

CAROLINA MOREIRA OLIVEIRA

**SEQUENCIAMENTO INTELIGENTE E ADAPTATIVO DE
ENUNCIADOS EM PROGRAMAÇÃO DE
COMPUTADORES**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Andrey Ricardo Pimentel

Coorientador: Prof. Dr. Eleandro Maschio

CURITIBA

2016

CAROLINA MOREIRA OLIVEIRA

**SEQUENCIAMENTO INTELIGENTE E ADAPTATIVO DE
ENUNCIADOS EM PROGRAMAÇÃO DE
COMPUTADORES**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Andrey Ricardo Pimentel

Coorientador: Prof. Dr. Eleandro Maschio

CURITIBA

2016

O48

Oliveira, Carolina Moreira

Sequenciamento inteligente e adaptativo de enunciados em programação de computadores / Carolina Moreira Oliveira. – Curitiba, 2016.

120 f. : il. ; 30 cm.

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

Orientador: Andrey Ricardo Pimentel

Coorientador: Eleandro Maschio

Bibliografia: p. 98-103.

1. Programação (Computadores). 2. Sistemas tutoriais inteligentes. 3. Aprendizado do computador. I. Pimentel, Andrey Ricardo. Maschio, Eleandro . III. Universidade Federal do Paraná. IV. Título.

CDD: 006.31




MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor CIÊNCIAS EXATAS
Programa de Pós Graduação em INFORMÁTICA
Código CAPES: 40001016034P5

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **CAROLINA MOREIRA OLIVEIRA**, intitulada: "**Sequenciamento Inteligente e Adaptativo de Enunciados em programação de Computadores**", após terem inquirido a aluna e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO.

Curitiba, 12 de Setembro de 2016.


ANDREY RICARDO PIMENTEL
Presidente da Banca Examinadora (UFPR)


ALEXANDER ROBERT KUTZKE
Avaliador Externo (UFPR)


EDUARDO TODT
Avaliador Interno (UFPR)


ELEANDRO MASCHIO KRYNSKI
Coorientador - Avaliador Externo (UFPR)



A eles, que antes de mim acreditaram.

AGRADECIMENTOS

A Deus e à Nossa Senhora Aparecida.

Aos meus pais, Murilo e Valéria, que sempre apoiaram as minhas decisões. Essa realização é nossa!

Ao meu orientador, professor Andrey Ricardo Pimentel, pela confiança e, principalmente, por acreditar no nosso trabalho em todos os momentos.

Ao professor Alexandre Ibrahim Direne, pelas longas conversas e pelo auxílio na pesquisa.

Ao professor Eleandro Maschio, por ceder horas de seu precioso tempo. Serei eternamente grata!

Aos professores Eduardo Todt e Alexander Kutzke, integrantes da banca, pelas valiosas sugestões para aprimorar a pesquisa.

Ao corpo docente da UFPR, que transmitiu seus ensinamentos com sabedoria.

Aos meus queridos amigos da pós-graduação, Ernani Gottardo, José Antonio Buiar e Simone Dominico. Agradeço também a todos os meus colegas de laboratório.

Aos professores da UTFPR, câmpus Guarapuava. Especialmente aos professores Diego Marczal, Renata Stange e Roni Banaszewski, pela contribuição para a pesquisa. Também, ao professor Alex Sandro de Castilho, pela ajuda com a fundamentação matemática.

À minha família, que possibilitou que eu permanecesse em Curitiba. Principalmente à minha avó Erondina, que me deu suporte em todos os momentos.

À minha amiga Maisa, que me cedeu o seu escritório para que eu pudesse trabalhar tranquila. Obrigada pelo seu imenso coração.

Às minhas amigas Maisa, Vanessa, Larissa, Patrícia e Priscila, que sempre me alegraram em dias de angustia. Estamos sempre juntinho!

Ao meu querido amigo Vinícius, por me apoiar e estar presente há tanto tempo.

Aos meus queridos alunos, que me fazem evoluir constantemente.

A todos os meus familiares e amigos, que sabem o quanto sou grata por toda a ajuda recebida.

*“Não deve haver limites para o esforço humano.
Somos todos diferentes.
Por pior do que a vida possa parecer,
sempre há algo que podemos fazer em que podemos obter sucesso.
Enquanto houver vida, haverá esperança.”*

Stephen Hawking

SUMÁRIO

| | |
|--|-------------|
| LISTA DE SIGLAS E ACRÔNIMOS | viii |
| LISTA DE FIGURAS | xi |
| RESUMO | xii |
| ABSTRACT | xiii |
| 1 INTRODUÇÃO | 1 |
| 1.1 Problema Central | 1 |
| 1.2 Objetivo | 2 |
| 1.3 Justificativa | 3 |
| 1.4 Estrutura da Dissertação | 5 |
| 2 RESENHA LITERÁRIA | 7 |
| 2.1 Múltiplas Representações Externas | 7 |
| 2.1.1 Taxonomia Funcionalista | 11 |
| 2.2 Modelo de Sobreposição | 13 |
| 2.3 Grafos Genéticos | 16 |
| 2.3.1 Conceitos e Terminologia | 17 |
| 2.3.2 Contribuições para a Tutoria Inteligente | 19 |
| 2.3.2.1 Base para a Modelagem | 19 |
| 2.3.2.2 Base para a Tutoria | 20 |
| 2.3.2.3 Base para a Aprendizagem | 21 |
| 2.4 Aquisição de Conhecimento em Programação de Computadores | 22 |
| 2.5 Sequenciamento de Enunciados | 25 |
| 2.5.1 JV ² M | 28 |
| 2.5.2 SQL Tutor | 30 |

| | | |
|----------|---|-----------|
| 2.5.3 | PHP Intelligent Tutoring System | 33 |
| 2.5.4 | Duolingo | 35 |
| 2.5.5 | ADAPTFARMA | 45 |
| 2.5.6 | Khan Academy | 48 |
| 3 | FORMALISMOS ADOTADOS NA SOLUÇÃO DO PROBLEMA | 55 |
| 3.1 | Sequenciamento Automático, Adaptativo e Inteligente de Enunciados . . . | 55 |
| 3.1.1 | Sequenciamento Automático Simples | 55 |
| 3.1.2 | Sequenciamento Adaptativo | 56 |
| 3.1.3 | Sequenciamento Inteligente | 56 |
| 3.1.4 | Discussão | 56 |
| 3.2 | Metodologia para Sequenciamento de Enunciados | 57 |
| 3.2.1 | Modelagem do Aprendiz | 60 |
| 3.2.2 | Sugestão do Tópico Tutorado/Exercitado | 61 |
| 3.2.2.1 | Priorização Absoluta da Perícia Inicial | 62 |
| 3.2.2.2 | Perícias Insuficientemente Desenvolvidas | 63 |
| 3.2.2.3 | Perícias Localizadas na Fronteira de Conhecimento do Aprendiz | 64 |
| 3.2.2.4 | Distância até a Perícia Inicial | 64 |
| 3.2.2.5 | Estratégias de Tutoria Bem-sucedidas | 65 |
| 3.2.2.6 | Navegabilidade Proporcionada ao Aprendiz | 68 |
| 3.2.3 | Sequenciamento de Enunciados | 68 |
| 3.2.4 | Atualização do Modelo do Aprendiz | 70 |
| 3.2.5 | Acompanhamento do Processo de Aprendizagem | 72 |
| 3.2.6 | Implementação de um Protótipo de Ferramenta | 75 |
| 4 | DINÂMICA DE FUNCIONAMENTO DA SOLUÇÃO PROPOSTA | 77 |
| 4.1 | Primeiro Ciclo de Sequenciamento | 77 |
| 4.2 | Segundo Ciclo de Sequenciamento | 81 |
| 4.3 | Terceiro Ciclo de Sequenciamento | 83 |

| | |
|--|------------|
| 5 ESTUDO EMPÍRICO | 85 |
| 5.1 Sujeitos | 85 |
| 5.2 Instrumentos | 85 |
| 5.2.1 Questionário | 86 |
| 5.3 Procedimentos | 86 |
| 5.3.1 Questão 1 | 86 |
| 5.3.2 Questão 2 | 88 |
| 5.3.3 Questão 3 | 88 |
| 5.4 Resultados Coletados | 88 |
| 5.4.1 Questão 1 | 88 |
| 5.4.2 Questão 2 | 91 |
| 5.4.3 Questão 3 | 91 |
| 5.5 Discussão dos Resultados | 92 |
| 6 RESULTADOS | 93 |
| 7 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS | 95 |
| 7.1 Trabalhos Futuros | 96 |
| REFERÊNCIAS | 98 |
| A SUBCONJUNTO DE PERÍCIAS NO DOMÍNIO DE PROGRAMAÇÃO DE COMPUTADORES | 104 |
| B CATÁLOGO DE ENUNCIADOS | 115 |
| C ENUNCIADOS COMPLEMENTARES | 120 |

LISTA DE SIGLAS E ACRÔNIMOS

| | |
|--------------|---|
| CBT | <i>Case-Based Teaching</i> (Ensino Baseado em Casos) |
| CBR | <i>Case-Based Reasoning</i> (Raciocínio Baseado em Casos) |
| CE | Complexidade Estrutural |
| DI | Detalhes de Implementação |
| FARMA | Ferramenta de Autoria para a Remediação de Erros com Mobilidade na Aprendizagem |
| IA | Inteligência Artificial |
| IAC | Instrução Assistida por Computador |
| IHC | Interação Humano-Computador |
| IIAC | Instrução Inteligente Assistida por Computador |
| ITS | <i>Intelligent Tutoring Systems</i> (Sistemas Tutores Inteligentes) |
| LOC | <i>Lines of Code</i> (Linhas de Código) |
| MRE | Múltiplas Representações Externas |
| MS | Método de Sequenciamento |
| MSA | Método de Sequenciamento Adaptativo |
| MSD | Método de Sequenciamento Ordenado por Grau de Dificuldade |

| | |
|------------|---|
| MSP | Método de Sequenciamento Definido pelo Professor |
| MSR | Método de Sequenciamento Randômico |
| RE | Representação Externa |
| STI | Sistemas Tutores Inteligentes |
| XML | <i>Extensible Markup Language</i> (Linguagem de Marcação Extensível) |
| XP | Pontos de Experiência |

LISTA DE FIGURAS

| | | |
|------|--|----|
| 2.1 | Taxonomia funcionalista de MREs [24] | 11 |
| 2.2 | Exemplo de grafo genético contextualizado no domínio de Programação de Computadores [25] | 18 |
| 2.3 | JV ² M [15] | 28 |
| 2.4 | SQL-Tutor: Escolha da cláusula a ser exercitada [31] | 31 |
| 2.5 | SQL-Tutor: Escolha do exercício a ser resolvido [31] | 33 |
| 2.6 | PHP ITS: Escolha do exercício a ser resolvido [50] | 34 |
| 2.7 | Duolingo: estruturação das unidades de conteúdo em árvore | 36 |
| 2.8 | Duolingo: lições de uma unidade | 36 |
| 2.9 | Duolingo: progresso do aprendiz pelas unidades | 37 |
| 2.10 | Duolingo: percentual de fluência no idioma | 37 |
| 2.11 | Duolingo: aprendizado de uma palavra nova por meio de imagens | 38 |
| 2.12 | Duolingo: aprendizado de palavra pela indicação em um texto para traduzir do idioma ensinado | 39 |
| 2.13 | Duolingo: tradução para o idioma ensinado | 39 |
| 2.14 | Duolingo: seleção da palavra que falta em uma frase no idioma ensinado | 39 |
| 2.15 | Duolingo: ditado de uma frase | 39 |
| 2.16 | Duolingo: pronúncia de uma frase | 40 |
| 2.17 | Duolingo: tradução oral para o idioma ensinado | 40 |
| 2.18 | Duolingo: escolha da(s) tradução(ões) corretas | 40 |
| 2.19 | Duolingo: pergunta de múltipla escolha (aplicativo para celular) | 41 |
| 2.20 | Duolingo: associação de palavras e traduções (aplicativo para celular) | 41 |
| 2.21 | Duolingo: <i>personal trainer</i> | 43 |
| 2.22 | Duolingo para as Escolas: painel centralizado de acompanhamento | 44 |
| 2.23 | Khan Academy: guia passo a passo | 50 |
| 2.24 | Khan Academy: desafio de programação passo a passo | 50 |

| | | |
|------|--|----|
| 2.25 | Khan Academy: projeto | 51 |
| 2.26 | Khan Academy: estruturação do conteúdo | 52 |
| 2.27 | Khan Academy: painel de aprendizagem (<i>learning dashboard</i>) | 52 |
| 2.28 | Khan Academy: medalhas | 53 |
| 2.29 | Khan Academy: relatório de progresso de habilidades do estudante (painel do tutor) | 53 |
| 2.30 | Khan Academy: relatório de progresso de habilidades da classe toda (painel do tutor) | 54 |
| 3.1 | Ferramenta para a descrição do conhecimento do domínio [26] | 58 |
| 3.2 | Ferramenta para a elicitación e catalogação de enunciados [26] | 59 |
| 3.3 | Aspecto de sobreposição do conhecimento do aprendiz frente ao conhecimento do domínio [25] | 60 |
| 3.4 | Indicação de regiões do grafo a serem exploradas [25] | 62 |
| 3.5 | Sobreposição de um enunciado frente ao conhecimento do domínio [25] | 69 |
| 3.6 | Catálogo de enunciados conforme o gabarito do conhecimento do domínio [25] | 69 |
| 3.7 | Atualização do Modelo do Aprendiz [25] | 71 |
| 3.8 | Sintetização do Modelo do Aprendiz de um grupo [25] | 74 |
| 3.9 | Progresso do aprendiz (ou de um grupo) ao longo do tempo [25] | 74 |
| 4.1 | Situar aprendiz (Primeiro Ciclo de Sequenciamento) | 78 |
| 4.2 | Sugerir perícia (Primeiro Ciclo de Sequenciamento) | 78 |
| 4.3 | Sugerir enunciado (Primeiro Ciclo de Sequenciamento) | 79 |
| 4.4 | Avaliar solução (Primeiro Ciclo de Sequenciamento) | 80 |
| 4.5 | Atualizar Modelo do Aprendiz (Primeiro Ciclo de Sequenciamento) | 80 |
| 4.6 | Situar aprendiz (Segundo Ciclo de Sequenciamento) | 81 |
| 4.7 | Situar aprendiz (Terceiro Ciclo de Sequenciamento) | 83 |
| 5.1 | Questão 3 | 87 |

RESUMO

A presente dissertação detalha uma abordagem para tratar do sequenciamento inteligente e adaptativo de enunciados, considerando a modelagem dinâmica do aprendiz, no domínio de Programação de Computadores. A metodologia utiliza grafos genéticos como base para a representação interna. Com isso, aspectos de sobreposição destacam o progresso do aprendiz frente ao conhecimento do domínio, além da contribuição individual de cada enunciado. Propõe-se, então, um processo de ordenação heurística que indica capacidades a serem desenvolvidas pelo aprendiz, bem como o sequenciamento adaptativo de enunciados que contribuam nessa evolução. A atualização do modelo e a extração de informações úteis sobre a aprendizagem, por fim, são analisadas. A implementação de um protótipo de ferramenta suporta a abordagem descrita.

Palavras-chave: sequenciamento de enunciados, ensino de programação de computadores, representação de conhecimento, sistemas tutores inteligentes, múltiplas representações externas.

ABSTRACT

This dissertation details an approach to deal with intelligent and adaptive exercise ordering, considering the student dynamic modeling in Computer Programming domain. The methodology uses genetic graphs as a basis for the internal representation. Thus, overlaying aspects highlight the progress of the learner compared to the domain knowledge, and to the individual contribution of each exercise. Then is proposed a heuristic ordering process that indicates capabilities to be developed by the student, as also an adaptive exercise ordering that contributes in this evolution. The model update and the extraction of useful information about learning, lastly, are analyzed. An implemented software tool prototype supports the approach here described.

Keywords: exercise ordering, teaching of computer programming, knowledge representation, intelligent tutoring systems, multiple external representations.

CAPÍTULO 1

INTRODUÇÃO

1.1 Problema Central

O ensino em Programação de Computadores tem sido um desafio por décadas [32]. Embora se trate de uma disciplina fundamental para qualquer curso da área de Ciência da Computação, o aprendizado dela tem sido bastante difícil para muitos iniciantes. Isso se corrobora pelo alto índice de desistência dessas disciplinas e, até mesmo, de evasão dos cursos [28]. Assim, há a necessidade de encontrar meios que tornem o assunto menos impactante para aqueles que iniciam na área.

Além disso, as pessoas que mostram interesse em aprender a programar são bastante diversas. Elas se distinguem em aspectos como idade, sexo, nível de escolaridade, maneira como aprendem e aptidão para resolver problemas lógicos. Como consequência, é extremamente difícil criar um material único que atenda a toda essa diversidade de perfis.

Dentre os vários métodos sugeridos para melhorar o processo de ensino, a adoção de aulas individuais tem se provado eficaz [50]. No entanto, aulas individuais com instrutores humanos exigem proporcionalmente mais recursos e, logo, tornam-se impraticáveis de serem aplicadas em contextos com muitos aprendizes, como no ensino superior.

A criação e difusão dos Sistemas Tutores Inteligentes (STIs) se deu como alternativa viável em resposta a essas necessidades [49, 33, 45]. Os STIs são, portanto, programas de computador projetados para incorporar técnicas de Inteligência Artificial (IA) de modo que saibam como ensinar, o que ensinar e a quem ensinar [33]. Essa abordagem tenta se aproximar das aulas individuais, considerando as características particulares do aprendiz para alterar a interação.

Nesse sentido, a modelagem do aprendiz responsabiliza-se por armazenar a representação atual do conhecimento e do desempenho do indivíduo sobre os tópicos tutorados. A partir dela, o sistema consegue definir o nível de dificuldade relativo de um conteúdo

frente ao conhecimento do aprendiz, como também propor sugestões personalizadas. Em mais alto nível, a inteligência de um sistema pode conseguir reconhecer as dificuldades de um aprendiz e prover diagnóstico para um tutoramento efetivo [11].

Paralelamente, qualquer ambiente de ensino que utilize enunciados precisa, de alguma maneira, determinar a sequência em que eles serão oferecidos ao aprendiz. Alguns sistemas apresentam os enunciados em ordem predefinida [47]. Outros, permitem que os aprendizes selecionem o próximo enunciado a partir de uma lista estática de alternativas [48]. Em ambas as situações, a escolha do enunciado não considera as capacidades atuais do aprendiz.

Levando em consideração a modelagem do aprendiz, um STI tem a capacidade de sugerir enunciados próprios ao nível de conhecimento do indivíduo. Isso se mostra positivo, pois enunciados com dificuldade muito abaixo desse nível podem causar entediamento, e do contrário, sendo muito acima, provocar desmotivação [38, 10]. Ambos os casos são passíveis do abandono da atividade proposta.

Dessa forma, o sequenciamento adaptativo de enunciados busca auxiliar diferentes perfis de aprendizes, independentemente do nível de conhecimento. Pode-se diversificar a ordem e mesmo a quantidade de enunciados com o objetivo de potencializar o processo de ensino-aprendizagem para cada aprendiz em particular.

1.2 Objetivo

O objetivo geral desta pesquisa foi propor uma abordagem para o sequenciamento adaptativo de enunciados, frente a uma modelagem dinâmica do aprendiz, no contexto de Programação de Computadores.

Constituem-se como objetivos específicos:

- Implementar a modelagem dinâmica do aprendiz diante das capacidades do domínio;
- Prover um processo de ordenação heurística que oriente conhecimentos a serem explorados pelo aprendiz e que, igualmente, seja utilizado no sequenciamento adaptativo de enunciados;

- Fornecer técnicas que promovam o sequenciamento adaptativo e inteligente de enunciados ao aprendiz;
- Pesquisar e indicar alternativas para que ocorra a atualização do Modelo do Aprendiz conforme a avaliação de desempenho nas soluções dos enunciados sugeridos;
- Analisar a possibilidade de extrair dos modelos informações úteis à didática do instrutor, tais como:
 - A sintetização do Modelo do Aprendiz de um conjunto (por exemplo, uma sala de aula) para que se observe como as capacidades são ensinadas por aquele instrutor;
 - A apresentação de um histórico do desenvolvimento do aprendiz (ou de um conjunto deles) ao longo do tempo;
- Disseminar o conhecimento alcançado, por meio de publicações da área.

1.3 Justificativa

Na literatura, conforme será detalhado no Capítulo 2, são esparsas as pesquisas sobre sequenciamento automático de enunciados em ambientes de ensino. Esses estudos tornam-se ainda mais reduzidos quando se considera o sequenciamento adaptativo no nicho de Sistemas Tutores Inteligentes. Trata-se de uma abordagem promissora, pela tentativa de proporcionar uma sucessão personalizada de enunciados ao aprendiz. Faz-se isso considerando características particulares, como também relacionadas ao nível de conhecimento atual do indivíduo.

Deste modo, a adaptação inteligente de sequências de enunciados abre espaço para algumas possibilidades, diante das seguintes perspectivas:

- Do aprendiz:
 - potencializar o ensino pela estratégia de prover enunciados com dificuldade pouco acima do nível de conhecimento do aprendiz;

- permitir que se retrocedam níveis de dificuldade dos enunciados a fim de reforçar conteúdos nos quais o aprendiz possa apresentar dúvidas;
 - sugerir enunciados alinhados às características cognitivas do aprendiz, favorecendo a maneira com que o indivíduo adquire conhecimento;
 - considerar preferências particulares que podem influenciar na aprendizagem;
 - auxiliar em aspectos motivacionais do aprendiz, pela adaptabilidade da sucessão de enunciados ao seu perfil;
 - influenciar positivamente no tempo de estudo extraclasse;
- Do instrutor:
 - oferecer aos seus aprendizes sequências de enunciados individualmente personalizadas sem a necessidade de retrabalho;
 - abordar eficientemente grupos grandes e heterogêneos de aprendizes;
 - observar eventuais dificuldades do grupo frente aos conteúdos propostos e seus enunciados correspondentes;
 - identificar a demanda de enunciados diante de frações descobertas do conteúdo;
 - favorecer a indicação de enunciados entre instrutores;
 - reduzir o tempo necessário para a seleção de enunciados;
 - Da autoria:
 - eliminar a necessidade do autor especificar sequências de enunciados;
 - dinamizar o processo de autoria, aumentando a granularidade das contribuições, pois se favorece que os ambientes de ensino incorporem enunciados individuais ao invés de sequências inteiras;
 - concentrar o esforço de autoria em fornecer oportunidades para o desenvolvimento de conhecimentos mais específicos;
 - reduzir o tempo necessário ao processo de autoria;

- favorecer o intercâmbio de enunciados entre instrutores/atores;
- fomentar repositórios colaborativos de enunciados.

Diante dessas possibilidades, no ensino Programação de Computadores, um estudo realizado no final da década de 90 [38] descreveu medidas cognitivas como uma maneira de obter metaconhecimento sobre enunciados. Foi desenvolvida a ferramenta de autoria Sequence, que sugere uma sequência inicial de enunciados e, observando um conjunto reduzido de capacidades, consegue adaptá-la ao nível de conhecimento atual do aprendiz.

Recentemente, outra pesquisa [25] promoveu a continuidade desses estudos. Primeiro, atuando na revisão do conjunto de capacidades identificado por [38]. Depois, definindo um arcabouço conceitual para a modelagem do processo de aquisição de conhecimento apoiado por ambientes inteligentes. Os esforços de implementação da pesquisa se concentraram na descrição do conhecimento do domínio de Programação de Computadores, bem como na consequente catalogação e elicitación de enunciados diante desse conhecimento.

Todavia, a pesquisa não se estendeu ao estágio de sequenciar os enunciados de maneira adaptativa frente a uma modelagem dinâmica do aprendiz. Com isso, observa-se um espaço de contribuição que pode se favorecer de uma potencialidade pouco explorada dos estudos de [25].

Por fim, justifica-se que, embora as pesquisas em tutoria inteligente tenham crescido nas últimas décadas, não se trata de um conceito amplamente difundido entre educadores [50]. Uma das principais razões para isso é que apesar de muitos STIs terem sido construídos, poucos deles são usados em situações reais de ensino, restringindo-se ao caráter experimental. Toma-se esse fato como indício de que existe espaço significativo para melhorias na área de STIs.

1.4 Estrutura da Dissertação

O Capítulo 2 concentra o referencial teórico desta dissertação, compreendendo: múltiplas representações externas, modelo de sobreposição, grafos genéticos, aquisição de conhecimento em Programação de Computadores e sequenciamento de enunciados. Depois, o

Capítulo 3 apresenta os formalismos adotados na solução do problema. Descreve-se, no Capítulo 4, a dinâmica de funcionamento da solução proposta. O Capítulo 5, em seguida, contém o estudo empírico realizado com o protótipo que suporta a abordagem descrita. No Capítulo 6, discutem-se os resultados alcançados pela pesquisa. O Capítulo 7, por sua vez, traz as considerações finais e a perspectiva de trabalhos futuros. Os anexos encerram o documento.

CAPÍTULO 2

RESENHA LITERÁRIA

O referencial teórico e as pesquisas correlatas desta dissertação fundamentam-se em cinco temas principais, cujo detalhamento compete ao presente capítulo: Múltiplas Representações Externas (2.1), Modelo de Sobreposição (2.2), Grafos Genéticos (2.3), aquisição de conhecimento em Programação de Computadores (2.4) e sequenciamento de enunciados (2.5).

2.1 Múltiplas Representações Externas

O termo Representação Externa (RE) diz respeito ao uso de técnicas para representar, organizar e apresentar conhecimento [8]. O conceito abrange desde modelos puramente proposicionais/sentenciais até modelos gráficos/diagramáticos, passando por representações que associam elementos gráficos e textuais (como as tabelas). São exemplos de RE [8]: sentenças em linguagem natural, sentenças em linguagens formais (e.g. lógica de primeira ordem), listas, tabelas, grafos, mapas, projetos e diagramas. Adicionando-se recursos como multimídia e interação, os exemplos passam a incluir animações e, até mesmo, as realidades aumentada e virtual.

Conforme a definição de [35], qualquer RE deve ser descrita em termos:

1. do mundo representado;
2. do mundo representante;
3. dos aspectos do mundo representado que estão sendo expressos;
4. dos aspectos do mundo representante que compõem a modelagem;
5. da correspondência entre esses dois mundos.

Portanto, pode-se evidenciar que inúmeras atividades cotidianas compreendem o uso de REs, mesmo a elaboração de uma simples lista de compras. Como outros exemplos, nesse sentido, citam-se: fichas de cadastro, receitas, tabelas nutricionais, esboços de reformas, agendas de compromissos, placas em aeroportos, trajetos rodoviários, entre outros [8].

Em decorrência do suporte que as REs trazem ao dia-a-dia, elas passaram a ser empregadas na resolução de problemas mais formais, prestando-se como eficiente auxílio cognitivo em várias áreas do conhecimento. Seguem alguns exemplos [24]:

1. Arquitetura e *design*: esboços (croquis), projetos 2D e 3D, passeios virtuais em vídeo ou navegáveis;
2. Raciocínio lógico e analítico: abstração e estruturação visual de problemas, re-representação;
3. Classificação de informações hierárquicas: organogramas, árvores de classificação, árvores e tabelas de decisão, regras e sentenças *if-then*;
4. Matemática: notação matemática, equações, plano cartesiano, diagramas de Venn, grafos;
5. Física: diagramas que envolvem entidades concretas e abstratas (como velocidade, aceleração, força, movimento, calor, correntes elétricas, ondas);
6. Química: tabela periódica, modelo atômico de Bohr, fórmulas estruturais;
7. Finanças: extratos bancários, balancetes contábeis, holerites, planilhas, gráficos setoriais;
8. Música: partituras, tablaturas, cifras.

Embora a característica mais notável das REs seja facilitar o processo de memorização, reduzindo a carga cognitiva necessária para realizar uma tarefa, os seguintes benefícios ainda foram observados [52]:

1. Prover informações possíveis de serem diretamente percebidas e utilizadas sem nenhuma interpretação ou formulação explícita;
2. Fixar e estruturar comportamento cognitivo, porque a estrutura física das REs restringe a possibilidade de comportamentos (alguns são permitidos e outros proibidos);
3. Modificar a natureza das tarefas em questão, tornando-as potencialmente mais fáceis de serem realizadas.

Estudos posteriores incidiram sobre REs dinâmicas, bem como o compartilhamento de REs construídas por aprendizes. As primeiras têm se mostrado úteis pela facilidade de apresentarem explicitamente a atividade de um sistema ou processo, admitindo, inclusive, interação [5]. As experiências com o segundo recurso, por sua vez, sustentam que há benefícios quando os aprendizes constroem (ou até criam), compartilham e avaliam as representações uns dos outros [16]. Trata-se de uma abordagem mais ativa, ou mesmo proativa, se comparada à simples exposição de REs por parte do instrutor.

Além disso, a pesquisa de [2] traz evidências que enfatizam as vantagens do emprego de REs como auxílio ao ensino e à resolução de problemas, elevando-se o desempenho do aprendiz quando utilizadas apropriadamente na interação. Essas REs são consideradas de natureza tipicamente metacognitiva, por promoverem que o aprendiz desenvolva seu conhecimento sobre os próprios processos cognitivos, as formas de operá-los e a capacidade de controlá-los [7]. Ambientes de ensino-aprendizagem têm se beneficiado, justamente, dessas características para alcançarem melhores resultados.

Nas duas últimas décadas, pesquisas têm sido focadas na combinação de diferentes REs como apoio a situações de aprendizado. O conjunto resultante dessa combinação passa a ser denominado e constituído por Múltiplas Representações Externas (MREs) [2].

Desde então, os estudos têm indagado acerca dos benefícios, como também das possíveis desvantagens, decorrentes do uso simultâneo de REs com formatos diferentes na aprendizagem [8]. Um conjunto multirrepresentacional tende a fornecer suporte às necessidades cognitivas diversas de um grupo heterogêneo de aprendizes. Em síntese, isso ocorre principalmente por se representar uma mesma informação de maneiras diferentes.

Por consequência, dá-se espaço a novas ideias e estratégias, fazendo com que tanto a aprendizagem quanto a resolução de problemas sejam mais flexíveis e intuitivos. Por meio da combinação das propriedades de cada RE, espera-se que o aprendiz se prevaleça da soma das benefícios envolvidos.

Adicionalmente, foi evidenciado que tanto a construção quanto a observação de MREs trazem benefícios ao aprendiz [8]. E, nesse mesmo sentido, a habilidade de transitar entre essas representações proporciona um melhor entendimento sobre o conteúdo abordado, por conta das múltiplas perspectivas trazidas [24]. Por tudo isso, [19] destaca que as MREs podem conseqüentemente trazer efeitos positivos na motivação e no interesse do aprendiz.

Em contraste, nem sempre todas essas vantagens são atingidas facilmente. Primeiro, pela dificuldade de se alcançar generalizações sobre o emprego de MREs. Depois, porque a própria composição de um conjunto de representações, em determinado contexto, não consiste em uma tarefa trivial [4, 3] e deve considerar:

- o número de representações;
- a informação representada;
- a forma (texto, som, imagem, híbridos, entre outros), tanto individual quanto do conjunto;
- a sequência das representações; e
- a tradução entre elas.

Mais recentemente, as pesquisas de [19] foram retomadas pelos próprios autores, no intuito de delinear um panorama do uso de REs na resolução de problemas, considerando a lacuna de vinte anos entre ambas as publicações [9]. Os principais progressos citados foram: a adaptatividade das (múltiplas) representações ao perfil do aprendiz; a mineração de dados para que isso seja possível; bem como o incremento da largura de banda e a evolução da *web* (em padronizações e implementações) como tecnologias de suporte.

Como tendências para futuros avanços significativos, a mesma pesquisa prospectou: o fortalecimento tanto teórico, quanto em termos de evidências práticas, dos resultados para otimizar o uso de REs nas situações de aprendizado; a mineração de dados em larga escala, a fim de que o aprendiz se valha do conhecimento coletivo; a exploração de tecnologias interativas contemporâneas, como telas sensíveis ao toque e sensores de movimento, para construir REs.

2.1.1 Taxonomia Funcionalista

Conforme [2, 24], as MREs exercem três principais funções, em situações de aprendizado: complementar, restringir e construir. Uma mesma RE pode desempenhar mais de uma função (ou papel) em determinada situação de aprendizado. Essas funções consideram os recursos semânticos oferecidos pelas REs.

