

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA NA EDUCAÇÃO**

VINICIUS HARTMANN FERREIRA

**A APRENDIZAGEM DE PROGRAMAÇÃO APOIADA PELO MODELO SOCIAL  
ABERTO DO ESTUDANTE**

Porto Alegre  
2019

VINICIUS HARTMANN FERREIRA

**A APRENDIZAGEM DE PROGRAMAÇÃO APOIADA PELO MODELO SOCIAL  
ABERTO DO ESTUDANTE**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática na Educação (PGIE) da Universidade Federal do Rio Grande do Sul (UFRGS) como parte das exigências para obtenção do título de Doutor em Informática na Educação.

Orientador: Dr. Eliseo Berni Reategui

Linha de pesquisa: Ambientes Informatizados e Ensino a Distância

Porto Alegre  
2019

### CIP - Catalogação na Publicação

Ferreira, Vinicius Hartmann  
A APRENDIZAGEM DE PROGRAMAÇÃO APOIADA PELO MODELO  
SOCIAL ABERTO DO ESTUDANTE / Vinicius Hartmann  
Ferreira. -- 2019.  
112 f.  
Orientador: Eliseo Berni Reategui.

Tese (Doutorado) -- Universidade Federal do Rio  
Grande do Sul, Centro de Estudos Interdisciplinares em  
Novas Tecnologias na Educação, Programa de  
Pós-Graduação em Informática na Educação, Porto  
Alegre, BR-RS, 2019.

1. Aprendizagem de programação. 2. Meta-cognição.  
3. Modelo social aberto do estudante. I. Reategui,  
Eliseo Berni, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os  
dados fornecidos pelo(a) autor(a).



SERVIÇO PÚBLICO FEDERAL  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
CENTRO INTERDISCIPLINAR DE NOVAS TECNOLOGIAS NA EDUCAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA NA EDUCAÇÃO

**ATA SOBRE A DEFESA DE TESE DE DOUTORADO  
VINICIUS HARTMANN FERREIRA**

Às quatorze horas do dia dois de maio de dois mil e dezenove, na sala 329 do PPGIE/CINTED, nesta Universidade, reuniu-se a Comissão de Avaliação, composta pelos Professores Doutores: Marcus Vinicius de Azevedo Basso, André Luís Alice Raabe e Crediné Silva de Menezes para a análise da defesa de Tese de Doutorado intitulada “**A Aprendizagem de Programação Apoiada pelo Modelo Social Aberto do Estudante**”, do doutorando do Programa de Pós-Graduação em Informática na Educação Vinicius Hartmann Ferreira, sob a orientação do Prof. Dr. Eliseo Berni Reategui. A Banca, reunida, após a apresentação e arguição, emite o parecer abaixo assinalado.

- Considera a Tese aprovada  
() sem alterações;  
( ) sem alterações, com voto de louvor;  
( ) e recomenda que sejam efetuadas as reformulações e atendidas as sugestões contidas nos pareceres individuais dos membros da Banca;

[ ] Considera a Tese reprovada.

Considerações adicionais (a critério da Banca):

*A banca destaca a relevância do trabalho a adequação da metodologia e a excelência do resultado. Recomenda-se que as sugestões individuais dos avaliadores sejam consideradas na versão final.*

Prof. Dr. Eliseo Berni Reategui  
Orientador

Prof. Dr. Marcus Vinicius de Azevedo Basso  
PPGIE/UFRGS

(vídeo conferência)

Prof. Dr. André Luís Alice Raabe  
UNIVALI

Prof. Dr. Crediné Silva de Menezes  
PPGEDU/UFRGS

## AGRADECIMENTOS

Agradeço inicialmente ao meu orientador, professor Eliseo Reategui. Seus conselhos e sua contribuição foram essenciais para a realização desta tese.

Agradeço ao Instituto Federal do Rio Grande do Sul pelo apoio e também pelo incentivo à qualificação de seus servidores.

Agradeço a banca avaliadora, cujas contribuições tornaram esta tese mais rica.

Agradeço a minha esposa, Vanessa, que sempre está ao meu lado em todos os momentos. Também agradeço ao Nicolau, nosso querido filho, por sempre agraciar meus dias com sinceros sorrisos.

Agradeço aos amigos que fizeram parte desta caminhada. Compartilhar deste processo com eles foi imprescindível.

Por fim, agradeço aos meus pais por todos os esforços e pelo apoio.

*“Nunca conheci quem tivesse levado porrada.  
Todos os meus conhecidos têm sido campeões em tudo.  
[...] Arre, estou farto de semideuses!  
Onde é que há gente no mundo? [...]”*

Fernando Pessoa  
“Poema em linha reta”

## RESUMO

A programação de computadores é essencial para quem trabalha na área de tecnologia e projeta-se que será tão importante quanto a leitura e as matemáticas em um futuro não tão distante. No entanto, aprender a programar não é uma tarefa trivial. Dentre as principais dificuldades enfrentadas pelos estudantes estão as de compreender e avaliar suas próprias capacidades e limitações, sendo estas dificuldades relacionadas aos processos meta-cognitivos. Assim, a presente tese tem como objetivo investigar como o *scaffolding* meta-cognitivo por meio da interação com o Modelo Social Aberto do Estudante (MSAE) pode contribuir para a aprendizagem de programação. Esta tese está apoiada no modelo de meta-cognição proposto por Tobias e Everson, no qual a capacidade de avaliar o próprio conhecimento é um pré-requisito para os demais processos de regulação, e em pesquisas que demonstram que a interação dos estudantes com o MSAE contribui para a regulação da própria aprendizagem. Nesta tese foi realizado um quasi-experimento com duas turmas de Introdução a Programação que utilizaram um Ambiente Virtual de Aprendizagem integrado ao MSAE durante um semestre. A coleta de dados se deu por meio de respostas ao questionário Metacognitive Awareness Inventory (MAI), da obtenção dos índices de Precisão no Monitoramento do Conhecimento (KMA) e Viés no Monitoramento do Conhecimento (KMB), do desempenho na resolução de exercícios, do *log* de uso do Ambiente Virtual de Aprendizagem e de entrevista semi-estruturada com um grupo de estudantes. Como resultado verificou-se que os estudantes utilizaram o MSAE como um recurso útil para organização, para avaliação e monitoramento e para solicitar ajuda durante o processo de aprendizagem. Além disso, por mais que não tenha sido verificada alteração nos aspectos meta-cognitivos dos estudantes a partir do uso do MSAE, aqueles estudantes que realizaram mais exercícios corretamente obtiveram melhor desempenho geral que os demais.

**Palavras-chave:** Aprendizagem de Programação; Meta-cognição; Modelo Social Aberto do Estudante.

## **ABSTRACT**

*Computer programming is essential for anyone who works in the area of technology, and is projected to be as important as reading and Maths in the not-so-distant future. However, learning to program is not a trivial task. Among the main difficulties faced by students are to understand and evaluate their own abilities and limitations, these difficulties being related to meta-cognitive processes. Thus, the present thesis aims to investigate how metacognitive scaffolding through interaction with the Open Social Student Model (OSSM) can contribute to programming learning. This thesis is supported by the meta-cognition model proposed by Tobias and Everson, in which the ability to assess knowledge itself is a prerequisite for the other regulatory processes, and in research that demonstrates that students' interaction with the OSSM contributes for the regulation of learning. In this thesis was a quasi-experiment with two classes of Introduction to Programming that used a Virtual Learning Environment integrated to the OSSM during a semester. The data collection was done through answers to the Metacognitive Awareness Inventory (MAI) questionnaire, obtaining Knowledge Monitoring Accuracy (KMA) and Knowledge Monitoring Bias (KMB) indexes, exercise resolution, logging use of the Virtual Environment and semi-structured interview with a group of students. As a result, it was found that students used the OSSM as a useful resource for organization, evaluation and monitoring, and to request help during the learning process. In addition, although there was no change in the meta-cognitive aspects of the students from the use of the OSSM, those students who performed more exercises correctly obtained better overall performance than the others.*

**Keywords:** *Programming learning, Metaconition, Open Social Student Model.*

## SUMÁRIO

1	INTRODUÇÃO .....	16
1.1	OBJETIVOS .....	20
1.1.1	Objetivo geral.....	20
1.1.2	Objetivos específicos .....	20
1.2	JUSTIFICATIVA .....	20
2	APRENDIZAGEM DE ALGORITMOS E PROGRAMAÇÃO .....	22
2.1	DIFICULDADES NA APRENDIZAGEM DE ALGORITMOS .....	25
2.2	CONSIDERAÇÕES SOBRE O CAPÍTULO.....	34
3	META-COGNIÇÃO .....	35
3.1	A APRENDIZAGEM AUTORREGULADA .....	35
3.2	A DEFINIÇÃO DE META-COGNIÇÃO.....	37
3.3	MODELOS DE META-COGNIÇÃO.....	38
3.3.1	O modelo de Flavell (1979) .....	38
3.3.2	O modelo de Brown (1989) .....	39
3.3.3	O modelo de Tobias e Everson (2002).....	40
3.4	AVALIAÇÃO DE META-COGNIÇÃO.....	42
3.4.1	Metacognitive Awareness Inventory .....	42
3.4.2	Precisão no Monitoramento do Conhecimento (KMA).....	43
3.4.3	Viés no Monitoramento do Conhecimento (KMB) .....	45
3.5	SCAFFOLDING META-COGNITIVO .....	46
3.6	CONSIDERAÇÕES SOBRE O CAPÍTULO.....	51
4	MODELO ABERTO DO ESTUDANTE .....	53
4.1	CONSIDERAÇÕES SOBRE O CAPÍTULO.....	61
5	PROCEDIMENTOS METODOLÓGICOS .....	63
5.1	AMOSTRA INICIAL .....	63

5.2	DESIGN DO EXPERIMENTO.....	63
5.3	AMBIENTE VIRTUAL DE APRENDIZAGEM .....	65
5.3.1	Primeira versão do Ambiente Virtual de Aprendizagem.....	66
5.3.2	Versão final do Ambiente Virtual de Aprendizagem.....	67
5.4	RESULTADOS PRELIMINARES .....	68
6	RESULTADOS.....	70
6.1	AMOSTRA FINAL .....	70
6.2	A CONSCIÊNCIA META-COGNITIVA.....	71
6.3	O DESEMPENHO.....	76
6.4	O ACESSO AO AMBIENTE VIRTUAL DE APRENDIZAGEM .....	82
7	CONCLUSÃO .....	86
7.1	LIMITAÇÕES DO ESTUDO.....	90
	REFERÊNCIAS .....	91
	ANEXO A .....	104
	ANEXO B.....	108
	ANEXO C.....	109
	ANEXO D .....	110

## ÍNDICE DE ILUSTRAÇÕES

Figura 1. Ambiente de programação Scratch. ....	32
Figura 2. Tela principal do sistema AnnotaEx. ....	33
Figura 3. Modelo de autorregulação. ....	36
Figura 4. Modelo de meta-cognição de Flavell. ....	39
Figura 5. Modelo de meta-cognição de Brown. ....	40
Figura 6. Modelo de meta-cognição de Tobias e Everson.....	41
Figura 7 – Ficha de acompanhamento de sprint. ....	49
Figura 8 – Comparação entre expectativa e resultado obtido.....	50
Figura 9 – Exemplo de visualização de Modelagem Aberta do Estudante. ....	54
Figura 10. Representação gráfica do MAE para aprendizagem de SQL.....	55
Figura 11. Apresentação dos usuários da comunidade Comtella. ....	57
Figura 12 – Exemplo de visualização de Modelagem Social Aberta do Estudante. ....	59
Figura 13 – Exemplo de visualização da MSAE na interface MasteryGrids. ....	60
Figura 14 – Comparação entre contribuições de usuários na plataforma Topolor. ....	61
Figura 15 – Implementação do MSAE no ambiente de aprendizagem. ....	66
Figura 16 – Visualização do modelo do estudante e dos seus colegas.....	67
Figura 17 – Visualização dos modelos dos estudantes na versão final do AVA.....	68
Figura 18. Resultado geral para subescalas do MAI. ....	73
Figura 19. Desempenho dos grupos nas avaliações por tópico. ....	78
Figura 20. Desempenho dos grupos nos exercícios centrais por tópico. ....	79
Figura 21. Desempenho dos grupos nos exercícios complementares por tópico. ....	80
Figura 22. Quantidade de exercícios realizados por tópico. ....	80

## ÍNDICE DE TABELAS

Tabela 1. Avaliação geral do MAI.....	72
Tabela 2. Resultados para <i>conhecimento cognitivo</i> .....	72
Tabela 3. Resultados para <i>regulação da cognição</i> .....	72
Tabela 4. Resultados do pré-teste e do pós-teste.....	76
Tabela 5. Resultados obtidos nos exercícios disponíveis no AVA.....	77

## ÍNDICE DE QUADROS

Quadro 1. Categorias de problemas relacionados a aprendizagem de programação. ....	27
Quadro 2. Problemas de aprendizagem de natureza meta-cognitiva. ....	29
Quadro 3. Intervalos possíveis no índice KMA.....	44
Quadro 4. Intervalos possíveis no índice KMA revisto.....	44
Quadro 5. Classificação dos estudantes com base no KMA.....	45
Quadro 6. Intervalos possíveis no índice KMB. ....	45
Quadro 7. Classificação dos estudantes com base no KMB.....	46
Quadro 8. Estudantes entrevistados ao final do quasi-experimento. ....	71
Quadro 9. Auxílio fornecido pelo MSAE às dificuldades de aprendizagem.....	89
Quadro 10. Subescalas do questionário MAI utilizadas no experimento. ....	104
Quadro 11. Questões da subescala Conhecimento declarativo.....	104
Quadro 12. Questões da subescala Conhecimento processual. ....	105
Quadro 13. Questões da subescala Conhecimento condicional.....	105
Quadro 14. Questões da subescala Planejamento. ....	105
Quadro 15. Questões da subescala Gestão da informação.....	106
Quadro 16. Questões da subescala Monitoramento.....	106
Quadro 17. Questões da subescala Depuração. ....	107
Quadro 18. Questões da subescala Avaliação. ....	107

## ÍNDICE DE EQUAÇÕES

Equação 1. Knowledge Monitoring Accuracy original .....	43
---	----

## **LISTA DE SIGLAS**

AVA	Ambiente Virtual de Aprendizagem
KMA	Knowledge Monitoring Accuracy
KMB	Knowledge Monitoring Bias
MAE	Modelo Aberto do Estudante
MAI	Metacognitive Awareness Inventory
MOODLE	Modular Object-Oriented Dynamic Learning Environment
MSAE	Modelo Social Aberto do Estudante
PPGIE	Programa de Pós-Graduação em Informática na Educação
UFRGS	Universidade Federal do Rio Grande do Sul

## 1 INTRODUÇÃO

A tecnologia computacional está cada vez mais presente no cotidiano da sociedade. Antes utilizados apenas por pessoal especializado para atividades profissionais, como o controle do estoque de uma empresa varejista, os computadores, software e hardware, assumiram papel importante nas relações humanas, no meio acadêmico e nos meios de comunicação. Assim, com a popularização da internet e dos dispositivos conectados a ela, o acesso aos recursos proporcionados pelos computadores e smartphones abrange usuários com diferentes níveis de experiência e interesses distintos.

Em decorrência deste cenário cresce o interesse dos usuários por não mais apenas consumir aplicativos ou serviços desenvolvidos por outros, mas sim produzir os seus. Isto se verifica pelo surgimento de diferentes plataformas que ofertam cursos online de programação, como Khan Academy<sup>1</sup>, Udacity<sup>2</sup> e Coursera<sup>3</sup>, e pelo grande número de usuários que aderem a elas.

O interesse pelo desenvolvimento de software não se restringe apenas aos usuários que querem criar a sua própria rede social ou mudar a cor dos alertas em seu smartphone. O meio acadêmico também vem se transformando e hoje em grande parte do tempo os engenheiros, físicos, economistas, biólogos, entre outros, trabalham com o desenvolvimento de modelos computacionais criados a partir de algoritmos (BLIKSTEIN, 2008). De forma similar, cresceu nos últimos anos a busca das empresas por profissionais qualificados para o desenvolvimento de software, serviços online ou aplicativos. Porém, muitas são as vagas nesta área que não são preenchidas justamente pela falta destes profissionais (CETIC, 2016). Assim, verifica-se que saber desenvolver softwares também é uma necessidade do mercado de trabalho.

Programar computadores está relacionado à habilidade de resolução de problemas e com isso exige e estimula o desenvolvimento da abstração, da compreensão, do planejamento e da criatividade. Além disso, a programação tem sido indicada como uma das formas de estimular o Pensamento Computacional, que se refere à resolução de problemas enfatizando o uso de estratégias oriundas de diferentes disciplinas da Ciência da Computação (WING, 2006). Desta forma, há um movimento que reivindica o ensino do pensamento computacional, incluindo nisto a programação, desde as séries iniciais.

---

<sup>1</sup> <https://pt.khanacademy.org/>

<sup>2</sup> <https://br.udacity.com/>

<sup>3</sup> <https://pt.coursera.org/>

A importância do ensino de programação é evidenciada por iniciativas como a Hora do Código, criada pela Code.org<sup>4</sup>, uma instituição pública sem fins lucrativos que tem por objetivo popularizar a Ciência da Computação. A Hora do Código é um evento que estimula a aprendizagem de programação por 1 hora durante uma semana e que pode ser organizado por voluntários. Para isso, são disponibilizadas atividades em 45 idiomas que devem ser realizadas pelos aprendizes. Até o início do ano de 2017 haviam sido registrados cerca de 166.000 eventos distribuídos em 180 países, dentre eles o Brasil.

Nesse contexto, em 2016 o presidente dos Estados Unidos da América, Barack Obama, iniciou um programa intitulado “Ciência da Computação para todos”. A proposta da iniciativa foi de estimular o ensino do Pensamento Computacional por meio da programação nas escolas, do jardim de infância até o ensino médio. Para que isso se tornasse possível foram disponibilizados U\$4 bilhões como fundos para o desenvolvimento de infraestrutura adequada nas escolas e para a formação dos professores (SMITH, 2016).

No Brasil não existem políticas públicas específicas para o ensino de programação em nível básico. Porém, trabalhos como os realizados por Andrade *et al.* (2013) e Bombasar *et al.* (2015) apresentam possibilidades de atividades e metodologias com este foco. Além disso, iniciativas como as propostas pelo site Programaê!<sup>5</sup>, que disponibiliza cursos online e está associada ao evento Hora do Código, e pelo Code Club Brasil<sup>6</sup>, que dá suporte para a criação de clubes de programação para crianças, têm evidenciado o interesse pelo ensino do Pensamento Computacional e da programação no país.

Embora aprender programação contribua para a vida acadêmica dos estudantes, trata-se de uma aprendizagem desafiadora. As disciplinas de programação apresentam alto índice de reprovação e, em muitos casos, o insucesso enfrentado nelas faz com que os estudantes abandonem o curso (WATSON; LI, 2014).

As dificuldades de aprendizagem ocorrem por diferentes razões. A programação envolve linguagens específicas, através das quais o programador dá comandos para o computador. Por ser algo novo, distante do cotidiano, muitos estudantes encontram dificuldades em lembrar dos comandos. Ao mesmo tempo, a programação exige que o estudante elabore soluções que combinem uma linguagem de programação e raciocínio lógico. Neste ponto, os estudantes apresentam dificuldades para elaborar, estruturar e formalizar uma solução que atenda ao solicitado por um problema.

---

<sup>4</sup> <https://hourofcode.com/br>

<sup>5</sup> <http://programae.org.br/>

<sup>6</sup> <http://www.codeclubbrasil.org.br/>

Outro fator apontado como motivo das dificuldades na aprendizagem de programação relaciona-se à metodologia de ensino. Nesse caso, exercícios com enunciados confusos ou pouco explicativos podem contribuir para a dificuldade em solucionar um problema. A indisponibilidade dos docentes em atender de forma adequada todos os estudantes de uma turma, identificando dificuldades ou lacunas individuais, também contribui para a dificuldade de aprendizagem.

Fatores relacionados à afetividade também são apontados como motivo para as dificuldades de aprendizagem. Ao se defrontarem com o insucesso na disciplina, muitos estudantes sentem-se deslocados na turma e desmotivam-se em aprender. Em relação à motivação, também são apontadas as várias tentativas sem sucesso para a resolução de um problema. Desta forma, tanto a desmotivação pode levar as dificuldades quanto o contrário (RAABE, 2005; IEPSSEN, 2013).

Bergin, Reilly e Traynor (2005) afirmam que os estudantes que conseguem planejar, monitorar e avaliar o seu próprio progresso alcançam melhores resultados de aprendizagem. Estas habilidades estão relacionadas à meta-cognição, definida como a capacidade de monitorar, regular e orquestrar os próprios processos cognitivos para alcançar um objetivo (FLAVELL, 1979). Assim, ao não conhecer suas próprias capacidades e lacunas de conhecimento em relação ao conteúdo estudado, o estudante não consegue identificar exercícios e materiais adequados ou avaliar seu progresso a fim de solicitar auxílio (LOKSA; KO, 2016).

Esta tese está alinhada ao modelo de meta-cognição proposto por Tobias e Everson (2002), no qual os processos meta-cognitivos são divididos em três componentes, o conhecimento sobre a meta-cognição, o monitoramento dos processos de aprendizagem e o controle sobre estes processos. Neste modelo há uma hierarquia na qual estipula-se que o conhecimento sobre as próprias capacidades, a consciência sobre o que se sabe e o que não se sabe, é pré-requisito para ativar os demais processos meta-cognitivos. Assim, é essencial que o estudante desenvolva a habilidade de reconhecer suas capacidades.

Dar suporte ou estimular o desenvolvimento da meta-cognição tem apresentado resultados positivos em diferentes áreas de conhecimento. Pesquisas que enfocam em fornecer auxílio gradual aos estudantes, por meio de uma estratégia conhecida com *scaffolding* meta-cognitivo, têm demonstrado que podem potencializar a independência e o desenvolvimento cognitivo dos estudantes (LUCKIN E HAMMERTON, 2002).

Trabalhos sobre o *scaffolding* meta-cognitivo na aprendizagem de programação também têm sido desenvolvidos e têm demonstrado potencializar a produtividade, o desempenho, a

independência e a confiança dos estudantes. Estas pesquisas possuem diferentes enfoques, como o suporte automático, dado por um Sistema Tutor Inteligente (FRANÇA; TEDESCO, 2014), o uso de instrumentos físicos para controlar o próprio progresso (BORGES; MORAES; OKUYAMA, 2013; LOKSA; KO2016) ou recursos de visualização que auxiliam o estudante na reflexão sobre sua aprendizagem, como o Modelo Aberto do Estudante (MAE) e o Modelo Social Aberto do Estudante (MSAE).

O MAE caracteriza-se pela disponibilização para o estudante das informações registradas pelo sistema sobre seu desempenho e uso dos recursos disponíveis. Baseando-se na Teoria da Comparação Social (FESTINGER, 1954), sobretudo na premissa de que os indivíduos tendem a se comparar com os demais, o MSAE caracteriza-se pela possibilidade de visualizar o próprio MAE e também o dos colegas. Com isso, o uso do modelo aberto facilita a reflexão e a monitoração sobre a própria aprendizagem ao possibilitar que os estudantes visualizem de forma concreta seu desempenho ou o acesso aos materiais disponibilizados em um curso. O modelo aberto também contribui para o planejamento das etapas de aprendizagem, uma vez que o estudante tende a analisar seu desempenho e o dos colegas e decidir os próximos passos a partir da análise.

O uso do modelo aberto na aprendizagem de programação é relatado em trabalhos como os de Mitrovic e Martin (2007), que incorporaram o MAE em um ambiente de aprendizagem de banco de dados com a linguagem SQL e de Hsiao *et al.* (2011), que apresenta a incorporação do MSAE ao QuizzJET, uma ferramenta para produção e disponibilização de questões parametrizadas sobre programação em Java. Ainda, podem ser citados os trabalhos de Loboda *et al.* (2014), Brusilovsky *et al.* (2015) e Guerra *et al.* (2016), que apresentam o uso do MSAE por meio de uma ferramenta de código aberto intitulada *MasteryGrids*.

Em todos os trabalhos que incorporaram o MSAE verificou-se maior acesso aos materiais disponibilizados, sobretudo aqueles considerados complementares, e maior participação na resolução das atividades. Além disso, os estudantes que apresentavam dificuldades de aprendizagem apresentaram melhor rendimento ao responder os exercícios. Desta forma, evidencia-se a contribuição do MSAE não só para o desenvolvimento da metacognição mas também para um maior engajamento dos estudantes.

A possibilidade de comparar o próprio modelo com o dos colegas facilita a sensação de pertencimento a um grupo, pois o estudante encontrará colegas com desempenho similar ao dele. Ao mesmo tempo, quando o estudante sente que os demais estão se comparando com ele tende a assumir uma posição de maior responsabilidade com a própria aprendizagem. Além

disso, o estudante também pode comparar seu desempenho com o de colegas, o que lhe permite verificar sua posição junto ao grupo e readequar suas próprias estratégias de aprendizagem.

Esta teste propõe a construção de um ambiente de aprendizagem para programação que implemente o MSAE. Partindo deste ambiente, pretende-se investigar como o *scaffolding* meta-cognitivo, proporcionado pela interação com o MSAE, pode contribuir para o processo de aprendizagem de programação.

É importante destacar que iniciativas que estimulem o ensino de programação devem vir acompanhadas do suporte às dificuldades de aprendizagem dos estudantes. Caso contrário, a iniciativa apenas reforçará o insucesso acadêmico do aprendiz. Assim, esta tese contribui também para a popularização do ensino de programação.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo geral

Investigar como o *scaffolding* meta-cognitivo proporcionado pela interação com o Modelo Social Aberto do Estudante pode contribuir para o processo de aprendizagem de programação.

### 1.1.2 Objetivos específicos

- Analisar se a possibilidade de o estudante comparar o seu desempenho com o dos colegas contribui para o desenvolvimento da regulação da própria aprendizagem;
- Avaliar se há relação entre o desempenho dos estudantes e a capacidade de regulação da própria aprendizagem; e
- Identificar como os estudantes utilizaram o Modelo Social Aberto do Estudante durante o processo de aprendizagem de programação.

## 1.2 JUSTIFICATIVA

Diante de um cenário no qual a habilidade de programar começa a ser percebida como essencial para o sucesso profissional e acadêmico, cresce o interesse da comunidade acadêmica em dar suporte àqueles estudantes que apresentam dificuldades de aprendizagem em disciplinas de programação. Este interesse é percebido pelo crescimento de publicações relacionadas ao tema, verificado por meio de diferentes revisões da literatura.

A programação exige capacidades individuais essenciais para a aprendizagem com sucesso, como o acompanhamento da própria aprendizagem por meio da meta-cognição. Com isso, dar suporte para que o estudante desenvolva estas habilidades é fundamental. Assim, disponibilizar para o estudante uma representação do seu conhecimento, por meio do MAE, deve auxiliá-lo a monitorar seu progresso. Acrescenta-se a isto o compartilhamento deste modelo com os demais colegas, por meio do MSAE, que permite a ele realizar comparações e ajustar suas próprias ações.

Também neste contexto, torna-se fundamental estimular o estudante a perceber suas capacidades com maior eficácia. Com isso, incorporar mecanismos ao MSAE que o façam refletir sobre o que ele realmente sabe pode auxiliá-lo a readequar suas estratégias de aprendizagem. Assim, a contribuição desta tese reside em investigar se um mecanismo de apoio ao desenvolvimento da meta-cognição por meio do registro e disponibilização de uma representação do conhecimento do estudante e de seus pares contribui para o processo de aprendizagem de programação. Esse mecanismo permitirá ao estudante compreender seu próprio desempenho, comparar seu desempenho com o dos demais e refletir sobre a percepção que tem sobre si mesmo.

## 2 APRENDIZAGEM DE ALGORITMOS E PROGRAMAÇÃO

O desenvolvimento e o crescimento da participação da tecnologia computacional no cotidiano de famílias, governos, empresas e instituições de ensino e saúde nos últimos 30 anos é visível. De grandes computadores com pouca capacidade de processamento e armazenamento, utilizados apenas por um grupo restrito de funcionários capacitados, passou-se neste tempo para computadores pessoais de mesa, *notebooks*, *tablets* e *smartphones* com grande poder de processamento e alta capacidade de armazenamento, seja por meio de memória física ou acesso à servidores na nuvem por meio da internet.

