

Departamento de Ciências e Tecnologias da Informação (DCTI-IUL)

**A APRENDIZAGEM DA PROGRAMAÇÃO NO ENSINO SUPERIOR:
A ADOÇÃO DE CURSOS *ONLINE* GAMIFICADOS**

Martinha do Rosário Fonseca Piteira
Mestre

Orientador:

Professor Doutor Carlos Jorge da Costa, Professor Associado no ISEG (Lisbon School of Economics & Management), Universidade de Lisboa

Coorientadora:

Professora Doutora Maria Manuela Símões Aparício da Costa, Professora Auxiliar no Instituto Universitário de Lisboa (ISCTE-IUL)

Dezembro de 2017

Departamento de Ciências e Tecnologias da Informação (DCTI-IUL)

**A APRENDIZAGEM DA PROGRAMAÇÃO NO ENSINO SUPERIOR:
A ADOÇÃO DE CURSOS *ONLINE* GAMIFICADOS**

Tese apresentada como requisito parcial para obtenção do Grau de Doutor em
Ciências e Tecnologias da Informação

Martinha do Rosário Fonseca Piteira

Mestre

Júri:

Professor Doutor Jorge Rodrigues da Costa, Professor Associado (ISCTE-IUL)

Professor Doutor António Maria Palma dos Reis, Professor Catedrático (ISEG -
Lisbon School of Economics & Management), Universidade de Lisboa

Professora Doutora Isabel Maria Mendes Pedrosa, Professora Adjunta (ISCAC)

Professor Doutor Abílio Gaspar Oliveira, Professor Auxiliar (ISCTE-IUL)

Professor Doutor Carlos Jorge da Costa, Professor Associado (ISEG - Lisbon School
of Economics & Management), Universidade de Lisboa

Professora Doutora Maria Manuela Símões Aparício da Costa, Professora Auxiliar
(ISCTE-IUL)

Aos meus pais,

Carlos e Sílvia

AGRADECIMENTOS

Em primeiro lugar quero deixar um reconhecimento profundo ao meu orientador, Professor Doutor Carlos J. da Costa, pela sua infinita paciência e disponibilidade durante todos estes longos anos, primeiro no mestrado, e agora no doutoramento. Foram preciosos os seus ensinamentos, a sua exigência sempre pelo aperfeiçoamento das minhas competências e pela excelência do trabalho que pretendia ver realizado. Certamente a concretização desta tese não teria sido possível sem a sua sabedoria em apontar os melhores caminhos de investigação.

À Professora Doutora Manuela Aparício, agradeço a constante disponibilidade, todos os contributos para a concretização deste trabalho, a sua confiança e permanente motivação.

À minha família, o meu suporte que durante todo este percurso, sempre estiveram presentes no seu apoio, compreensão e confiança.

Aos meus pais, Carlos e Sílvia, pilares fundamentais no meu crescimento enquanto pessoa e profissional. Os ensinamentos que passaram pelo seu exemplo fizeram de mim aquilo que sou hoje.

Aos meus sobrinhos Carla e Roberto, com um particular agradecimento à Carla, pela sua inestimável ajuda e apoio nas revisões do documento, nas revisões e traduções para o inglês dos artigos e pelos seus excelentes contributos gráficos. O meu muito obrigado!

Ao meu irmão José Fernando e à minha cunhada Deolinda o reconhecido agradecimento pelo apoio aos pais quando necessário, libertando-me assim para que conseguisse dedicar-me à realização do trabalho necessário para a concretização da tese.

À Geresa Marcolino, o meu agradecimento pelo seu apoio e constante incentivo.

Ao Nélio Folgôa, um reconhecido agradecimento pelos seus contributos na implementação do curso *online*. Foram longas as horas que passámos a analisar ferramentas, a construir e a validar a base de dados das perguntas para os testes *online*.

Ao Miguel Boavida e João Paulo Santos, meus colegas, presentes desde o início, incentivando-me e ouvindo-me vezes sem conta sobre os meus avanços e recuos. Obrigada por todo o incentivo e apoio na concretização da tese.

Ao Professor Doutor Nuno Nunes, o meu obrigado pelo apoio e suporte numa das fases do desenvolvimento da tese em que foi necessário dar forma a um artefacto físico.

Aos meus amigos Samir R. Haddad, Paulo Coelho e Isabel Vilhena, agradeço o apoio e incentivo.

Uma palavra de agradecimento em especial para o Samir R. Haddad, que desde a altura em que fomos colegas de mestrado, sempre foi uma presença e um suporte incondicional, em todo o meu percurso.

Ao Fernando Bento colega de doutoramento, um agradecimento pelo apoio e incentivo.

A todos que nomeei nestes agradecimentos, mas também para aqueles que de alguma forma contribuíram para a concretização deste trabalho, o meu reconhecido agradecimento!

RESUMO

A temática da aprendizagem da programação tem sido objeto de estudo crescente nos últimos anos. Como forma de mitigar as dificuldades de aprendizagem da programação diversas soluções foram propostas. Procura-se assim motivar e criar experiências de aprendizagem ricas e envolventes, centrar o estudante na aprendizagem, contribuindo assim para que o estudante não desista na primeira dificuldade encontrada. A presente tese estuda em primeiro lugar os diferentes tipos de problemas, percebidos por alunos e professores, na aprendizagem da programação, e a aplicação de soluções que possam criar um maior envolvimento por parte dos estudantes. Baseado em diversas teorias nomeadamente na teoria da gamificação e do *flow* é aqui proposta e validada uma framework teórica de cursos *online* no contexto da aprendizagem de programação. Neste sentido, foi desenvolvido numa plataforma *online* um curso gamificado para operacionalização da framework teórica. Foi realizado um estudo com estudantes de diversos cursos de ensino superior das áreas de engenharia, para aferir dos resultados percebidos acerca do curso *online*. Nesta tese foi também desenvolvido, um modelo teórico de adoção tecnológica de cursos *online* para a aprendizagem da programação, no qual se identificam os principais determinantes dessa adoção. Este modelo foi objeto de estudo e foi realizado um inquérito para validação das hipóteses aqui definidas. Como dimensões determinantes dessa adoção estão: fatores de personalidade dos estudantes, a facilidade de utilização da plataforma e do curso de aprendizagem de programação, o prazer na utilização do curso, a utilidade percebida do curso e o *flow*. Os vários estudos conduzidos no âmbito desta tese, demonstram que o desenho de cursos gamificados e os fatores de personalidade dos estudantes são fatores chave na aprendizagem da programação e na adoção de cursos *online* realizados para o efeito.

Palavras-chave: Gamificação, Aprendizagem de Programação, *eLearning*, Adoção da Tecnologia, *Flow*, Framework Teórica, CANOE.

ABSTRACT

The crescent importance of programming learning has been subjected to several studies in the last years. Those studies propose some solutions to mitigate programming learning difficulties. In this context, this dissertation proposes motivating rich experiences to engage students in programming learning, contributing to ensure perseverance in learners. This dissertation starts to present a study related to the difficulties in programming learning perceptions of students and teachers and presenting some solutions to engage students better. Based on gamification and flow theory, here is proposed a theoretical framework for programming learning online courses. In this setting, it was developed an online gamified course to instantiate the theoretical framework. It was conducted a study with several students of engineering programs, to understand their perception of the gamified online course. In this dissertation, was developed a theoretical, technological adoption model of programming learning online courses, in which were identified the main adoption determinants. This model was studied and conducted an empirical study, in which was surveyed to validate the theoretical hypotheses. Students' personality characteristics, course platform ease of use, enjoy, course usefulness, and course flow, are among the main determinant adoption dimensions. The various studies of this dissertation demonstrate that a gamified course design, and students' personality characteristics are key drivers in programming learning on-line adoption courses.

Keywords: Gamification, Programming Learning, *eLearning*, Technology Adoption, *Flow*, Theoretical Framework, CANOE.

LISTA DE PUBLICAÇÕES

- Piteira, M., Costa, C. J., & Aparicio, M. (2017). CANOE e Fluxo: determinantes na adoção de curso de programação online gamificado. *Iberian Journal of Information Systems and Technologies (RISTI)*, 2017(25). *In Press*.
- Piteira, M., Costa, C., & Aparício, M. (2017). A conceptual framework to implement gamification on online courses of computer programming learning: implementation. *ICERI2017 Proceedings* (pp. 7022–7031). Seville, SPAIN: IATED.
- Piteira, M., & Costa, C. J. (2017). Gamification: Conceptual framework to online courses of learning computer programming. *12th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–7). ISCTE-IUL - Lisboa.
- Piteira, M., & Costa, C. J. (2014). Aprendizagem da Programação: Problemas e Soluções (pp. 46-67). Apresentado na Conferência da Associação Portuguesa de Sistemas de Informação, Santarém. <http://dx.doi.org/10.18803/capsi.v14.46-67>.
- Piteira, M., & Costa, C. (2013). Learning Computer Programming: Study of Difficulties in Learning Programming. *Proceedings of the 2013 International Conference on Information Systems and Design of Communication* (pp. 75–80). New York, NY, USA: ACM. <https://doi.org/10.1145/2503859.2503871>.
- Piteira, M., & Costa, C. (2012). Computer Programming and Novice Programmers. *Proceedings of the Workshop on Information Systems and Design of Communication* (pp. 51–53). New York, NY, USA: ACM. <http://doi.org/10.1145/2311917.2311927>.
- Piteira, M., Costa, C., & Haddad, S. (2012). Educational Computer Programming Tools. *Proceedings of the Workshop on Open Source and Design of Communication* (pp. 57–60). New York, NY, USA: ACM. <http://doi.org/10.1145/2316936.2316947>.
- Piteira, M.; Costa, C. (2009). e-Games for e-Learning. In: *Eunis International Congress - IT: Key of the European Space Knowledge*. (v. 184, pp. 92). Santiago de Compostela. Universidade de Santiago de Compostela publicaci3ns.

ÍNDICE

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
LISTA DE PUBLICAÇÕES	VII
LISTA DE ABREVIATURAS E SIGLAS	XIX
1 INTRODUÇÃO	1
1.1 Fundamento teórico	2
1.2 Motivação científica	3
1.3 Questões e objetivos de investigação	4
1.4 Abordagem metodológica	6
1.5 Metodologia de investigação.....	7
1.5.1 <i>Design science research</i>	7
1.6 Contribuições	12
1.7 Estrutura da tese	13
2 REVISÃO CRÍTICA DA LITERATURA	15
2.1 Introdução.....	15
2.2 Problemas na aprendizagem da programação	15
2.3 Soluções propostas de suporte à aprendizagem da programação.....	18
2.3.1 Ferramentas de visualização gráfica	19
2.3.2 Artefactos tangíveis.....	21
2.3.3 Jogos educacionais digitais	23
2.3.4 Gamificação	25
2.4 Adoção da tecnologia	33
2.4.1 Modelos de adoção e uso da tecnologia	34
2.5 Conclusões	36
3 IDENTIFICAÇÃO DAS DIFICULDADES PERCECIONADAS PELOS ESTUDANTES NA APRENDIZAGEM DA PROGRAMAÇÃO - ESTUDO 1	39
3.1 Introdução.....	39
3.2 Objetivos	39
3.3 Abordagem metodológica	39

3.4	Questionários.....	40
3.4.1	Amostra	44
3.5	Recolha e tratamento dos dados	44
3.6	Apresentação dos resultados	45
3.6.1	Situações de aprendizagem	45
3.6.2	Conceitos de programação	46
3.6.3	Contextos de aprendizagem	48
3.6.4	Materiais/recursos	50
3.6.5	Atitude face à programação.....	51
3.7	Discussão de resultados.....	52
3.8	Conclusões do Estudo 1	54
4	COMPARAÇÃO DAS DIFICULDADES PERCECIONADAS NA APRENDIZAGEM DA PROGRAMAÇÃO – ESTUDO 2.....	57
4.1	Introdução.....	57
4.2	Objetivos	57
4.3	Abordagem metodológica	57
4.4	Resultados	58
4.4.1	Dificuldade nos conceitos de programação – Estudantes vs. Professores	58
4.4.2	Dificuldade nas situações de aprendizagem – Estudantes vs. Professores	59
4.4.3	Utilidade dos contextos de aprendizagem – Estudantes vs. Professores.....	59
4.4.4	Utilidade dos materiais/recursos – Estudantes vs. Professores	60
4.4.5	Dificuldade percecionada nos conceitos de programação vs. Resultados das provas de avaliação	61
4.5	Análise e discussão de resultados.....	62
4.6	Conclusões do Estudo 2	64
5	PROPOSTA E OPERACIONALIZAÇÃO DE <i>FRAMEWORK</i> TEÓRICA-CONCETUAL GAMIFICADA PARA CURSOS DE APRENDIZAGEM ONLINE DE PROGRAMAÇÃO – ESTUDO 3	67
5.1	Introdução.....	67
5.2	Objetivo.....	67
5.3	Abordagem metodológica	67
5.4	Fase 1 e Fase 2 - Enquadramento teórico.....	71
5.5	Fase 3: <i>Design</i> e desenvolvimento – <i>Framework</i> - Dimensões	73

5.6	Fase 4: Demonstração - Implementação da <i>framework</i> gamificada para cursos <i>online</i> de programação	80
5.6.1	Infraestrutura tecnológica de suporte à implementação do curso <i>online</i>	80
5.6.2	Elementos de desenho educacional	83
5.6.3	Operacionalização da <i>framework</i>	84
5.6.4	Método de avaliação das aprendizagens	95
5.6.5	Certificação das aprendizagens	96
5.6.6	Princípios de desenho educacional e mecânicas de jogo aplicados ao curso <i>online</i>	96
5.7	Fase 5: avaliação - resultados - atitudes dos estudantes face aos elementos de gamificação	106
5.8	Fase 6: Comunicação	110
5.8.1	Comunicação em eventos:.....	110
5.8.2	Comunicação: Publicações Científicas	112
5.9	Discussão de resultados.....	113
5.10	Conclusões do Estudo 3	115
6	PROPOSTA DE MODELO DE ADOÇÃO PARA CURSOS ONLINE DE PROGRAMAÇÃO GAMIFICADOS - ESTUDO 4	117
6.1	Introdução.....	117
6.2	Objectivos.....	117
6.3	Enquadramento teórico	117
6.3.1	A adoção da tecnologia	117
6.3.2	Conceito de <i>Flow</i>	118
6.3.3	As cinco dimensões da personalidade	119
6.4	Modelo proposto e hipóteses.....	119
6.4.1	Operacionalização das dimensões e recolha de dados	123
6.4.2	Resultados	125
6.4.3	Modelo de medida e avaliação	126
6.4.4	Avaliação do modelo estrutural.....	127
6.4.5	Discussão de resultados.....	129
6.4.6	Conclusões	130
7	CONCLUSÕES, LIMITAÇÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS	
	131	
7.1	Conclusões por objetivos de investigação.....	131
7.2	Limitações	134

7.3	Implicações Teóricas e Práticas	134
7.4	Trabalhos Futuros.....	134
	REFERÊNCIAS	137
	ANEXOS	159
	Anexo A – Questionário Dificuldades na Aprendizagem da Programação	161
	Anexo B - Questionário Percepções dos Estudantes – Gamificação	173
	Anexo C – Cross-Loadings	191
	Anexo D – Correlação entre Variáveis Latentes	193
	Anexo E – Modelo Estrutural.....	195
	Anexo F – Total Effects	197
	Anexo G – Path Coefficients.....	199
	Anexo H – Outer Weights	201
	Anexo I – Outer Loadings.....	203

LISTA DE TABELAS

Tabela 1 - Resultados da Metodologia <i>Design Science Research</i> (Järvinen, 2005; March & Smith, 1995).....	8
Tabela 2 - Diretrizes para a Condução/Avaliação do <i>Design Science Research</i> (Hevner et al., 2004).....	9
Tabela 3 - <i>Framework</i> de Investigação (March & Smith, 1995)	10
Tabela 4 – Estudos, Métodos e Técnicas de Investigação	11
Tabela 5 - Problemas na Aprendizagem da Programação, adaptado de (Costa, Aparicio, & Cordeiro, 2012)	18
Tabela 6 – Problemas e Soluções	24
Tabela 7 – Itens do Questionário - Situações de Aprendizagem.....	41
Tabela 8 – Itens do Questionário – Conceitos de Programação.....	41
Tabela 9 – Itens do Questionário – Contextos de Aprendizagem	42
Tabela 10 - Itens do Questionário - Materiais / Recursos	43
Tabela 11 – Itens do Questionário - Atitude Face à Programação.....	43
Tabela 12 - Mecânicas de Jogo	76
Tabela 13 - Princípios de Desenho Educacionais (adaptado de Dicheva et al., (2015).....	77
Tabela 14 - Personalidade (dimensões e adjetivos associados) – adaptado de John & Srivastava (1999)	78
Tabela 15 - Tabela resumo (medalhas e critérios aplicados)	104
Tabela 16 - Tabela Resumo das Mecânicas Aplicadas no Curso.....	105
Tabela 17 - Dimensão Barra de Progresso	106
Tabela 18 – Dimensão - Quadro de Honra (<i>LeaderBoard</i>).....	107
Tabela 19 – Dimensão - Medalhas	108
Tabela 20 – Dimensão - Pontos.....	109
Tabela 21 – DSRM (Fase 6): Comunicação - Publicações Científicas.....	112
Tabela 22 - Dimensões do Modelo Proposto	120
Tabela 23 - Variáveis Sociodemográficas - Estudo 4	124
Tabela 24 – Operacionalização das Dimensões	125
Tabela 25 - Resultados do Modelo de Medida.....	126
Tabela 26 - Testes de Hipóteses – Resultados	127

Tabela 27 - Objetivos de Investigação vs Estudos de Investigação..... 131

LISTA DE FIGURAS

Figura 1 - Abordagem Metodológica	6
Figura 2 - Taxonomia de Métodos de Investigação, adaptado de Järvinen (2004).....	11
Figura 3 - Estudos e Metodologias.....	12
Figura 4 - Gamificação Entre o Jogo e Jogar, o Todo e as Partes (Deterding et al., 2011)	26
Figura 5 - Hierarquia dos Elementos de Jogo (Werbach & Hunter, 2012).....	27
Figura 6 - Modelo TAM, adaptado de Davis (1989).....	35
Figura 7 – Modelo TAM2 (fonte: adaptado de (Venkatesh & Davis, 2000)	36
Figura 8 – Caracterização da Amostra	44
Figura 9 - Modelo de Processo da <i>Design Science Research Methodology (DSRM)</i> , adaptado de Vaishnavi & Kuechler (2004)	69
Figura 10 – Fases de <i>Design Science Research Methodology</i>	70
Figura 11- <i>Framework</i> Teórico-Concetual (dimensões).....	74
Figura 12 – <i>Framework</i> Gamificada para Cursos <i>Online</i> de Programação (Piteira & Costa, 2017).....	80
Figura 13 – Moodle: Ferramentas de suporte às atividades	82
Figura 14 – Moodle: Tipos de questões (testes online).....	82
Figura 15 - Instanciação da <i>Framework</i> Teórica-Concetual (Piteira, Costa, & Aparício, 2017)	85
Figura 16 - Fundamentos de Programação - SmartMOOC.....	85
Figura 17 – Módulo 0 – Ambientação à Plataforma - SmartMOOC	86
Figura 18 - Módulo 1 – Introdução às Linguagens de Programação - SmartMOOC	87
Figura 19 - Módulo 2 – Variáveis - SmartMOOC	88
Figura 20 – Módulo 3 – Entrada e Saída de Dados - SmartMOOC.....	89
Figura 21 - Módulo 4 – Operadores - SmartMOOC	90
Figura 22 - Módulo 5 – Estruturas de Seleção -SmartMOOC	91
Figura 23 - Módulo 6 – Estruturas de Repetição - SmartMOOC.....	92
Figura 24 - Módulo 7 – Vetores - SmartMOOC	93
Figura 25 - Módulo 8 – Subprogramas - SmartMOOC	94
Figura 26 - Certificado de Conclusão de Curso Online - SmartMOOC	96
Figura 27 – Barra de Progresso	97

Figura 28 - Progressão dos Estudantes nas Atividades do Curso	97
Figura 29 - Submissão das Avaliações – Retorno ao Estudante	98
Figura 30 - Fórum Pergunta-Resposta	99
Figura 31 - Acesso Condicionado às Atividades	100
Figura 32 - Acesso Restrito às Atividades	100
Figura 33 - Medalha Caloiro	102
Figura 34 - Medalhas da Categoria <i>Geek</i>	102
Figura 35 - Medalhas da Categoria <i>Super Geek</i>	102
Figura 36 – Medalhas da Categoria <i>Guru</i>	103
Figura 37 - Medalha Categoria <i>Super Guru</i>	103
Figura 38 – Apresentação do SmartMOOC na CodeWeek, ISCTE, abril 2017	111
Figura 39 - Entrega dos Certificados de Conclusão do SmartMOOC, abril 2017	111
Figura 40 – Divulgação do SmartMOOC na Noite dos Investigadores, 2017 in Site http://noitedosinvestigadores.org/programa/lisboa/	112
Figura 41 - Modelo de Investigação.....	123
Figura 42 - Resultados do Modelo e Análise Estrutural	128

LISTA DE ILUSTRAÇÕES

Ilustração 1 - Etapas da concepção do questionário de investigação	40
Ilustração 2 - Etapas da Concepção do Questionário de Investigação – Gamificação	124

LISTA DE GRÁFICOS

Gráfico 1 – Dificuldade nas Situações de Aprendizagem – Percepção dos Estudantes	45
Gráfico 2 – Dificuldade nas Situações de Aprendizagem – Percepção dos Professores	46
Gráfico 3 – Dificuldade nos Conceitos de Programação – Percepção dos Estudantes	47
Gráfico 4 - Dificuldade nos Conceitos de Programação – Percepção dos Professores.....	48
Gráfico 5 – Utilidade dos Contextos de Aprendizagem – Percepção dos Estudantes	49
Gráfico 6 – Utilidade dos Contextos de Aprendizagem – Percepção dos Professores	49
Gráfico 7 - Utilidade dos Materiais/Recursos – Percepção dos Estudantes.....	50
Gráfico 8 – Utilidade dos Materiais/Recursos – Percepção dos Professores.....	51
Gráfico 9 - Atitude dos Estudantes Face à Programação	52
Gráfico 10 – Dificuldade nos Conceitos de Programação - Estudantes vs. Professores.....	58
Gráfico 11 – Dificuldade nas Situações de Aprendizagem - Estudantes vs. Professores	59
Gráfico 12 – Utilidade dos Contextos de Aprendizagem - Estudantes vs. Professores.....	60
Gráfico 13 – Utilidade dos Materiais/Recursos - Estudantes vs. Professores.....	60
Gráfico 14 – Dificuldade nos Conceitos de Programação – (Professores vs. Estudantes vs. Provas de Avaliação).....	61
Gráfico 15 – Atitude dos Estudantes Face à Barra de Progresso	106
Gráfico 16 – Atitude dos Estudantes Face ao Quadro de Honra.....	107
Gráfico 17 – Atitude dos Estudantes Face às Medalhas	108
Gráfico 18 – Atitude dos Estudantes Face aos Pontos	109
Gráfico 19 - Atitude dos Estudantes Face às Medalhas vs. Pontos.....	110

LISTA DE ABREVIATURAS E SIGLAS

BS	<i>Behavioral Science</i>
CANOE	<i>Conscientiousness, Agreeableness, Neuroticism, Openness to Experience, Extroversion</i>
cMOOC	<i>Connective Massive Open Online Course</i>
CSE	<i>Computer Science Education</i>
DSR	<i>Design Science Research</i>
DSRM	<i>Design Science Research Methodology</i>
IDE	<i>Integrated Developer Environment</i>
LMS	<i>Learning Management System</i>
MEE	Modelo de Equações Estruturais
MOOC	<i>Massive Open Online Courses</i>
NS	<i>Natural Science</i>
PLS	<i>Partial Least Square</i>
SEM	<i>Structural Equation Models</i>
TAM	<i>Technology Acceptance Model</i>
TI	Tecnologias de Informação
TPB	<i>Theory of Planned Behavior</i>
TRA	<i>Theory of Reasoned Action</i>
UTAUT	<i>Unified Theory of Acceptance and Use of Technology</i>
VME	Variância Média Extraída
xMOOC	<i>Extended Massive Open Online Course</i>

1 INTRODUÇÃO

A aprendizagem da programação é uma das principais áreas no ensino da informática, através da qual o estudante adquire competências quer ao nível prático quer concetual. Contudo, ao longo dos tempos, têm-se vindo a verificar diversos problemas relativos à aprendizagem da programação, entretanto reportados na literatura (Bouvier et al., 2016; Chao, 2016; Hawi, 2010; Lahtinen, Ala-Mutka, & Järvinen, 2005; Malik & Coldwell-Neilson, 2017; Pears et al., 2007; Piteira & Costa, 2013; Rahmat et al., 2012; Renumol, Janakiram, & Jayaprakash, 2010; Robins, Rountree, & Rountree, 2003; Salleh, Shukur, & Judi, 2013). Por exemplo, Pears et al. (2007) conduziram uma revisão sistemática da literatura e identificaram vários problemas associados ao currículo, pedagogia, paradigma de programação e à linguagem de programação utilizada para ensinar estudantes iniciantes. Identificaram também possíveis soluções que podem contribuir para reduzir e ultrapassar essas dificuldades. Jenkins (Jenkins, 2002) discute questões associadas à atitude, fatores cognitivos, estilos de aprendizagem, motivação e Lahtinen, Ala-Mutka e Järvinen (2005) discutem questões relacionadas com a perceção da dificuldade nos conceitos de programação. A maioria dos estudantes acredita que a programação é uma matéria de difícil compreensão. Esta perceção é passada aos novos estudantes transmitindo uma impressão de dificuldade na aprendizagem da programação (Rahmat et al., 2012). Estes problemas refletem-se nas taxas de reprovação e de desistência que as disciplinas de programação apresentam e apesar dos problemas na aprendizagem serem uma temática estudada e investigada, constata-se que os problemas persistem e que é necessário continuar a investigar e a propor novas abordagens que contribuam para os mitigar (Denning, 2004; Kinnunen, Marttila-Kontio, & Pesonen, 2013; Rahmat et al., 2012) e que contribuam para estimular e incentivar os estudantes para a aprendizagem, de forma que consigam ultrapassar as dificuldades encontradas e não desmotivem ou desistam da aprendizagem.

Neste sentido, esta investigação, estuda os problemas na aprendizagem da programação e propõe uma solução *online* que integra características de jogos como forma de envolver e estimular os estudantes para a aprendizagem da programação. A utilização dos elementos com características de jogos é designada por *gamification* (usualmente designada por gamificação em língua portuguesa). Este é um conceito recente e tem vindo a ser introduzido em diversas áreas, em particular na educação. Contudo, a sua utilização e aceitação em contexto educacional

e particularmente na programação, não tem ainda uma explicação concisa e fundamentada. Assim, esta tese estuda os fatores determinantes na utilização e adoção da gamificação em contexto de aprendizagem da programação *online*.

A investigação empírica é apoiada na revisão da literatura centrada no tema da investigação e estrutura-se em quatro estudos, visando assim responder às questões de investigação e aos objetivos propostos.