Nesse sentido, [2] define uma taxonomia para essas funções, apresentada na Figura 2.1 e detalhada na sequência:

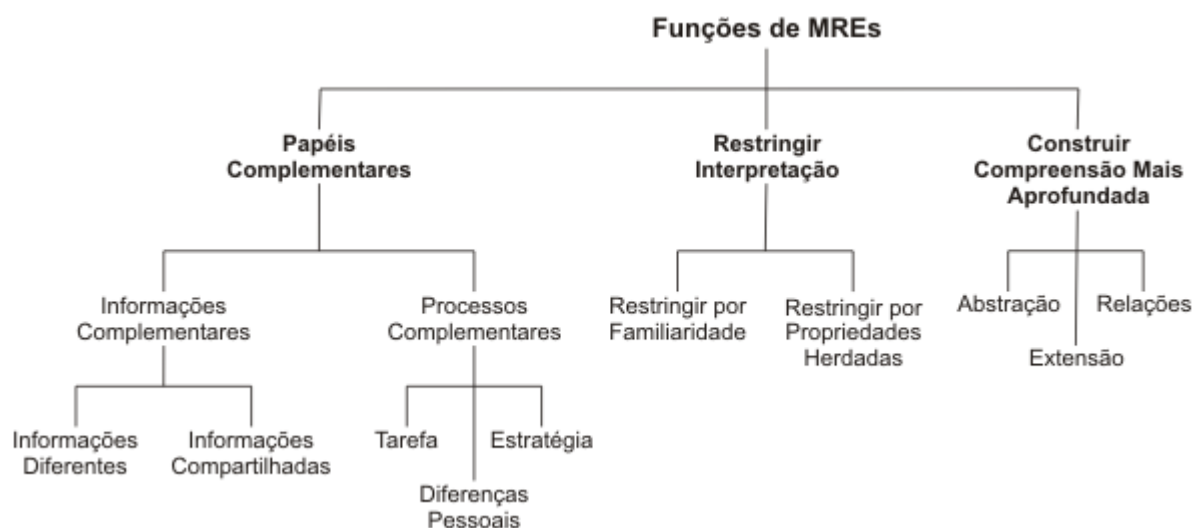


Figura 2.1: Taxonomia funcionalista de MREs [24]

1. **Papéis complementares:** quando duas ou mais REs do conjunto diferem, ou na informação que expressam ou no processo que suportam. Divide-se em:
 - (a) Suporte a informações complementares: considera as diferenças na informação

trazida individualmente por cada RE. Pode ocorrer quando uma única RE não consegue expressar toda a informação necessária, ou em situações que se teria sobrecarga com a tentativa de incluir tal conteúdo em uma RE. São subclasses:

- i. Informações diferentes: ambas as REs modelam aspectos únicos do domínio e trazem informações exclusivas;
- ii. Informações complementares: existe interseção entre as informações modeladas pelas REs do conjunto.

Exemplos: ligações químicas explícitas em uma fórmula estrutural completa, interface de um simulador de direção veicular, realce sintático de um código;

- (b) Suporte a processos complementares: indica que processos e inferências complementares podem ser alcançados pelo uso de MREs, mesmo que elas contenham informações equivalentes. Relacionam-se a:
 - i. Diferenças pessoais: possibilidade de um aprendiz escolher, conforme critérios particulares, a RE mais adequada;
 - ii. Compatibilidade com a tarefa: alinhamento entre a estrutura da informação requerida pelo problema e o formato da RE utilizada;
 - iii. Interação com estratégias: aderência das REs ao uso de estratégias mais (ou menos) efetivas para resolver determinado problema.

Exemplos: diagramas de Veen nas operações entre conjuntos, tabelas para expressar a relação entre duas variáveis (como cidade e distância), simulação de um programa sendo executado;

2. **Restringir interpretação:** quando uma RE é usada com o objetivo de restringir a interpretação de outra. As restrições podem ocorrer por:

- (a) Familiaridade: emprego de uma RE mais familiar para auxiliar na interpretação de outra menos familiar. Exemplos: os desenhos de chocolate em barra como metáfora à notação matemática de fração, a tablatura como facilitadora para o ensino da partitura, uma receita culinária sendo usada para introduzir o conceito de algoritmo;

- (b) Propriedades herdadas: uso de uma RE mais específica para limitar, ou até forçar, a interpretação de outra mais ambígua. Exemplos: um balancete contábil evidenciando os créditos e os débitos, a sintetização de dados em um gráfico setorial, um trecho de código com uma estrutura condicional sendo esclarecido por um fluxograma.
3. **Construir compreensão mais aprofundada:** quando a informação trazida pelas MREs proporciona uma compreensão que seria difícil de ser atingida por meio de uma única RE. Pode acontecer por:
- (a) Abstração: processo que cria entidades mentais (referências) para sustentar conceitos que possuem nível mais alto de organização. Exemplos: a tabela periódica dos elementos químicos, o esboço visual de um problema (como a Torre de Hanói), a visualização dos valores atribuídos às variáveis de um programa;
- (b) Extensão: ato de estender o conhecimento anterior para novas situações sem que, necessariamente, se reconheça a natureza desse conhecimento. Exemplos: uma criança aprendendo a somar números reais (conta de adição com vírgula) depois dos inteiros, a compreensão do sistema cartesiano com uma terceira coordenada, o entendimento de uma matriz bidimensional a partir de uma unidimensional;
- (c) Relações: concerne à compreensão relacional, ou seja, o processo de associar duas ou mais REs sem que haja reorganização de conhecimento. Exemplos: a correspondência de palavras entre textos traduzidos, a construção de um gráfico dada uma equação, o realce da linha de código que está sendo executada em uma depuração.

2.2 Modelo de Sobreposição

Diante da literatura da área [33, 49], observou-se que, até o início da década de 90, inexistiam padrões quanto às arquiteturas que baseavam os Sistema Tutores Inteligentes. Assim,

ocorriam variações tanto de quais eram os módulos componentes quanto da organização desses elementos.

Entretanto, diante do que se tinha implementado na época, foi reconhecido que uma arquitetura básica seria composta dos três primeiros componentes detalhados na sequência. Pesquisas consequentes logo identificaram e adicionaram um quarto e último componente. Dessa forma, embora não haja arquitetura única para a construção de um STI, uma abordagem clássica sugere os seguintes módulos:

1. **Modelo do Domínio (ou do Especialista):** faz uma representação do conhecimento de um especialista, por meio de objetos, regras ou procedimentos;
2. **Modelo do Aprendiz:** remete à competência do aprendiz no conteúdo tutorado, considerando o Modelo do Domínio, bem como suas preferências relativas ao processo de aquisição de conhecimento;
3. **Modelo Pedagógico (ou Didático-Pedagógico):** contém as estratégias de ensino que, mediante interação com os modelos anteriores, possibilitam que a apresentação do conteúdo seja correspondente ao progresso e às preferências de cada aprendiz; e
4. **Módulo de Interface:** responsabiliza-se pela interação entre o aprendiz (e outros usuários, como autores, tutores humanos, monitores de turma) e o sistema.

O Modelo do Aprendiz, portanto, contém uma representação do estado atual de conhecimento e do desempenho do aprendiz sobre o domínio ensinado [45]. O modelo é atualizado de forma dinâmica pelo sistema e representa informações de aprendizes em particular. Todavia, o modelo pode ser adaptado a necessidades específicas, representando informações coletivas ao invés de individuais, ou ainda por tempo fixado em detrimento de um histórico completo.

Seria ideal que o modelo armazenasse todos os aspectos cognitivos e comportamentais que pudessem influenciar no aprendizado. Contudo, na atualidade, algo tão completo seria praticamente impossível pela maneira convencional de comunicação entre o aprendiz e os

sistemas tutores. Estes, diferentes de tutores humanos, ignoram dados como expressões faciais, alterações na voz e fatores psicológicos como motivação ou tédio, entre outros passíveis de conter informações cruciais ao aprendizado.

Assim, o Modelo do Aprendiz pode variar na maneira com que representa internamente as informações obtidas. Na sequência, são resumidos os quatro principais modelos cognitivos para essa finalidade, conforme [13, 40]:

1. **Modelo de Medição de Performance:** procura descrever o aprendiz em relação ao domínio de conhecimento com base na medição de sua performance durante a resolução de problemas da área. Trata-se de uma abordagem simplificada, uma vez que se compara o desempenho (não o conhecimento) do aprendiz com o de um perito. A medição pode ser realizada de formas variadas;
2. **Modelo de Sobreposição (*Overlay*):** representa o conhecimento do aprendiz como um subconjunto do conhecimento do domínio. A sobreposição desses conjuntos evidencia os conhecimentos que devem ser apropriados pelo aprendiz. Essa abordagem implica que a representação de ambos os modelos (do domínio e do aprendiz) sejam comparáveis;
3. **Modelo de Perturbação ou de Descrição de Erros (*Buggy*):** assume que os erros do aprendiz podem provir tanto da ausência de conceitos quanto de concepções errôneas adquiridas no processo de ensino. A abordagem também considera o Modelo do Aprendiz como um subconjunto do conhecimento do domínio, entretanto enfoca a ausência de conceitos e as concepções equivocadas, ao invés da proficiência no conhecimento;
4. **Modelo de Simulação:** consiste em um modelo (geralmente procedural) que simula o comportamento do aprendiz em um domínio de conhecimento específico. Esse comportamento atua como uma predição das ações que o aprendiz executaria para resolver um problema, alcançando desempenho semelhante ao que ele teria. Mostra-se uma abordagem útil para explicar os passos necessários para a resolução de um problema.

A corrente pesquisa concentra-se no Modelo de Sobreposição, uma vez que remete continuidade aos estudos de [25], embasados nessa abordagem de representação. A próxima seção apresenta os Grafos Genéticos e enfatiza a aderência deles com a abordagem do Modelo de Sobreposição.

2.3 Grafos Genéticos

Cabe a esta seção abordar os conceitos de grafo genético fundamentados por Ira P. Goldstein [14]. O grafo genético é, basicamente, “um modelo evolutivo para representação de conhecimento procedimental que serve como base para o desenvolvimento de tutores inteligentes” [25]. Nesse sentido, o termo “genético” provém do chamado Método Genético que consiste no estudo da gênese e desenvolvimento dos objetos de estudo. Assim, a modelagem pode ser vista como um exercício de Epistemologia Genética, pois se preocupa em explicar a ordem de sucessão em que as diferentes capacidades cognitivas se constroem, importando-se no estudo de como o conhecimento se origina e se desenvolve [37]. Destaca-se que não existe associação direta com o termo homônimo da Computação Natural ou Bioinspirada, referente à hereditariedade.

Os grafos genéticos foram introduzidos à comunidade científica em um contexto no qual se desenvolvia uma nova geração de sistemas de Instrução Assistida por Computador (IAC) cujo destaque era o emprego de técnicas da Inteligência Artificial. Essa nova geração trouxe o termo Instrução Inteligente Assistida por Computador (IIAC) que evoluiu para a denominação Sistema Tutor Inteligente.

Um avanço frente aos ambientes anteriores (baseados em *scripts*) foi a inclusão de um módulo que representasse o conhecimento do domínio. Contudo, as estratégias pedagógicas presentes nesses sistemas tutores ainda eram bastante rudimentares.

Ainda assim, já se começava a tratar, na época, o ensino sob a perspectiva de subconjuntos. Nela, o conhecimento do aprendiz é modelado como um subconjunto da perícia do domínio, que consiste em um conjunto de fatos e regras. A tutoria, portanto, busca estimular o crescimento do subconjunto, intervindo nas situações em que um fato ou regra faltantes são componentes críticos para se chegar à solução correta.

Mesmo que a concepção de subconjuntos fosse um resumo do processo de ensino, pesquisas já investiam em representar as perícias de um domínio. Todavia, a interpretação que se tinha na época não se preocupava em descrever como o novo conhecimento evoluía a partir do que se conhecia anteriormente. Goldstein criticou que eram desconsiderados processos evolutivos como analogia, generalização, depuração e refinamento. Ele então “propôs o grafo genético como um arcabouço para a representação de conhecimento procedimental sob uma perspectiva evolucionária” [25].

2.3.1 Conceitos e Terminologia

O grafo genético modela a evolução de conhecimento por meio da representação de regras como vértices e de relações como arestas. Ambos os componentes são descritos na sequência. Em caráter de exemplo, a Figura 2.2 apresenta uma região de um grafo genético que modela o conhecimento do domínio de Programação de Computadores.

Regras

Os vértices do grafo genético representam as regras procedimentais do domínio de conhecimento em questão. São equivalentes às perícias ou, mais precisamente, às subperícias do domínio representado. Assim, o processo de aprendizagem consiste na apropriação de cada uma destas regras que denotam perícias a serem desenvolvidas.

Relações Genéticas

Os arcos do grafo representam as relações genéticas, ou seja, as associações evolucionárias entre perícias (vértices). Consistem em:

- Generalização / Especialização;
- Analogia;
- Simplificação / Refinamento;
- Desvio / Correção.

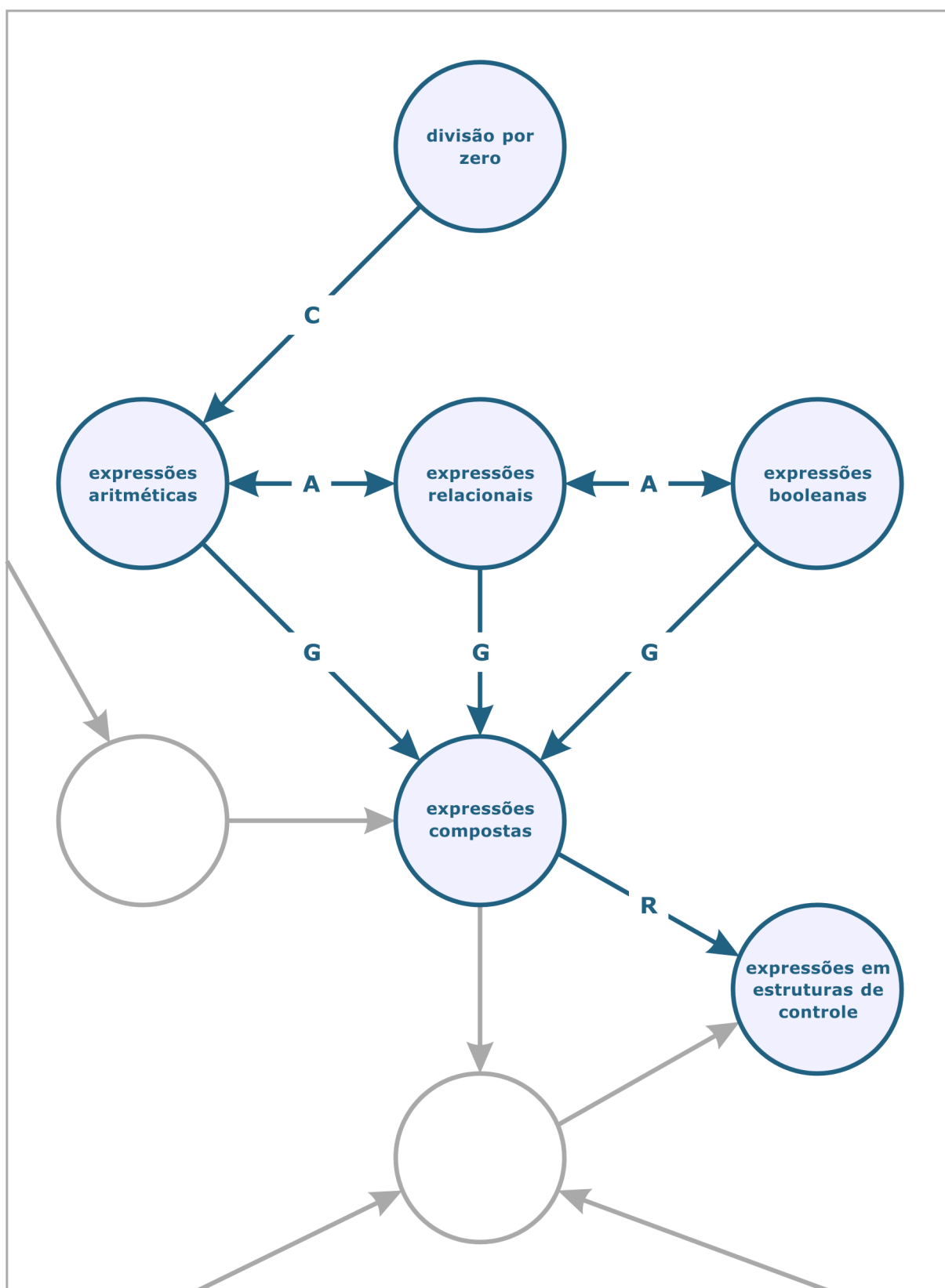


Figura 2.2: Exemplo de grafo genético contextualizado no domínio de Programação de Computadores [25]

Extensões do Grafo Genético

Goldstein, na continuidade, propôs estender o grafo genético para que as relações fossem além das perícias individuais. Ele considerou as relações entre grupos de perícias, como também os fatos declarativos que explicam e justificam o uso de delas. O modelo evolutivo passa a ser chamado de **grafo genético estendido** e conta com as seguintes ferramentas, que ampliam o potencial de representação do grafo em casos e domínios específicos:

- agrupamento de perícias em **ilhas de conhecimento**;
- representação de **conhecimento declarativo** além do procedimental; e
- representação da **ordem**, ou seja, determinação das relações genéticas de pré e pós-requisitos entre perícias.

2.3.2 Contribuições para a Tutoria Inteligente

Goldstein destaca que a contribuição dos grafos genéticos para a área de Sistemas Tutores Inteligentes se expressa diante de três perspectivas: base para a modelagem, base para a tutoria e base para a aprendizagem. Cada uma delas é abordada na sequência.

2.3.2.1 Base para a Modelagem

No contexto do grafo genético, a **sobreposição** enfatiza que a estrutura correspondente ao Modelo do Aprendiz deriva do conhecimento do domínio. Dessa forma, sendo o conhecimento do domínio um grafo genético, modela-se o aprendiz sob a perspectiva de um subgrafo dele. Portanto, interpreta-se que o aprendiz domina um subconjunto das perícias que compõem a capacidade de um experto.

A modelagem evolutiva ainda contribui para que a sobreposição não se restrinja ao conjunto de perícias. Adicionalmente, ela pode acontecer em termos de todos os outros componentes do grafo genético, tais como as subperícias e as próprias conexões (generalizações, analogias, simplificações, desvios, requisitos).

Assim, enquanto os **vértices** representam o conhecimento do aprendiz frente ao domínio, as **conexões** do grafo servem como uma estrutura complementar que modela o comportamento da aprendizagem desse indivíduo. Tais conexões acabam indicando qual foi o caminho percorrido pelo aprendiz no desenvolvimento de cada perícia. Com isso, a natureza das relações genéticas trazem informações sobre quais estratégias de aprendizagem se mostram mais eficientes àquele indivíduo.

2.3.2.2 Base para a Tutoria

O grafo genético pode guiar o Modelo Pedagógico de duas maneiras diferentes. A primeira delas, mediante a sugestão de qual perícia deve ser abordada como tópico de estudo pelo aprendiz, dentre aquelas que pertencem à fronteira da sua posição atual no grafo. A outra, explicando uma perícia (escolhida como tópico de estudo) por mais de uma via, através de relações evolutivas com outras perícias anteriormente adquiridas.

O grafo genético, contudo, não resolve o problema da **escolha do tópico a ser ensinado**. Ainda assim, ele provê uma fronteira de perícias que constituem possíveis conteúdos a serem escolhidos, tidos como mais adequados ao aprendiz. As perícias com maior prioridade de serem ensinadas são aquelas localizadas nessa fronteira, entre o sub-grafo do Modelo do Aprendiz e o restante do conhecimento do domínio. Cabe ao Modelo Pedagógico eleger um tópico dentre as perícias do conjunto, utilizando, por exemplo, um processo de busca heurística que oriente regiões do grafo a serem exploradas pelo aprendiz. Outra possibilidade para esse modelo é o uso de estratégias específicas relacionadas ao aprendiz.

Após a seleção do tópico de estudo, uma importante técnica de ensino apoia-se na capacidade de explicar esse tópico por mais de uma maneira, ou seja, o **suporte a múltiplas explicações**. Isso é possível porque o grafo genético permite que uma nova perícia seja explicada por meio das suas próprias conexões. Logo, para cada tipo de relacionamento, pode-se adotar uma estratégia diferente de explicação.

A escolha da melhor explicação para uma perícia, assim como a seleção do próximo tópico a ser ensinado, não é determinada pelo grafo genético. Devem ser considerados o

aprendiz em questão e o seu estágio no momento, dependendo da aplicação de heurísticas de ensino e de estratégias específicas relacionadas às particularidades do indivíduo. Pode-se, por exemplo, variar as explicações e evitar estratégias anteriormente malsucedidas.

2.3.2.3 Base para a Aprendizagem

A maioria dos recursos fundamentados pela Teoria dos Grafos se estendem aos grafos genéticos. Assim, consegue-se estabelecer uma relação entre os conceitos dessa teoria e a **complexidade de aprendizagem**.

Desse modo, por exemplo, podem ser fornecidos meios de orientar o tutor sobre regiões do grafo que precisam de atenção, como também de sinalizar perícias difíceis de serem adquiridas pelo aprendiz. Percebe-se então que o amparo de algoritmos próprios da área (ex. caminho mínimo) e o reconhecimento de características do grafo (ex. ordem, tamanho e conectividade) se revelam ferramentas bastante úteis para o Modelo Pedagógico.

Como outro exemplo, considerando essas características, uma perícia que possua várias relações sugere uma menor dificuldade de ser aprendida, pois oferece múltiplos caminhos para ser alcançada. Em compensação, uma perícia mais isolada pode requerer orientação tutorial de maior intensidade.

Outra possibilidade indicada por Goldstein reside na **simulação de aprendizes**, criados em termos de regiões do grafo genético, como metodologia de pesquisa. A intenção seria experimentar diferentes estratégias pedagógicas reproduzindo a conquista do grafo completo.

Por fim, o autor conjectura sobre a suposição de modelar habilidades de estudo em um grafo genético, igual e paralelamente ao conhecimento do domínio. A **modelagem de processos metacognitivos**, conforme denominação, abriria espaço para que essas habilidades também fossem tutoradas.

2.4 Aquisição de Conhecimento em Programação de Computadores

O processo de aprendizagem em domínios de natureza prática e complexa fundamenta-se na aquisição de conhecimentos sobre **princípios** e consequente desenvolvimento deles até que constituam **perícias** na área [20]. A **aquisição de princípios** corresponde à assimilação, pelo aprendiz, dos fundamentos de um domínio específico, ou seja, do conhecimento formal inicialmente repassado. O **desenvolvimento de perícias** acaba sendo o processo de construção de conhecimento por meio da experiência, levando ao crescimento da aptidão pela prática. Esse processo envolve a integração dos princípios aprendidos com a experiência que se alcançou até o momento.

Mesmo que o conhecimento sobre princípios seja semelhante em peritos e aprendizes, o conhecimento experiencial pode ser muito distante entre eles. Segundo analogia de [25], os princípios funcionam como um conjunto de ferramentas básicas que se tem à disposição, enquanto a perícia se constitui da habilidade de manipular essas ferramentas. Assim, embora contando com as mesmas ferramentas disponíveis, aprendizes e peritos empregam-nas com níveis de habilidade diferentes na resolução de um problema.

Diante da mesma perspectiva [20], conforme [25], a aprendizagem em Programação de Computadores pode ser entendida como a aquisição de conhecimentos sobre os princípios de lógica de programação e consequente desenvolvimento deles até consolidarem perícias na área. Os princípios envolvem a compreensão das instruções isoladas, considerando a rigidez léxica, sintática e semântica da linguagem utilizada. As perícias se tratam do encadeamento e aninhamento dessas instruções na construção de um algoritmo. Portanto, a aplicação de ambas as categorias de conhecimento são exigidas no ato de programar.

Nesse sentido, [38] identificou o seguinte conjunto de capacidades presentes no estereótipo de um programador perito:

1. Precisão sintática;
2. Precisão semântica;

3. Identificação de estruturas principais no programa fonte (busca por palavra chave);
4. Simulação mental dos estados do computador durante a execução;
5. Catálogo de erros;
6. Mapeamento mental das estruturas do programa;
7. Checagem de pré-condições;
8. Análise do problema;
9. Integração dos subproblemas;
10. Generalização da solução;
11. Reutilização de soluções já conhecidas; e
12. Catálogo de soluções.

Em continuidade, a pesquisa de [25] atuou na revisão do conjunto de capacidades identificado por [38]. Fez-se isso mediante a observação e análise do material produzido por alunos de disciplinas da área de Programação de Computadores, durante cinco anos. O objetivo da pesquisa residia em catalogar enunciados de programação frente ao conjunto de capacidades revisado.

Nesse estudo, foi constatada a necessidade de definir perícias de mais baixo nível de abstração que, por sua vez, correspondessem à prática em Programação de Computadores. A necessidade foi justificada pela dificuldade de relacionar enunciados com as perícias isoladas do conjunto definido por [38]. A perícia de *precisão sintática* foi um dos exemplos apresentados, por ser exigida em praticamente qualquer exercício da área.

Considerando isso, [25] reconheceu que o conjunto de capacidades de [38] se caracterizava pelo alto nível de abstração e pela sobrejacência a outras perícias mais próximas dos elementos instrucionais da programação. Entretanto, eram justamente essas últimas perícias que se faziam relevantes à modelagem e evolução do aprendiz frente ao conhecimento do domínio, bem como à catalogação e ao sequenciamento de enunciados.

A abordagem para identificar esse segundo conjunto de perícias foi reconhecer frações mais restritas de habilidades em correspondência às instruções e princípios de programação. Assim, uma primeira tentativa identificou 335 perícias, hierarquizadas em cinco níveis. Considerou-se apenas o seguinte subconjunto de Programação de Computadores:

1. Conceitos iniciais. Análise e abstração de informações na resolução de problemas
2. Composição e efetivação de algoritmos em linguagem natural;
3. Declaração de variáveis e constantes. Instruções de entrada, saída e atribuição;
4. Expressões aritméticas, relacionais e booleanas;
5. Estruturas condicionais;
6. Estrutura de repetição.

Embora o detalhamento atingido fosse esclarecedor como metaconhecimento do domínio, acabava sendo negativo para a perspectiva de autoria e catalogação de enunciados. A elicitación de um enunciado ocorreria frente à alternativa de contribuição para o desenvolvimento de 335 perícias, adicionando complexidade ao processo de autoria.

Além disso, a pesquisa intencionava formalizar as perícias identificadas por meio de um grafo genético (técnica de modelagem detalhada na Seção 2.3). Dessa forma, a quantidade elevada de perícias implicaria em sobrecarga de interconexões, além da alta densidade e frequência de cruzamentos de arestas no grafo resultante.

Diante da necessidade, o conjunto de perícias foi reduzido em termos daquelas pertencentes ao primeiro nível hierárquico e, depois, passou por fases de aprimoramento. O grafo genético que modela o conjunto é composto por 41 perícias e 60 relações (Anexo A), sendo apontado como versão experimental e sujeito a alterações pela comunidade científica.

Os estudos de [25] ainda trouxeram uma revisão, propriamente dita, das características do estereótipo de um programador perito definidas em [38]. Entretanto, esse conjunto recebeu a denominação de **perícias sobrejacentes de alto nível**. Apresentou-se o seguinte acréscimo às 12 perícias inicialmente identificadas:

13. Velocidade de resolução;
14. Legibilidade do código escrito;
15. Otimização de soluções;
16. Capacidade de depuração;
17. Definição de casos básicos de teste;
18. Construção de diálogo adequado de interface;
19. Autoconhecimento sobre habilidades metacognitivas.

O aprofundamento das últimas perícias começa a extrapolar o domínio de Programação de Computadores e a abranger outras áreas, como Engenharia de Software, Interação Humano-Computador (IHC) e Psicologia Cognitiva.

2.5 Sequenciamento de Enunciados

Contribuições anteriores evidenciaram a complexidade dos enunciados propostos como um dos principais componentes de motivação do aprendiz [10]. Equivalentemente, [38] sustenta que “a repetição sistemática de enunciados completamente diferentes, porém de graus de complexidade aproximadamente iguais pode elevar consideravelmente a autoconfiança do aprendiz, mantendo-o motivado e produtivo”.

Em contrapartida, [25] destacou que a extrema rigidez na ordenação das galerias de enunciados é um fator adverso em ambientes de ensino. Justificou que, embora determinada sequência de enunciados tenha sido resultado de decisões fundamentadas, não se garante de que ela privilegie a heterogeneidade de perfis e competências de diferentes aprendizes. Com isso, tais aprendizes são negligenciados à proporção que se afastam do perfil idealizado. No ensino de Programação de Computadores, [38] descreve **medidas cognitivas**, diante da perspectiva do desenvolvimento de perícias de [20], que se mostram úteis no sequenciamento automático e adaptativo de enunciados. As medidas

cognitivas são utilizadas para obter metaconhecimento sobre os enunciados. Essas medidas possuem a função de quantificar cognitivamente um enunciado, sendo responsáveis de estimar quanto ele exige de um aprendiz em termos de conhecimentos adquiridos e de capacidades desenvolvidas na progressão do aprendizado.

Com o mesmo propósito, a **carga cognitiva** de um enunciado é definida como a capacidade de ele exercitar o aprendiz na construção de um programa de computador, contribuindo para o desenvolvimento de perícia na área. Assim, a carga cognitiva pode ser dividida em subcomponentes que se responsabilizam por medir contribuições distintas proporcionadas pelo enunciado ao aprendiz.

A complexidade do software é um dos subcomponentes citados pelo estudo. Entretanto, como, na fase inicial do aprendizado, a maioria dos programas resultantes são pequenos, a própria complexidade passa a ser melhor determinada pelo uso de linhas de código (LOC), somadas à complexidade estrutural (CE) e aos detalhes de implementação (DI). Cabe à complexidade estrutural avaliar a quantidade de estruturas condicionais e de repetição, e a interdependência entre elas (aninhamentos). Detalhes de implementação remetem à quantidade condições específicas que dificultam o cumprimento do enunciado.

Em conjunto com os subcomponentes relacionados à complexidade do software, a carga cognitiva considera a contribuição de cada enunciado para que o aprendiz desenvolva características do estereótipo de um programador perito (Seção 2.4). Assim, avalia-se a contribuição/exigência de um enunciado diante de cada característica desse estereótipo.

A carga cognitiva de um enunciado, então, é a ponderação das suas medidas cognitivas. Tal ponderação é necessária porque diferentes medidas possuem níveis distintos de contribuição em um enunciado. Os pesos são determinados por meio de informações obtidas de especialistas no domínio.

Dessa maneira, as medidas cognitivas conseguem amparar a escolha do próximo enunciado em uma sessão de ensino. Elas auxiliam na classificação de enunciados por grau de exigência/contribuição, podendo adaptar a sequência deles ao nível do aprendiz naquele momento.

No que diz respeito à implementação, a mesma pesquisa desenvolveu a ferramenta de

autoria **Sequence**, inicialmente concebida para a área médica e depois adaptada ao ensino de programação. Ela avalia um enunciado quanto à complexidade (LOC, CE e DI), considerando as perícias de precisão sintática e semântica, análise do problema, reutilização de soluções e simulação mental. A partir disso, a ferramenta sugere uma sequência inicial de enunciados, podendo adaptá-la a cada situação por meio das informações obtidas do Modelo do Aprendiz.

Recentemente, conforme antedito, os estudos de [38] foram retomados por [25]. A pesquisa revisou as capacidades do estereótipo de um programador experto e modelou essas perícias em um grafo genético (seções 2.4 e 2.3). Nessa abordagem, tanto o conhecimento do aprendiz quanto a contribuição de cada enunciado constituem subgrafos do conhecimento do domínio. A referida pesquisa implementou duas ferramentas, uma destinada à **descrição do conhecimento do domínio** e outra que se concentra na **catalogação e elicitación de enunciados**.

Entretanto, ainda não foram voltados esforços de pesquisa à continuidade dos estudos de [25]. Uma possível continuidade imediata, objeto desta dissertação de mestrado, reside na implementação da modelagem dinâmica do aprendiz. Com isso, deseja-se prover um processo de busca heurística que oriente conhecimentos a serem explorados e, igualmente, seja utilizado no sequenciamento adaptativo de enunciados ao aprendiz. Diante disso, é conseqüente incidir sobre a alimentação do Modelo do Aprendiz, conforme seu desempenho na resolução dos enunciados propostos.

Na continuidade dessa seção, são apresentados adicionalmente seis sistemas tutores que promovem o sequenciamento automático de enunciados. Evidencia-se a dificuldade de encontrar pesquisas correlatas devido à falta de terminologia técnica que denote a escolha do próximo enunciado em uma sessão de ensino. As buscas melhores sucedidas em bases científicas consistiam nos termos: “*exercise selection*”, “*exercise suggestion*”, “*exercise ordering*” e “*next exercise*”. A substituição de “*exercise*” por “*problem*” fornecia ocorrências de quantidade e qualidade equivalentes.

Além disso, são escassas as publicações que forneçam detalhes dos mecanismos que apoiam a escolha do próximo enunciado nos sistemas desenvolvidos.

2.5.1 JV²M

O estudo apresentado por [15] e concentra na intersecção do ensino baseado em casos¹ e o ensino baseado em jogos. Com isso, ele propõe o sistema tutor inteligente JV²M destinado ao ensino do processo de compilação em linguagens orientadas a objetos por meio da metáfora de uma aventura em 3D (conforme Figura 2.3).



Figura 2.3: JV²M [15]

Trata-se de uma abordagem de *learning-by-doing* (aprender fazendo) em que o aprendiz se concentra em tarefas que refletem objetivos pedagógicos, contando com explicações em momentos específicos. Em justificativa, segundo [42], o melhor momento para intervir é quando o aprendiz comete um erro, pois se consegue utilizar a própria falha como oportunidade focada de explicação.