Inicialmente utilizados apenas por setores específicos de determinadas empresas, como o Centro de Processamento de Dados (CPD), os computadores atualmente estão presentes na maioria dos setores de pequenas, médias e grandes empresas, tornando-se em grande parte dos casos a principal ferramenta de trabalho. Além disto, os computadores embarcados estão presentes no cotidiano dos lares, desde o forno micro-ondas até o controlador remoto do ar condicionado, e nos meios de transporte, como nos sistemas de piloto automático dos aviões e em sistemas de GPS em automóveis.

Muitas são as empresas com foco na área de Tecnologia da Informação e Comunicação (TIC) que se destacam no atual cenário. Em uma lista divulgada pela revista Forbes em 2016, das 10 marcas mais valiosas do mundo 5 são empresas do ramo das TIC, sendo elas as empresas Apple (1º), Google (2º), Microsoft (3º), Facebook (5º) e IBM (7º). Neste cenário, surge uma grande demanda por profissionais capazes de projetar, desenvolver, testar e implantar sistemas computacionais (BADENHAUSEN, 2016).

No cenário brasileiro a demanda por profissionais especializados na área de TIC também é grande. Conforme dados de pesquisa realizada pelo Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br), empresas de grande porte (com 250 ou mais funcionários) não conseguiram preencher as vagas relativas ao setor de TIC por motivo de falta de candidatos especializados, pelo alto custo destes profissionais e pela falta de candidatos especialistas com experiência na área (CETIC, 2016).

Diante da demanda por profissionais capacitados para a área de TIC, sobretudo focados no desenvolvimento de software, muitas são as iniciativas que estimulam a aprendizagem de programação de computadores. Dentre elas destacam-se os cursos gratuitos e pagos ofertados

por meio de plataformas online como Khan Academy<sup>7</sup>, Code Academy<sup>8</sup> e Udacity<sup>9</sup>. Através destas iniciativas é possível aprender do básico de programação até o desenvolvimento de páginas web complexas em cursos criados em parceria com empresas mundialmente reconhecidas, como Google e AT&T.

Mais do que apenas pensar na demanda do setor comercial, Dan Crow (2014) defende que as crianças deveriam aprender a programar nas escolas. Em um artigo publicado no jornal *The Guardian*, o autor defende que em um futuro próximo não saber programação será semelhante a não saber ler ou somar, habilidades hoje aprendidas nas escolas. Para além do desenvolvimento de aplicativos e sites, Wing (2006) apresentou o conceito de Pensamento Computacional, definido como a habilidade de empregar técnicas computacionais, como a elaboração de algoritmos e de programação, para resolver problemas do cotidiano. Desta forma, percebe-se que ensinar e aprender programação, seja para construir aplicações ou para estimular o uso do raciocínio lógico por meio de algoritmos, é atualmente um tema de grande relevância.

Um algoritmo pode ser definido como uma sequência de passos dados para se chegar a solução de um problema ou também como a descrição de um padrão de comportamento expresso através de um conjunto finito de ações (ENGELBRECHT *et al.*, 2012; DIJKSTRA, 1971). Partindo destas definições, podem ser citados como exemplos de algoritmos instruções para o preparo de um alimento, através dos passos descritos por uma receita, ou um guia sobre como trocar o pneu furado de um veículo.

Este trabalho terá seu foco nos algoritmos computacionais, que representam um conjunto finito de instruções que quando executadas, em um tempo finito, resolvem um problema computacional (KNUTH, 1968). Ainda, um algoritmo computacional também pode ser definido como uma descrição lógica de um programa de computador que pode ser implementado em uma linguagem de programação (RAABE, 2005).

Saber dar instruções à um computador é uma habilidade imprescindível nos cursos da área de informática, como nos bacharelados em Ciência da Computação e Engenharia da Computação, nos cursos tecnológicos como Análise e Desenvolvimento de Sistemas e Sistemas para internet e nos cursos técnicos. A esta ação de dar instruções a um computador se dá o nome de programar. Por programar entende-se a tradução de uma solução lógica para um problema, um algoritmo, em instruções compreendidas pela máquina através de linguagens de programação, adaptando a solução à sintaxe e à semântica própria da linguagem utilizada. Com

---

<sup>7</sup> <https://pt.khanacademy.org/>

<sup>8</sup> <https://www.codecademy.com/pt>

<sup>9</sup> <https://br.udacity.com/>

isto, pode-se concluir que um mesmo algoritmo pode ser implementado em uma ou mais linguagens de programação.

A construção de um algoritmo computacional, ou a programação de um software, não se resume apenas à elaboração de instruções. Para Farrel (2010), podem ser identificados 6 passos para a elaboração de um programa de computador, sendo eles:

1. Compreensão do problema;
2. Planejamento da lógica;
3. Codificação do programa;
4. Compilação do código escrito para linguagem de máquina;
5. Execução e testes com o programa codificado; e
6. Disponibilizar o programa para uso.

Na execução destes passos o programador pode identificar problemas de sintaxe (erros referentes as regras de escrita da linguagem de programação escolhida) ou de semântica (erros referentes a lógica de resolução do problema). Destes dois tipos de erros, apenas o primeiro é identificado pela compilação do programa, ficando então sob responsabilidade do programador verificar se o programa produz os resultados desejados (IEPSEN, 2013).

Dada a importância da programação para os cursos da área de informática, e mesmo para áreas afim, a disciplina com foco em introduzir os conceitos de algoritmos e programação está presente, em grande maioria dos cursos, no primeiro semestre (MEC, 2016). Estas disciplinas assumem diferentes nomes (Algoritmos e Programação, Lógica de algoritmos, Programação I, entre outros), porém compartilham ementas similares, nas quais constam assuntos como: (i) representação de dados; (ii) operações; (iii) entrada e saída de dados; (iv) desvios condicionais; (v) laços de repetição; (vi) dados compostos; e (vii) modularização (RAABE, 2005).

No trabalho de Júnior e Rapkiewicz (2004) são apresentadas habilidades que precisam ser desenvolvidas para alcançar a competência de elaborar algoritmos e conseqüentemente programar softwares, que são: (i) interpretação e compreensão de texto; (ii) resolução de problemas; (iii) formalização de solução proposta; e (iv) construção de algoritmos. Assim, conclui-se que aprender a programar não é uma tarefa fácil, exige a compreensão de conceitos abstratos e a elaboração de soluções que exigem raciocínio lógico (DIJKSTRA, 1971; ROBINS; ROUNTREE; ROUNTREE, 2003).

## 2.1 DIFICULDADES NA APRENDIZAGEM DE ALGORITMOS

A disciplina introdutória de algoritmos e programação é conhecida por ter altas taxas de reprovação e ser considerada uma das disciplinas mais difíceis dos cursos da área de informática, porém, são poucos os trabalhos que apresentam dados quantitativos que demonstrem isso. Em um dos trabalhos que aborda este tema Bennedsen e Caspersen (2007) contataram instituições de nível superior que ofertavam cursos na área de Informática nos Estados Unidos da América (34%), na Austrália (4%) e em países do leste (12%) e oeste (50%) europeu. Através de um questionário enviado para estas instituições foi possível concluir que a média de alunos que são aprovados na disciplina é de 67%, embora existam grandes variações entre alunos que são aprovados, reprovados ou que abandonaram o curso. Também verificou-se que turmas formadas por menos de 30 alunos, consideradas turmas pequenas, possuem maior taxa de aprovação. Ainda, foi possível concluir que o paradigma de programação escolhido pelo professor (procedural ou orientado a objetos) não influencia nas taxas de aprovação.

Em outro estudo, realizado por Watson e Li (2014), uma nova pesquisa foi realizada com o intuito de expandir e investigar os resultados obtidos por Bennedsen e Caspersen (2007). Nesta nova pesquisa realizou-se uma revisão sistemática da literatura sobre artigos que tratassem sobre a reprovação em disciplinas de algoritmos e programação. Concluiu-se que a taxa média de aprovação dos alunos é de cerca de 67%, conforme encontrado no trabalho de Bennedsen e Caspersen (2007). Além disso, concluiu-se que o número de aprovações tem variação entre diferentes países, que nos últimos 60 anos este número não aumentou de forma relevante e a linguagem de programação utilizada nos cursos não possui grande influência nos resultados.

Dado o alto número de reprovações nas disciplinas introdutórias de algoritmos, alguns trabalhos investigam as causas da dificuldade em se aprender a elaborar algoritmos e a programar. As causas identificadas são atribuídas tanto aos professores quanto aos alunos. Com relação aos professores, identifica-se que em muitos casos a metodologia adotada não favorece a aprendizagem dos alunos. Considerando que uma turma é heterogênea em todos os aspectos, conduzir a disciplina sem variar as estratégias com o intuito de favorecer alunos com diferentes características pode tornar o aprendizado mais difícil do que já o é e conseqüentemente desmotivador (GOMES; HENRIQUES; MENDES, 2008).

A falta de possibilidade de acompanhar o progresso individual dos alunos também acaba por prejudicá-los. Embora no Brasil o número recomendado de alunos em sala de aula seja de no máximo 50 alunos (RAMOS *et al.*, 2015), os professores trabalham com mais de uma turma

e devido à sobrecarga de horas em sala de aula poucos são os professores que conseguem acompanhar o aprendizado de seus alunos de forma eficaz, com feedback qualificado e em tempo de sanar todas as dúvidas (RAABE, 2005).

Em se tratando do aluno, a desmotivação e a frustração podem contribuir para as suas dificuldades de aprendizagem. Ainda, em decorrência destes fatores, o estudante enfrenta dificuldades em sentir-se parte do grupo, da turma, e isto também o leva ao insucesso (IEPSEN, 2008; VIHAVAINEM; AIRAKSINEN; WATSON, 2014).

Considerando a natureza abstrata da programação, deficiências no raciocínio lógico trazidas desde o ensino fundamental podem ter impacto negativo no aprendizado, conduzindo então o aluno a falta de interesse pela disciplina. A deficiência em regular a própria aprendizagem por meio de processos meta-cognitivos faz com que os estudantes não consigam selecionar exercícios ou materiais adequados e nem readequar suas estratégias de aprendizagem (FLAVELL, 1979; BULL; KAY, 2013; LOKSA; KO, 2016).

Uma investigação foi realizada duas vezes na Universidade de Otago, na Nova Zelândia, com turmas introdutórias de programação que contam com um total de 250 alunos. Esta pesquisa permitiu identificar quais eram as principais dificuldades dos estudantes durante a disciplina, considerando aspectos técnicos da programação e aspectos externos ao ato de programar. Em ambas as execuções do experimento concluiu-se que os alunos possuem dificuldade em compreender os enunciados dos problemas apresentados e que possuem dificuldade em trabalhar com o conceito de estruturas de dados compostas, conhecidas como vetores ou *arrays*. É interessante destacar que o conteúdo relacionado a vetores é trabalhado também em disciplinas de matemática nos cursos, o que permite traçar um paralelo entre o desempenho nas duas disciplinas (GARNER; HADEN; ROBINS, 2005; ROBINS; HADEN; GARNER, 2006).

Uma síntese que categoriza as principais dificuldades enfrentadas na aprendizagem de algoritmos e programação foi elaborada por Raabe (2005) com base na literatura e em percepções sobre turmas desta disciplina na Universidade do Vale do Itajaí (UNIVALI). Nesta síntese, adaptada por Iepsen (2013) e apresentada no Quadro 1, as dificuldades são categorizadas em problemas de natureza didática, cognitiva e afetiva.

Quadro 1. Categorias de problemas relacionados a aprendizagem de programação.

<b>Problemas de natureza didática</b>	
Grande número de alunos por turma	Turmas com 40 ou 50 alunos, dificultam avaliações e acompanhamento individualizado.
Dificuldade do professor compreender a lógica do aluno	Uso de um único raciocínio lógico para a solução do problema; impede o professor de compreender a lógica individual de cada aluno – que os leva a construir soluções equivocadas de Algoritmos
Diferença de experiência e ritmo de aprendizagem entre os alunos	Turmas mesclando alunos com alguma experiência em programação e outros sem nenhum conhecimento prévio
Ambiente de realização das provas	As provas são normalmente o momento em que o aluno percebe a diferença entre observar e fazer. Além de outros fatores que não favorecem a concentração e o raciocínio, tais como, tempo limitado, pressão e stress.
Pouco uso dos monitores da disciplina	Os alunos com dificuldades de aprendizagem geralmente não aproveitam a ajuda de monitores da disciplina
Ausência de bons materiais	Os professores comumente utilizam apenas livros e estes, no geral, apresentam o conteúdo de forma que o aluno tem dificuldade de compreender.
Alunos desorientados na escolha do curso	Muitos alunos não tem uma visão correta sobre o perfil do curso, descobrindo isso apenas no decorrer das aulas. O que, por vezes, gera um ambiente de incompreensão e descaso frente aos desafios impostos pela disciplina.
<b>Problemas de natureza cognitiva</b>	
Alunos sem perfil para solução de problemas	Carência relacionada ao desenvolvimento de estratégias de solução de problemas durante o ensino médio acarreta maiores dificuldades no ensino superior.
Alunos sem base operatório-formal	Indicativos de que o raciocínio operatório formal, que é a base para a compreensão do raciocínio lógico, não foi desenvolvido no ensino médio.
Conteúdo sem proximidade com o conteúdo escolar	Como a lógica algorítmica é algo totalmente novo para a maioria dos alunos, eles não conseguem estabelecer relações com conteúdos anteriores – principalmente matemática.
<b>Problemas de natureza afetiva</b>	
Ocasionais	Problemas esporádicos de natureza pessoal que afetam o aluno, dificultando sua concentração e influenciando seu desempenho nas avaliações.
Constantes	Baixa autoestima, pouca motivação, aversão ao conteúdo ou ao professor, insegurança são exemplos de emoções que podem afetar negativamente o aprendizado do aluno.

Fonte: Iepsen (2013).

Na categoria didática encontram-se problemas relacionados à atuação do professor, de sua relação com os alunos e também da forma com que a disciplina é organizada. Na categoria

cognitiva estão problemas relacionados ao mau desenvolvimento de funções cognitivas para resolução de problemas algorítmicos, normalmente oriundos do ensino fundamental e médio. E na categoria afetiva encontram-se problemas relacionados ao estado afetivo do aluno que podem ter influência em sua aprendizagem. Nesta categorização os problemas afetivos foram classificados de acordo com a frequência com que ocorrem (RAABE, 2005; IEPSSEN, 2013).

Podem ser encontradas dentro desta disciplina diferentes práticas pedagógicas, conforme apresenta o trabalho de Vihavainen, Airaksinen e Watson (2014), porém a mais comum é a de resolução de problemas. Nesta prática são disponibilizadas listas de exercícios (problemas) que devem ser resolvidos pelo aluno desenvolvendo algoritmos e implementando-os através de uma linguagem de programação.

Diante do enunciado de um problema o aluno pode adotar diferentes estratégias para solucioná-lo, porém existem três passos que invariavelmente serão seguidos:

1. Abstração dos dados;
2. Elaboração da solução; e
3. Validação da solução desenvolvida.

Estes passos apresentados no trabalho de Zanini (2013) estão intrinsecamente ligados aos seis passos para desenvolvimento de um programa descritos por Iepsen (2013). É importante destacar que os três passos possuem igual importância no produto final da resolução de um problema e que são dependentes uns dos outros, ou seja, caso um dos passos não seja realizado adequadamente há chances de o programa construído não funcionar corretamente.

O primeiro destes três passos envolve habilidades que vão muito além do conhecimento da sintaxe de uma linguagem de programação. A abstração dos dados envolve a boa leitura, a compreensão do que se leu e a inferência das informações pertinentes. É neste passo que o aluno define o que deve fazer, com o que deve fazer e qual é o resultado esperado. Entra neste assunto também o contexto do problema proposto. A familiaridade que o aluno tem sobre o problema que deve resolver tem impacto direto no resultado do seu programa (ZANINI, 2013).

No segundo passo o aluno deve elaborar uma sequência lógica de instruções com o intuito de resolver o problema. Esta sequência lógica não está somente ligada a instrução dadas através da linguagem de programação escolhida, mas sim aos passos lógicos necessários para resolução do problema, o mapeamento do problema para e a formalização da solução. Estas habilidades são fundamentais para a construção de algoritmos e caso o aluno não tenha desenvolvido elas de forma sólida no ensino fundamental e médio pode apresentar maiores dificuldades na aprendizagem de programação (RAABE, 2005; JESUS; BRITO, 2009; NOSCHANG *et al.*, 2014).

Conforme Havenga *et al.* (2013), o emprego da meta-cognição na resolução de problemas de programação é fundamental. Quanto maior a complexidade do problema a ser resolvido maior será a necessidade de aplicação dos processos meta-cognitivos. Desta forma, fica evidente que é necessário que o estudante saiba planejar uma solução, monitorar o desenvolvimento e desempenho do programa que está sendo escrito e avaliar sua produção, refletindo sobre o resultado e os erros encontrados. Além disso, é fundamental que o estudante conheça suas dificuldades e a partir disso identifique conteúdos que devam ser reforçados e exercícios que se adequem ao seu nível de conhecimento. Assim, os estudantes que conseguem regular sua aprendizagem, empregando de forma efetiva processos meta-cognitivos, obtém melhores resultados na aprendizagem (BERGIN; REILLY; TRAYNOR, 2005).

Com isso, acrescentam-se ao Quadro 1 problemas de natureza meta-cognitiva, ou seja, ligados à autorregulação da aprendizagem (Quadro 2). Desta forma são elencados problemas ligados à seleção de exercícios e materiais adequados; ao reconhecimento das próprias capacidades; à adoção das estratégias de estudo mais eficientes; à compreensão dos erros identificados na solução dos problemas; e à avaliação da solução proposta para um problema (FLAVELL, 1979; BULL; KAY, 2013; LOKSA; KO, 2016).

Quadro 2. Problemas de aprendizagem de natureza meta-cognitiva.

<b>Problemas de natureza meta-cognitiva</b>	
Alunos com dificuldades para selecionar recursos adequados ao seu nível de conhecimento	Diante de uma variedade de recursos e atividades disponibilizados nas disciplinas de programação os alunos enfrentam dificuldades em identificar aqueles que são mais adequados ao seu nível de conhecimento ou que irão contribuir para a sua aprendizagem.
Alunos que avaliam de forma imprecisa o seu conhecimento sobre os conteúdos.	O aluno não consegue avaliar quais conteúdos já domina e em quais precisa focar os estudos.
Alunos que não conseguem selecionar a estratégia de estudo mais eficiente.	Ao apresentar dificuldades para compreender determinado conteúdo o estudante não consegue avaliar e readequar a sua estratégia de estudo.
Alunos com dificuldade para refletir sobre os erros identificados na solução de um problema.	Ao compilar a solução para um problema o estudante recebe alertas de erros mas não consegue fazer os ajustes necessários para corrigi-los.
Alunos que não conseguem avaliar a solução elaborada para um problema.	Quando o aluno não tem certeza se a solução que ele propôs resolve o problema em questão.

Fonte: Elaborada pelo autor (2017).

Considerando a complexidade da aprendizagem de algoritmos e programação e as dificuldades apresentadas pelos alunos, pesquisas que buscam encontrar formas de auxiliar o processo de ensino e de aprendizagem nesta disciplina vem sendo realizadas. Em uma revisão sistemática da literatura realizada nos anais do Simpósio Brasileiro de Informática na Educação (SBIE) e no Workshop sobre Ensino em Computação (WEI), no período entre 1999 e 2004, Júnior e Rapkiewicz (2004) classificaram os artigos encontrados em três categorias: (i) ferramentas; (ii) estratégias; (iii) estratégias com ferramenta. A distribuição das produções indicou que 50% dos artigos encontrados tratava sobre ferramentas, 25% sobre estratégias e 25% sobre estratégias com ferramentas. Ainda, Aureliano e Tedesco (2012) realizaram uma revisão sistemática similar à realizada por Júnior e Rapkiewicz (2004) abrangendo o período entre 2002 e 2011. A partir desta nova revisão verificou-se que cerca de 63% das publicações realizadas trata sobre ferramentas computacionais que dão suporte ao ensino e a aprendizagem de programação, resultado similar ao encontrado na revisão anterior.

Uma revisão sistemática com o intuito de investigar as principais abordagens utilizadas para o ensino e a aprendizagem de programação entre 2009 e 2013 nos principais eventos e revistas da área no Brasil foi realizada por Silva *et al.* (2015). A partir deste trabalho identificou-se que a maioria dos trabalhos apresenta o desenvolvimento de ferramentas que possam dar suporte ao ensino e a aprendizagem e que o foco das pesquisas é o ensino superior na modalidade presencial. Ainda, quanto às principais abordagens utilizadas identificou-se o uso de robótica e desenvolvimento de jogos digitais, e quanto às abordagens pedagógicas foram identificadas a Aprendizagem Significativa, a Taxonomia de Bloom, original e revisada, o Construtivismo, o Ciclo de Kolb, a Construcionista e a Instrucionista.

Ramos *et al.* (2015) realizaram também uma revisão sistemática, essa com o intuito de comparar o ensino de algoritmos e programação no Brasil e no exterior por meio das publicações em eventos nacionais da área entre o período de 2001 e 2014. Nesta revisão buscou-se identificar artigos que apresentassem taxas de aprovação antes e depois da aplicação do experimento relatado. Constatou-se que poucos experimentos possuem resultados concretos sobre sua efetividade e que dos artigos que apresentaram esses dados a maioria utilizava o desenvolvimento de jogos como metodologia para motivar os estudantes.

Souza, Batista e Barbosa (2016) também realizaram uma revisão sistemática sobre problemas e dificuldades enfrentados na aprendizagem de programação. Neste trabalho foram consideradas publicações da área de Ciência da Computação indexadas na base SCOPUS entre 2010 e 2014. A partir dos dados coletados verificou-se que as principais dificuldades enfrentadas no processo de aprendizagem estão relacionadas à compreensão e aplicação dos

conceitos de programação e à falta de motivação dos estudantes para realizar as atividades propostas. Além disso, identificou-se que as principais abordagens utilizadas para amenizar estas dificuldades são a visualização de programas e algoritmos, a utilização de *serious games* e o desenvolvimento de ambientes pedagógicos específicos para o ensino e a aprendizagem de programação.

O problema de pesquisa abordado por esta proposta trata sobre estudantes que apresentam dificuldades devido à deficiência ao regular sua própria aprendizagem. Neste sentido, trabalhos que buscam dar suporte ao desenvolvimento da meta-cognição têm obtido bons resultados, sobretudo quanto à aprendizagem de programação (BULL e KAY, 2013). Diferentes abordagens para apoiar os estudantes têm sido utilizadas. Por exemplo, Resnick (2007) utiliza a plataforma Scratch de modo a dar ao estudante a possibilidade de refletir sobre a sua solução para um problema a partir das avaliações e comentários dos demais colegas e da visualização imediata dos programas construídos., Já Hsiao e Brusilovsky (2011) propõem ambiente colaborativo que permite que os estudantes comentem exemplos de programas e avaliem os comentários dos colegas, na plataforma AnnotaEx.

O Scratch é um ambiente de programação baseado em blocos visuais que tem por objetivo permitir que os programadores visualizem a sua produção de forma imediata (Figura 1). Além disto, ao considerar um ambiente que possibilita concretizar os comandos abstratos da programação, iniciantes em programação sentem-se mais confortáveis ao utilizar os blocos para programar seus jogos, músicas ou animações. A partir da plataforma que hospeda a ferramenta o estudante pode compartilhar suas produções, avaliar as produções dos colegas e criar novos projetos a partir de projetos compartilhados, processo conhecido como mixagem (RESCNICK, 2007; RESNICK *et al.*, 2009).

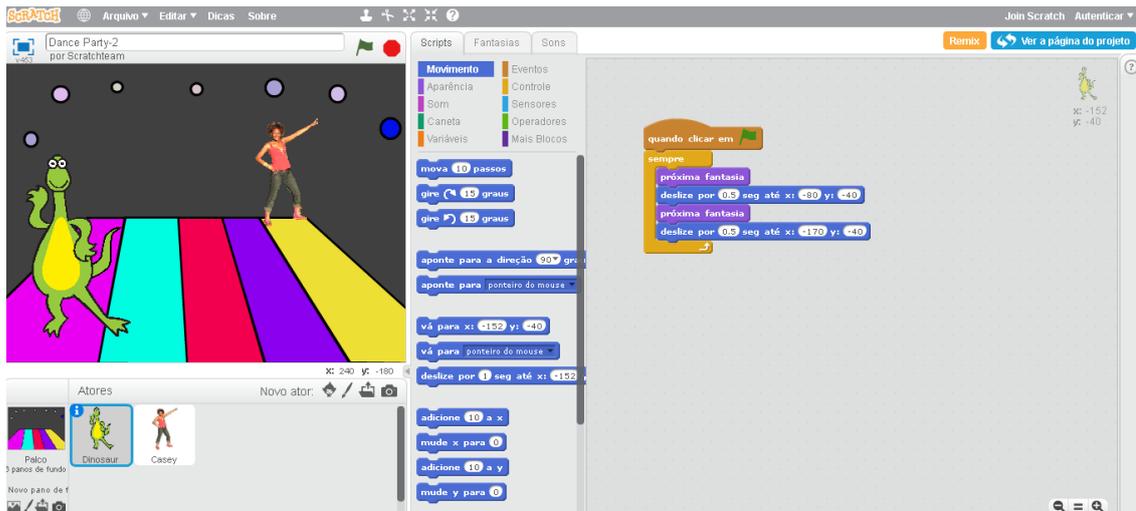


Figura 1. Ambiente de programação Scratch.

Fonte: Resnick *et al.* (2009).

A ferramenta Scratch vem sendo indicada como um recurso apropriado para a aprendizagem de programação, isso se deve também as suas características que estimulam a meta-cognição. Isto se verifica através da possibilidade do programador visualizar de forma instantânea o resultado do seu código, assim ele pode monitorar se o resultado obtido é o que ele esperava. Outro ponto ligado a meta-cognição tem relação com a comunidade através da qual os estudantes podem compartilhar suas produções e receber avaliações dos demais membros. Por meio da avaliação dos colegas, os estudantes são convidados a refletirem sobre sua produção, readequando seu código quando necessário.

Farzan e Brusilovsky (2008) apresentam o sistema AnnotaEd que implementa mecanismos de anotação e de navegação social para páginas Web. Por meio deste sistema um usuário pode inserir anotações em páginas e compartilhá-las com outros usuários; e acessar os links ou materiais que estão sendo mais acessados pelos outros usuários. Baseando-se na premissa do AnnotaEd, Hsiao e Brusilovsky (2011) desenvolveram o AnnotaEx, focado no ensino e na aprendizagem de programação.

O sistema AnnotaEx permite que os usuários comentem, linha por linha, soluções para problemas de programação. Cada usuário, além de inserir seus comentários, pode visualizar, avaliar e comentar as anotações dos demais usuários. A Figura 2 apresenta a tela principal do sistema AnnotaEx. Nela é possível visualizar os problemas nos quais o usuário já inseriu anotações, identificados por um círculo verde, e a avaliação que suas anotações receberam dos demais usuários.

The screenshot shows the main interface of the AnnotEx system. At the top, there is a navigation bar with links for Home, Authoring, System Management, HELP, My Account, and Logout. Below this, a welcome message is displayed for Sharon Hsiao, indicating her current group is VB .NET. The main content area is divided into three sections: 'Your Assignment:', 'Annotated/Commented', and a table of examples.

**Your Assignment:**

- Give Annotation:** simple, loop1
- Give Ratings:** simple, loop1, char4, array4.c, loop2
- Give Reannotation:**

**Annotated/Commented**

Example Name	Rating
<input checked="" type="radio"/> Ch1 Changing Button	4.0
<input checked="" type="radio"/> Ch1 Las Vegas	0.0
<input checked="" type="radio"/> Ch2 Numbers And Dates Average	4.0
<input checked="" type="radio"/> Ch2 Numbers And Dates Dates Difference	4.0
<input type="radio"/> Ch2 Numbers And Dates Radio Buttons	0.0
<input type="radio"/> Ch2 Timer And Text Left Right Middle	0.0
<input checked="" type="radio"/> Ch2 Timer And Text Timer	4.0
<input type="radio"/> Ch2 Timer And Text Trim Upper Replace	0.0
<input checked="" type="radio"/> Ch3 Case2 Do Loops	4.0
<input type="radio"/> Ch3 Case2 Factorial	0.0
<input checked="" type="radio"/> Ch3 Case2 For Loop Demo	4.5
<input checked="" type="radio"/> Ch3 Case2 Search Array	0.0
<input type="radio"/> Ch3 Case3 Add To Listbox	0.0
<input type="radio"/> Ch3 Case3 Remove From Listbox	0.0
<input type="radio"/> Ch3 Case4 Function Factorial	0.0
<input type="radio"/> Ch3 Case6 Random Numbers	0.0

Figura 2. Tela principal do sistema AnnotaEx.