1.1 Fundamento teórico

Saber programar, é atualmente uma das competências mais procuradas e valorizadas no mercado de trabalho (Gardiner, Aasheim, Rutner, & Williams, 2017; Yadav, Stephenson, & Hong, 2017). Corresponde ainda a um conjunto de conhecimentos basilares aos cursos de informática no ensino superior. No entanto, a programação tem taxas de reprovação significativas no primeiro ano e são uma evidência de que a aprendizagem da programação para os estudantes é uma tarefa difícil (Rahmat et al., 2012). Diversos autores têm estudado esta problemática e identificado alguns dos aspetos que podem contribuir para a dificuldade e conseqüente retenção. Por exemplo, (Lahtinen et al., 2005; Schulte & Bennedsen, 2006) apontam como principais causas, os métodos tradicionais de ensino, baseados em aulas expositivas sobre as sintaxes das linguagens, os quais não permitem envolver os estudantes em atividades de programação com sentido, contribuindo de forma negativa para a motivação dos estudantes. Também as linguagens que se ensinam são apontadas como tendo um impacto negativo, principalmente para os estudantes iniciantes (Jenkins, 2002). A natureza abstrata da programação representa também um desafio e uma barreira à aprendizagem. Os estudantes têm dificuldade com os conceitos abstratos (Lahtinen et al., 2005; Piteira & Costa, 2012, 2013, 2014; Rahmat et al., 2012) porque não têm uma analogia direta com a vida real. Mesmo os conceitos mais básicos são por vezes de difícil compreensão e de aplicação, principalmente quando têm de ser utilizados em conjunto para a resolução de um determinado problema (Winslow, 1996). Contudo, as dificuldades continuam a persistir e diversos autores e.g., (Hawi, 2010; Rahmat et al., 2012) apontam como necessária mais investigação relacionada com os problemas na aprendizagem da programação. De forma a ultrapassar as dificuldades que os estudantes enfrentam, diversos autores têm proposto diferentes soluções (Helminen, 2009; Ibáñez, Di-Serio, & Delgado-Kloos, 2014; McGill, 2012b; Sorva & Sirkiä, 2010). Por exemplo, soluções centradas na utilização das tecnologias, nomeadamente ferramentas de visualização

gráfica (Sorva & Sirkiä, 2010), robótica educacional (McGill, 2012a) e as recentes soluções baseadas na gamificação (Ibáñez et al., 2014).

1.2 Motivação científica

Os problemas que os estudantes enfrentam na aprendizagem da programação têm sido estudados por diversos autores e.g., (Konecki, 2014; Rahmat et al., 2012; Salleh et al., 2013; Watson & Li, 2014), assim como propostas para ultrapassar essas dificuldades, têm sido apresentadas e.g. (Álvarez & Larrañaga, 2013, 2013; Liu, Lee, Kang, & Liu, 2016; Piteira, Costa, & Haddad, 2012). Contudo, os estudantes continuam a enfrentar dificuldades de natureza diversa na aprendizagem da programação (Piteira & Costa, 2013, 2014; Yadav et al., 2017). Nesse sentido, o aprofundamento do conhecimento referente às dificuldades sentidas pelos estudantes na aprendizagem da programação é uma necessidade, de modo a contribuir para a sua mitigação com a consequente motivação dos estudantes para a aprendizagem e redução nas taxas de reprovação. Por outro lado, o modo como hoje se ensina e se aprende tem vindo a ser alterado, em grande medida impulsionado pela evolução da tecnologia. O estudante deixou de estar confinado a um espaço e limitado a um período temporal, podendo efetuar a aprendizagem ao seu ritmo. Através da Web, podemos atualmente aceder a conteúdos pedagógicos *online*, interagir e colaborar com colegas e com os educadores, realizar atividades pedagógicas, submeter trabalhos online, realizar testes online com a possibilidade de obter retorno imediato dessas avaliações (Lam, McNaught, Lee, & Chan, 2014). Nos últimos anos assistimos a uma mudança vertiginosa no ensino superior, no modo como o ensino/aprendizagem acontece, impulsionada pela educação à distância e *online* e pelas plataformas de *e-Learning*, que emergiram em resultado da evolução da tecnologia, permitindo assim criar oferta formativa online ou complementar às aulas presenciais (Lam et al., 2014). Por outro lado, e apesar das plataformas de *e-Learning* disponibilizarem um conjunto diversificado de ferramentas tecnológicas educacionais, os estudantes nem sempre estão motivados para interagir com a ferramenta e consequentemente, com os conteúdos e com as atividades (Fryer, Nicholas Bovee, & Nakao, 2014). Torna-se assim necessário estudar e investigar mecanismos que contribuam para envolver o estudante com a aprendizagem, em particular com a aprendizagem da programação.

Nesse sentido, a integração de elementos que estimulem, envolvam e criem uma absorção do estudante com a aprendizagem da programação, parecem ser uma abordagem a explorar. A gamificação tem sido referida e apontada por diversos autores (Deterding, Dixon, Khaled, &

Nacke, 2011; Dicheva, Dichev, Agre, & Angelova, 2015; Ibáñez et al., 2014; Kim & Lee, 2015; Seaborn & Fels, 2015) como uma abordagem com potencial para aumentar e estimular o envolvimento dos estudantes com as atividades de aprendizagem. Por exemplo, Kim & Lee (2015) referem que o propósito da gamificação é aumentar o envolvimento do indivíduo com o sistema educacional através do uso de mecânicas de jogo, encorajando e permitindo experiências ricas de aprendizagem. Contudo, a integração da gamificação em cursos *online* de aprendizagem da programação suportados por uma plataforma online, requer uma *framework* (quadro conceitual teórico) que guie os educadores na estruturação do curso e na integração dos elementos de jogo. Apesar de existirem na literatura diversas *frameworks* (Chou, 2015; Werbach & Hunter, 2012), nenhuma delas é específica para a implementação da gamificação em cursos *online* de aprendizagem da programação. Apesar dos benefícios referidos por diversos autores (Dicheva et al., 2015; Ibáñez et al., 2014) na utilização da gamificação em contexto educacional, a adoção e o uso de cursos *online* gamificados, pode não estar dependente da abordagem em si, mas também das características individuais dos estudantes e da capacidade da abordagem em envolver, estimular e absorver os estudantes na aprendizagem. Diversos estudos avaliaram o impacto da gamificação e.g. (Attali & Arieli-Attali, 2015; Hamari, 2017; Seixas, Gomes, & Filho, 2016), contudo, nenhum estudo avaliou as características do indivíduo e o modo como essas características influenciam a adoção da gamificação na aprendizagem da programação, como também a influência que os estados de imersão e o envolvimento (*flow*), que o estudante experiencia durante a aprendizagem, pode ter nessa adoção.

A presente tese contextualiza-se assim, no conhecimento referente às dificuldades na aprendizagem da programação e nas soluções propostas. Sendo a gamificação emergente no ensino, este estudo pretende também explorar a sua utilização na aprendizagem da programação e particularmente, a sua utilização numa plataforma *online* de aprendizagem (Dicheva et al., 2015), e identificar os determinantes que influenciam a adoção da gamificação nos cursos *online*.

1.3 Questões e objetivos de investigação

Estima-se que nos próximos anos o mercado de trabalho necessite de um grande número de profissionais com competências em programação de computadores (EU, 2016; Yadav et al., 2017). Contudo, devido à natureza abstrata da programação, verifica-se que as taxas de reprovação nesta unidade curricular se mantêm elevadas, levando a que os estudantes desmotivem e desistam ou fiquem retidos por mais tempo nos cursos de informática (Pappas,

Giannakos, & Jaccheri, 2016). Nos últimos anos, têm surgido várias soluções com características de jogos, que têm sido utilizadas em diversos contextos, tendo tido uma grande aceitação por parte dos educadores e dos investigadores (Dicheva et al., 2015; Ibáñez et al., 2014). Essas soluções são consideradas emergentes, inovadoras e com potencial para a motivação dos estudantes na aprendizagem da programação. Assim estas temáticas, dificuldades na aprendizagem da programação e a utilização da gamificação, conduziram a duas questões de investigação, concretamente:

Q1 – Quais as dificuldades dos estudantes na aprendizagem da programação e soluções que contribuam para ultrapassar essas dificuldades?

Q2 – Qual o impacto da gamificação nos cursos *online*?

A investigação realizada foi orientada pelas questões de investigação, mais especificamente, pelos seguintes objetivos de investigação:

OBJ1: Identificar as dificuldades percecionadas na aprendizagem da programação.

A identificação das dificuldades percecionadas pelos estudantes, a identificação das dificuldades percecionadas pelos professores relativamente aos estudantes e a comparação dessas perceções com as provas de avaliação, é o principal resultado obtido com a concretização deste objetivo.

OBJ2: Identificar as soluções facilitadoras da aprendizagem da programação.

Através da identificação na revisão da literatura das soluções facilitadoras, obtivemos como principal resultado a matriz de problemas e possíveis soluções para esses problemas.

OBJ3: Propor *framework* (quadro concetual teórico) gamificada para cursos *online* de programação.

Na literatura revista, é largamente utilizado o termo *framework*, como referência a quadro concetual teórico. Nesse sentido, nesta tese adotamos o termo *framework*.

A *framework* teórica é o resultado deste objetivo. É suportado na revisão da literatura e em estudos anteriormente realizados, no âmbito da presente tese.

OBJ4: Implementar a *framework* teórica gamificada.

A implementação da *framework* teórica é o primeiro resultado deste objetivo. Esta implementação foi efetuada através do desenvolvimento de um curso *online* de fundamentos de programação.

O segundo resultado obtido com a concretização deste objetivo, foi a construção de um questionário para obtenção das perceções dos estudantes.

OBJ5: Proposta de um modelo de adoção da gamificação para cursos *online* gamificados.

O modelo de adoção é o resultado deste objetivo. É suportado na revisão da literatura e nos estudos anteriores. O desenvolvimento do modelo foi efetuado em duas fases: construção e validação.

1.4 Abordagem metodológica

Na Figura 1, descreve-se a abordagem metodológica que orientou o desenvolvimento da presente tese.

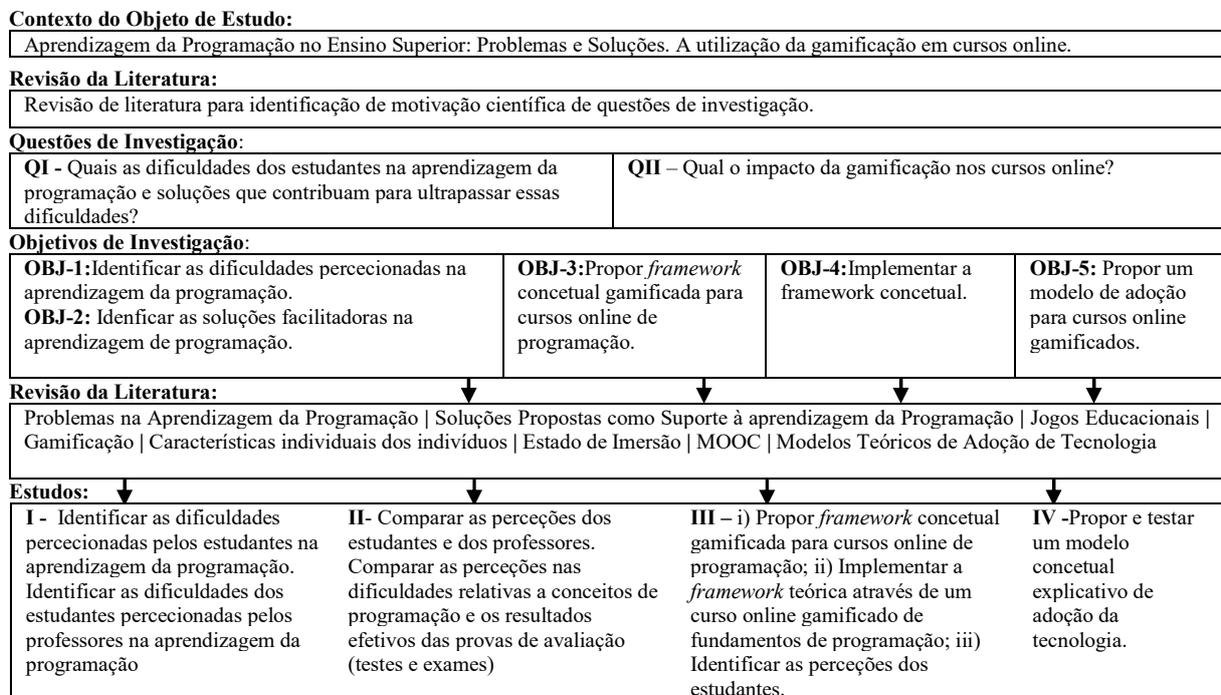


Figura 1 - Abordagem Metodológica

1.5 Metodologia de investigação

O desenvolvimento de um trabalho de investigação deve ser suportado e conduzido por uma metodologia científica, para desse modo poder ser considerado sólido e relevante. A metodologia assegura que o trabalho é desenvolvido de uma forma lógica, com recurso a métodos e técnicas, como meio para atingir o fim do estudo do problema em análise e poder retirar conclusões. Neste sentido, descrevem-se nos subcapítulos seguintes, as metodologias e métodos que suportam esta tese.

1.5.1 *Design science research*

Os métodos de investigação em sistemas de informação têm sido frequentemente divididos em dois paradigmas de investigação: Ciências Comportamentais (*Behavioral Science*) e Ciências Naturais (*Natural Science*) (Hevner & Chatterjee, 2010; Hevner, March, Park, & Ram, 2004; Järvinen, 2004, 2005; March & Smith, 1995; Simon, 1996; Vaishnavi & Kuechler, 2004). Contudo, essas divisões têm ignorado um paradigma de investigação que é essencialmente criativo e orientado através da solução de problemas, a ciência do *design* (*design science*). A ciência do *design* foi conceptualizada por Simon (1996) e suporta uma pesquisa pragmática que representa a criação de artefactos inovadores para resolver problemas reais. Esta pesquisa em ciência do *design*, combina o foco no artefacto de sistemas de informação com uma elevada prioridade na relevância da aplicação no seu domínio. Hevner e Chatterjee (2010) descrevem a ciência do *design* como: “*Design science research is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence. The designed artifacts are both useful and fundamental in understanding that problem*” (Hevner & Chatterjee, 2010, p. 5).

March e Smith (1995) identificaram dois processos na ciência do *design*: Construir e Avaliar. O processo de Construir é um passo importante a partir do *design*, dado que é um processo mais abstrato que o Construir, sendo este uma realização física do *design*. O avaliar é um processo de análise onde é determinada a eficácia dos artefactos desenvolvidos. Estes autores identificaram quatro tipos de artefactos de *design* criados pelo *design science research* (DSR), mais concretamente: dimensões (*constructs*), modelos (*models*), métodos (*methods*) e instanciações (*instantiations*). As dimensões são os conceitos elementares da conceção do problema. Os modelos incluem relações relevantes entre as dimensões. Os métodos especificam o processo de como realizar ou implementar uma determinada tarefa. O produto da tarefa deverá ser um artefacto, em sistemas de informação e a ciência do *design* deverá ter como resultado

final, a construção de um artefacto que se adequa à interação entre os sistemas, as tecnologias e as pessoas. Os artefactos são também designados por instanciações (Hevner & Chatterjee, 2010; Hevner et al., 2004; Järvinen, 2004, 2005; March & Smith, 1995). As instanciações são realizações do *design* físico ou abstrato do produto. Esse produto poderá posteriormente ser aplicado, numa organização ou num ambiente social. Na Tabela 1 resumem-se, os resultados da metodologia *Design Science Research*.

Tabela 1 - Resultados da Metodologia *Design Science Research* (Järvinen, 2005; March & Smith, 1995)

Resultados de Investigação (Artefactos)	Descrição	Autor
Dimensões (<i>constructs</i>)	As dimensões formam o vocabulário de um domínio. Constituem uma concetualização utilizada para descrever os problemas dentro do domínio e para especificar as respetivas soluções. As concetualizações são importantes em ambas as ciências: natural e de <i>design</i> . Definem os termos usados para descrever e pensar sobre as tarefas. Podem ser de grande utilidade para <i>designers</i> e investigadores.	
Modelos (<i>models</i>)	Um modelo é um conjunto de proposições ou declarações que expressam as relações entre as dimensões. Nas atividades de <i>design</i> , modelos representam situações como problema e solução. Pode ser visto como uma descrição, i.e., uma representação de como as coisas são. Em ciências naturais utiliza-se por vezes o termo “modelo” como sinónimo de “teoria” ou “modelos” quando as teorias são ainda incipientes. Em <i>Design Science</i> , a preocupação é a utilidade de modelos, não a adesão da sua representação à Verdade. Contudo, embora o modelo tenda a ser impreciso sobre detalhes, precisa de capturar a estrutura da realidade para ser uma representação útil.	(Järvinen, 2005; March & Smith, 1995)
Métodos (<i>methods</i>)	Um método é um conjunto de passos (um algoritmo ou orientação) utilizado para executar uma tarefa. Os métodos baseiam-se em dimensões subjacentes (linguagem) e uma representação (modelo), num espaço de solução. Os métodos podem ser ligados aos modelos, nos quais as etapas do método podem utilizar partes do modelo como uma entrada que o compõe. Os métodos podem ainda ser utilizados para traduzir um modelo ou representação em curso para resolução de um problema. O método é a descrição do processo científico.	
Instanciações (<i>instantiations</i>)	Uma instanciação é a concretização de um artefacto no seu ambiente. Instanciações operacionalizam dimensões, modelos e métodos. Contudo, uma instanciação pode na prática, preceder a articulação completa das suas dimensões, modelos e métodos. As instanciações demonstram a viabilidade e a eficácia dos modelos e dos métodos que as instanciações contemplam.	

De forma a orientar a investigação em sistemas de informação e ajudar os investigadores a desenvolver atividades relacionadas com a ciência do *design* Hevner et. al. (2004), desenvolveram uma *framework* teórica concetual e um conjunto de diretrizes e princípios para

a condução e avaliação de uma investigação sólida e robusta em *Design Science Research*. Na Tabela 2 descrevem-se, as diretrizes propostas.

Tabela 2 - Diretrizes para a Condução/Avaliação do *Design Science Research* (Hevner et al., 2004)

Diretrizes	Descrição
Design como artefacto	A investigação em <i>Design Science</i> deve produzir um artefacto viável na forma de dimensões, modelos, métodos ou instanciações.
Relevância do Problema	O objetivo da investigação em <i>Design Science</i> é desenvolver soluções baseadas em tecnologias para problemas importantes e relevantes de um determinado contexto.
Avaliação do Design	A utilidade, qualidade e eficácia do artefacto, deve ser demonstrada por métodos de avaliação bem executados.
Contribuições do Design	Uma investigação em <i>Design Science</i> , deve gerar contribuições claras e verificáveis nas áreas específicas dos artefactos desenvolvidos e apresentar fundamentação do <i>design</i> e das metodologias do design.
Rigor da Investigação	A investigação em <i>Design Science</i> é baseada, na aplicação de métodos rigorosos, tanto na construção como na avaliação dos artefactos.
Design como processo de investigação	A procura por um artefacto efetivo, requer a utilização de meios que estejam disponíveis, para alcançar os fins pretendidos, ao mesmo tempo que cumpre com as leis que regem o ambiente que está a ser estudado.
Comunicação da investigação	A investigação, em <i>Design Science</i> deve ser apresentada de forma efetiva para o público orientado à tecnologia, como ao público orientado à gestão da tecnologia.

Tendo por base as diretrizes propostas por Hevner et. al. (2004) e na framework de investigação proposta por March e Smith (1995), apresentam-se na Tabela 3, as atividades de investigação desenvolvidas na presente tese, orientadas pelas metodologias de investigação em Ciências do *Design* e Ciências Naturais. Apresentam-se também os resultados de investigação obtidos em cada uma dessas atividades. Assim, para o resultado de investigação *Modelo*, identificou-se na atividade de investigação Construir, a Proposta de *Framework* Gamificada e na atividade de investigação Avaliar, identificou-se a *Framework* Concetual de Gamificação. No resultado de investigação *Método*, foi definido para a atividade de investigação Construir e Avaliar, o Processo de Gamificação. No resultado de investigação *Instanciação*, foi definido para as atividades Construir e Avaliar, o Curso Gamificado “*SmartMOOC*”. Nos resultados de investigação associados às Ciências Naturais, foram definidos para as *Dimensões* e atividade de investigação Justificar o *Flow* e o *CANOE*. Nos resultados de investigação *Modelo* das ciências naturais foi identificado para a atividade Teorizar o Modelo de Adoção da Tecnologia e na atividade de investigação *Justificar*, o Modelo de Adoção do Sistema Gamificado.

Tabela 3 - *Framework* de Investigação (March & Smith, 1995)

		Atividades de Investigação			
		Ciências do Design		Ciências Naturais	
Resultados da Investigação		Construir	Avaliar	Teorizar	Justificar
	Dimensões				<i>Flow</i> <i>CANOE</i>
	Modelo	Proposta de <i>framework</i> de Gamificação	<i>Framework</i> de Gamificação	Modelo de Adoção da Tecnologia	Modelo da Adoção do Sistema Gamificado
	Método	Processo de Gamificação			
	Instanciação	Curso Gamificado SmartMOOC			

O desenvolvimento da presente tese suporta-se, também na taxonomia de métodos de investigação, proposta por Pertti Järvinen (2004). Esta taxonomia foi desenvolvida tendo por base o princípio de *top-down*, ou seja, todas as abordagens de investigação são primeiramente divididas em duas classes. Jarvinen define a taxonomia a partir de duas abordagens de investigação: as abordagens que estudam a realidade e as abordagens matemáticas. Relativamente às abordagens que estudam a realidade, derivam duas classes, uma referindo-se a questões de investigação sobre do que é (parte de) a realidade e uma outra, focada na utilidade de um artefacto (algo feito por seres humanos). A taxonomia diferencia ainda as abordagens concetuais-analíticas e as abordagens para estudos empíricos. Segundo Järvinen, quando o passado e o presente são empiricamente estudados, diferenciam-se os métodos Testar Teoria, do Criar Teoria, dependendo de ser uma teoria, um modelo ou uma *framework* para orientar um estudo, ou se é uma investigação que desenvolve uma nova teoria suportada em novos dados. Relativamente aos artefactos o autor propõe uma diferenciação entre construir e avaliar (Järvinen, 2004).

Com base na taxonomia de Järvinen na Figura 2, descrevem-se os métodos de investigação que orientaram e guiaram o desenvolvimento da tese.

Após a identificação das atividades de investigação e dos resultados associados definiram-se os métodos e técnicas de investigação para cada objetivo proposto. Na Tabela 4, sintetizam-se os objetivos de investigação, os estudos que foram realizados para responder aos objetivos e as respetivas técnicas e métodos que foram utilizados.

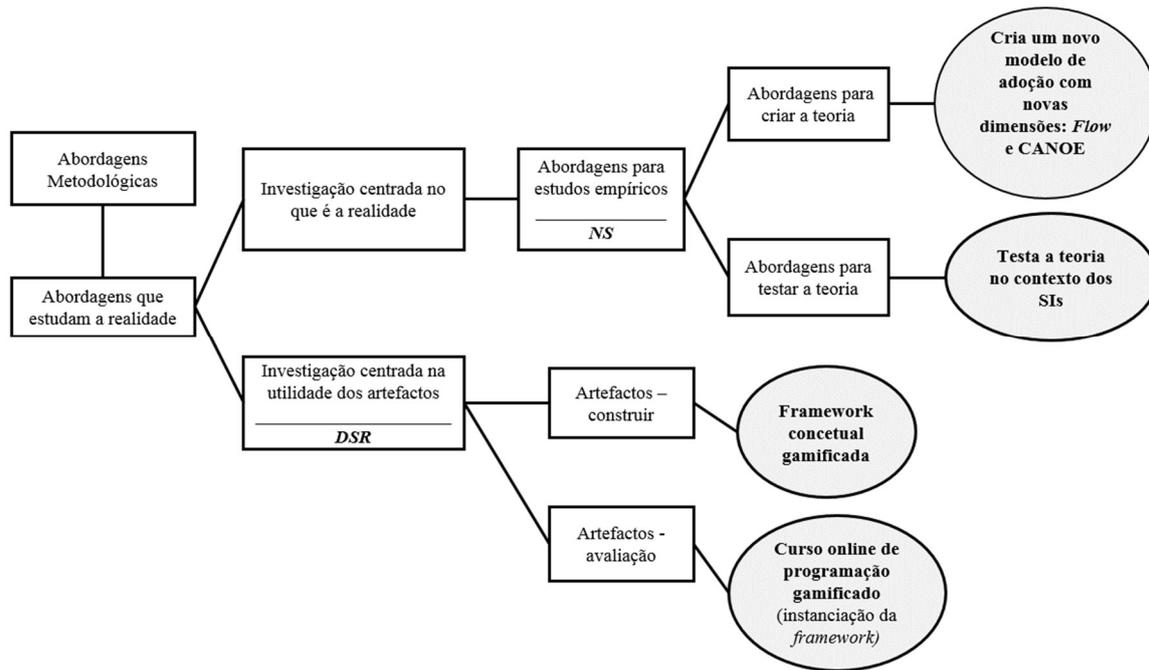


Figura 2 - Taxonomia de Métodos de Investigação, adaptado de Järvinen (2004)

Tabela 4 – Estudos, Métodos e Técnicas de Investigação

Objetivos de Investigação	Estudos				Artefacto	Método/Técnica		
	1	2	3	4		Instrumento Questionário	Quantitativo	RL
OBJ-1: Identificação das dificuldades	✓	✓				✓	✓	✓
OBJ-2: Soluções								✓
OBJ-3: <i>Framework</i>			✓		✓			✓
OBJ-4: Implementação			✓		✓			
OBJ-5: Avaliação				✓	✓	✓	✓	✓

Nota: OBJ – Objetivos de investigação; RL – Revisão da literatura

O desenvolvimento da presente tese focou-se em duas metodologias: *Design Science Research* (DSR) e *Natural Science* (NS). Seguindo a metodologia do *Design Science Research*, efetuou-se a identificação dos problemas na aprendizagem da programação. Após essa identificação foi proposta a *framework* e a sua implementação, tendo sido aplicado na implementação o método de gamificação definido para o curso *online*. Na avaliação do artefacto, utilizou-se também a metodologia de investigação *Natural Science*. A partir da literatura formulou-se um conjunto de hipóteses que foram testadas a partir de um modelo de adoção da gamificação, permitindo assim a validação do modelo de adoção proposto e do artefacto. Exemplificam-se na Figura 3,

as metodologias utilizadas e as atividades de investigação associadas a cada uma das metodologias.