O JV²M apresenta problemas cada vez mais complexos ao aprendiz, dando-lhe explicações contextualizadas. O próximo exercício é apresentado quando se detecta que o

¹Do inglês, *Case-Based Teaching* (CBT), que compartilha raízes com o raciocínio baseado em casos (*Case-Based Reasoning*, CBR)

aprendiz tem conhecimento suficiente para avançar ao próximo nível. Isso é realizado por meio de quatro elementos seguintes:

- **Base de cenários (casos):** armazena todos os exercícios (situações-problema) que podem ser apresentados ao aprendiz;
- **Modelo do aprendiz:** mantém informações sobre a proficiência de cada aprendiz em relação ao conhecimento tutorado;
- **Módulo pedagógico:** decide o próximo exercício para o aprendiz, conforme o seu conhecimento atual, baseado em informações dos dois primeiros módulos; e
- **Ambiente interativo de aprendizagem:** representa, para os autores do estudo, o espaço de interface onde o aprendiz interage para resolver o exercício.

O ciclo de execução do aplicativo se baseia na integração desses quatro módulos, ocorrendo da seguinte maneira:

1. o exercício é selecionado pelo módulo pedagógico, de acordo com o modelo do aprendiz em questão;
2. o ambiente apresenta o mundo virtual, onde o aprendiz interage com os objetos a fim de resolver o exercício; e finalmente
3. o modelo do aprendiz é atualizado conforme sua performance na resolução do exercício.

Diante disso, a escolha do próximo exercício pode ser observada como um processo composto por duas fases: a seleção de conceitos a serem ensinados e a sugestão de exercícios baseados nesses conceitos. A primeira fase é cumprida pelo módulo pedagógico, de acordo com a evolução do conhecimento do aprendiz (vide modelo). A outra fase é desempenhada pela base de cenários, respondendo à requisição do módulo pedagógico.

Infortunadamente, os autores declararam que a maneira a qual os exercícios são escolhidos, ou seja, o detalhamento do processo de duas fases, excedia o escopo do artigo. Além disso, não se encontrou outra publicação dos autores que trouxesse essas informações à comunidade científica.

2.5.2 SQL Tutor

Por sua vez, [31] trouxe um estudo avaliativo que compara duas estratégias de seleção de problemas para um STI. Ambas consideram o estágio atual de conhecimento do aprendiz. A primeira delas utiliza a complexidade de cada problema, especificada no momento da autoria, para escolher aqueles mais apropriados ao aprendiz. A outra estratégia é mais adaptativa, pois a dificuldade de um determinado problema é calculada individualmente para cada aprendiz e a seleção é feita por meio de medidas dinâmicas de dificuldade. O sistema SQL Tutor foi utilizado como contexto de aplicação.

O SQL-Tutor é um STI destinado ao ensino da linguagem de pesquisa SQL (para banco de dados relacionais) a estudantes universitários. Ele é composto dos módulos de interface e pedagógico, bem como do modelo do aprendiz. O sistema possui as definições de várias bases de dados, um conjunto de problemas e suas soluções ideais. Um nível estático de complexidade é atribuído a cada problema, por um especialista no domínio, baseado nos conceitos necessários para a resolução. O SQL-Tutor não dispõe de um solucionador de problemas para avaliar a exatidão da resposta dada pelo aprendiz. Ao invés disso, ele compara a resposta com a solução ideal, usando o conhecimento do domínio representado por mais de 600 restrições.

O aprendiz pode selecionar problemas no sistema de diferentes maneiras:

1. fazendo do seu próprio modo, por meio de uma série de problemas para cada base de dados;
2. decidindo a partir de uma lista de alternativas;
3. requisitando ao sistema que escolha um problema com base no seu perfil (modelo do aprendiz);
4. informando a cláusula SQL que deseja exercitar para que o sistema, igualmente, opte por um problema de acordo com o seu perfil.

As duas estratégias de seleção de problemas que o estudo desejava comparar foram implementadas em versões alternativas do sistema. Em ambas as versões, no momento

em que é feita a requisição de um novo problema, apresenta-se uma página que traz o modelo do aprendiz e pergunta a cláusula que se quer exercitar (vide Figura 2.4).

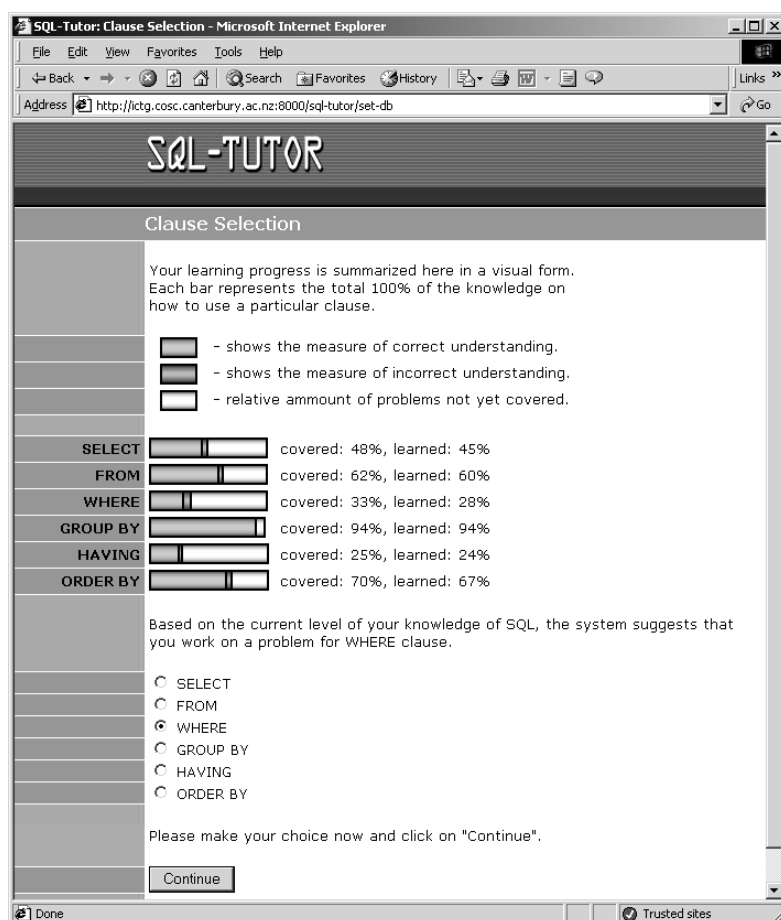


Figura 2.4: SQL-Tutor: Escolha da cláusula a ser exercitada [31]

A intenção de mostrar essas informações é encorajar o aprendiz a refletir sobre o próprio conhecimento, identificando suas dificuldades. Segundo [30], a visualização do modelo do aprendiz tem efeito positivo no processo de ensino, especialmente para aqueles com mais dificuldades, também sendo útil na seleção de enunciados apropriados para saná-las.

Entretanto, como a base de restrições do sistema é numerosa, tornou-se inviável apresentar o progresso do aprendiz diretamente nesses termos. Optou-se, ao invés, em resumir as informações em seis partes, correspondendo às cláusulas de uma consulta SQL (SELECT, FROM, WHERE, GROUP BY, HAVING e ORDER BY).

Para medir o progresso no aprendizado de cada cláusula, foi calculado o percentual de restrições relevantes utilizadas até então. O modelo do aprendiz acompanha como cada

restrição foi empregada e calcula uma estimativa da compreensão que se teve, considerando a recência (últimos usos). O percentual estimado acaba denotando o entendimento do aprendiz sobre todas as restrições relevantes para o conhecimento de uma cláusula específica. Assim, para cada cláusula, a primeira metade da barra de progresso representa o percentual de restrições corretamente aprendidas, enquanto aquelas pendentes constituem a outra metade.

O SQL-Tutor sugere a categoria de problemas que o aprendiz deveria cumprir. Na Figura 2.4, por exemplo, o sistema aponta que a cláusula WHERE deveria ser exercitada. Para sugerir uma cláusula, o modelo pedagógico observa o nível do aprendiz, que varia de 1 a 9 e é proporcional ao número de restrições existentes na base. Caso esse nível seja menor do que 3, uma das três cláusulas iniciais (SELECT, FROM e WHERE) é selecionada, visto que envolvem os problemas mais fáceis. Para os aprendizes que excederem esse limite, consideram-se as seis cláusulas na escolha do problema. A cláusula candidata passa a ser aquela em que o aprendiz mostrou maior dificuldade. Isso é feito por meio de uma métrica simples: são consideradas todas as restrições relevantes para uma cláusula e então se calcula a média das probabilidades do aprendiz conhecer essas restrições.

Uma vez que se escolha a cláusula a ser exercitada, o SQL-Tutor procura por problemas correspondentes. Dentre todos os problemas relevantes à cláusula em questão, o sistema seleciona aqueles apropriados ao estágio atual do aprendiz. Tratam-se daqueles problemas cujos níveis são iguais ou superiores à habilidade corrente do aprendiz.

A Figura 2.5 apresenta todos os problemas (de uma base de dados denominada PRODUCTS) considerados relevantes para a cláusula WHERE. Os problemas sugeridos são ordenados de acordo com as suas complexidades/dificuldades, sendo que um deles é destacado pelo sistema. O aprendiz é livre para aceitar o problema destacado ou para escolher qualquer um dos outros sugeridos, incluindo aqueles previamente resolvidos.

Os resultados da pesquisa, considerando o contraste entre as duas estratégias de seleção de problemas, mostraram que a segunda, baseada em medidas dinâmicas de dificuldade, trouxe maiores benefícios ao desempenho dos aprendizes no processo de ensino.

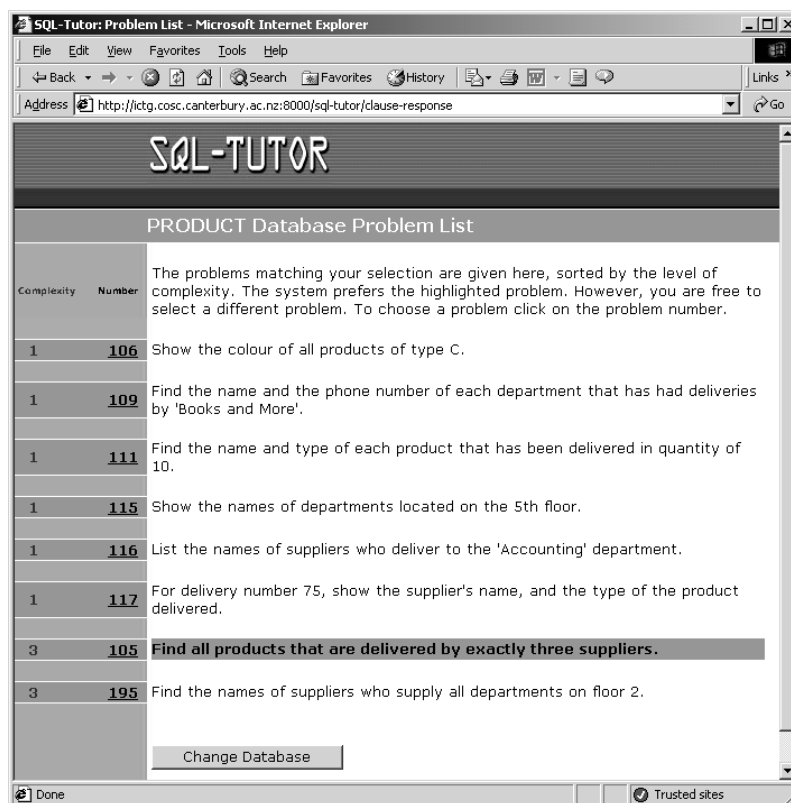


Figura 2.5: SQL-Tutor: Escolha do exercício a ser resolvido [31]

2.5.3 PHP Intelligent Tutoring System

O PHP Intelligent Tutoring System (PHP ITS) [50] destina-se a ensinar desenvolvimento *web* básico para programadores iniciantes. Ele procura orientar cada aprendiz para que estude os conteúdos mais adequados ao seu nível de conhecimento atual. Faz-se isso por meio de uma lista de exercícios sugeridos, classificados por ordem de indicação. O exercício mais indicado é a escolha padrão proposta pelo sistema, embora o aprendiz possa selecionar qualquer outro exercício da lista. A Figura 2.6 apresenta essa interação.


O autor justifica a seleção do próximo exercício como benéfica para individualizar e aperfeiçoar a interação em um STI. Contudo, antecipa as desvantagens de que o PHP ITS nem sempre é exato ao estimar o conhecimento do aprendiz. E, além disso, mesmo que o sistema faça corretamente essa estimativa, há espaço para que alguns aprendizes não aceitem as sugestões, seja porque desejam exercitar mais sobre um conteúdo, ou mesmo pular para outro.

PHP Intelligent Tutoring System [Help](#) [Skillometer](#) [ChangePassword](#) [Logout](#)

Let the ITS suggest exercises
 Search for exercises

Select the next exercise that you would like to attempt and then click the 'Proceed to exercise' button

The exercise that is most suitable for you

PHP012  2012-02-29
 You have not yet attempted this exercise

```

Display the text "Hello World" on a web page. Use PHP instead of HTML.
<html>
<body>
<?php
enter your code here
?>
</body>
</html>
  
```

Other exercises in order of decreasing suitability



| Select | Exercise Code | Already Attempted | Successfully Completed | Exercise |
|-----------------------|---------------|-------------------|------------------------|--|
| <input type="radio"/> | PHP013 | No | | Store your name into a variable named "myname". Display the string "Hello! My name is" followed by the contents of the above variable using a double quoted string.  2012-02-29 |
| <input type="radio"/> | PHP014 | No | | Store your name into a variable called "myname". Display the string "Hello! My name is" followed by the contents of the above variable using a single quoted string and the PHP concatenation operator.  2012-02-29 |

Figura 2.6: PHP ITS: Escolha do exercício a ser resolvido [50]

A seleção do próximo (melhor) exercício no PHP ITS é baseada em um método próprio que identifica o aprendizado. Esse método estabelece um limite de probabilidade acima do qual um conteúdo é considerado como aprendido. A pesquisa assumiu a probabilidade de 0,85 com a intenção de que o aprendiz não precisasse atingir o aproveitamento total para ser reputado como apto naquele conteúdo.

Segundo o autor, o exercício mais adequado ao indivíduo, em um determinado momento, é aquele que abrange o menor número de conteúdos ainda não aprendidos. Caso haja mais de um exercício com a mesma quantidade de conteúdos a serem promovidos, a escolha pode ser feita aleatoriamente.

O motivo para selecionar os exercícios com menos conteúdos desconhecidos assegura a pouca quantidade de material novo a ser exibido. Isso permite que o aprendiz construa seu conhecimento sem que ocorra a sobrecarga de contemplar múltiplos conteúdos novos em um único exercício. Trata-se de uma abordagem baseada na Zona de Desenvolvimento Proximal de Vygotsky [46], que consiste no espaço em que o indivíduo consegue aprender confortavelmente. Ela excede um pouco o conhecimento atual do aprendiz, mas não se

projeta para muito além disso.

Portanto, a lista de exercícios inicialmente mencionada (Figura 2.6) é constituída seguindo essa abordagem. Os exercícios do final da lista contém uma grande quantidade de conteúdos ainda não aprendidos, enquanto aqueles mais do topo possuem poucos conteúdos a serem aprendidos. Assim, a ordenação acaba funcionando como um critério que auxilia o aprendiz a decidir qual exercício ele resolverá na sequência.

2.5.4 Duolingo

O Duolingo² é uma plataforma gratuita destinada ao ensino de idiomas, acessível por meio de um *website* e de um aplicativo para celular. Ela oferece cursos de 50 idiomas, para falantes de 23 línguas, e propõe uma abordagem colaborativa que tem fomentado o desenvolvimento de outros 30 cursos (ditos em processo de incubação). A plataforma ultrapassou o número de 100 milhões de usuários, no segundo semestre de 2015 [39].

O projeto foi iniciado na Universidade de Carnegie Mellon, pelo professor Luis von Ahn e seu orientando Severin Hacker. Na época, a intenção era que o serviço funcionasse de maneira que os aprendizes progredissem nas lições ao mesmo tempo que traduzissem conteúdo real da Internet [27]. Outra motivação foi prover uma alternativa gratuita, contestando o custo geralmente elevado, para que pessoas de países subdesenvolvidos aprendessem idiomas.

Para cada idioma, o processo de aprendizagem é estruturado e representado externamente por uma árvore contendo as unidades a serem desenvolvidas (Figura 2.7)³. Cada unidade pode ser composta por uma ou mais lições (Figura 2.8). A organização hierárquica da árvore implica a regra estrutural de que unidades pertencentes a um mesmo nível podem ser aprendidas na ordem que se desejar. Entretanto, uma unidade mais abaixo nessa hierarquia mantém-se bloqueada até que aquelas respectivamente anteriores sejam aprendidas (pois são pré-requisitos). As lições de uma determinada unidade devem ser aprendidas na sequência, ou seja, somente se tem acesso à uma terceira lição após o cum-

²Disponível em www.duolingo.com.br.

³Existem diferenças em algumas representações externas, entre a interface para *web* e o aplicativo para celular. A descrição apresentada considera a primeira.

primento das duas primeiras, por exemplo. Ao lado de cada unidade ainda não concluída são informados o número de lições realizadas e o número total de lições nela presentes.

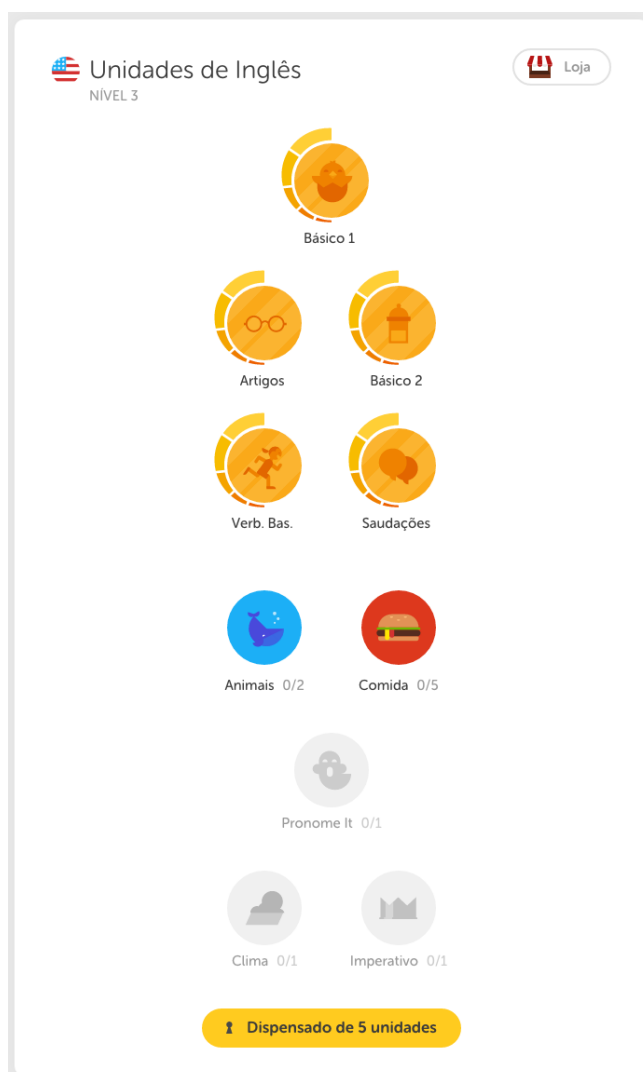


Figura 2.7: Duolingo: estruturação das unidades de conteúdo em árvore

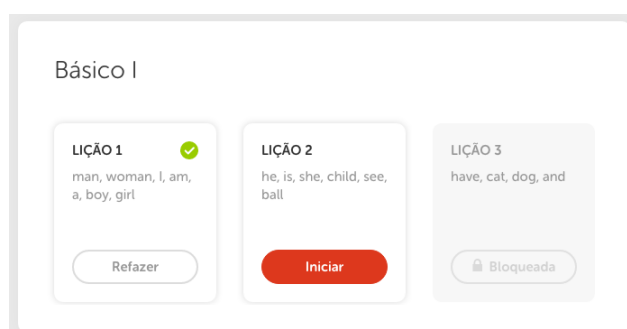


Figura 2.8: Duolingo: lições de uma unidade

A mesma árvore que sumariza as capacidades a serem desenvolvidas acaba funcionando como uma representação múltipla para situar o estado atual do aprendiz frente a esse conhecimento. Faz-se isso pela sobreposição gráfica, de maneira que os ícones de unidades aprendidas tenham coloração amarela e alaranjada (remetendo ao dourado, em metáfora a um prêmio conquistado). Unidades parcialmente desenvolvidas são indicadas pela proporcionalidade de coloração mais intensa de seus ícones, até então em cores esmaecidas. A Figura 2.9 destaca essas particularidades.

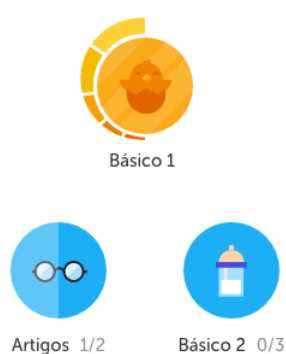


Figura 2.9: Duolingo: progresso do aprendiz pelas unidades

A visualização dessa árvore pelo aprendiz, na página inicial do idioma, tende a conduzi-lo por entre um processo reflexivo e metacognitivo acerca do próprio desenvolvimento. Ele passa a ter ciência do seu estado atual, bem como sobre a maneira como o conteúdo se encontra estruturado. Em adição à árvore, a mesma página traz um escudo laureado que informa o percentual de fluência no idioma (Figura 2.10).



Figura 2.10: Duolingo: percentual de fluência no idioma

Os conteúdos são tutorados por meio de lições rápidas, constituídas por exercícios multimodais, sendo eles:

- aprendizado de palavras novas por meio de imagens (Figura 2.11);
- aprendizado de palavras pela indicação em um texto para traduzir do idioma ensinado (Figura 2.12);
- tradução para e a partir do idioma ensinado (Figura 2.13);
- seleção da palavra que falta em uma frase no idioma ensinado (Figura 2.14);
- ditado de palavras e frases (Figura 2.15);
- pronúncia de palavras e frases (Figura 2.16);
- traduções orais para o idioma ensinado (Figura 2.17);
- escolha da(s) tradução(ões) corretas (Figura 2.18);
- pergunta de múltipla escolha (Figura 2.19); e
- associação de palavras e traduções (Figura 2.20), presente apenas no aplicativo para celular.

Selecione a tradução de "jornal"



Figura 2.11: Duolingo: aprendizado de uma palavra nova por meio de imagens

Traduza este texto



Figura 2.12: Duolingo: aprendizado de palavra pela indicação em um texto para traduzir do idioma ensinado

Traduza este texto

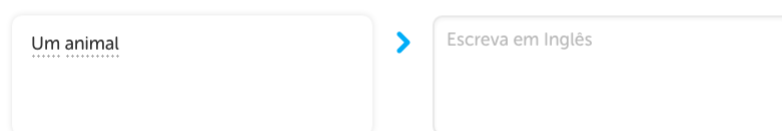


Figura 2.13: Duolingo: tradução para o idioma ensinado

Selecione a palavra que falta

She a book.

Figura 2.14: Duolingo: seleção da palavra que falta em uma frase no idioma ensinado

Escreva o que escutar

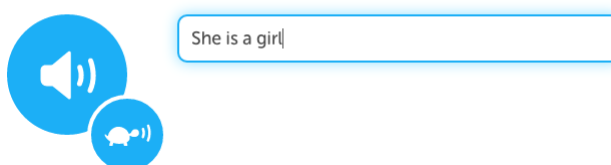


Figura 2.15: Duolingo: ditado de uma frase

Clique no microfone e diga:

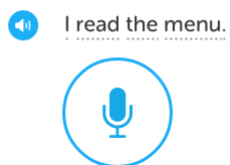


Figura 2.16: Duolingo: pronúncia de uma frase

Clique no microfone e traduza para Inglês:

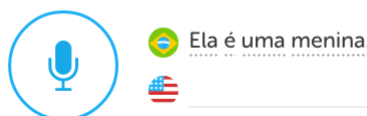


Figura 2.17: Duolingo: tradução oral para o idioma ensinado

Marque todas as traduções corretas

O arquiteto é o homem com a camisa branca.

- The architect is the man with the white shirt.
- The architect is the man with the white skirt.
- The architect is the man in the white coat.

Figura 2.18: Duolingo: escolha da(s) tradução(ões) corretas

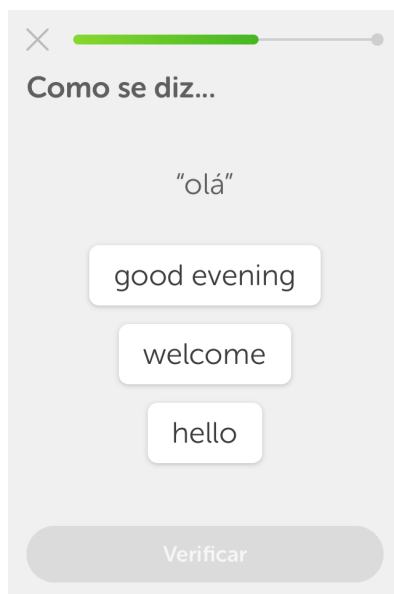


Figura 2.19: Duolingo: pergunta de múltipla escolha (aplicativo para celular)

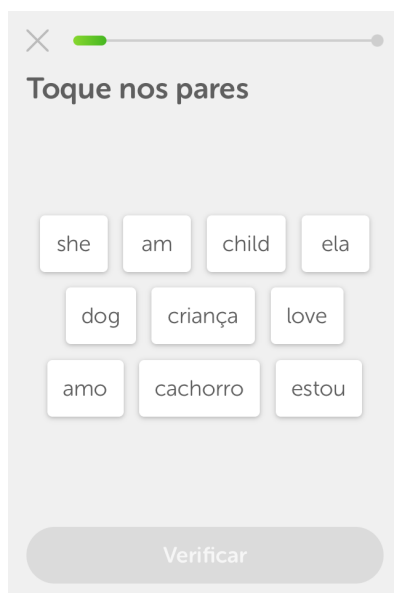


Figura 2.20: Duolingo: associação de palavras e traduções (aplicativo para celular)

Caso o aprendiz tenha errado a resposta de um exercício, tal questão é novamente proposta em outro formato na mesma lição. Assim, se o aprendiz errou ao responder um exercício de pronúncia, por exemplo, a mesma questão é apresentada no formato de ditado, e assim, sucessivamente, de maneira adaptativa.

Duas outras características interessantes na plataforma são a necessidade de reforçar o conhecimento e os possíveis testes de proficiência. Considera-se que os conteúdos precisam ser reforçados à proporção do tempo, para que não ocorra o esquecimento do que foi aprendido. A necessidade é indicada por meio de uma barra ao lado de cada unidade cumprida. A redução dessa barra indica que aquele conteúdo precisa ser reforçado. Além disso, a própria página inicial dispõe da opção intitulada “reforçar unidades” que oferece um conjunto de exercícios fazendo um apanhado do reforço necessário para cada unidade aprendida.

A possibilidade do aprendiz testar sua proficiência é garantida tanto por unidade quanto por conjunto de unidades (denominado seção). Conforme a quantidade de acertos no teste, o aprendiz mostra que domina aqueles conteúdos e pode avançar, fazendo com que aquelas unidades sejam consideradas como cumpridas.

O uso de estratégias de *gamificação* [18] é outro atrativo da plataforma, sendo promovido por recursos como:

- Avanço de níveis e conquista de pontos de experiência (XP) à medida que se aprende. Um idioma é geralmente fracionado em 25 níveis;
- Acesso a um tutor virtual, chamado de “*personal trainer*”, para quem o aprendiz determina uma meta diária (dentro os níveis: básico, casual, regular, forte e insano) e é motivado a alcançá-la (Figura 2.21);
- Contabilização dos chamados “dias de ofensiva”, que se trata do número de dias consecutivos que o aprendiz cumpriu a meta diária estipulada;
- Recebimento de *lingots*, uma moeda virtual da plataforma, pelas conquistas. Essa moeda pode ser usada para comprar itens como o bloqueio das ofensivas por um dia de inatividade;

- Prática cronometrada, em que 20 questões devem ser respondidas em 30 segundos;
- Compartilhamento de resultados com amigos;
- Possibilidade de constar o percentual de fluência em um idioma na rede social de negócios LinkedIn⁴.



Figura 2.21: Duolingo: *personal trainer*

A recompensação por pontos de experiência e o recebimento de *lingots*, conforme o desempenho e as atividades concluídas, são sempre informados ao aprendiz. Entretanto, a própria medição de desempenho, em termos do percentual de fluência em um idioma, não se mostra evidente no processo. Ocorre, por exemplo, a estagnação desse percentual por um longo período de tempo, acompanhada de avanços altos e súbitos. Nesse mesmo sentido, infelizmente, detalhes sobre os critérios de sequenciamento dos enunciados, ou mesmo sobre outros mecanismos internos de funcionamento da plataforma, não foram divulgados à comunidade científica.

Recentemente, a plataforma Duolingo passou a oferecer o Duolingo Test Center e o Duolingo para as Escolas como recursos adicionais. O Duolingo Test Center⁵ oferece

⁴Disponível em www.linkedin.com.

⁵Disponível em testcenter.duolingo.com.

certificação oficial e reconhecida para a língua inglesa, por meio da Internet. A prova tem duração de aproximadamente 20 minutos e é supervisionada remotamente por um fiscal humano, utilizando a câmera e o microfone do computador de quem realiza o teste. A legitimidade e o nível de confiança da certificação é comparável ao TOEFL e ao IELTS, tanto que ela tem sido aceita por, cada vez mais, universidades e empresas ao redor do mundo. Um grande diferencial, entretanto, reside no custo de US\$ 20, algo um décimo do valor de certificações equivalentes oferecidas até o momento, desconsiderando despesas de viagem. O valor destina-se à contratação de um fiscal humano para aplicar a prova. Outra vantagem é a obtenção dos resultados em até 48 horas após a realização do teste. A plataforma pretende expandir a certificação para outros idiomas.

O Duolingo para as Escolas⁶, por sua vez, consiste em uma ferramenta para que professores acompanhem o progresso de seus alunos no aprendizado de idiomas. Ela apresenta um painel centralizado (Figura 2.22) que ajuda o professor a identificar dificuldades e facilidades de cada aluno, com o objetivo de auxiliar na otimização dos métodos de ensino empregados. A ferramenta também identifica padrões de desempenho e comportamento de cada aluno, oferecendo exercícios e lições para reforçar os conteúdos nos quais ele possui dificuldade.

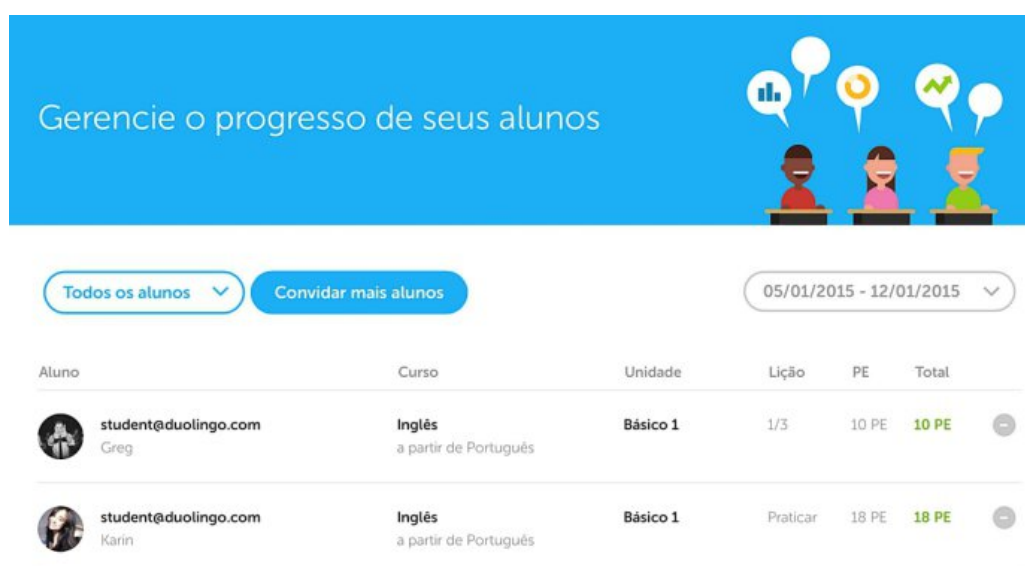


Figura 2.22: Duolingo para as Escolas: painel centralizado de acompanhamento

⁶Disponível em schools.duolingo.com.

2.5.5 ADAPTFARMA

A pesquisa de [44], concentrou-se em ajustar (calibrar, na terminologia do autor) a dificuldade dos enunciados propostos de acordo com o nível de habilidade atual do aprendiz, em se tratando de tutoria inteligente. Nesse intuito, evidenciou condições que deveriam ser consideradas no processo. Dentre as condições observadas, a ocorrência de um aprendiz acertar uma questão que a maioria dos demais errou pode indicar tanto perícia mais elevada daquele aprendiz, quando um possível maior grau de dificuldade no referido exercício. Isso, assumindo-se que os enunciados estejam adequadamente formulados. De modo análogo, um aprendiz errar uma questão que a maioria acertou pode mostrar falta de perícia daquele aprendiz e um possível menor grau de dificuldade no exercício.