Fonte: Hsiao e Brusilovsky (2011).

Para avaliar a contribuição que o feedback da comunidade de usuários teria nas anotações realizadas pelos usuários foi realizado um experimento no qual um grupo recebia avaliações e outro grupo não. Com este experimento verificou-se que o grupo que recebeu feedback da comunidade produziu mais e melhores anotações nos códigos, sobretudo aqueles estudantes que apresentavam maiores dificuldades de aprendizagem. Neste ponto, evidencia-se que a avaliação dos demais colegas levou os estudantes a refletirem sobre suas anotações e a elaborarem-nas de melhor forma.

Também é relevante citar a linha de pesquisa abordada por Blikstein *et al.* (2014), que utiliza técnicas de Aprendizagem de Máquina para compreender o processo adotado pelos aprendizes de programação. A partir do registro das ações dos estudantes nos ambientes de programação, é possível identificar padrões e fases comuns aos estudantes iniciantes e aquelas que são comuns aos estudantes experientes. Ao analisar os registros capturados a partir de 370 estudantes verificou-se que existem caminhos de aprendizagem que levam ao sucesso e outros que levam ao insucesso, o que significa a existência de um indício de aprovação ou reprovação em disciplinas de programação. Além disso, verificou-se que à medida que os estudantes desenvolvem suas habilidades como programadores são adquiridos hábitos comuns, como a menor alteração do código durante o processo de programação.

## 2.2 CONSIDERAÇÕES SOBRE O CAPÍTULO

Dentre as principais causas identificadas para a dificuldade em aprender a programar, esta proposta irá abordar a deficiência em regular o próprio processo de aprendizagem por meio de processos meta-cognitivos. Quando o estudante não consegue estimar com precisão suas capacidades, como os conteúdos que ele já sabe e os conteúdos que ele ainda não sabe, também não consegue selecionar exercícios ou materiais adequados para sua aprendizagem, identificar momentos em que necessita de ajuda, avaliar as suas soluções para um problema e refletir sobre os erros identificados nos testes.

Iniciativas que têm por objetivo auxiliar os estudantes a regular a própria aprendizagem a partir do monitoramento do seu desempenho e dos colegas, apoiadas por Flavell (1979), Tobias e Everson (2002) e Bull e Kay (2013), têm apresentado resultados relevantes. Estes tópicos, que embasam esta proposta, serão aprofundados nos Capítulos 3 e 4.

### 3 META-COGNIÇÃO

Este capítulo apresenta os conceitos fundamentais relacionados à meta-cognição. Para isso, introduz o conceito de aprendizagem autorregulada, da qual a meta-cognição é componente, e define o conceito de meta-cognição utilizado por esta tese. Também são apresentados o conceito de *scaffolding* meta-cognitivo e trabalhos que o relacionam à aprendizagem de programação.

#### 3.1 A APRENDIZAGEM AUTORREGULADA

A habilidade de regular o próprio processo de aprendizagem é necessária para o sucesso acadêmico e profissional. A pesquisa sobre aprendizagem autorregulada postula que os estudantes são meta-cognitivo, motivacional e comportamentalmente ativos na sua própria aprendizagem (ZIMMERMAN, 1986; PINTRICH, 2000).

Em se tratando de aspectos meta-cognitivos, o estudante autorregulado planeja, organiza, instrui-se e avalia-se em diferentes estágios da aquisição de conhecimento. Quanto ao aspecto motivacional, o estudante autorregulado percebe-se como eficaz, autônomo e intrinsecamente motivado. Ainda, em relação ao aspecto comportamental, o estudante autorregulado seleciona, estrutura e cria ambientes físicos e sociais que potencializam a sua aprendizagem. Desta forma, um estudante eficiente toma consciência da relação funcional entre seus padrões de pensamentos e ações e os resultados sociais e ambientais (ZIMMERMAN, 2002).

Um modelo de autorregulação composto por diferentes processos que auxiliam o estudante na aprendizagem foi definido por Zimmerman (2002). Este modelo (Figura 3), é composto por três fases: (i) fase prévia: processos e crenças que ocorrem antes dos esforços para aprender; (ii) fase de realização: realização da tarefa para alcançar os objetivos previamente definidos; e (iii) auto-reflexão: avaliação e reflexão sobre os resultados obtidos.

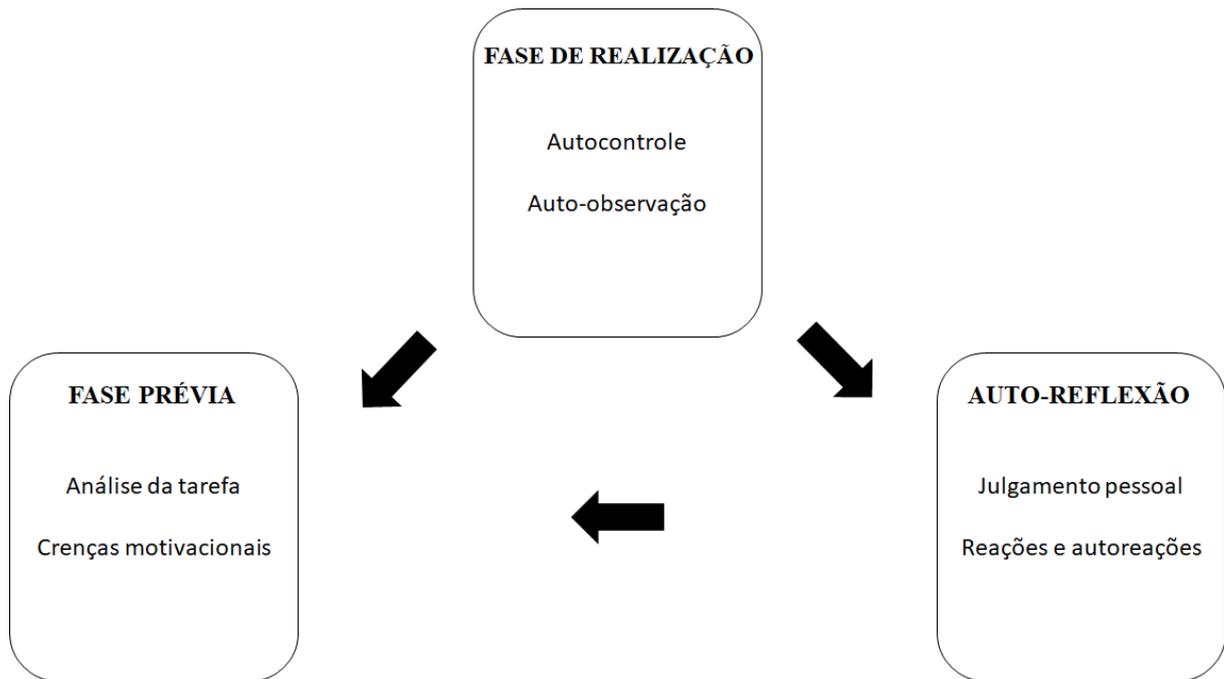


Figura 3. Modelo de autorregulação.

Fonte: Adaptado de Zimmerman (2002).

A fase prévia é composta de dois processos, a análise da tarefa e a automotivação. A análise da tarefa está relacionada a definição de metas e o planejamento para alcançar tais metas. A automotivação refere-se às crenças do próprio estudante sobre a aprendizagem. Este processo também está relacionado a motivação intrínseca, o interesse, em alcançar determinado resultado. Neste aspecto, estudantes que motivam-se pelo interesse em desenvolver suas habilidades ao invés de comparar seus resultados com os dos colegas ou apenas obter uma boa nota apresentam melhores resultados (Zimmerman, 2002).

Na fase de realização situam-se os processos de autocontrole e de auto-observação. O autocontrole está relacionado ao uso das estratégias definidas na fase anterior, como buscar um ambiente adequado para estudar. A auto-observação refere-se a perceber ou experimentar situações ou estratégias que levam a determinados acontecimentos, como a percepção de que estudando sozinho o estudante alcançou melhores resultados do que estudante com determinado colega.

A última fase, intitulada auto-reflexão, é composta pelos processos de autojulgamento e autoreação. O autojulgamento envolve atividades de análise, como a avaliação de um resultado perante um padrão esperado ou a sua comparação com o resultado obtido por outra pessoa. A autoreação pode se manifestar por meio da autossatisfação ao atingir uma meta, o que influenciará positivamente sua motivação para aprender. Ao mesmo tempo, a autoreação pode se manifestar na forma de reações defensivas, como não realizar uma tarefa pelo medo de

não a acertar. Ainda, a autoreação pode se manifestar diante de reações adaptativas, como na mudança de estratégia de estudos.

Como o modelo proposto por Zimmerman (2002) é cíclico, a auto-avaliação irá influenciar as decisões tomadas no estabelecimento de metas e seleção de estratégias adequadas para alcançá-las. Porém, se o estudante não conseguir avaliar o que efetivamente sabe enfrentará dificuldades em controlar sua aprendizagem ou avaliar de forma realista seus resultados. Desta forma, torna-se essencial que os estudantes sejam estimulados a tomarem consciência de sua própria aprendizagem.

### 3.2 A DEFINIÇÃO DE META-COGNIÇÃO

A importância dos processos meta-cognitivos é evidenciada por estudos que demonstram que os estudantes que desenvolvem melhor estas habilidades alcançam resultados acadêmicos melhores (HÄNDEL; ARTELT; WEINERT, 2013). Estes estudantes continuam e constantemente avaliam, planejam e regulam o seu progresso, o que os torna conscientes de sua própria aprendizagem (LUCKIN E HAMMERTON, 2002).

De acordo com Flavell (1979), a meta-cognição pode ser definida como o conhecimento sobre o próprio conhecimento ou a capacidade de monitorar, regular e orquestrar os próprios processos cognitivos para alcançar um objetivo. Assim, quando um estudante percebe que aprende o conteúdo A com mais facilidade que o conteúdo B ou sente que deveria fazer anotações sobre um conteúdo para não esquecê-lo são evidenciados processos meta-cognitivos.

Acrescenta-se ainda que a meta-cognição está relacionada a compreensão do conhecimento. Por sua vez, essa compreensão pode ser aplicada no uso efetivo ou na descrição do conhecimento em questão. Assim, pode-se afirmar que um estudante compreende determinada atividade cognitiva se consegue aplicá-la ou discutir sobre ela (BROWN, 1987).

É importante diferenciar de forma clara o que é cognição e o que é meta-cognição. A habilidade de ler um texto, cognitiva, é diferente daquela necessária para monitorar a compreensão do que se está lendo, sendo esta uma habilidade meta-cognitiva. Outro exemplo é a diferença entre conhecer as estruturas necessárias para programar um software, uma habilidade cognitiva, e a percepção de que se é melhor fazendo o projeto do software do que desenvolvendo ele, outro exemplo de habilidade meta-cognitiva (GAMA, 2004).

Flavell (1979) afirma que a meta-cognição e a cognição são similares em se tratando de forma e qualidade, ambas podem ser adquiridas ou esquecidas. Para o autor, elas diferenciam-se quanto ao conteúdo e a função. O conteúdo da meta-cognição são as habilidades, o

conhecimento e informações sobre a cognição, parte do mundo mental, enquanto para a cognição o conteúdo trata sobre ambos os mundos, mental e real. Por outro lado, a função da cognição é a resolução de problemas, enquanto a função da meta-cognição é regular os processos cognitivos empregados na resolução de um problema.

### 3.3 MODELOS DE META-COGNIÇÃO

Diferentes modelos de concepção da meta-cognição são encontrados na literatura. Na próxima seção serão apresentados os modelos propostos por Flavell (1979), um dos precursores nos estudos sobre o tema e o primeiro a apresentar componentes da meta-cognição e sua relação; por Brown (1989), que apresenta distinção entre a monitoração do conhecimento e a regulação da cognição; e por Tobias e Everson (2002), que apresentam um modelo modular e hierárquico.

#### 3.3.1 O modelo de Flavell (1979)

Para Flavell (1979), a meta-cognição ocorre a partir da ação e da interação entre quatro fenômenos, o conhecimento meta-cognitivo, as experiências meta-cognitivas, objetivos (atividades) e ações (estratégias) (Figura 4). O conhecimento meta-cognitivo é definido como a consciência sobre suas próprias capacidades e limitações cognitivas. Este conhecimento por sua vez é dividido em três subcategorias: (i) variáveis da pessoa; (ii) variáveis da tarefa; e (iii) variáveis da estratégia.

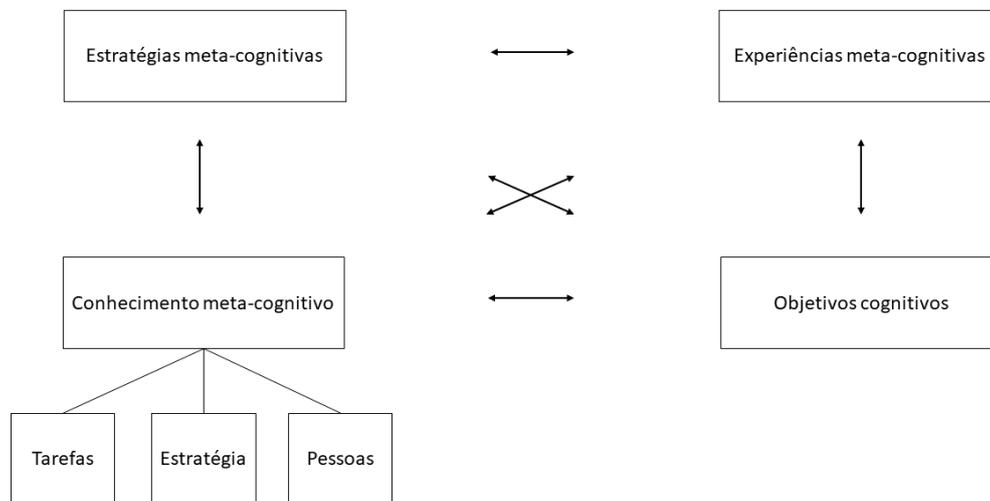


Figura 4. Modelo de meta-cognição de Flavell.

Fonte: Traduzido de Flavell (1979).

As variáveis da pessoa estão relacionadas ao conhecimento do próprio funcionamento cognitivo e afetivo. As variáveis da tarefa dizem respeito ao conhecimento adquirido sobre como lidar com diferentes informações. E as variáveis da estratégia são classificadas em cognitivas, o resultado de uma tarefa, e meta-cognitivas, a eficiência do resultado.

Os objetivos, ou atividades, estão relacionadas às metas estabelecidas para cada envolvimento cognitivo. As ações, ou estratégias, dizem respeito às realizações necessárias para alcançar as metas estabelecidas. As experiências meta-cognitivas.

A experiência meta-cognitiva está relacionada a consciência das experiências cognitivas e afetivas identificadas em cada envolvimento cognitivo. Este fenômeno inclui as atividades de planejamento, monitoramento, regulação e avaliação. O planejamento envolve a identificação e seleção de estratégias adequadas, a alocação de recursos e a definição de objetivos. O monitoramento e a regulação estão relacionados ao atendimento e a tomada de consciência sobre a compreensão e o desempenho nas tarefas, podendo também envolver o auto teste. E a avaliação está relacionada a revisitar e revisar os próprios objetivos a partir da avaliação do desempenho alcançado e dos processos regulatórios da aprendizagem.

### 3.3.2 O modelo de Brown (1989)

O modelo para meta-cognição proposto por Brown (1989) diferencia-se do proposto por Flavell (1979) ao dividir a meta-cognição em dois componentes: (i) cognição do conhecimento;

e (ii) regulação da cognição. Conforme pode ser visto na Figura 5, estes dois componentes estão intimamente relacionados, sendo um fonte do outro, ao mesmo tempo em que são facilmente diferenciados.

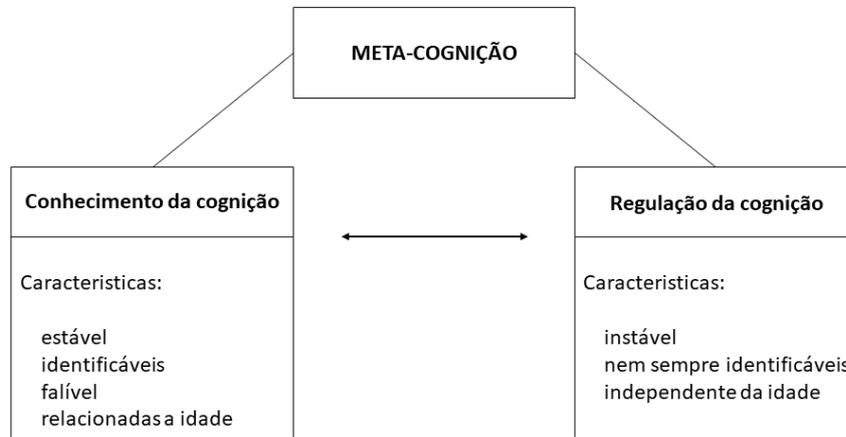


Figura 5. Modelo de meta-cognição de Brown.

Fonte: Traduzido de Brown (1989).

A monitoração do conhecimento está relacionada a atividades que envolvem reflexão consciente sobre habilidades e atividades cognitivas. Refere-se a informações estáveis, identificáveis, por vezes falíveis e desenvolvidas com atraso, que os estudantes tem sobre seus próprios processos cognitivos. Este componente por vezes é identificado como “saber que”.

A regulação da cognição consiste em atividades para regular e vigiar a aprendizagem. Incluem-se entre estes processos as atividades de planejamento para resolução de problemas, de monitoramento durante a aprendizagem e da avaliação dos resultados. Estas atividades são consideradas instáveis, não identificáveis e relativamente independentes da idade do estudante. Ainda, Brown (1989) introduz o estado de piloto automático, que tenta explicar o porquê de estudantes meta-cognitivos não conseguirem identificar com clareza suas estratégias mesmo utilizando-as com efetividade.

### 3.3.3 O modelo de Tobias e Everson (2002)

Tobias e Everson (2002) propõe um modelo no qual a meta-cognição é composta por habilidades e conhecimento, focando-se no conhecimento sobre a cognição, monitoramento dos seus processos cognitivos e de aprendizagem e o controle sobre estes processos. Neste ponto, o modelo proposto por esses autores se diferencia por organizar-se de forma hierárquica,

definindo que a monitoração sobre o conhecimento é pré-requisito para ativação dos demais processos meta-cognitivos (Figura 6).

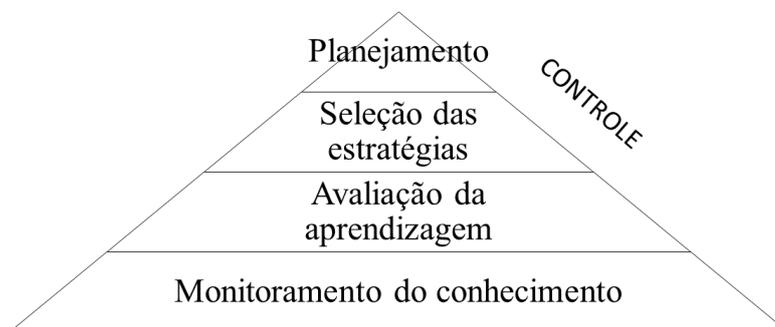


Figura 6. Modelo de meta-cognição de Tobias e Everson.

Fonte: Traduzido de Tobias e Everson (2002).

A monitoração do conhecimento é definida como a capacidade de identificar aquilo que se sabe e aquilo que não se sabe. A partir disso, considera-se que o estudante que realiza esta atividade com sucesso consegue utilizar processos meta-cognitivos mais complexos, como avaliar sua aprendizagem, selecionar estratégias mais adequadas. Por outro lado, quando o estudante não consegue identificar se sabe ou não determinado conteúdo enfrenta dificuldades em concentrar-se em conteúdos novos ou em materiais que auxiliaram sua compreensão (TOBIAS; EVERSON, 2002).

Desta forma, sob esta perspectiva, o estudante que consegue identificar com mais exatidão aquilo que sabe e aquilo que não sabe tem maior vantagem na aprendizagem, pois poderá focar seus esforços em conteúdos que são novos e apenas revisar aqueles que já conhece. Dada a importância do conhecimento que um estudante tem sobre suas próprias capacidades os autores elaboraram um instrumento de avaliação, intitulado *Knowledge Monitoring Accuracy* (KMA) e traduzido nesta tese como Precisão no Monitoramento do Conhecimento, que permite inferir a percepção que o estudante tem sobre sua compreensão a respeito dos conteúdos estudados.

Esta tese irá considerar o modelo de Tobias e Everson (2002) para a compreensão da meta-cognição. Esta escolha se dá pela adequação do modelo ao domínio em questão, a programação e conseqüentemente a resolução de problemas, e pela compreensão de que a monitoração do conhecimento se torna essencial para o uso dos demais processos meta-cognitivos. Além disso, acrescenta-se a disponibilidade de um instrumento de avaliação atrelado a um processo meta-cognitivo específico.

### 3.4 AVALIAÇÃO DE META-COGNIÇÃO

Avaliar a meta-cognição, ou o nível em que cada um de seus componentes é empregado na realização de uma atividade, não é uma tarefa trivial. Dentre os métodos propostos para tal citam-se os questionários, nos quais os estudantes indicam hábitos ou crenças sobre suas experiências ou indicam suas expectativas ao realizar uma atividade, e as entrevistas e auto relatos, nos quais os estudantes são convidados a refletirem sobre suas ações ou a externalizarem seus pensamentos (GAMA, 2004).

Na literatura são encontrados indícios que apontam para o fato de que nenhum dos métodos e métricas utilizadas é infalível (PINTRICH, 2000). Desta forma, defende-se a combinação de diferentes métodos que não compartilhem da mesma fonte. Conforme Schoenfeld (1985), cada instrumento e método de avaliação trará clareza sobre determinado aspecto da meta-cognição, assim um pode complementar as informações identificadas pelo outro.

Nesta tese pretende-se inferir informações sobre dois aspectos da meta-cognição, o monitoramento do conhecimento e o controle da cognição, presentes no modelo de Tobias e Everson (2002). Para isto, serão apresentados nas seções seguintes instrumentos que permitem avaliar estes aspectos, sendo eles o *Metacognitive Awareness Inventory* (MAI), proposto por Schraw e Dennison (1994), e os índices de Precisão no Monitoramento do Conhecimento (KMA), definido por Tobias e Everson (2002) e adaptado por Gama (2004), e *Knowledge Monitoring Bias* (KMB), definido por Gama (2004) e traduzido nesta tese como Viés no Monitoramento do Conhecimento.

#### 3.4.1 Metacognitive Awareness Inventory

O Metacognitive Awareness Inventory (MAI), proposto por Schraw e Dennison (1994), é um instrumento de auto-relato utilizado para medir a consciência meta-cognitiva de um indivíduo. O instrumento é um questionário composto de 52 questões nas quais o respondente deve informar o quanto aquela afirmação é verdadeira para si a partir de uma escala que pode variar entre 1 e 5 ou 1 e 10, que é o caso do instrumento utilizado por esta tese (ANEXO A).

Estas 52 questões são agrupadas de forma a avaliar 2 componentes maiores: (i) *o conhecimento cognitivo*, relacionado a consciência do indivíduo sobre seus pontos fortes e fracos e o conhecimento sobre estratégias e porquê e quando utilizá-las; e (ii) *a regulação da cognição*, relacionada ao planejamento, implementação, monitoramento e avaliação da estratégia a ser utilizada.

O componente *conhecimento cognitivo* agrupa três subescalas, definidas como: (a) conhecimento declarativo: o conhecimento sobre si e sobre estratégias; (b) conhecimento condicional: o conhecimento sobre o porquê e quando utilizar determinada estratégia; e (c) conhecimento processual: o conhecimento sobre como utilizar determinada estratégia. Já o componente *regulação da cognição* agrupa cinco subescalas, definidas como: (a) planejamento; (b) gestão da informação; (c) monitoramento; (d) depuração; (e) avaliação.

O resultado do instrumento é obtido a partir da média aritmética obtida para cada subescala. Ainda, os resultados podem ser interpretados sob três perspectivas, o resultado geral, referente a consciência meta-cognitiva do indivíduo, por componente, referente a especificidade dos dois componentes ou por subescala, referente a cada uma das subescalas específicas.

### 3.4.2 Precisão no Monitoramento do Conhecimento (KMA)

Tobias e Everson (2002) propuseram o índice KMA com o objetivo de inferir a capacidade de monitoração do conhecimento, um pré-requisito para ativar os demais processos meta-cognitivos. Na versão original os estudantes deveriam indicar antes de realizar uma atividade se iriam conseguir resolvê-la ou não. Assim, o KMA é o resultado da comparação entre o que o estudante crê que consegue resolver e aquilo que ele realmente consegue.

O índice KMA é obtido a partir de diversas situações em que o estudante deve avaliar sua capacidade de resolver um problema e é calculado conforme a Equação 1. Nesta equação *a* representa o número de vezes em que o estudante previu que teria sucesso e o obteve; *b* o número de vezes em que previu que irá fracassar e obteve sucesso; *c* o número de vezes em que previu que iria obter sucesso e fracassou e; *d* representa o número de vezes em que previu que iria fracassar e fracassou.

$$KMA = ((a + d) - (b + c)) / (a + b + c + d)$$

Equação 1. Knowledge Monitoring Accuracy original

Fonte: Tobias e Everson (2002).

Os possíveis valores para o índice KMA variam entre -1 e 1, sendo que 1 indica que o estudante possui grande habilidade em prever seu desempenho e -1 que ele tem dificuldades em fazer isso. Acrescenta-se o valor 0, que indica que esta habilidade é variante (Quadro 3).

Quadro 3. Intervalos possíveis no índice KMA.

Desempenho	Estimativa	
	Resolverá com sucesso	Não resolverá com sucesso
Correto	1	-1
Incorreto	-1	1

Fonte: Tobias e Everson (2002).

Diante da pouca flexibilidade do índice KMA proposto por Tobias e Everson (2002), Gama (2004) propôs inserir a variante parcial tanto na estimativa quanto no desempenho. Desta forma torna-se possível diferenciar entre o estudante que consegue prever parcialmente seu resultado e aquele que consegue prever com sucesso. Ainda, nesta adaptação o KMA é calculado a partir da média de KMA obtida para um determinado ciclo, como um conteúdo em específico. Para isso, Gama (2004) propôs novos valores possíveis para o índice KMA (Quadro 4).

Quadro 4. Intervalos possíveis no índice KMA revisito.

Desempenho	Estimativa		
	Resolverá com sucesso	Resolverá parcialmente	Não resolverá com sucesso
Correto	1	-0,5	-1
Parcialmente correto	-0,5	1	-0,5
Incorreto	-1	-0,5	1

Fonte: Gama (2004).

Gama (2004) propõem que um rótulo seja definido para o KMA do estudante, de forma a facilitar sua reflexão. Para isso, foram atribuídas 3 classificações possíveis, estudante com alto, médio ou baixo KMA, conforme pode ser visto no Quadro 5.

Quadro 5. Classificação dos estudantes com base no KMA.

<b>KMA</b>	<b>Classificação</b>	<b>Interpretação</b>
[-1,-0.25]	KMA baixo	O estudante não estima seu conhecimento corretamente na maioria das situações.
[-0.25,0.5]	KMA médio	O estudante estima seu conhecimento corretamente com frequência, porém em algumas situações estima de forma parcialmente correta ou completamente errada.
[0.5,1]	KMA alto	O estudante estima seu conhecimento corretamente em grande parte do tempo.

Fonte: Gama (2004).

### 3.4.3 Viés no Monitoramento do Conhecimento (KMB)

O índice KMB, proposto por Gama (2004), foi criado com o intuito de trazer mais informações sobre os motivos que levam o estudante a avaliar seu conhecimento com ineficiência, representados nesse caso por um viés na avaliação. Este índice é dependente do KMA e também é calculado a partir da média de KMB obtido em todos os exercícios de um determinado ciclo ou conteúdo.

Assim como o índice KMA, o valor de KMB pode assumir valores identificados a partir de uma matriz 3x3 (Quadro 6). Estes valores representam que o estudante pode ser realista (0), muito (-1) ou parcialmente pessimista (-0.5) e muito (1) ou parcialmente otimista (0.5).

Quadro 6. Intervalos possíveis no índice KMB.

<b>Desempenho</b>	<b>Estimativa</b>		
	Não resolverá com sucesso	Resolverá parcialmente	Resolverá com sucesso
Incorreto	0	0,5	1
Parcialmente correto	-0,5	0	0,5
Correto	-1	-0,5	0

Fonte: Gama (2004).