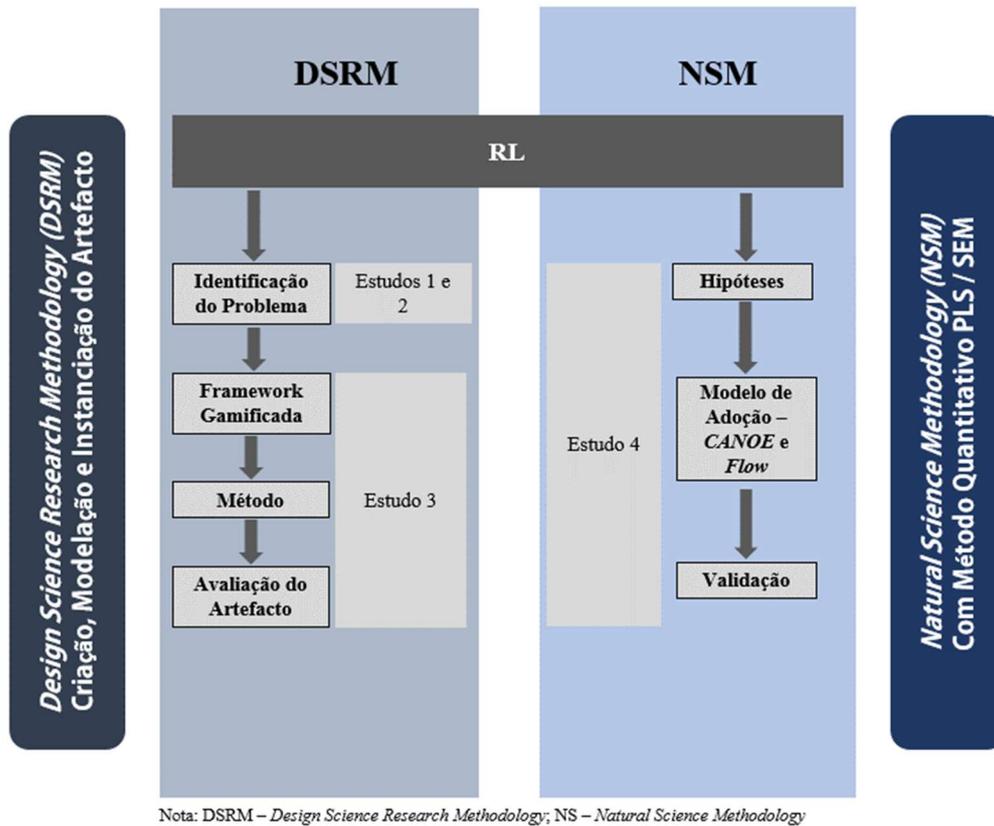


Figura 3 - Estudos e Metodologias

1.6 Contribuições

A presente tese faz contribuições teóricas e contribuições práticas. As principais contribuições são a *framework* concetual para cursos *online* gamificados e o modelo de adoção da gamificação para cursos online gamificados. Relativamente às contribuições teóricas, aprofunda-se o conhecimento científico referente às dificuldades na aprendizagem da programação. Através do estudo das perceções das dificuldades na aprendizagem da programação e a comparação com os resultados obtidos nas provas de avaliação, reforçámos o conhecimento dessas dificuldades. Esse conhecimento será útil aos educadores da área da informática, permitindo que a partir das dificuldades encontradas, sejam estabelecidos mecanismos que ajudem os estudantes a ultrapassar essas mesmas dificuldades. Por outro lado, e tendo em consideração que são necessárias soluções que contribuam para mitigar os problemas e ajudar os estudantes a

ultrapassar esses problemas, propõe-se uma framework conceitual gamificada para cursos *online*. Essa framework permite operacionalizar cursos *online* gamificados, tanto no contexto da programação como em outros contextos, sendo esta uma inovação, dado não existir na literatura, uma framework que operacionalize a gamificação em cursos *online*. De forma a explicar a influência que o estado de absorção experienciado pelos estudantes (*flow*) e as suas características individuais (CANOE) têm na adoção da gamificação em cursos *online*, propomos um novo modelo teórico de adoção da gamificação, contribuindo assim para a explicação teórica do impacto do *flow* e do CANOE na adoção da gamificação em cursos online gamificados de aprendizagem da programação.

Como implicações práticas, os resultados da tese são um contributo para os educadores, de forma a facilitar e a incentivar a adoção da gamificação na aprendizagem da programação. Devem também ser consideradas no desenho e na implementação dos cursos online gamificados as dimensões identificadas na presente tese.

1.7 Estrutura da tese

A tese encontra-se estruturada em sete capítulos. No Capítulo 1, sendo o presente capítulo, efetua-se a apresentação da temática e da justificação para a investigação que se pretende realizar. Identificam-se também as questões de partida, a abordagem metodológica adotada com a definição dos objetivos de investigação e a metodologia de investigação que orientou o desenvolvimento da tese. Por último, apresentam-se as contribuições da tese.

O Capítulo 2 é constituído por cinco subcapítulos, onde são abordadas as temáticas da investigação: problemas na aprendizagem da programação; soluções propostas para a aprendizagem da programação; gamificação; adoção da tecnologia. Finaliza-se o capítulo com a conclusão.

No Capítulo 3, descreve-se o estudo empírico realizado através do qual foram identificadas as dificuldades na aprendizagem da programação percecionadas por estudantes e as dificuldades percecionadas pelos professores relativamente aos estudantes. As dificuldades são descritas em dois grupos: situações de aprendizagem e conceitos de programação. Identificou-se também a utilidade percecionada referente a contextos de aprendizagem e a materiais/recursos, bem como a atitude dos estudantes face à programação. Posteriormente apresentam-se os resultados e respetiva conclusão.

No Capítulo 4, é descrito o estudo empírico que compara as percepções entre professores e estudantes relativamente aos conceitos de programação, situações de aprendizagem, contextos de aprendizagem e materiais/recursos. Adicionalmente neste estudo, analisam-se e comparam-se os resultados obtidos nas provas de avaliação (exames e testes) com as dificuldades percebidas pelos professores e estudantes. Finaliza-se o capítulo com a discussão de resultados e conclusões.

No Capítulo 5, é proposto uma framework teórica-concetual de gamificação para implementação da gamificação em cursos *online* de programação. Apresenta-se a abordagem metodológica seguida e descrevem-se as várias dimensões que compõem a framework. Descreve-se ainda a operacionalização da framework e respetiva validação.

No capítulo 6, é proposto um novo modelo teórico-concetual explicativo da adoção da gamificação em cursos *online* gamificados.

Finaliza-se o documento com o Capítulo 7, onde são apresentadas as conclusões gerais organizadas por objetivos de investigação, limitações à investigação, implicações teóricas e práticas, e trabalho futuro.

2 REVISÃO CRÍTICA DA LITERATURA

2.1 Introdução

Neste capítulo, é apresentada a revisão da literatura, que se traduz em apresentar, enquadrar e contextualizar, as temáticas e conceitos abordadas nesta tese. Identificaram-se na literatura os problemas na aprendizagem da programação e as soluções propostas para mitigação dos problemas na aprendizagem da programação. São igualmente apresentados os conceitos de gamificação, de adoção de tecnologia e os modelos teóricos explicativos referentes ao comportamento e adoção da tecnologia.

A revisão da literatura, seguiu as boas práticas na seleção dos artigos mais relevantes, publicados nas principais revistas científicas da área. Estes foram complementados com estudos realizados pelos membros da equipa de investigação.

2.2 Problemas na aprendizagem da programação

Neste subcapítulo é descrita a importância no contexto atual da programação, assim como os problemas que os estudantes enfrentam no processo de aprendizagem da programação de computadores.

A procura por competências de programação está a aumentar a uma escala global. Projeções recentes do *United States of Labour Statistics* (2017), sugerem um crescimento nas carreiras ligadas às áreas de informática até 2024 e a procura pelas competências tecnológicas continuará a crescer no futuro. De forma a dar resposta às necessidades do mercado de trabalho, diversos governos pelo mundo inteiro estão a introduzir a programação nos primeiros anos de aprendizagem e a desenvolver campanhas direcionadas aos estudantes que pretendam ingressar na universidade, para que optem por formações relacionadas com a tecnologia, de forma a prepará-los para o trabalho na economia digital. Saber programar é uma das competências mais procuradas e valorizadas no mercado de trabalho atualmente (Franklin, 2015; Yadav et al., 2017). Contudo, a aprendizagem da programação de computadores é uma das áreas de conhecimento que acarreta maior dificuldade de aprendizagem e se revela desafiante para os estudantes, principalmente para os que estão a iniciar a aprendizagem nesta área (Lahtinen et

al., 2005; Milne & Rowe, 2002; Mow, 2008). Esta dificuldade é expressa nas elevadas taxas de desistência ou de retenção que os cursos da área da informática registam todos os anos (Bergin & Reilly, 2005; Watson & Li, 2014). As taxas de retenção e de desistência são uma evidência dessas dificuldades Bergin & Reilly (2005) referem mesmo que *“It is well known in the Computer Science Education (CSE) community that students have difficulty with programming courses and this can result in high drop-out and failure rates. (p. 293).”*

Diversos trabalhos de investigação têm sido realizados focando-se na identificação dos vários fatores envolvidos na aprendizagem da programação, sendo um contributo relevante para a área. Esses trabalhos têm identificado e reportado os mais diversos problemas na aprendizagem da programação e.g. Berglund & Lister (2010); Eckerdal (2009); Lister et al., (2004) Ma (2007); Milne & Rowe (2002); Mow (2008); Pappas et al., (2016); Rahmat et al., (2012); Robins et al., (2003); Winslow (1996). Por exemplo, a identificação dos modelos mentais dos estudantes iniciantes (Ma, 2007); proposta de soluções baseadas em ferramentas educacionais, (Eckerdal & Thuné, 2005; Lister et al., 2004); identificação do impacto do tipo de linguagem de programação que é ensinada (Winslow, 1996), entre outros. Tendo por base a revisão da literatura efetuada, descrevem-se de seguida alguns dos estudos realizados.

Milne & Rowe (2002) realizaram um trabalho de investigação que teve como objetivo a identificação das dificuldades na aprendizagem da programação na perspetiva dos estudantes e dos professores. Os autores centraram o seu estudo na identificação das dificuldades referentes aos conceitos de programação. Os conceitos identificados pelos autores como sendo os mais difíceis, foram os ponteiros e a alocação dinâmica da memória. Acreditam os autores, que a deficiente compreensão dos estudantes relativamente ao modo como a memória do computador funciona após a execução dos programas, é um dos motivos para bloquearem na compreensão total dos conceitos. Esta situação, apenas poderá ser ultrapassada quando os estudantes adquirirem um modelo mental adequado, relativamente à forma como os programas funcionam, do que é guardado na memória e de como os objetos se relacionam entre eles na memória.

Resultados similares foram alcançados pelos autores Lathinen et. al. (2005), que conduziram um trabalho de investigação para identificar as dificuldades na aprendizagem da programação, no qual os conceitos considerados mais difíceis foram também os ponteiros e alocação da memória. Referem os autores que os conceitos abstratos são normalmente aqueles que apresentam maiores dificuldades para os estudantes na sua aprendizagem. Estes autores chegaram à conclusão que a prática é um dos fatores que poderá contribuir para uma melhor

compreensão destes conceitos, entre outros fatores (Lahtinen et al., 2005). Estudantes e professores consideram que as aprendizagens práticas da programação são as mais úteis. Argumentam os autores que quanto mais concretas as situações de aprendizagem forem, mais se aprende. Os estudantes até podem compreender os conceitos individualmente, mas apresentam dificuldades quando têm que os aplicar em conjunto. Dado que muitas das dificuldades estão relacionadas com questões mais avançadas e não propriamente com os conceitos em si os autores Lahtinen et al., (2005), referem que materiais/recursos e situações de aprendizagem, deveriam estar mais orientados para as competências de desenvolvimento, modificação e depuração do programa.

Robins et. al. (2003) conduziram um trabalho de revisão da literatura e discussão sobre o ensino e a aprendizagem da programação onde sugerem que o foco não deve ser no professor, mas sim na aprendizagem do estudante, através de uma efetiva comunicação entre estudante e professor, com o objetivo de aprofundar os princípios da aprendizagem e as competências, criando assim estudantes independentes e refletivos, proporcionando-lhes objetivos claros, estimulando-lhes o interesse, envolvendo-os na aprendizagem através do material/recurso do curso, com avaliações apropriadas e retorno das suas ações. Refere o autor em consonância com Winslow (1996), que a boa pedagogia requer que o professor mantenha os factos iniciais, modelos e regras simples, apenas expandindo e refinando as regras, à medida que o estudante for adquirindo experiência.

Schulte et. al. (2010) referem no seu trabalho de investigação, que tarefas que suportem o desenvolvimento de um modelo de domínio são necessárias. Atividades de aprendizagem que simplesmente envolvam a leitura de exemplos de código para recordação ou para resumir ou documentar, não ajudam os estudantes a desenvolver modelos de domínio. Em contrapartida, os autores referem que as atividades que envolvam modificar ou reutilizar o exemplo para atender a diferentes necessidades de aprendizagem podem ser particularmente úteis para os estudantes desenvolverem um modelo de domínio (Schulte et al., 2010).

Seguidamente resumem-se na Tabela 5, os problemas na aprendizagem da programação revistos na literatura.

Tabela 5 - Problemas na Aprendizagem da Programação, adaptado de (Costa, Aparicio, & Cordeiro, 2012)

	Menor Experiência	Novidade	Muitas Competências	Desenhar o Programa	Compreensão do Programa	Tipo de Linguagem de Programação	Duração do curso – tempo	Diferentes Capacidades	Atitudes Face à Programação	Fatores Cognitivos	Autoconfiança	Motivação	Modelos Mentais Ineficientes
(Wilson & Shrock, 2001)				✓				✓					✓
(Byrne & Lyons, 2001)	✓								✓		✓		
(Jenkins, 2002)		✓	✓	✓	✓	✓			✓	✓			
(Milne & Rowe, 2002)			✓		✓		✓			✓			
(Robins et al., 2003)	✓		✓	✓	✓		✓	✓	✓	✓			
(Lahtinen et al., 2005)			✓	✓	✓		✓	✓		✓			
(Ala-Mutka, 2005)			✓	✓	✓	✓		✓		✓			
(Gomes & Mendes, 2007)			✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
(Ma, 2007)	✓		✓									✓	✓
(Mow, 2008)			✓	✓	✓		✓	✓	✓	✓			✓
(Radošević, Orehovački, & Lovrenčić, 2009)			✓	✓	✓		✓	✓	✓	✓			
(Renumul et al., 2010)			✓	✓	✓		✓	✓	✓	✓			
(Kordaki, 2010)	✓		✓	✓	✓	✓	✓	✓					
(Rahmat et al., 2012)	✓		✓									✓	✓
(Álvarez & Larrañaga, 2013)												✓	
(Salleh et al., 2013)							✓					✓	
(Konecki, 2014)	✓											✓	
(Rudolph, 2015)	✓	✓	✓								✓	✓	✓
(Bouvier et al., 2016)			✓	✓	✓			✓					
(Chao, 2016)				✓	✓							✓	
(Malik & Coldwell-Neilson, 2017)					✓				✓	✓	✓		

2.3 Soluções propostas de suporte à aprendizagem da programação

Na literatura relacionada com o ensino da informática, identificaram-se diversos trabalhos de investigação que descrevem o uso de ferramentas, ambientes, linguagens de programação, artefactos tangíveis, jogos educacionais digitais e abordagens que integram características de jogos. Essas soluções propostas têm como objetivo incentivar, motivar e estimular o estudante para a aprendizagem da programação. Nos diversos trabalhos, observa-se que todos os autores

reportam a importância de providenciar aos estudantes iniciantes em programação de computadores, ambientes que sejam confortáveis e não intimidantes, que motivem o estudante para a aprendizagem, para que possam assim colocar questões e melhorar as suas competências em programação. Em resultado da revisão da literatura realizada, apresenta-se seguidamente uma síntese das soluções revistas.

2.3.1 Ferramentas de visualização gráfica

Neste subcapítulo é efetuada uma síntese das ferramentas de visualização gráfica revistas na literatura. Estas ferramentas utilizam uma abordagem baseada em metáforas (micromundos) e permitem ao estudante através de um ambiente gráfico, criar, definir comportamentos e interagir com os objetos criados.

2.3.1.1 Alice

A ferramenta Alice nasceu na Universidade da Virgínia em 1995 (Ty et al., 1995) como protótipo de uma aplicação para experiências de realidade virtual, com o objetivo de ser acessível por não programadores e contribuir para o avanço da realidade virtual (Tabela 6). Posteriormente, os criadores derivaram esta ferramenta para o desenvolvimento de gráficos 3D e incorporaram a codificação *drag and drop* (Cooper, Dann, & Paush, 2000). Desde então, a ferramenta gráfica 3D tem evoluído e integrado diversas componentes. A ferramenta combina narrativa, linguagem por camadas e ferramenta visual de programação. O ambiente utiliza objetos tridimensionais e está direcionado para a programação orientada por objetos, tais como: objetos, classes, herança, expressões, condições, ciclos, variáveis, tabelas, eventos e recursividade. A construção dos programas é feita com recurso a segmentos de código através do *drag and drop*, em vez de ser escrito o segmento. Este tipo de programação despreocupa os estudantes com a sintaxe da linguagem. A ferramenta oferece um ambiente a partir do qual os utilizadores podem selecionar o *template*, os objetos, alterar a propriedade dos objetos, programar as ações dos objetos e manipular uma câmara para criar a história ou um jogo interativo. Atualmente, existem duas versões para *download*: Alice 2 e Alice 3.

A versão Alice 2, adequa-se mais aos estudantes do ensino médio e estudantes iniciantes do ensino superior, sendo este ambiente menos complexo de utilizar. A versão Alice 3 é uma versão mais avançada e destina-se aos estudantes que pretendam expandir a interface *drag and drop*, e trabalhar diretamente com o código Java. Os autores Cooper, Dann & Pausch (2003) e Dann, Cosgrove, Slater, Culyba & Cooper (2012) estudaram e utilizaram esta ferramenta como suporte à aprendizagem da programação. Esta ferramenta pode ser obtida em

<http://www.alice.org> e pode ser utilizada de forma gratuita. É suportada por vários *sponsors*, entre os quais a Universidade Carnegie Mellon e a Oracle.

2.3.1.2 GreenFoot

A ferramenta *Greenfoot* disponibiliza uma representação gráfica do código criado (Kölling & Henriksen, 2004). Os estudantes escrevem instruções de código Java de modo a programar as ações dos objetos a duas dimensões, aprendendo desta forma os conceitos de programação orientada a objetos. Esta ferramenta contribui para a aprendizagem dos estudantes relativamente a objetos, instruções, expressões, condições, ciclos, variáveis, tabelas e eventos. Esta ferramenta pode ser obtida gratuitamente através do endereço (<http://www.greenfoot.org>). Neste endereço são disponibilizados um conjunto diversificado de tutoriais, vídeos, entre outros, que ajudam na iniciação ao ambiente de desenvolvimento. A ferramenta encontra-se disponível para instalação nos sistemas operativos Windows, Mac e Linux. Esta ferramenta foi utilizada num estudo de investigação conduzido por Vilner, Zur & Tavor (2011), que integrou a ferramenta numa aula de herança. Os estudantes foram inquiridos sobre a ferramenta e 325 responderam que tinham gostado de utilizar a ferramenta, a qual os ajudou na compreensão do conceito de herança.

2.3.1.3 Jeroo

A ferramenta *Jeroo* pretende ser um contributo para a aprendizagem dos conceitos de programação: objetos, instruções, expressões, condições, ciclos e variáveis (Tabela 6). Esta ferramenta tem por base uma metáfora baseada num animal australiano conhecido por *jeroo*, animal semelhante a um canguru. O objetivo é mover o animal pela ilha e evitar a água ou as armadilhas e colecionar recursos. Os estudantes são convidados a programar os seus *jeroos* com recurso às linguagens de programação: C++, C#, Java ou Python. Esta ferramenta pode ser utilizada sem qualquer custo e existem no *website* (<http://home.cc.gatech.edu/dorn/38>) uma variedade de recursos que ajudam a utilizar a ferramenta. Os autores Sanders & Dorn (2003), realizaram um estudo de investigação para medirem a confiança dos estudantes na programação antes e após a utilização da ferramenta. Após a obtenção dos dados e com recurso ao teste *t* concluíram que o aumento da confiança era estatisticamente significativo.

2.3.1.4 Scratch

O *Scratch* foi desenvolvido pelo MIT Media Lab no ano de 2007 e disponibiliza um ambiente de narrativa de *drag and drop* que permite aos estudantes a criação de programas através da

manipulação de blocos gráficos de código (Tabela 6). Esta ferramenta permite a aprendizagem dos conceitos relativos a objetos, instruções, expressões, condições, ciclos, variáveis, *threads* e eventos. Pode ser obtido gratuitamente a partir do endereço (<https://scratch.mit.edu/>) e disponibiliza um conjunto diverso de recursos. O *Scratch* foi recentemente indicado pelo Ministério da Educação de Portugal, como ferramenta a adotar num projeto amplo de iniciação à programação nos primeiros anos de ensino básico. O projeto EduScratch visa promover a utilização educativa do *Scratch* através do apoio, formação e partilha de experiências na comunidade educativa (WebSite EduScratch, 2017). Daly (2009) conduziu um estudo para medir a perceção dos estudantes sobre a utilização da ferramenta e os resultados demonstraram que a sua utilização foi considerada uma experiência de aprendizagem positiva e divertida de usar.

Para além das ferramentas descritas anteriormente, diversas outras foram propostas e revistas na literatura, e.g. Raptor (Carlisle, Wilson, Humphries, & Hadfield, 2004), acessível a partir do endereço: <http://raptor.martincarlisle.com/>; Jeliot (Moreno, Myller, Sutinen, & Ben-Ari, 2004), acessível a partir do endereço: <https://cs.joensuu.fi/jeliot/index.php>; Verificator (Radošević et al., 2009); Jype (Helminen, 2009); Tracka2 (Nikander et al., 2004); Ville (Rajala, Laakso, Kaila, & Salakoski, 2007), acessível a partir do endereço: <http://ville.cs.utu.fi/old/> e UUhistle (Sorva & Sirkiä, 2010), acessível a partir do endereço: <http://www.uuhistle.org/index.php>.

2.3.2 Artefactos tangíveis

As abordagens utilizadas para ensinar os conceitos de programação com recurso a sistemas físicos do mundo real, têm tido recentemente um crescente interesse e a sua utilização em diversos contextos, tem sido amplamente documentada (Álvarez & Larrañaga, 2013; Arlegui, Menegatti, Moro, & Pina, 2008; Blank, Kumar, Meeden, & Yanco, 2003; Frangou, Papanikolaou, & Aravecchia, 2008; McGill, 2012b). A utilização dos micros robots como suporte à aprendizagem, são um dos exemplos dessas abordagens. Descrevem-se de seguida, alguns dos exemplos reportados na literatura.

McGill (2012b) conduziu um estudo que teve como objetivo estudar os efeitos motivacionais gerados pela utilização de robots físicos na aprendizagem da programação em estudantes que não pertenciam a cursos de informática. O autor utilizou para esse efeito um robot desenhado com fins educacionais, pelo *Intitute for Personal Robots in Education* (IPRE).

Também McWhorter (2008), conduziu um estudo com o objetivo de identificar a efetividade da utilização de atividades robóticas através dos *Lego MindStorms*. O sistema *LEGO* é um brinquedo cujo conceito se baseia em partes que se encaixam permitindo muitas combinações. O *Lego Mindstorms* é uma linha do brinquedo *LEGO*, direcionada para a educação tecnológica (Kim & Jeon, 2009). McWhorter (2008) analisou em que medida as atividades realizadas com os *Lego MindStorms*, influenciaram a aprendizagem autorregulada na disciplina de introdução à programação na *University of North Texas*. De acordo com o autor, a utilização de artefactos tangíveis na aprendizagem de programação, oferece significantes vantagens. Por exemplo, a visualização permite especificar o problema mais facilmente. Observar o artefacto, como por exemplo, um robot num ambiente físico, torna a correção de erros mais fácil. A visualização realça um dos processos do cérebro humano mais importante, que é a entrada visual. Contribui também para facilitar a aprendizagem da programação, é divertido e um dos maiores contributos é na pedagogia (McWhorter, 2008).

Brauner et. al. (2010) conduziram um estudo que teve como objetivo comparar o efeito da utilização de robots tangíveis, com a utilização de ferramentas de visualização numa disciplina de introdução à programação. Os resultados mostraram que a utilização dos robots é benéfica para a aprendizagem.

Apesar dos benefícios da utilização dos robots na aprendizagem da programação, um dos aspetos referidos na revisão da literatura como menos positivo, é a necessidade na compreensão de aspetos técnicos relacionados com a robótica, desviando a atenção do principal: a aprendizagem da programação. Com a necessidade dessa compreensão prévia os estudantes que não se sintam motivados para a aprendizagem desses aspetos técnicos, podem desinteressar-se pela aprendizagem que interessa: aprender a programar (Blank et al., 2003; Braught, 2012; Kumar, 2004). Tendo por base os constrangimentos técnicos Braught (2012), propõe uma biblioteca para múltiplas plataformas como meio que permita manipular e aceder ao *hardware* do robot, através de sensores e atuadores, tornando assim mais fácil a interação com o curso e diminuindo o tempo e a necessidade de compreensão dos detalhes robóticos, não criando obstáculos que prejudiquem o curso. Contudo, esta biblioteca proposta pelo autor destina-se a ser utilizada não numa fase inicial da aprendizagem da programação, mas sim já numa fase em que é suposto os estudantes terem adquirido os conhecimentos básicos sobre os principais mecanismos de programação.

2.3.3 Jogos educacionais digitais

Neste subcapítulo, descrevem-se o conceito e alguns dos jogos educacionais digitais utilizados no contexto da aprendizagem da programação.

O conceito de jogos educacionais digitais é normalmente utilizado, para descrever os jogos de computador que são utilizados como ferramentas de suporte à aprendizagem e que fornecem atividades interativas e atrativas para os estudantes. Pretende-se com os jogos educacionais, reforçar a motivação intrínseca dos estudantes através da incorporação do desafio; despertar a curiosidade; estimular a imaginação. Através dos jogos, os estudantes podem alcançar determinados objetivos e visualizar de imediato os resultados alcançados, sendo este um processo que estimula a sua autoconfiança e os ajuda a ter mais confiança nas suas competências (Malliarakis, Satratzemi, & Xinogalos, 2014).

Por exemplo, Barnes, Powell, Chaffin & Lipford (2008) propuseram um ambiente baseado em jogo com o objetivo de envolver e motivar os estudantes principiantes na aprendizagem da programação. Os autores avaliaram o retorno e o desempenho dos estudantes nos protótipos iniciais concluindo que os mesmos se divertem a aprender com o jogo e que as recompensas incorporadas no jogo, são vitais para a sua motivação durante o processo de aprendizagem. Eagle e Barnes (2009) propuseram um jogo “*Wu’s Castle*” com o objetivo de melhorar a aprendizagem e as atitudes dos estudantes face aos cursos de informática e particularmente, à programação. Neste jogo, os estudantes são desafiados a programar ciclos e tabelas num ambiente interativo e num modo visual. O jogo introduz retorno imediato aos estudantes e permite a execução e visualização do código num ambiente seguro. Os autores avaliaram o jogo com um grupo de estudantes que utilizaram o jogo e com um grupo de controlo. Os resultados demonstraram que os estudantes que utilizaram o jogo tinham uma compreensão mais profunda e robusta dos conceitos de programação.

Piteira & Haddad (2011) propuseram um protótipo de jogo “*The Cube Game*” com o objetivo de motivar e envolver os estudantes na aprendizagem da programação (Tabela 6). O jogo proposto é baseado na web, de forma a permitir aos estudantes o acesso a qualquer hora e a partir de qualquer lugar. Pretende-se assim facilitar aos estudantes o acesso e a realização da sua aprendizagem através do jogo, sem restrições temporais e de espaço. Por outro lado, pretende-se igualmente registar e analisar a interação e os resultados dos estudantes, tendo sido escolhida para arquitetura do jogo, a arquitetura Web com um suporte de base de dados. Os conceitos de programação considerados para a implementação do jogo, foram os seguintes:

variáveis, operadores aritméticos, operadores lógicos, operados booleanos, instruções sequenciais, instruções de repetição, estruturas de seleção e funções. As características de jogos incorporados foram: desafio, colaboração e personalização.