Diante disso, o estudo se inspirou no sistema de *rating* (classificação) de jogos adversaristas, como o xadrez, para avaliar o desempenho de aprendizes na tutoria inteligente. Nesses jogos, o *rating* de um competidor é um número real, em um intervalo fechado predefinido, atualizado a cada torneio. Essa medida fornece um indicativo tal que, quanto maior o valor, maior a perícia de um jogador. Assim, o competidor com maior *rating* tem maior probabilidade de vencer uma partida do que o seu adversário, fazendo com que o sistema possa ser usado para prever resultados nesses jogos.

Embora a competição não seja uma abordagem usual na tutoria inteligente, os aprendizes são desafiados a realizarem atividades que têm o propósito tanto de avaliá-los quanto de desenvolvê-los em termos de perícia. Com isso, o desempenho de cada aprendiz pode ser comparado e também usado para orientar ações de tutoria, individuais e coletivas.

A pesquisa então propôs uma fórmula para fazer a classificação de um aprendiz, avaliando automaticamente seu desempenho. A fórmula considera os acertos, os erros e o número de tentativas, tanto do próprio aprendiz quanto dos demais. Ela parte do princípio de que o grau de dificuldade de cada enunciado pode ser medido por meio da taxa de aprendizes que o acertaram (ou erraram). O conjunto dessas informações é depois usado para avaliar o desempenho do aprendiz individualmente.

As seguintes diretrizes orientam a fórmula:

- o *rating* de um aprendiz é um valor real no intervalo $[1, 10]$;
- para um dado enunciado, quanto mais aprendizes o acertaram, menor é o incremento do *rating* para aqueles que conseguiram isso e maior o decremento do *rating* para os que erraram. Por hipótese, um enunciado tem nível fácil caso seja resolvido corretamente pela maioria dos aprendizes. Inversamente, considera-se um enunciado difícil caso a maioria dos aprendizes errem-no;
- cada enunciado também é classificado por meio de um valor real, proporcional ao respectivo grau de dificuldade, no intervalo $[1, 10]$;
- aprendizes que, na primeira tentativa, resolvem corretamente um enunciado atingem uma pontuação maior quando comparados com aqueles que precisaram de várias tentativas;
- no caso do aprendiz errar a resposta de um enunciado, quanto maior o número de tentativas, maior o decremento do *rating*;
- um enunciado não resolvido (pulado) é tido como um erro.

Essa fórmula atua como se implicitamente houvesse uma competição entre todos os aprendizes do grupo.

Em continuidade, o estudo desenvolveu um algoritmo de sequenciamento adaptativo de enunciados, que se baseia nos respectivos graus de dificuldade, e é guiado dinamicamente pelo desempenho do aprendiz. O algoritmo sequencia os enunciados em ordem crescente de dificuldade, combinado com um mecanismo semelhante à interpolação numérica, tendo como diretrizes:

- a calibragem dos enunciados deve ter sido previamente efetuada;
- uma sequência mínima de enunciados é definida para ser inicialmente apresentada ao aprendiz, correspondendo aos enunciados que serão resolvidos caso as respostas sempre sejam corretas;

- a sequência mínima começa com o enunciado mais fácil e termina com o mais difícil;
- os enunciados com dificuldade intermediária na sequência mínima são distribuídos uniformemente entre o mais fácil e o mais difícil, considerando o tamanho do passo, ou seja, o número de enunciados que podem ser pulados quando o aprendiz acerta uma solução. O tamanho do passo pode ser configurado no momento da autoria, de maneira que o número de enunciados na sequência mínima tenha, pelo menos, 25% de todo o conjunto;
- o número de tentativas que um aprendiz tem para resolver um enunciado é limitado pela média de tentativas, obtida na fase de calibragem dos enunciados;
- quando o número de tentativas excede esse limite, recomenda-se ao aprendiz o enunciado que se encontra no ponto médio de dificuldade entre aquele atual e o último respondido corretamente. Isto é, em síntese, retrocede-se meio passo.

Os algoritmos que calculam o grau de dificuldade dos enunciados e os *ratings* dos aprendizes, bem como o algoritmo que promove o sequenciamento adaptativo foram implementados sobre a ferramenta FARMA [23]. A FARMA (Ferramenta de Autoria para a Remediação de erros com Mobilidade de Aprendizagem) destina-se à autoria de objetos de aprendizagem por meio de interface *web*. A derivação desenvolvida pela pesquisa foi chamada de ADAPTFARMA.

Com a finalidade de avaliar a aprendizagem promovida pelos algoritmos propostos, o estudo realizou um experimento com aprendizes reais. O experimento teve 119 alunos do ensino médio técnico, entre 15 e 17 anos, como sujeitos. O material utilizado foi um objeto de aprendizagem sobre logaritmos, com 10 páginas conceituais e 30 exercícios. A avaliação aconteceu por meio de pré e pós-testes.

Os sujeitos foram divididos em quatro grupos, cada qual utilizando um método de sequenciamento diferente (MS), a saber:

1. randômico (MSR);
2. definido pelo professor (MSP);

3. ordenado por grau de dificuldade (MSD); e
4. adaptativo (MSA).

A análise dos resultados indicou que todos os métodos, com exceção do MSD, ou seja, a ordenação por grau de dificuldade, não adaptativa ao perfil do aprendiz, tiveram aumento significativo na nota do pós-teste com relação a do pré-teste. Entre esses outros métodos, o aumento foi bastante similar. Essa falta de diferença significativa foi anteriormente trazida por pesquisas como [22, 29].

Além disso, observou-se que, surpreendentemente, o MSR obteve o melhor resultado, enquanto o MSD teve o pior. Tal fato contradiz o conhecimento coletivo trazido por muitas pesquisas sobre prática pedagógica com o uso de computadores [12], que estabelece a superioridade sobre um método randômico como parâmetro de comparação.

Com isso, a análise colocou em evidência o caráter de navegabilidade proporcionado ao aprendiz, além da distinção entre os métodos adotados. O melhor desempenho, obtido pelo MSR, pode ter se devido à possibilidade do aprendiz navegar livremente pelos enunciados sugeridos, conseguindo resolvê-los na ordem que desejasse, segundo seus próprios critérios.

2.5.6 Khan Academy

A Khan Academy⁷ é uma organização educacional sem fins lucrativos criada e mantida por Salman Khan [41]. A plataforma homônima, tem a missão de oferecer educação de qualidade para qualquer um, em qualquer lugar. A organização produz e disponibiliza aulas curtas, em vídeos, abrangendo diversos domínios de conhecimento. No início, as aulas eram predominantemente sobre conteúdos de Matemática. Com o incentivo financeiro de caráter filantrópico, passaram a ser abrangidos temas de História, Saúde, Medicina, Finanças, Física, Química, Biologia, Astronomia, Cosmologia, Educação Civil Americana, História da Arte, Economia, Música e Ciência da Computação.

Embora atendam a propósitos diferentes, a Khan Academy possui tanto destaque quanto a plataforma Duolingo na atualidade. Diante disso, os vídeos das aulas já foram

⁷Disponível em www.khanacademy.org.

traduzidos para 65 idiomas e a interface *web* para 23 idiomas. No Brasil, a Khan Academy passou a ser traduzida para o português pela Fundação Lehman⁸, em 2014. As traduções encontram-se em andamento e, por enquanto, frases em ambas as línguas (inglês e português) se intercalam por entre a interface e o conteúdo.

Em se tratando do aprendizado de temas relacionados à Programação de Computadores, em língua portuguesa, a plataforma oferece os seguintes cursos:

1. Introdução a JavaScript: desenho e animação;
2. Introdução a HTML/CSS: criação de páginas *web*;
3. Introdução a SQL: consulta e gerenciamento de dados;
4. JavaScript Avançado: jogos e visualizações;
5. JavaScript Avançado: simulações naturais;
6. HTML/JavaScript: tornando páginas *web* interativas;
7. HTML/JavaScript: tornando páginas *web* interativas com jQuery;
8. Conheça o profissional.

Os fundamentos de programação são apresentados no primeiro curso, que utiliza JavaScript como linguagem. Diferentemente dos outros domínios de conhecimento, que são ensinados por meio de vídeos na plataforma, os conteúdos de programação são explicados pelos chamados guias passo a passo (Figura 2.23). Um guia passo a passo é um objeto de aprendizagem que funciona como um vídeo interativo, que pode ser pausado a qualquer momento, e que permite ao aprendiz alterar o código correspondente e fazer derivações do programa exemplificado. Depois que um conteúdo é introduzido por esse guia, propõe-se ao aprendiz o desafio de programação passo a passo (Figura 2.24), em que mensagens e dicas auxiliam no cumprimento da tarefa dada. Ainda existem os projetos, que são mais aprofundados do que os desafios passo a passo, demandam mais tempo e exigem

⁸Disponível em pt.khanacademy.org.

Informações sobre o projeto em www.fundacaolemann.org.br/khan-academy/.

maior esforço criativo (Figura 2.25). Desses projetos, alguns são avaliados apenas pelo próprio aprendiz e outros por seus pares, proporcionando abordagens metacognitivas e colaborativas.

Desenhando mais formas com código Compartilhar 75 Votos

Learn how to draw rectangles and lines with code (JavaScript and ProcessingJS).

```

1
2
3
4 rect(76, 45, 250, 300); // face
5
6 rect(126, 250, 152, 60); // mouth
7
8 rect(140, 150, 30, 30); // left eye
9
10 rect(240, 150, 30, 30); // right eye
11
12 rect(175, 345, 50, 50); // neck
13
14 // unibrow
15 line(140, 129, 270, 129);
16

```

5:24 porque você provavelmente vai querer que as formas
5:25 sejam desenhadas em uma certa ordem.
5:28 E se você não consegue encontrar uma forma que você tenha desenhado,
5:31 você pode descobrir que está se escondendo debaixo de uma outra forma.
5:34 Já que você provavelmente não tem visão de raio-x para ver isso,
5:37 uma coisa que você pode fazer é apenas reorganizar o seu código
5:40 para descobrir isso.
5:41 **E agora nós temos novamente a nossa boca.**
5:43 Ok, então nós aprendemos muito nesta lição,

Figura 2.23: Khan Academy: guia passo a passo

Desafio: um H para Hopper Compartilhar 2541 Votos

Hopper é a nossa castor, ela o guiará pelos desafios. Faça-a feliz e desenhe um "H" Trabalhando com a etapa 2 de 3

Agora, o lado direito.

Para o lado direito do H, queremos um retângulo que é exatamente igual ao do lado esquerdo, exceto pela posição de x. Não tem certeza de como começar? Você pode copiar o código que escreveu para a primeira etapa e, em seguida, alterar o valor de x (o primeiro número) até que o segundo retângulo esteja no lado direito.

Dica [\(O que é isto?\)](#)

```

rect(80, 70, 60, 240);
rect(100, 70, 60, 240);

```

```

1 rect(80, 70, 60, 240);
2 rect(

```

Todas as alterações foram salvas. Desfazer alteração Recomeçar Hmm...

Figura 2.24: Khan Academy: desafio de programação passo a passo

Derivado de "Project: What's for Dinner?"

Compartilhar

Program Guidelines

Agora que você aprendeu como desenhar e colorir formas, use suas habilidades para fazer algo divertido: jantar!

O programa abaixo inclui apenas um prato simples, cabe a você colocar comida no prato. Use formas como `rect()`, `ellipse()`, e `triangle()` para desenhar a comida no prato. Use `fill()` e `stroke()` para colorir a comida. Não se lembra de como usar alguma coisa? Dê uma olhada na guia "Documentação" e clique para obter exemplos.

Quando terminar, mostre para seus amigos e para a sua família. Se ofereça para fazer o jantar de amanhã. :-)

Você pode trabalhar em seu projeto quanto tempo quiser. Quando terminar, clique neste botão.

Feito!

```

1 background(186, 145, 20); // wooden table
2 ellipse(200, 200, 350, 350); // plate
3 ellipse(200, 200, 300, 300);
4
5 // uvas!
6 fill(195, 116, 207);
7 ellipse(100, 100, 30, 30);
8 ellipse(123, 100, 30, 30);
9 ellipse(149, 100, 30, 30);
10 ellipse(174, 100, 30, 30);
11 ellipse(106, 120, 30, 30);
12 ellipse(131, 120, 30, 30);
13 ellipse(157, 120, 30, 30);
14 ellipse(123, 140, 30, 30);
15 ellipse(149, 140, 30, 30);
16 ellipse(136, 160, 30, 30);

```

Recomeçar Solicitar ajuda Salvar

Figura 2.25: Khan Academy: projeto

O curso citado, por exemplo, é composto por 40 guias passo a passo, 35 desafios e 9 projetos. A duração do curso varia entre 15 e 40 horas, dependendo da complexidade que o aprendiz interponha em seus projetos. Como possibilidade adicional, no sentido de promover um ambiente de livre exploração, a plataforma permite que o aprendiz crie um novo programa desatrelado dos enunciados do curso, consultando a documentação disponível como auxílio.

Em cada curso, os conteúdos são linearmente estruturados, mas se admite que o aprendiz acesse lições e atividades seguintes sem que as anteriores tenham sido cumpridas (Figura 2.26). O progresso em um curso pode ser acompanhado por um painel de aprendizagem (*learning dashboard*) que sumariza as habilidades dominadas, aquelas não iniciadas e o percentual cumprido (Figura 2.27). A mesma página traz recomendações de habilidades a serem desenvolvidas, preservando a sequência do conteúdo. A adaptatividade dessas recomendações apenas ocorre pela intervenção humana de um treinador (*coach*) na plataforma. Esse papel colaborativo de treinador pode ser desempenhado por professores, tutores, pais, ou mesmo por outro aprendiz, e tem objetivo de incentivar e potencializar

os resultados de quem recebe a ajuda.



Figura 2.26: Khan Academy: estruturação do conteúdo

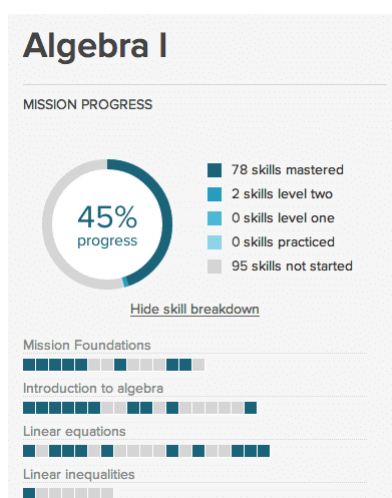


Figura 2.27: Khan Academy: painel de aprendizagem (*learning dashboard*)

De maneira semelhante ao Duolingo, a experiência é gamificada por meio de elementos como avatares, que podem ser desbloqueados, e “pontos de energia”, conquistados como uma métrica de esforço despendido. Além disso, existe a recompensação por medalhas (Figura 2.28), categorizadas pela seguinte terminologia: meteorito (comuns e fáceis de serem obtidas por iniciantes), lua (incomuns e indicam investimento na aprendizagem),

terra (raras e requerem quantidade significativa de conhecimento), sol (épicas e que exigem muita dedicação), buraco negro (lendárias e desconhecidas, sendo a premiação mais rara da plataforma) e desafio (prêmios especiais conferidos ao se completar os desafios de tópico).

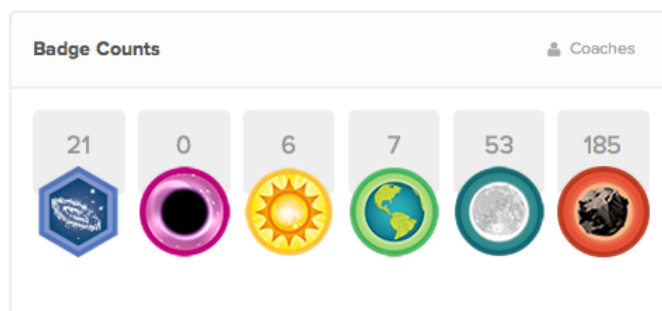


Figura 2.28: Khan Academy: medalhas

Mais recentemente, a plataforma tem investido em recursos para que pais e professores acompanhem o desenvolvimento de seus filhos ou alunos. Com o painel do tutor, por exemplo, é possível observar o perfil detalhado de cada aprendiz (Figura 2.29), bem como o desempenho geral da turma inteira (Figura 2.30). Por meio disso, pode-se perceber quando um aprendiz está tendo dificuldades, ou quando ele tem avançado mais rapidamente que o restante da turma.

| Manage Students | | | Student Progress | Skill Progress | Grid | Activity | Real Time | Coach Resour |
|--------------------------------|----------|---------------|---|----------------|------|-----------------------------|-----------|--------------|
| Class: Girls Who Code - Lowell | | | Mission: Intro to JS | | | Activity from: Last 30 days | | |
| Student Name | Points | Points | Gigi L | | | | | |
| algiersrocks | 0 | 0 | Topic Videos Badges Activity | | | | | |
| annaybannay | 0 | 0 | Intro to JS: Drawing & Animation <input type="checkbox"/> Only show completed content | | | | | |
| chlpunk.ebl | 15 | 1,000 | <ul style="list-style-type: none"> What Is Programming? <input checked="" type="checkbox"/> Completed A Tour of Programming on Khan Academy <input type="checkbox"/> Not started Intro to Drawing <input type="checkbox"/> Started Challenge: H for Hopper <input checked="" type="checkbox"/> Completed More Drawing! <input checked="" type="checkbox"/> Completed | | | | | |
| Christine Van | 0 | 0 | | | | | | |
| cora.monokandlios | 0 | 500 | | | | | | |
| Crystal Chan | 0 | 0 | | | | | | |
| David | 0 | 1,000 | | | | | | |
| ellenreller | 10 | 16,099 | | | | | | |
| ericakong0331 | 0 | 0 | | | | | | |
| Gigi L | 0 | 12,500 | | | | | | |
| GIGI Lu | 0 | 0 | | | | | | |
| grace.ohair.sherman | 0 | 0 | | | | | | |

Figura 2.29: Khan Academy: relatório de progresso de habilidades do estudante (painel do tutor)

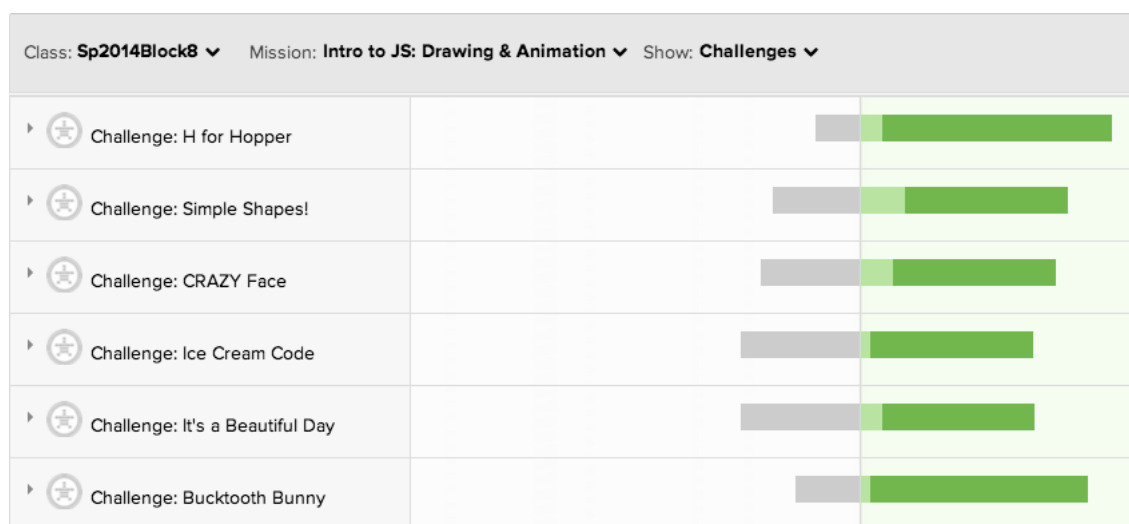


Figura 2.30: Khan Academy: relatório de progresso de habilidades da classe toda (painel do tutor)

CAPÍTULO 3

FORMALISMOS ADOTADOS NA SOLUÇÃO DO PROBLEMA

O presente capítulo apresenta a formalização do arcabouço conceitual adotado nesta pesquisa. Primeiramente, são discutidos os conceitos de automaticidade, adaptatividade e inteligência no escopo do problema (Seção 3.1). Na sequência, propõe-se uma metodologia inteligente e adaptativa para sequenciamento de enunciados, baseada em grafos genéticos (Seção 3.2).

3.1 Sequenciamento Automático, Adaptativo e Inteligente de Enunciados

A seguir, são prescritos os conceitos de automaticidade, adaptatividade e inteligência assumidos pelos autores da pesquisa no escopo do sequenciamento de enunciados. Foram consideradas as perspectivas das áreas de Inteligência Artificial [44, 21] e Interação Humano-Computador [1], ambas aplicadas à Educação. Por meio de exemplos, destacam-se os contrastes entre esses conceitos, indicando que possuem peculiaridades e que não devem ser considerados sinônimos.

3.1.1 Sequenciamento Automático Simples

Automatiza a geração de enunciados sem implicar, invariavelmente, que esse processo seja adaptativo ou inteligente (isto é, usar técnicas da IA).

Como exemplo, pode-se gerar uma sequência de 10 enunciados:

1. De maneira aleatória;
2. Sendo 4 fáceis, 4 médios e 2 difíceis, conforme categorização prévia pelo instrutor;

3. Diferente de outra anteriormente proposta.

3.1.2 Sequenciamento Adaptativo

Adapta a sequência de enunciados segundo um conjunto de critérios, não necessariamente inteligentes.

São exemplos:

1. Excluir da lista de enunciados candidatos aqueles que abrangem o mesmo tópico e têm dificuldade muito próxima de exercícios recém-resolvidos pelo aprendiz;
2. Propor um enunciado de dificuldade levemente menor, após a tentativa malsucedida de resolução de determinado exercício pelo aprendiz;
3. Reduzir ou aumentar a quantidade de enunciados da sequência, em resposta ao desempenho do aprendiz.

3.1.3 Sequenciamento Inteligente

Promove o sequenciamento de enunciados baseado em critérios inteligentes. Muito dessa inteligência se estabelece sobre a precisão da modelagem do aprendiz frente ao conhecimento do domínio. Com isso, pode-se variar radicalmente tanto a ordem e quanto a quantidade de enunciados fornecidos, considerando, por exemplo:

1. A dificuldade relativa de um enunciado frente à modelagem do aprendiz;
2. Características individuais de aprendizagem capturadas na interação;
3. Fatores coletivos dos demais aprendizes do grupo, que seriam ponderados com os individuais.

3.1.4 Discussão

Nestes termos, uma estratégia de sequenciamento pode ser automática, mas não adaptativa. Ou ainda, ser adaptativa sem que seja inteligente. Contudo, atenuando o contraste

entre as duas últimas, o comportamento resultante de um processo adaptativo pode ser visto como inteligente se considerada a perspectiva de [6], de que a inteligência está nos olhos de quem observa. Segundo o autor, ela pode apenas ser determinada pela manifestação do comportamento no ambiente.

Assim, como exemplo, pode-se considerar a comparação entre dois robôs hipotéticos, percorrendo o caminho da entrada até a saída de um labirinto específico. Assume-se que o primeiro foi anteriormente programado para aquele labirinto e apenas executa as instruções que realizam o trajeto. O segundo robô utiliza de diversos sensores para avaliar onde se encontra e, por meio de processos inteligentes, procura otimizar as instruções a fim de encontrar a saída. Ele consegue ser genérico para qualquer labirinto, mas precisa reconhecer, constantemente, onde se encontra. Esse robô, por vezes, realiza passos desnecessários e precisa corrigi-los.

Diante de observação externa, a perspectiva de [6] reconhece o primeiro robô como sendo mais inteligente, dada a atuação de ambos no labirinto para o qual esse robô foi programado. Ele cumpre o percurso de maneira ininterrupta enquanto o outro precisa se situar com frequência, sendo menos eficiente e aparentando maior dificuldade ao observador.

3.2 Metodologia para Sequenciamento de Enunciados

Descreve-se, nesta seção, a metodologia proposta nesta pesquisa para promover o sequenciamento inteligente e adaptativo de enunciados. Os seguintes passos metodológicos foram estabelecidos para a solução pretendida, sendo que cada um deles é detalhado individualmente na sequência:

1. Modelagem do aprendiz;
2. Sugestão do tópico tutorado/exercitado;
3. Sequenciamento de enunciados;
4. Atualização do Modelo do Aprendiz;

5. Acompanhamento do processo de aprendizagem;
6. Implementação de um protótipo de ferramenta.

Alguns destes aspectos metodológicos foram antecipados por [25], pesquisa cuja corrente dissertação estabelece continuidade, de acordo com o que foi explicado na Seção 2.5. Em se tratando de implementação, o mesmo estudo desenvolveu duas ferramentas (figuras 3.1 e 3.2, respectivamente). A seguir, faz-se uma breve caracterização dessas ferramentas, com a finalidade de contextualizar os passos anteriores da solução proposta [26].

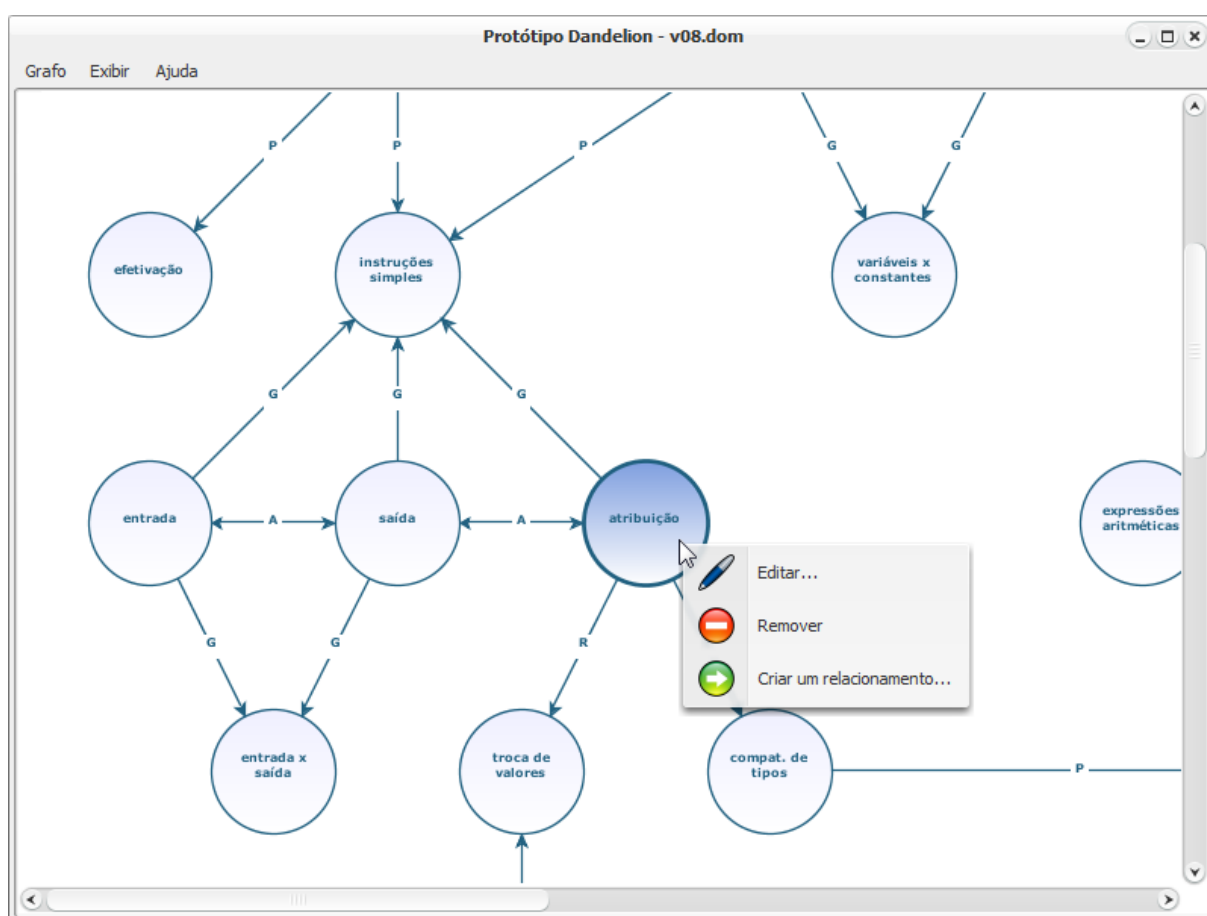


Figura 3.1: Ferramenta para a descrição do conhecimento do domínio [26]

A primeira ferramenta se concentra na **descrição do conhecimento do domínio** por meio do detalhamento das perícias componentes. A própria interação com a ferramenta ocorre por meio da representação visual do grafo genético, sendo que *menus* de contexto proporcionam acesso às opções de edição. Basicamente, permite-se: **(a)** a inserção, edição

e remoção de perícias; (b) o estabelecimento das relações evolucionárias entre perícias; (c) a definição de uma perícia inicial; e (d) a validação do modelo descrito quanto à alcançabilidade das demais perícias a partir daquela inicial. A persistência do modelo é feita em formato XML¹.

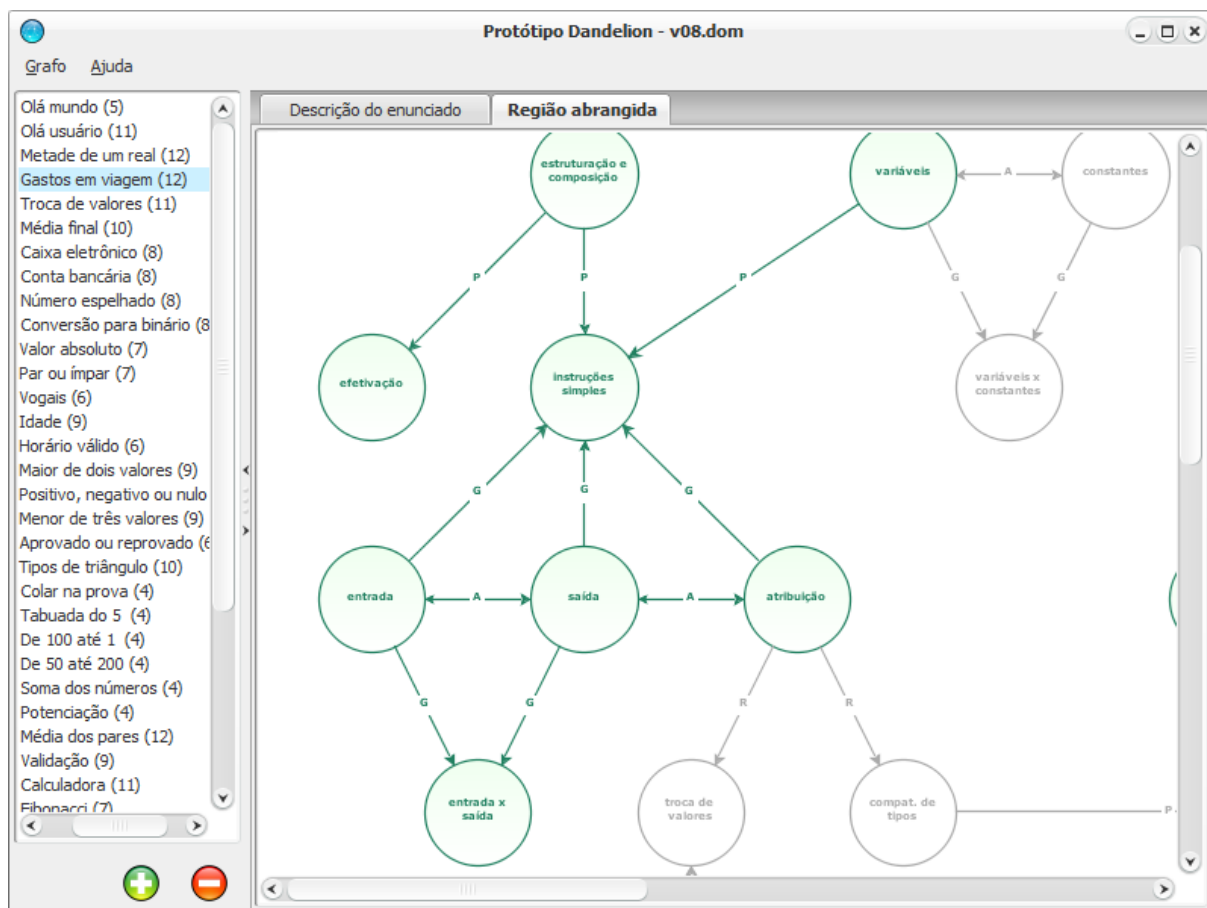


Figura 3.2: Ferramenta para a elicitaco e catalogaco de enunciados [26]

A segunda ferramenta, que usa a mesma representaco visual, destina-se à **catalogaco e elicitaco de enunciados** descritos como subgrafos do conhecimento de domnio formalizado pela primeira. Dessa forma, possibilita-se que sejam indicadas as percias que cada enunciado contempla. O aspecto de sobreposio do enunciado, no conhecimento do domnio, destaca a regio abrangida do grafo gentico. Alm disso, a ferramenta permite inspecionar a quantidade de enunciados que exercitam determinada percia, bem como a existncia de percias no abordadas pelo conjunto (catlogo) de enunciados corrente. No

¹Do ingls, *Extensible Markup Language* (Linguagem de Marcao Extensvel).

mesmo sentido, existe uma análise mais profunda sobre a coesão do catálogo e consequente requisição da autoria de enunciados adicionais.