Ainda, assim como para o índice KMA, Gama (2004) propôs um rótulo para classificação dos estudantes conforme o viés na estimativa de seu conhecimento. Esta classificação tem por objetivo auxiliar a compreender a visão que o estudante tem de suas próprias capacidades (Quadro 7).

Quadro 7. Classificação dos estudantes com base no KMB.

KMB	Classificação	Interpretação
KMA alto	Realista	O estudante estima corretamente seu conhecimento.
[0.25,1]	Otimista	O estudante tende a estimar que irá resolver os problemas com sucesso, mas não consegue na maioria das vezes.
[-1,-0.25]	Pessimista	O estudante tende a estimar que não irá resolver os problemas corretamente, mas na maioria das vezes consegue.
[-0.25,.0.25]	Aleatório	A estimativa do estudante sobre seu desempenho é ora otimista e ora pessimista.

Fonte: Gama (2004).

Os três instrumentos apresentados permitem inferir informações essenciais para auxiliar o estudante na compreensão e acompanhamento de sua própria aprendizagem. O MAI permitirá analisar quais são as estratégias de aprendizagem utilizadas pelo estudante e qual é a sua consciência meta-cognitiva. Os índices KMA e KMB possibilitarão inferir o quanto o estudante monitora seus conhecimentos.

Dentre os instrumentos citados, o KMA é identificado como uma medida natural e robusta que apresenta confiabilidade e validade. Além disso, quando comparado a outros instrumentos como o MAI, o KMA apresenta maior correlação com os resultados obtidos na aprendizagem. Outro ponto importante do KMA é que ele é menos suscetível a problemas de compreensão por parte dos estudantes, dado que ele refere-se exclusivamente a avaliação que o estudante tem sobre suas capacidades (PINTRICH, 2000; GAMA, 2004).

### 3.5 SCAFFOLDING META-COGNITIVO

Diante da importância que os processos meta-cognitivos têm na aprendizagem, trabalhos que buscam dar suporte ao desenvolvimento destas habilidades têm apresentado resultados relevantes. Estes trabalhos têm focado em dar suporte gradual aos estudantes de forma que em dado momento eles mesmos consigam monitorar a sua aprendizagem e atinjam resultados satisfatórios (LUCKIN; HAMMERTON, 2002).

Esta abordagem, intitulada *scaffolding*, consiste em dar auxílio aos estudantes a resolver problemas ou realizar tarefas que eles não conseguem sozinhos e diminuir o grau de auxílio à medida em sua suas habilidades aumentam. O *scaffolding* pode ser descrito também como um conjunto de ferramentas, estratégias e guias, humanos ou computacionais, que dão suporte na

compreensão de determinado tema para além da compreensão possível naquele exato momento (WOOD *et al*, 1976).

Quando o auxílio dado está relacionado ao aprimoramento de processos meta-cognitivos, intitula-se *scaffolding* meta-cognitivo. Neste caso, utilizam-se diferentes estratégias, que podem ser instrumentos analógicos, *prompts*, *templates* ou Sistemas Tutores Inteligentes, que estimulam o estudante a refletir sobre sua própria aprendizagem. O objetivo deste tipo de *scaffolding* é dar suporte para que os estudantes se tornem independentes, auto-suficientes e menos dependentes de instruções. É importante salientar que o *scaffolding* também pode ser aplicado quanto ao aprendizado de um determinado conteúdo, ao uso de uma ferramenta ou na adaptação a diferentes contextos, lugares ou culturas (AZEVEDO, HADWIN, 2005; GAMA, 2004).

Neste contexto, Luckin e Hammerton (2002) apresentam o Ecolab 2, que é um ambiente interativo de aprendizagem sobre teias e cadeias alimentares. Na sua primeira versão, Ecolab 1, o sistema implementava um assistente para a realização das atividades propostas. Este assistente, observando o nível de ajuda que o estudante necessitava para completar uma tarefa, disponibilizava exercícios adaptados para o seu nível de conhecimento. Para isso, o sistema mantinha um modelo do estudante no qual eram representados os componentes curriculares daquele conteúdo na forma de um grafo. Cada nodo do grafo representava um conteúdo e tinha atrelado a ele duas marcações, uma referente à crença do sistema sobre o quanto o estudante possuía independência sobre o tema e outra sobre a quantidade de ajuda que o estudante necessitou para completar as atividades referentes aquele conteúdo. A partir deste modelo o assistente inferia o potencial do estudante e poderia então identificar quais seriam os próximos conteúdos a serem estudados.

O Ecolab 2 tem por objetivo dar suporte aos processos meta-cognitivos auxiliando os estudantes na busca por ajuda, na seleção de exercícios adequados e na reflexão sobre seu desempenho. Para isso, o Ecolab 2 adicionou às informações mantidas no modelo do estudante do Ecolab 1 a crença do sistema sobre a consciência do estudante a respeito de suas necessidades de aprendizagem. Assim, além da assistência disponível no Ecolab 1 que auxiliava o estudante a selecionar o que aprender em sequência, o Ecolab 2 se preocupa em fazê-lo refletir sobre suas dificuldades, sobre quais atividades deve desenvolver e sobre quais conteúdos deve estudar a partir de mensagens de ajuda, sendo elas:

- Não esqueça que o Ecolab pode ajudá-lo;
- Por que não pedir ao Ecolab por mais ou menos ajuda?
- Tente a ajuda do nível X.

Um experimento foi conduzido com 32 estudantes, com idades entre 9 e 11 anos, que já haviam estudado cadeias e teias alimentares. No experimento os estudantes responderam a um pré-teste escrito e também no Ecolab 2 e preencheram um questionário com informações sobre sua experiência com computadores. Em seguida os estudantes foram encorajados a explorar os assuntos e exercícios disponíveis na ferramenta. Por fim, os estudantes responderam novamente ao mesmo teste na ferramenta e de forma escrita. A partir do experimento foi possível concluir que no Ecolab 2 os estudantes menos capazes obtiveram ganho maior entre o pré-teste e o pós-teste do que no Ecolab 1. Desta forma, evidencia-se a contribuição dada pelo estímulo a reflexão sobre a aprendizagem naqueles estudantes que possuem deficiência em realiza-la.

A programação de computadores está relacionada à resolução de problemas, que envolve passos como a compreensão do problema, a busca e avaliação de problemas ou soluções similares, a implementação e a avaliação da solução proposta, e com isso monitorar, planejar, refletir e avaliar suas ações de forma eficiente torna-se imprescindível para o aprendiz alcançar sucesso (BERGIN; REILLY; TRAYNOR, 2005; LOKSA; KO, 2016). Neste contexto, o *scaffolding* meta-cognitivo também tem apresentado resultados relevantes.

Borges, Moraes e Okuyama (2013) apresentam o resultado de uma prática docente para aprendizagem de desenvolvimento de sistemas envolvendo a metodologia ágil Scrum. Na atividade, os estudantes deveriam desenvolver um projeto Web utilizando a metodologia Scrum, na qual são previstas as atividades e o tempo para desenvolvê-las (sprints). Conforme pode ser visto na Figura 7, uma ficha de controle do andamento do trabalho deveria ser preenchida por cada grupo.

### Avaliação do Sprint

Projeto: \_\_\_\_\_

**Casos de Uso do Projeto:**

- 1- \_\_\_\_\_  
 2- \_\_\_\_\_  
 3- \_\_\_\_\_  
 4- \_\_\_\_\_

**Evolução da Implementação dos Casos de Uso**

Casos de Uso	Implementação do modelo objeto-relacional			Implementação do CRUD			Regras de Negócio			Usabilidade			Tratamento de Erros e Exceções		
	☹	😊	😄	☹	😊	😄	☹	😊	😄	☹	😊	😄	☹	😊	😄
1															
2															
3															
4															

**Evolução dos Sprints**

	Sprint1	Sprint2	Sprint3	Sprint4
Team lider				
Data				
Objetivos				
Conclusão				

Figura 7 – Ficha de acompanhamento de sprint.

Fonte: Borges, Moraes e Okuyama (2013).

O resultado da prática docente indicou que ao fazer com que os estudantes acompanhassem seu desempenho, refletissem sobre seu comprometimento e sobre as estratégias adotadas a prática obteve resultados positivos. Dentre estes resultados evidencia-se mudanças em relação a prioridade das tarefas e alocação do tempo.

França e Tedesco (2014) desenvolveram o modelo PenC para apoiar estudantes do ensino médio na resolução de problemas associados ao Pensamento Computacional por meio do estímulo de processos meta-cognitivos. O modelo proposto é composto por 4 fases: (a) pré-reflexão: antecede a resolução de um problema e envolve a reflexão sobre indicadores de sua própria aprendizagem e a sua própria avaliação sobre seus conhecimentos (KMA e KMB); (b) resolução: envolve a resolução de problemas com aporte de prompts de perguntas que convidam à reflexão geradas a partir de sua experiência na resolução de exercícios anteriores; (c) avaliação por pares: nesta fase a solução enviada para o problema é avaliada pelos colegas, que também tem o suporte de prompts para refletir sobre a avaliação; e (d) pós-reflexão: nesta fase o estudante visualizar e reflete sobre a avaliação dos colegas ao seu problema e também visualiza um comparativo entre a sua expectativa ao resolver o problema e o resultado obtido (Figura 8).

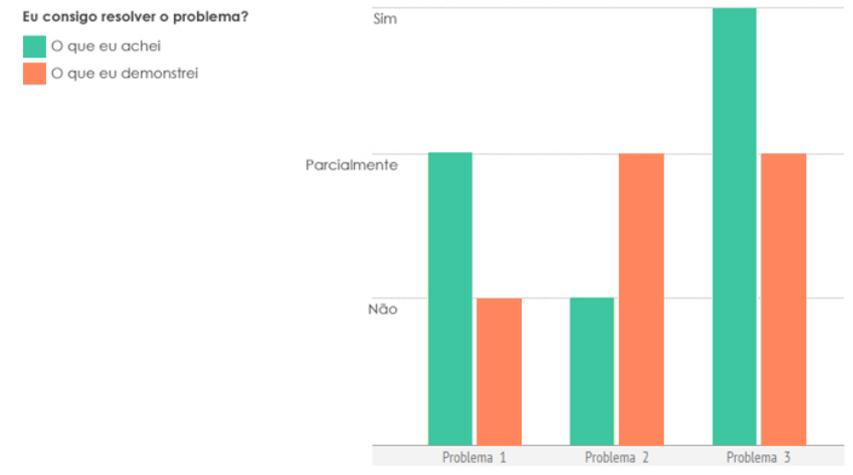


Figura 8 – Comparação entre expectativa e resultado obtido.

Fonte: França e Tedesco (2014).

O modelo PenC foi integrado ao ambiente PenC+, que disponibiliza exercícios de programação, para a realização de um quase-experimento em um curso de desenvolvimento de jogos digitais. Nesse experimento, um grupo utilizou o PenC+ com o modelo PenC implementado e outro sem o modelo. A partir dos resultados obtidos através do índice KMA, KMB e do questionário MSLQ, verificou-se que o grupo experimental teve maior variação do índice KMB, ou seja, os estudantes tornaram-se mais realistas acerca de seus conhecimentos, além de que seus processos meta-cognitivos, avaliados com o questionário MSLQ, foram mais estimulados entre o pré-teste e o pós-teste.

Para Loksa e Ko (2016) o ensino de programação deve também englobar o ensino das habilidades necessárias para a resolução de problemas. Desta forma, propuseram uma intervenção que consistia em: (i) apresentar instruções explícitas sobre a resolução de determinado problema, indicando os pequenos objetivos que deveriam ser atingidos para alcançar o objetivo maior; (ii) acompanhar o progresso da resolução do problema fazendo com que toda vez que fosse solicitada ajuda o estudante devesse informar em que estágio do desenvolvimento estava ao analisar um indicador físico, fazendo-o refletir sobre isso; e (iii) suporte sensível ao contexto por meio de uma IDE que, ao identificar erros de sintaxe de programação em uma linha, dá dicas de solução.

Um experimento foi realizado com 48 estudantes durante 2 semanas de um curso de programação. Ao final de cada dia do curso os estudantes respondiam a questões acerca de sua consciência meta-cognitiva (“Como você resolveu este problema?”), sua auto-eficácia (“Eu consigo escrever códigos em Javascript corretamente”) e sua confiança (“Eu não consigo mudar minha atitude sobre programação”) e também analisava-se a produtividade através dos logs obtidos na IDE. Por meio do experimento identificou-se que dar suporte explícito e sensível ao

contexto e estimular a reflexão acerca da solução do problema potencializa a produtividade, a independência, a consciência meta-cognitiva e a confiança dos estudantes.

O *scaffolding* meta-cognitivo, sobretudo para a aprendizagem de programação, implica em dar suporte gradual ao estudante, seja por meio de um sistema ou tutor que o auxilia a refletir sobre sua solução ou monitorar seu progresso. Este suporte gradual pode ser automático, quando o sistema identifica que o estudante adquiriu habilidades necessárias e não precisa mais auxiliá-lo em uma determinada tarefa, pode ser semi-automático, quando a ajuda é solicitada pelo estudante, ou pode ser realizado por meio de recursos que o estudante pode utilizar conforme sua necessidade.

Em relação ao terceiro cenário, no qual o recurso para suporte à meta-cognição fica ao alcance do estudante, trabalhos que disponibilizam o modelo do conhecimento do estudante registrado por um sistema computacional, e até mesmo permitem que ele compare o seu modelo com o dos outros colegas, têm apresentado resultados relevantes no apoio ao desenvolvimento de habilidade meta-cognitivas. O uso do Modelo Aberto do Estudante e Modelo Social Aberto do Estudante, quando os modelos dos colegas também são visíveis, tem demonstrado capacidade de estimular a reflexão e o monitoramento da aprendizagem (BULL; KAY, 2010). Desta forma, no próximo capítulo estes tópicos serão aprofundados.

### 3.6 CONSIDERAÇÕES SOBRE O CAPÍTULO

A aprendizagem de programação está relacionada à resolução de problemas, que exige processos meta-cognitivos para compreender o problema, para planejar, implementar e avaliar a solução proposta. Com isso, os estudantes que conseguem monitorar sua aprendizagem de forma mais eficiente, tomando consciência de suas capacidades e dificuldades, alcançam melhores resultados na aprendizagem.

Neste capítulo definiu-se o modelo de meta-cognição que será adotado para a Tese, sendo este o modelo proposto por Tomas e Everson (2002). Esses pesquisadores elaboraram também um índice de avaliação que permite inferir o quanto o estudante consegue estimar seu conhecimento sobre um assunto (KMA), uma característica determinante para o uso efetivo dos processos meta-cognitivos. Acrescenta-se ainda o índice KMB (GAMA, 2004), referente ao viés inculcido na avaliação pela percepção positiva ou negativa que o estudante tem sobre sua aprendizagem.

Também neste capítulo foi discutida a contribuição do *scaffolding* meta-cognitivo, o apoio gradual para o desenvolvimento da meta-cognição, para a aprendizagem de programação.

Este apoio pode ser dado por meio de sistemas inteligentes ou através da disponibilização de recursos que auxiliam o estudante a refletir e monitorar sua aprendizagem, como por meio da interação com o Modelo Social Aberto do Estudante.

#### 4 MODELO ABERTO DO ESTUDANTE

A representação dos usuários em um ambiente online é o meio pelo qual estes se identificam e conhecem uns aos outros. No contexto de sistemas que apresentam características de personalização, conhecidos como Sistemas Adaptativos, esta representação, intitulada modelagem ou modelo do usuário, é um elemento essencial. O modelo, que registra o estado atual do usuário no domínio do sistema, suas preferências e informações, é o meio pelo qual os mecanismos de personalização adaptam trajetórias de aprendizagem, geram recomendações de materiais ou reorganizam interfaces (BRUSILOVSKY; MILLÁN, 2007).

Na maioria dos sistemas adaptativos o modelo não é disponibilizado para o usuário. Porém, esta iniciativa tem apresentado indícios de que pode estimular a auto regulação e motivar o estudante no processo de aprendizagem. Este modelo, conhecido como Modelo Aberto do Estudante (MAE), do inglês *Open Learner Model*, se caracteriza por ser possível de ser visualizado pelo estudante e por seus tutores. Na maioria das vezes é apresentado de forma visual e em algumas vezes permite interação (BULL; PAIN, 1995; BULL; NGHIEM, 2002; BULL; KAY, 2010). O MAE pode ser útil para de diferentes formas, como:

- Promover atividades meta-cognitivas como reflexão, planejamento e auto monitoramento;
- Permitir ao estudante maior controle e responsabilidade sobre sua aprendizagem, encorajando a sua independência;
- Facilitar o acesso a materiais dentro de um ambiente virtual;
- Dar suporte à avaliação do desempenho do estudante;
- Aumentar a eficiência do modelo do estudante, considerando que este pode adicionar informações ou corrigi-las;
- Dar transparência ao sistema, permitindo que o usuário visualize as informações que são mantidas sobre ele pelo sistema; e
- Aumentar a confiança na personalização fornecida pelo sistema, apresentando quais são as informações inferidas pelo sistema para realizar adaptações.

O MAE não consiste apenas em tornar disponível para o usuário as informações que um sistema registra sobre ele, dado que estas informações podem ser de difícil compreensão, mas também apresentá-las por meio de uma interface de fácil navegação. A maioria dos modelos de usuário podem ser abertos e a forma de visualizá-los pode variar conforme o propósito de torná-los acessíveis, os usuários alvos, o contexto da aprendizagem e as atividades a serem desenvolvidas (BULL; KAY, 2007).

A Figura 9 apresenta uma ferramenta de visualização para MAE intitulada *IntrospectiveViews* para um estudante em uma disciplina de programação Orientada a Objetos (HSIAO *et al.*, 2011). Nesta ferramenta os tópicos de um determinado domínio são representados na forma de uma fatia de pizza. As cores de cada fatia variam conforme o desempenho do estudante nas atividades propostas e no acesso aos materiais disponíveis em um ambiente virtual, sendo a cor verde referente ao bom desempenho e a cor vermelha ao mau desempenho.

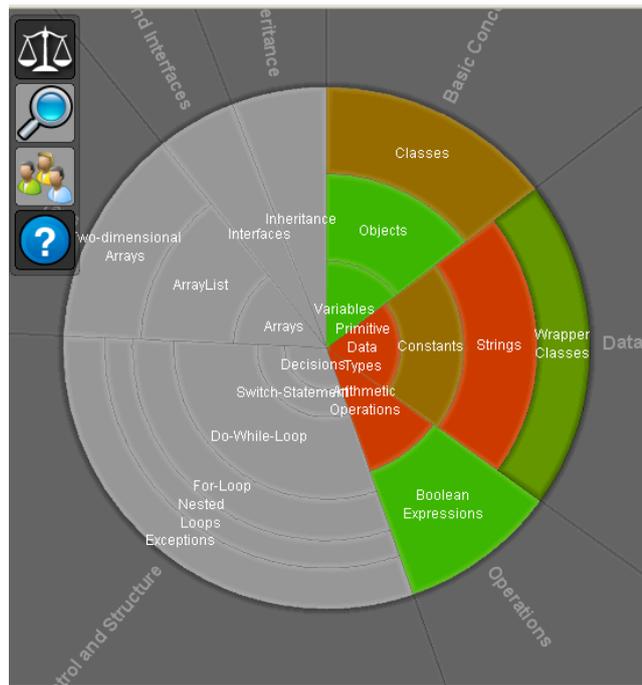


Figura 9 – Exemplo de visualização de Modelagem Aberta do Estudante.

Fonte: Hsiao *et al.*, 2011.

Conforme Bull e Kay (2016), não há um padrão para a representação gráfica do MAE. Dentre as representações encontradas na literatura podem ser citados mapas nos quais cada ilha representa um conceito e o caminho entre as ilhas o progresso do estudante, barras que representam o domínio de um usuário sobre determinado tema (Figura 10), mapas mentais, nuvens de *tags* ou simples tabelas.

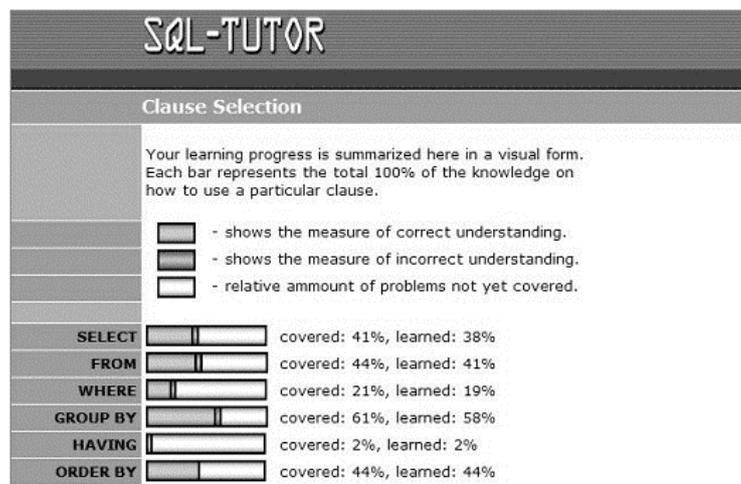


Figura 10. Representação gráfica do MAE para aprendizagem de SQL.

Fonte: Mitrovic e Martin (2007).

Outra proposta de uso da MAE é o seu compartilhamento com os demais membros de uma plataforma virtual, conhecido como Modelo Social Aberto do Estudante (MSAE), ou do inglês *Open Social Student Modeling*. Esta ideia, proposta com base na Teoria da Comparação Social, dá ao estudante a possibilidade de comparar o seu progresso com o de um colega, de um grupo específico ou de uma turma toda.

Conforme a Teoria da Comparação Social, proposta por Festinger (1954), os indivíduos tendem a comparar suas ações ou conquistas com aqueles que eles julgam serem semelhantes a eles de alguma forma. Caso não sejam encontrados pares, a comparação do indivíduo é realizada em escala muito menor. Por outro lado, quando um indivíduo sabe que os demais estão se comparando com ele tende a ter atitudes mais responsáveis.

A comparação com outro indivíduo pode levar à competição, que de certa forma pode ter efeitos negativos no processo de ensino e de aprendizagem. Porém, a comparação com indivíduos que apresentam melhor desempenho pode fazer com que o estudante o considere um modelo a ser seguido, fazendo da comparação um instrumento para instigar o desenvolvimento próprio.

O uso da MSAE para promover a comparação social entre estudantes em ambientes virtuais tem demonstrado que pode diminuir o desgaste das relações entre membros de uma comunidade, aumentar a produtividade e a motivação para aprender e estimular maior participação nas atividades propostas (BULL; PAIN, 1995; SHEPHERD *et al.*, 1996; LOBODA *et al.*, 2014; BRUSILOVSKY *et al.*, 2015). Ainda, além das contribuições já proporcionadas pelo MAE, o MSAE pode ser útil ao estudante de diferentes formas, como:

- Promover ou dar suporte a interações colaborativas ou competitivas entre grupos de estudantes;
- Facilitar as interações entre estudantes e tutores; e
- Promover o acesso a materiais que contribuíram para a aprendizagem de outros colegas.

Assim como para o MAE, a visualização dos modelos é fundamental para o MSAE. Erickson (2003) propôs seis reivindicações a respeito da visualização em contextos sociais elencadas a partir de sua experiência com o tema, sendo elas: (i) que todos os usuários vejam as mesmas informações que os demais, sem possibilidade de customização do que pode ou não ser compartilhado; (ii) que sejam dispostas informações que permitam a cada usuário sua interpretação dos fatos; (iii) é importante que seja possível reproduzir comportamentos sociais do cotidiano, como ignorar um material ou fingir interesse; (iv) deve ser possível visualizar as informações em um contexto micro e macro; (v) o foco deve ser em prover informações, deixando a inferência por conta dos usuários; (vi) o usuário compreenderá o sistema de visualização a partir do reflexo de suas próprias ações, com isso é importante que ele próprio também visualize o seu perfil.

Neste contexto, Vassileva e Sun (2008) apresentam a comunidade online Comtella, que permite que os usuários pertencentes a um laboratório ou turma compartilhem, pesquisem e avaliem recursos online (arquivos ou urls). Além disso, com o objetivo de investigar formas de motivar a participação de usuários em comunidades online, a plataforma incorpora um mecanismo de visualização de MSAE.

Na comunidade Comtella cada usuário é representado na forma de uma estrela em uma constelação. Existem quatro níveis de participação na comunidade, remetendo a reputação, sendo cada uma delas representada por meio de diferentes cores. Os usuários com maior reputação são os usuários da categoria ouro (amarelo), seguidos por prata (branca), bronze (vermelha) e plástico (verde), sendo esta última a categoria inicial de todos participantes. Além disso, o brilho da estrela corresponde à qualidade das contribuições, sendo a mais brilhante a que possui maior qualidade. Ainda, para representar quantidade de contribuições as estrelas variam de tamanho (Figura 11).

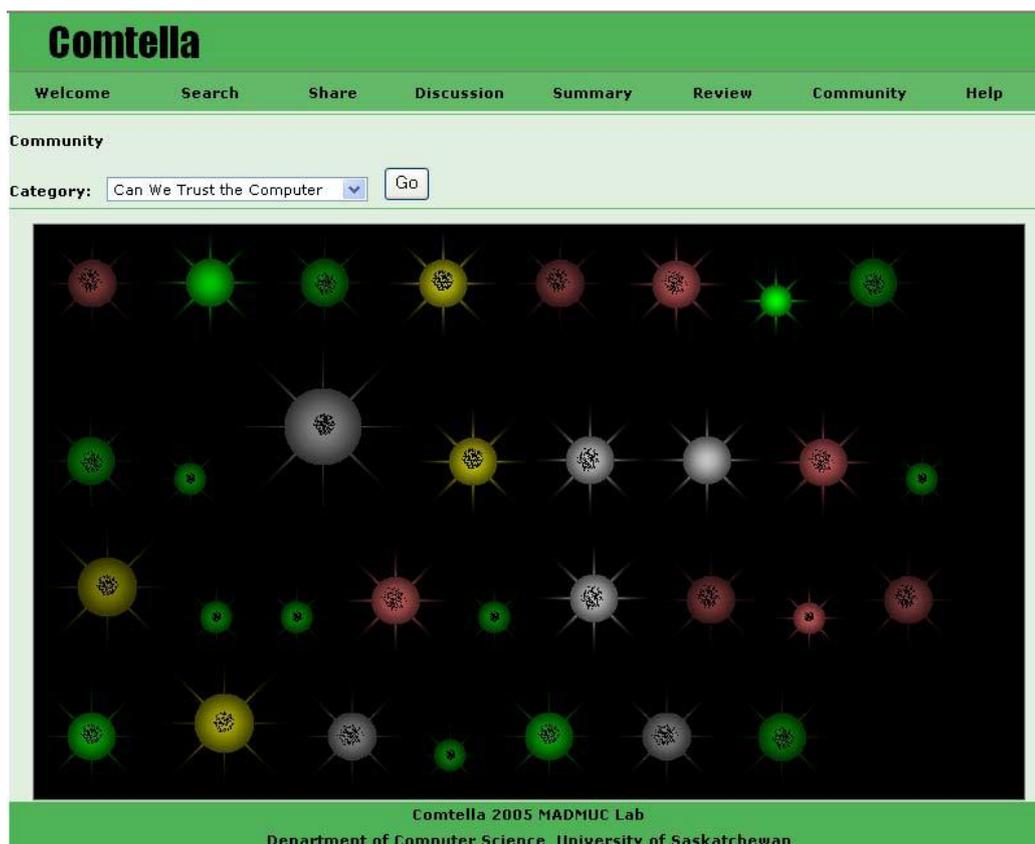


Figura 11. Apresentação dos usuários da comunidade Comtella.

Fonte: Vassileva e Sun (2007).

Em um experimento conduzido com dois grupos, sendo um deles um grupo de controle, verificou-se que a interface de visualização do MSAE motivou maior participação dos usuários na comunidade. Porém, é importante destacar que este resultado se mostrou mais aparente em usuários que apresentam características competitivas. Além disso, concluiu-se que é importante que a visualização reflita qual tipo de contribuição é mais valiosa para a comunidade, fazendo com isso com que o valor da comunidade cresça.