Tabela 6 – Problemas e Soluções

Soluções	Problemas na Aprendizagem da Programação									
	Desenhar o Programa	Compreensão do Programa	Tipo de Linguagem de Programação	Diferentes Capacidades	Atitudes Face à Programação	Fatores Cognitivos	Autoconfiança	Motivação	Modelos Mentais Ineficientes	
Alice (Cooper et al., 2000)	✓	✓		✓					✓	
Jeliot (Moreno et al., 2004)	✓	✓			✓	✓		✓	✓	
Tracka2 (Nikander et al., 2004)	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Greenfoot (Henriksen & Kölling, 2004; Kölling, 2010)										
Raptor (Carlisle et al., 2004)	✓	✓	✓	✓	✓	✓	✓	✓	✓	
GP-Robocode (Shichel, Ziserman, & Sipper, 2005)					✓		✓	✓		
Object Karel (Xinogalos, Satratzemi, & Dagdilelis, 2006)	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Ville (Rajala et al., 2007)	✓	✓	✓		✓	✓	✓	✓	✓	
Scratch (MIT Media Lab, 2007)	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Verificador (Radošević et al., 2009)	✓	✓		✓	✓	✓	✓	✓		
Wu's Castle (Eagle & Barnes, 2009)						✓	✓	✓		
Jype (Helminen, 2009; Helminen & Malmi, 2010)	✓	✓	✓		✓	✓	✓	✓	✓	
EleMental (Chaffin, Doran, Hicks, & Barnes, 2009)	✓	✓					✓	✓		
Solução Robótica (McWhorter, 2008)	✓	✓			✓		✓	✓		
Uuhistle (Sorva & Sirkiä, 2010)	✓	✓			✓	✓			✓	
The Cube Game (Piteira & Haddad, 2011)	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Mit App Inventor (Wolber, 2011)	✓	✓	✓		✓		✓	✓		
Solução Robótica (McGill, 2012a)		✓		✓	✓	✓	✓	✓	✓	
Program Your Robot (Kazimoglu, Kiernan, Bacon, & Mackinnon, 2012)	✓				✓		✓	✓		
Solução Robótica (Álvarez & Larrañaga, 2013)								✓		
CMX (Malliarakis et al., 2014)	✓				✓		✓	✓		
Robot On! (Miljanovic & Bradbury, 2016)	✓	✓	✓		✓		✓	✓	✓	
Serious Game (Liu et al., 2016)	✓				✓			✓		
Ambiente de Programação Visual (Chao, 2016)	✓	✓						✓		

Chaffin et. al. (2009) propõem um jogo tridimensional com o nome *EleMental: The Recurrence*, que tem como objetivo ensinar os estudantes na execução da recursividade. O jogo utiliza dois avatares de nome *Ele* e *Cera*, os quais orientam os estudantes durante a sua interação com o jogo.

Mit App Inventor (Wolber, 2011) é uma ferramenta que foi criada em resultado de um projeto-piloto do Google e atualmente mantido pelo Massachusetts Institute of Technology (MIT). Esta ferramenta é *cloud based* e tem como objetivo o desenvolvimento de aplicações funcionais para dispositivos Android (<http://appinventor.mit.edu/explore/>). Permite a criação de programas de uma forma interativa com recurso à técnica de *drag and drop*. É utilizado o pseudocódigo, que é uma forma genérica de escrever um algoritmo, utilizando uma linguagem simples, sem necessidade de conhecer a sintaxe de nenhuma linguagem de programação, em alternativa a uma linguagem de programação. Para a criação das aplicações a ferramenta disponibiliza blocos de código com significado, tais como ciclos de repetição, ciclos de seleção, variáveis, entre outros conceitos. Diversos estudos de investigação utilizaram esta ferramenta como suporte à aprendizagem da programação (Hsu & Ching, 2013; Karakus, Uludag, Guler, Turner, & Ugur, 2012). Efetuada a identificação dos problemas e das soluções, na Tabela 6 resumiram-se as soluções revistas na literatura para cada problema na aprendizagem da programação previamente identificado no subcapítulo 2.2.

2.3.4 Gamificação

Neste subcapítulo, é descrito o conceito de gamificação, a sua origem, a sua aplicação em contexto educacional, com particular ênfase na aprendizagem da programação.

O conceito de gamificação começou a ser utilizado de uma forma mais generalizada no ano de 2010, quando diversos jogadores e oradores da área dos jogos em conferências, o popularizaram (Deterding et al., 2011). Na literatura encontramos referências a ludificação e.g. (Costa, 2012) como tradução do termo *gamification*, contudo o termo gamificação proliferou e é largamente utilizado e aceite pela comunidade científica. Nesse sentido e no âmbito desta tese, adota-se o termo gamificação.

A gamificação é o processo de incorporar elementos de jogo em contexto de não jogo (Deterding et al., 2011). Os elementos de jogo caracterizam-se pela utilização de pontos, medalhas, níveis, barras de progresso, quadro de honra, moeda virtual, avatares, entre outros e as implementações comuns de gamificação, aplicam estes elementos em contexto de aprendizagem (Deterding et al., 2011). Apesar do conceito de gamificação ter sido definido por Deterding, persistem ainda diversas interpretações relativamente ao mesmo. Assim, a gamificação não é um jogo ou um processo de transformar em jogo, mas sim a utilização de abstrações e metáforas originárias da cultura dos videojogos, em áreas que por norma não estão associadas a videojogos (Deterding et al., 2011). Esta definição é importante para a

compreensão da utilização da gamificação na educação e a sua diferenciação da utilização dos videojogos na educação, nomeadamente jogos digitais educacionais e aprendizagem baseada em jogos (Dichev & Dicheva, 2017) . Diversos estudos têm definido e centrado o conceito de gamificação e um desses estudos definiu o conceito com recurso a duas dimensões de *playing/gaming* e *parts/whole* (Deterding et al., 2011). A dimensão jogar/brincar (*playing/gaming*) foi utilizada para destacar a diferença entre o design lúdico (*playful design*) e brinquedos (*toys*) e a dimensão parte/todo (*parts/whole*) foi utilizada para distinguir os jogos ou *serious games* que estão presentes na gamificação. A Figura 4 ilustra o modo como a gamificação é definida tendo por base essas dimensões.

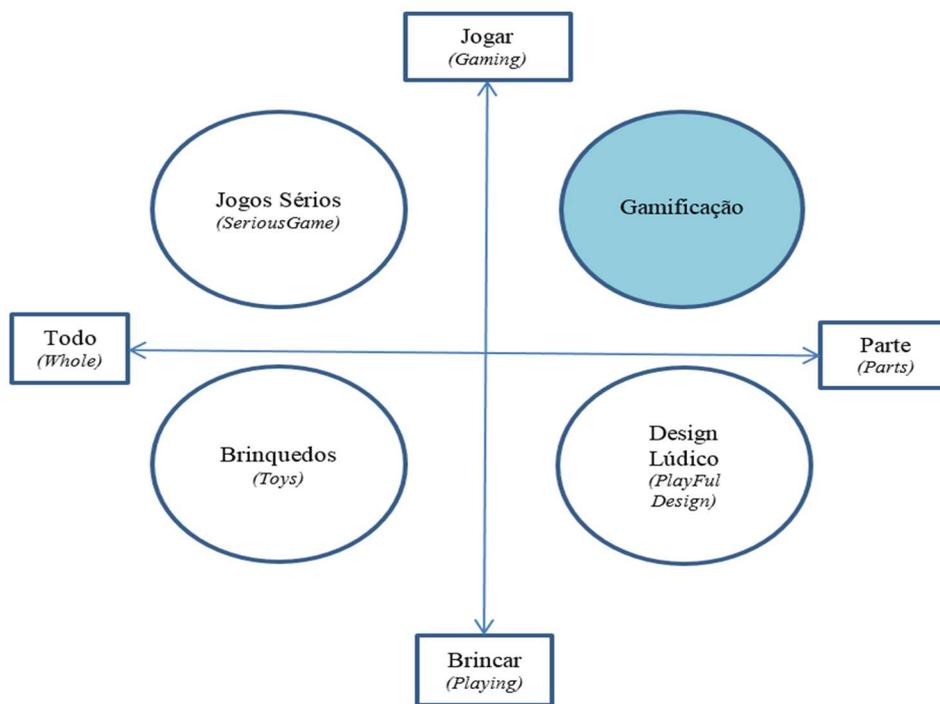


Figura 4 - Gamificação Entre o Jogo e Jogar, o Todo e as Partes (Deterding et al., 2011)

Vamos compreender um pouco melhor o que são os elementos de jogo. Os elementos de jogo são característicos dos jogos. Estes elementos são significativos para a jogabilidade e estão associados à maioria dos jogos de entretenimento.

Werbach & Hunter (2012) propuseram uma *framework* (Figura 5), na qual identificaram três tipos de elementos, nomeadamente, mecânicas, dinâmicas e componentes. As dinâmicas são os elementos que aplicam fatores motivacionais através da narrativa, interação social, emoções,

progressão, entre outros. As mecânicas são os elementos impulsionadores do envolvimento do utilizador e inclui aspetos como o desafio, recompensa, competição, colaboração e retorno. Os componentes são os elementos que implementam as dinâmicas e as mecânicas e são compostos por elementos como pontos, medalhas, níveis e quadros de honra.

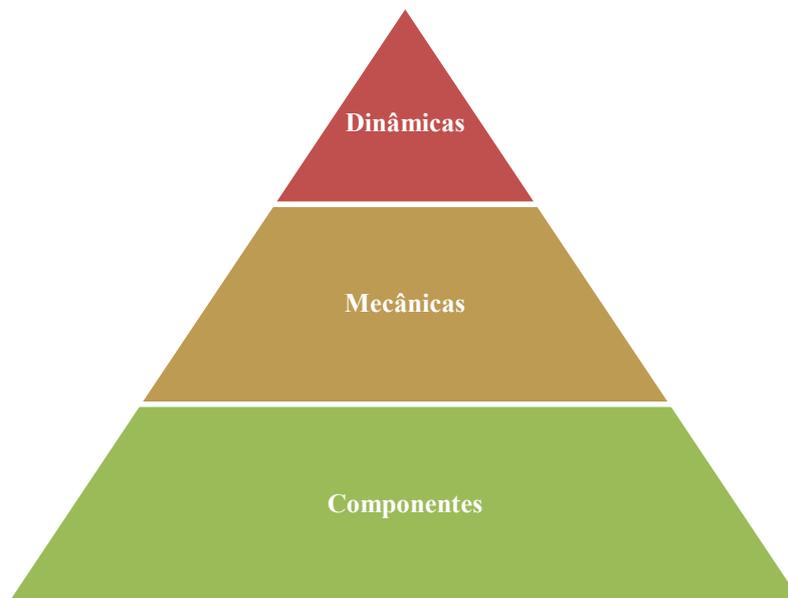


Figura 5 - Hierarquia dos Elementos de Jogo (Werbach & Hunter, 2012)

Tendo por base a pirâmide dos elementos de jogo de Werbach & Hunter (2012) os autores (Costa, Aparicio, Aparicio, & Aparicio, 2017) identificaram a partir da literatura os elementos de jogo relativos a dinâmicas, mecânicas e componentes. Nas dinâmicas de jogo identificaram as emoções, narrativa, restrições e progressão. Para as mecânicas de jogo identificaram os desafios, competição, cooperação, retorno, recompensa e transações. Por último nos componentes de jogo identificaram os avatares, as medalhas, os níveis, os pontos, os bens virtuais, os quadros de honra e conteúdo bloqueado. Por exemplo, os pontos (componentes) fornecem recompensas (mecânicas) e criam uma sensação de progressão (dinâmicas), (Dichev & Dicheva, 2017). Os termos referentes a estes três elementos têm sido utilizados na literatura de forma interrelacionada dado que não existe uma definição e aceitação geral definida (Dichev & Dicheva, 2017; Dicheva et al., 2015). Por exemplo, Burke (2016) descreve as mecânicas de jogo como pontos, medalhas, e quadros de honra, entre outros, enquanto (Werbach & Hunter,

2012) classifica estes elementos como componentes. Por exemplo (Dichev & Dicheva, 2017) utilizam o termo elementos de jogo quando se referem aos componentes.

De todos os elementos de jogo anteriormente identificados, observa-se da análise à literatura que uns são mais utilizados que outros. Por exemplo, Zichermann & Cunningham (2011) descrevem que os elementos de jogo mais utilizados são: pontos, níveis, quadros de honra, medalhas e desafios. Estes elementos foram também identificados por Dicheva et al. (2015) e por Werbach & Hunter (2015) como os mais utilizados. Seguidamente, apresenta-se uma descrição sumária de cada um desses elementos.

Os Pontos são um dos elementos centrais nos jogos. Os utilizadores recebem pontos quando interagem com o jogo e podem ser utilizados com os mais variados propósitos. Contudo, são normalmente utilizados para orientar o utilizador no seu progresso.

Os Níveis representam o progresso no jogo à semelhança dos pontos. O jogo tem diversos níveis e em cada um dos níveis é requerido o alcance de um ou mais objetivos. Idealmente, à medida que se vai subindo nos níveis, o grau de dificuldade deverá aumentar relativamente ao anterior. Contudo, Zichermann e Cunningham (2011) sugerem que o grau de dificuldade nos níveis, não deverá ser exponencial nem linear. Deve-se variar a dificuldade em cada nível, de forma a permitir aos utilizadores, experienciar diferentes emoções, tais como o orgulho e a confiança.

Os Quadro de Honra (*leaderboards*) são um dos elementos que integram uma vertente social na gamificação. Através do quadro de honra é possível visualizar os resultados dos diversos utilizadores, facilitando assim a comparação dos pontos, dos níveis completados ou das medalhas obtidas. Este elemento permite assim introduzir um fator de competição no jogo. O utilizador poderá assim experienciar o orgulho por ter alcançado uma posição no quadro de honra e sentir-se desafiado a ir mais além e quem não alcançou essa posição, poderá sentir-se desafiado a alcançar uma nesse quadro.

As Medalhas são elementos que são representados através de símbolos gráficos e que são atribuídos aos utilizadores quando completam uma determinada tarefa ou objetivo. Para o utilizador, a medalha pode ser representativa de orgulho e de satisfação. Nos sistemas, gamificados existem diversas medalhas que o utilizador pode colecionar, sendo um fator motivacional para alguns utilizadores, o conseguir obter todas as medalhas disponíveis e poder exibi-las como resultado da sua evolução. Zichermann e Cunningham (2011) referem que alguns utilizadores gostam das medalhas e de as colecionar apenas por razões estéticas.

Os Desafios são elementos que implementam objetivos a serem alcançados de forma a poder progredir. Os desafios podem ser implementados nas mais diversas formas, como a limitação de tempo para uma determinada tarefa (e.g. responder a uma questão no menor tempo possível).

A aplicação da gamificação no contexto educacional pode contribuir para aumentar a motivação dos estudantes na aprendizagem (Zichermann & Cunningham, 2011) e tem o potencial de aumentar o envolvimento do estudante na aprendizagem, enquanto fornece retorno dessa aprendizagem (Seaborn & Fels, 2015). Por conseguinte, parece existir um bom ajuste entre a gamificação e a programação. Um dos princípios de desenho da gamificação é fornecer retorno aos jogadores para que alcancem uma sensação de domínio (Richards, Thompson, & Graham, 2014). Concluir uma atividade e aprender através do erro, são na maioria representados na gamificação na forma de realizações, medalhas, quadro de honra, alcançar o nível máximo, entre outros (Richards et al., 2014). A gamificação oferece ciclos curtos de retorno e permite aos estudantes avaliar as suas competências e capacidades e criar um ambiente onde o esforço de aprendizagem é recompensado. Os estudantes, em alternativa, podem ver o falhar como uma oportunidade de aprendizagem, sendo particularmente útil para os estudantes que tendem a desistir com facilidade, quando os resultados não são os esperados. O conceito de gamificação em si, não é novo, uma vez que medalhas e rankings têm sido largamente utilizados em contextos militares, bem como outros desde há largos anos (Kapp, 2012). Contudo, o que atualmente torna a gamificação emergente é a conjugação de um conjunto de factores, como por exemplo a tecnologia mais barata, a obtenção / análise de dados pessoais, o sucesso iminente e a prevalência de características de jogo (Deterding et al., 2011). Na literatura identificam-se diversos estudos que abordam a utilização da gamificação, contudo, os seus efeitos ainda permanecem incipientes (Dicheva et al., 2015; Ibáñez et al., 2014; Iosup & Epema, 2014; Richards et al., 2014). A gamificação em contexto educacional tem sido investigada e reportada a utilização de algumas das mecânicas, dinâmicas e componentes, contudo, estudos empíricos mais aprofundados sobre a efetividade da incorporação dos elementos de gamificação em ambientes de aprendizagem, continuam escassos (Dicheva et al., 2015). No entanto, a maioria dos autores partilha da opinião que a gamificação tem potencial para melhorar a aprendizagem, se for bem desenhada e usada corretamente (Dichev & Dicheva, 2017). Consequentemente, mais investigação empírica é necessária para investigar em particular, os efeitos dos elementos num contexto educacional específico (Dicheva et al., 2015). Muitos desses trabalhos centram a sua investigação em ferramentas e em protótipos que os autores desenvolveram, sendo assim necessária uma investigação que explore a utilização

desses elementos nas atividades de aprendizagem regulares e principalmente, nas atividades *online* suportadas por plataformas de aprendizagem. A introdução destas plataformas nos últimos anos, permitiu uma inovação nas metodologias de aprendizagem, suportadas pela tecnologia. As escolas podem tirar vantagens destas tecnologias para tornar a aprendizagem mais acessível e mais efetiva. Contudo existem poucos estudos que exploram a integração de elementos de jogos nessas plataformas e qual o impacto nos estudantes (Dicheva et al., 2015).

A utilização da gamificação tem como objetivo melhorar a experiência, o envolvimento, a fidelidade e a diversão do utilizador nas mais diversas áreas (Deterding et al., 2011; Lee & Hammer, 2011). Nos últimos anos, a gamificação tem sido utilizada em diversos contextos, tais como, o marketing (Muntean, 2011), saúde (Brazil et al., 2018; Lee & Hammer, 2011), setor bancário (Rodrigues, Oliveira, & Costa, 2016b) e na educação (Buckley & Doyle, 2016; Dicheva et al., 2015; Domínguez et al., 2013; Ibáñez et al., 2014; Iosup & Epema, 2014; Kapp, 2012; Kyewski & Krämer, 2018; Landers & Callan, 2011; McLeod, Hewitt, Gibbs, & Kristof, 2017; Richards et al., 2014; Seixas et al., 2016; Yildirim, 2017).

Considerando que através da gamificação é possível criar experiências ricas e envolventes, a gamificação tem vindo a ser adotada por empresas e organizações nas mais diversas áreas, como por exemplo, a Starbucks e a Nike.

A Starbucks é uma conhecida cadeia internacional de cafés, que desenvolveu um programa de pontos designado “*My Starbucks Rewards*”. Dessa forma, introduziu a gamificação no programa de fidelização dos clientes. As compras que os clientes fazem, são recompensadas com pontos ou estrelas, que posteriormente podem ser trocadas por produtos. O programa “*My Starbucks Rewards*” disponibiliza uma aplicação móvel através do qual o utilizador pode interagir, efetuar a compra e o pagamento e automaticamente ser recompensado pelas compras que fez (Kim & Ahn, 2017).

A Nike é uma empresa mundial de equipamentos de desporto, incorporou a gamificação na sua aplicação Nike + Fuel, através da qual os utilizadores podem fazer o registo da sua atividade física. A aplicação recolhe a atividade realizada e informa o utilizador do desempenho alcançado. Permite depois ao utilizador partilhar os seus resultados com outros utilizadores e efetuar a comparação com utilizadores do mundo inteiro, que estão conectados na aplicação. O desempenho é transformado em pontos, recompensando os utilizadores pelo seu esforço. Após

completarem diferentes níveis de atividade física, os utilizadores são recompensados com troféus e medalhas (Blohm & Marco, 2013).

Outras organizações adotaram a gamificação dos seus produtos e serviços, como a SAP, que através da gamificação da sua comunidade em rede, aumentou a utilização em 400% e o *feedback* da comunidade em 96%. Também a Delloite, através do seu programa de formação interno, verificou que o tempo de conclusão foi reduzido em 50% e melhorou significativamente o envolvimento. Igualmente, a Cisco utilizou estratégias de jogo para incentivar as vendas *online*, verificando-se que reduziram o tempo da chamada em cerca de 15% e melhoraram as vendas em torno de 10%. (fonte: <http://yukaichou.com/gamification-examples/gamification-stats-figures/>).

2.3.4.1 Gamificação na aprendizagem da programação

Neste subcapítulo, é realizada uma síntese da revisão da literatura referente à utilização da gamificação na aprendizagem da programação.

Ibañez et al. (2014) conduziram um estudo com o propósito de investigar o impacto das técnicas de gamificação no envolvimento cognitivo dos estudantes e na aprendizagem da linguagem de programação C. Os autores criaram uma plataforma que designaram como *Q-Learning-G* com elementos básicos de gamificação. Os estudantes eram reconhecidos pelas atividades realizadas e os seus feitos eram tornados visíveis no quadro de honra e na área das medalhas. O estudo foi conduzido com uma amostra de 22 alunos. Os dados foram recolhidos através da aplicação de questionários. De acordo com os autores, após os estudantes obterem os cem pontos que foram definidos como objetivo pedagógico, os alunos continuaram a trabalhar, o qual, segundo os autores, era uma evidência de envolvimento cognitivo. Salientam os autores, que outras das razões para este envolvimento, relaciona-se com o facto de os estudantes quererem colecionar todas as medalhas disponíveis, estas conclusões reportadas pelos autores estão em consonância com (Hamari, 2017). Por outro lado, segundo os autores, a ausência das medalhas desencoraja os estudantes a continuar o seu trabalho. Esta observação é suportada por Hamari (2017), que refere que as medalhas fornecem retorno, que é considerado como um importante antecedente, para o *flow* e para o envolvimento (Csikszentmihalyi, 1975). Concluem os autores, que do ponto de vista académico, a gamificação é um sucesso, suportados pela comparação dos resultados dos pré e pós testes, onde claramente se denota que os estudantes melhoraram os seus conhecimentos na linguagem C. A aceitação e o envolvimento dos estudantes com as atividades de aprendizagem foram também observadas pelos autores Knutas et al. (2014), que realizaram

um trabalho de investigação com o objetivo de aumentar a colaboração *online* entre estudantes, numa disciplina de introdução à programação. O estudo foi realizado por 249 participantes, quatro assistentes e um professor. A maioria dos estudantes frequentava o primeiro ano da universidade. A disciplina tinha duas horas por semana, de aula teórica e duas horas de laboratório, com uma avaliação *online* e avaliada pelo sistema. Ao sistema foi adicionado um sistema *online* de colaboração assíncrona com elementos de gamificação. O sistema reconhecia e recompensava quem contribuía construtivamente com perguntas, comentários ou respostas. Os utilizadores podiam também publicar e tornar visíveis os pontos e as medalhas obtidas no sistema. Os dados foram recolhidos a partir dos *logs* do sistema, do perfil dos utilizadores e da análise correlacionada entre os dois. Os resultados mostraram que a introdução dos elementos de gamificação, contribuiu para melhorar a colaboração e interação *online*, com os estudantes participativos e a fazerem contribuições para a comunidade online. Verificaram também uma redução de questões colocadas aos assistentes e uma diminuição de perguntas repetidas. Estes resultados mostram que a utilização dos elementos de gamificação, e gamificação no geral, podem influenciar o comportamento dos estudantes, devido à comparação social (Hamari, 2017) e suportado pela teoria da comparação social estudada por Festinger (1954). A conclusões similares, chegaram os autores Fotaris, Mastoras, Leinfellner & Rosunally (2016) que avaliaram o impacto da gamificação, através de um estudo empírico que comparou dois grupos: gamificado e não gamificado. Os autores utilizaram uma combinação de sistemas, concretamente o *Kahot!*, a versão para sala de aula do programa de TV “*Who Wants To Be A Millionaire*” e a plataforma *online Codecademy* (<https://www.codecademy.com/>). Esta experiência foi aplicada numa turma de introdução à linguagem de programação *Python*. Os autores utilizaram um questionário *online* com quinze questões, para obterem o retorno relativo à eficácia da experiência gamificada. Referem os autores que, por exemplo, com a utilização da plataforma *online Codecademy*, os pontos e as medalhas são motivadores para completar e praticar os exercícios. Ao mesmo tempo, os estudantes estão intrinsecamente motivados para completar os exercícios e obter um melhor desempenho que os seus colegas.

Recentemente, com a massificação do ensino *online* e a distância e com o surgimento de novas abordagens, como o ensino aberto e massificado, surgiram diversas plataformas online. Algumas dessas plataformas disponibilizam cursos nas mais diversas áreas, como é o exemplo do *Coursera* (<http://coursera.org>). Especificamente para a aprendizagem da programação, existem a *Khan Academy* (<https://www.khanacademy.org/>), o *Stackoverflow* (<https://stackoverflow.com/>) e o *Codecademy* (<http://codecademy.org>), atualmente

renomeada para (<https://www.fullstackacademy.com/>) e atualizada com novas funcionalidades. Estas plataformas integraram recentemente nas suas atividades de aprendizagem elementos de gamificação. Em seguida descrevem-se sumariamente a plataforma *Khan Academy*, a *Stackoverflow* e o processo de gamificação utilizado.

A *Khan Academy* é uma organização sem fins lucrativos com o objetivo de mudar a educação, facilitando o acesso à educação no mundo inteiro, sem qualquer custo associado. A plataforma *Khan Academy*, tem mais de 4500 lições que são acedidas através de vídeos *online* e integra um mecanismo de autoavaliação *web-based*, através do qual são geradas questões para os estudantes, tendo em conta o nível em que se encontram e o desempenho obtido anteriormente. A *Khan Academy* integrou diversos elementos de gamificação, tais como: medalhas e pontos. As medalhas foram categorizadas em meteorito, lua, terra, sol, buraco negro e desafio e os estudantes podem ganhar essas medalhas representando os objetivos alcançados. As medalhas são listadas no perfil do utilizador e podem ser partilhadas na rede social *Facebook*. Os pontos foram designados por pontos de energia. Após completar as atividades, o utilizador recebe pontos de energia. Os pontos ficam visíveis num painel, assim como o número de vídeos visualizados e as medalhas obtidas. O utilizador acumula pontos e consoante os pontos acumulados, recebe a medalha específica para os pontos obtidos num dado momento (Morrison & DiSalvo, 2014).

O *Stackoverflow* é uma rede online mundial baseada em pergunta e resposta para programadores, através da qual é construído e partilhado conhecimento. Esta rede online tem implementado um conjunto diversificado de elementos de gamificação: pontos; medalhas; e níveis. À medida que os utilizadores ganham reputação com a obtenção de pontos, são desbloqueados novos recursos, como a possibilidade de votar, comentar ou editar os posts dos outros utilizadores. Atualmente, o *Stackoverflow* está a disponibilizar um novo projeto denominado *Documentation*, ao qual aplicam o mesmo sistema de gamificação (Richter, Raban, & Rafaeli, 2015).

2.4 Adoção da tecnologia

A adoção da tecnologia é uma das teorias largamente utilizadas, para explicar o uso e a aceitação individual dos sistemas de informação (SI) e das tecnologias de informação (TI). Com base nesta teoria, diversos estudos e.g. (Picoto, Bélanger, & Palma-dos-Reis, 2014; Soares-Aguiar & Palma-Dos-Reis, 2008) foram realizados com o objetivo de identificar fatores intrínsecos e

extrínsecos envolvidos nas decisões, intenções e satisfação individual, na aceitação e uso dos SI e das TI, através de diversos métodos e testes (Venkatesh, Speier, & Morris, 2002). Diversos modelos têm sido propostos para estudar a adoção da tecnologia e um dos mais utilizados pelos investigadores, foi proposto por Davis em (1986) e tem como objetivo explicar o comportamento dos utilizadores na utilização do computador. O modelo explica o relacionamento causal entre as variáveis de aceitação do utilizador e a utilização atual do sistema, procurando o comportamento do utilizador através do conhecimento da utilidade e da facilidade percebida. Este modelo é útil, não apenas para prever, mas também para descrever e identificar a razão da rejeição de uma determinada tecnologia ou sistema e consequentemente, implementar medidas corretivas (Davis, 1986, 1989).

2.4.1 Modelos de adoção e uso da tecnologia

Um dos modelos largamente utilizado pelos investigadores, para estudar e explicar a adoção e uso da tecnologia, é o modelo de aceitação da tecnologia TAM (*Technology Acceptance Model*). Neste subcapítulo, descrevem-se a sua origem e evolução e identificam-se outros modelos de adoção da tecnologia revistos na literatura.

O modelo TAM (Figura 6) resulta numa adaptação do modelo *Theory of Reasoned Action* (TRA) (Ajzen & Fishbein, 1972). De acordo com o TRA, as convicções influenciam as atitudes, as quais influenciam as intenções, as quais podem gerar comportamentos positivos ou negativos. O TAM adapta as convicções-atitudes-intenções-comportamento, a um modelo de aceitação de tecnologia (Hu, Chau, Sheng, & Tam, 1999). O TAM tem como objetivo explicar os determinantes da aceitação da tecnologia, contudo, não atua apenas como preditor, pretendendo também explicar a razão pela qual a tecnologia não é aceite (Davis, 1989).