3.2.1 Modelagem do Aprendiz

Admitindo a modelagem em grafo genético das perícias no domínio de Programação de Computadores (Anexo A), o conhecimento de um aprendiz é considerado como o subconjunto (ou subgrafo) do conhecimento de um experto. Assim, as perícias faltantes ao aprendiz são destacadas pela sobreposição do segundo grafo no primeiro.

A Figura 3.3 mostra um exemplo simplificado de grafo genético, que representa o estágio atual do aprendiz frente ao conhecimento do domínio. A sobreposição propriamente dita é evidenciada.

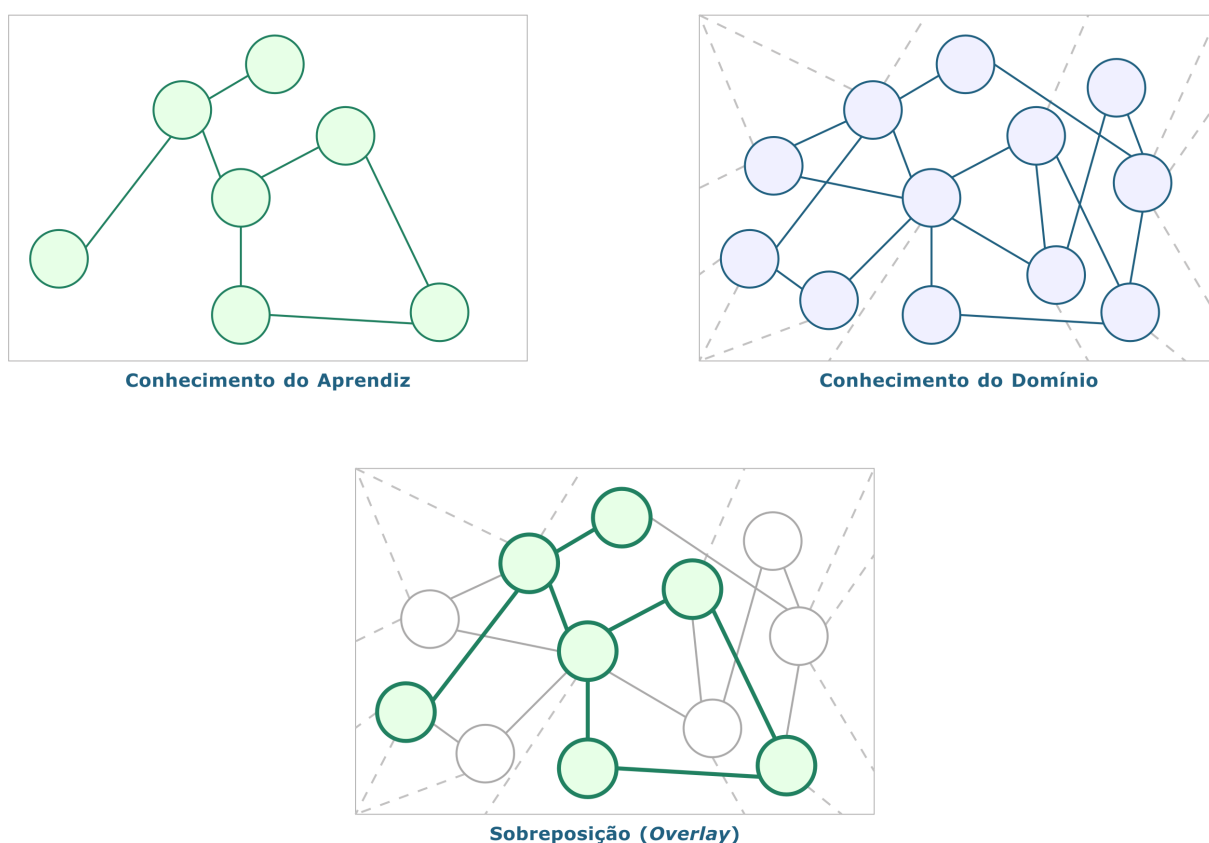


Figura 3.3: Aspecto de sobreposição do conhecimento do aprendiz frente ao conhecimento do domínio [25]

Em termos de implementação, as perícias (vértices) que compõem o conhecimento do domínio (grafo genético) são alocadas em uma tabela de espalhamento (*hash*), que

tem um identificador inteiro exclusivo como chave. As relações entre perícias (arestas) são armazenadas em listas de adjacências, associadas às perícias correspondentes. A definição desse conhecimento provém da ferramenta para a descrição do conhecimento do domínio, anteriormente apresentada neste capítulo, cuja interoperabilidade ocorre por meio da serialização dessas informações em formato XML.

Por sua vez, o Modelo do Aprendiz relaciona, em outra tabela de espalhamento, o identificador inteiro da perícia com o percentual aprendido (um número real entre 0.0 e 1.0). As perícias são referenciadas no conhecimento do domínio, por meio dos identificadores, enfatizando a característica de subconjunto.

A modelagem considerou, além das perícias desenvolvidas pelo aprendiz, a estrutura complementar que fornece indícios de como ele aprendeu (relações evolucionárias). Apesar do armazenamento dessas informações ser trivial, ou seja, uma lista com os identificadores dessas arestas, existe dificuldade em inferir tais elementos por meio da interação com aprendiz. A Seção 3.2.4 menciona a alternativa utilizada.

Adicionalmente, propõe-se a persistência dos eventos que ocorrem no modelo do aprendiz. Isso pode ser útil no aprimoramento do protótipo desenvolvido pela pesquisa, proporcionando que seja analisada a evolução do aprendiz ao longo do tempo. Os eventos seguintes relacionam a data e a hora às informações abaixo citadas:

- **Desenvolvimento de perícia:** perícia desenvolvida (identificador) e o incremento dado no percentual aprendido; e
- **Tentativa de resolver um enunciado:** enunciado sugerido (identificador) e se a resolução estava correta (booleano).

3.2.2 Sugestão do Tópico Tutorado/Exercitado

Partindo dos modelos do aprendiz e do domínio, proveu-se um processo de ordenação heurística que orienta conteúdos a serem explorados pelo aprendiz. Indicam-se perícias que deveriam ser prioritariamente desenvolvidas (compreendidas e exercitadas) pelo indivíduo, conforme o seu nível atual de conhecimento (ilustrado pela Figura 3.4).

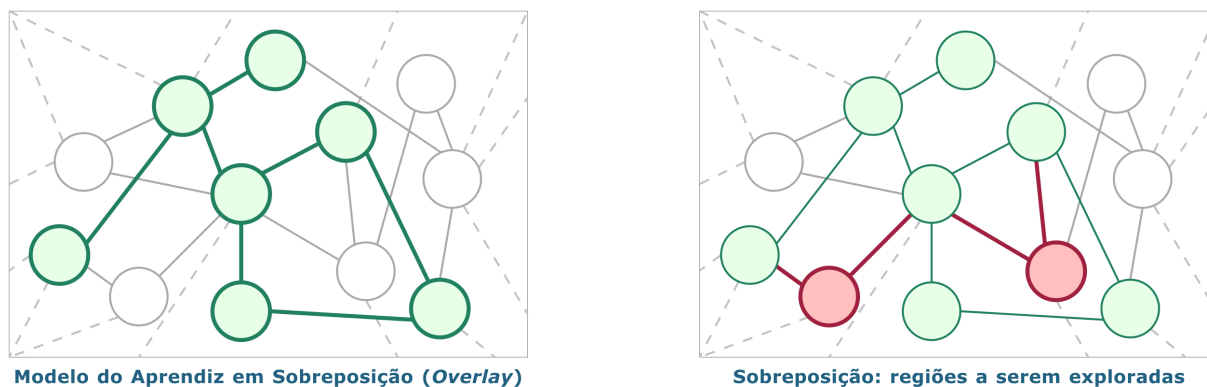


Figura 3.4: Indicação de regiões do grafo a serem exploradas [25]

Para isso, a abordagem proposta considera os seguintes critérios, detalhados em subseções próprias:

1. Priorização absoluta da perícia inicial;
2. Priorização de perícias insuficientemente desenvolvidas;
3. Perícias localizadas na fronteira entre o subgrafo do Modelo do Aprendiz e o restante do conhecimento do domínio;
4. Distância das perícias desse subconjunto até aquela definida como inicial para o conhecimento do domínio;
5. Estratégias de tutoria bem-sucedidas para aquele aprendiz (tipos de relações genéticas, como generalização, analogia, simplificação e desvio);
6. Navegabilidade proporcionada para que o aprendiz escolha entre o conjunto de perícias sugeridas.

3.2.2.1 Priorização Absoluta da Perícia Inicial

A primeira perícia sugerida ao aprendiz é aquela determinada como inicial na definição do conhecimento do domínio, feita anteriormente. A sugestão do tópico tutorado possui, como pré-condição, que a perícia inicial tenha sido aprendida. Caso contrário, ela é

obrigatoriamente recomendada como perícia exclusiva a ser desenvolvida na sessão de sequenciamento, ou seja, possui prioridade absoluta sobre as demais perícias.

Com isso, evita-se a anomalia do aprendiz avançar pelo conhecimento do domínio sem desenvolver a perícia inicial. Como medida preventiva, a situação é corrigida imediatamente quando constatada, independentemente da causa (e.g. alteração feita manualmente no Modelo do Aprendiz).

Para contexto, no domínio de conhecimento considerado, a perícia inicial possui o mnemônico *algoritmo* e refere-se à:

Domínio do conceito de algoritmo.

- (a) Conceito de algoritmo. Entendimento do conceito de algoritmo como uma sequência finita de instruções a fim de realizar determinada tarefa.
- (b) Conceito de instrução.
- (c) Entendimento do fluxo sequencial de execução.

Conforme antedito, informações detalhadas sobre as perícias do domínio encontram-se no Anexo A.

3.2.2.2 Perícias Insuficientemente Desenvolvidas

A metodologia indica a priorização de perícias insuficientemente desenvolvidas, ou seja, aquelas cujo percentual aprendido é maior que 0,0 e menor que 0,85. As perícias que possuem esse percentual igual a zero ainda não começaram a ser desenvolvidas. Por sua vez, aquelas com percentual igual ou superior a 0,85 são prescritas como aprendidas, a fim de que o indivíduo não precise atingir o aproveitamento total para ser determinado apto naqueles conteúdos. Nesse sentido, perícias com percentual de aprendizado mais próximo do limite de 0,85 teriam maior prioridade de serem aprendidas.

Esse critério foi embasado na pesquisa de [50], resenhada na Seção 2.5.3. Por ora, decidiu-se não incorporá-lo na metodologia porque dependem de detalhes da granularidade do processo, no que diz respeito ao quanto se aprende com a resolução de um exercício. Sabe-se que a resolução de um único enunciado que contemple uma perícia, pode não significar que ela foi desenvolvida. Deve-se, portanto, admitir uma progressão mais gradual que denote esse crescimento.

Contudo, em primeiro momento, preferiu-se focar os demais critérios para que a granularidade do processo fosse retomada em um trabalho futuro, após a condução de experimentos adicionais.

3.2.2.3 Perícias Localizadas na Fronteira de Conhecimento do Aprendiz

O conjunto de perícias candidatas à sugestão é constituído por aquelas localizadas na fronteira entre subgrafo do Modelo do Aprendiz e o restante do conhecimento do domínio. Trata-se do conhecimento imediatamente seguinte ao estágio atual do aprendiz.

A fronteira é determinada considerando, de maneira isolada, a adjacência de cada perícia pertencente ao Modelo do Aprendiz. Caso uma dessas perícias adjacentes ainda não tenha sido aprendida, ela é incluída no conjunto de perícias candidatas à sugestão, pois se localiza na fronteira imediata do modelo.

Inicialmente, todas as perícias da fronteira são candidatas à sugestão. Os critérios seguintes permitem estabelecer prioridades a cada perícia e, com isso, restringir o número de sugestões oferecidas ao aprendiz.

3.2.2.4 Distância até a Perícia Inicial

O primeiro fator para priorizar as perícias candidatas é a distância entre cada uma delas e aquela considerada como inicial no conhecimento do domínio. Como essa distância é constante desde a abertura do Modelo do Domínio, faz-se o cálculo nesse momento, dispensando a necessidade de realizá-lo a cada sessão de sequenciamento. O algoritmo que calcula as distâncias é uma adaptação da busca em amplitude [43] e assume que todas as relações (arestas) possuem custo unitário. A distância é armazenada em um atributo homônimo nos objetos que representam as perícias.

A princípio, não se restringe a quantidade máxima de perícias a serem sugeridas. Entretanto, as perícias candidatas são colocadas em uma lista de prioridades. O fator de maior impacto na priorização de uma perícia é a respectiva distância até a inicial.

Perícias mais próximas da inicial tendem a representar um conhecimento de necessidade mais imediata e com menor nível de dificuldade. Esses níveis, denotados pela distância, podem ser entendidos como camadas concêntricas que têm a perícia inicial como núcleo.

3.2.2.5 Estratégias de Tutoria Bem-sucedidas

A consideração das estratégias de tutoria bem-sucedidas atua como segundo fator para priorizar as perícias candidatas, depois da distância de cada uma delas até a perícia inicial. O Modelo do Aprendiz armazena, além das perícias desenvolvidas, a estrutura complementar que fornece indícios de como elas foram aprendidas (relações evolucionárias). Cada tipo de relação evolucionária ampara a capacidade de explicar (e de entender) determinada perícia por mais de uma maneira. Por exemplo, uma perícia pode ser compreendida como uma especialização de outra já adquirida, ou como uma analogia de duas outras.

Observando o número de ocorrências das relações evolucionárias no Modelo do Aprendiz, procura-se inferir quais delas foram melhores sucedidas para aquele indivíduo. Logo, a priorização das perícias a serem sugeridas pode ser refinada, ponderando-se as relações que as conectam com o conhecimento anterior. Os pesos são determinados em função do número de ocorrências de cada tipo de relação no modelo.

No início, existem poucas relações armazenadas no Modelo do Aprendiz, pois pouco conhecimento foi adquirido, e uma ponderação baseada no número de ocorrências poderia superestimar equivocadamente alguma delas. Por esse motivo, quando o modelo possui menos de 10 relações evolucionárias, são assumidos os pesos definidos na Tabela 3.1 (denominada padrão). A referida constante, de 10 relações, denota o número de ocorrências limite que condiciona a personalização da tabela ao aprendiz. O valor considera o dobro da quantidade de tipos de relações (que são 5) e foi estabelecido em caráter experimental.

Os pesos foram distribuídos entendendo-se que o grafo genético incorpora a teoria de que o conhecimento, por meio de relações genéticas, evolui:

- Do pré-requisito ao pós-requisito;
- Do desvio à correção (importantes independentemente da orientação);

- De uma analogia a outra (bilateral);
- Da especialização à generalização; e
- Da simplificação ao refinamento.

Tabela 3.1: Tabela de priorização das relações genéticas (estratégias de aprendizagem)

| Tipo de Relação | Pesos | |
|--------------------------------|--------|---------|
| | Origem | Destino |
| Pré-requisito / Pós-requisito | 10 | 1 |
| Desvio / Correção | 4 | 4 |
| Analogia | 3 | 3 |
| Especialização / Generalização | 3 | 2 |
| Simplificação / Refinamento | 2 | 1 |

Em contraste, havendo 10 ou mais relações no modelo, define-se uma tabela personalizada que procura exprimir como o aprendiz melhor se desenvolve. A composição dessa tabela respeita as seguintes diretrizes quanto às relações:

1. Pré-requisito tem prioridade máxima;
2. Pós-requisito tem prioridade mínima;
3. Desvio / correção sempre é colocada em segundo lugar, independentemente do número de ocorrências, e possui peso único;
4. A analogia, por ser uma relação bilateral, também possui peso único.

Com isso, os pesos das relações de pré-requisito/pós-requisito e de desvio/correção permanecem prioritários e inalterados. Depois, os demais pesos são distribuídos priorizando as relações com maior número de ocorrências no Modelo do Aprendiz, começando com o valor 3. Caso duas relações tenham o mesmo número de ocorrências, faz-se o desempate conforme a ordem da Tabela 3.1 (padrão).

A tabela de priorização das relações genéticas é definida no momento em que se situa o aprendiz frente ao conhecimento do domínio. Portanto, antecede qualquer passo de

sugestão e é utilizada por entre todo o ciclo de sequenciamento, até a atualização do Modelo do Aprendiz.

Na prática, os dois fatores de priorização atuam em conjunto para cada perícia localizada na fronteira do conhecimento do aprendiz, antes que ela seja inserida na lista. Por se tratar de uma lista de prioridades, a prioridade de cada perícia precisa ser conhecida anteriormente à respectiva inserção na estrutura de dados. Isto posto, a prioridade de uma perícia é definida por:

$$prioridade(perícia) = 10 \cdot distância(perícia) - relações(perícia)$$

onde:

prioridade(perícia): calcula a prioridade da perícia em questão, sendo que, quando menor o resultado, maior a prioridade;

distância(perícia): é a distância da perícia até aquela determinada como inicial no conhecimento do domínio; e

relações(perícia): é a soma ponderada das relações que conectam a perícia ao Modelo do Aprendiz, considerando os pesos da tabela de priorização das relações genéticas. Essa soma é obtida pela seguinte fórmula:

$$relações(perícia) = \sum_{relação = 1}^n \left[ocorrência(relação, perícia) \cdot peso(relação) \right]$$

onde:

relação: itera sobre os tipos de relações genéticas presentes na tabela de priorização, ou seja, sobre os elementos da tabela, começando pelo primeiro;

n: quantidade de tipos de relações genéticas considerados na modelagem;

ocorrências(relação, perícia): retorna o número de ocorrências de um determinado tipo de relação, conectando a nova perícia ao Modelo do Aprendiz;

peso(relação): peso daquele tipo de relação genética, dado pela tabela de priorização.

Como exemplo, uma perícia x , que está localizada há três níveis da inicial e se conecta ao Modelo do Aprendiz por duas analogias e uma generalização, assumindo a Tabela 3.1, possui a seguinte prioridade:

$$\text{prioridade}(x) = 10 \cdot (3) - [(2 \cdot 3) + 2] = 22$$

Dessa forma, uma perícia conectada ao Modelo do Aprendiz por meio de uma relação mais promissora será sugerida com maior prioridade do que uma segunda, cuja única relação apresentou pouco sucesso anterior, considerando que possuam a mesma distância até a perícia inicial. Além disso, quanto melhor uma perícia sugerida estiver relacionada com o que se aprendeu antes (múltiplas conexões), mais fácil ela tende de ser assimilada pelo aprendiz, justamente pela possibilidade de ser compreendida por várias maneiras.

Ainda por meio da tabela definida, ao se sugerir uma perícia, pode-se também indicar a estratégia de tutoria (relação genética) mais promissora para que o aprendiz a desenvolva. Para isso, basta considerar a relação com maior peso na tabela, dentre aquelas que conectam a perícia sugerida ao conhecimento anterior do aprendiz.

3.2.2.6 Navegabilidade Proporcionalada ao Aprendiz

Proporciona-se navegabilidade a fim de que o aprendiz considere as perícias sugeridas e respectivas prioridades mas, ainda assim, possa decidir sobre o próximo conteúdo a ser explorado. Esse critério foi destacado pela pesquisa de [44], resenhada na Seção 2.5.5, quando a atividade de sequenciamento obteve melhor desempenho ao possibilitar que o aprendiz navegasse livremente pelas sugestões de enunciados, resolvendo-os conforme a ordem desejada.

3.2.3 Sequenciamento de Enunciados

Escolhida a perícia (ou conjunto delas) como tópico a ser exercitado, encontram-se enunciados que correspondam a essa demanda. Isso se torna possível porque o mesmo grafo genético (que representa o conhecimento do domínio) é empregado como gabarito para

os enunciados a serem propostos. Nessa perspectiva, cada enunciado contribui para que se desenvolvam perícias específicas do aprendiz, sendo também um subgrafo do conhecimento do domínio (Figura 3.5). Trata-se de uma extensão conceitual que [25] propôs para os estudos de [14].

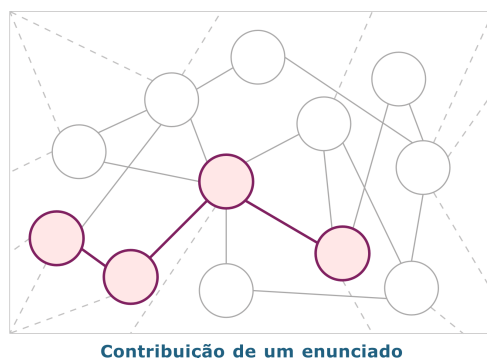


Figura 3.5: Sobreposição de um enunciado frente ao conhecimento do domínio [25]

Diante dessa possibilidade, o conjunto de enunciados selecionados foi catalogado de acordo com o referido gabarito (Figura 3.6). O sequenciamento já poderia ocorrer de maneira automática, considerando, por exemplo, a quantidade total de perícias envolvidas na resolução.

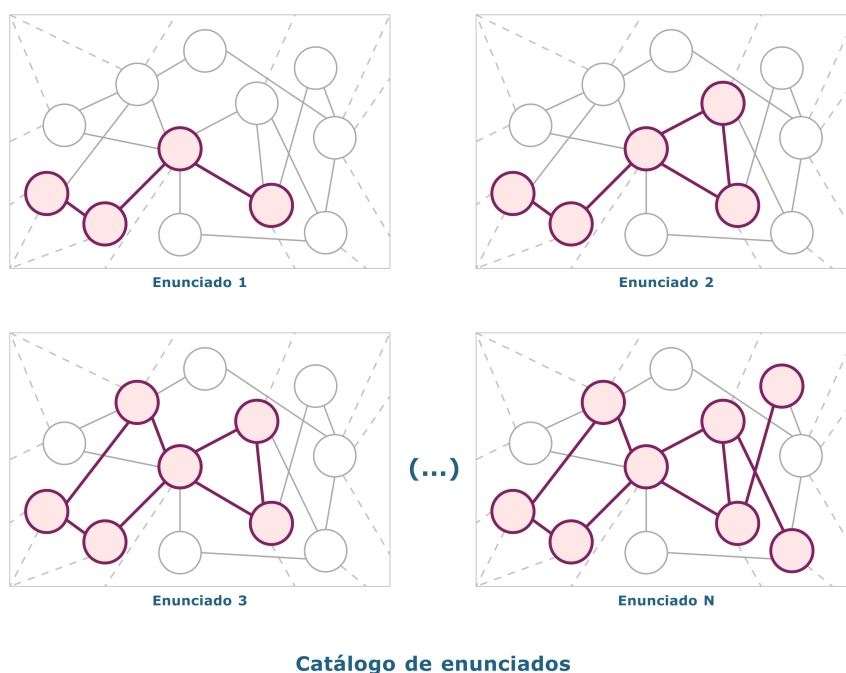


Figura 3.6: Catálogo de enunciados conforme o gabarito do conhecimento do domínio [25]

Contudo, a pesquisa investe na adaptação inteligente desses enunciados de acordo com o perfil atual do aprendiz. Com isso, diferentes exercícios podem ser sugeridos para a tutoria de uma mesma perícia, considerando aprendizes distintos. Adotou-se, então, a abordagem de priorizar os enunciados que abranjam o menor número de perícias não desenvolvidas, isto é, que sejam mais próximos do conhecimento do aprendiz naquela ocasião. Caso dois enunciados se igualem nesse quesito, prioriza-se aquele que exercita o menor número de perícias ao todo (mesmo que já tenham sido desenvolvidas).

Abordagem semelhante foi resenhada na Seção 2.5.3 e ambas se baseiam Zona de Desenvolvimento Proximal de Vygotsky [46], que excede moderadamente o conhecimento atual do aprendiz. Assim, procura-se evitar que o próximo exercício seja sobrecarregado por múltiplos conteúdos novos, sem que haja sugestão de enunciados intermediários.

Por fim, de maneira análoga à sugestão do tópico tutorado, o aprendiz pode escolher livremente entre as opções de enunciados da lista de prioridades. A quantidade de novas perícias, bem como número total de perícias exercitadas, são informações apresentadas juntamente com o identificador de cada enunciado, como auxílio à escolha.

3.2.4 Atualização do Modelo do Aprendiz

Após o cumprimento do enunciado, faz-se necessária uma avaliação para verificar se ocorreu o desenvolvimento das perícias abordadas. A alimentação do Modelo do Aprendiz é indispensável para que se passe a refletir o nível atual de conhecimento do indivíduo, considerando o possível progresso. Entretanto, apenas a sugestão do enunciado para o aprendiz não é condição suficiente para se inferir que as perícias foram desenvolvidas.

Esta pesquisa não se concentra no mérito de implementar um corretor automático para as soluções. Ela precisa de uma alternativa que forneça parâmetros para que o Modelo do Aprendiz seja atualizado, permitindo que a próxima sugestão de enunciado já considere o avanço que se possa ter conquistado.

Uma das alternativas para essa finalidade é supor a avaliação das resoluções por um especialista externo (humano), levando em conta o gabarito de perícias fornecido pelo enunciado. A Figura 3.7 expõe tal possibilidade, assumindo que a avaliação externa

assinala quais foram as perícias desenvolvidas daquelas exercitadas. O procedimento registra o progresso do aprendiz e consequente evolução na estrutura do grafo genético.

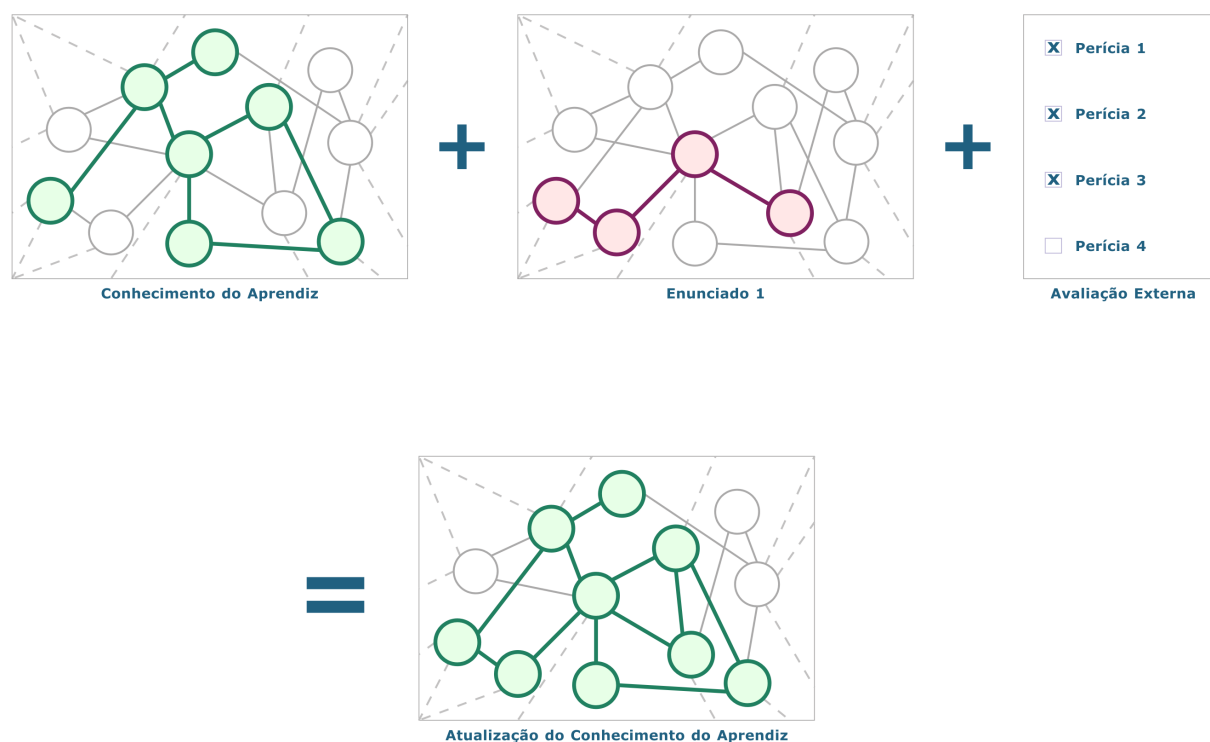


Figura 3.7: Atualização do Modelo do Aprendiz [25]

Outra alternativa seria que o especialista externo desse lugar a um diagnóstico automático simples, a exemplo do que ocorre nas maratonas de programação. A abordagem geralmente consiste em comparar a saída esperada com aquela fornecida pela resolução do aprendiz, tendo o mesmo conjunto de dados como entrada. Isso depende que, internamente, se realize a chamada de um compilador.

Ocorre que essa segunda alternativa é bastante sensível à natureza dos enunciados. No momento da autoria, tais enunciados deverão ser compostos de maneira que a resolução não seja possível sem o emprego das perícias que se pretende desenvolver.

Por exemplo, a soma dos inteiros de um intervalo não seria um bom enunciado para que se diagnosticasse o conhecimento sobre estruturas de repetição. O aprendiz poderia resolver o enunciado utilizando a fórmula da soma dos termos de uma progressão aritmética.

Para o mesmo conteúdo, o enunciado de exibir os cinco primeiros números da sequência

de Fibonacci igualmente não auxiliaria no diagnóstico. Poderia ocorrer que o aprendiz fixasse como literais os números esperados (0, 1, 1, 2, 3) em uma simples instrução de saída.

Demarcadas essas possibilidades, em caráter de simplificação, o protótipo implementado suprime a avaliação das resoluções. Contudo, encontra-se em fase final de desenvolvimento um módulo que realiza o diagnóstico automático, conforme exposto na segunda alternativa. Como trabalho futuro imediato, convém acoplar a avaliação feita por esse módulo ao processo de sequenciamento.

Atualmente, a solução admite que todas as perícias exercitadas foram desenvolvidas, desde que a resolução do aprendiz seja assinalada como correta no processo de sequenciamento. Além disso, infere-se quais foram as estratégias de tutoria utilizadas, considerando aquelas que anteriormente se mostraram melhores sucedidas ao aprendiz em particular.

3.2.5 Acompanhamento do Processo de Aprendizagem

Na metodologia proposta, o acompanhamento do processo de aprendizagem é suportado por múltiplas representações externas. Tal multiplicidade ocorre porque diferentes aspectos são representados visualmente, por meio de sobreposições sucessivas no grafo genético.

De acordo com a taxonomia funcionalista de Ainsworth [2] (Seção 2.1), são recursos semânticos do conjunto multirrepresentacional utilizado:

1. **Construção de Compreensão Aprofundada:** consegue-se isso porque as representações externas fornecem subsídio metacognitivo ao aprendiz sobre o processo de sequenciamento. Assim, eleva-se o nível da **abstração** que o aprendiz constrói acerca da estruturação do conteúdo, bem como do próprio estágio de desenvolvimento. A **relação** direta entre as etapas do processo e a correspondência com elementos do grafo genético também contribuem para que essa compreensão aprofundada exista;
2. **Papéis Complementares:** ocorre o **suporte a informações complementares** porque cada camada sobreposição, quando adicionada, denota avanço nas etapas

de sequenciamento e acrescenta informações sobre a aprendizagem. O **suporte a processos complementares** também acontece, uma vez que tais informações, quando expressas no grafo, podem melhor subsidiar estratégias de decisão do aprendiz quanto às perícias e aos enunciados;

3. **Restrição da Interpretação:** possíveis ambiguidades nas informações do processo de sequenciamento são esclarecidas pelo grafo genético, que atua como representação mais específica desse mesmo conteúdo. Portanto, as **propriedades herdadas** do grafo restringem a interpretação de elementos textuais apresentados na interface da solução.

Conforme descrito, evidencia-se não se tratar de uma classificação excludente. Assim, uma mesma representação pode desempenhar mais de uma função (ou papel) no acompanhamento do processo de aprendizagem.

Além disso, a representação em grafo abre oportunidades de extrair dos modelos informações úteis à didática do instrutor. A corrente pesquisa suporta um conjunto dessas possibilidades, considerando:

- Sintetizar o Modelo do Aprendiz de um conjunto, como uma sala de aula, para que se observe como as capacidades são ensinadas pelo instrutor (Figura 3.8). Com isso, torna-se possível notar que um instrutor se sobressai no ensino de alguns conteúdos, enquanto um segundo acaba negligenciando outros;
- Apresentar um histórico do desenvolvimento de perícias do aprendiz durante um intervalo de tempo específico (Figura 3.9). Pode-se, por exemplo, exibir o grafo sendo conquistado (coloração dos vértices) à proporção do tempo, ou seja, uma representação externa dinâmica;
- Esse mesmo recurso pode ser aplicado à sintetização do Modelo do Aprendiz de um conjunto, auxiliando a visualizar a progressão do aprendizado de uma sala inteira, ou de um curso inteiro (ainda na Figura 3.9).