Existem duas correntes principais de pesquisa sobre MAE e MSAE, a primeira tem como foco dar suporte à auto reflexão e ao planejamento dos estudantes por meio da visualização do seu modelo e a segunda se preocupa em encorajar os estudantes a participar da construção do seu modelo, estimulando com isso a colaboração e a negociação. Em ambas as correntes os resultados de experimentos indicam a promoção de maior acesso aos conteúdos disponíveis, da interação nos ambientes virtuais e da percepção sobre o desempenho na aprendizagem (HSIAO *et al.*, 2011). Esta tese está relacionada à primeira linha de pesquisa, na qual a visualização do desempenho, tanto o próprio quanto o dos colegas, busca estimular o desenvolvimento de processos meta-cognitivos, essenciais para o sucesso da aprendizagem.

A MSAE cria uma ruptura com a modelagem tradicional do estudante, tornando visíveis aspectos antes disponíveis apenas para o sistema e para o próprio estudante. Desta forma, torna-se uma ferramenta que pode auxiliar na tomada de consciência e no monitoramento da aprendizagem (GUERRA *et al.*, 2015). Bull e Kay (2013) relatam diferentes implementações da MSAE que contribuem para o uso de processos meta-cognitivos, como indicadores de desempenho, possibilidades de negociação do modelo entre o estudante e o sistema e a possibilidade de visualização de desempenho em diferentes níveis de detalhe. Ainda, ao tornar o modelo do estudante visível para os colegas torna-se possível encontrar colaboradores, comparar os modelos e adquirir consciência de grupo.

Bull e Kay (2007) apresentam vantagens de tornar o modelo do estudante aberto em relação ao desenvolvimento dos processos meta-cognitivos:

- Melhorar a precisão do modelo do aluno, permitindo que ele contribua para isso;
- Promover a reflexão ao confrontar o estudante com a representação do seu conhecimento;
- Facilitar o planejamento e o monitoramento da aprendizagem;
- Facilitar a colaboração entre os estudantes;
- Facilitar a competição entre os estudantes;
- Apoiar a navegação no ambiente de aprendizagem;
- Dar o direito de acesso a informações armazenadas sobre si;
- Dar controle e responsabilizar o estudante pela sua aprendizagem;
- Promover a confiança no modelo do estudante;
- Estimular a avaliação formativa; e
- Justificar a avaliação sumativa.

Assim, ao possibilitar que o estudante visualize uma representação das suas habilidades, de seu acesso aos materiais e do seu progresso no desenvolvimento das atividades propostas estimula-se os processos meta-cognitivos, ele poderá planejar, monitorar, avaliar e readequar suas estratégias de aprendizagem. Ainda, ao permitir a visualização do modelo dos colegas, o estudante é estimulado a refletir sobre seu progresso em relação ao dos outros, fazendo com que ele reveja o quanto tem sido frequente no acesso aos materiais propostos e compare seu desempenho nas atividades.

O trabalho de Hsiao *et al.* (2011) apresenta a incorporação da MSAE ao QuizzJET, uma ferramenta para produção e disponibilização de questões parametrizadas acerca de programação em Java. A visualização da MSAE proposta neste trabalho se dá por meio da

ferramenta *IntrospectiveViews*, já apresentada na Figura 9. Conforme pode ser visto na Figura 12, durante a interação com a ferramenta o estudante pode acompanhar o seu progresso e compará-lo com o de outros colegas ou com a média da turma. No experimento relatado por Hsiao *et al.* (2011), que durou um semestre, verificou-se que os estudantes utilizaram o recurso tanto para acompanhar seu progresso quanto para navegar no ambiente virtual. Além disso, foi possível identificar maior sucesso nas respostas às questões quando comparadas com o sistema sem o recurso de *IntrospectiveViews*.



Figura 12 – Exemplo de visualização de Modelagem Social Aberta do Estudante.

Fonte: Hsiao *et al.*, 2011.

Os trabalhos de Loboda *et al.* (2014), Brusilovsky *et al.* (2015) e de Guerra *et al.* (2016) relatam experimentos com uma interface de código aberto para MSAE intitulada *MasteryGrids*. Esta interface permite visualizar o modelo do estudante, de seus colegas, de seus grupos e de sua turma. Conforme pode ser visto na Figura 13, as colunas da interface representam os tópicos e as linhas o progresso no tópico em questão. Ao clicar em uma célula é possível visualizar em detalhes o desempenho em um tópico específico.

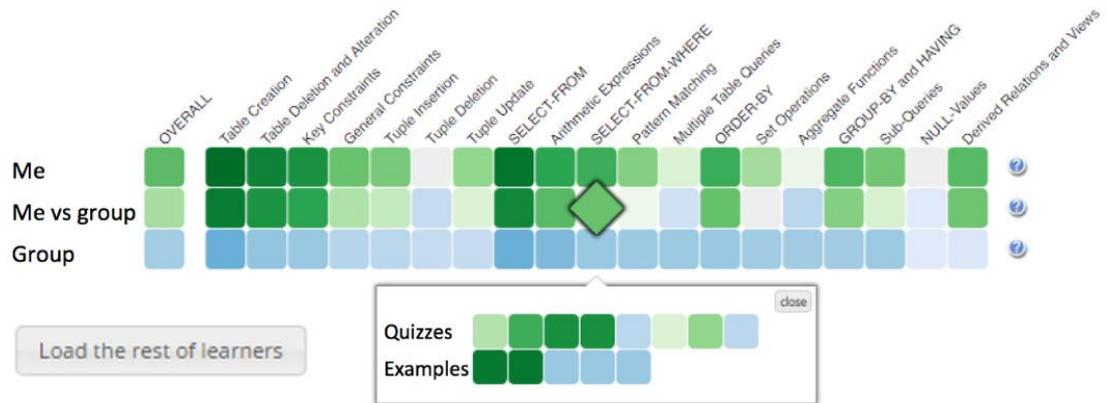


Figura 13 – Exemplo de visualização da MSAE na interface MasteryGrids.

Fonte: Brusilovsky *et al.* (2015).

Ainda que os experimentos realizados com a interface MasteryGrids tenham abordado temas diferentes relacionados à computação, foi possível verificar em todos que os estudantes acessaram mais materiais quando comparado com interfaces de MAE. Além disso, como no experimento relatado por Hsiao *et al.* (2011), o número de acertos quando comparado com a MAE foi maior, sobretudo quando foi analisado o desempenho daqueles alunos com mais dificuldades.

Shi, Cristea e Hadzidedic (2014) incorporaram à plataforma Topolor, um ambiente virtual de aprendizagem adaptativo baseado em aspectos sociais, um mecanismo de visualização do MSAE considerado multifacetado. Esta característica se dá pelo fato de que o usuário pode acessar o modelo dos demais estudantes em diferentes granularidades, partindo de um contexto maior como uma turma até um menor como o de um único recurso. Além disso, o MSAE implementado na plataforma Topolor considera o desempenho dos estudantes (em atividades ou no acesso a materiais) e suas contribuições, considerando eles consumidores e produtores de conhecimento.

A plataforma Topolor permite que os usuários visualizem suas contribuições, como materiais postados e atividades resolvidas, e também compare seus dados com os demais usuários. Nesta comparação com os demais usuários é possível analisar desde a quantidade de materiais compartilhados até a trajetória de aprendizagem dentro do ambiente, sempre levando em consideração a granularidade desejada (Figura 14).

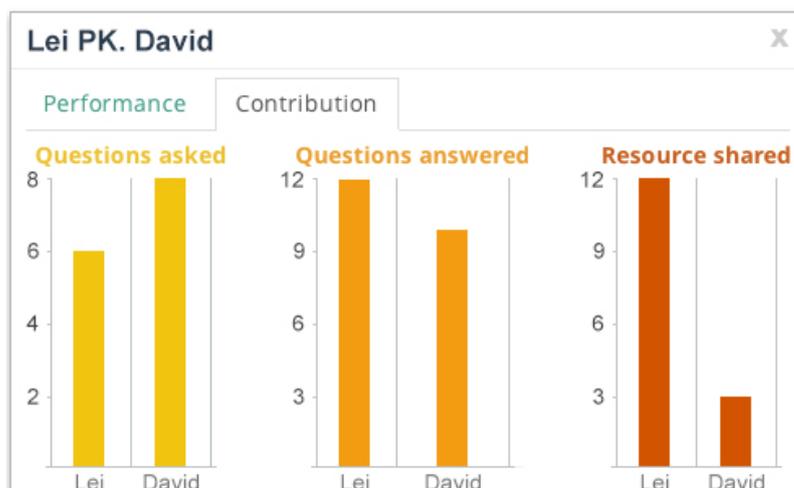


Figura 14 – Comparação entre contribuições de usuários na plataforma Topolor.

Fonte: Shi, Cristea e Hadzidedic (2014).

Um experimento com estudantes de uma turma de Mestrado foi conduzido para avaliar aspectos de usabilidade da plataforma desenvolvida. Os resultados obtidos apontaram para uma boa aceitação da plataforma, ainda que poucos estudantes tenham participado do experimento, sendo 12 o total. Para o futuro da plataforma espera-se investigar o efeito de atividades que estimulem auto regulação no ambiente e também a incorporação de mecanismos de personalização na visualização dos MSAE.

As variadas formas de visualização e implementação do MSAE podem contribuir em diferentes aspectos para o desenvolvimento dos processos meta-cognitivos. Ao permitir que o estudante insira informações sobre seu desempenho no modelo, em um processo de negociação com o sistema, estimula-se o controle e a responsabilidade sobre o próprio processo de aprendizagem. Por outro lado, o MSAE torna-se um recurso útil para promover o monitoramento da própria aprendizagem, pois convida o estudante a refletir e monitorar seu desempenho ao mesmo tempo que permite que ele compare-se com os demais colegas. Ainda, podem ser citadas a possibilidade de o estudante acessar materiais complementares a partir da comparação com os demais colegas e a discussão do seu modelo com um facilitador ou tutor, estimulando desta forma a reflexão.

#### 4.1 CONSIDERAÇÕES SOBRE O CAPÍTULO

O MAE possibilita que os usuários de um sistema visualizem as informações armazenadas sobre eles. Estas informações variam de acordo com o foco do sistema, podendo estas serem referentes ao seu desempenho em atividades, seu acesso aos materiais disponibilizados ou sobre suas interações com os demais colegas. Quando este modelo é compartilhado com os colegas

de uma turma passa a ser chamado de MSAE. Nesta modalidade, os estudantes conseguem visualizar o modelo dos colegas e comparar com seu.

Conforme relatado nos trabalhos apresentados neste capítulo, a possibilidade de visualizar o seu modelo e também o modelo dos colegas traz benefícios para o processo de ensino e de aprendizagem. Dentre estes benefícios está o auxílio ao desenvolvimento de processos meta-cognitivos, essenciais para o sucesso na aprendizagem. Isto é possível porque as ferramentas de visualização dos modelos permitem que o estudante monitore seu desempenho, compare com o dos colegas, avalie seu progresso e readéque suas estratégias.

## 5 PROCEDIMENTOS METODOLÓGICOS

Este Capítulo apresenta os procedimentos metodológicos que foram adotados para alcançar o objetivo desta tese, definido como: investigar como o *scaffolding* meta-cognitivo proporcionado pela interação com o Modelo Social Aberto do Estudante pode contribuir para o processo de aprendizagem de programação.

Quanto ao objetivo, a pesquisa realizada se caracteriza como explicativa na medida em que busca identificar fatores que contribuem para a ocorrência de um fenômeno, apontando relações de causa e efeito (GIL, 2002). Trata-se de um estudo de caráter quasi-experimental com coleta e análise de dados utilizando abordagens quantitativa e qualitativa.

A seção 5.1 descreve a amostra de estudantes que foram selecionados para o experimento, a seção 5.2 especifica os métodos para experimentação utilizados e detalha aspectos relacionados à abordagem da pesquisa e à coleta de dados. Por fim, a seção 5.3 apresenta o Ambiente Virtual de Aprendizagem (AVA) desenvolvido para a realização do experimento proposto.

### 5.1 AMOSTRA INICIAL

Foram selecionadas para participar da pesquisa duas turmas do ensino superior da disciplina de *Introdução à Programação*. Durante um semestre foi realizada uma intervenção com ambas as turmas utilizando o Modelo Aberto do Estudante. A Turma A, do 1º semestre do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistema, registrou a matrícula de 33 estudantes. A Turma B, do 4º semestre do Curso Superior de Engenharia Química, registrou a matrícula de 17 estudantes.

### 5.2 DESIGN DO EXPERIMENTO

Para alcançar o objetivo proposto nesta tese foi adotada a abordagem quali-quantitativa em uma modalidade quasi-experimental sem grupo de controle, com pré-teste e pós-teste. Assim, de forma a viabilizar a coleta de dados, foi desenvolvido um AVA centrado no MSAE que foi utilizado pelas duas turmas durante um semestre.

Neste AVA foram disponibilizados materiais de apoio e também exercícios de programação que poderiam ser visualizados por meio do MSAE. O conteúdo da disciplina foi dividido em cinco tópicos comuns as disciplinas de introdução a programação, sendo eles:

- Introdução: variáveis, tipos de dados, entrada e saída, atribuição e operações aritméticas;
- Condicional: estruturas condicionais SE, SENÃO, SE-SENÃO e operações lógicas;
- Repetição: estruturas de repetição PARA e ENQUANTO, acumuladores e contadores;
- Tipos compostos: vetores e matrizes;
- Modularização: funções com retorno e sem retorno.

Para cada tópico da disciplina foram selecionados dez exercícios, divididos em duas categorias, cinco exercícios centrais e cinco exercícios complementares. Os exercícios centrais são aqueles considerados essenciais para os quais foi disponibilizado tempo em algumas aulas para serem resolvidos. Os exercícios complementares foram definidos como exercícios úteis para a prática fora da sala de aula, como forma de complementar os estudos.

O processo de definição dos exercícios iniciou pela seleção de um conjunto de exercícios pelo pesquisador. Após uma classificação inicial a categorização foi validada por três professores de programação. Assim, foram disponibilizados para os estudantes no AVA cinquenta exercícios de programação divididos por tópicos que deveriam ser realizados livremente durante o semestre.

É importante ressaltar que nos exercícios os estudantes deveriam indicar sua previsão quanto à resolução de cada questão proposta, sendo as opções possíveis: (i) Consigo resolver o exercício corretamente; (ii) Consigo resolver o exercício parcialmente; e (iii) Não consigo resolver o exercício. A partir desta resposta foi calculado o índice KMA e KMB de cada estudante. Ao terminar cada tópico os estudantes também realizaram um teste individual e sem consulta abordando especificamente o conteúdo trabalhado que tinha nota máxima 10. Também foi coletada a nota final de cada estudante na disciplina.

Para enriquecer os dados da pesquisa, foram selecionados sete estudantes para participar de uma entrevista semi-estruturada com base no desempenho final obtido na disciplina (ANEXO B). O professor da disciplina também registrou observações semanais em diário de campo e foram coletadas informações sobre o uso do AVA por meio do sistema de *logs* do MOODLE.

Desta forma, para analisar se a possibilidade de o estudante comparar o seu desempenho com o dos colegas contribui para o desenvolvimento da regulação da própria aprendizagem os estudantes responderam questionário MAI (ANEXO A) ao iniciar e a finalizar o semestre. De

modo a verificar se ocorreram mudanças significativas durante o período na consciência metacognitiva dos estudantes, as respostas ao MAI foram analisadas estatisticamente com o teste *t de Student*.

Para avaliar a possível relação entre o desempenho dos estudantes e a capacidade de regulação da própria aprendizagem, foi analisada a correlação, a partir do cálculo da correlação de Pearson, entre as respostas ao MAI, o desempenho nos testes, o desempenho nos exercícios, o desempenho final na disciplina e os valores de KMA e KMB. Além disso, os estudantes também realizaram um pré-teste e um pós-teste (ANEXO C) com questões que tratavam sobre todos os conteúdos da disciplina. Verificou-se então se houve diferença significativa entre estes resultados utilizando o teste *t de Student*, observando-se ainda a correlação com os demais dados.

Para identificar como os estudantes interagem com o Modelo Social Aberto do Estudante na aprendizagem de programação, foram considerados os dados coletados pelos logs do AVA. Além disso, também foram consideradas as respostas dadas pelos entrevistados e informações registradas no diário de campo durante toda a investigação.

### 5.3 AMBIENTE VIRTUAL DE APRENDIZAGEM

O AVA utilizado nesta tese foi desenvolvido sob a plataforma MOODLE<sup>10</sup> (Modular Object-Oriented Dynamic Learning Environment). Optou-se pela utilização dessa plataforma por ser gratuita e com o código aberto, permitindo assim a realização de personalização conforme a necessidade por meio de *plugins*. A plataforma MOODLE disponibiliza de forma nativa recursos para comunicação síncrona e assíncrona entre os usuários, registro da navegação no ambiente por meio de *logs*, recursos para criação, correção e envio de atividades e a possibilidade de inserir materiais de diferentes mídias e extensões.

Os exercícios de programação e os materiais de apoio foram disponibilizados no AVA de forma que apenas os estudantes que possuem acesso ao curso de Programação pudessem acessá-los. Foram elaboradas duas versões do Ambiente Virtual de Aprendizagem, cada uma utilizando um *plugin* diferente para a implementação do MSAE, conforme descrito nas próximas seções.

---

<sup>10</sup> <http://www.moodle.org>

### 5.3.1 Primeira versão do Ambiente Virtual de Aprendizagem

Na primeira versão do AVA a implementação do MSAE se deu pelo uso do *plug-in Completion Progress*<sup>11</sup>, desenvolvido pela comunidade MOODLE e disponibilizado no repositório de *plug-ins* da plataforma de forma gratuita. Desenvolvido na forma de um bloco, podem ser inseridas diferentes instâncias desse recurso em um curso. A função principal do bloco *Completion Progress* é apresentar para os estudantes uma representação do seu desempenho na forma de retângulos coloridos. Na configuração original do bloco todos recursos aparecem representados como retângulos azuis; quando há sucesso na realização da atividade o retângulo correspondente assume a cor verde e quando há insucesso assume a cor vermelha.

A Figura 15 apresenta o Modelo Aberto de um estudante em relação ao acesso aos materiais de apoio. Neste caso basta que o estudante acesse o material para que seja considerada concluída a atividade. Quando se trata dos exercícios três avaliações são possíveis: resolvido com sucesso (verde); resolvido parcialmente (amarelo); e resolução incorreta (vermelho). Com isso, o retângulo que corresponde a um exercício assumirá a cor verde apenas se o estudante obtiver sucesso na realização do exercício. É importante destacar também que ao posicionar o mouse sobre os retângulos é possível acessar o material ou a atividade em questão.



Figura 15 – Implementação do MSAE no ambiente de aprendizagem.

Fonte: Elaborada pelo autor (2017).

Ao clicar no botão “Visão geral de alunos” o estudante podia visualizar o seu modelo junto ao dos seus colegas. Isto se tornou possível pela flexibilidade em configurar permissões na plataforma MOODLE, já que esta função originalmente está disponível apenas para os professores. Conforme pode ser visto na Figura 16 o estudante poderia comparar o seu

<sup>11</sup> [https://moodle.org/plugins/block\\_completion\\_progress](https://moodle.org/plugins/block_completion_progress)

desempenho com o dos colegas e também acessar os recursos clicando diretamente nos retângulos.

Visão geral de alunos			
Papel: Estudante			
	Nome / Sobrenome	Último no curso	Progresso de Conclusão
<input type="checkbox"/>	Aluno Um	sábado, 17 Jun 2017, 16:48	<div style="width: 100%;"><div style="width: 25%; background-color: #76b82a;"></div><div style="width: 25%; background-color: #3498db;"></div><div style="width: 25%; background-color: #76b82a;"></div><div style="width: 25%; background-color: #e74c3c;"></div></div>
<input type="checkbox"/>	Aluno Trs	sábado, 17 Jun 2017, 16:47	<div style="width: 100%;"><div style="width: 25%; background-color: #76b82a;"></div><div style="width: 25%; background-color: #3498db;"></div><div style="width: 25%; background-color: #3498db;"></div><div style="width: 25%; background-color: #3498db;"></div></div>
<input type="checkbox"/>	Aluno Quatro	sábado, 17 Jun 2017, 16:48	<div style="width: 100%;"><div style="width: 25%; background-color: #3498db;"></div><div style="width: 25%; background-color: #e74c3c;"></div><div style="width: 25%; background-color: #3498db;"></div><div style="width: 25%; background-color: #3498db;"></div></div>
<input type="checkbox"/>	Aluno Dois	sábado, 17 Jun 2017, 16:46	<div style="width: 100%;"><div style="width: 25%; background-color: #e74c3c;"></div><div style="width: 25%; background-color: #76b82a;"></div><div style="width: 25%; background-color: #3498db;"></div><div style="width: 25%; background-color: #3498db;"></div></div>

Figura 16 – Visualização do modelo do estudante e dos seus colegas.

Fonte: Elaborada pelo autor (2017).

O *plugin Completion Progress* foi desenvolvido como um bloco, assim, conforme a estrutura de blocos do MOODLE, o recurso fica disponível em segundo plano no curso. Desta forma, para que o estudante conseguisse visualizar o MSAE seria necessário que ele clicasse em “Visão Geral de alunos” (Figura 16). Considerando a importância da visualização do MSAE para a tese, constatou-se que o acesso a ele deveria ser facilitado, colocando-o em primeiro plano dentro do curso. Para contornar essa limitação, buscou-se uma nova forma de incorporar o MSAE no AVA.

### 5.3.2 Versão final do Ambiente Virtual de Aprendizagem

Para que o MSAE ocupasse o primeiro plano dentro do curso no AVA optou-se então pela utilização do *plugin eTask format course*<sup>12</sup>. Este *plugin* diferencia-se do anterior por ser desenvolvido como um formato de curso e não um bloco. Devido a isso, a partir de sua instalação e configuração, o curso todo no AVA adota o formato de um MSAE (Figura 17). Com o novo *plugin*, ao acessar o curso no AVA, o MSAE era a primeira tela que os alunos visualizavam. Com isso, o objetivo de tornar o AVA centrado no MSAE tornou-se possível.

<sup>12</sup> [https://moodle.org/plugins/format\\_etask](https://moodle.org/plugins/format_etask)

	A7	Q50	Q49	Q48	Q47	Q46	Q45	Q44	Q43	Q42	Q41
	-	-	3	2	2	3	3	3	3	3	2
	-	3	3	3	3	3	3	3	3	3	3
	-	3	3	3	3	3	3	3	3	3	3
	-	3	3	3	3	3	-	-	-	-	3
	-	-	-	-	-	-	-	-	-	-	-
	-	3	-	2	2	3	2	3	3	2	3
	-	-	-	-	-	-	-	-	-	-	-

Figura 17 – Visualização dos modelos dos estudantes na versão final do AVA.

Fonte: Elaborada pelo autor (2018).

Ao acessar o curso os alunos se deparavam com o MSAE indicando em verde se haviam resolvido o exercício corretamente, ou em vermelho se haviam resolvido parcial ou incorretamente. Conforme pode ser visualizado na Figura 17, o valor interno dos blocos possui diferente representação para os estudantes. O valor 3 indica a resolução correta, o valor 2 a resolução parcialmente correta e o valor 1 a resolução incorreta. Ainda, aquelas atividades que não foram realizadas aparecem para o usuário com um bloco sem valor e sem cor.

Além da representação do conhecimento dos estudantes, o MSAE também pode ser utilizado para acessar os exercícios e materiais. Ao clicar no identificador da coluna há um link que redireciona para o exercício em questão. Assim, é possível que o estudante utilize tanto o MSAE para navegar no curso quanto os links tradicionalmente dispostos no MOODLE.

#### 5.4 RESULTADOS PRELIMINARES

Artigos oriundos da pesquisa desenvolvida nesta primeira versão do ambiente virtual de aprendizagem foram publicados em anais de eventos e em um periódico. Todos os artigos tratam sobre a aprendizagem de programação e apresentam possibilidades de suporte à esse processo.

Em Ferreira e Reategui (2016a) é apresentado o resultado da construção de uma Trajetória de Aprendizagem Conceitual para a aprendizagem de programação. Neste artigo é proposta uma sequência e são definidos pré-requisitos entre os conceitos aprendidos nas disciplinas de programação de modo que os estudantes teriam possibilidade de optarem pela

melhor estratégia de estudo. Este estudo envolvia também a auto-avaliação da aprendizagem, assim o estudante poderia avaliar quando e qual conteúdo deveria ser estudado.

Em Ferreira e Reategui (2017a) são apresentados os resultados de um experimento utilizando a ferramenta de mineração de textos Sobek para identificação dos principais termos de enunciados de programação e a sua apresentação na forma de um grafo. Considerando que a representação visual das informações contribui para a sua compreensão, o experimento tinha como objetivo verificar a viabilidade de utilização do Sobek para auxiliar os estudantes a compreenderem os enunciados dos problemas de programação, e conseqüentemente auxiliá-los à obter melhores resultados na resolução. Ao final do experimento verificou-se que devido a maioria dos enunciados de exercícios de programação apresentarem poucas informações a contribuição do uso do Sobek se torna limitada.

Em Ferreira e Reategui (2016b) e Ferreira e Reategui (2017b) é apresentada uma proposta de Comunidade Virtual de Prática para aprendizagem de programação que incorpora o MSAE. A comunidade proposta disponibilizaria para seus membros recursos para comunicação síncrona e assíncrona, de forma que o conhecimento seria construído de forma coletiva. Destacava-se na comunidade o *scaffolding* entre os membros e a possibilidade de identificação dos membros mais experientes por meio do MSAE. A partir da proposta apresentada nestes dois artigos chegou-se ao desenho do estudo final desenvolvido nesta tese.

## 6 RESULTADOS

Neste Capítulo são apresentados e discutidos os resultados obtidos a partir do *quasi*-experimento realizado. A seção 6.1 descreve o grupo de estudantes que participou da pesquisa; a seção 6.2 apresenta os resultados referentes à consciência meta-cognitiva dos estudantes; na seção 6.3, são discutidos os resultados referentes ao desempenho dos alunos; e na seção 6.4 são apresentados os resultados referentes ao acesso ao AVA e ao MSAE.

### 6.1 AMOSTRA FINAL

Conforme mencionado no item 5.1, matricularam-se na disciplina de Introdução à Programação 33 estudantes na Turma A e 17 estudantes na Turma B. Porém, foram considerados apenas os dados dos estudantes que:

- não reprovaram por falta;
- responderam duas vezes ao questionário MAI;
- realizaram o pré-teste e o pós-teste e;
- concordaram em participar do experimento assinando o Termo de Consentimento Livre e Esclarecido (ANEXO D).

Assim, foram considerados os dados coletados para 26 estudantes da Turma A e 14 estudantes da Turma B, totalizando 40 participantes.

Como a abordagem metodológica não teve por finalidade comparar os resultados entre a Turma A e a Turma B, considerando que ambas passaram pela mesma intervenção em todos os aspectos, a análise dos resultados irá considerar apenas um grupo, denominado Grupo Geral, reunindo todos os 40 estudantes das duas turmas.

Do Grupo Geral foram selecionados sete estudantes a partir do desempenho final obtido nos cursos. Desta forma, foram selecionados dois estudantes que obtiveram desempenho superior a 75% das turmas (alto), dois estudantes que obtiveram desempenho superior a 25% da turma (intermediário) e três estudantes que obtiveram desempenho abaixo de 75% da turma (baixo) (Quadro 9).

Quadro 8. Estudantes entrevistados ao final do quasi-experimento.

<b>Identificador</b>	<b>Turma</b>	<b>Nível de desempenho</b>	<b>Sexo</b>
Estudante A	B	Alto	Feminino
Estudante B	A	Alto	Masculino
Estudante C	A	Intermediário	Masculino
Estudante D	B	Intermediário	Feminino
Estudante E	A	Baixo	Masculino
Estudante F	A	Baixo	Masculino
Estudante G	B	Baixo	Masculino

Fonte: Elaborado pelo autor (2018).

Os estudantes selecionados participaram de uma entrevista semi-estruturada após o final do semestre. A entrevista (ANEXO B) consistia de perguntas relacionadas à percepção dos estudantes quanto as estratégias de estudo e ao próprio desempenho no curso, à comparação com o desempenho dos colegas e sobre a apresentação visual do MSAE no AVA.

## 6.2 A CONSCIÊNCIA META-COGNITIVA

Ao iniciar e finalizar o curso os estudantes responderam a uma versão traduzida e validada do questionário MAI com o objetivo de coletar dados sobre a sua consciência meta-cognitiva e verificar alterações deste perfil ao final do semestre após o uso do MSAE. A organização do questionário MAI permite avaliar os resultados de forma geral, a partir da média de todos os itens respondidos; por componente, a partir da média das subescalas que compõem cada componente; e por subescala.