O modelo de aceitação da tecnologia TAM é composto pelas dimensões “utilidade percebida” e “facilidade de utilização”. Estas dimensões assumem um papel relevante, na aceitação e uso das tecnologias da informação. Estas dimensões foram inicialmente utilizadas para estudar a aceitação da tecnologia pelos novos utilizadores (Davis, 1986), sendo mais tarde confirmados como bons preditores para o uso continuado da tecnologia.

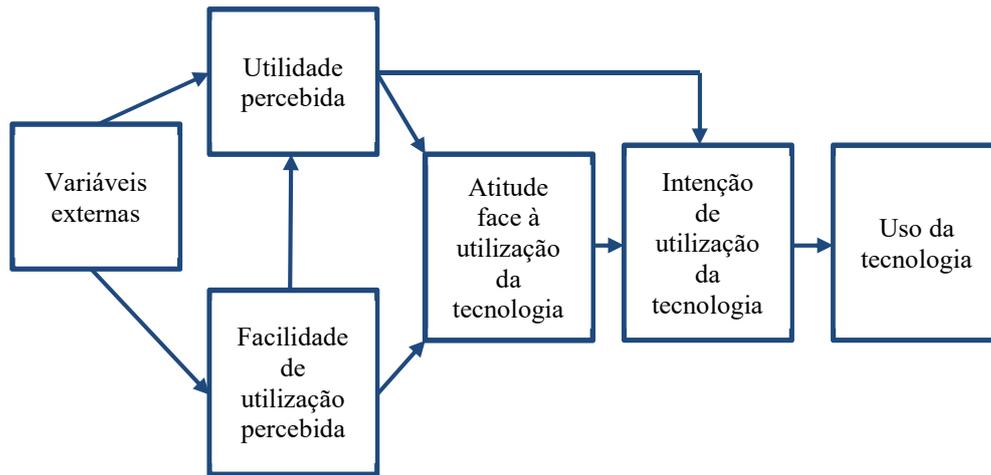


Figura 6 - Modelo TAM, adaptado de Davis (1989)

Em resultado dos estudos de investigação associados aos modelos de aceitação de tecnologia, o modelo TAM evoluiu e passou a integrar duas novas dimensões (Figura 7): processos de instrumentação cognitiva e processos de influência social (Venkatesh & Davis, 2000). Os processos de instrumentação cognitiva são compostos pelas seguintes dimensões: relevância do emprego, demonstrabilidade de resultados e percepção de facilidade de uso. Os de influência social estão relacionados com a norma subjetiva, o carácter voluntário e o impacto que têm sobre a aceitação da tecnologia. Ao longo dos últimos anos, o TAM tem sido utilizado nos mais diversos contextos, mais especificamente, na área da educação e formação e têm sido adicionadas novas dimensões consideradas relevantes na atitude e adoção das tecnologias, como por exemplo: a absorção cognitiva (Agarwal & Karahanna, 2000); o *flow* (Koufaris, 2002); a diversão e o prazer (Van der Heijden, 2004); e gamificação (Rodrigues, 2016). Diversas outras dimensões têm sido adicionadas ao modelo TAM, conforme referido por (Rodrigues, 2016), no seu trabalho de investigação. Na literatura, identificaram-se outros modelos de adoção de tecnologia, e.g.: Teoria do Comportamento Planeado (*Theory of Planned Behavior*) (Icek Ajzen, 1991); Ajuste Tecnologia-Tarefa (*Task-Technology Fit*) (Goodhue & Thompson, 1995); Teoria Unificada de Aceitação e Utilização da Tecnologia (*Unified Theory of Acceptance and Use of Technology*) (Venkatesh, Morris, Davis, & Davis, 2003); e Teoria Unificada de Aceitação e Utilização da Tecnologia2 (Venkatesh, Thong, & Xu, 2012).

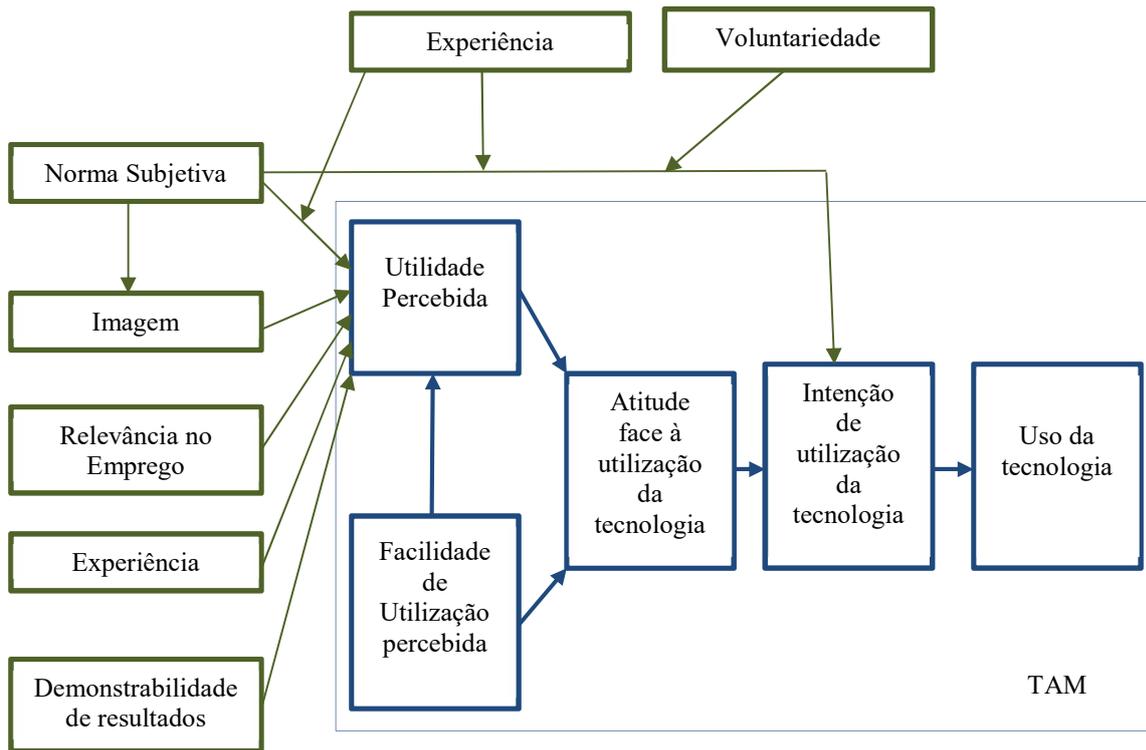


Figura 7 – Modelo TAM2 (fonte: adaptado de (Venkatesh & Davis, 2000))

2.5 Conclusões

Neste capítulo, foram revistos na literatura os problemas na aprendizagem da programação e as soluções propostas para auxiliar os estudantes a ultrapassar os problemas enfrentados durante a aprendizagem da programação. Foi também revisto o conceito de gamificação e descrita a sua utilização em contexto educacional. Descreveu-se também a teoria de adoção da tecnologia e identificaram-se os modelos propostos para estudar a adoção da tecnologia em diversos contextos de utilização.

A aprendizagem da programação é atualmente uma das competências mais solicitadas. Estima-se que na próxima década exista um déficit de profissionais com competências digitais capazes de desenvolver atividades no mercado digital. É assim necessário formar e cativar estudantes para as áreas mais tecnológicas. Verifica-se, no entanto, que a aprendizagem de uma das competências base, a programação, apresenta vários problemas revistos na literatura, que contribuem para a desmotivação dos estudantes, com a consequente desistência da aprendizagem desta temática. Descrevem-se seguidamente os problemas e as soluções revistas

na literatura. Os problemas revistos na literatura são de natureza diversa. É referido, por exemplo, que a natureza abstrata da programação é uma barreira à aprendizagem, dado requerer dos estudantes, uma capacidade de abstração, capacidade esta, que os estudantes iniciantes ainda não desenvolveram totalmente. Devido à natureza abstrata da programação, foram reportados problemas associados com a compreensão e o desenho do algoritmo/programa para a resolução de um determinado problema. Também o paradigma e a linguagem de programação utilizados foram apontados como fatores que podem contribuir para a dificuldade na aprendizagem da programação, principalmente para a aprendizagem de estudantes iniciantes. Por outro lado, foram identificados os modelos mentais que são necessários à compreensão da natureza da programação e que são ainda ineficientes, assumindo-se como um entrave à compreensão total. Outro problema reportado está relacionado com a atitude que os estudantes adotam face à programação, i.e., percecionam-na como sendo uma matéria de difícil aprendizagem. Quando confrontados com as dificuldades, têm tendência a desistir, a desmotivar e a perder a sua autoconfiança, desistindo do estudo da programação. As soluções revistas na literatura foram propostas com o objetivo de ajudar os estudantes a ultrapassar as diversas dificuldades sentidas durante o processo de aprendizagem. Diversas soluções suportadas pela tecnologia foram propostas, como por exemplo, ferramentas de visualização gráfica, robótica educacional, jogos educacionais e as recentes soluções baseadas na gamificação. Considerando que um dos problemas relacionados com a aprendizagem da programação é a desmotivação dos estudantes, quando confrontados com os problemas na aprendizagem, soluções que permitam criar experiências ricas, envolventes e motivadoras, podem ser um contributo relevante para manter os estudantes em contacto com os conteúdos programáticos e motivá-los para realizar as atividades de aprendizagem. A gamificação surge como uma solução que pode contribuir para ultrapassar esses problemas e sendo uma solução emergente e com um enorme potencial de aceitação por parte dos educadores, pode desempenhar um papel relevante ao centrar os estudantes na aprendizagem. Nesse sentido, neste capítulo, reviram-se o conceito de gamificação, os diversos contextos onde tem sido aplicada a gamificação e como tem sido aplicada.

Dado que a introdução de novas abordagens educacionais suportadas pela tecnologia, podem ter impactos na aceitação por parte do estudante, efetuou-se uma revisão da literatura relativamente à adoção da tecnologia e identificaram-se os modelos de aceitação e utilização da tecnologia na área dos sistemas de informação.

A identificação dos problemas na revisão da literatura, serviram de suporte à elaboração do estudo que se descreve no Capítulo 3, no qual identificámos as dificuldades percecionadas pelos estudantes, as quais se compararam com as percecionadas pelos professores, relativamente aos estudantes. Pretende-se com os resultados obtidos, complementar e aprofundar ainda mais o conhecimento relativamente aos problemas na aprendizagem da programação.

3 IDENTIFICAÇÃO DAS DIFICULDADES PERCECIONADAS PELOS ESTUDANTES NA APRENDIZAGEM DA PROGRAMAÇÃO - ESTUDO 1

3.1 Introdução

No capítulo anterior efetuou-se a revisão da literatura referente às dificuldades na aprendizagem da programação. Identificaram-se as soluções que podem contribuir para ultrapassar as dificuldades que os estudantes enfrentam na aprendizagem da programação. Com base na revisão da literatura efetuada no capítulo anterior, descreve-se no presente capítulo, um estudo de investigação que tem como objetivos: a identificação das dificuldades na aprendizagem da programação, percecionadas por estudantes; e a perceção dos professores, relativamente às dificuldades sentidas pelos estudantes. Após a recolha e tratamento de dados, apresentam-se os resultados obtidos.

3.2 Objetivos

Neste estudo, foi definido como objetivo geral, a identificação das dificuldades percecionadas na programação, tanto por estudantes, como por professores. As dificuldades foram categorizadas em conceitos de programação e situações de aprendizagem. Neste estudo, foi também identificada a utilidade percebida dos contextos de aprendizagem e dos materiais/recursos, como também a atitude dos estudantes face à programação.

3.3 Abordagem metodológica

A abordagem metodológica adotada seguiu os seguintes procedimentos: identificação das principais dificuldades referidas na revisão da literatura, desenvolvimento de instrumento de recolha de dados (questionário), lançamento de questionário a amostra de estudantes e professores, respetiva recolha e análise dos resultados. A análise dos dados foi efetuada com recurso ao método de análise quantitativa. Foi utilizada a técnica de estatística simples, com o cálculo das médias, desvio padrão e erro no desvio padrão.

3.4 Questionários

No presente estudo, foram aplicados dois questionários. Um questionário para recolha das perceções dos estudantes e um outro para a recolha das perceções dos professores. O questionário para recolha das perceções, foi criado e adaptado com base na revisão da literatura efetuada e desde a conceção até à sua disponibilização *online*, seguiu um conjunto de etapas, conforme demonstra a Ilustração 1. É composto por questões fechadas e a sua classificação foi definida tendo por base a escala de *likert* de sete valores.

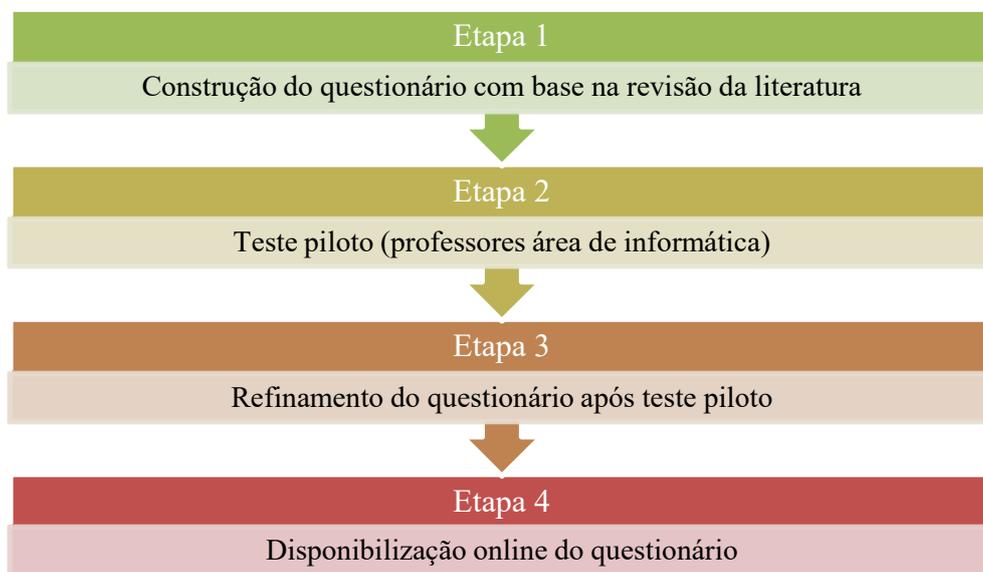


Ilustração 1 - Etapas da conceção do questionário de investigação

O questionário (Anexo A) encontra-se dividido em cinco grupos de perguntas. O primeiro contém perguntas de caracterização sociodemográfica dos participantes; o segundo aborda questões relacionadas com as dificuldades nas situações de aprendizagem (Tabela 7); o terceiro está relacionado com a dificuldade nos conceitos de programação (Tabela 8); o quarto demonstra a utilidade nos contextos de aprendizagem (Tabela 9); o quinto debruça-se sobre a utilidade dos materiais/recursos de aprendizagem (Tabela 10); o sexto demonstra as questões relacionadas com a atitude face à programação (Tabela 11); e o sétimo aborda outras questões relacionadas com o tema de investigação. Para cada grupo foram identificados os respetivos itens, que se detalham no subcapítulo seguinte. No instrumento de recolha de dados das perceções dos professores, foram adotados os seguintes procedimentos: adaptação do

instrumento de recolha das perceções dos estudantes, mais concretamente, a reformulação das perguntas, o lançamento do questionário a uma amostra de professores, a respetiva recolha, o tratamento dos dados e a análise.

Tabela 7 – Itens do Questionário - Situações de Aprendizagem

Dimensão	Código	Variáveis	Descrição	Autor
Situações de Aprendizagem	I1	Na utilização do ambiente de desenvolvimento (IDE)	Utilização do programa de desenvolvimento do <i>software</i> (<i>debugs</i> e organização de ficheiros).	(Lahtinen et al., 2005; Schulte & Bennedsen, 2006)
	I2	Na estruturação adequada do programa	Estruturar um programa nos seus diferentes blocos de forma adequada.	(Lahtinen et al., 2005; Schulte & Bennedsen, 2006)
	I3	Aprendizagem da sintaxe da linguagem de programação	Compreender e aplicar a sintaxe correta de uma linguagem.	(Lahtinen et al., 2005; Schulte & Bennedsen, 2006)
	I4	Conceber um programa para a resolução de um determinado problema	Desenhar e implementar um programa (algoritmo) para a resolução correta de um problema.	(Lahtinen et al., 2005; Robins et al., 2003; Winslow, 1996)
	I5	Dividir as funcionalidades em procedimentos	Agrupar as funcionalidades do programa em blocos (procedimentos).	(Lahtinen et al., 2005; Winslow, 1996)
	I6	Encontrar erros no meu próprio programa	Interpretar os erros do programa e saber corrigir.	(Lahtinen et al., 2005; Schulte & Bennedsen, 2006)

Tabela 8 – Itens do Questionário – Conceitos de Programação

Dimensão	Código	Variáveis	Descrição	Autor
Conceitos de Programação	C1	Variáveis	Dificuldade na compreensão e na aplicação das variáveis: definir tipo variável, tempo de vida e âmbito.	(Lahtinen et al., 2005; Milne & Rowe, 2002)
	C2	Estruturas de Seleção	Dificuldade em compreender e em aplicar as estruturas de seleção (IF simples e encadeados).	(Dale, 2006; Lahtinen et al., 2005; Milne & Rowe, 2002; Schulte & Bennedsen, 2006)
	C3	Estruturas de Repetição	Dificuldade em compreender e em aplicar as estruturas de repetição (for, <i>whiles</i> simples e encadeados).	(Lahtinen et al., 2005; Milne & Rowe, 2002; Schulte & Bennedsen, 2006)
	C4	Tabelas	Dificuldade em compreender e em aplicar os diferentes tipos de tabelas.	(Dale, 2006; Lahtinen et al., 2005; Milne & Rowe, 2002)
	C5	Ponteiros, Referências	Dificuldade na compreensão e na utilização dos conceitos de Ponteiros e Referências no desenvolvimento de um algoritmo.	(Lahtinen et al., 2005; Milne & Rowe, 2002; Schulte & Bennedsen, 2006)

Dimensão	Código	Variáveis	Descrição	Autor
	C6	Parâmetros (passagem por valor)	Dificuldade em compreender e em aplicar num programa, os parâmetros com passagem por valor.	(Dale, 2006; Lahtinen et al., 2005; Milne & Rowe, 2002)
	C7	Parâmetros (passagem por referência)	Dificuldade em compreender e em aplicar num programa, os parâmetros com passagem por referência.	(Lahtinen et al., 2005; Milne & Rowe, 2002; Schulte & Bennedsen, 2006)
	C8	Tipos Estruturados de Dados	Dificuldade em compreender e em aplicar tipos estruturados de dados (ex: <i>struct</i>).	(Lahtinen et al., 2005; Milne & Rowe, 2002)
	C9	Tipos Abstratos de Dados	Dificuldade em compreender e em aplicar tipos abstratos de dados (ex: pilhas e filas).	(Lahtinen et al., 2005; Milne & Rowe, 2002)
	C10	Manuseamento de entrada e saída de dados	Dificuldade em compreender e em aplicar a entrada de dados no programa e a sua correta manipulação na saída.	(Lahtinen et al., 2005)
	C11	Tratamento de Erros	Dificuldade em saber aplicar no programa, estruturas que façam o tratamento do erro.	(Lahtinen et al., 2005)
	C12	Utilização das Bibliotecas da Linguagem	Dificuldade em entender e em aplicar as bibliotecas de uma linguagem no dado contexto do programa.	(Lahtinen et al., 2005; Schulte & Bennedsen, 2006)

Tabela 9 – Itens do Questionário – Contextos de Aprendizagem

Dimensão	Código	Variáveis	Descrição	Autor
Contextos de Aprendizagem	S1	Nas aulas teóricas.	Em que medida as aulas teóricas permitem aprender programação.	(Lahtinen et al., 2005)
	S2	Em sessões de resolução de exercícios com os colegas	Em que medida os exercícios feitos com o apoio de colegas, permitem aprender programação.	(Lahtinen et al., 2005)
	S3	Nas aulas de laboratório	Em que medida as aulas de laboratório permitem aprender programação.	(Lahtinen et al., 2005)
	S4	Quando estudo sozinho	O aluno consegue aprender quando estuda sozinho.	(Lahtinen et al., 2005)
	S5	Quando trabalho sozinho nos exercícios da unidade curricular	Quando realiza exercícios da UC sozinho consegue aprender a programar.	(Lahtinen et al., 2005)
	S6	Em sessões de esclarecimento de dúvidas com o docente da unidade curricular (horário de dúvidas)	As sessões de dúvidas permitem aprender e melhorar a aprendizagem em programação.	Nossa autoria
	S7	Exercícios resolvidos em sala de aula	Os exercícios resolvidos em sala de aula ajudam na aprendizagem.	(Lahtinen et al., 2005)
	S8	Fóruns de discussão na Internet	O aluno consegue aprender através dos fóruns de discussão sobre a temática	Nossa autoria

Tabela 10 - Itens do Questionário - Materiais / Recursos

Dimensão	Código	Variáveis	Descrição	Autor
Materiais	M1	Livro recomendado da unidade curricular	Utilidade do livro recomendado na UC para a aprendizagem.	(Lahtinen et al., 2005)
	M2	Apontamentos tirados nas aulas teóricas	Utilidade para a aprendizagem dos apontamentos obtidos em sala de aula.	(Lahtinen et al., 2005)
	M3	Exemplos de programas completos	Utilidade para a aprendizagem de programas completos (codigo + executável).	(Lahtinen et al., 2005)
	M4	Imagens estáticas de programas estruturados	Utilidade de código de programas em imagens estáticas (ex: PDF ou powerpoint).	(Lahtinen et al., 2005)
	M5	Visualizações interativas através de ferramentas de visualização gráfica como por exemplo (Alice, BlueJ, RoboCode ou outra similar)	Utilidade para a aprendizagem da utilização de aplicações, que permite visualizações interativas do código.	(Lahtinen et al., 2005)
	M6	Tutoriais disponíveis na Internet	Utilidade para a aprendizagem dos tutoriais consultados na Internet.	Autoria Própria
	M7	Vídeos educativos disponíveis no Youtube	Utilidade para a aprendizagem dos vídeos educativos sobre programação na Internet.	Autoria Própria
	M8	Twitter - Páginas dedicadas à aprendizagem da programação	Utilidade para a aprendizagem de conteúdo sobre programação no Twitter.	Autoria Própria
	M9	Facebook- Páginas dedicadas à aprendizagem da programação	Utilidade para a aprendizagem de conteúdo sobre programação no Facebook.	Autoria Própria
	M10	Cópia dos acetatos	Utilidade dos acetatos para a aprendizagem.	(Lahtinen et al., 2005)
	M11	Recursos disponibilizados na plataforma Moodle	Utilidade dos recursos na plataforma Moodle.	Autoria Própria

Tabela 11 – Itens do Questionário - Atitude Face à Programação

Dimensão	Código	Variáveis	Autor
Atitude Face à Programação	AT1	Gosto de aprender programação!	(Korkmaz & Altun, 2014)
	AT2	Gosto de programar!	(Korkmaz & Altun, 2014)
	AT3	Acho que programar vai ser útil para a minha vida profissional!	(Korkmaz & Altun, 2014)
	AT4	Programar é fácil!	Autoria Própria
	AT5	Acho que programar é útil na minha formação!	Autoria Própria
	AT6	Invisto muito tempo na aprendizagem da programação!	Autoria Própria
	AT7	Sei programar muito bem!	(Korkmaz & Altun, 2014)

3.4.1 Amostra

A amostra selecionada para o presente estudo de investigação, diz respeito aos estudantes do ensino superior que já tinham frequentado, pelo menos uma vez, uma disciplina de introdução à programação de computadores. No total, responderam duzentos estudantes ao questionário. Do total de respostas obtidas, quarenta foram do género feminino e cento e sessenta do género masculino (Figura 8). A média de idades da amostra é de 23 anos. Relativamente à amostra para recolha das perceções dos professores, foram selecionados os professores de informática, tendo respondido um total de dez professores de uma instituição do ensino superior politécnico.

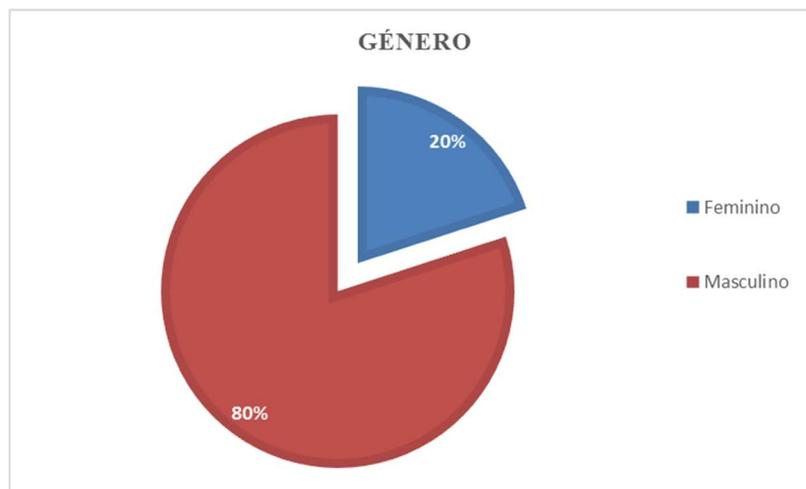


Figura 8 – Caracterização da Amostra

3.5 Recolha e tratamento dos dados

A recolha de dados decorreu em diversas fases. Numa primeira fase, o questionário foi submetido a uma validação por parte de professores da área de informática, de modo a aferir a clareza das perguntas, a estrutura e a organização do questionário. Seguidamente e tendo em consideração os comentários, foi efetuado o refinamento do questionário. Posteriormente, o questionário foi disponibilizado *online* através da ferramenta *Google Forms*.

Para a recolha dos dados, foram selecionadas as turmas dos cursos de engenharia da Escola Superior de Tecnologia de Setúbal, que já tinham frequentado pelo menos uma vez uma disciplina de programação. A partir dessa identificação, a investigadora agendou horários com os professores dessas turmas. Nesses horários, que por norma correspondiam a aulas de laboratório, a investigadora explicou a investigação que estava a realizar e qual o objetivo.

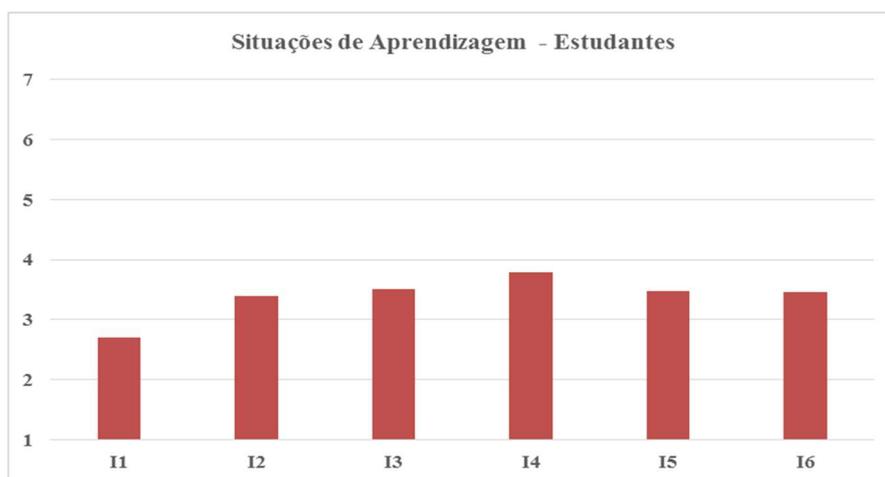
Seguidamente, solicitava a colaboração dos estudantes nessa investigação, através das respostas ao questionário *online*. A recolha dos dados decorreu durante dois semestres em diferentes anos letivos, à qual se seguiu a fase de tratamento dos mesmos.