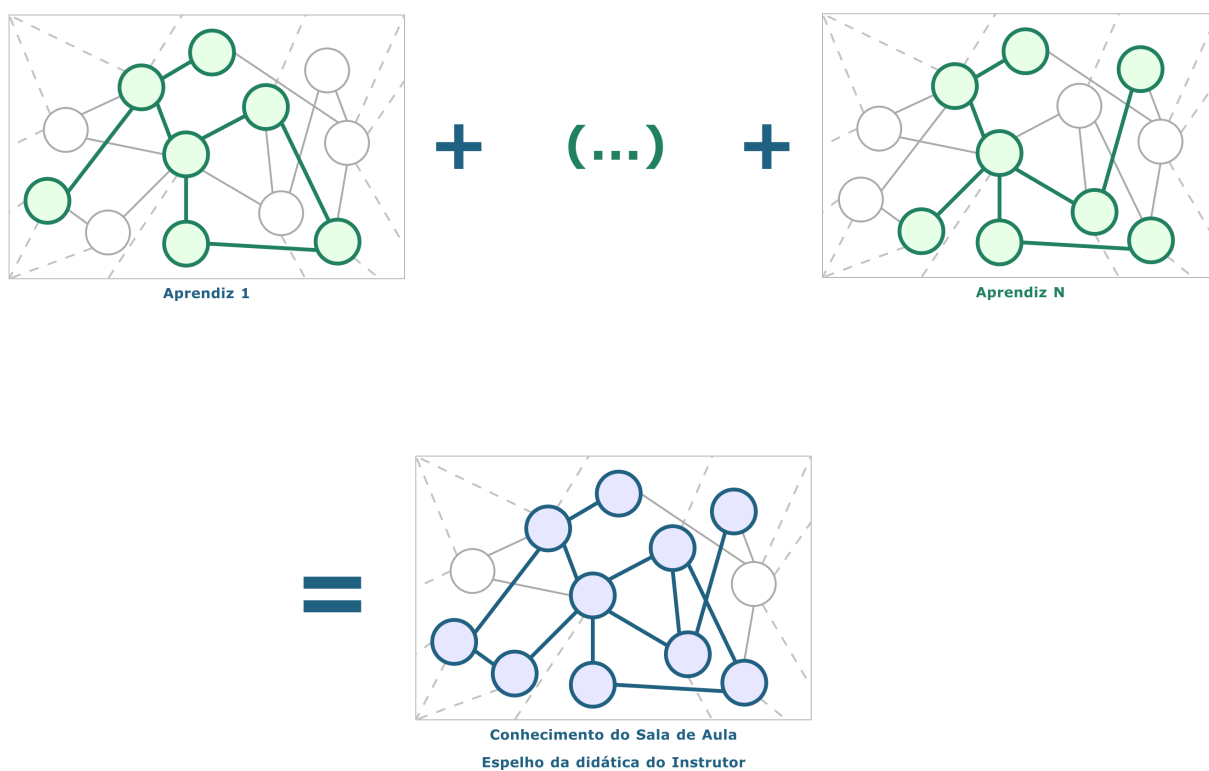


Figura 3.8: Sintetização do Modelo do Aprendiz de um grupo [25]

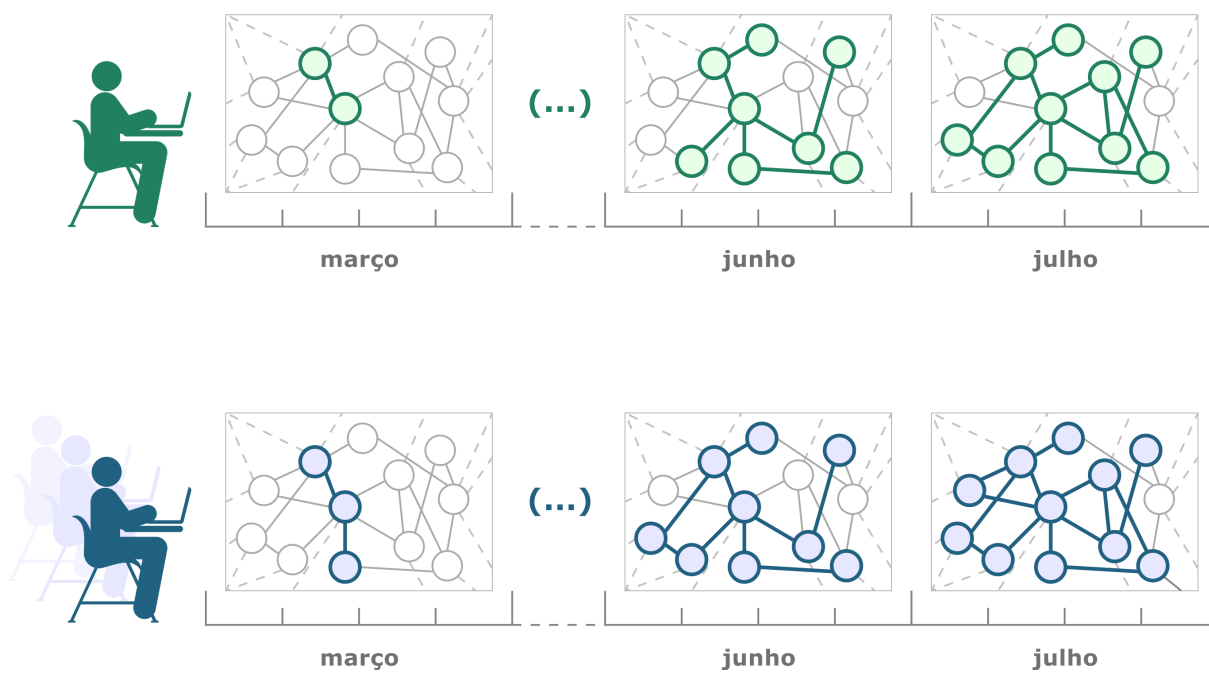


Figura 3.9: Progresso do aprendiz (ou de um grupo) ao longo do tempo [25]

O suporte às possibilidades citadas acontece porquanto a Modelagem do Aprendiz (descrita na Seção 3.2.1) faz a persistência da cronologia de eventos, que ocorrem na ferramenta implementada. Registram-se a data e a hora, relacionados ao desenvolvimento de perícias e às tentativas de resolver enunciados feitas por cada aprendiz. As últimas representações externas, contudo, não foram implementadas no protótipo.

3.2.6 Implementação de um Protótipo de Ferramenta

As funcionalidades descritas foram instanciadas em um protótipo de ferramenta que contempla o seguinte ciclo de uso, assistido por múltiplas representações externas:

1. Situar o estágio atual do aprendiz frente ao domínio de conhecimento;
2. Indicar perícias a serem prioritariamente desenvolvidas pelo aprendiz, considerando esse estágio;
3. Sequenciar e sugerir enunciados que contemplem o desenvolvimento das perícias objetivadas; e
4. Atualizar o Modelo do Aprendiz conforme a avaliação de desempenho nas soluções dos enunciados escolhidos.

A ferramenta, entretanto, consiste em um **simulador** para acompanhar o processo de sequenciamento de enunciados, fundamentado na metodologia inteligente e adaptativa proposta. O foco primário incide em observar a solução e analisar em funcionamento, não na fluência da interação com possíveis atores do cenário. A dinâmica de funcionamento dessa solução é detalhada no Capítulo 4.

Posteriormente, objetiva-se que a solução seja embarcada em um ambiente de ensino² para promover a tutoria inteligente. Nesses termos, precisa ser considerado o interfaceamento próprio, alinhado aos diversos perfis que interagirão com o referido ambiente (aprendizes, instrutores, monitores, entre outros). Além disso, será estudada a representação externa de mecanismos adicionais que permitam extrair e apresentar informações

²Considera-se a FARMA [23], desenvolvida pelo mesmo grupo de pesquisa da UFPR, como primeira opção.

úteis ao processo de ensino-aprendizagem, bem como a possível implementação de outras funcionalidades orientadas ao mesmo propósito.

CAPÍTULO 4

DINÂMICA DE FUNCIONAMENTO DA SOLUÇÃO PROPOSTA

Expõe-se, nesta seção, a dinâmica de funcionamento da solução proposta, contextualizada no domínio de Programação de Computadores. O cenário de utilização simula e acompanha o processo de sequenciamento de enunciados para um aprendiz.

A explicação tem caráter resumido, procurando enfoque na fluência desse processo. As seções foram divididas em ciclos de sequenciamento. Cada um desses ciclos consiste nas etapas de: **(1)** situar o aprendiz; **(2)** sugerir perícia; **(3)** sugerir enunciado; e **(4)** avaliar solução. Para o exemplo, foi suposta a aplicação do processo a um aprendiz que teve apenas uma introdução teórica aos primeiros tópicos da área.

4.1 Primeiro Ciclo de Sequenciamento

O Modelo do Aprendiz é um subconjunto nulo do conhecimento do domínio, pois nenhuma perícia do grafo genético foi desenvolvida ainda. A ferramenta, em primeiro momento, situa o aprendiz frente ao conhecimento do domínio. Faz-se isso como amparo metacognitivo, por meio de representações externas, de caráter visual (sobreposição no grafo genético) e textual (painel de estatísticas), como pode ser observado na Figura 4.1.

A ferramenta sugere exclusivamente a perícia inicial, que possui o identificador *algoritmo*, para ser desenvolvida na sessão de sequenciamento. Conforme o Anexo A, essa perícia abrange a compreensão do conceito de algoritmo. A localização da perícia sugerida é destacada no conhecimento do domínio (Figura 4.2).

Admitindo a perícia *algoritmo* como prioritária a ser exercitada, cabe à ferramenta encontrar enunciados que contribuam para esse desenvolvimento. Para isso, toma-se o catálogo de enunciados do Anexo B, assumido integralmente de [25]. Esse catálogo foi complementado com os enunciados do Anexo C.

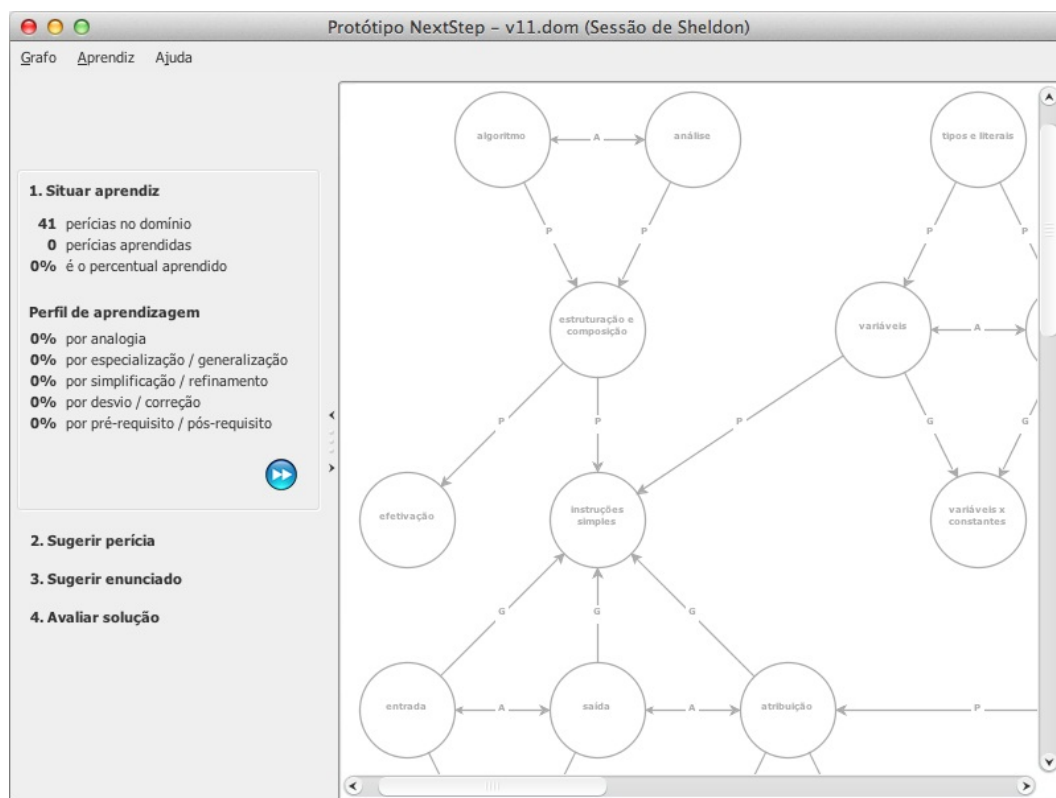


Figura 4.1: Situar aprendiz (Primeiro Ciclo de Sequenciamento)

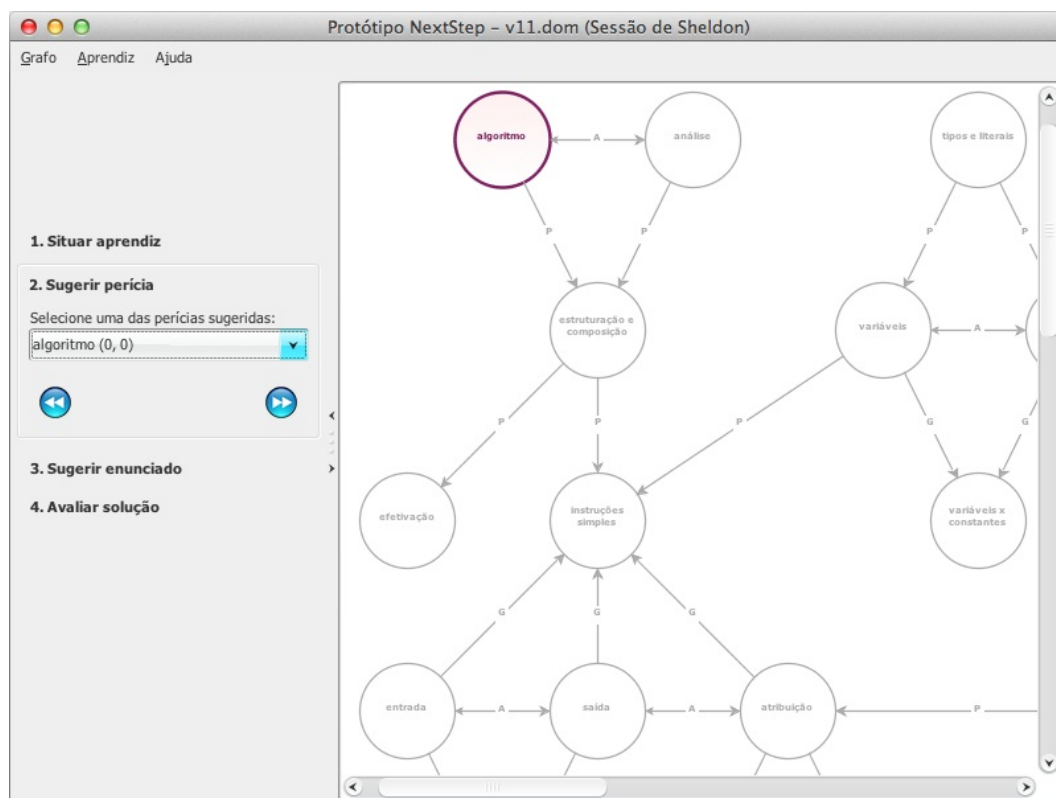


Figura 4.2: Sugerir perícia (Primeiro Ciclo de Sequenciamento)

De todo o catálogo, apenas os enunciados *Olá mundo*, *Olá usuário* e *Metade de um real* contribuem para a apropriação da perícia *algoritmo*. Convém destacar que, embora todos os enunciados de um catálogo envolvam, por exemplo, instruções de saída, nem todos se propõem a desenvolver esse conhecimento.

Como se tem mais de um enunciado que contempla a perícia sugerida, oferece-se navegabilidade ao aprendiz. A escolha é orientada pela ordem das sugestões, priorizando os enunciados que abrangem o menor número de perícias não desenvolvidas e, depois, aqueles que exercitam o menor número de perícias ao todo (mesmo que já tenham sido desenvolvidas). Essas duas informações são apresentadas ao lado do identificador de cada enunciado. Adicionalmente, quando se seleciona um enunciado, as perícias que ele objetiva desenvolver (regiões do grafo) são destacadas no conhecimento do domínio (Figura 4.3).

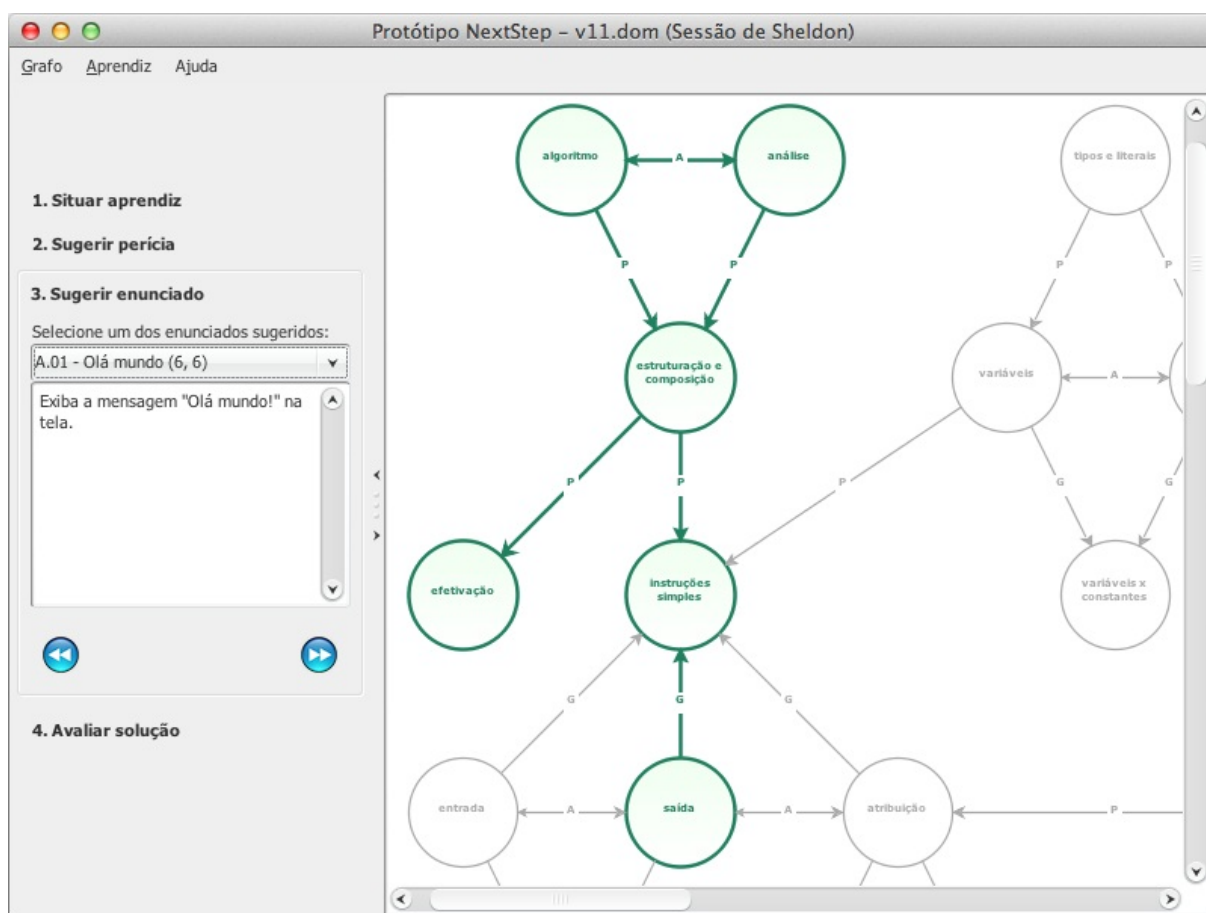


Figura 4.3: Sugerir enunciado (Primeiro Ciclo de Sequenciamento)

Então, supondo-se que o primeiro enunciado (*Olá mundo*) tenha sido a escolha do aprendiz para exercitar a perícia *algoritmo* e que a resolução seja avaliada como correta na ferramenta (Figura 4.4). Depois disso, o Modelo do Aprendiz é atualizado de modo que passe a refletir que as perícias exercitadas foram desenvolvidas. Exibe-se um sumário dessa atualização, indicando quais outras perícias foram desenvolvidas no processo e que o ciclo de sequenciamento foi completado (Figura 4.5).

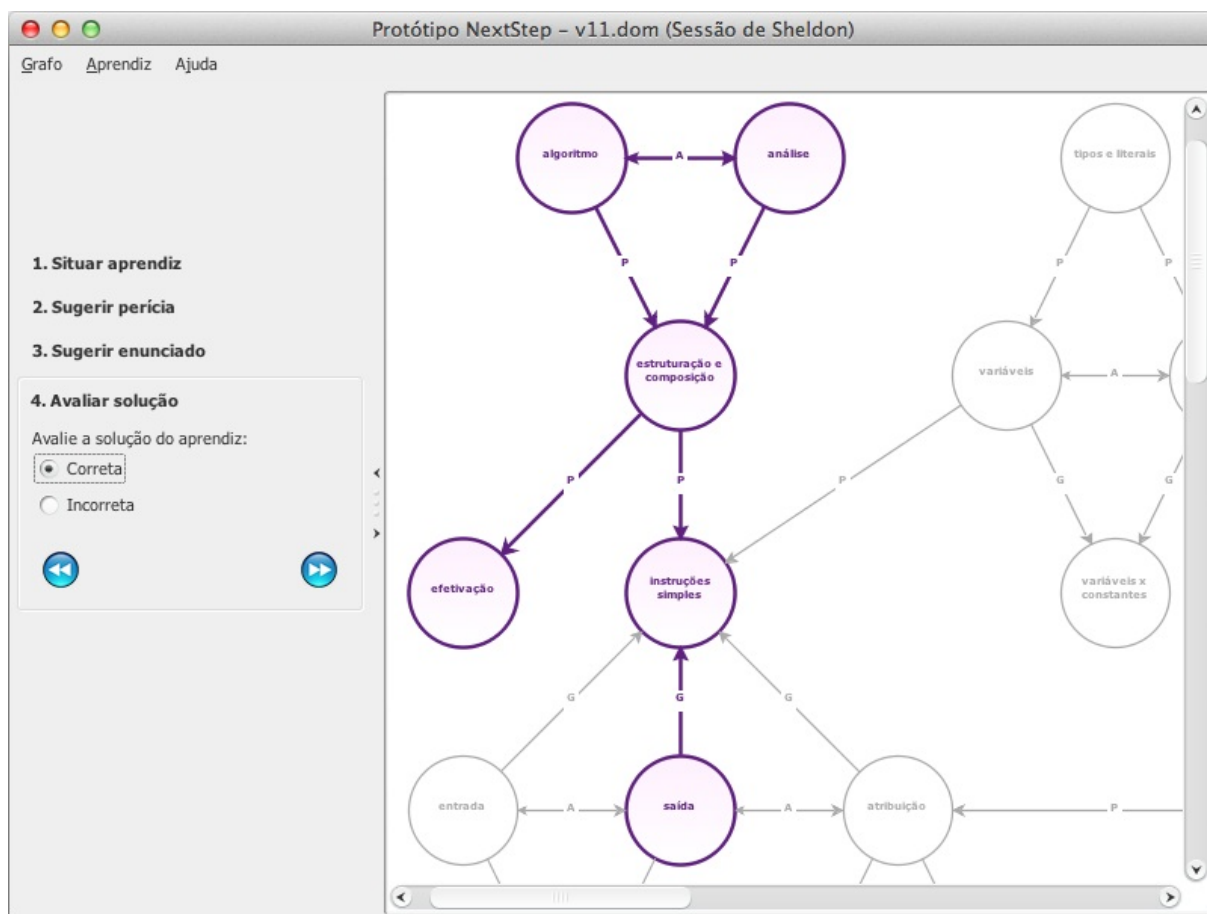


Figura 4.4: Avaliar solução (Primeiro Ciclo de Sequenciamento)

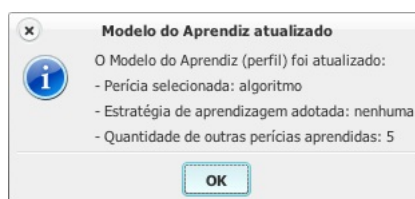


Figura 4.5: Atualizar Modelo do Aprendiz (Primeiro Ciclo de Sequenciamento)

4.2 Segundo Ciclo de Sequenciamento

Considerando o Modelo do Aprendiz atualizado, cabe à ordenação heurística da ferramenta sugerir a próxima perícia para ser desenvolvida e, depois, um enunciado que a atenda. Cada ciclo de sequenciamento começa situando o aprendiz sobre o estado atual dele frente ao conhecimento do domínio (Figura 4.6).

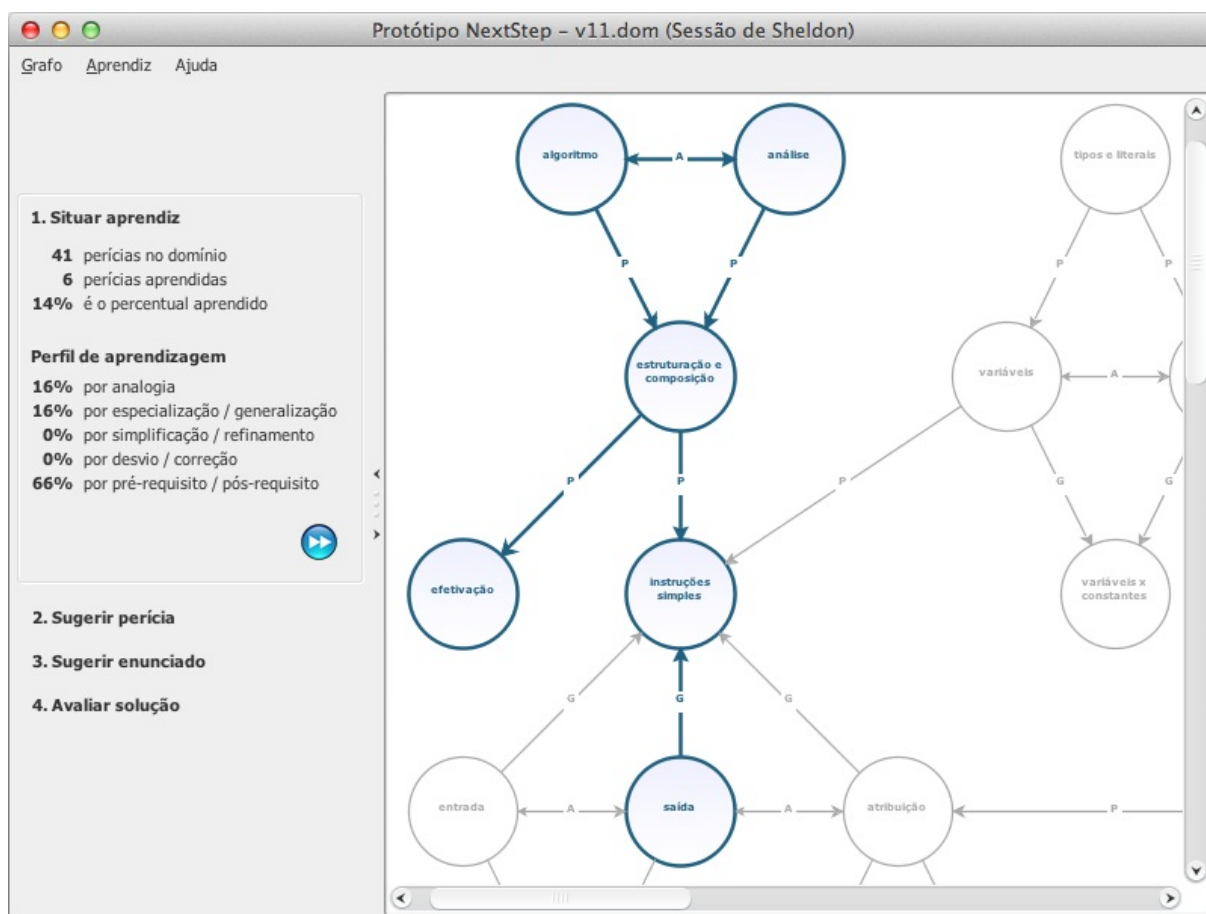


Figura 4.6: Situar aprendiz (Segundo Ciclo de Sequenciamento)

Agora, que a perícia inicial já foi desenvolvida, a ferramenta provê uma lista de prioridades com sugestões de perícias. A ordenação heurística ponderou critérios como a fronteira do Modelo do Aprendiz e o restante do conhecimento do domínio, a distância dessas perícias até aquela definida como inicial e, por fim, as estratégias de tutoria bem-sucedidas para o perfil em particular.

Apresentam-se as sugestões de perícias para escolha do aprendiz. A ferramenta in-

forma, ao lado do identificador de cada perícia sugerida, a distância dela até a perícia inicial e a prioridade definida, respectivamente. A prioridade de uma perícia considera a citada distância e a ponderação das relações genéticas que a conectam ao Modelo do Aprendiz (sendo que, quanto menor o número, maior a prioridade). No momento da seleção, além de cada perícia sugerida ser destacada no conhecimento do domínio, também se indica a estratégia de tutoria mais promissora para desenvolvê-la, conforme o perfil do aprendiz.

Como exemplo, as seguintes sugestões de perícias são propostas ao aprendiz, adicionadas das informações recém-descritas:

1. *variáveis* (3, 20): pré-requisito;
2. *atribuição* (3, 24): analogia;
3. *entrada* (3, 24): analogia; e
4. *entrada × saída* (4, 38): generalização.

Assume-se que a primeira sugestão, *variáveis*, tenha sido selecionada. Logo, deve-se procurar por enunciados que contemplem o reconhecimento de variáveis na análise e na resolução de problemas. Diante do catálogo, tem-se as sugestões abaixo, relacionadas com a quantidade de perícias não desenvolvidas e o total de perícias exercitadas:

1. *Olá usuário* (5, 11);
2. *Metade de um real* (6, 12); e
3. *Gastos em viagem* (8, 13).

A primeira sugestão, *Olá usuário*, foi aceita pelo aprendiz. Ela abrange uma menor quantidade de perícias não desenvolvidas e, por isso, encontra-se mais próxima do seu estágio atual de conhecimento (Zona de Desenvolvimento Proximal). Procedendo-se corretamente com a resolução do exercício, o Modelo do Aprendiz é atualizado a fim de registrar que as perícias exercitadas foram desenvolvidas.

4.3 Terceiro Ciclo de Sequenciamento

Novamente, o aprendiz é situado frente ao conhecimento do domínio (Figura 4.7) e a ordenação heurística sugere uma perícia a ser desenvolvida. Procedendo da mesma maneira que antes, a perícia *expressões aritméticas*, por exemplo, é escolhida. Ela diz respeito à avaliação e construção de expressões aritméticas.

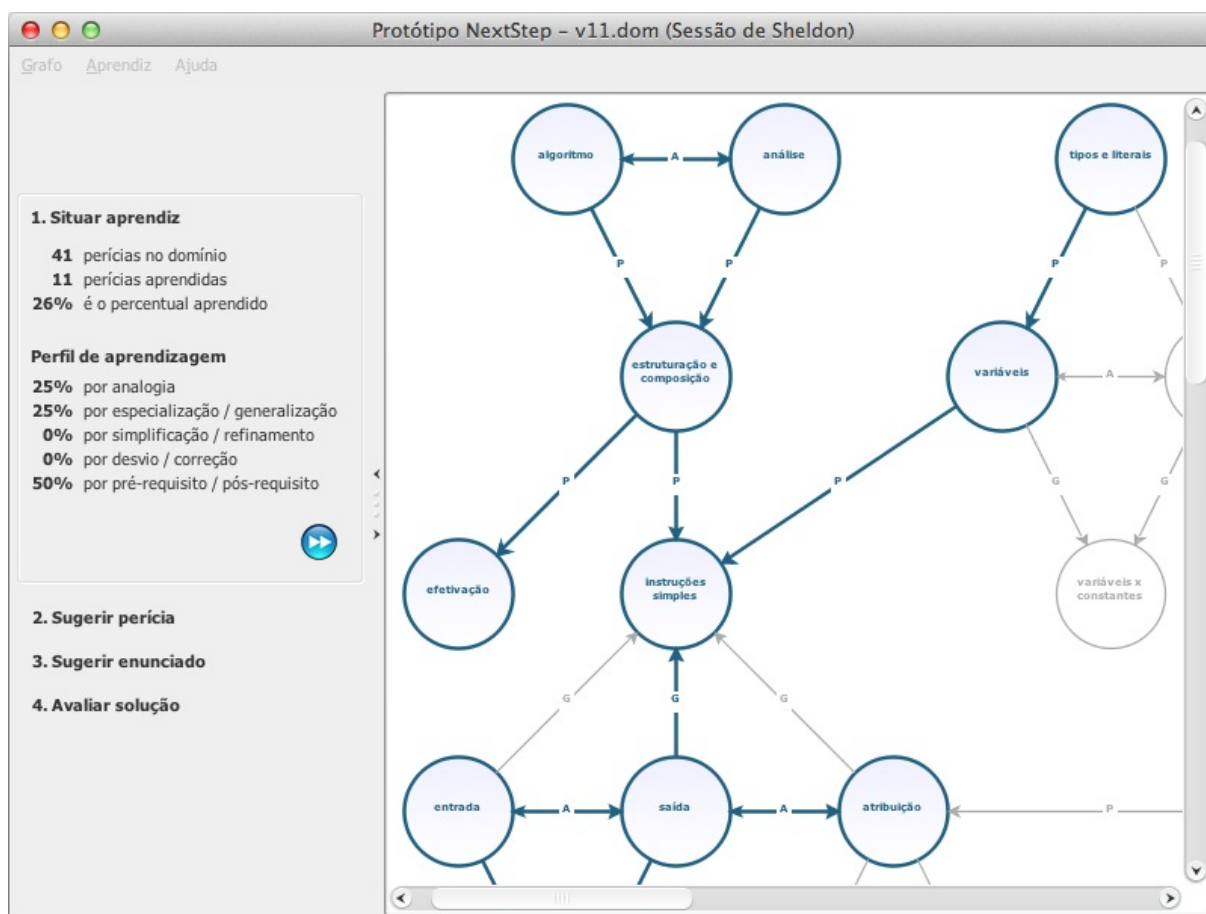


Figura 4.7: Situar aprendiz (Terceiro Ciclo de Sequenciamento)

Os seguintes enunciados do catálogo, por objetivarem essa perícia, são indicados ao aprendiz:

1. *Metade de um real* (1, 11);
2. *Média final* (1, 11); e
3. *Gastos em viagem* (2, 12).