Na Tabela 1 são apresentados os resultados referentes ao valor geral obtido pelos estudantes ao responderem o questionário. Após análise com o teste *t de Student* verificou-se que não ocorreu variação significativa entre o pré-teste e o pós-teste. Este resultado demonstra que a percepção geral que os estudantes tinham sobre suas habilidades meta-cognitivas não se alterou após a intervenção proposta. Neste sentido, o fato de não ser foco desta tese prover alguma forma de treino meta-cognitivo pode justificar este resultado, uma vez que os estudantes construíram de forma autônoma as estratégias de uso do MSAE.

Tabela 1. Avaliação geral do MAI.

	<b>Pré</b>	<b>Mediana</b>	<b>D. P.</b>	<b>Pós</b>	<b>Mediana</b>	<b>D.P.</b>
<b>Geral</b>	7,55	7,60	1,11	7,45	7,58	0,91

Fonte: Elaborada pelo autor (2018).

As respostas ao questionário também foram analisadas agrupando as subescalas por componentes. Na Tabela 2 são apresentados os resultados obtidos para o componente *conhecimento cognitivo*, que se refere à consciência sobre o próprio conhecimento e ao saber quando e como utilizar estratégias corretas para resolver um problema. Ao analisar as respostas para o componente *conhecimento cognitivo* com o teste de *t de Student* não se encontrou diferença significativa entre a primeira e a segunda aplicação do questionário.

Tabela 2. Resultados para *conhecimento cognitivo*.

	<b>Início</b>	<b>Mediana</b>	<b>D. P.</b>	<b>Final</b>	<b>Mediana</b>	<b>D.P.</b>
<b>Geral</b>	7,39	7,63	1,16	7,20	7,31	1,01

Fonte: Elaborada pelo autor (2018).

Na Tabela 3 são apresentadas as respostas para o componente *regulação da cognição*, que está relacionado às estratégias específicas que são empregadas na resolução de problemas. Da mesma forma que as respostas ao componente *conhecimento cognitivo*, as respostas para o componente *regulação da cognição* foram analisadas com o teste *t de Student*. E, assim como para o outro componente, não se verificou diferença significativa entre o pré-teste.

Tabela 3. Resultados para *regulação da cognição*.

	<b>Início</b>	<b>Mediana</b>	<b>D. P.</b>	<b>Final</b>	<b>Mediana</b>	<b>D.P.</b>
<b>Geral</b>	7,64	7,72	1,15	7,61	7,69	0,96

Fonte: Elaborada pelo autor (2018).

Conforme já mencionado no início da seção, a análise do questionário MAI pode ser realizada com maior nível de detalhe, ou seja, a partir das subescalas de cada componente. Assim, a Figura 18 apresenta um gráfico construído a partir da média aritmética dos valores obtidos no pré-teste e no pós-teste para todos os estudantes.

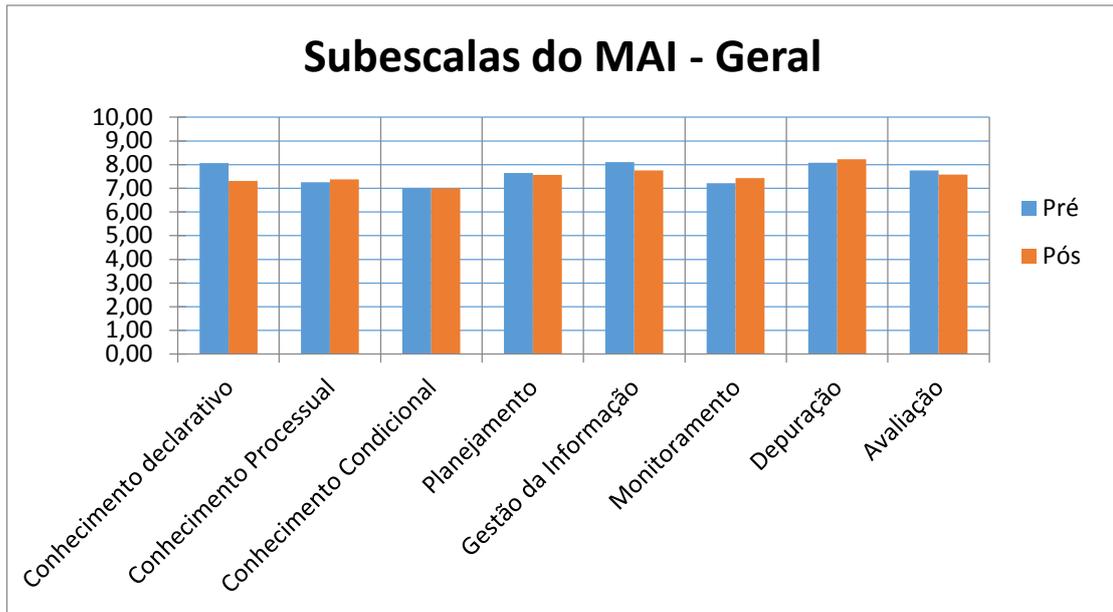


Figura 18. Resultado geral para subescalas do MAI.

Fonte: Elaborada pelo autor (2018).

Conforme ilustrado na Figura 18, verifica-se que os resultados do pré-teste e do pós-teste são similares, exceto pela subescala Conhecimento Declarativo, que é menor no pós-teste. Ao realizar análise estatística com o teste *t de Student* se verificou diferença significativa ( $p = 0,0006$ ) para ela após a intervenção proposta. Assim, pode-se concluir que a intervenção contribuiu para que os estudantes questionassem seu conhecimento sobre si, suas capacidades e suas estratégias, e o avaliassem de forma diferente após a intervenção proposta.

Por outro lado, não se verificou alteração significativa em relação as outras subescalas. Conforme já mencionado, considera-se que a ausência de um treino meta-cognitivo durante a intervenção pode ter contribuído para isso. Todavia, conforme apresentado na sequência deste Capítulo, o MSAE foi utilizado como suporte aos processos meta-cognitivos durante a aprendizagem de programação, ainda que alterações significativas sobre as demais subescalas não tenham sido encontradas.

Neste sentido, os estudantes entrevistados foram questionados sobre como estudavam para a disciplina de programação e a resposta de todos foi que estudavam resolvendo os exercícios propostos através do AVA. Assim, identifica-se que a resolução de exercícios é a estratégia normalmente adotada pelos estudantes para aprender programação, independente do grau de sua consciência meta-cognitiva.

*[...] no primeiro momento eu olhava os slides e tentava fazer os exemplos das aulas sozinha, sem ajuda. Aí, se eu tivesse alguma dificuldade eu olhava como o professor*

*tinha feito. Aí, eu fazia umas duas vezes cada exercício (Estudante A, sexo feminino, 22 anos).*

*Eu buscava rever os conteúdos passados em aula e buscava realizar as atividades propostas (Estudante E, sexo masculino, 24 anos).*

Com o objetivo de aprofundar a análise sobre os resultados para as subescalas os estudantes foram questionados a respeito de como se organizavam para estudar para a disciplina de programação. A partir das respostas ficou evidente que existia diferença com relação ao planejamento entre eles.

*[...] eu distribuo horários. Vou estudar tanto tempo pra essa disciplina e tanto pra essa. Na que eu tenho mais dificuldade eu estudo mais umas horinhas pra aquela (Estudante D, sexo feminino, 23 anos).*

*[...] pra mim o que fez mais diferença foi fora de sala de aula do que dentro da sala. O final de semana eu uso pra descansar. Se eu organizasse melhor meu tempo dava, mas eu fico entre programar e assistir tv, daí é difícil... (Estudante E, sexo masculino, 24 anos).*

Ainda que não tenha sido verificada grande diferença na consciência meta-cognitiva após a intervenção com o MSAE, ao serem questionados como decidiam se já haviam estudado suficiente programação, os estudantes com desempenho alto ou intermediário demonstraram utilizar processos meta-cognitivos de Depuração e Avaliação favorecidos pelo uso do MSAE, diferente do estudante de desempenho baixo. Neste sentido, verificou-se de forma clara o distinto uso destes processos por estudantes que apresentaram desempenhos diferentes ao final da disciplina.

*[...] eu fazia duas vezes todos os exercícios disponíveis [...] quando eu conseguia resolver de primeira eu dizia que já tinha entendido. Mas tinha aqueles que eu tinha dificuldade. Aí eu tentava fazer mais vezes. Como os de laços de repetição, eu tinha dificuldade em um, aí eu tentava fazer mais vezes. Eu não fazia eles na mesma ordem sempre, pra não decorar. [...] Então quando eu tentava resolver e conseguia, eu considerava que tinha entendido (Estudante A, sexo feminino, 22 anos).*

*[...] eu parava de estudar quando eu dominava o conteúdo [...] quando eu conseguia ler o enunciado e resolver o problema sem ter que escrever código (Estudante C, sexo masculino, 26 anos).*

*[...] eu nunca sabia se eu tinha estudado o suficiente, só sabia depois da prova (Estudante F, sexo masculino, 22 anos).*

É importante relatar que durante o semestre aconteceram situações em que estudantes enviaram e-mail para o professor perguntando se poderiam submeter novamente a solução de um exercício para o qual haviam conseguido elaborar uma solução correta. Neste sentido, conforme demonstra a resposta do estudante G, a informação apresentada pelo MSAE representa o estudo, a dedicação ou o conhecimento sobre aquele determinado assunto.

*[...] tinha aquelas provas que eu via que eu estudei mais no quadro e eu ia melhor. Era visual o meu estudo (Estudante G, sexo masculino, 21 anos).*

Com isso, por mais que existam trabalhos que argumentem que o acesso ao MSAE promova o desenvolvimento de aspectos meta-cognitivos (BULL; KAY, 2007), não há relatos de avaliações que comprovem esta evolução com instrumentos como o MAI. Assim, a partir da análise estatística proposta nesta tese, pode-se afirmar que o acesso ao MSAE no estudo proposto relativo à disciplina de Introdução à Programação não promoveu alterações significativas nas habilidades meta-cognitivas dos estudantes participantes.

Por outro lado, verificou-se que a possibilidade de comparar o seu desempenho com o dos colegas e da turma por meio do MSAE foi o meio pelo qual os estudantes que apresentaram desempenho alto e intermediário monitoraram, avaliaram e decidiram quais estratégias utilizar para alcançar seus objetivos na disciplina. Isso se tornou evidente quando os estudantes responderam na entrevista a respeito da possibilidade de comparar-se com os colegas:

*[...] comparar meio que me fez sentir uma invejinha dos colegas. Tu ver os caras lá com tudo verde lá e o cara não conseguir atingir o objetivo...ter o mesmo nível de progresso que o outro. [...] essa inveja serviu, pra mim, como uma motivação para tentar também deixar tudo lá com verde (Estudante E, sexo masculino, 24 anos).*

*[...] eu achei muito legal isso de ver o desempenho de todo mundo [...] eu conseguia ver tipo se eu tinha que começar a prestar mais atenção...me puxar mais, to seguindo um caminho errado (Estudante B, sexo masculino, 25 anos).*

Com isso, pode-se concluir que por mais que as respostas dadas ao questionário MAI não demonstrem estatisticamente que houve variação na consciência meta-cognitiva dos estudantes ao interagirem com o MSAE, verifica-se que os estudantes utilizaram seus recursos para sustentar seus processos meta-cognitivos. Desta forma, torna-se relevante avaliar e analisar de que forma o uso do MSAE e a consciência meta-cognitiva relacionam-se ao desempenho dos estudantes durante a aprendizagem de programação.

### 6.3 O DESEMPENHO

Com o objetivo de analisar a relação do desempenho e da consciência meta-cognitiva dos estudantes na disciplina de programação ao iniciar e finalizar o *quasi*-experimento foi proposto aos estudantes que resolvessem um teste (ANEXO C) que compreendia questões referentes a todos os conteúdos estudados durante o semestre. Na Tabela 4 são apresentados os resultados alcançados.

Tabela 4. Resultados do pré-teste e do pós-teste.

	<b>Pré-teste</b>	<b>Mediana</b>	<b>D. P.</b>	<b>Pós-teste</b>	<b>Mediana</b>	<b>D.P.</b>
<b>Geral</b>	1,22	0	2,49	7,20	7,60	2,23

Fonte: Elaborada pelo autor (2018).

Os resultados no pré-teste indicam que os estudantes possuíam pouco ou nenhum conhecimento sobre os temas tratados na disciplina antes de cursarem, com a nota média se aproximando de 1 em um total de 10. Neste sentido, verificou-se diferença significativa entre o resultado do pré-teste e o resultado do pós-teste com o teste *t de Student*.

Além do pré-teste e do pós-teste os estudantes resolveram exercícios dispostos no MSAE quantificados e classificados conforme descrito no Capítulo 5. Ao resolver cada exercício os estudantes também deveriam indicar sua percepção sobre a resolução, respondendo se acreditavam ter resolvido o exercício corretamente, parcialmente correto ou incorretamente. Esta informação era então utilizada para calcular os índices KMA e KMB, responsáveis por aferir indícios sobre a percepção dos estudantes acerca de suas capacidades. Além disso, ao final de cada tópico os estudantes também resolveram uma avaliação específica sobre o tema tratado naquele tópico.

Na Tabela 5 são apresentados os resultados referentes à resolução do conjunto de exercícios como um todo, tanto centrais quanto complementares, das avaliações e o desempenho final na disciplina. Além disso, também é apresentada a média aritmética de questões resolvidas por estudante e o índice KMA e KMB.

Tabela 5. Resultados obtidos nos exercícios disponíveis no AVA.

		<b>Média</b>	<b>Mediana</b>	<b>Desvio P.</b>
<b>Geral</b>	<b>Complementares</b>	6,06	6,50	2,99
	<b>Centrais</b>	8,52	8,80	1,27
	<b>Quantidade</b>	7,91	8,20	1,78
	<b>KMA</b>	0,69	0,73	0,29
	<b>KMB</b>	0,02	0,05	0,15
	<b>Avaliações</b>	7,54	8,04	2,03
	<b>Geral</b>	79,55	81,50	13,62

Fonte: Elaborada pelo autor (2018).

Ao analisar a correlação entre os dados verificou-se que há correlação entre o desempenho nos exercícios complementares e o desempenho geral ( $p = 0,60$ ) e entre a quantidade de exercícios realizados e o desempenho geral ( $p = 0,51$ ). Assim, percebe-se que aqueles estudantes que não se limitaram a resolver apenas os exercícios centrais obtiveram melhores resultados.

Os exercícios estavam disponíveis no MSAE da mesma forma, sem qualquer diferenciação visual quanto a sua categorização. É importante relatar que foi observado que os estudantes se preocupavam também com o seu desempenho nos exercícios complementares, pois enviavam e-mails para o professor com novas soluções para os exercícios complementares submetidos inicialmente de forma incorreta.

Observou-se também que os estudantes em geral obtiveram KMA alto e KMB aleatório. Isto significa que os estudantes demonstraram avaliar corretamente suas capacidades e se avaliaram com um viés aleatório, ora otimista e ora pessimista, não se observando um padrão estável.

É importante destacar que a partir da observação do professor os estudantes buscaram resolver e se preocuparam com a resolução de exercícios, sendo este um resultado do uso do MSAE percebido pelo professor. Assim, a correlação entre a quantidade de exercícios e o desempenho neles com o desempenho final na disciplina demonstra que quanto mais se favorece a resolução de exercícios melhor será o desempenho final da turma.

Ao focar a análise nos índices KMA (a capacidade de avaliar o próprio conhecimento) e KMB (o viés empregado na avaliação do próprio conhecimento) verificou-se que no geral há correlação moderada entre o valor do índice KMA e da nota obtida nas avaliações ( $p = 0,62$ ), entre o KMA e quantidade de exercícios resolvidos ( $p = 0,43$ ), entre o KMA e o desempenho

nos exercícios complementares ( $p = 0,51$ ) e entre o KMA e o desempenho geral ( $p = 0,63$ ). Também foi possível verificar correlação entre o índice KMB e o desempenho nos exercícios centrais ( $p = -0,43$ ).

Isso corrobora o fato de que os estudantes que conseguem avaliar melhor suas capacidades tendem a obter melhores resultados. É possível também concluir que estudantes com melhores capacidades de avaliar sua aprendizagem tendem a fazer mais exercícios e, conforme já demonstrado anteriormente, conseqüentemente tendem a obter maior desempenho.

A correlação entre o desempenho geral e nos exercícios com os resultados obtidos no pré-teste e no pós-teste também foi analisada. Ao realizar a análise com o teste t de *Student* foi verificada correlação forte entre o resultado do pós-teste e o desempenho geral ( $p = 0,83$ ) e entre o pós-teste e o KMA ( $p = 0,58$ ).

Considerando que os exercícios e os testes realizados foram classificados por tópicos, torna-se relevante observar o desempenho dos estudantes em cada tópico. A Figura 19 apresenta o desempenho por tópico a partir da média aritmética das notas obtidas nas avaliações.

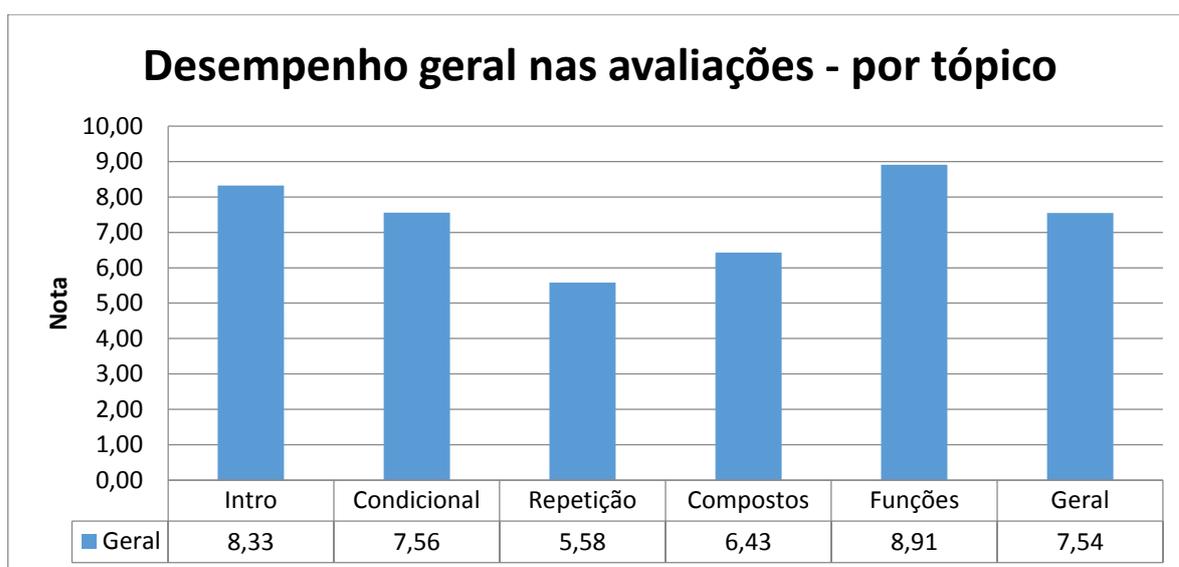


Figura 19. Desempenho dos grupos nas avaliações por tópico.

Fonte: Elaborada pelo autor (2018).

É possível observar que o tópico com menor média geral é o de repetição, que trata sobre estruturas de controle que possibilitam construir laços de repetição nos códigos. Em sequência, o tópico sobre tipos compostos aparece com a menor média. Pode-se relacionar os dois tópicos tendo em vista que o primeiro é necessário para a utilização do segundo. Assim, observa-se que a deficiência na aprendizagem do tópico sobre estruturas de repetição incide no desempenho sobre tipos compostos.

Na Figura 20 é apresentada uma comparação do desempenho dos estudantes nos exercícios centrais por tópico. Conforme se percebe pelo gráfico, o tópico de tipos compostos é aquele no qual os estudantes apresentaram pior desempenho, seguido pelo tópico sobre estruturas condicionais. Conforme análise realizada sobre o desempenho nas avaliações, o tópico sobre tipos compostos seguiu sendo um no qual os estudantes apresentaram pior desempenho. Porém, no caso dos exercícios centrais, os estudantes apresentaram desempenho inferior sobre estruturas condicionais. Isto pode estar relacionado com o fato de que é a primeira estrutura complexa com a qual se defrontam, pois no tópico de introdução apresentaram o melhor desempenho.

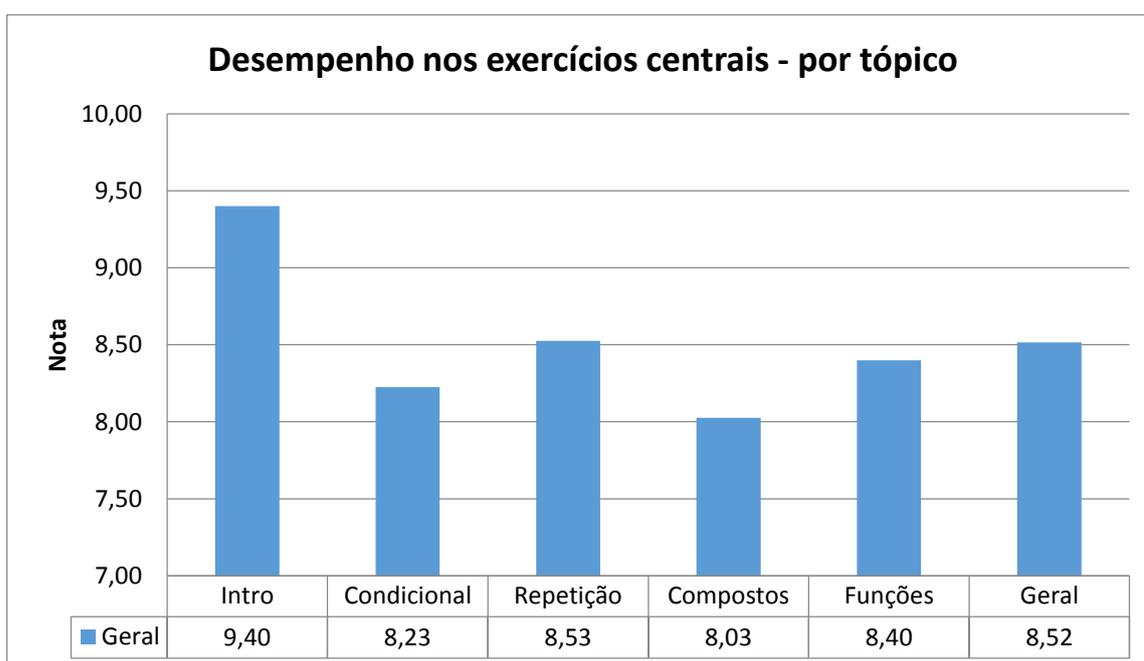


Figura 20. Desempenho dos grupos nos exercícios centrais por tópico.

Fonte: Elaborada pelo autor (2018).

Na Figura 21 é apresentada comparação entre os resultados obtidos pelos estudantes nos exercícios complementares. Observa-se que as médias menores são as relacionadas aos tópicos de estruturas condicionais, repetição e tipos compostos, o que se refletiu no desempenho nas avaliações.

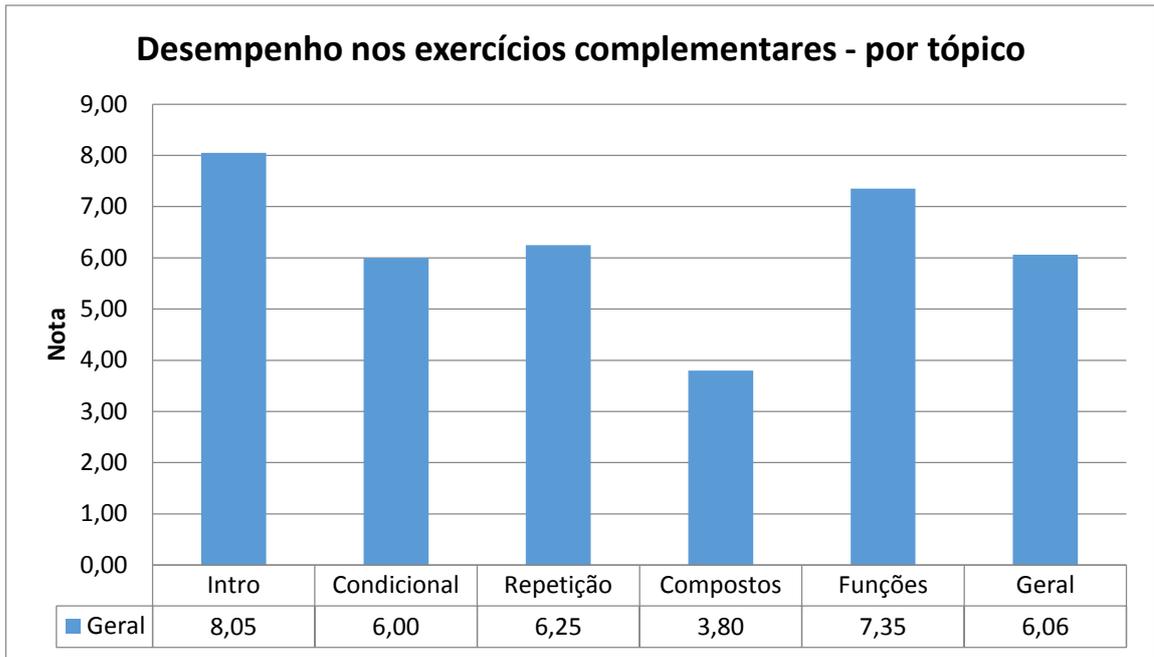


Figura 21. Desempenho dos grupos nos exercícios complementares por tópico.

Fonte: Elaborada pelo autor (2018).

Na Figura 22 é apresentada a comparação entre a média de exercícios realizados por tópicos. Conforme se pode visualizar, na média os estudantes realizaram pelo menos 6 exercícios em um total de 10 por tópico.

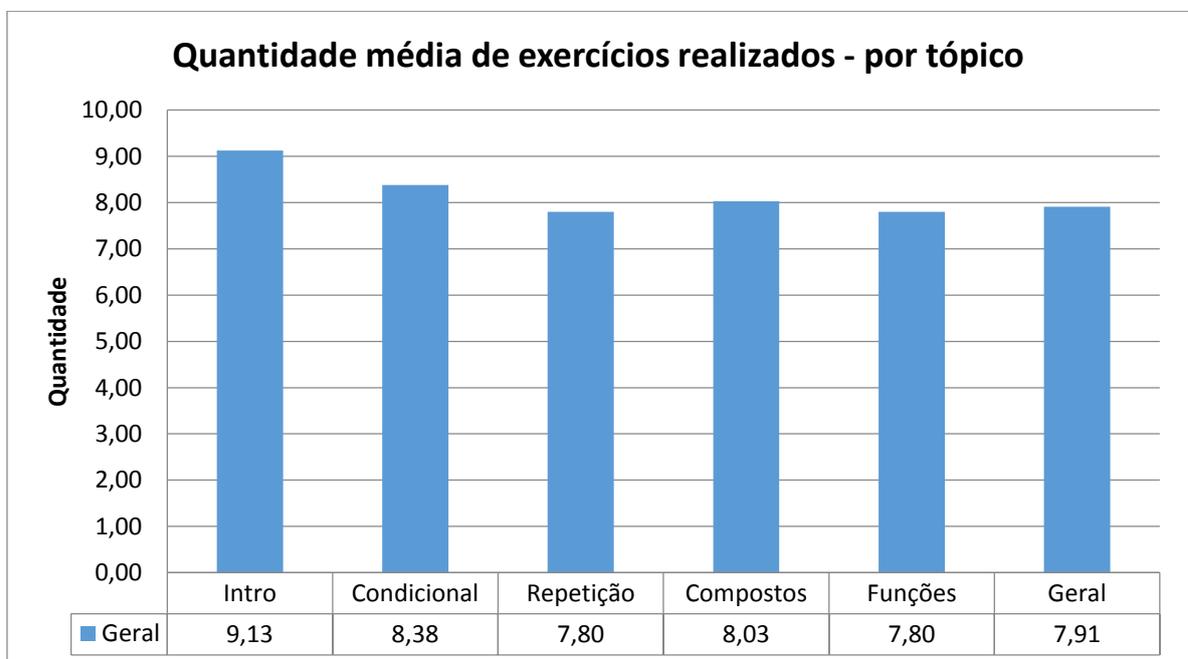


Figura 22. Quantidade de exercícios realizados por tópico.

Fonte: Elaborada pelo autor (2018).