3.6 Apresentação dos resultados

Após a fase do tratamento dos dados, seguiu-se a fase da apresentação dos resultados. A apresentação dos resultados é efetuada tendo por base os dois grupos de dificuldades na aprendizagem da programação: situações de aprendizagem e conceitos de programação. Os dois grupos de utilidade na aprendizagem da programação são: os contextos de aprendizagem e os materiais/recursos. De seguida apresentam-se os resultados da atitude dos estudantes face à programação e finaliza-se com a descrição dos resultados.

3.6.1 Situações de aprendizagem

Nas situações de aprendizagem foram analisados os itens: utilização do ambiente de desenvolvimento, compreensão na estruturação do programa, aprendizagem da sintaxe da linguagem, desenhar o programa para a resolução de uma determinada tarefa, dividir as funcionalidades em procedimentos e interpretar e resolver erros no programa (Gráfico 1).



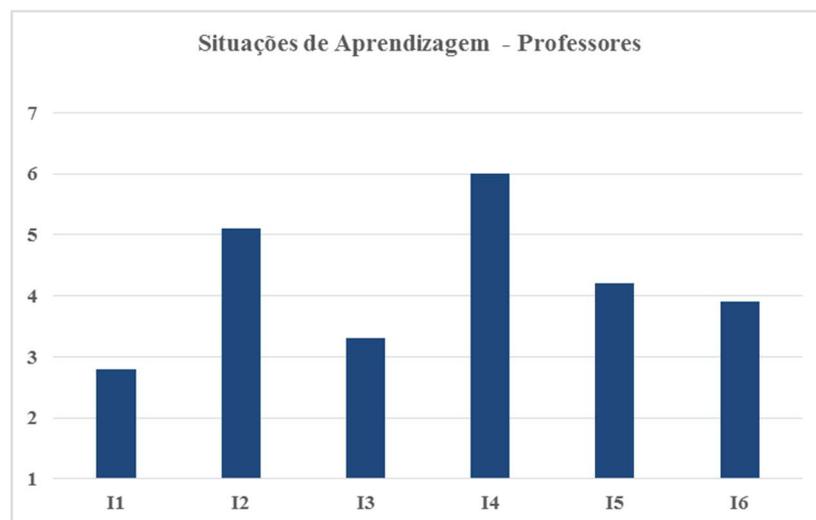
(1-fácil de aprender), (7-difícil de aprender)

(I1) – utilização do ambiente de desenvolvimento; (I2) – compreensão na estruturação do programa; (I3) – aprendizagem da sintaxe da linguagem; (I4) – desenhar o programa para resolução de uma determinada tarefa; (I5) – dividir as funcionalidades em procedimentos; (I6) – interpretar e resolver erros no programa.

Gráfico 1 – Dificuldade nas Situações de Aprendizagem – Perceção dos Estudantes

Da análise aos resultados, podemos observar que os estudantes consideraram ter tido menor dificuldade, na utilização do ambiente de desenvolvimento (IDE). Desenhar o programa para a resolução de uma determinada tarefa (I4), é considerado pelos estudantes, como sendo a situação de aprendizagem, na qual têm maior dificuldade. Dividir as funcionalidades em procedimentos (I5), interpretar e resolver erros no programa (I6), são situações de aprendizagem consideradas pelos estudantes de dificuldade média, comparativamente à situação da aprendizagem tida como a mais difícil.

Nas situações de aprendizagem percebidas pelos professores (Gráfico 2), observam-se que as situações de aprendizagem que estes interpretaram como sendo as mais difíceis para os estudantes, são as seguintes: desenhar o programa para a resolução de uma determinada tarefa (I4); compreender a estruturação do programa (I2); e dividir as funcionalidades em procedimentos (I5). Relativamente às situações de aprendizagem percebidas como tendo sido as que menor dificuldade apresentaram, identificaram-se a utilização do ambiente de desenvolvimento (I1) e a aprendizagem da sintaxe da linguagem (I3).



(1-fácil de aprender), (7-difícil de aprender)

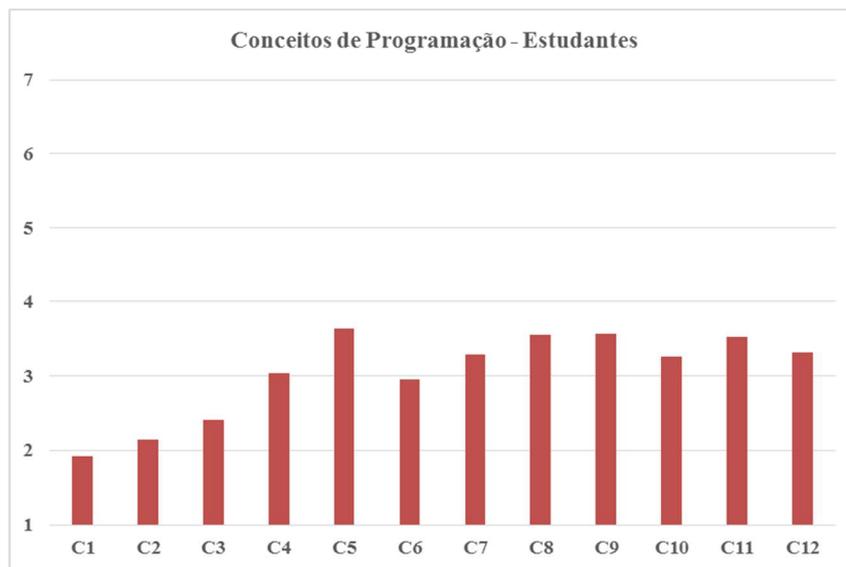
(I1) – utilização do ambiente de desenvolvimento; (I2) – compreensão na estruturação do programa; (I3) – aprendizagem da sintaxe da linguagem; (I4) – desenhar o programa para resolução de uma determinada tarefa; (I5) – dividir as funcionalidades em procedimentos; (I6) – interpretar e resolver erros no programa.

Gráfico 2 – Dificuldade nas Situações de Aprendizagem – Percepção dos Professores

3.6.2 Conceitos de programação

Nos conceitos de programação, foram analisados os seguintes itens: variáveis, estruturas de seleção, estruturas de repetição, parâmetros, tipos estruturados de dados, tipos abstratos de

dados, manuseamento de entrada e saída de dados, tratamento de erros e utilização das bibliotecas da linguagem. Da análise aos resultados (Gráfico 3), observa-se que os estudantes percecionam os ponteiros e referências (C5); tipos estruturados de dados (C8); tipos abstratos de dados (C9); e tratamento de erros (C11), como conceitos de programação de natureza difícil para a aprendizagem da programação. Dos conceitos que se apresentaram como sendo aqueles em que os estudantes tiveram menor dificuldade, identificaram-se as variáveis (C1), estruturas de seleção (C2) e estruturas de repetição (C3). Os restantes conceitos de programação, como as tabelas (C4), os parâmetros (C6-C7), o manuseamento de entrada e saída de dados (C10) e a utilização das bibliotecas da linguagem (C12), representaram uma dificuldade intermédia para os estudantes.



(1-fácil; 7-difícil)

(C1) – variáveis (C2) – estruturas de seleção; (C3) – estruturas de repetição; (C4) – tabelas; (C5) – ponteiros, referências; (C6) – parâmetros (valor); (C7) – parâmetros (referência); (C8) – tipos estruturados de dados; (C9) – tipos abstratos de dados; (C10) – manuseamento de entrada e saída de dados; (C11) – tratamento de erros; (C12) – utilização das bibliotecas da linguagem.

Gráfico 3 – Dificuldade nos Conceitos de Programação – Percepção dos Estudantes

Com base na observação dos resultados obtidos, relativamente às dificuldades percecionadas pelos professores (Gráfico 4), verifica-se que os professores percecionam que os estudantes têm mais dificuldades nos conceitos de programação ponteiros e referências (C5), parâmetros (C7), tipos estruturados de dados (C8) e tipos abstratos de dados (C9). Os conceitos de programação percecionados pelos professores como sendo os mais fáceis, são as estruturas de seleção (C2), a utilização das bibliotecas da linguagem (C12) e as variáveis (C1). São percecionados como

tendo uma dificuldade intermédia, os seguintes conceitos de programação: estruturas de repetição (C3); tabelas (C5); manuseamento e entrada e saída de dados (C10).

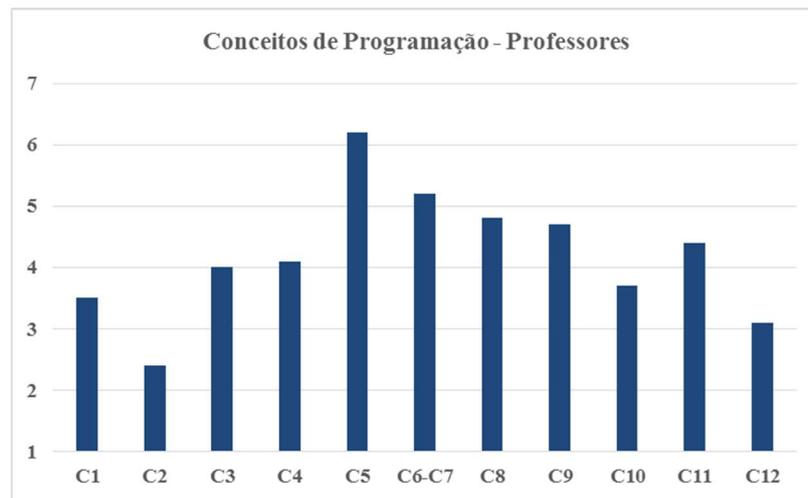
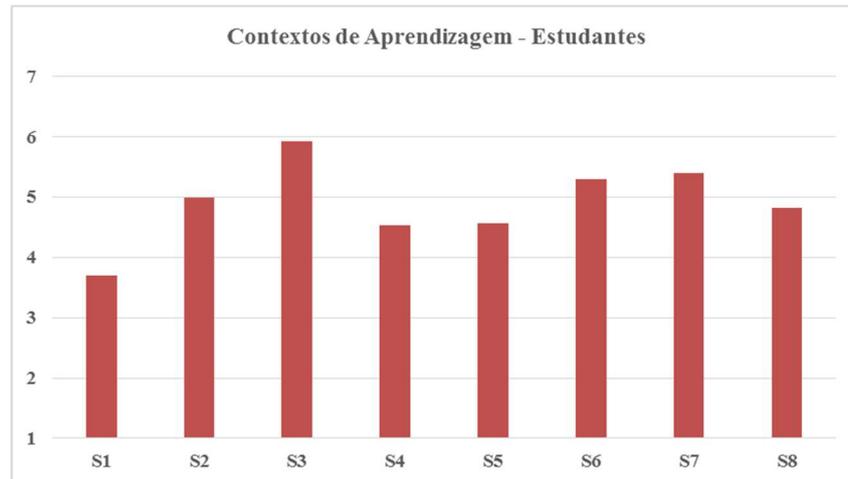


Gráfico 4 - Dificuldade nos Conceitos de Programação – Percepção dos Professores

3.6.3 Contextos de aprendizagem

Foram analisados os seguintes contextos de aprendizagem (Gráfico 5) e identificada a respetiva utilidade: nas aulas teóricas; em sessões de resolução de exercícios com os colegas; em sessões práticas de resolução de exercícios (aulas de laboratório); quando estudo sozinho; quando estudo/ trabalho sozinho na resolução de exercícios da disciplina; em sessões de esclarecimento de dúvidas com o docente (horário de dúvidas); em sessões de resolução de exercícios na aula teórica; em fóruns de discussão na Internet.

Observa-se que os contextos de aprendizagem com maior utilidade para os estudantes são: as sessões práticas de resolução de exercícios (aulas de laboratório) (S3); as sessões de resolução de exercícios com os colegas (S2); as sessões de esclarecimentos de dúvidas com o docente (S6); e as sessões de resolução de exercícios nas aulas teóricas (S7). Os contextos de aprendizagem com menor utilidade para os estudantes são as aulas teóricas (S1). Os contextos de aprendizagem percebidos como sendo de utilidade média, são: quando estuda sozinho (S4); quando estuda/trabalha na resolução de exercícios da disciplina (S5); e os fóruns de discussão na Internet (S8).

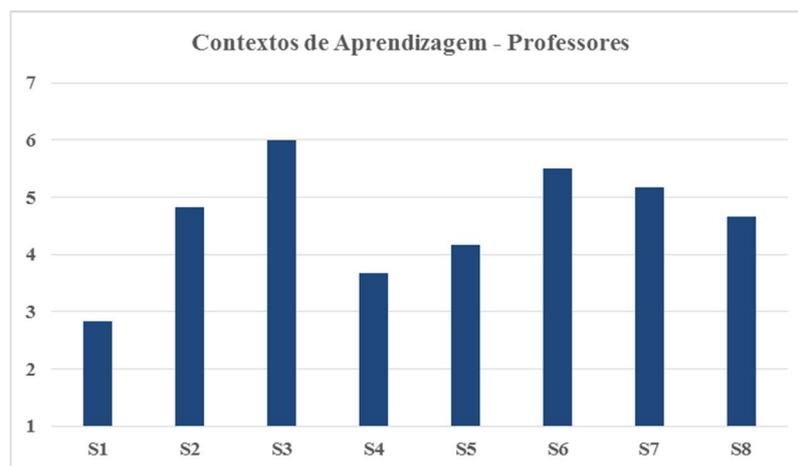


(1-pouco útil), (7-muito útil)

(S1) – Nas aulas teóricas; (S2) – Em sessões de resolução de exercícios com os colegas; (S3) – Em sessões práticas de resolução de exercícios (aulas de laboratório); (S4) – Quando estudo sozinho; (S5) – Quando estudo / trabalho sozinho na resolução de exercícios da disciplina; (S6) – Em sessões de esclarecimento de dúvidas com o docente (horário de dúvidas); (S7) – Em sessões de resolução de exercícios na aula teórica; (S8) – Em fóruns de discussão na Internet

Gráfico 5 – Utilidade dos Contextos de Aprendizagem – Percepção dos Estudantes

As aulas teóricas (S1) são o contexto de aprendizagem, percebido pelos professores, com menor utilidade para os estudantes (Gráfico 6).



(1-pouco útil), (7-muito útil)

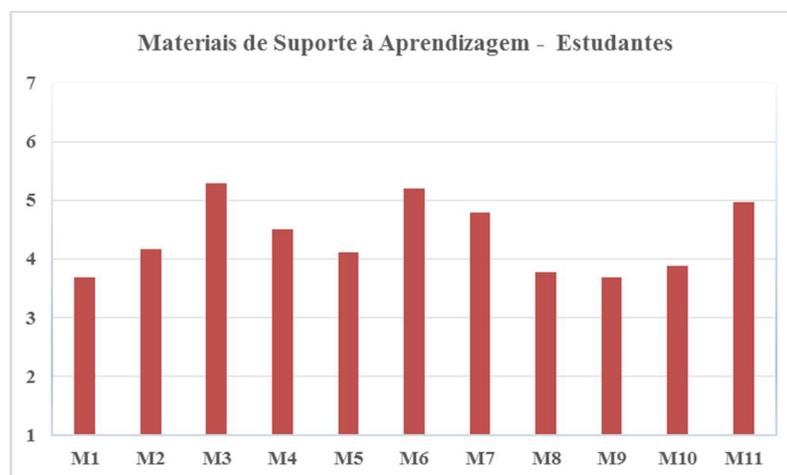
(S1) – Nas aulas teóricas; (S2) – Em sessões de resolução de exercícios com os colegas; (S3) – Em sessões práticas de resolução de exercícios (aulas de laboratório); (S4) – Quando estudo sozinho; (S5) – Quando estudo / trabalho sozinho na resolução de exercícios da disciplina; (S6) – Em sessões de esclarecimento de dúvidas com o docente (horário de dúvidas); (S7) – Em sessões de resolução de exercícios na aula teórica; (S8) – Em fóruns de discussão na Internet

Gráfico 6 – Utilidade dos Contextos de Aprendizagem – Percepção dos Professores

Os professores perceberam que os estudantes consideraram, quando estudo sozinho (S4), quando estudo/trabalho sozinho na resolução de exercícios da disciplina (S5), como contextos de aprendizagem de relativa utilidade, enquanto os fóruns de discussão na Internet (S8), se assumem como contextos de aprendizagem de utilidade média. Os contextos de aprendizagem que os professores consideram ser valorizados pelos estudantes, pela grande utilidade junto dos estudantes, são as sessões práticas de resolução de exercícios (aulas de laboratório) (S3), as sessões de esclarecimento de dúvidas com o docente (horário de dúvidas) (S6) e as sessões de resolução de exercícios na aula teórica (S7).

3.6.4 Materiais/recursos

Nos materiais/recursos de suporte à aprendizagem (Gráfico 7), verificou-se que os estudantes consideraram como tendo maior utilidade, os exemplos de programas completos (M3), os tutoriais disponíveis na Internet (M6) e os conteúdos disponibilizados na plataforma de *e-Learning Moodle* (M11).



(1-pouco útil), (7-muito útil)

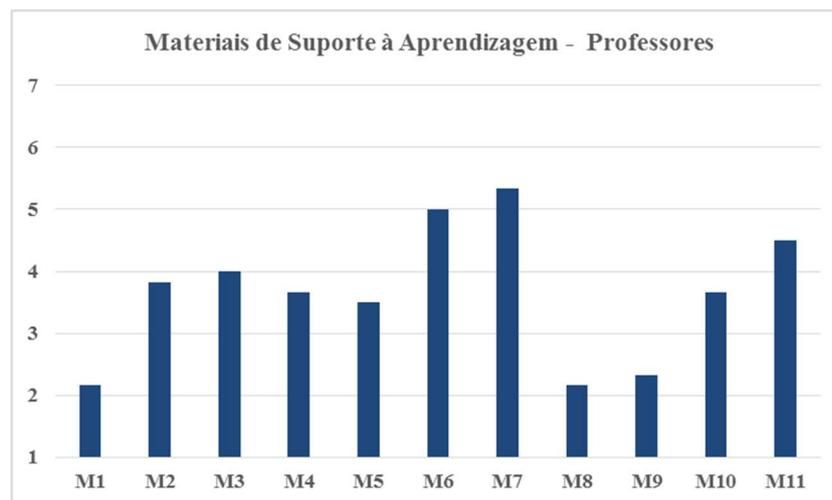
(M1) – Livro recomendado para a disciplina; (M2) – Apontamentos tirados nas aulas; (M3) – Exemplos de programas completos (código + executável); (M4) – Imagens estáticas de programas estruturados; (M5) – Visualizações interactivas (com ferramentas tipo Alice, BlueJ, Robocode ou outras similares); (M6) – Tutoriais disponíveis na Internet; (M7) – Vídeos educacionais disponibilizados no Youtube; (M8) – Twitter – Páginas relacionadas com a aprendizagem da programação; (M9) – Facebook – Páginas relacionadas com a aprendizagem da programação; (M10) – Cópias das transparências / Powerpoint; (M11) – Conteúdos disponibilizados na plataforma.

Gráfico 7 - Utilidade dos Materiais/Recursos – Percepção dos Estudantes

Os materiais/recursos considerados como sendo menos úteis pelos estudantes, são: o livro recomendado para a disciplina (M1); Facebook (páginas de ensino da programação) (M9); Twitter (twitees de ensino da programação) (M8); e cópias das transparências/ PowerPoint (M10). Como sendo de utilidade média, os estudantes percebem os apontamentos tirados

nas aulas (M2), as imagens estáticas de programas estruturados (M4) e os vídeos educacionais disponibilizados na Internet (M6).

Nos materiais de suporte à aprendizagem (Gráfico 8) percebidos pelos professores, como sendo de maior utilidade para a aprendizagem dos estudantes, identificaram-se os vídeos educacionais disponibilizados no *Youtube* (M7), tutoriais disponíveis na Internet (M6), conteúdos disponibilizados na plataforma de *e-Learning* (M11). Através da análise ao gráfico, verifica-se que os materiais/ recursos percebidos com sendo de menor utilidade, são o livro recomendado para a disciplina (M1), o Twitter (páginas relacionadas com a programação) (M8) e o Facebook (páginas relacionadas com a programação) (M9). São de utilidade percebida como intermédia, os apontamentos tirados nas aulas (M2), os exemplos de programas completos (M3) e as imagens estáticas de programas estruturados.



(1-pouco útil), (7-muito útil)

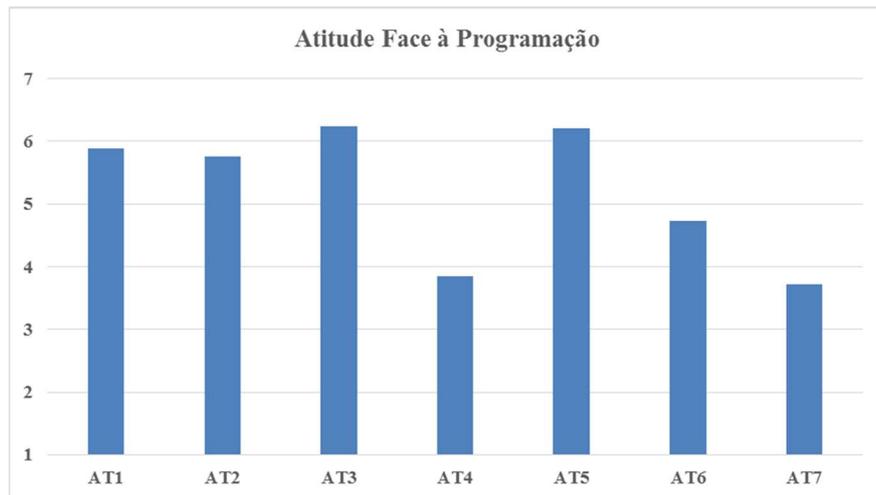
(M1) – Livro recomendado para a disciplina; (M2) – Apontamentos tirados nas aulas; (M3) – Exemplos de programas completos (código + executável); (M4) – Imagens estáticas de programas estruturados; (M5) – Visualizações interativas (com ferramentas tipo Alice, BlueJ, Robocode ou outras similares); (M6) – Tutoriais disponíveis na Internet; (M7) – Vídeos educacionais disponibilizados no Youtube; (M8) – Twitter – Páginas relacionadas com a aprendizagem da programação; (M9) – Facebook – Páginas relacionadas com a aprendizagem da programação; (M10) – Cópias das transparências / Powerpoint; (M11) – Conteúdos disponibilizados na plataforma

Gráfico 8 – Utilidade dos Materiais/Recursos – Percepção dos Professores

3.6.5 Atitude face à programação

Na atitude face à programação (Gráfico 9), foram tidos em conta os seguintes itens: gosto de aprender programação; gosto de programar; acho que programar vai ser útil na minha vida profissional; programar é fácil; acho que programar é útil na minha formação; invisto muito tempo na aprendizagem da programação; sei programar muito bem. Da análise aos resultados,

observa-se que os estudantes destacaram a importância que a programação pode ter na sua vida profissional e na sua formação. Observa-se também que existe uma diferença entre o gostar de aprender programação e o gostar de programar. Os estudantes gostam mais de aprender programação, do que propriamente programar. Por outro lado, os estudantes consideraram que a programação não é fácil e que não sabem programar muito bem.



(1-discordo totalmente), (7-concordo totalmente)

(AT1) – gosto de aprender programação; (AT2) – gosto de programar; (AT3) – acho que programar vai ser útil na minha vida profissional; (AT4) – programar é fácil; (AT5) – acho que programar é útil na minha formação; (AT6) – invisto muito tempo na aprendizagem da programação; (AT7) – sei programar muito bem

Gráfico 9 - Atitude dos Estudantes Face à Programação

3.7 Discussão de resultados

Da análise aos dados, constatou-se que os estudantes se mostraram confiantes na perceção das suas dificuldades, estando estes resultados em consonância com observações similares feitas por Lahtinen et al., (2005) Milne & Rowe (2002). Os estudantes acreditam perceber sobre um dado tópico, mas quando os resultados são avaliados, constata-se que a maioria tem excesso de confiança. Este excesso de confiança, pode ser um problema para o sucesso dos estudantes na aprendizagem da programação, dado que cria uma expectativa positiva sobre o conhecimento adquirido e quando confrontados com os exames e exercícios nos quais precisam de os aplicar, acabam por falhar. Confrontados com resultados menos positivos, os estudantes ficam desmotivados e conseqüentemente assiste-se a uma menor dedicação ao estudo, terminando mesmo eventualmente com a desistência da disciplina. Observam-se também nas situações de aprendizagem e nos conceitos de programação, que as situações e os conceitos percecionados como sendo os mais difíceis, são os que requerem dos estudantes um maior raciocínio abstrato.

Estes resultados estão também consonantes com os autores Lahtinen et al., (2005); Milne & Rowe (2002) e Piteira & Costa (2013, 2014). Conceitos de programação como ponteiros e referências, tipos abstratos de dados, tipos estruturados de dados, desenhar um programa para a resolução de uma determinada tarefa e dividir o programa em funcionalidades, são situações de aprendizagem e conceitos de programação que requerem raciocínio abstrato, como também uma compreensão e manipulação de várias “entidades”, para os quais os estudantes ainda não desenvolveram capacidades e treino específico. Milne & Rowe (2002) referem que os conceitos de programação, como os ponteiros, são usualmente fáceis de explicar, no entanto, são a sua implementação e o conseqüente comportamento durante a execução do programa, que causam os maiores problemas aos estudantes, no que diz respeito à sua compreensão. Observa-se da análise aos dados, que quando os estudantes necessitam de utilizar os conceitos de programação em conjunto, esta tarefa se revela como sendo difícil. Os estudantes parecem compreender os conceitos individualmente, mas quando têm que os aplicar em conjunto, manifestam dificuldade na sua utilização e na compreensão do modo como os devem aplicar. Estes resultados encontram-se alinhados com os resultados similares, reportados por Piteira & Costa (2013, 2012); Renumol et al., (2010) e Robins et al., (2003).

Na análise aos resultados obtidos nos contextos de aprendizagem, verificou-se que os estudantes privilegiam contextos de aprendizagem com componentes mais práticas, ao invés de contextos meramente teóricos. Inclusive, os estudantes consideram ser de maior utilidade, a aprendizagem que realizam quando estudam sozinhos, comparativamente com a aprendizagem que realizam nas aulas teóricas. Pode assim afirmar-se, tendo por base os resultados, que as aulas puramente teóricas não são atrativas e não envolvem os estudantes na aprendizagem, contribuindo mesmo para a sua desmotivação. Esta observação está suportada precisamente no contexto de aprendizagem percecionado como sendo de maior utilidade, como o são as sessões práticas de resolução de exercícios em laboratório e a resolução de exercícios práticos com apoio do docente. Os estudantes preferem a prática, pois implica aprender através do fazer. Através da observação da implementação algorítmica para a resolução de um dado problema e da sua execução, os estudantes podem contextualizar e fazer uma relação entre o real e o abstrato, permitindo assim uma melhor compreensão. Por outro lado, tendo em conta que quanto maior for a complexidade dos conceitos, maior será a necessidade do estudante em suportar a sua aprendizagem em conteúdos e materiais, que permitam uma compreensão em profundidade dos mesmos. Nesse sentido, materiais interativos e dinâmicos, serão facilitadores do processo de aprendizagem (Lahtinen et al., 2005). E é precisamente na utilidade percebida

referente aos materiais/conteúdos, que os estudantes preferem ter programas executáveis, como os tutoriais e os vídeos educativos na Internet, os quais são tidos também como sendo de grande utilidade no processo de aprendizagem. Contrariamente à confiança que os estudantes percecionam nos seus conhecimentos relativos aos conceitos de programação, quando questionados sobre se programar é fácil, mostram-se menos confiantes e consideram que programar não é fácil. Pode-se assim afirmar que globalmente, os estudantes têm a noção clara de que a temática em si é difícil, embora quando questionados sobre os conceitos individualmente, mostram-se confiantes na sua aprendizagem. Esta observação confirma em certa medida que os estudantes compreendem individualmente os conceitos, mas utilizá-los em conjunto na resolução de um determinado problema, torna-se mais difícil e é aí que na maioria das vezes, se deparam com as dificuldades. Quando questionados sobre o tempo que investem no estudo e na sua percepção de saberem programar muito bem, os estudantes reconhecem que não investem muito tempo na aprendizagem e que não sabem programar muito bem. Têm contudo uma noção muito clara, de que a programação é útil na sua formação e será também útil para a sua vida profissional.