Admite-se a primeira sugestão, *Metade de um real*, como a escolha feita pelo aprendiz. Hipoteticamente, foi constatado insucesso na tentativa do aprendiz resolver o exercício. Possíveis alternativas que tratam da remediação de erros foram desconsideradas na pesquisa. Nesse caso, pode ser escolhido outro enunciado, como *Média final* para desenvolver a mesma perícia em uma nova abordagem. Outra alternativa possível seria retroceder às sugestões de perícias e optar por desenvolver outra ao invés de *expressões aritméticas*.

A ferramenta prossegue avaliando a resolução do aprendiz, atualizando o seu modelo, sugerindo perícias a serem desenvolvidas e enunciados correspondentes.

CAPÍTULO 5

ESTUDO EMPÍRICO

O estudo empírico relatado neste capítulo observa a aderência didática do protótipo, tendo em vista a comparação das sugestões fornecidas pelo processo de sequenciamento com outras dadas por tutores humanos. Considerou-se, especialmente, a proximidade entre os arranjos e a ocorrência de inversões severas, na ordem dos tópicos e dos enunciados sugeridos.

5.1 Sujeitos

Foram sujeitos da pesquisa três docentes¹ da área de Ciência da Computação, todos doutores, que ministram disciplinas introdutórias de Programação de Computadores. Nenhum dos docentes teve envolvimento anterior com a pesquisa.

5.2 Instrumentos

Um questionário com três questões foi utilizado como instrumento para coleta de dados dos docentes. Adicionalmente, foram fornecidos os seguintes materiais, entregues impressos aos docentes:

1. Um texto de apresentação resumindo a pesquisa;
2. O conhecimento do domínio representado em grafo genético, plotado; e
3. O catálogo de enunciados do Anexo B, diagramado no formato de cartões em tamanho A5. Cada cartão continha um enunciado e a região abrangida do grafo genético;

¹Coordenação do Curso de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná (UTFPR), Câmpus Guarapuava.

5.2.1 Questionário

Os questionários entregue aos docentes foi reproduzido integralmente na sequência:

Questão 1

Ordene os enunciados do catálogo na sequência didática que você considere mais conveniente. Entregue os cartões nessa ordem.

Questão 2

Considere que o aprendiz domina o conceito de algoritmo, conhece os tipos primitivos de dados, variáveis, constantes, bem como as instruções de entrada, de saída e de atribuição. Diante do catálogo de enunciados representado pelos cartões, qual seria o próximo enunciado que você recomendaria? Justifique.

Questão 3

Considere o estágio de desenvolvimento do aprendiz ilustrado pela Figura 5.1. Diante do catálogo de enunciados representado pelos cartões, qual seria o próximo enunciado que você recomendaria? Justifique.

5.3 Procedimentos

Detalha-se, a seguir, as orientações fornecidas e o propósito de cada uma das questões componentes. Para comparação, foram realizadas simulações no protótipo em resposta às mesmas questões perguntadas aos docentes. Como procedimento, assumiu-se sempre a primeira sugestão de perícia e de enunciado oferecida pela ferramenta.

5.3.1 Questão 1

Procura saber como os docentes ordenariam didaticamente o mesmo catálogo de enunciados utilizado pela pesquisa. Os cartões foram entregues embaralhados aos docentes, que ficaram livres para considerar a representação em grafo de cada enunciado.

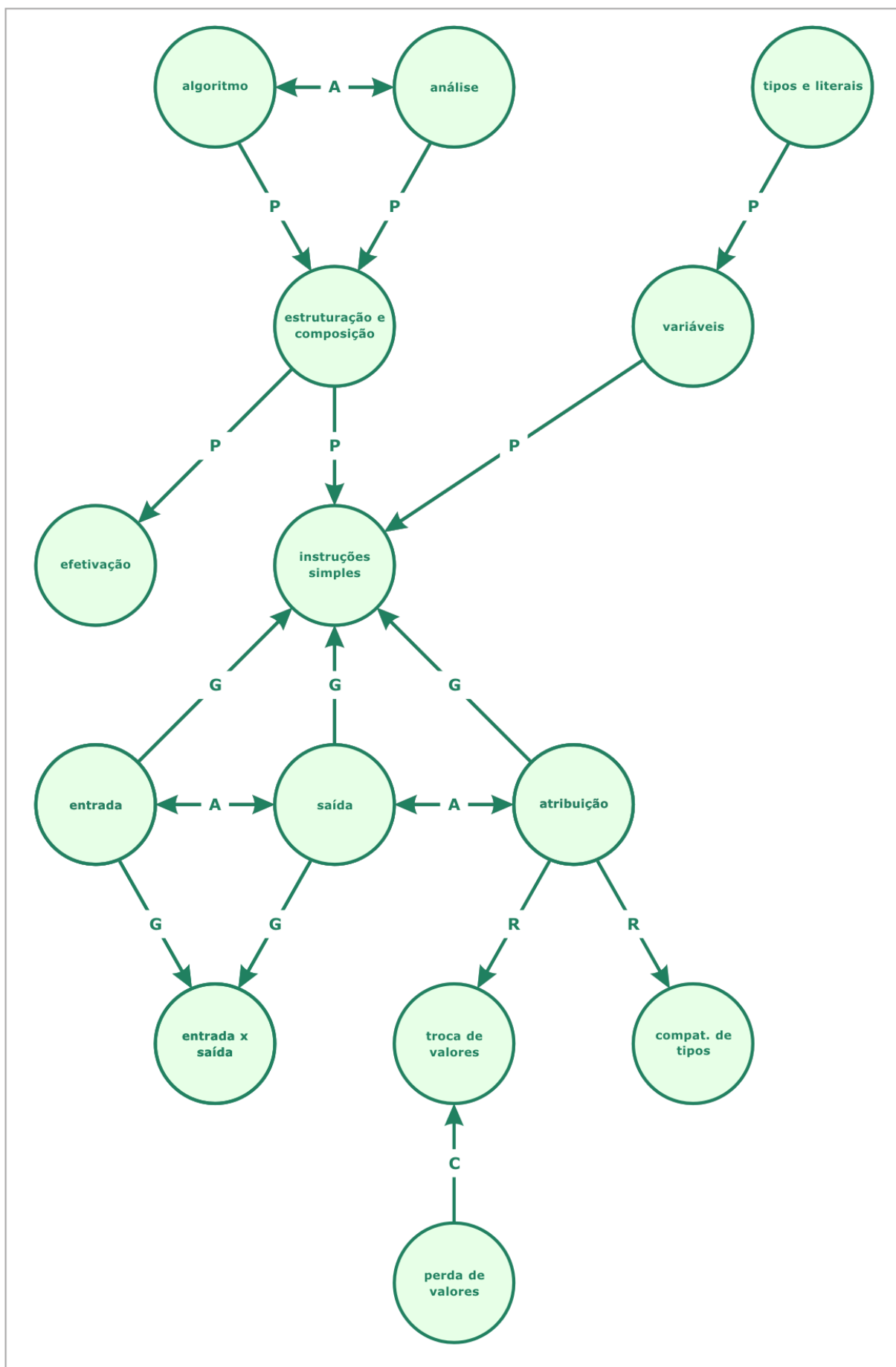


Figura 5.1: Questão 3

5.3.2 Questão 2

Constrói um cenário do conhecimento atual do aprendiz para que o docente recomende o próximo enunciado, dentre aqueles do catálogo. Não faz referência à representação em grafo.

5.3.3 Questão 3

Em contraste com a questão anterior, remete-se diretamente ao grafo genético e induz que o docente decida com base nas mesmas informações assumidas pelo protótipo, dado um cenário para que um enunciado seja sugerido. Propositalmente, a imagem que representa o Modelo do Aprendiz (Figura 5.1) foi privada da sobreposição no conhecimento do domínio. Assim, não se tem o contexto das perícias vizinhas.

5.4 Resultados Coletados

Os resultados foram coletados por meio dos instrumentos de observação especificados na Seção 5.2. São interpretados conforme segue.

5.4.1 Questão 1

A ordenação do catálogo de enunciados pelos docentes consta na Tabela 5.1. Na análise dessas informações, primeiramente foi observado se houve alguma inversão de conteúdos. Pode-se perceber isso porque o catálogo contempla os seguintes conteúdos:

1. Instruções de entrada, de saída e de atribuição;
2. Estruturas de condicionais; e
3. Estruturas de repetição.

Nesses termos, foi examinado se algum enunciado que contemplasse conteúdo posterior foi antecipado. Ou ainda, se ocorreu o contrário, de um enunciado que requereria conteúdo

Tabela 5.1: Resultados dos docentes para a Questão 1

| Conteúdo | Ordem | Professor 1 | Professor 2 | Professor 3 |
|---|-------------------|----------------------------|----------------------------|----------------------------|
| Instruções de Entrada, de Saída e de Atribuição | 1 | Olá mundo | Olá mundo | Olá mundo |
| | 2 | Olá usuário | Olá usuário | Olá usuário |
| | 3 | Metade de um real | Metade de um real | Troca de valores |
| | 4 | Troca de valores | Gastos em viagem | Metade de um real |
| | 5 | Média final | Média final | Média final |
| | 6 | Conta bancária | Conversão para binário | Gasto com viagem |
| | 7 | Gasto em viagem | Conta bancária | Número espelhado |
| | 8 | Caixa eletrônico | Número espelhado | Conversão para binário |
| | 9 | Número espelhado | Caixa eletrônico | Conta bancária |
| | 10 | Conversão para binário | Troca de valores | Caixa eletrônico |
| Estruturas Condicionais | 11 | Valor absoluto | Valor absoluto | Par ou ímpar |
| | 12 | Par ou ímpar | Par ou ímpar | Positivo, negativo ou nulo |
| | 13 | Horário válido | Vogais | Valor absoluto |
| | 14 | Aprovado ou reprovado | Idade | Maior de três valores |
| | 15 | Menor de três valores | Horário válido | Menor de três valores |
| | 16 | Maior de dois valores | Aprovado ou reprovado | Horário válido |
| | 17 | Positivo, negativo ou nulo | Maior de dois valores | Vogais |
| | 18 | Vogais | Menor de três valores | Aprovado ou reprovado |
| | 19 | Tipos de triângulo | Positivo, negativo ou nulo | Tipos de triângulo |
| | 20 | Idade | Tipos de triângulos | Idade |
| Estruturas de Repetição | 21 | Colar na prova | Colar na prova | Colar na prova |
| | 22 | Maioridade | De 1 até 100 | De 50 até 200 |
| | 23 | Soma dos números | De 50 até 200 | De 1 até 100 |
| | 24 | De 50 até 200 | Soma dos números | Tabuada do 5 |
| | 25 | De 1 até 100 | Tabuada do 5 | Tabuadas do 1 ao 10 |
| | 26 | Tabuada do 5 | Potenciação | 1/n |
| | 27 | Tabuadas do 1 ao 10 | Fatorial | Potenciação |
| | 28 | Potenciação | Tabuadas do 1 ao 10 | Fatorial |
| | 29 | Fatorial | Maior média | Soma dos números |
| | 30 | Menor valor | Número primo | Maioridade |
| | 31 | Pares e ímpares | Primos após o 100 | Pares e ímpares |
| | 32 | Menor e maior peso | Maioridade | Menor valor |
| | 33 | Validação | Pares e ímpares | Menor e maior peso |
| | 34 | Calculadora | Menor valor | Média dos pares |
| | 35 | Média dos pares | Menor e maior peso | Validação |
| | 36 | Maior média | Calculadora | Calculadora |
| | 37 | Fibonacci | Validação | Número primo |
| | 38 | 1/n | Média dos pares | Primos após o 100 |
| 39 | Número primo | Fibonacci | Maior média | |
| 40 | Primos após o 100 | 1/n | Fibonacci | |

mais básico ser prorrogado. Esses casos não aconteceram e a mesma tabela foi seccionada em três partes, em correspondência respectiva aos conteúdos citados.

Depois, constatou-se que o sequenciamento dos enunciados que contemplam um determinado conteúdo é bastante sensível à didática do docente. Assim, por exemplo, o enunciado *Vogais*, que remete ao conceito de estrutura condicional com seleção múltipla, variou bastante na sequência, porque o Professor 1 preferiu postergar esse conteúdo, enquanto os outros dois decidiram antecipá-lo.

A proposta desta questão aos docentes desempenhou um papel importante na pesquisa. Ela indicou a necessidade de alterações na primeira versão do protótipo porque, em algumas situações, havia antecipação significativa de conteúdos. Foi necessário reexaminar sobre como os enunciados estavam anteriormente elicitados na ferramenta.

A última simulação feita com o protótipo, apresentada pela Tabela 5.2, forneceu um sequenciamento próximo aos dos docentes e também não trouxe inversão entre os três conteúdos. O sequenciamento sugerido possui menos enunciados porque, conforme mencionado, a configuração atual do protótipo assume que a resolução correta de um único enunciado já proporciona o desenvolvimento da perícia contemplada. Assim, não são repetidos enunciados que abranjam unicamente perícias já desenvolvidas.

Tabela 5.2: Resultados do protótipo para a Questão 1

| Ordem | Protótipo |
|-------|-----------------------|
| 1 | Olá mundo |
| 2 | Olá usuário |
| 3 | Metade de um real |
| 4 | Reajuste de preço |
| 5 | Troca de valores |
| 6 | Gastos com viagem |
| 7 | Valor absoluto |
| 8 | Horário válido |
| 9 | Vogais |
| 10 | Idade |
| 11 | Menor de três valores |
| 12 | Colar na prova |
| 13 | Validação |
| 14 | Tabuadas do 1 ao 10 |
| 15 | Número primo |

5.4.2 Questão 2

Os três docentes sugeriram um enunciado que introduzisse estruturas condicionais ao aprendiz, a saber:

- Professor 1: *Valor absoluto*;
- Professor 2: *Valor absoluto*;
- Professor 3: *Par ou ímpar*.

As escolhas dos docentes corroboraram a ordenação de enunciados que foi recomendada na questão anterior, uma vez que a sugestão de cada um coincide com o primeiro enunciado que trata o referido conteúdo naquela ordenação (Tabela 5.1). O protótipo alcançou o mesmo comportamento de dois docentes, sugerindo o enunciado *Valor absoluto*.

5.4.3 Questão 3

Diante do conhecimento do domínio, os docentes tinham três opções de perícias:

- *Constantes*;
- *Variáveis \times constantes*;
- *Expressões aritméticas*.

Entretanto, o catálogo do Anexo B não possui enunciado que contemple perícias relacionadas a constantes. Essa situação foi propositalmente avaliada para elaborar o questionário.

Foram sugestões dos docentes:

- Professor 1: *Média final*, por contemplar a perícia *expressões aritméticas*;
- Professor 2: a autoria de um enunciado que introduzisse a perícia *constantess*;
- Professor 3: *Metade de um real*, por contemplar a perícia *expressões aritméticas*.

O protótipo sugere a perícia *constantes* e, no próximo passo, não recomenda enunciados, indicando a necessidade de autoria. Ele permite retroceder e optar pela perícia *expressões aritméticas*, para então sugerir o enunciado *Metade de um real*.

5.5 Discussão dos Resultados

Diante do exposto, a solução proposta obteve resultados bastante próximos àqueles fornecidos por tutores humanos. Destaca-se que não houve inversões de ordem significativas nas sugestões de perícias ou de enunciados feitas pelo protótipo. Além disso, foi notado que a ordem dos enunciados diferiu mesmo entre os docentes sujeitos da pesquisa, entendendo-se ser sensível, em certo grau, à didática de cada tutor humano em particular. A solução, por sua vez, acaba sendo influenciada pela perspectiva com que se elicitam os enunciados, no momento da catalogação, quanto às perícias contempladas (regiões do grafo).

Assim, acredita-se que informações interessantes possam provir de experimentos adicionais. Algumas possibilidades, ainda na comparação com tutores humanos, residem na condução de dois outros experimentos distintos, um excluindo os detalhes da metodologia (tais como terminologia e representação) e outro imergindo os docentes nessas informações. Os resultados seriam utilizados para adequação dos parâmetros e aprimoramento da metodologia proposta, bem como para uma investigação mais profunda do problema.

CAPÍTULO 6

RESULTADOS

Diante do que se apresentou, sumarizam-se abaixo os principais resultados que foram alcançados pela pesquisa:

1. Implementação da modelagem dinâmica do aprendiz, assumindo a representação tanto interna quanto externa por grafos genéticos;
2. Determinação dos critérios a serem considerados pela ordenação heurística, para indicar quais perícias devem ser prioritariamente desenvolvidas pelo aprendiz;
3. Consideração inicial da natureza de cada uma das relações presentes no conhecimento do domínio, na sugestão de perícias a serem tutoradas;
4. Inferência das estratégias de tutoria mais promissoras para um aprendiz em particular;
5. Definição dos critérios que adaptem, ao perfil atual do aprendiz, a ordem dos enunciados que contemplam a perícia sugerida;
6. Atualização do Modelo do Aprendiz, para que passe a refletir o nível atual de conhecimento do indivíduo, considerando o progresso após a resolução de um enunciado;
7. Encaminhamento sobre as possibilidades para que ocorra a avaliação das resoluções do aprendiz, presumindo que seja detectado o desenvolvimento de apenas algumas das perícias contempladas;
8. Implementação de um protótipo de ferramenta que funciona como um simulador para acompanhar o processo de sequenciamento de enunciados, fundamentado na metodologia inteligente e adaptativa proposta;

9. Estudo empírico que relaciona o sequenciamento fornecido pelo protótipo com outros gerados por tutores humanos;
10. Disseminação de resultados parciais da pesquisa no I Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg/2015), evento satélite do IV Congresso Brasileiro de Informática na Educação (CBIE/2015) [34];
11. Obtenção do 3^o lugar entre os melhores artigos publicados no evento citado.

CAPÍTULO 7

CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A pesquisa objetivou tratar do sequenciamento adaptativo de enunciados considerando a modelagem dinâmica do aprendiz, no domínio de Programação de Computadores. O projeto decorreu da observação de uma potencialidade pouco explorada dos estudos de [25], favorecendo um possível espaço de contribuição.

Com esse propósito, o referencial teórico abrangeu os seguintes temas: múltiplas representações externas, modelo de sobreposição, grafos genéticos, aquisição de conhecimento em Programação de Computadores e sequenciamento de enunciados. Diante do levantamento realizado, percebeu-se a quantidade reduzida de pesquisas que abordassem o sequenciamento automático de enunciados em ambientes de ensino. Mais especificamente, foi notado que tais estudos se mostram ainda mais dispersos em se tratando do sequenciamento adaptativo em tutoria inteligente.

Dentre os formalismos adotados na solução, discutiu-se sobre os conceitos de automaticidade, adaptatividade e inteligência no escopo do problema. Depois disso, foi detalhada uma proposta de metodologia inteligente e adaptativa para o sequenciamento de enunciados, baseada em grafos genéticos. Em seguida, expôs-se a dinâmica de funcionamento dessa solução, assumindo o protótipo implementado. A avaliação da metodologia e da ferramenta foi realizada por um estudo empírico.

Como principal resultado, espera-se contribuir para a dinamização do aprendizado na área, por meio da adaptação de sequências de enunciados às particularidades dos aprendizes. Por fim, acredita-se que parte dos resultados alcançados podem se estender à aquisição de conhecimento em outros domínios de natureza prática e complexa.

7.1 Trabalhos Futuros

Indicam-se possíveis trabalhos futuros que podem ampliar a pesquisa realizada:

1. Condução de experimentos adicionais para avaliar os benefícios da solução proposta, bem como a necessidade de possíveis adequações;
2. Avaliação e ajuste de parâmetros do processo a fim de garantir que a adaptatividade não seja tendenciosa;
3. Extensão dos critérios assumidos para a sugestão de perícias a serem tutoradas. Dois critérios que podem ser considerados são a necessidade de reforço [18] e o cometimento de erros pelo aprendiz [17]. O primeiro observaria a demanda de reforço dos conteúdos à proporção do tempo, evitando o esquecimento do que foi aprendido. O outro contemplaria os erros cometidos pelo aprendiz, tentando inferir se provieram da ausência de conceitos, de concepções errôneas, de esquecimento, ou ainda se foram apenas deslizes pontuais;
4. Consideração mais aprofundada da natureza de cada uma das relações do conhecimento do domínio, na sugestão de perícias a serem tutoradas. Como passo subsequente, pode-se explorar padrões de relacionamento entre duplas ou triplas de perícias. Ou ainda, critérios como penalizar a sugestão de tópicos que possuam pré-requisitos indiretos ainda não desenvolvidos;
5. Realização de testes e acoplamento, ao protótipo, do módulo que realiza o diagnóstico automático das resoluções do aprendiz;
6. Estudo e melhoria do processo de avaliação das resoluções do aprendiz, objetivando a detecção de aspectos como o desenvolvimento de apenas algumas perícias do enunciado;
7. Implementação da granularidade para a atualização do Modelo do Aprendiz, pois a resolução de um único enunciado que contemple determinada perícia pode não

significar que ela foi desenvolvida. Deve-se admitir uma progressão mais gradual que denote esse crescimento;

8. Estudo sobre o impacto da remodelagem do conhecimento do domínio, expresso no Anexo A, de modo que perícia sugerida possa se localizar sempre na imediata fronteira do Modelo do Aprendiz, tornando a busca heurística mais fluente. Faz-se isso, por exemplo, invertendo alguns relacionamentos, de modo que uma perícia vizinha se conecte individualmente a cada uma das especializações, ao invés da perícia que as generaliza. Tais artifícios podem ser necessários porque o grafo genético original privilegiou aspectos visuais, enquanto essa outra abordagem exige que mais relações sejam definidas. Ainda assim, toda a semântica do grafo genético original será preservada;
9. Análise da viabilidade de admitir ilhas, recurso do grafo genético estendido, na remodelagem do conhecimento do domínio;
10. Tratamento das perícias iniciais do conhecimento do domínio, considerando que podem envolver enunciados de raciocínio lógico anteriores à utilização de uma linguagem formal (construção de algoritmos ou implementação de programas);
11. Admissão da proficiência do aprendiz, para que não seja necessário sujeitá-lo a todos os enunciados que contribuiriam para o desenvolvimento de perícias já adquiridas;
12. Embarcamento da solução em um ambiente de ensino para promover a tutoria inteligente (e.g. FARMA);
13. Investimento na representação externa de mecanismos adicionais, nesse ambiente, que permitam extrair e apresentar informações úteis ao processo de ensino-aprendizagem (mineração de dados).

REFERÊNCIAS

- [1] Abdul-Rahim Ahmad, Otman Basir, e Khaled Hassanein. Adaptive user interfaces for intelligent e-learning: Issues and trends. *The Fourth International Conference on Electronic Business (ICEB)*, páginas 925–934, 2004.
- [2] Shaaron Ainsworth. A functional taxonomy of multiple representations. *Computers and Education*, 33(2/3):131–152, 1999.
- [3] Shaaron Ainsworth. DeFT: A conceptual framework for considering learning with multiple representations. *Learning and Instruction*, 2006.
- [4] Shaaron Ainsworth e Nicolas Van Labeke. Using a multi-representational design framework to develop and evaluate a dynamic simulation environment. Dynamic Information and Visualisation Workshop, julho de 2002.
- [5] Shaaron Ainsworth e Nicolas Van Labeke. Multiple forms of dynamic representation. *Learning and Instruction*, 14(3):241–255, 2004.
- [6] Rodney A. Brooks. Intelligence without reason. *Computers And Thought, IJCAI-91*, páginas 569–595. Morgan Kaufmann, 1991.
- [7] A. Brown. *Metacognition, Motivation, and Understanding*, capítulo Metacognition, executive control, self-regulation, and other more mysterious mechanisms, páginas 65–116. LEA, Hillsdale, NJ, 1987.
- [8] Richard Cox e Paul Brna. Supporting the use of external representations in problem solving: the need for flexible learning environments. *Journal of Artificial Intelligence in Education*, (6):239–302, 1995.
- [9] Richard Cox e Paul Brna. Twenty years on: Reflections on “supporting the use of external representations in problem solving”. *International Journal of Artificial Intelligence in Education*, 26(1):193–204, 2015.

- [10] Teresa del Soldado e Benedict du Boulay. Implementation of motivational tactics in tutoring systems. *Journal of Artificial Intelligence in Education*, 6(4):337–378, 1995.
- [11] Michel C. Desmarais e Ryan S. J. d. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, volume 22, páginas 9–38. Springer Netherlands, 2012.
- [12] Alexandre Direne. Authoring intelligent systems for teaching visual concepts. *International Journal of Artificial Intelligence in Education*, (1):3–14, 1990.
- [13] Lucia Maria Martins Giraffa. Fundamentos de sistemas tutores inteligentes. Relatório técnico, Pontifícia Universidade Católica do Rio Grande do Sul, 2003.
- [14] Ira P. Goldstein. The genetic graph: A representation for the evolution of procedural knowledge. *International Journal of Man-Machine Studies*, 11(1):51–77, janeiro de 1979.
- [15] Marco A. Gómez-Martín, Pedro P. Gómez-Martín, e Pedro A. González-Calero. Game-based learning as a new domain for case-based reasoning. *1st Workshop on Computer Gaming and Simulation Environments*, 2005.
- [16] Teresa Hübscher-Younger e N. Hari Narayanan. Features of shared student-created representations. *External Representations in AIED: Multiple Forms and Multiple Roles - AI-ED 2001 Workshop*, San Antonio, Texas, EUA, maio de 2001. IOS Press.
- [17] Alexander Robert Kutzke. *Informática Educacional e a Mediação do Erro na Educação: Um Estudo Teórico-Crítico e Uma Proposta de Instrumento Computacional*. Tese de doutorado, Universidade Federal do Paraná, Curitiba, agosto de 2015.
- [18] Vilson J. Leffa. Gamificação adaptativa para o ensino de línguas. *Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación*, páginas 1–12, 2014.
- [19] Maici Duarte Leite. *Arquitetura para Remediação de Erros Baseada em Múltiplas Representações Externas*. Tese de doutorado, Universidade Federal do Paraná, Curitiba, 2013.

- [20] Alan Lesgold, Harriet Rubinson, Paul Feltovich, Robert Glaser, Dale Klopfer, e Yen Wang. *The nature of expertise*, capítulo Expertise in a Complex Skill: Diagnosing X-ray Pictures, páginas 311–342. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, England, 1988.
- [21] Guilherme Medeiros Machado, Vinícius Maran, José Palazzo Moreira de Oliveira, Isabela Gasparini, e Ana Pernas. Uma revisão sistemática sobre as abordagens ubíquas para recomendação educacional: Estariam elas se tornando adaptativas? *Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE)*, páginas 170–179, 2015.
- [22] Nigel Major e Han Reichgelt. *Intelligent Tutoring Systems: Second International Conference, ITS '92 Montréal, Canada, June 10–12 1992 Proceedings*, capítulo COCA: A shell for intelligent tutoring systems, páginas 523–530. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.
- [23] Diego Marczal, Alexandre Direne, Andrey Ricardo Pimentel, e Eleandro Maschio. FARMA: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos. *Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE/2015)*, páginas 23–32, 2015.
- [24] Eleandro Maschio. Abordagem metacognitiva através de múltiplas representações externas para o ensino de programação de computadores. Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba, abril de 2007. Alexandre Ibrahim Direne (orientador).
- [25] Eleandro Maschio. *Modelagem do Processo de Aquisição de Conhecimento Apoiado por Ambientes Inteligentes*. Tese de doutorado, Universidade Federal do Paraná, Curitiba, agosto de 2013.
- [26] Eleandro Maschio e Alexandre Ibrahim Direne. Múltiplas representações externas no suporte à aquisição de conhecimento em programação de computadores. *Revista Brasileira de Informática na Educação (RBIE)*, 23(3):91–96, 2015.

- [27] Viktor Mayer-Schönberger e Kenneth Cukier. *Learning with Big Data: The Future of Education*. Houghton Mifflin Harcourt, 2014.
- [28] Iwona Miliszewska e Grace Tan. *Befriending Computer Programming: A Proposed Approach to Teaching Introductory Programming*, volume 4, páginas 277–289. *Issues in Informing Science and Information Technology*, 2007.
- [29] Antonija Mitrovic. An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education*, (13):173–197, 2003.
- [30] Antonija Mitrovic e Brent Martin. Evaluating the effects of open student models on learning. *Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference*, volume 2347, páginas 296–305, Malaga, Spain, maio de 2002. Springer Berlin Heidelberg.
- [31] Antonija Mitrovic e Brent Martin. Evaluating adaptive problem selection. *Adaptive Hypermedia and Adaptive Web-Based Systems: Third International Conference*, volume 3137, páginas 185–194, Eindhoven, Netherlands, agosto de 2004. Springer Berlin Heidelberg.
- [32] Peter Norvig. Teach yourself programming in ten years. <http://norvig.com/21-days.html>, 2001.
- [33] Hyacinth S. Nwana. Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, (4):251–277, 1990.
- [34] Carolina Moreira Oliveira, Andrey Ricardo Pimentel, e Eleandro Maschio. Estudo sobre o sequenciamento inteligente e adaptativo de enunciados em programação de computadores. *Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE 2015)*, páginas 1320–1329, 2015.
- [35] Stephen E. Palmer. Fundamental aspects of cognitive representation. páginas 259–303, 1978.

- [36] Fillipi Domingos Pelz. Correção automática de algoritmos no ensino introdutório de programação. Dissertação de Mestrado, Universidade do Vale do Itajaí, Itajaí, junho de 2011. André Luis Alice Raabe (orientador).
- [37] Jean Piaget. *Epistemologia Genética*. WMF Martins Fontes, 2002.
- [38] Andrey Ricardo Pimentel e Alexandre Ibrahim Direne. Medidas cognitivas no ensino de programação de computadores com sistemas tutores inteligentes. *Anais do IX Simpósio Brasileiro de Informática na Educação. SBIE 1998*, volume 9, páginas 206–215, Fortaleza, CE, Brasil, 1998. Sociedade Brasileira de Computação.
- [39] Emil Protalinski. 100m users strong, duolingo raises 45m led by google at a 470m valuation to grow language-learning platform. <http://venturebeat.com/2015/06/10/100m-users-strong-duolingo-raises-45m-led-by-google-at-a-470m-valuation-to-grow-language-learning-platform/>, 2015.
- [40] André Luís Alice Raabe. *Uma Proposta de Arquitetura de Sistema Tutor Inteligente Baseada na Teoria das Experiências de Aprendizagem Mediadas*. Tese de doutorado, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005.
- [41] José A. Ruipérez-Valiente, Pedro J. Muñoz-Merino, Derick Leony, e Carlos Delgado Kloos. Alas-ka: A learning analytics extension for better understanding the learning process in the khan academy platform. *Computers in Human Behavior*, 47:139 – 148, 2015.
- [42] Roger Schank e Chip Cleary. *Engines for Education*. Lawrence Erlbaum Associates Publishers, Hillsdale, NJ, England, 1995.
- [43] Robert Sedgewick e Kevin Wayne. *Algorithms*. Addison-Wesley, Boston, MA, 4 ed., 2011.
- [44] Rômulo César Silva. *Sequenciamento Adaptativo de Exercícios Baseados na Correspondência entre a Dificuldade da Solução e o Desempenho Dinâmico do Aprendiz*. Tese de doutorado, Universidade Federal do Paraná, Curitiba, agosto de 2015.

- [45] Marcelo A. S. Turine, Marcus Vinicius Maltempi, e Ricardo Hasegawa. Sistemas tutores inteligentes: uma revisão descritiva. Relatório técnico, Universidade de São Paulo, 1994.
- [46] Lev Vygotsky. *Interaction between learning and development*, páginas 34–41. Readings on the Development of Children, 1978.
- [47] Gerhard Weber e Peter Brusilovsky. Elm-art: An adaptive versatile system for web-based instruction. *International Journal of Artificial Intelligence in Education*, volume 12, páginas 351–384, 2001.
- [48] Gerhard Weber e Antje Möllenberg. *Cognition and Computer Programming*, capítulo 14 - ELM programming environment: A tutoring system for LISP beginners, páginas 373–408. Ablex Publishing Corporation, Norwood, New Jersey,, 1995.
- [49] Etienne Wenger. *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann, 1987.
- [50] Dinesha Samanthi Weragama. *Intelligent Tutoring System for Learning PHP*. Tese de doutorado, Queensland University Of Technology, Brisbane, Austrália, 2013.
- [51] Adriana Salvador Zanini. Avaliação da influência dos enunciados na resolução de problemas de programação introdutória. Dissertação de Mestrado, Universidade do Vale do Itajaí, Itajaí, fevereiro de 2013. André Luis Alice Raabe (orientador).
- [52] Jiajie Zhang. The nature of external representations in problem solving. *Cognitive Science: A Multidisciplinary Journal*, (21):179–217, 1997.