Conforme já mencionado, os estudantes utilizaram o MSAE para acompanhar o desempenho dos colegas e da turma como um todo como forma de avaliar e monitorar as próprias estratégias e desempenho a partir da comparação. Neste sentido, os estudantes entrevistados foram questionados se comparavam seu desempenho com algum colega específico. Conforme as respostas dos alunos, as comparações eram realizadas com alunos que estavam com desempenho pior ao deles e desempenho melhor. As duas comparações tinham motivações diferentes. A comparação com os estudantes de melhor desempenho fazia com que eles buscassem alcançá-los, conforme relatado pelos estudantes A, B, C e D.

*[...] eu acompanhava aqueles colegas que eu sabia que eram bons. Quando tu acerta e aquele que é bom ainda não fez ou não acertou, tu te sente aliviado, dá um sentimento de retorno (Estudante A, sexo feminino, 22 anos).*

*Eu sempre fui de fazer isso, de comparar meu desempenho com os outros. [...] eu sou um pouco competitivo, se eu via um cara acertando tudo eu pensava ah, eu tenho que fazer igual aquele cara (Estudante B, sexo masculino, 25 anos).*

*Quando eu sei que os outros tão acertando, eu me motivo a ir atrás (Estudante D, sexo feminino, 23 anos).*

*Chamava a atenção o colega que gabaritava, por isso eu olhava. Se tal pessoa acertou todas, por que eu não posso? (Estudante C, sexo masculino, 25 anos).*

Por outro lado, conforme relatam os estudantes A, E e G, a comparação com estudantes de desempenho inferior ao deles fazia com que eles sentissem conforto por não ter obtido o resultado desejado. Todavia, como já exposto, também é importante que os estudantes se sintam parte do grupo no processo de aprendizagem, e a comparação com colegas de desempenho inferior teve esse papel.

*Tu tem a nota de toda turma, então tu sabe se tu tá mal e a turma tá bem ou se a turma tá bem e tu tá mal. [...] a gente sempre se compara com o melhor. Aconteceu as vezes eu errar e o colega que ia bem errar também, daí me dava um conforto, tava ok... (Estudante E, sexo masculino, 24 anos).*

*[...] eu vi que os colegas tinham as mesmas dificuldades que eu. Eu meio que senti um comodismo. Se todos erraram a mesma questão, então tá bom... (Estudante G, sexo masculino, 26 anos).*

Assim, ao debruçar-se sobre a relação entre a consciência meta-cognitiva e o desempenho dos estudantes, foi possível identificar que há relação entre a capacidade de avaliação das próprias capacidades, o desempenho nos exercícios e o desempenho na disciplina de programação.

Na seção 6.3 concluiu-se que os estudantes utilizam o MSAE para avaliar e monitorar sua aprendizagem de programação através da possibilidade de comparar seu desempenho com o dos colegas. A partir das observações trazidas nesta seção, foi possível identificar que a comparação que os estudantes realizam tem duas consequências, motivá-los em buscar alcançar resultados melhores quando se comparam com estudantes com melhor desempenho que o deles e confortá-los quanto ao seu desempenho quando se comparam com estudantes de desempenho similar ou inferior ao deles.

#### 6.4 O ACESSO AO AMBIENTE VIRTUAL DE APRENDIZAGEM

Com o objetivo de investigar de que forma se deu o uso do AVA e do MSAE foram coletados dados a partir do *log* do MOODLE. A partir desses dados, enriquecidos pelas entrevistas e pelo diário de campo, foi possível analisar o papel do MSAE integrado no Ambiente Virtual de Aprendizagem no processo de aprendizagem de programação. A partir dos *logs* foi possível analisar a média aritmética de quantos acessos ao curso no AVA os estudantes realizaram. Por acesso ao curso compreende-se a efetivação de um login no sistema. Esta informação é válida porque ao acessar o AVA os alunos deparavam-se imediatamente com o MSAE. Assim, verificou-se que em média os estudantes acessam 223,10 vezes o curso no AVA.

Buscando verificar se existia alguma relação entre os acessos ao curso e aspectos meta-cognitivos e de desempenho realizou-se análise estatística com o cálculo do coeficiente de Pearson, porém, nenhuma correlação foi encontrada. No geral, este resultado difere dos demais trabalhos sobre MSAE nos quais o número de acessos ao modelo estava relacionado ao desempenho.

Com o objetivo de verificar o número de acessos aos recursos disponibilizados no AVA por meio do MSAE novamente foram consultados os logs do Moodle. A partir dessa consulta

verificou-se que em média os estudantes acessaram 762,30 recursos no AVA. Por acesso aos recursos compreende-se aqui o acesso aos exercícios e aos materiais de apoio disponibilizados.

Ao verificar a correlação do número de acessos aos recursos com os dados coletados pelo questionário MAI e os dados referentes aos exercícios e avaliações, não foi encontrada nenhuma correlação. Assim, na intervenção proposta nesta tese não se verificou relação entre o acesso aos recursos pelo MSAE e o desempenho na disciplina.

Explorando questões relacionadas ao acesso ao AVA e ao MSAE os estudantes entrevistados indicaram que as relações entre eles também tinham influência no momento de fazer comparações. Conforme os estudantes C, D e G, a proximidade com os colegas também era um fator que os motivava a fazer comparações.

*Como eu tenho amizade com essa pessoa, eu sabia que ela não trabalhava, então de certa forma ela devia ir melhor que eu. Por isso ela devia gabaritar. Então se essa pessoa gabaritou é porque ela não trabalha, tem mais tempo de dia pra estudar...[...] Eu via lá no site que esse meu amigo tinha feito o exercício já, então eu tinha que fazer também... (Estudante C, sexo masculino, 26 anos).*

*Eu acompanhava mais o desempenho dos do meu grupinho no início porque a gente estuda junto (Estudante D, sexo feminino, 23 anos).*

*Quando eu comecei a conviver com duas pessoas que se puxam mais, eu comecei a me puxar mais. A gente via mesmo não estando na aula quem tinha conseguido acertar (Estudante G, sexo masculino, 21 anos).*

A partir das respostas dos entrevistados A, B e E relacionadas às comparações por meio do MSAE, foi possível identificar que os estudantes utilizavam o MSAE como um recurso de suporte para identificar para quais colegas pedir ajuda ou até mesmo aqueles que estavam enfrentando dificuldades e oferecer auxílio. Neste sentido, pode-se perceber que o MSAE não só auxiliou os estudantes a identificarem colegas experientes ou com dificuldades, mas também criar relações na turma.

*Eu achei bem interessante saber como eu estou e também saber quantas meu colega acertou. Aí tu podia pedir me ajuda aí...ou se ele não acertou podia ajudar ele nas que eu tinha acertado (Estudante A, sexo feminino, 22 anos).*

*Tu tava ali fazendo o exercício e não tava conseguindo. Daí tu via lá que o colega tinha feito, já tinha até a nota. Aí tu podia pedir ajudar pra ele...pra ver como ele fez (Estudante B, sexo masculino, 25 anos).*

*Nas outras disciplinas tu tem que ficar perguntando “tu conseguiu fazer?” pra saber quem tá indo melhor. Ali no moodle tinha como saber certo pra quem perguntar (Estudante D, sexo feminino, 23 anos).*

*Quando tu começa a analisar as notas dos colegas de cara tu já vê quem são os caras que entendem do assunto e quem tá como tu. Então tu sabendo quem são as pessoas melhores na turma obviamente tu já sabe com quem tirar dúvidas.[...] aí eu saber que ele é um cara mais experiente foi um gatilho pra eu interagir mais com ele, criar uma amizade (Estudante E, sexo masculino, 24 anos).*

O MSAE também foi utilizado como um recurso que auxiliou os estudantes a se organizarem na disciplina. Conforme relatam os estudantes A, B, C e D na entrevista quando questionados sobre a interface do AVA com o MSAE, tornou-se mais fácil do que nas outras disciplinas identificar quais conteúdos estavam sendo trabalhos naquele momento e acessar os exercícios que ainda não havia resolvido.

*Achei bastante legal ver como tu tá na disciplina só vendo quantos vermelhinhos e quantos verdinhos tu tem (Estudante A, sexo feminino, 22 anos).*

*A principal diferença do moodle das outras disciplinas é pra organização, pra ver como tá indo nos estudos. [...] eu vi como uma ferramenta de organização o moodle dessa disciplina, dava pra saber a ordem do que tinha pra fazer. Eu acessava os exercícios direto pelos que eu tinha pra fazer, em cima (Estudante B, sexo masculino, 25 anos).*

*Eu acessei bem mais esse moodle que os outros por causa da comparação, pra ver se tal pessoa já fez, ou pra rever os exercícios pras provas, aqueles que eu tinha errado e aqueles que tinha acertado também. [...] eu achei bem intuitiva a navegação, não tinha nada que eu mudaria ali. Eu sempre acessava os exercícios pelo início, pela comparação, embaixo eu acho que ficava mais bagunçado (Estudante C, sexo masculino, 26 anos).*

*Foi útil o moodle, sempre meu nome tava em primeiro, não precisava pesquisar numa lista pra ver os exercícios (Estudante D, sexo masculino, 22 anos).*

Assim, ao analisar o uso do MSAE, foi identificado que os estudantes se comparam com aqueles com os quais tem maior proximidade. Nesse sentido, também foi identificado que o MSAE é utilizado como um suporte para a solicitação de ajuda no processo de aprendizagem de programação. A partir dele os estudantes conseguem identificar estudantes mais experientes, para os quais pedir auxílio, e estudantes com dificuldades, para os quais podem se colocar à disposição para ajudar. Também há relatos de estudantes que criaram laços a partir das comparações realizadas por meio do modelo e passaram a estudar juntos.

A representação visual proporcionada pelo MSAE também foi citada como uma ferramenta de organização, de forma que por meio dela os estudantes puderam encontrar os exercícios que ainda não haviam realizado conforme o andamento do curso, algo diferente do que acontece em AVA sem a integração com o MSAE. Ainda, observou-se e foi relatado nas entrevistas que o MSAE foi utilizado como acesso principal aos exercícios e materiais disponibilizados.

## 7 CONCLUSÃO

A aprendizagem de programação é desafiadora, o que faz com que as disciplinas de introdução à programação tenham um número elevado de desistências e reprovações. Dentre as dificuldades apontadas situam-se aquelas ligadas ao uso dos processos meta-cognitivos no processo de aprendizagem. Neste sentido, trabalhos com os de Bull e Pain (1995), Bull e Nghiem (2002) e Bull e Kay (2010) indicam que a disponibilização de uma representação visual do desempenho dos estudantes de forma compartilhada por meio do MSAE com os colegas pode auxiliar o estudante a utilizar os processos meta-cognitivos durante a aprendizagem.

Neste contexto, esta tese teve como objetivo investigar como o apoio meta-cognitivo por meio do MSAE pode contribuir com o processo de aprendizagem de programação. Para isso, foram definidos como objetivos específicos investigar como o MSAE era utilizado e se existia relação entre os processos meta-cognitivos apoiados e o desempenho.

Partindo disso, foi conduzido um quasi-experimento com duas turmas de Introdução à Programação no ensino superior, totalizando 40 estudantes. Estes estudantes acessaram durante um semestre um AVA no qual foi integrado o MSAE por meio de um *plugin*. Este modelo apresentava para os estudantes a representação do seu desempenho em 50 exercícios categorizados nos 5 tópicos estudados na disciplina. Além dos exercícios, ao final de cada tópico os estudantes realizaram uma avaliação específica sobre o conteúdo tratado. Também foi realizado um pré-teste e um pós-teste que abrangia todos os conteúdos estudados durante o semestre e foi levada em consideração a nota final obtida por cada estudante na disciplina.

Os estudantes também responderam ao questionário MAI no início e no final do curso e, por meio da sua percepção acerca dos acertos e erros nos exercícios, foram calculados os índices KMA e KMB. Além disso, também foram coletadas informações por meio do log do AVA, registrado um diário de campo e realizadas entrevistas com 7 estudantes.

Com isso, verificou-se que os estudantes iniciaram a disciplina sem conhecimento algum sobre os conteúdos tratados, sendo esta uma turma homogênea em se tratando de conhecimentos prévios sobre programação. Desta forma, foi encontrada diferença significativa entre o pré-teste e o pós-teste.

Verificou-se também que há uma relação entre o desempenho nos exercícios e a quantidade de exercícios resolvidos e o desempenho na disciplina. Dessa forma, considerando que a representação e compartilhamento do desempenho tornou importante para os estudantes resolver exercícios, tanto centrais quanto complementares, estabelece-se uma correlação entre o uso do MSAE e o desempenho.

Quando o enfoque foram os aspectos meta-cognitivos, não foi verificada diferença significativa entre as respostas do MAI do início do semestre e as do final, demonstrando dessa forma que não houve alterações significativas provocadas pelo apoio proporcionado pelo MSAE. Assim, ao contrário do que afirmam Bull e Kay (2007), nesse estudo observou-se que o uso do MSAE não provocou o desenvolvimento da meta-cognição, ainda que tenha sido verificado que ele favorece o seu uso.

Ainda sobre os aspectos meta-cognitivos verificou-se que os estudantes demonstraram em sua maioria alta capacidade em estimar seus conhecimentos (KMA) e um viés aleatório nesta avaliação (KMB) desde o início do quasi-experimento, não sendo otimistas nem pessimistas. Porém, verificou-se que os estudantes que conseguiam avaliar melhor suas capacidades fizeram mais exercícios, obtiveram melhor desempenho nos exercícios e também na disciplina em geral.

Desta forma, considerando o objetivo de investigar a relação entre o MSAE, o desenvolvimento da meta-cognição e o desempenho na aprendizagem de programação, foi possível observar com a pesquisa que o apoio do MSAE não favoreceu o desenvolvimento da meta-cognição. Com exceção da capacidade de avaliar seu conhecimento, não houve relação observável entre os processos meta-cognitivos e o desempenho na aprendizagem de programação introdutória. Porém, como já mencionado, ao favorecer o monitoramento e a avaliação da aprendizagem, os estudantes foram estimulados a resolver exercícios e, a partir da resolução de mais exercícios, obtiveram melhor desempenho.

Esta tese também contribuiu identificando como o MSAE é utilizado durante o processo de aprendizagem de programação. Foram identificadas três formas de utilização: (i) para avaliação e monitoramento; (ii) para pedir ajuda; e (iii) para organização. Neste sentido, todas as três formas estão relacionadas aos processos meta-cognitivos.

Através do aspecto social do MSAE, os estudantes defrontavam-se sempre que acessavam o curso no AVA com uma representação do seu desempenho e do dos colegas. A partir dessa situação, os estudantes realizavam comparações que serviam como forma de monitorar e avaliar o seu percurso, conforme a Teoria da Comparação Social (FESTINGER, 1954). Foi identificado que os estudantes se comparavam em sua maioria com colegas com os quais tinham maior proximidade. Além disso, também foi identificado que as comparações tinham dois alvos, os estudantes com desempenho maior, que resultava em motivação para estudar mais, e os estudantes com desempenho inferior, que resultava em conforto diante do fracasso.

Além de utilizarem o MSAE, e conseqüentemente a comparação, como forma de se avaliar e monitorar, também utilizaram o modelo como um recurso para solicitar ou fornecer ajuda. Com outro olhar sobre a representação do desempenho, os estudantes conseguiram identificar quem eram os estudantes mais experientes e quem eram aqueles com mais dificuldades. A partir disso solicitavam ou forneciam ajuda para os colegas. É importante destacar que essa ação já acontece na sala de aula tradicional. Todavia, o MSAE traz a facilidade de o estudante já saber exatamente a quem se dirigir. Além disso, também é importante relatar que este uso do MSAE pode fomentar a construção de relações entre os estudantes, sendo este apontado como um fator positivo para o bom desempenho acadêmico.

A interface do MSAE também foi apontada como facilitadora da organização durante o curso. Ao acessar o curso imediatamente o estudante visualizava seu nome em primeiro lugar no MSAE, o que facilitava a visualização do seu desempenho. Para além do desempenho, também foi identificado que a grande maioria dos estudantes utilizaram os links criados no MSAE para acessar os exercícios e materiais. Esta opção se deu pelo fato de as cores representarem seu acerto, erro ou não resolução, facilitando a identificação de qual deveria ser o foco no momento do acesso. Também foi citado que se tornou mais fácil encontrar os materiais e identificar a seqüência do conteúdo.

Desta forma, com o objetivo de formalizar as contribuições desta tese, retomam-se aqui as causas relacionadas à meta-cognição apontadas como dificuldades na aprendizagem de programação, elencadas na seção 2.1, e são descritas as formas como o MSAE pode contribuir para atenuá-las (Quadro 9).

Quadro 9. Auxílio fornecido pelo MSAE às dificuldades de aprendizagem.

<b>Problemas de natureza meta-cognitiva</b>	
<b>Causa</b>	<b>Auxílio por meio do MSAE</b>
Alunos com dificuldades para selecionar recursos adequados ao seu nível de conhecimento	O sequenciamento proporcionado pela interface do MSAE favoreceu a identificação dos conteúdos que deveriam ser estudados no momento.
Alunos que avaliam de forma imprecisa o seu conhecimento sobre os conteúdos.	Visualizar uma representação do seu desempenho, com cores que claramente estavam relacionadas ao resultado obtido, fazia com que os estudantes percebessem se haviam alcançado seus objetivos.
Alunos que não conseguem selecionar a estratégia de estudo mais eficiente.	Ao favorecer a identificação de estudantes mais experientes, os estudantes podem trocar experiências. Além disso, ao comparar seu desempenho com a turma, o estudante reavalia sua dedicação e suas estratégias.
Alunos com dificuldade para refletir sobre os erros identificados na solução de um problema.	Ao comparar o resultado da sua submissão com a dos colegas, o estudante se sente estimulado a buscar interpretar os erros informados pelo feedback de forma a alcançar o resultado desejado.
Alunos que não conseguem avaliar a solução elaborada para um problema.	O fato de o estudante ter consciência de que o resultado da sua submissão será compartilhado com os colegas faz com que ele avalie mais vezes o código antes de submetê-lo. Assim, fomenta-se que ele faça uma avaliação mais rigorosa antes de se colocar diante de avaliação.

Fonte: Elaborado pelo autor (2018).

A partir da pesquisa foi possível concluir que o suporte meta-cognitivo proporcionado pelo MSAE contribui para o processo de aprendizagem de programação introdutória. Porém, a contribuição não se relaciona de forma direta ao desempenho e ao desenvolvimento da meta-cognição. A grande contribuição do MSAE para a aprendizagem de programação está em facilitar o uso de habilidades meta-cognitivas, como a organização, o monitoramento e a avaliação. E, dado que não houve alterações demonstradas de forma empírica na consciência meta-cognitiva dos estudantes, pode-se concluir também que quem usará melhor estes recursos são aqueles estudantes que já possuem estes processos meta-cognitivos melhor desenvolvidos e sabem como empregá-los no processo de aprendizagem.

Com isso, podem ser elencadas duas frentes como possibilidade de trabalhos futuros. A primeira delas está relacionada a desenvolver representações visuais para o MSAE que estimulem o desenvolvimento de processos meta-cognitivos em estudantes de baixo desempenho e que apresentem dificuldades em utilizar os processos meta-cognitivos no processo de aprendizagem. A segunda frente está relacionada a compreender como enriquecer

o MSAE de forma que ele possa auxiliar de forma automatizada os processos utilizados pelos estudantes descritos nesta tese (monitoramento, avaliação e organização).

## 7.1 LIMITAÇÕES DO ESTUDO

Dentre as limitações para este estudo cita-se que o mesmo poderia ser replicado com turmas maiores, dado que esta tese desenvolveu um quasi-experimento com 40 estudantes oriundos de duas turmas diferentes. Neste sentido, também se cita como uma limitação o fato de que 12 estudantes do total de 40 eram oriundos de um curso que não é estritamente ligado à área de Tecnologia da Informação.

Além disso, também pode ser citado como limitação deste estudo a forma como os estudantes tinham possibilidade de resolver os exercícios propostos. Dado que apenas os exercícios centrais dispunham de tempo em aula para serem resolvidos, os exercícios complementares eram resolvidos fora do período de aula. Com isso, não se pode garantir que a solução elaborada pelos estudantes era elaborada de forma autônoma, conforme relatos nas entrevistas e registros no diário de campo. Ao mesmo tempo, alguns estudantes relataram que em alguns momentos tornou-se mais importante fazer seu desempenho aparecer em verde para os demais do que realmente aprender o conteúdo.

Acrescenta-se também que o estudante F, de baixo desempenho, apresentou um perfil completamente diferente da turma. O estudante mostrou-se completamente alheio ao MSAE, afirmando que jamais se comparou com os colegas e que isso não exercia qualquer influência sobre sua avaliação de desempenho. Porém, além da entrevista, não há indícios estatísticos de que isso afetou seu desempenho.

## REFERÊNCIAS

ANDRADE, D.; CARVALHO, T.; SILVEIRA, J.; CAVALHEIRO, S.; FOSS, L.; FLEISCHMANN, A. M.; AGUIAR, M.; REISER, R. Proposta de atividades para o desenvolvimento do Pensamento Computacional no Ensino Fundamental. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 24, 2013, Campinas. **Anais...** Campinas: SBC, 2013.

AURELIANO, V. C. O.; TEDESCO, P. C. A. R. Ensino-aprendizagem de programação para iniciantes: uma revisão sistemática da literatura focada no SBIE e WIE. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 23, 2012, Rio de Janeiro. **Anais...** Rio de Janeiro: SBC, 2012.

AZEVEDO, R.; HADWIN, A. F. Scaffolding self-regulated learning and metacognition – Implications for the design of computer-based scaffolds. **Instructional Science**, vol. 33, n. 5, p. 367-379, 2005.

BADENHAUSEN, K. The World's Most Valuable Brands, 2016. Disponível em:< [http://www.forbes.com/powerful-brands/#/tab:rank\\_page:2](http://www.forbes.com/powerful-brands/#/tab:rank_page:2)>. Acesso em: 11 de jan. 2017.

BARATA, G.; GAMA, S.; JORGE, J.; GONÇALVES, D. Improving participation and learning with gamification. In: INTERNATIONAL CONFERENCE ON GAMEFUL DESIGN, RESEARCH, AND APPLICATIONS, 1, 2013, Toronto, Ontario, Canada. **Proceedings...** New York: ACM, p. 10-17, 2013.

BENNEDSEN, J.; CASPERSEN, M. E. Failure rates in introductory programming. **SIGCSE Bulletin**, v. 39, n. 2, p. 32-36, 2007.

BERGIN, S.; REILLY, R.; TRAYNOR, D. Examining the role of self-regulated learning on introductory programming performance. In: INTERNATIONAL COMPUTING EDUCATION RESEARCH, 2005, Seattle, USA. **Proceedings...** New York: ACM, p. 81-86, 2005.

BLIKSTEIN, P. O pensamento computacional e a reinvenção do computador na educação. 2008. Disponível em: [http://www.blikstein.com/paulo/documents/online/ol\\_pensamento\\_computacional.html](http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html). Acesso em: 06/06/2017.

BLIKSTEIN, P.; WORSLEY, M.; PIECH, C.; SAHAMI, M.; COOPER, S.; KOLLER, D. Programming pluralism: using learning analytics to detect patterns in the learning of computer programming. **Journal of the Learning Sciences**, vol. 23, n. 4, p. 1-39, 2014.

BOEKAERTS, M. Self-regulated learning: A new concept embraced by researchers, policy makers, educators, teachers, and students. **Learning and Instruction**, vol. 7, p. 161–186, 1997.

BOMBASAR, J. R.; SANTIAGO, R.; MIRANDA, E. M.; RAABE, A. L. A. Ferramentas para o Ensino-Aprendizagem do Pensamento Computacional: onde está Alan Turing? In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 26, 2015, Maceió. **Anais... Maceió: SBC**, 2015.

BORGES, K. S.; MORAES, M. A. C.; OKUYAMA, F. Y. Autorregulação da aprendizagem em computação com apoio da metodologia Scrum. **Texto livre: linguagem e tecnologia**, vol. 6, n. 2, 2013.

BROWN, A. L. Metacognition, executive control, self-regulation and other more mysterious mechanisms. In: F. E. WEINERT; R. H. KLUWE (eds.), **Metacognition, Motivation, and Understanding**, pp. 65–116. Lawrence Erlbaum Associates, Hillsdale, New Jersey. 1987.

BRUSILOVSKY, P.; MILLÁN, E. User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: **The Adaptive Web**. Berlin: Springer Berlin Heidelberg. 2007. p. 3-53.

BRUSILOVSKY, P.; SOMYŮREK, S.; GUERRA, J.; HOSSEINI, R.; ZADOROZHNY, V. The value of social: comparing open student modeling and open social student modeling. In: **CONFERENCE ON USER MODELING, ADAPTATION AND PERSONALIZATION**, 23., 2015, Dublin. Proceedings. Dublin: Springer, 2015.

BULL, S.; KAY, J. The SMILI (student models that invite the learner in): open learner modelling framework. **International Journal of Artificial Intelligence in Education**, Berlin, v. 17, n. 2, p. 89–120. 2007.

BULL, S.; KAY, J. Open Learner Models. In: **Advances in Intelligent Tutoring Systems**. Berlin: Springer Berlin Heidelberg. 2010. p. 301-322.

BULL, S.; KAY, J. Open Learner Models as drivers for Metacognitive processes. **International Handbook of Metacognition and Learning Technologies**, vol. 28, p. 349-365, 2013.

BULL, S.; KAY, J. SMILIJ: A Framework for Interfaces to Learning Data in Open Learner Models, Learning Analytics and Related Fields. **International Journal of Artificial Intelligence in Education**, v. 26, n. 1, New York: Springer New York, p. 293-331. 2016.

BULL, S.; NGHIEM, T. Helping Learners to Understand Themselves with a Learner Model Open to Students, Peers and Instructors. In: INTERNATIONAL CONFERENCE ON INTELLIGENT TUTORING SYSTEMS, 2002. **Proceedings of Workshop on Individual and Group Modelling Methods that Help Learners Understand Themselves, International Conference on Intelligent Tutoring Systems**, 2002, pp. 5-13.

BULL, S.; PAIN, H. 'Did I say what I think I said, and do you agree with me?': Inspecting and Questioning the Student Model. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, Charlottesville, VA, 1995. **Proceedings of World Conference On Artificial Intelligence In Education**, Charlottesville: AACE, 1995, p. 501-508.

CETIC. TIC empresas 2015. Disponível em: <<http://cetic.br/tics/empresas/2015/geral/F2/>>. Acesso em: 11 de jan. 2017.

CETIC. Proporção de domicílios com acesso à internet. 2016. Disponível em: <<http://data.cetic.br/cetic/explore>>. Acesso em: 13 de jan. 2017.

CHAGAS SCHNEIDER, F. **Comunidade virtual de prática na perspectiva da inclusão escolar: o perfil, os discursos e as práticas de educadores no exercício da cultura da**

participação. 2016. 165 f. Tese (Doutorado em Informática na Educação) – Programa de Pós-Graduação em Informática na Educação, UFRGS, Porto Alegre, 2016.

CHEN, G. D.; LI, L. Y.; WANG, C. Y. A community of practice approach to learning programming. **The Turkish Online Journal of Educational Technology**, v. 22, n. 2, p. 15-26. 2012.

CORDENONZI, W.; MÜLLER, T. J.; AMARAL, R. H.; PIOVESAN, S. D.; REATEGUI, E.; TAROUÇO, L. M. R.; LIMA, J. V. Mobile Q: construção de uma comunidade de prática sobre mobile learning. **RENOTE - Revista Novas Tecnologias na Educação**, Porto Alegre, v. 11, n. 1, p. 1-10. 2013.

CROW, D. Why every child should learn to code. 2014. Disponível em: <<https://www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick>>. Acesso em: 11 de jan. 2017.

DETERDING, S.; SICART, M.; NACKE, L.; OHARA, K.; DIXON, D. Gamification: Using Game Design Elements in Non-Gaming Contexts. In: INTERNATIONAL ACADEMIC MINDTREK CONFERENCE: ENVISIONING FUTURE MEDIA ENVIRONMENTS, 15, Tampere, Finland, 2011. **Proceedings...** New York: ACM, p. 9-15, 2011.

DIJKSTRA, E. W. A short introduction to the art of programming. Technological University Endhove, 1971. Disponível em: <<https://www.cs.utexas.edu/users/EWD/transcriptions/EWD03xx/EWD316.html>>. Acesso em: 02/02/2016.

DRUCKENMILLER, D. A.; MITTLEMAN, D. A Design theory for digital habitats: building virtual communities of practice. In: INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 48, 2015, Hawaii. **Proceedings...** Hawaii: IEEE, 2015.

EL-HANI, C. N.; GRECA, I. ComPratica: A Virtual Community of Practice for Promoting Biology Teachers' Professional Development in Brazil. **Research in Science Education**, v. 43, n. 4, p. 1327-1359. 2013.