3.8 Conclusões do Estudo 1

Neste capítulo, identificaram-se as dificuldades percecionadas pelos estudantes relativamente às situações de aprendizagem e conceitos de programação, assim como a a utilidade percebida referente aos materiais/conteúdos e contextos de aprendizagem. Neste capítulo, foram ainda identificadas, as percepções dos professores relativamente às dificuldades sentidas pelos estudantes nos conceitos de programação e em situações de aprendizagem, assim como a utilidade relativamente aos contextos de aprendizagem e materiais/recursos. Os conceitos de programação identificados como fáceis foram: variáveis e estruturas de seleção. Os conceitos considerados como mais difíceis foram: ponteiros, parâmetros, tipos estruturados de dados, tipos abstratos de dados e gestão de erros. As situações de aprendizagem consideradas como fáceis foram a utilização do ambiente de aprendizagem. Os que se identificaram como sendo os mais difíceis, foram: desenhar o programa para a resolução de uma determinada tarefa; dividir as funcionalidades em procedimentos; a aprendizagem da sintaxe da linguagem; e interpretar e resolver erros no programa. Os contextos de aprendizagem com maior utilidade para os estudantes são: sessões práticas de resolução de exercícios no laboratório; sessões de resolução de exercícios na aula teórica; sessões de dúvidas com o docente; e sessões de resolução de exercícios com os colegas. Os contextos identificados como sendo de menor utilidade, foram:

aulas teóricas; estudar sozinho; e quando estuda sozinho na resolução de exercícios da disciplina. Os materiais/recursos percebidos como sendo de maior utilidade para a aprendizagem, foram: exemplos de programas completos (código e executável); tutoriais na Internet; e vídeos educacionais disponíveis no *Youtube*. Os materiais/recursos identificados como sendo de menor utilidade, foram: o livro recomendado para a disciplina e as cópias das transparências/ acetatos.

Foram também identificadas as atitudes dos estudantes face à programação, mais concretamente: gosto de aprender a programar; gosto de programar; acho que programar vai ser útil na minha vida profissional; programar é fácil; acho que programar é útil para a minha formação; invisto muito tempo na aprendizagem da programação; e sei programar muito bem. Verificou-se que os estudantes têm a noção da dificuldade na aprendizagem da programação, que não investem muito tempo no estudo da programação e assumem que não sabem programar muito bem. Reconhecem que a programação será útil na sua formação, bem como na vida profissional, preferindo no entanto, aprender a programar, do que propriamente programar.

Tendo por base os dados analisados no presente estudo, no próximo capítulo, comparar-se-ão as perceções entre professores e estudantes e será efetuada a comparação entre as dificuldades nos conceitos de programação e os resultados efetivos das provas de avaliação (testes e exames).

4 COMPARAÇÃO DAS DIFICULDADES PERCECIONADAS NA APRENDIZAGEM DA PROGRAMAÇÃO – ESTUDO 2

4.1 Introdução

A partir dos dados obtidos no Estudo 1, através do qual foram identificadas as dificuldades percecionadas pelos estudantes na aprendizagem da programação e as percepções dos professores sobre a dificuldade dos estudantes, pretende-se, no presente estudo, efetuar a comparação entre as percepções dos estudantes e dos professores, comparando-as por fim, com os resultados obtidos pelos estudantes nas provas de avaliação. Finaliza-se o capítulo com as discussões dos resultados e as respetivas conclusões.

4.2 Objetivos

No presente estudo foram definidos como objetivos, a comparação das percepções entre estudantes e professores e a comparação das percepções destes, com os resultados efetivos obtidos pelos estudantes nas provas de avaliação (testes e/ou exames).

4.3 Abordagem metodológica

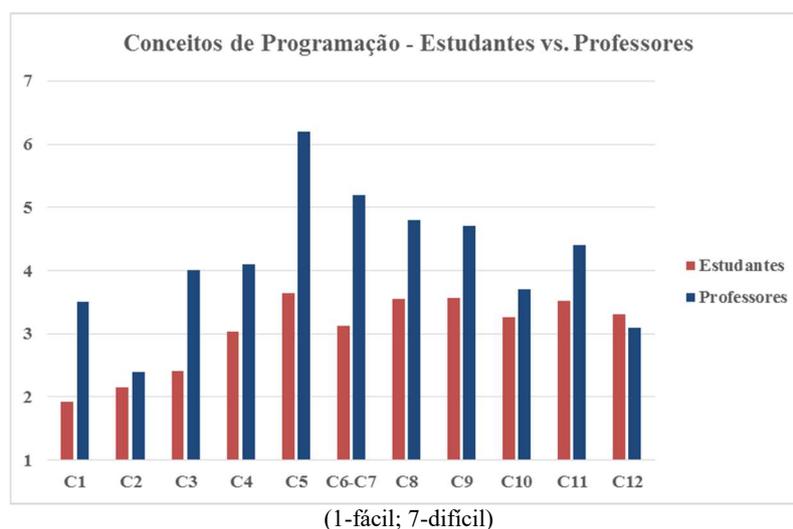
A abordagem metodológica adotada no presente estudo foi a seguinte: com base nos resultados do Estudo 1, onde se identificaram as percepções dos estudantes e dos professores, relativamente às situações de aprendizagem, contextos de aprendizagem, conceitos de programação e materiais/recursos, no presente estudo comparam-se as percepções entre professores e estudantes, em cada uma dessas dimensões. Foi ainda adicionado um terceiro elemento de comparação, nomeadamente, as provas de avaliação (teste e exames). Os dados recolhidos e tratados, referem-se aos anos letivos de 2011/2012 e 2012/2013. Relativamente às provas de avaliação, adotou-se o seguinte procedimento: (a) por cada uma das questões presentes nas provas de avaliação (testes ou exames), foram identificados os conceitos de programação presentes; (b) seguidamente, por cada questão, foram identificados os conceitos de programação e a percentagem obtida pelos alunos em cada um desses conceitos; (c) finalmente, foi efetuada a conversão da cotação obtida pelo aluno em percentagem, para uma escala de sete valores. No total, foram analisadas trezentas provas de avaliação.

4.4 Resultados

Neste subcapítulo, apresenta-se a comparação das dificuldades percebidas pelos professores e pelos estudantes, relativamente aos conceitos de programação e às situações de aprendizagem. Apresenta-se também a comparação da utilidade percebida, relativamente aos materiais/recursos e aos contextos de aprendizagem. Por último, efetua-se a comparação da dificuldade percebida, tanto por estudantes como por professores e comparam-se com os resultados efetivos das provas de avaliação.

4.4.1 Dificuldade nos conceitos de programação – Estudantes vs. Professores

Da análise aos resultados referentes à comparação das dificuldades nos conceitos de programação (Gráfico 10), verifica-se que os professores percebem que os estudantes sentem uma maior dificuldade na maioria dos conceitos. Os estudantes são mais confiantes nas suas dificuldades, percecionando um grau de dificuldade mais baixo que os professores. Verifica-se apenas uma alteração das perceções, no item C12, o qual é percebido pelos professores, como tendo um nível de dificuldade inferior ao percebido pelos estudantes. As diferenças nas perceções entre professores e estudantes, são mais visíveis nos conceitos variáveis (C1), nas tabelas (C5), nos parâmetros (C6-C7), nos tipos estruturados de dados (C8), nos tipos abstratos de dados (C9) e na utilização das bibliotecas da linguagem (C11).



(C1) – variáveis (C2) – estruturas de seleção; (C3) – estruturas de repetição; (C4) – tabelas; (C5) – ponteiros, referências; (C6) – parâmetros (valor); (C7) – parâmetros (referência); (C8) – tipos estruturados de dados; (C9) – tipos abstratos de dados; (C10) – manuseamento de entrada e saída de dados; (C11) – tratamento de erros; (C12) – utilização das bibliotecas da linguagem.

Gráfico 10 – Dificuldade nos Conceitos de Programação - Estudantes vs. Professores

4.4.2 Dificuldade nas situações de aprendizagem – Estudantes vs. Professores

Nas situações de aprendizagem (Gráfico 11), verifica-se a tendência observada nos conceitos de programação. Os professores percecionam um grau de dificuldade, maioritariamente superior ao percecionado pelos estudantes. Os professores percecionam que os estudantes manifestam uma maior dificuldade nas seguintes situações de aprendizagem: compreensão na estruturação do programa (I2); e desenhar o programa para a resolução de uma determinada tarefa (I4). Nas SA de utilização do ambiente de desenvolvimento (I1), aprendizagem da sintaxe da linguagem (I3) e interpretar e resolver erros no programa (I6), verifica-se que a perceção dos professores, continua a ser superior à perceção dos estudantes. É apenas na SA aprendizagem da sintaxe da linguagem (I3), que se verifica uma inversão na perceção, em que os estudantes entendem esta SA, como sendo ligeiramente mais difícil.

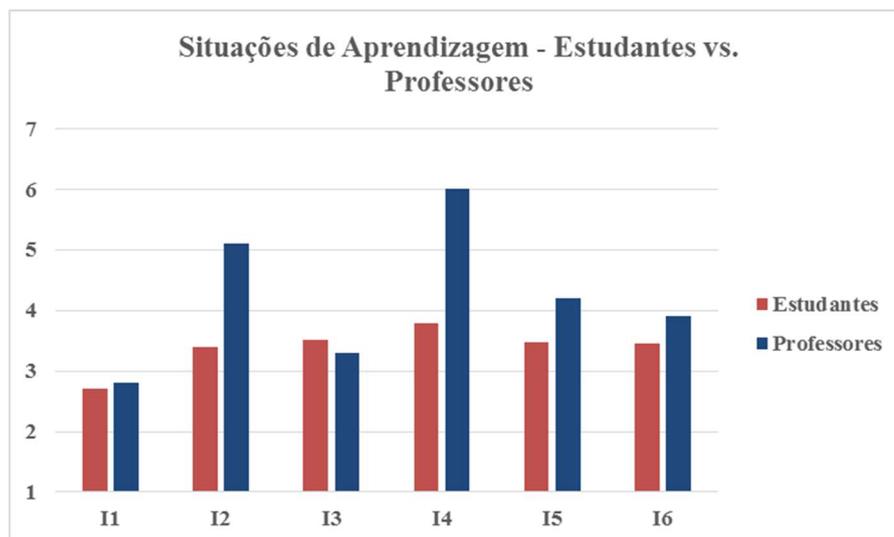


Gráfico 11 – Dificuldade nas Situações de Aprendizagem - Estudantes vs. Professores

4.4.3 Utilidade dos contextos de aprendizagem – Estudantes vs. Professores

Relativamente à utilidade dos contextos de aprendizagem (Gráfico 12), verifica-se que estudantes e professores têm a percepções mais próximas, no que diz respeito à utilidade das sessões práticas de resolução de exercícios nas aulas laboratoriais (S3), em sessões de resolução de exercícios com os colegas (S3), em sessões de resolução de exercícios na aula teórica (S8) e em fóruns de discussão na Internet (S8).

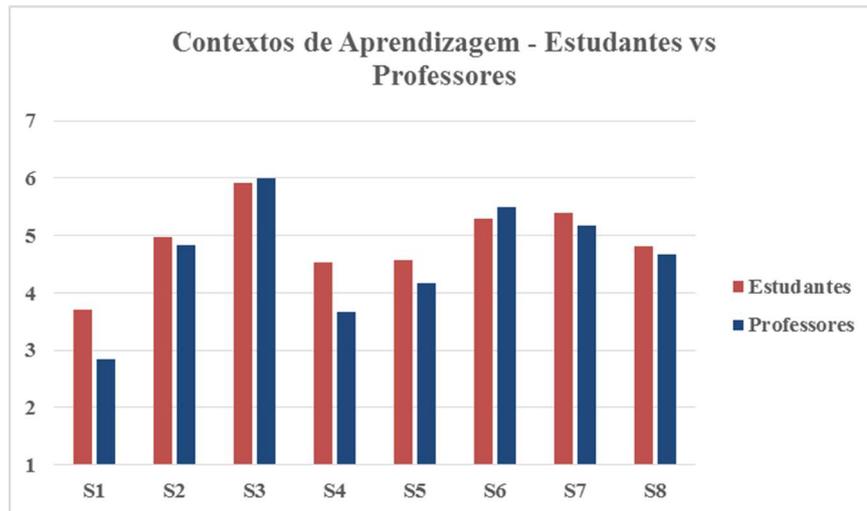


Gráfico 12 – Utilidade dos Contextos de Aprendizagem - Estudantes vs. Professores

4.4.4 Utilidade dos materiais/recursos – Estudantes vs. Professores

Após análise aos resultados, observa-se que os estudantes consideraram todos os materiais (Gráfico 13), como tendo maior utilidade, comparativamente à utilidade considerada pelos professores, excetuando os vídeos educacionais (M7). Este material é entendido pelos professores, com tendo uma utilidade superior, à utilidade percebida pelos estudantes.

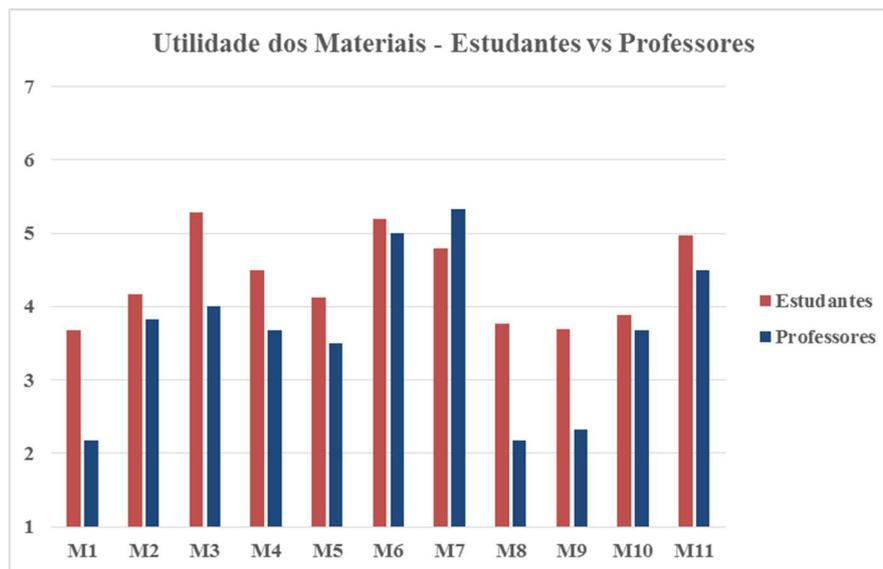
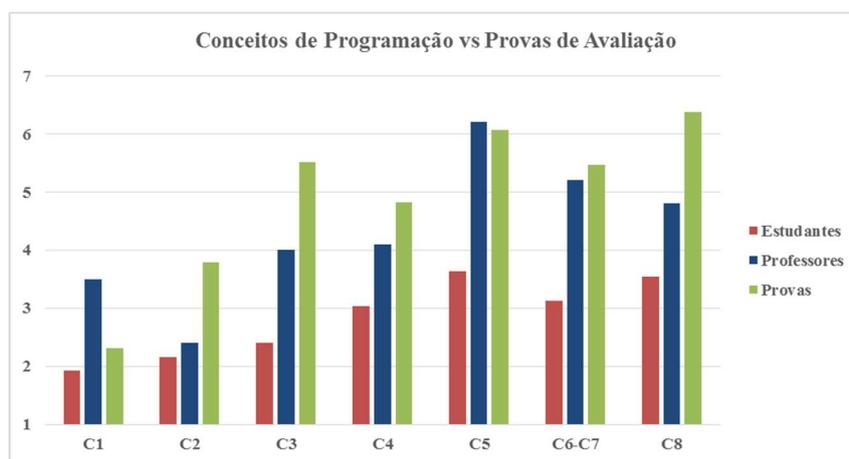


Gráfico 13 – Utilidade dos Materiais/Recursos - Estudantes vs. Professores

Verifica-se que os materiais percecionados pelos estudantes como tendo maior utilidade, são os exemplos de programas completos (código e executável), os tutoriais disponíveis na Internet, os vídeos educacionais disponibilizados no *Youtube* e os conteúdos disponibilizados na plataforma de *e-Learning*. Dos materiais considerados como tendo menor utilidade, destaca-se o livro recomendado para a disciplina (M1), com uma diferença significativa de perceção entre professores e estudantes. Curiosamente, verifica-se que os professores percecionam uma utilidade mais baixa, que a utilidade percecionada pelos estudantes. Por outro lado, salienta-se a diferença de perceção entre as cópias das transferências/*powerpoint* (M11) e os conteúdos disponibilizados na plataforma de *e-Learning*. Os estudantes consideram como tendo maior utilidade, os conteúdos disponibilizados na plataforma de *e-Learning*.

4.4.5 Dificuldade percecionada nos conceitos de programação vs. Resultados das provas de avaliação

Após a recolha, análise e tratamento dos dados recolhidos no presente estudo, efetuou-se a comparação com os resultados obtidos no Estudo 1. O resultado dessa comparação, pode ser observado no gráfico (Gráfico 14). Verifica-se que em todos os conceitos, com exceção do C1, os professores têm uma perceção mais elevada sobre a dificuldade dos estudantes. Verifica-se também na observação dos dados, que o grau de dificuldade nas provas de avaliação, é superior ao percecionado tanto pelos professores, como pelos estudantes, à exceção do conceito de programação variáveis (C1). Neste conceito, a dificuldade percecionada pelos professores, é superior aos resultados efetivos obtidos nas provas de avaliação.



(1-fácil; 7-difícil)

(C1) – variáveis (C2) – estruturas de seleção; (C3) – estruturas de repetição; (C4) – tabelas; (C5) – ponteiros, referências; (C6) – parâmetros (valor); (C7) – parâmetros (referência); (C8) – tipos estruturados de dados

Gráfico 14 – Dificuldade nos Conceitos de Programação – (Professores vs. Estudantes vs. Provas de Avaliação)

4.5 Análise e discussão de resultados

Da análise aos dados, observa-se que os estudantes se mostram confiantes na percepção das suas dificuldades. Esta diferença na percepção das dificuldades, pode resultar da ineficiente compreensão que os estudantes têm sobre o que sabem e o que deveriam na realidade saber, não tendo uma noção exata das dificuldades. Em contrapartida, os professores têm uma noção mais próxima das dificuldades, advindo por um lado, dos resultados que os estudantes obtêm nas provas de avaliação e por outro, da sua experiência no contacto direto com os estudantes em sala de aula. Esta conclusão é suportada pela análise ao Gráfico 10. Observa-se que as provas de avaliação em todas as dimensões, apresentam valores mais próximos de muito difícil. Aproximam-se desses valores, as percepções dos professores, ficando mais distantes as percepções dos estudantes, confirmando assim, que os estudantes são mais confiantes nas suas dificuldades. Estes resultados estão em consonância com as observações feitas por Lahtinen et al., (2005) e Milne & Rowe (2002), em que os estudantes acreditavam perceber sobre um dado tópico, mas quando analisadas as questões dos exames ou inquiridos pelos professores, constatou-se que estavam errados na sua confiança. Esta percepção pode ser um problema para o sucesso dos estudantes na aprendizagem da programação, dado que cria autoconfiança sobre o conhecimento adquirido, pois quando confrontados com os exames e com os exercícios, nos quais precisam de aplicar esse conhecimento, acabam por falhar. Uma quebra na autoconfiança, poderá contribuir para que o estudante abandone a disciplina, se desmotive e dedique menos tempo de estudo à programação de computadores.

Verificou-se, tanto nas situações de aprendizagem (Gráfico 2), como nos conceitos de programação (Gráfico 1), que as dimensões percecionadas como mais difíceis, são aquelas que necessitam de uma compreensão mais abstrata. Estes resultados têm como suporte os trabalhos dos autores Lahtinen et al., (2005) e Milne & Rowe (2002). Conceitos como ponteiros e referências, tipos abstratos de dados, tipos estruturados de dados, desenhar um programa para a resolução de uma determinada tarefa ou dividir o programa em funcionalidades, são situações de aprendizagem e conceitos de programação, que tanto requerem um raciocínio abstrato, como também uma compreensão e manipulação de várias “entidades”, para os quais os estudantes ainda não desenvolveram capacidades e treino específico. Milne & Rowe (2002) referem que conceitos como os ponteiros, são usualmente fáceis de explicar, no entanto, são a sua implementação e o conseqüente comportamento durante a execução do programa, que causam os maiores problemas aos estudantes, na sua compreensão. Verificou-se também que em situações onde os estudantes devem utilizar os conceitos de programação em conjunto, o grau

de dificuldade é percebido como sendo maior. Os estudantes parecem compreender os conceitos individualmente, mas quando os devem utilizar em conjunto, manifestam dificuldade na sua utilização e na compreensão do modo de como os devem de utilizar.

Assim, em resposta à questão de investigação Q1, baseados na análise dos dados, pode-se afirmar que as maiores dificuldades percebidas pelos estudantes nas situações de aprendizagem, foram a compreensão na estruturação do programa, a aprendizagem da sintaxe da linguagem e o desenhar o programa para a resolução de uma determinada tarefa, enquanto as maiores dificuldades percebidas nos conceitos de programação, foram os ponteiros e referências, os tipos estruturados de dados e as tabelas.

Da análise aos resultados obtidos nos contextos de aprendizagem (Gráfico 3), verificou-se que os estudantes preferem sessões práticas, em alternativa às sessões meramente teóricas. Inclusive, os estudantes mostram-se mais confiantes na aprendizagem que fazem quando estudam sozinhos, ao invés da aprendizagem realizada em aulas teóricas. Pode-se assim afirmar, tendo por base os resultados, que as aulas meramente expositivas não contribuem para uma efetiva aprendizagem da programação.

Ao relacionar-se os resultados obtidos nos contextos de aprendizagem, com os materiais, verificou-se precisamente que os estudantes preferem ter programas completos (código e executável), como material de estudo. Pode-se afirmar, tendo por base estas observações, que os estudantes preferem aprender através do fazer. Estas observações têm suporte nos trabalhos de Robins et al., (2003), verificando-se que tanto estudantes como professores, estão em concordância neste ponto.

Tendo em consideração que um dos aspetos que contribui para as dificuldades na aprendizagem, é a crescente complexidade que alguns dos conceitos de programação apresentam, assim como a utilização conjunta de diferentes conceitos, os materiais/recursos disponibilizados aos estudantes podem contribuir para ultrapassar essas dificuldades, se tiverem em consideração a visualização e a interatividade no sentido de lhes permitir, a criação de um modelo mental adequado e correto na sua compreensão, ajudando à resolução dos problemas.

Na percepção da utilidade dos materiais/recursos, observa-se que os estudantes preferem os programas executáveis, os tutoriais e os vídeos educativos na Internet (Gráfico 4). Esta percepção confirma, que o suporte audiovisual e a interatividade nos materiais influenciam positivamente, a motivação do estudante para a aprendizagem da programação.

Por último, na análise efetuada aos dados nos contextos de aprendizagem e materiais/recursos, pode-se afirmar que os contextos percecionados como tendo maior utilidade, são as sessões práticas de resolução de exercícios-laboratório, as sessões de resolução de exercícios-aula teórica, as sessões de esclarecimento de dúvidas com o docente e as sessões de resolução de exercícios com os colegas. Nos materiais/recursos, foram percecionados como sendo mais úteis, os exemplos de programas completos (código e executável), os tutoriais disponíveis na Internet, os vídeos educacionais disponibilizados no *Youtube* e os conteúdos disponibilizados na plataforma Moodle.

4.6 Conclusões do Estudo 2

Neste capítulo, efetuaram-se as comparações das perceções entre professores e estudantes sobre as dificuldades verificadas na aprendizagem da programação, nomeadamente, nas situações de aprendizagem, nos conceitos de programação, nos contextos de aprendizagem e nos materiais/recursos de suporte à aprendizagem. Foram também realizadas as comparações entre as perceções dos estudantes e dos professores, com as provas de avaliação, relativamente aos conceitos de programação.

Tendo por base os resultados, pode-se concluir que os estudantes são mais confiantes nas suas perceções das dificuldades, quando comparados esses resultados, com os resultados das perceções dos professores e os resultados das provas de avaliação. Os estudantes manifestam uma dificuldade acrescida em conceitos mais abstratos como ponteiros, referências e tipos abstratos de dados. A compreensão dos conceitos individualmente, parece não ser uma dificuldade para os estudantes, contudo, verifica-se que quando é necessário utilizar os conceitos em conjunto, esta situação é percecionada como tendo uma dificuldade acrescida. A vertente prática da aprendizagem, destaca-se na maior utilidade percebida por estudantes e professores, relativamente a alguns dos itens mais práticos nos contextos de aprendizagem. A componente da proximidade entre estudante e professor é também destacada como útil. Os estudantes consideram que estar com o docente a esclarecer dúvidas, é útil para a sua aprendizagem na programação.

Os materiais/recursos percebidos com tendo maior utilidade, por estudantes e por professores, são os tutoriais disponibilizados na Internet, os vídeos educacionais disponibilizados no *Youtube*, os conteúdos disponibilizados na plataforma institucional de *e-Learning* e os exemplos de código completo (código e executável). Conclui-se assim, que os estudantes

valorizam conteúdos que possam ser facilmente acedidos e com uma vertente mais dinâmica e interativa, características que os conteúdos como os tutoriais e os vídeos, possuem.

Através da realização e da conclusão do presente estudo e em resposta à questão de investigação “*Q1 – Quais as dificuldades dos estudantes na aprendizagem da programação?*”, identificou-se que as maiores dificuldades sentidas pelos estudantes, são a compreensão na estruturação do programa, a aprendizagem da sintaxe da linguagem e o desenhar o programa para a resolução de uma determinada tarefa, sendo que as maiores dificuldades sentidas pelos estudantes, no que diz respeito aos conceitos de programação, foram os ponteiros e as referências, os tipos estruturados de dados e as tabelas.

Tendo por base a revisão da literatura e os estudos, anteriormente realizados, no próximo capítulo pretende-se propor um modelo concetual teórico, para cursos *online* gamificados.

5 PROPOSTA E OPERACIONALIZAÇÃO DE *FRAMEWORK* TEÓRICA-CONCETUAL GAMIFICADA PARA CURSOS DE APRENDIZAGEM ONLINE DE PROGRAMAÇÃO – ESTUDO 3

5.1 Introdução

No presente capítulo, propõe-se uma *framework* concetual para cursos *online* gamificados de programação. Nesse sentido, para a realização deste estudo de investigação, foi definido como objetivo, a identificação dos elementos/ características de gamificação, que são utilizados em contexto educacional. Consequentemente, foram definidas duas questões de partida para este estudo: 1) Quais os elementos/ características de gamificação, que são utilizados em contexto educacional? E quais os que são utilizados especificamente na aprendizagem da programação *online*?

Pretende-se também operacionalizar a *framework* teórica através do desenho e da criação de um curso *online* de fundamentos de programação.

5.2 Objetivo

Tendo por base as questões de partida para o presente estudo, foram definidos os seguintes objetivos: i) propor uma *framework* teórica concetual gamificada para cursos *online* de programação; ii) implementar a *framework* teórica concetual gamificada; iii) identificar as perceções dos estudantes em relação aos elementos de gamificação utilizados no desenho do curso *online*.