ANEXO A

SUBCONJUNTO DE PERÍCIAS NO DOMÍNIO DE PROGRAMAÇÃO DE COMPUTADORES

O subconjunto de perícias no domínio de Programação de Computadores foi modelado por [25] com a intenção de fornecer linhas gerais para que sejam adaptadas às peculiaridades de cada linguagem de programação. Considerou-se o paradigma Imperativista.

1. Domínio do conceito de algoritmo.
 - (a) Conceito de algoritmo. Entendimento do conceito de algoritmo como uma sequência finita de instruções a fim de realizar determinada tarefa.
 - (b) Conceito de instrução.
 - (c) Entendimento do fluxo sequencial de execução.
2. Análise e abstração de informações trazidas pelo enunciado do problema.
 - (a) Leitura atenta do enunciado.
 - (b) Identificação do problema.
 - (c) Identificação da entrada, do processamento e da saída do problema.
3. Composição e estruturação do algoritmo.
 - (a) Domínio da estrutura de um algoritmo.
 - (b) Refinamento sucessivo das tarefas às instruções para resolver o problema determinado.
 - (c) Transcrição do algoritmo em linguagem natural ou formal.
4. Efetivação do algoritmo.
 - (a) Simulação (mental) do algoritmo alcançado.
 - (b) Simulação do algoritmo através de testes.
 - (c) Retroalimentação, correção e aperfeiçoamento do algoritmo.
5. Distinção entre diferentes tipos de dados e literais.
 - (a) Booleano.
 - (b) Inteiro.
 - (c) Real.
 - (d) Caractere.
 - (e) Texto.
 - (f) Compreensão da Tabela ASCII.

- i. Distinção entre maiúsculas e minúsculas.
 - ii. Caracteres especiais.
6. Reconhecimento de variáveis na análise e na resolução do problema.
 - (a) Domínio do conceito de variável.
 - (b) Sintaxe da declaração de variáveis.
 - (c) Identificação do tipo de dado correspondente.
 - (d) Definição de nomes apropriados para variáveis.
 - (e) Declaração de variáveis.
 - i. Declaração de variáveis do tipo booleano.
 - ii. Declaração de variáveis do tipo inteiro.
 - iii. Declaração de variáveis do tipo real.
 - iv. Declaração de variáveis do tipo caractere.
 - v. Declaração de variáveis do tipo texto.
7. Reconhecimento de constantes na análise e na resolução do problema.
 - (a) Domínio do conceito de constante
 - (b) Sintaxe da declaração de constantes.
 - (c) Identificação do tipo de dado correspondente.
 - (d) Convenções para a denominação de constantes.
 - (e) Declaração de constantes.
 - i. Declaração de constantes do tipo booleano.
 - ii. Declaração de constantes do tipo inteiro.
 - iii. Declaração de constantes do tipo real.
 - iv. Declaração de constantes do tipo caractere.
 - v. Declaração de constantes do tipo texto.
8. Distinção conceitual e pragmática entre variáveis e constantes.
9. Instruções simples.
10. Domínio sobre a instrução de atribuição.
 - (a) Conceito de atribuição.
 - (b) Sintaxe da instrução de atribuição.
 - i. Atribuição de variáveis do tipo inteiro.
 - ii. Atribuição de variáveis do tipo real.
 - iii. Atribuição de variáveis do tipo caractere.
 - iv. Atribuição de variáveis do tipo texto.
 - v. Atribuição de variáveis do tipo booleano.
 - (c) Conceito de inicialização.
 - i. Inicialização de variáveis do tipo inteiro.
 - ii. Inicialização de variáveis do tipo real.
 - iii. Inicialização de variáveis do tipo caractere.

- iv. Inicialização de variáveis do tipo texto.
 - v. Inicialização de variáveis do tipo booleano.
11. Troca de valores entre duas variáveis com o uso de uma auxiliar.
 12. Sobreposição e consequente perda de valores armazenados.
 13. Cautela com a compatibilidade de tipos.
 - (a) Incorrência de tipos incompatíveis.
 - (b) Resultado da divisão de dois inteiros.
 - (c) Conversão de tipos.
 14. Domínio sobre a instrução de entrada.
 - (a) Conceito de entrada.
 - (b) Sintaxe da instrução de entrada.
 - i. Entrada de uma variável do tipo inteiro.
 - ii. Entrada de uma variável do tipo real.
 - iii. Entrada de uma variável do tipo texto.
 - iv. Entrada de uma variável do tipo caractere.
 - v. Entrada de uma variável do tipo booleano.
 - (c) Entrada de múltiplos valores.
 - (d) Captura de teclas (especiais).
 15. Domínio sobre a instrução de saída.
 - (a) Conceito de saída.
 - (b) Sintaxe da instrução de saída.
 - i. Saída de literais.
 - ii. Saída de caracteres especiais.
 - iii. Saída de variáveis.
 - iv. Saídas compostas.
 - v. Sequências de escape.
 - vi. Formatação da saída.
 - A. Alinhamento.
 - B. Separador de milhares.
 - C. Decimais.
 16. Distinção conceitual e pragmática entre instruções de entrada e de saída.
 17. Avaliação e construção de expressões compostas.
 - (a) Precedência de operadores aritméticos, relacionais e booleanos.
 - (b) Parênteses.
 - i. Alteração da precedência de operadores através de parênteses.
 - ii. Desambiguação de expressões através de parênteses.
 18. Avaliação e construção de expressões aritméticas.

- (a) Operadores aritméticos.
 - i. Adição.
 - ii. Subtração.
 - iii. Multiplicação.
 - iv. Divisão.
 - v. Divisão inteira.
 - vi. Resto.
 - (b) Precedência dos operadores aritméticos.
 - (c) Avaliação de expressões aritméticas.
 - (d) Pragmática destacada:
 - i. Quadrado de um número.
 - ii. Média aritmética.
 - A. Média aritmética simples.
 - A soma precede a divisão.
 - B. Média aritmética composta.
 - A soma dos produtos precede a divisão.
 - iii. Porcentagem.
 - A. Acréscimo percentual.
 - B. Decréscimo percentual.
 - iv. Regra de três.
 - A. Regra de três simples.
 - B. Regra de três composta.
 - v. Ordem dos algarismos de um inteiro.
 - A. Decomposição dos dígitos de um inteiro através dos operadores DIV e MOD.
 - B. Recomposição dos dígitos de um inteiro através do operador de multiplicação.
19. Incorrência da divisão por zero.
20. Avaliação e construção de expressões relacionais.
- (a) Resultado de expressões relacionais.
 - (b) Operadores relacionais.
 - i. Igual.
 - ii. Diferente.
 - iii. Maior que.
 - iv. Menor que.
 - v. Maior ou igual.
 - vi. Menor ou igual.
 - (c) Precedência dos operadores relacionais.
 - (d) Avaliação de expressões relacionais.
21. Avaliação e construção de expressões booleanas.
- (a) Resultado de expressões booleanas.

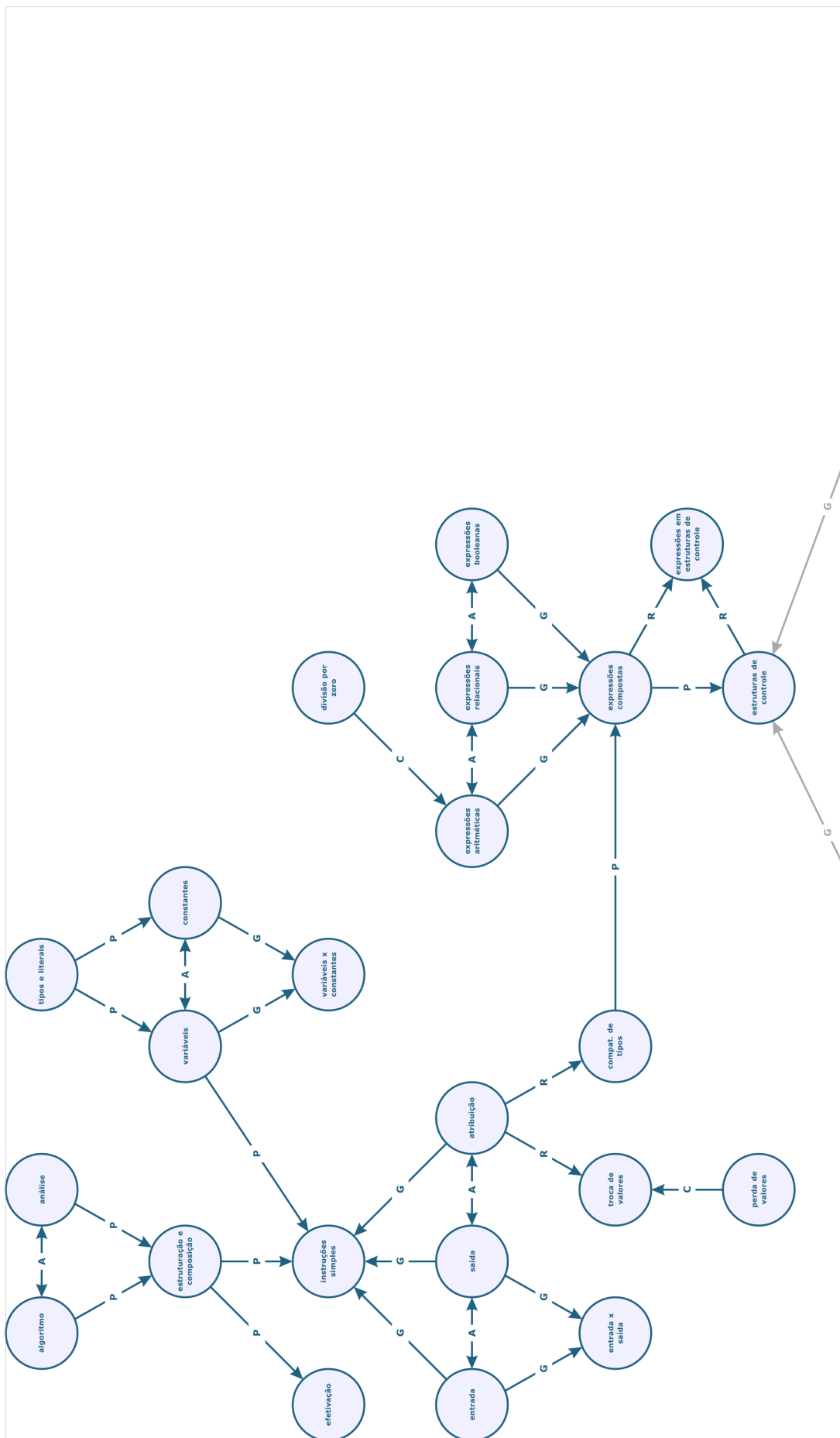
- (b) Operadores booleanos e respectivas tabelas verdade.
 - i. Negação.
 - ii. Conjunção.
 - iii. Disjunção.
 - iv. Disjunção exclusiva.
 - A. Emulação da disjunção exclusiva através dos operadores booleanos fundamentais.
 - (c) Precedência dos operadores booleanos.
 - (d) Avaliação de expressões booleanas.
22. Projeto de (conjuntos de) expressões condicionais coesas e concisas para estruturas de controle.
- (a) Entendimento de que expressões condicionais precisam ter resultado booleano.
 - (b) Cuidado com a repetição indevida de condições face à exclusividade do fluxo de execução.
 - (c) Distinguição entre maiúsculas e minúsculas na construção de expressões condicionais, envolvendo os tipos caractere ou texto.
23. Domínio sobre estruturas de controle.
- (a) Composição de subcontextos de instruções coesos e concisos para estruturas de controle.
24. Domínio sobre as estruturas condicionais.
- (a) Conceito de desvio condicional progressivo.
 - i. Entendimento da necessidade de controle do fluxo de execução.
 - (b) Pragmática destacada:
 - i. Determinação de datas válidas.
 - ii. Reconhecimento de vogais e consoantes.
 - (c) Composição de subcontextos de instruções coesos e concisos para estruturas condicionais.
 - i. Evitamento da repetição indevida de instruções.
25. Domínio sobre as estruturas condicionais simples e composta.
- (a) Semântica das estruturas condicionais simples e composta.
 - i. Compreensão da existência de dois caminhos possíveis e exclusivos.
 - (b) Sintaxe das estruturas condicionais simples e composta.
 - i. Indentação.
 - ii. Blocos com uma única instrução.
 - (c) Pragmática destacada:
 - i. Classificação de um número como nulo ou não-nulo.
 - ii. Classificação de um número como positivo, negativo ou nulo.
 - A. Módulo ou valor absoluto de um número.
 - iii. Reconhecimento de multiplicidade numérica.

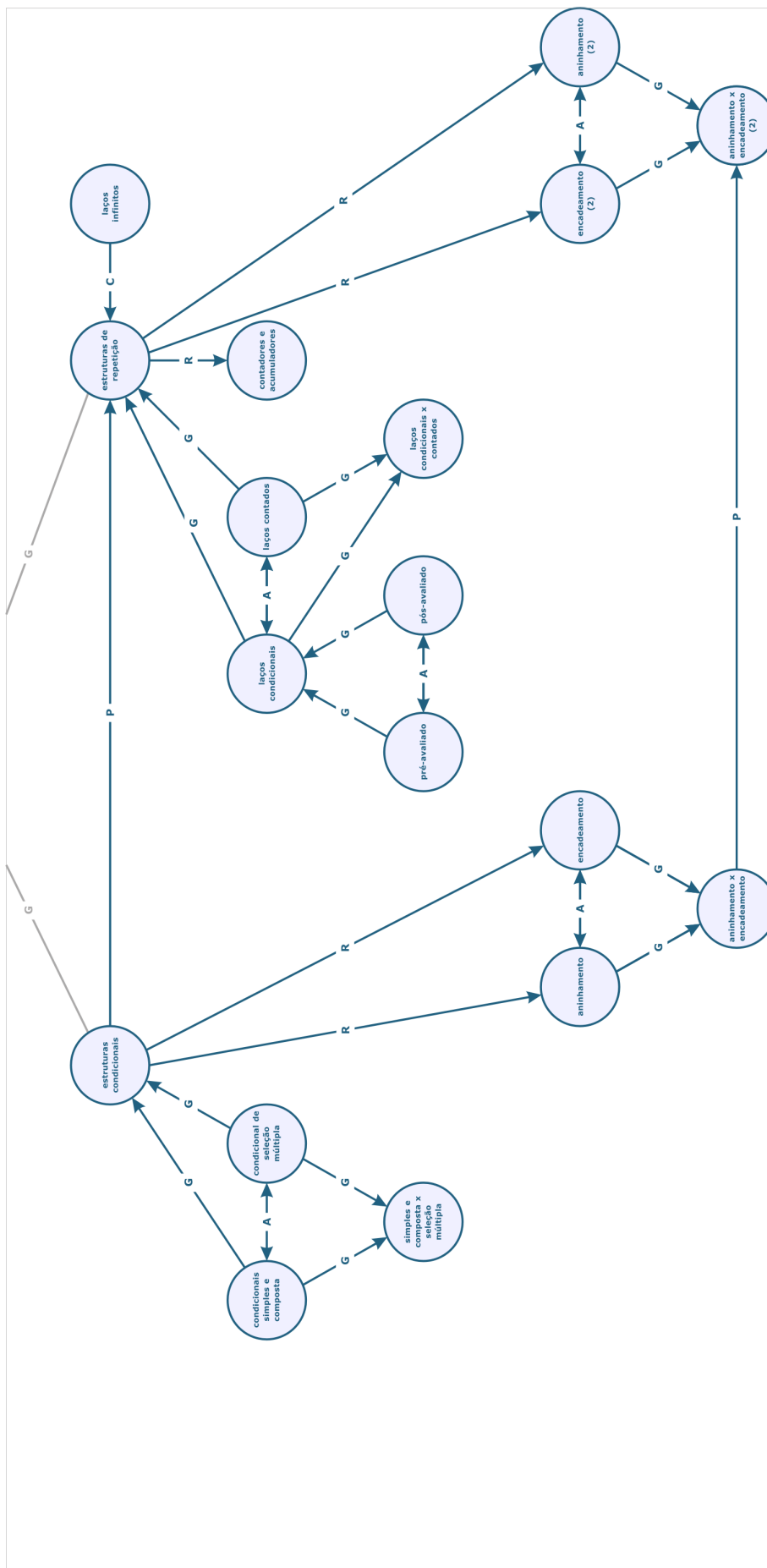
- A. Reconhecimento de um número par (ou ímpar).
 - iv. Determinação de que uma data é posterior (ou anterior) a outra.
 - v. Verificação de anos bissextos.
 - vi. Validação de entradas.
 - A. Determinar se uma data é válida.
 - vii. Determinação do maior (ou menor) valor.
 - A. Entre dois valores.
 - B. Entre três valores.
 - C. Entre quatro valores.
 - viii. Ordenação.
 - A. De dois valores.
 - B. De três valores.
26. Domínio sobre a estrutura condicional de seleção múltipla.
- (a) Semântica da estrutura condicional de seleção múltipla.
 - i. Restrição de tratar somente tipos contáveis.
 - A. Inteiro.
 - B. Caractere.
 - C. Booleano.
 - ii. Impossibilidade de tratar tipos incontáveis.
 - A. Real.
 - B. Texto.
 - iii. Uso do caso-contrário.
 - (b) Sintaxe da estrutura condicional seleção múltipla.
 - i. Seleção de um valor específico.
 - ii. Seleção de uma escala de valores.
 - iii. Seleção de uma lista de valores.
 - iv. Seleção composta.
 - v. Indentação.
 - (c) Reconhecimento de problemas resolvíveis pela estrutura condicional de seleção múltipla.
27. Distinção conceitual e pragmática entre as estruturas condicionais da linguagem utilizada.
- (a) Entendimento da generalidade das estruturas condicionais simples e composta.
 - (b) Entendimento da especificidade da estrutura condicional de seleção múltipla.
 - (c) Reconhecimento de situações que privilegiem uma ou outra estrutura condicional.
28. Distinção e concomitância de aninhamento e encadeamento de estruturas condicionais.
- (a) Importância da indentação.
29. Aninhamento de estruturas condicionais para classificações exclusivas.
30. Encadeamento de estruturas condicionais para classificações não-exclusivas.
31. Domínio sobre as estruturas de repetição.

- (a) Conceito de laço ou desvio regressivo.
 - i. Entendimento da necessidade de controle do fluxo de execução.
 - (b) Conceito de subcontextos ciclados.
 - (c) Composição de subcontextos de instruções coesos e concisos para estruturas de repetição.
 - i. Evitamento da repetição indevida de instruções.
 - ii. Noções básicas de complexidade temporal de algoritmos.
 - (d) Pragmática destacada:
 - i. Validação e repetição de entradas.
 - ii. Repetição de entradas.
 - A. Leitura de uma quantidade determinada de valores.
 - B. Leitura de uma quantidade indeterminada de valores.
 - iii. Potenciação.
 - iv. Fatorial.
 - v. Séries numéricas.
 - A. Séries numéricas com um único sinal.
 - B. Séries numéricas com intercalação de sinais.
 - vi. Conjuntos finitos.
 - A. Determinação do maior (ou menor) valor de um conjunto finito.
 - Atentamento à inicialização da variável que armazena o maior (ou menor) valor.
 - Desconsideração do valor de escape quando se tem uma quantidade de valores previamente indeterminada.
 - B. Média aritmética simples de um conjunto numérico finito.
 - Entender a necessidade de dividir a somatória somente depois de ter acumulado todos os valores do conjunto.
 - C. Determinação de elementos com propriedades específicas (pares, ímpares, positivos, negativos).
 - Número de elementos com propriedades específicas.
 - Percentual de elementos com propriedades específicas.
 - vii. Verificação de números primos.
 - A. Verificação suficiente de números primos.
 - B. Verificação eficiente de números primos.
32. Domínio sobre os laços condicionais.
33. Domínio sobre a estrutura de repetição pré-avaliada.
- (a) Semântica da estrutura de repetição pré-avaliada.
 - i. Compreensão do pré-teste.
 - ii. Entendimento de que a ciclagem ocorre incondicionalmente ao final do bloco de instruções do subcontexto.
 - (b) Sintaxe da estrutura de repetição pré-avaliada.
 - i. Blocos com uma única instrução.
 - ii. Indentação.

34. Domínio sobre a estrutura de repetição pós-avaliada.
- (a) Semântica da estrutura de repetição pós-avaliada.
 - i. Compreensão do pós-teste.
 - ii. Entendimento de que primeira repetição do bloco de instruções do subcontexto ocorre incondicionalmente à avaliação do condicional.
 - (b) Sintaxe da estrutura de repetição pós-avaliada.
 - i. Blocos com uma única instrução.
 - ii. Indentação.
 - (c) Otimização do uso de estruturas de repetição.
 - i. Interromper a ciclagem tão logo o objetivo seja satisfeito.
 - ii. Identificar e evitar passos e ciclagens desnecessários.
 - A. Composição de expressões condicionais eficientes.
 - iii. Percepção de que subcontextos aninhados aumentam exponencialmente a complexidade de tempo em um trecho de código.
35. Domínio sobre laços contados.
- (a) Estrutura de repetição com variável de controle.
 - i. Semântica da estrutura com variável de controle.
 - A. Conceito de variável de controle.
 - Risco de se manipular a variável de controle no subcontexto da estrutura de repetição.
 - B. Contagem progressiva.
 - C. Contagem regressiva.
 - D. Incremento (ou passo).
 - ii. Sintaxe da estrutura de repetição específica de laços contados.
 - A. Parâmetros.
 - Variável de controle.
 - Valor inicial.
 - Valor final.
 - Valor do incremento (ou passo).
 - Indicador de contagem regressiva.
 - B. Blocos com uma única instrução.
 - C. Indentação.
36. Domínio conceitual e pragmático de contadores e acumuladores.
- (a) Contadores.
 - i. Conceito de contador.
 - ii. Inicialização de um contador.
 - A. Inicialização de um contador para uso em laços condicionais.
 - B. Inicialização de um contador com o uso de laços contados.
 - iii. Atualização de um contador.
 - A. Atualização de um contador em laços condicionais.
 - B. Atualização de um contador em laços contados.

- (b) Acumuladores.
 - i. Conceito de somas e produtos acumulativos.
 - ii. Inicialização de um acumulador.
 - iii. Atualização de um acumulador.
- 37. Incorrência de laços infinitos.
 - (a) Identificação de laços infinitos.
- 38. Distinção e concomitância de aninhamento e encadeamento de estruturas de repetição.
 - (a) Importância da indentação.
- 39. Aninhamento de estruturas de repetição.
- 40. Encadeamento de estruturas de repetição.
- 41. Distinção conceitual, pragmática e intercambiamento entre as estruturas de repetição da linguagem utilizada.
 - (a) Problemas de contagem.
 - i. Identificação de problemas que envolvam contagem (inclusive séries).
 - ii. Favorecimento dos problemas que envolvam contagem, pela estrutura de repetição com variável de controle.
 - iii. Restrição da estrutura de repetição com variável de controle aos problemas estritamente de contagem.
 - (b) Pré e pós-avaliação do condicional.
 - i. Distinção de quando uma resolução privilegia a pré ou a pós-avaliação do condicional em uma estrutura de repetição.
 - (c) Intercambiamento entre estruturas de repetição.
 - i. Entendimento de que as estruturas pré e pós-avaliada podem ser intercambiáveis e resolver qualquer problema de repetição (inclusive aqueles solucionáveis pela estrutura de repetição com variável de controle).
 - ii. Limitação do intercambiamento ao subconjunto dos problemas de contagem à estrutura de repetição com variável de controle.





ANEXO B

CATÁLOGO DE ENUNCIADOS

O presente catálogo subsidia o desenvolvimento de perícias no subconjunto do domínio de Programação de Computadores considerado pela tese. Consiste em uma adaptação de [36] que propõe um grupo de exercícios para abordar em extensão o conteúdo programático tradicional de disciplinas correspondentes. Observou-se, em acréscimo, resultados parciais que perfazem continuidade à pesquisa recém-citada e, por enquanto não disseminados, foram cordialmente cedidos pelo professor André Raabe (Univali).

Convém a advertência de que cada enunciado pode contemplar regiões ligeiramente diferentes das capacidades do domínio, em consequência das peculiaridades sintáticas e semânticas de cada linguagem de programação Imperativista. Por exemplo, há distinção entre estruturas condicionais de seleção múltipla, quando remetidas às linguagens C ou Pascal. O mesmo ocorre com estruturas de repetição para laços contados.

A fim de orientar a composição de novos enunciados, sugere-se consultar [51] que fez uma avaliação detalhada sobre a estrutura e o contexto usualmente empregados pelos exercícios.

Instruções Simples

1. **Olá, mundo**

Exiba a mensagem “Olá, mundo!” na tela.

2. **Olá, usuário**

Leia o nome do usuário e exiba-o, na saída, antecedido pela mensagem “Olá”.

3. **Metade de um real**

Leia um número real e exiba a metade do valor digitado.

4. **Gastos em viagem**

Calcule os gastos com combustível em uma viagem. Serão fornecidos: a distância a ser percorrida (em quilômetros), a média de consumo do carro (em quilômetros por litro) e o preço do combustível utilizado (em reais por litro). Exiba o valor, em reais, estimado dos gastos com combustível na viagem.

5. **Troca de valores**

Leia dois números inteiros, a e b , e faça com que eles troquem os valores entre si. Exiba os valores de a e b .

6. **Média final**

O sistema de avaliação de uma disciplina é composto por três provas com pesos respectivos

de 2, 4 e 6. Leia as notas obtidas por um acadêmico em cada prova e exiba a média final obtida.

7. **Caixa eletrônico**

Simule o saque em um caixa eletrônico. Leia o valor a ser retirado e apresente a quantidade de cada cédula que será entregue ao usuário. Assuma que o equipamento possui cédulas de 10, 5 e 1 reais. Utilize o menor número possível de cédulas.

8. **Conta bancária**

Dado o número de uma conta corrente com quatro dígitos, retorne o dígito verificador correspondente, calculado da seguinte maneira:

- Some os quatro dígitos;
- Multiplique os quatro dígitos,
- Subtraia o segundo resultado do primeiro;
- O dígito verificador será o resto da divisão do último resultado por 9.

9. **Número espelhado**

Dado um número no formato CDU, apresente-o espelhado: UDC. Por exemplo: 123 resultará em 321. O número deverá ser armazenado em outra variável antes de ser exibido.

10. **Conversão para binário**

Leia um número decimal, converta-o para binário e exiba o resultado. Considere que a entrada consistirá apenas dos números de 0 a 15.

Estruturas Condicionais

1. **Valor absoluto**

Leia um número inteiro e exiba-o na saída. Se o número digitado for negativo, transforme-o no equivalente positivo (módulo) antes de apresentá-lo.

2. **Par ou ímpar**

Leia um número inteiro e exiba se o valor é par ou ímpar.

3. **Vogais**

Leia uma letra e retorne se esta é, ou não, uma vogal. Atente-se às maiúsculas e minúsculas. Considere que a entrada consistirá apenas de letras sem acentuação ou sinalização.

4. **Idade**

Leia a data de nascimento de uma pessoa e a data atual. Calcule e exiba a idade do indivíduo em anos. Cada data é representada por três variáveis inteiras distintas e correspondentes ao dia, mês e ano.

5. **Horário válido**

Leia dois números inteiros que correspondem, respectivamente, às grandezas de horas e minutos de um suposto relógio. Retorne se o horário informado é válido. Considere a hora 24 como inválida.

6. **Maior de dois números**

Exiba o maior dentre dois números reais lidos. Identifique também e informe caso os números sejam iguais.

7. **Positivo, negativo ou nulo**

Leia um número inteiro e exiba se o valor é positivo, negativo ou nulo.

8. Menor de três valores

Leia três números inteiros e exiba somente o menor.

9. Aprovado ou reprovado

Leia o nome de um acadêmico e as notas obtidas em três avaliações. Calcule a média semestral (aritmética) e informe se o acadêmico foi aprovado ou reprovado. A aprovação é conquistada com média igual ou superior a 6.

10. Tipos de triângulos

Leia três valores, correspondentes aos lados de um triângulo, e apresente a classificação do referido triângulo, segundo as proporções relativas dos lados:

- Equilátero: três lados iguais;
- Isósceles: dois lados iguais;
- Escaleno: três lados diferentes.

Estruturas de Repetição

1. Colar na prova

Apresente, iterativamente, 30 vezes a mensagem “Não vou colar na prova”.

2. Tabuada do 5

Exiba iterativamente a tabuada do número 5, conforme o formato seguinte:

```
5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
(...)
5 X 10 = 50
```

3. De 100 até 1

Exiba iterativamente os números de 100 até 1.

4. De 50 até 200

Exiba iterativamente os números de 50 até 200.

5. Soma dos números

Calcule e apresente a soma de todos os inteiros no intervalo $[100, 200]$. O cálculo deve ser iterativo.

6. Potenciação

Dados a base e o expoente, ambos inteiros, calcule e exiba a potência correspondente, ou seja, a base elevada ao expoente. Assuma que o expoente fornecido nunca será negativo. Não utilize função predefinida para o cálculo da potência.

7. Média dos pares

Leia uma quantidade indeterminada de números inteiros. Calcule e exiba a média aritmética apenas dos números pares. A leitura deve ser interrompida quando o número zero for digitado. Desconsidere o zero no cálculo.

8. Validação

Solicite a entrada de um número real positivo. Exiba uma mensagem de advertência e repita a leitura enquanto o valor digitado seja negativo ou nulo. Saiba que, através do processo recém-descrito, valida-se a entrada e garante-se a consistência dos dados fornecidos.

9. Calculadora

Simule uma calculadora simples que leia dois operandos, permita a escolha da operação a ser realizada e exiba o resultado obtido. As operações possíveis (adição, subtração, multiplicação, divisão e potenciação) são indicadas pelos inteiros de 1 a 5. O funcionamento da calculadora é interrompido quando ambos os operandos informados forem iguais a zero. Exiba uma mensagem de erro, caso uma operação inválida seja escolhida. Atente-se também ao tipo de dado do resultado e aos tratamentos necessários para a realização do cálculo.

10. Fibonacci

Gere e exiba os 20 primeiros termos da série de Fibonacci. Os dois primeiros termos da série são 1 e os termos subsequentes resultam da soma dos dois anteriores imediatos: 1, 1, 2, 3, 5, 8, 13, 21, ...

11. 1/n

Dado n , representando o número de termos da série harmônica, calcule e exiba o valor de H , sendo:

$$H = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

12. Maioridade

Leia a idade de 10 pessoas e apresente quantas atingiram maioridade.

13. Menor valor

Leia 20 números inteiros e apresente o menor valor informado.

14. Maior e menor peso

Leia o peso de 25 pessoas, depois exiba o menor e o maior peso informados.

15. Pares e ímpares

Leia 20 números inteiros, então apresente a quantidade de valores pares e ímpares informados.

16. Fatorial

Calcule e exiba o fatorial de um número inteiro lido. Apresente uma mensagem de erro, caso seja fornecido um valor negativo. Não utilize função predefinida para o cálculo do fatorial.

17. Número primo

Constata e apresente se um número inteiro lido é primo. Um número natural n é primo se, e somente se, tiver exatamente dois divisores naturais: 1 e o próprio n . Por definição, os números 0 e 1 não são primos nem compostos. Realize uma verificação eficiente.

18. Tabuadas do 1 ao 10

Exiba iterativamente todas as tabuadas do 1 até o 10, conforme o formato seguinte:

```
1 X 1 = 1
1 X 2 = 2
1 X 3 = 3
```

(...)

$$10 \times 8 = 80$$

$$10 \times 9 = 90$$

$$10 \times 10 = 100$$

19. Primos após o 100

Encontre e apresente os 20 primeiros números primos após o 100.

20. Maior média

Leia o número de acadêmicos de uma turma e a quantidade de avaliações realizadas. Depois, leia o nome de um acadêmico por vez, seguido da nota obtida em cada uma das avaliações. Exiba, de imediato, a média (aritmética) atingida pelo acadêmico. Por fim, apresente a maior média obtida e o nome do acadêmico que a conquistou.

ANEXO C

ENUNCIADOS COMPLEMENTARES

A pesquisa de [25] constatou que, no catálogo adaptado de [36], não havia exercícios que incidissem sobre as perícias de: *reconhecimento de constantes na análise e resolução do problema* e *distinção conceitual e pragmática entre variáveis e constantes*. Os enunciados abaixo, portanto, complementam o catálogo original, contemplando essas perícias.

Instruções Simples

1. **Reajuste de preço**

Dado o preço de um produto, reajuste-o em 10% e exiba o resultado.

2. **Comprimento de uma circunferência**

Calcule e exiba o comprimento de uma circunferência, dado o respectivo raio. Considere a fórmula: $C = 2 \times \pi \times r$, onde π é uma constante de valor 3,14159.