ENGELBRECHT, A. M., DILERMANDO, P. J., NAKAMITI, G. S., BIANCHI, F.  
**Algoritmos e programação de computadores**. Editora Campus, São Paulo, 2012.

ERICKSON, T. Designing Visualizations of Social Activity: Six Claims. In: HUMAN FACTORS IN COMPUTING SYSTEMS, 2003, Ft. Lauderdale Florida, USA. **Proceedings...** New York: ACM Press. 2003.

ESTEVES, M.; ANTUNES, R.; FONSECA, B.; MORGADO, L.; MARTINS, P. Using Second Life in programming's communities of practice. In: **Groupware: Design, Implementation and Use**. Berlin: Springer Berlin Heidelberg. 2008. p. 99-106.

FARRELL, J. **Lógica e Design de Programação**. São Paulo: Editora Cengage, 2010.

FARZAN, R. BRUSILOVSKY, P. AnnotatEd: A social navigation and annotation service for web-based educational resources. **New Review of Hypermedia and Multimedia**, vol. 14, n. 1, 2008.

FESTINGER, L. A Theory of Social Comparison Processes. **Human Relations**, v. 7, n. 2, p.117-140, 1954

FLAVELL, J. H. Metacognition and cognitive monitoring: A new area of cognitive developmental inquiry. **American Psychologist**, vol. 34, p. 906-911, 1979.

FRANÇA, R. S.; TEDESCO, P. C. A. R. Um modelo colaborativo para a aprendizagem do pensamento computacional aliado à autorregulação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 25, Dourados, MS, 2014. **Anais...** Porto Alegre: SBC, p. 1133-1142, 2014.

FERREIRA, V. H.; REATEGUI, E. Uma Proposta de Trajetória de Aprendizagem Conceitual para Programação Introdutória. In: CICLO DE PALESTRAS SOBRE NOVAS TECNOLOGIAS NA EDUCAÇÃO, 24, 2016, Porto Alegre. **Anais...** Porto Alegre: UFRGS, v. 1. p. 267-271, 2016a.

FERREIRA, V. H.; REATEGUI, E. Uma Comunidade de Prática para o ensino e a aprendizagem de programação. **RENOTE. Revista Novas Tecnologias na Educação**, v. 13, n. 2, p. 1-10, 2016b.

FERREIRA, V. H.; REATEGUI, E. Uma Ferramenta de Mineração de Textos como Forma de Apoio à Compreensão de Enunciados de Exercícios de Programação. In: **COMPUTER ON THE BEACH**, 2017, Florianópolis. **Anais...** Itajaí: Univali, v. 1. p. 1-10, 2017a.

FERREIRA, V. H. REATEGUI, E. A Community of Practice to Support Learning of Computer Programming. In: **IEEE FRONTIERS IN EDUCATION CONFERENCE (FIE)**, 2017, Indianapolis, USA. **Proceedings...** Indianapolis: IEEE, 2017b. (Aceito)

GARNER, S.; HADEN, P.; ROBINS, A. My program is correct but it doesn't run: a preliminary investigation of novice programmers' problems. In: **AUSTRALASIAN COMPUTING EDUCATION CONFERENCE**, 7, 2005, Newcastle, Australia. **Proceedings...** Newcastle, Australia, p. 173-180, 2005.

GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002.

GOMES, A.; HENRIQUES, J.; MENDES, A. J. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. **Educação, Formação e Tecnologias**, vol. 1, n. 1, p. 93-103, 2008.

GUERRA, J.; SOMYUREK, S.; HOSSEINI, R.; BRUSILOVSKY, P. An intelligent interface for learning content: combining an open learner model and social comparison to support self-regulated learning and engagement. In: **INTERNATIONAL CONFERENCE ON INTELLIGENT USER INTERFACES**, 21, 2016, New York. Proceedings. New York: ACM, 2016.

GUERRA, J.; HUANG, Y.; HOSSEINI, R.; BRUSILOVSKY, P. Exploring the effects of Open Social Student Model beyond Social Comparison. In: **Workshop on Intelligent Support for Learning in Groups**, 4, 2015, Madrid. Proceedings. Madrid: AIED, 2015.

HÄNDEL, M.; ARTELT, C.; WEINERT, S. Assessing metacognitive knowledge:

Development and evaluation of a test instrument. **Journal for Educational Research Online**, vol. 5, n. 2, p. 162-168, 2013.

HARMS, K. J.; KERR, J. H.; ICHINCO, M.; SANTOLUCITO, M.; CHUCK, A.; KOSCIK, T.; CHOU, M.; KELLEHER, C. L. Designing a community to support long-term interest in programming for middle school children. In: International Conference on Interaction Design and Children, 11, 2012, Bremen, Germany. **Proceedings...** New York: ACM, p. 304-307, 2012.

HAVENGA, M.; BREED, B.; MENTZ, E.; GOVENDER, D.; GOVENDER, I.; DIGNUM, F. Metacognitive and problem-solving skills to promote self-directed learning in computer programming: teachers' experiences. **South African Journal of Education**, vol. 10, n. 2, 2013.

HENRIKSEN, P.; KÖLLING, M.; McCALL, D. Motivating programmers via an online community. **Journal of Computing Sciences in Colleges**, vol. 25, n. 3, Consortium for Computing Sciences in Colleges: USA, p. 82-93, 2010.

HOADLEY, C. What is a community of practice and how can we support it? In: Theoretical foundations of learning environments (2 ed). New York: Routledge. 2012.

HOSTINS, H.; RAABE, A. L. A. Auxiliando a aprendizagem de algoritmos com a ferramenta Webportugol. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 15, 2007, Rio de Janeiro. **Anais...** São Paulo: SBC, 2007.

HSIAO, I.; BRUSILOVSKY, P. The role of community feedback in the student example authoring process: An evaluation of AnnotEx. **British Journal of Educational Technology**, vol. 42, n. 3, p. 482-499, 2011.

HSIAO, I. H.; BAKALOV, F.; BRUSILOVSKY, P.; KÖNIG-RIES, B. Open social student modeling: visualizing student models with parallel introspective views. In: **User Modeling, Adaption and Personalization**. Berlin: Springer Berlin Heidelberg. 2011. p. 171-182.

IEPSEN, E. F. **Ensino de Algoritmos: Detecção do Estado Afetivo de Frustração**

**para Apoio ao Processo de Aprendizagem.** 2013. 157 f. Tese (Doutorado em Informática na Educação) – Programa de Pós-Graduação em Informática na Educação, UFRGS, Porto Alegre, 2013.

JESUS, A. de; BRITO, G. S. Concepção de ensino-aprendizagem de algoritmos e programação de computadores: a prática docente. In: ENCONTRO NACIONAL DE INFORMÁTICA NA EDUCAÇÃO, 1, 2009, Cascavel. **Anais...** Cascavel: UNIOESTE, Cascavel, 2009.

JÚNIOR, D. P.; FREITAS, R. L. Estratégias para melhorar os processos de abstração na disciplina de Algoritmos. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 21, 2010, João Pessoa. **Anais...** João Pessoa: SBC, 2010.

JÚNIOR, J. C. R. P. E RAPKIEWICZ, C. E. O Processo de ensino-aprendizagem de fundamentos de programação: uma visão crítica da pesquisa no Brasil. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 12, 2004, Salvador. **Anais...** São Paulo: SBC, 2004.

KNUTH, D. E. **The art of computer programming.** v.1 Reading, MA: Addison-Wesley, 1968.

KOIVULA, N.; HASSMEN, P.; HUNT, D. Performance on the Swedish scholastic aptitude test: Effects of self-assessment and gender. **Sex Roles: A Journal of Research**, v. 44, n. 12, p. 629–637, 2001.

LARA, B.; CAÑAS, F.; NADAL, A. V. N.; RIUS, F.; PAREDES, E.; MARAVALL, M. H. M. J.; FRANCH-NADAL, J.; BARBÉ, F.; MAURICIO, D. Knowledge management through two virtual communities of practice (Endobloc and Pneumobloc). **Health Informatics Journal**, p. 1-10. 2016.

LAHTINEN, E.; ALA-MUTKA, K.; JARVINEN, H-M. A study of the difficulties of novice programmers. In: INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION (ITiCSE), 10, 2005, New York. **Proceedings...** New York: ACM, 2005.

LAVE, J.; WENGER, E. **Situated learning**: Legitimate Peripheral Participation. Cambridge: Cambridge University Press. 1991.

LIMA FILHO, R. N.; BRUNI, A. L. Metacognitive Awareness Inventory: Tradução e Validação a partir de uma Análise Fatorial Confirmatória. **Psicologia: Ciência E Profissão**, v. 35, n. 4, p. 1275-1293, 2015.

LOBODA, T. D.; GUERRA, J.; HOSSEINI, R.; BRUSILOVSKY, P. Mastery grids: an open source social educational progress visualization. In: **Open Learning and Teaching in Educational Communities**. Berlin: Springer International Publishing. 2014. p. 235-248.

LOKSA, D.; KO, A. J. The Role of Self-Regulation in Programming Problem Solving Process and Success. In: CONFERENCE ON INTERNATIONAL COMPUTING EDUCATION RESEARCH, 2016, Melbourne. **Proceedings...** New York: ACM, p. 83-91, 2016.

LUCKIN, R.; HAMMERTON, L. Getting to Know Me: Helping Learners Understand Their Own Learning Needs through Metacognitive Scaffolding. In: CERRI, S.A.; GOUARDÈRES, G.; PARAGUAÇU, F. (eds). Intelligent Tutoring Systems. ITS 2002. Lecture Notes in Computer Science, vol. 2363. Springer, Berlin, Heidelberg, 2002.

MARTINEZ, M. E. What is metacognition? **Phi Delta Kappan**, vol. 87, n. 9, p. 696-699, 2006.

MEC. Diretrizes Curriculares Nacionais para os cursos de graduação em Computação.

Disponível em: <

[http://portal.mec.gov.br/index.php?option=com\\_docman&view=download&alias=11205-pces136-11-pdf&category\\_slug=julho-2012-pdf&Itemid=30192](http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=11205-pces136-11-pdf&category_slug=julho-2012-pdf&Itemid=30192)>. Acesso em: 02/02/2016.

MITROVIC, A.; MARTIN, B. Evaluating the Effect of Open Student Models on Self-Assessment. **International Journal of Artificial Intelligence in Education**, vol. 17, n. 2, p. 121-144, 2007.

NOSCHANG, L. F.; PELZ, F.; JESUS, E. A. de; RAABE, A. L. A. Portugol Studio: uma IDE para iniciantes em programação. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 22, 2014, Brasília. **Anais...** São Paulo: SBC, 2014.

PINTRICH, P. R. The role of goal orientation in self-regulated learning. In: M. BOEKAERTS, P. R. PINTRICH, & M. ZEIDNER (Eds.), **Handbook of self-regulation**, p. 451–502. San Diego, CA: Academic. 2000.

PINTRICH, P. R.; SMITH, D. A. F.; GARCIA, T.; MCKEACHIE, W. J. A manual for the use of the motivated strategies for learning questionnaire (mslq). **Technical report**, Ann Arbor, MI: National Center for Research to Improve Postsecondary Teaching and Learning. 1991.

RAABE, A. L. A. **Uma proposta de arquitetura de sistema tutor inteligente baseada na teoria das experiências de aprendizagem mediadas**. 2005. 155 f. Tese (Doutorado em Informática na Educação) – Programa de Pós-Graduação em Informática na Educação, UFRGS, Porto Alegre, 2005.

RAM, A.; AI, H.; RAM, P.; SAHAY, S. Open social learning communities. In: INTERNATIONAL CONFERENCE ON WEB INTELLIGENCE, MINING AND SEMANTICS, 2011, Sogndal, Norway. **Proceedings...** New York: ACM, 2011.

RAMOS, V.; FREITAS, M.; GALIMBERT, M.; MARIANI, A. C.; WAZLAWICK, R. A comparação da realidade mundial do ensino de programação para iniciantes com a realidade nacional: revisão sistemática da literatura em eventos brasileiros. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 26, 2015, Maceió. **Anais...** Maceió: SBC, p. 318-327, 2015.

RESNICK, M. All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In: SIGCHI CONFERENCE ON CREATIVITY & COGNITION, 6, 2007, Washington, DC, USA. **Proceedings...** New York: ACM, p. 1-6, 2007.

RESNICK, M.; MALONEY, J.; MONROY-HERNANDEZ, A.; RUSK, N.; EASTMOND, E.; BRENNAN, K.; MILLNER, A.; ROSENBAUM, E.; SILVER, J.; SILVERMAN, B.; KAFAI, Y. Scratch: Programming for All. **Communications of the ACM**, v. 52, n. 11, p. 60–67, 2009.

RIBEIRO, A. M.; SILVA, J. L. T.; BOFF, E.; VICCARI, R. M. Dos Ambientes de Aprendizagem às Comunidades de Prática. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 22, 2011, Aracaju. **Anais...** Aracaju: SBC, 2011.

ROBINS, A.; HADEN, P.; GARNER, S. Problem Distributions in a CS1 Course. In: AUSTRALASIAN COMPUTING EDUCATION CONFERENCE, 8, 2006, Hobart, Australia. **Proceedings...** Hobart, Australia, 2006.

ROBINS, A.; ROUNTREE, J.; ROUNTREE, N. Learning and teaching programming: a review and discussion. **Journal of Computer Science Education**, v. 13, n. 2, p. 137-172, 2003.

ROSA, O. S. **Aspectos Motivacionais do Cálculo Diferencial e Integral**. 2011. 117 f. Dissertação (Mestrado em Educação Matemática) – Programa de Pós-Graduação em Educação Matemática, Universidade Severino Sombra, Vassouras, RJ, 2011.

SCHOENFELD, A. H. **Mathematical Problem Solving**. New York: Academic Press, 1985.

SCHRAW, G.; DENNISON, R. S. Assessing metacognitive awareness. **Contemporary Educational Psychology**, v. 19, n. 4, p. 460-75, 1994.

SHEPHERD, M. M.; BRIGGS, R.O.; REINIG, B.A.; YEN, J.; NUNAMAKER J.F. Invoking Social Comparison to Improve Electronic Brainstorming: Beyond Anonymity. **Journal Of Management Information Systems**, v. 12, n. 3, p. 155-170, 1996.

SHI, L.; CRISTEA, A. I.; HADZIDEDIC, S. Multifaceted open social learner modelling. **Advances in Web-Based Learning – ICWL 2014**. Springer International Publishing, p. 32-42. 2014.

SILVA, T. R.; MEDEIROS, T. J.; MEDEIROS, H.; LOPES, R.; ARANHA, E. Ensino-aprendizagem de programação: uma revisão sistemática da literatura. **Revista Brasileira de Informática na Educação**, Porto Alegre, v. 23, n. 5, p. 182-196. 2015.

SMITH, M. Computer Science for all. 2016. Disponível em: <https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>. Acesso em: 06/06/2017.

SOUZA, D. M.; BATISTA, M. H. S.; BARBOSA, E. F. Problemas e dificuldades no ensino e na aprendizagem de programação: um mapeamento sistemático. **Revista Brasileira de Informática na Educação**, Porto Alegre, v. 24, n. 1, p. 39-52. 2016.

TOBIAS, S.; EVERSON, H. T. Knowing what you know and what you don't: further research on metacognitive knowledge monitoring. **College Board Research Report 2002-3**, College Entrance Examination Board: New York. 2002.

VASSILEVA, J.; SUN, L. Evolving A Social Visualization Design Aimed At Increasing Participation In A Class-Based Online Community. **International Journal of Cooperative Information Systems**, v. 17, n. 4, 2008.

VIHAVAINEN, A.; AIRAKSINEN, J.; WATSON, C. A systematic review of approaches for teaching introductory programming and their influence on success. In: INTERNATIONAL COMPUTING EDUCATION RESEARCH, 10, 2014, Glasgow, United Kingdom. **Proceedings...** New York: ACM, 2014.

VYGOTSKY, L. S. **A Formação Social da Mente**. São Paulo: Martins Fontes, 2007.

WATSON, C.; LI, F. W. B. Failure rates in introductory programming revisited. In: INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION (ITiCSE), 19, 2014, Uppsala, Sweden. **Proceedings...** New York: ACM, 2014.

WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, 2006. p. 33.

WOOD, D.; BRUNER, J.; ROSS, G. The role of tutoring in problem solving. **Journal of Child Psychology and Psychiatry and Allied Disciplines**. Vol. 17, p. 89–100, 1976.

ZANINI, A. S. **Avaliação da influência dos enunciados na resolução de problemas de programação introdutória**. 2013. 126 f. Dissertação (Mestrado em Computação Aplicada) – Programa de Pós-Graduação em Computação Aplicada, UNIVALI, Itajaí, 2013.n

ZIMMERMAN, B. J. Becoming a Self-Regulated Learner: which are the key ubprocesses? **Contemporary Educational Psychology**, vol. 11, n. 4, p. 307-313, 1986.

ZIMMERMAN, B. J. Becoming a self - regulated learner: an overview. **Theory into practice**, vol. 41, n. 2, p. 64-70, 2002.

## ANEXO A

### DETALHAMENTO DO QUESTIONÁRIO MAI

O questionário MAI utilizado nesta tese foi traduzido e validado por Lima Filho e Bruni (2015) a partir do questionário original proposto por Schraw e Dennison (1994). As categorias analisadas e as suas subescalas estão organizadas conforme o Quadro 10. Conforme pode ser observado, a categoria **Conhecimento cognitivo** será representada pela letra C e a categoria **Regulação da cognição** pela letra E.

Quadro 10. Subescalas do questionário MAI utilizadas no experimento.

Grupo	Subescala	Grupo	Subescala
C 1	Conhecimento declarativo	E 1	Planejamento
C 2	Conhecimento processual	E 2	Gestão da informação
C 3	Conhecimento condicional	E 3	Monitoramento
		E 4	Depuração
		E 5	Avaliação

Fonte: Lima Filho e Bruni (2015).

Os Quadros 11 à 13 apresentam as questões referentes às subescalas da categoria Conhecimento cognitivo.

Quadro 11. Questões da subescala Conhecimento declarativo.

Nº	Descrição das questões
5	Entendo meus pontos intelectuais fortes e fracos.
10	Eu sei que tipo de informação é mais importante aprender.
12	Sou bom em organizar informações.
16	Eu sei o que o professor espera que eu aprenda.
17	Eu sou bom em lembrar informações.
20	Eu tenho controle sobre o quanto eu aprendo.
32	Eu sou um bom juiz para saber como eu entendo as coisas.
46	Eu aprendo mais quando estou interessado no assunto.

Fonte: Lima Filho e Bruni (2015).

Quadro 12. Questões da subescala Conhecimento processual.

Nº	Descrição das questões
3	Tento usar estratégias que funcionaram no passado.
4	Eu me coloco em ritmo enquanto estou aprendendo para que eu tenha tempo suficiente.
27	Estou ciente das estratégias que utilizo quando estudo.
33	Utilizo estratégias de aprendizagem automaticamente.

Fonte: Lima Filho e Bruni (2015).

Quadro 13. Questões da subescala Conhecimento condicional.

Nº	Descrição das questões
15	Eu aprendo melhor quando eu já sei algo sobre o assunto.
18	Eu uso estratégias diferentes de aprendizagem, dependendo da situação.
26	Eu consigo me motivar a aprender, quando eu preciso.
29	Eu uso minhas forças intelectuais para compensar minhas fraquezas.
46	Eu aprendo mais quando estou interessado no assunto.

Fonte: Lima Filho e Bruni (2015).

Os Quadros 14 a 18 apresentam as questões referentes às subescalas da categoria Regulação da cognição.

Quadro 14. Questões da subescala Planejamento.

Nº	Descrição das questões
4	Eu me coloco em ritmo enquanto estou aprendendo para que eu tenha tempo suficiente.
6	Penso no que eu realmente preciso saber antes de iniciar uma tarefa.
8	Eu defino metas específicas antes de começar uma tarefa.
22	Faço-me perguntas sobre o material antes de começar a estudar.
23	Eu penso em várias maneiras de resolver um problema e escolho a melhor.
42	Eu leio as instruções cuidadosamente antes de começar uma tarefa.
45	Eu organizo meu tempo para realizar melhor meus objetivos.

Fonte: Lima Filho e Bruni (2015).

Quadro 15. Questões da subescala Gestão da informação.

<b>Nº</b>	<b>Descrição das questões</b>
9	Eu analiso com calma quando eu encontro informações importantes.
13	Eu conscientemente foco minha atenção nas informações importantes.
30	Concentro-me no sentido e importância das novas informações.
31	Eu crio meus próprios exemplos para tornar a informação mais significativa.
37	Construo imagens e diagramas para me auxiliar durante o estudo.
39	Eu tento traduzir novas informações em minhas próprias palavras.
41	Eu utilizo a estrutura organizacional do texto para me ajudar a aprender.
43	Pergunto-me se o que estou lendo é relacionado com o que eu já sei.
47	Eu tento dividir os estudos em etapas menores.
48	Minha concentração é global e não periférica.

Fonte: Lima Filho e Bruni (2015).

Quadro 16. Questões da subescala Monitoramento.

<b>Nº</b>	<b>Descrição das questões</b>
1	Pergunto-me periodicamente se estou atingindo as minhas metas.
2	Antes de resolver um problema analiso várias alternativas.
11	Pergunto-me se eu considere todas as opções quando estou resolvendo um problema.
21	Eu reviso periodicamente para me ajudar a entender as relações importantes.
28	Durante o estudo, consigo desenvolver estratégias de aprendizado.
34	Sempre faço pausas para verificar minha compreensão.
49	Analiso sobre o quão bem estou indo enquanto aprendo algo novo.

Fonte: Lima Filho e Bruni (2015).

Quadro 17. Questões da subescala Depuração.

<b>Nº</b>	<b>Descrição das questões</b>
25	Eu peço ajuda aos outros quando não entendo algo.
40	Eu mudo as estratégias quando eu não consigo entender.
44	Eu reavalio minhas suposições quando estou confuso.
51	Eu paro e volto quando uma informação não está clara.
52	Eu paro e releio quando fico confuso.

Fonte: Lima Filho e Bruni (2015).

Quadro 18. Questões da subescala Avaliação.

<b>Nº</b>	<b>Descrição das questões</b>
7	Eu sei o quão bem eu fiz quando eu termino uma tarefa.
19	Pergunto-me se tinha um jeito mais fácil de fazer coisas depois que termino uma tarefa.
24	Eu resumo tudo que aprendo.
36	Eu me pergunto o quão bem eu realizei meus objetivos quando concluo alguma atividade.
38	Pergunto-me se já considere todas as opções depois que eu resolvi algum problema.
50	Ao concluir um estudo, me pergunto se aprendi tanto quanto eu poderia ter aprendido.

Fonte: Lima Filho e Bruni (2015).

**ANEXO B****ROTEIRO DE ENTREVISTA SEMI-ESTRUTURADA**

- 1 - Como você estuda?
- 2 - Como você sabe que já estudou o suficiente?
- 3 - Você sentiu receio de compartilhar teu progresso com os colegas?
- 4 - Foi útil visualizar o teu progresso na disciplina?
- 5 - Foi útil ver teu progresso?
- 6 - Foi útil ver o progresso da turma?
- 7 - Você acompanhou o progresso de alguns colegas em específico?
- 8 - De que forma saber que os colegas veriam teu progresso te afetou?
- 9 - É importante para você comparar seu desempenho com o de algum colega?
- 10 - É importante para você comparar teu desempenho com a turma?
- 11 - Em algum momento se tornou mais importante aparecer teu nome com verde do que realmente aprender?
- 12 - Qual foi a diferença entre essa disciplina e outras?

**ANEXO C**

1. Faça um algoritmo que leia uma temperatura em graus Celsius e apresente-a convertida em graus Fahrenheit. A fórmula de conversão é:  $F = (9 * C + 160) / 5$ , na qual F é a temperatura em Fahrenheit e C é a temperatura em Celsius.
2. Faça um programa que solicite a idade dos indivíduos de um grupo enquanto a idade digitada não for 0 (zero). Após, informe qual foi a maior e a menor idade informada.
3. Faça um programa que preencha duas listas com 1000 valores inteiros em cada. Após, imprima a interseção dos valores das duas listas, ou seja, os valores que estão presentes nas duas.
4. Faça uma função que receba um caractere como parâmetro e retorne verdadeiro caso o caractere seja uma consoante e falso em caso contrário.

**ANEXO D****INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO  
GRANDE DO SUL – IFRS  
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO – PROPP  
COMITÊ DE ÉTICA EM PESQUISA – CEP****TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO**

Prezado (a) Senhor (a)

Você está sendo respeitosamente convidado (a) a participar do projeto de pesquisa intitulado: “A aprendizagem de programação apoiada pelo modelo social aberto do estudante”, cujo objetivo é investigar como o *scaffolding* meta-cognitivo proporcionado pela interação com o Modelo Social Aberto do Estudante pode contribuir para a aprendizagem de programação. Este projeto está vinculado ao Doutorado em Informática na Educação do Programa de Pós-Graduação em Informática na Educação da Universidade Federal do Rio Grande do Sul.

A pesquisa será feita no/a Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Feliz, através de entrevista, questionário e coleta de informações sobre a resolução de exercícios, que poderá ser gravada e/ou filmada, após minha autorização. Para a coleta de dados serão utilizados questionário, entrevista semi-estruturada e registros sobre a resolução de exercícios em ambiente virtual.

Fui alertado (a) que este estudo apresenta risco mínimo, isto é, serei levado a refletir sobre meu desempenho nos exercícios realizados e irei compartilhar com meus colegas informações sobre o meu desempenho, o que pode me causar desconforto. Caso isso ocorra, serei encaminhado para o coordenador do projeto, a fim de receber o acompanhamento necessário. Além disso, diante de qualquer tipo de questionamento ou dúvida poderei realizar o contato imediato com o pesquisador responsável pelo estudo que fornecerá os esclarecimentos necessários.

Foi destacado que minha participação no estudo é de extrema importância, uma vez que se espera compreender como a visualização e o compartilhamento do meu desempenho contribui para a aprendizagem de programação.

Estou ciente e me foram assegurados os seguintes direitos:

- da liberdade de retirar o meu consentimento, a qualquer momento, e deixar de participar do estudo, sem que isso me traga prejuízo de qualquer ordem;

- da segurança de que não serei identificado (a) e que será mantido caráter confidencial das informações relacionadas à minha privacidade;
- de que serão mantidos todos os preceitos ético-legais durante e após o término da pesquisa, de acordo com a Resolução 466/2016 do Conselho Nacional de Saúde;
- do compromisso de ter acesso às informações em todas as etapas do estudo, bem como aos resultados, ainda que isso possa afetar meu interesse em continuar participando da pesquisa;
- de que não haverá nenhum tipo de despesa ou ônus financeiro, bem como não haverá nenhuma recompensa financeira relacionada à minha participação;
- de que não está previsto nenhum tipo de procedimento invasivo, coleta de material biológico, ou experimento com seres humanos;
- de não responder qualquer pergunta que julgar constrangedora ou inadequada.

=====

Eu \_\_\_\_\_, portador do documento de identidade \_\_\_\_\_, aceito participar da pesquisa intitulada: “A aprendizagem de programação apoiada pelo modelo social aberto do estudante”. Fui informado (a) dos objetivos do presente estudo de maneira clara e detalhada, bem como sobre a metodologia que será adotada, sobre os riscos e benefícios envolvidos. Recebi uma cópia deste termo de consentimento e me foi dada a oportunidade de ler e esclarecer as minhas dúvidas.

### **Uso de gravação**

Autorizo o uso do áudio gravado de minha entrevista para fins da pesquisa, sendo seu uso restrito a análise do coordenador do projeto, que manterá em sigilo minha participação.

Local, \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_.

\_\_\_\_\_  
Assinatura do (a) participante

\_\_\_\_\_  
Assinatura do (a) pesquisador(a)

Em caso de dúvidas com respeito aos aspectos éticos deste estudo, você poderá consultar:

**CEP/IFRS****E-mail:** [cepesquisa@ifrs.edu.br](mailto:cepesquisa@ifrs.edu.br)**Endereço:** Rua General Osório, 348, Centro, Bento Gonçalves, RS, CEP: 95.700-000**Telefone:** (54) 3449-3340**Pesquisador(a) principal:** Vinicius Hartmann Ferreira**Telefone para contato:** 51 98308-6637**E-mail para contato:** [vinicius.ferreira@feliz.ifrs.edu.br](mailto:vinicius.ferreira@feliz.ifrs.edu.br)