5.3 Abordagem metodológica

A abordagem metodológica adotada no presente estudo, foi orientada pelo modelo de processo da *Design Science Research Methodology* (DSRM), proposto pelos autores (Vaishnavi & Kuechler, 2004) (Figura 9). O modelo de processo é composto por seis atividades: identificar o problema e a motivação; definir objetivos da solução; *design* e desenvolvimento; demonstração; avaliação; e comunicação. Seguidamente, descrevem-se as atividades do modelo de processo.

Fase 1 - Identificar o problema e a motivação: Nesta atividade deve-se especificar o problema de investigação e justificar o valor da solução proposta. A definição do problema, será posteriormente utilizada para o desenvolvimento do artefacto e fornecer uma solução efetiva, que poderá ser útil para especificar concetualmente o problema, para que a solução reflita a sua complexidade. A justificação do valor da solução permite alcançar, por um lado, a motivação do investigador e instigar a comunidade académica a perseguir a solução, aceitar os resultados e ajudar a compreender o raciocínio associado à compreensão do problema pelo investigador. Os recursos necessários para esta atividade, incluem o conhecimento do estado da arte do problema e a importância da sua solução. Após a identificação do problema, os passos seguintes são a definição dos objetivos para a solução, surgindo então a Fase 2.

Fase 2 – Definir os objetivos da solução: a partir da definição de um problema e do conhecimento do que é possível e factível, devem inferir-se os objetivos da solução. Os objetivos podem ser quantitativos, i.e., em termos do desejável que uma solução possa ser melhor que uma já existente, ou qualitativa, i.e., a descrição de como um novo artefacto pode suportar soluções para problemas que não foram ainda resolvidos. Os objetivos deverão ser inferidos racionalmente a partir da especificação do problema. Os recursos requeridos incluem o conhecimento do estado da arte, as soluções correntes e a sua eficácia.

Fase 3 – *Design* e desenvolvimento: criação do artefacto. Os artefactos são potencialmente dimensões, modelos, métodos ou instanciações. Concetualmente, a investigação do *design* do artefacto pode ser qualquer objeto, no qual a contribuição da investigação, esteja embebida no *design*. Nesta atividade, determinam-se a funcionalidade e a arquitetura do artefacto e finalmente a sua criação. Os recursos necessários para avançar para a atividade seguinte, incluem o conhecimento teórico que suporta a solução.

Fase 4 – Demonstração: demonstração da utilização do artefacto na resolução de uma ou mais instâncias do problema. Pode ser demonstrada através de experimentação, simulação, estudo de caso, prova de conceito ou outra atividade adequada. Os recursos requeridos para a demonstração, incluem o conhecimento efetivo de como utilizar o artefacto para resolver o problema. Após demonstração, é necessário avaliar se o artefacto é a solução para o problema, seguindo a Atividade 5.

Fase 5 – Avaliação: esta atividade tem como finalidade observar e medir, de modo a avaliar se o artefacto suporta a solução do problema. Envolve comparar objetivos da solução, com os

resultados observados na demonstração do artefacto. Requer conhecimento relevante de métricas e técnicas de análise. Dependendo da natureza do artefacto, a avaliação poderá assumir diversas formas. Poderá incluir a comparação das funcionalidades do artefacto com os objetivos da solução, medidas de desempenho quantitativas, como inquéritos de satisfação, retorno do utilizador ou simulações. Concetualmente, pode incluir qualquer evidência empírica ou lógica. No final da atividade, os investigadores podem decidir realizar a iteração do processo e voltar atrás para melhorar a eficiência do artefacto ou continuar para a atividade seguinte, de forma a proceder à comunicação dos resultados.

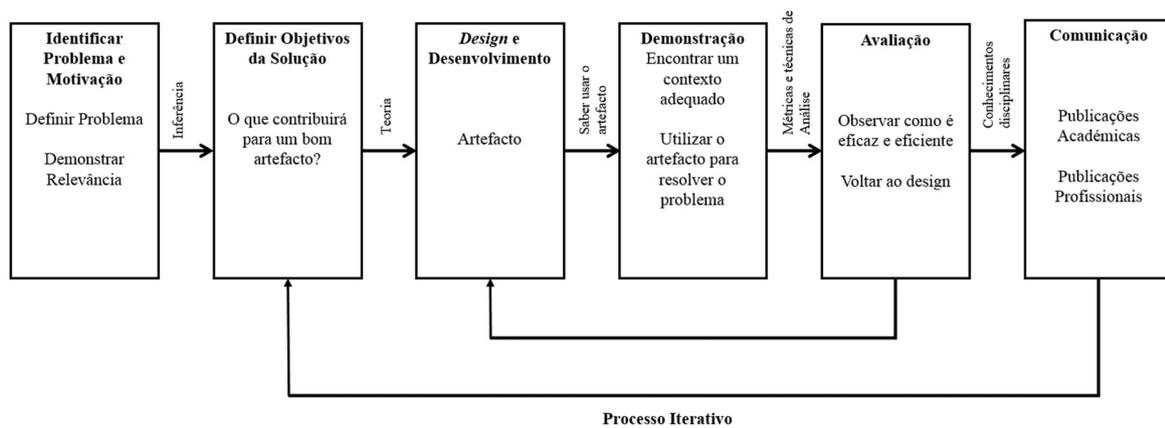


Figura 9 - Modelo de Processo da *Design Science Research Methodology (DSRM)*, adaptado de Vaishnavi & Kuechler (2004)

Fase 6 – Comunicação: Nesta fase, devem-se comunicar o problema e a sua importância, o artefacto, a sua utilidade e a sua inovação, o rigor do processo de *design*, a sua eficácia para os investigadores e outras audiências relevantes. Os investigadores podem utilizar a estrutura deste processo para a estruturação do artigo científico, adotando assim uma estruturação de investigação empírica (definição do problema, revisão da literatura, desenvolvimento das hipóteses, recolha de dados, análises, discussão e conclusão). A comunicação requer conhecimento da cultura disciplinar.

As fases metodológicas seguiram o modelo de processo DSRM, identificando-se para cada uma das seis fases que compõem o modelo, as diversas subfases realizadas (Figura 10) e descritas seguidamente.

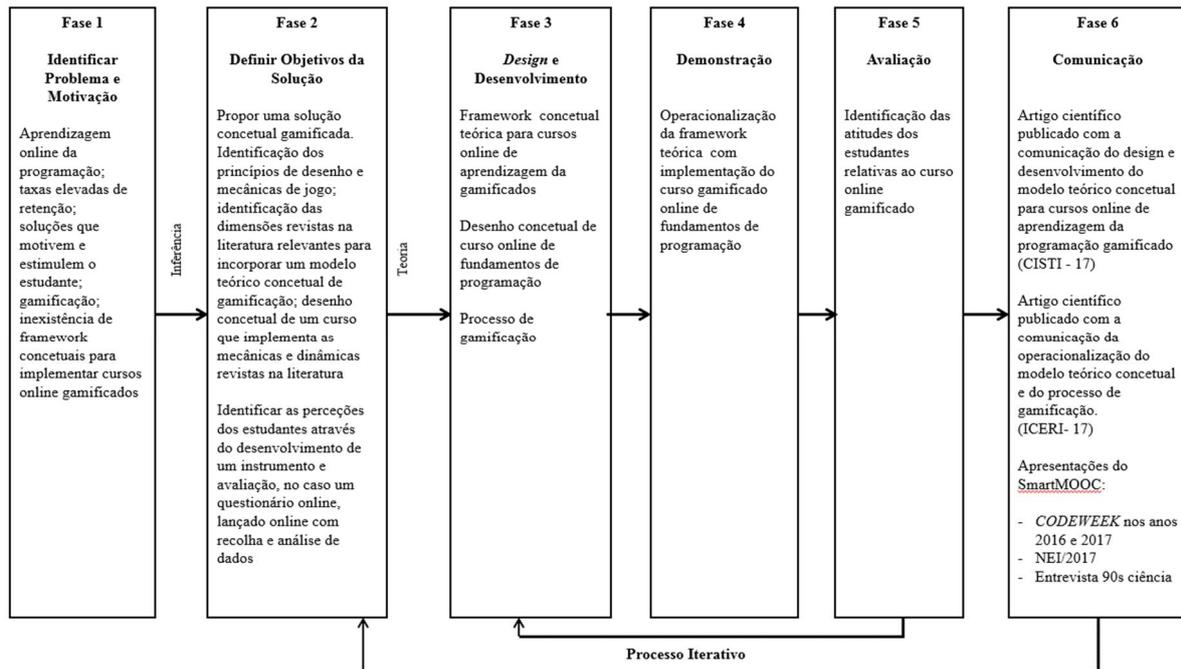


Figura 10 – Fases de *Design Science Research Methodology*

Nesse sentido, o problema identificado a partir da revisão da literatura, foram as taxas de reprovação e de desistência na aprendizagem da programação, assim como a desmotivação dos estudantes para a aprendizagem desta temática, quando não obtêm os resultados esperados. As taxas de reprovação são uma das evidências, de que os estudantes enfrentam dificuldades diversas, no decorrer da sua aprendizagem. Assim, a partir da identificação deste problema, procuraram-se soluções que incentivem e estimulem os estudantes à aprendizagem e ao contínuo contacto com os conteúdos e atividades, as quais sejam facilitadoras de uma experiência rica e envolvente na aprendizagem. Identificou-se a gamificação como solução emergente, dado que utiliza características de jogos, num contexto de não jogo. Estando a aprendizagem a tornar-se cada vez mais suportada por ferramentas tecnológicas, que permitem ao estudante aceder e realizar a sua aprendizagem a distância, identificou-se como necessidade, a existência de uma *framework* que oriente os educadores a desenhar cursos gamificados para a aprendizagem da programação *online*. A partir da identificação do problema, encontrou-se a motivação científica para a realização do estudo e definiram-se os objetivos gerais, mais concretamente, a proposta de framework conceitual gamificada, a identificação dos princípios de desenho, as mecânicas de jogo e a identificação das dimensões relevantes na literatura, para incorporarem a *framework* de gamificação. Foram estabelecidos como objetivos, o desenho conceitual do curso gamificado de programação e a recolha das percepções dos estudantes face

aos elementos de gamificação presentes no curso *online*, através de um instrumento de recolha de dados (questionário). Seguidamente, iniciou-se a atividade de *design* e respetivo desenvolvimento. Esta atividade foi desenvolvida com base num suporte teórico profundo e sólido, o qual permitiu a construção teórica-concetual da framework para os cursos *online* gamificados de programa. A partir da framework teórica desenvolvida, iniciou-se o desenho concetual do curso e do processo de gamificação. Após a conclusão desta atividade, passou-se para a fase da demonstração. Nesta atividade foi realizado todo o processo relacionado com a operacionalização da framework. Para a concretização da operacionalização da framework, foi desenvolvido um curso *online* de fundamentos de programação através da plataforma de aprendizagem online *Moodle*. Finalizada esta atividade, prosseguimos para a atividade de avaliação. Nesta atividade, foi desenvolvido um instrumento de recolha de dados (questionário), através do qual se recolheram as atitudes dos estudantes, face aos elementos de gamificação utilizados no curso *online* de fundamentos de programação. Por último, na atividade de comunicação e no contexto deste estudo, foram publicados dois artigos científicos e realizadas apresentações em eventos relacionados com a programação, em seminários e em aulas abertas.

Nos subcapítulos seguintes descrevem-se em detalhe as atividades que foram realizadas nas seis fases da metodologia.

5.4 Fase 1 e Fase 2 - Enquadramento teórico

A gamificação tem sido apontada como uma abordagem que tem o potencial de aumentar o envolvimento do estudante na aprendizagem, enquanto fornece retorno dessa aprendizagem (Costa et al., 2017; Domínguez et al., 2013; Fotaris et al., 2016; Ibáñez et al., 2014; Piteira & Costa, 2017; Piteira & Costa, 2017; Seaborn & Fels, 2015). Sendo a aprendizagem da programação, uma temática que é tida e percebida como difícil e que requer esforço e dedicação dos estudantes, as abordagens como a gamificação, que integra elementos com características de jogos, pode contribuir para incentivar e envolver os estudantes com os conteúdos e com as atividades de aprendizagem. O potencial de motivação e envolvimento que a utilização da gamificação representa para a educação, tem despertado um crescente interesse por parte dos investigadores e dos educadores. Esse interesse é evidente, quando observado o número de publicações científicas relacionadas com a temática da gamificação na educação. Apesar da gamificação ter esta atenção recente por parte dos investigadores, ainda é uma temática a

necessitar de mais estudos de investigação, que aprofundem a efetividade e a adoção da gamificação em ambientes de aprendizagem (Dicheva et al., 2015).

Verifica-se que muitos estudos descrevem a avaliação da gamificação, relativamente a ferramentas ou protótipos desenvolvidos para o efeito, sendo contudo escassos, os estudos que explorem e avaliem a utilização da gamificação nas atividades regulares de aprendizagem *online*, suportadas por uma plataforma de *e-Learning* (Dicheva et al., 2015). Nos últimos anos temos assistido a uma mudança de paradigma na educação. As universidades passaram a incorporar novas modalidades de aprendizagem: *e-Learning* e o *b-Learning*. O *e-Learning* permite ao estudante realizar toda a sua aprendizagem à distância suportada por ferramentas *online*. O *b-Learning* apresenta-se como uma modalidade mista, requerendo que o estudante faça a sua aprendizagem presencial, com complementos da aprendizagem *online*. Estas plataformas tornaram-se assim um suporte essencial a todo o processo educativo, introduzindo uma componente de inovação, através das quais são disponibilizados os conteúdos, definidas as atividades de aprendizagem e as atividades de avaliação. Nesse sentido e tendo em conta o papel que a gamificação pode ter no envolvimento do estudante, com a aprendizagem e a integração destas plataformas como suporte ao processo de ensino-aprendizagem, importa explorar e estudar a adoção da gamificação nestes ambientes *online* de aprendizagem. Contudo, incorporar a gamificação num curso online, coloca-me algumas questões, tais como: "Quais as mecânicas de jogo a utilizar?"; "Quais as dinâmicas de jogo que devem ser aplicadas?"; "Que tipo de gamificação devemos utilizar, estrutural ou de conteúdo?".

De acordo com os autores (Hamari, Koivisto, & Sarsa, 2014; Hamari, 2017; Landers, Bauer, & Callan, 2017), a gamificação tem a capacidade de envolver os utilizadores, sendo consensual que se for bem desenhada e usada corretamente, pode influenciar positivamente a aprendizagem. No entanto, a maioria dos estudos que têm abordado a temática da gamificação, não são suportados pela teoria e não utilizam *frameworks* teórico-conceituais de suporte ao desenho do sistema gamificado (Seaborn & Fels, 2015). Quando analisamos um pouco mais em detalhe a literatura, verifica-se a existência de *frameworks* e.g. (García, Pedreira, Piattini, Cerdeira-Pena, & Penabad, 2017; Rodrigues, Costa, & Oliveira, 2016), contudo, essas não são específicas do domínio da aprendizagem da programação e não são aplicáveis a cursos *online* gamificados.

Nesse sentido e após a revisão da literatura, propomos uma *framework* (Figuras 11 e 12) para cursos online de aprendizagem da programação. No subcapítulo seguinte, apresenta-se a *framework* teórica proposta.

5.5 Fase 3: *Design* e desenvolvimento – *Framework* - Dimensões

Neste subcapítulo, apresentam-se e descrevem-se as dimensões que compõem a *framework* para cursos *online* de aprendizagem da programação, assim como, as teorias que as suportam. A *framework* é uma generalização teórica (Carrol & Swatman, 2000; Lee & Baskerville, 2003) resultante da revisão da literatura, em aprendizagem da programação e em gamificação na educação. A *framework* teórica proposta, incorpora as dimensões revistas na literatura. As dimensões são: a) público-alvo; b) objetivos gerais; objetivos específicos e tópicos; c) conteúdos; d) princípios de desenho educacional; e) mecânicas de jogo; f) absorção cognitiva; g) *flow*; e h) personalidade. Seguidamente, descrevem-se as dimensões que compõem a *framework* proposta.

A caracterização do **público-alvo** serve de suporte para a definição dos objetivos gerais, resultados esperados e os tópicos que compõem um curso (Lederman & Abell, 2014).

É comumente aceite aquando da estruturação de um curso, a identificação prévia dos **Objetivos Gerais**, dos **Resultados Esperados** e dos **Tópicos** que serão lecionados, de modo a dar suporte aos objetivos gerais e aos resultados esperados. Esta aceitação é suportada pelas teorias da educação, e.g., teoria de Bloom (1956). Bloom definiu uma taxonomia designada de domínio cognitivo, estruturada em níveis de complexidade crescente. Essa complexidade crescente, significa que o estudante apenas deve adquirir uma nova competência depois de adquirir e dominar a competência que a antecede. Nesse sentido, a taxonomia apresenta um esquema para classificação e uma organização hierárquica dos processos cognitivos, de acordo com níveis de complexidade e objetivos do desenvolvimento cognitivo desejado e planeado. Os processos categorizados pela taxonomia dos objetivos cognitivos, de *Bloom*, além de representarem os resultados de aprendizagem esperados, são cumulativos, o que caracteriza uma relação de dependência entre os níveis e são organizados em termos de complexidades dos processos mentais (Bloom et al., 1956).

Nesse sentido e com base na teoria de *Bloom* e no *Computer Science Curricula da ACM*, publicado em 2013, consideraram-se estas dimensões na framework (Joint Task Force on Computing Curricula & Society, 2013).

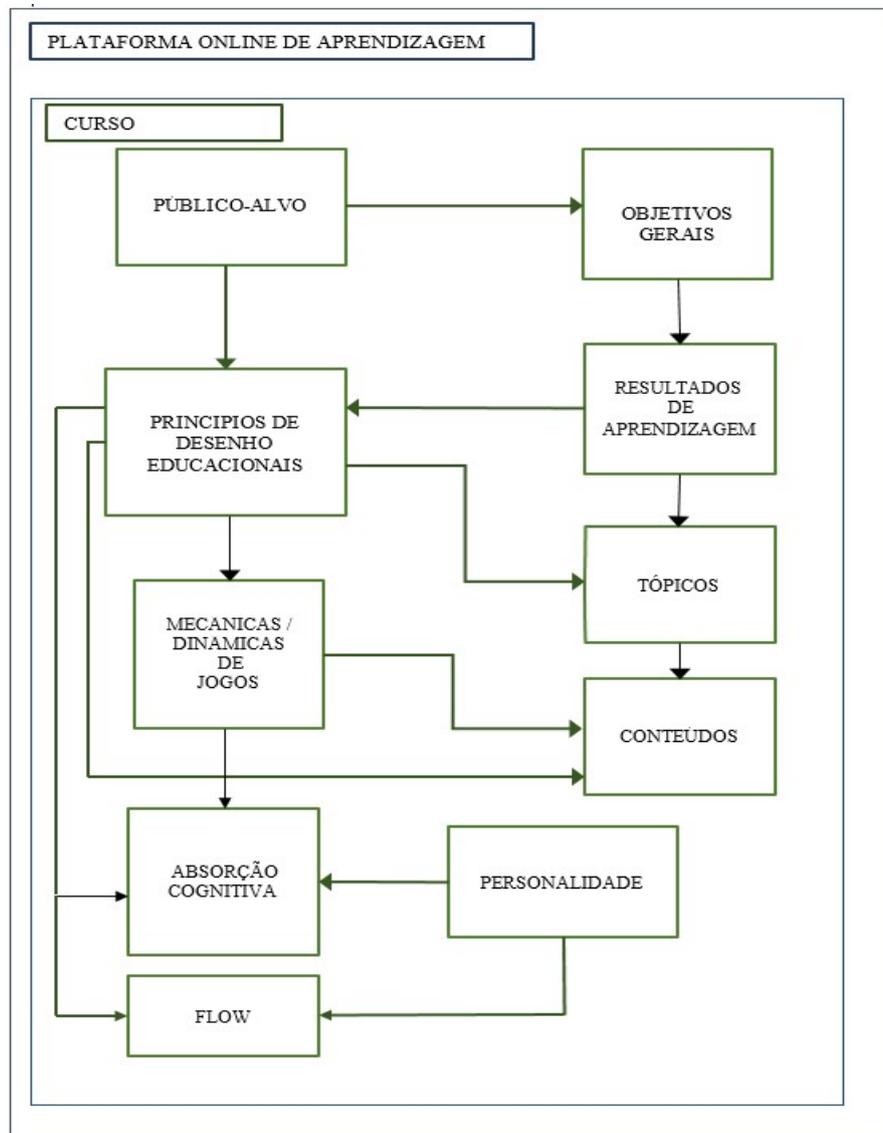


Figura 11- Dimensões da *Framework* Teórica

Numa plataforma *online* de aprendizagem, são diversas as tecnologias que suportam as estratégias educacionais. Oliver & Herrington (2003), construíram uma *framework* composta por elementos tecnológicos agrupados em três áreas da aprendizagem: recursos, suportes e atividades. Tendo por base a framework de Oliver & Herrington, Aparício et al. (Aparício, Bacao, & Oliveira, 2016), sumariza a relação entre as estratégias educacionais e as tecnologias e descrevem na sua framework de sistemas de aprendizagem *online* e a distância, as tecnologias,

que num ambiente *online* e a distância, suportam as várias estratégias educacionais. Tendo por base o trabalho destes autores, integrou-se a dimensão **Conteúdos**, na *framework*.

Dicheva et. al. (2015), definiram uma taxonomia de dois níveis para classificarem os diversos trabalhos de investigação relacionados com gamificação na educação. Essa taxonomia resultou da agregação da classificação de quatro níveis, propostos por *Deterding's*. Os níveis propostos por Dicheva são: princípios de desenho educacional gamificado e mecânicas de jogo. Tendo em consideração que o objetivo traçado para a elaboração da *framework* que se pretende propor, é servir de suporte à implementação de um curso *online*, aplicando um conjunto de elementos de jogo, as classificações de Dicheva et al. (2015), estão alinhadas com esse objetivo. Nesse sentido, adotou-se a classificação dos autores na *framework*. De seguida, descreve-se a classificação adotada.

Princípios de Desenho Educacional Gamificado. Após a identificação do público-alvo, objetivos gerais e resultados esperados, será necessário desenhar toda a estrutura do curso e planear de que forma o estudante interagirá com os conteúdos e com as atividades no curso *online*. Esta dimensão integra um conjunto de princípios que são utilizados no domínio da educação e não são específicos do desenho de jogos, sendo que muitos destes princípios, têm sido utilizados em sistemas instrucionais desde que eles existem (Dicheva et al., 2015). Estes princípios de desenho, são aplicados nos diferentes tópicos do curso e da sua aplicação, aos conteúdos e aos tópicos, será determinado o tipo de interação que o estudante terá com as diversas componentes do curso, alinhados com os resultados esperados. Estes princípios de desenho educacional, encontram-se descritos na Tabela 13.

As **Mecânicas de Jogo** agregam as componentes revistas na literatura: mecânicas, dinâmicas e componentes de jogos. Os componentes de jogo, são os elementos que podem ser incorporados no desenho do curso. As mecânicas e dinâmicas são planeadas e configuradas, tendo por base a interação que se pretenda que o estudante tenha com os diversos conteúdos e atividades, dependentes ou não, do seu percurso ao longo do curso. Medalhas, Pontos, Níveis, Avatares, Moeda Virtual, Quadros de Honra, são elementos que podem ser utilizados como mecânicas de jogo em contexto educativo (Deterding et al., 2011; Dicheva et al., 2015; Giannetto, Chao, & Fontana, 2013; Zichermann & Cunningham, 2011) e particularmente, na aprendizagem da programação. As medalhas são elementos simbólicos que são atribuídos aos estudantes por completar ou obter, uma determinada competência, um conhecimento, podendo ser visualizadas pelos restantes estudantes, demonstrando assim o seu domínio ou conhecimento. Os pontos

podem ser obtidos com a realização de uma variedade de tarefas e atividades tais como: *quizzes*, visualização de conteúdos, submissão de trabalhos, colocar *posts* nos fóruns, entre outras atividades *online*. Os quadros de honra, são mecânicas associadas aos pontos, os quais são agrupados, gerando assim uma classificação dos estudantes, que mais pontos conseguiram no total. Os avatares nos jogos de computador e em ambientes virtuais, representam o utilizador no mundo virtual. A utilização de um avatar e a sua customização, permite um nível emocional entre o jogo e o jogador. Em contexto educativo, a utilização de avatares pode aumentar a presença social, estabelecer conexões fortes entre o grupo e ainda contribuir para uma maior imersão no curso. Os níveis permitem dividir um jogo em pequenas partes, separar peças atingíveis e continuar para o próximo nível. Num contexto educativo e *online*, os níveis permitem uma progressão e sequenciação através de conteúdos e atividades.

Tabela 12 - Mecânicas de Jogo

Mecânicas	Descrição	Autor
Pontos e Quadro de Honra	Os pontos podem ser obtidos com a realização de uma variedade de tarefas e atividades, tais como: <i>quizzes</i> , visualização de conteúdos, submissão de trabalhos, colocar <i>posts</i> nos fóruns, entre outras atividades <i>online</i> . Os Quadros de Honra, são mecânicas associadas aos pontos, agrupando-os de modo a gerar um TOP dos alunos, que mais pontos conseguiram no total.	
Medalhas	As medalhas são símbolos simbólicos que são atribuídos aos alunos por completar ou obter uma competência, um conhecimento, podendo ser visualizadas por outros, mostrando assim o seu domínio ou conhecimento.	(Deterding et al., 2011; Iosup & Epema, 2014; Lee & Hammer, 2011; Simões, Redondo, & Vilas, 2013; Zichermann & Cunningham, 2011)
Níveis	Os níveis permitem dividir um jogo em pequenas partes, separar peças atingíveis e continuar para o próximo nível. Num contexto educativo e <i>online</i> , os níveis permitem uma progressão e sequenciação através de conteúdos e atividades.	
Moeda Virtual (Virtual Goods)	Representam a utilização virtual de moeda. Essa moeda pode ser utilizada, por exemplo, na extensão do prazo de trabalhos, na obtenção da possibilidade de mais do que uma tentativa de resposta em <i>quizzes</i> , entre outros.	(O'Donovan, Gain, & Marais, 2013)
Avatares	Os avatares nos jogos de computador e em ambientes virtuais, representam o utilizador no mundo virtual. A utilização de um avatar e a sua customização, permite um nível emocional entre o jogo e o jogador. Em contexto educativo, a utilização de avatares pode aumentar a presença social e estabelecer conexões fortes entre o grupo, contribuindo para uma maior imersão no curso.	(De Schutter & Abeele, 2014)

Tabela 13 - Princípios de Desenho Educacionais (adaptado de Dicheva et al., (2015))

Princípios de Desenho	Descrição	Autor
Progresso	Permite ao aluno visualizar a progressão no curso.	(Zichermann & Cunningham, 2011)
Retorno	Retorno imediato após submissão das avaliações.	(Kapp, 2012; Lee & Hammer, 2011; Zichermann & Cunningham, 2011)
Envolvimento Social	Inclui a cooperação e interação entre pares.	(Giannetto et al., 2013; Landers & Callan, 2011)
Estado visível	Reputação, credibilidade social e reconhecimento dos resultados obtidos. Pressupõe que os resultados obtidos são visíveis para todos.	(Deterding et al., 2011; Lee & Hammer, 2011)
Acesso bloqueado	Pressupõe que o conteúdo e as atividades, estão dependentes da realização e do desempenho das anteriores.	(Iosup & Epema, 2014)
Liberdade para falhar	Pressupõe baixo risco nas submissões de avaliação. São permitidas múltiplas tentativas.	(Deterding et al., 2011; Iosup & Epema, 2014; Lee & Hammer, 2011)
Restrição no tempo das submissões	Pressupõe que a submissão da avaliação, e.g., um <i>quiz</i> , tem definido um tempo limite para ser submetido	(Kapp, 2012)
Objetivos/Desafios	Definição de objetivos específicos claros, de dificuldade moderada e imediatos	(Kapp, 2012; J. J. Lee & Hammer, 2011)
Customização	Definição de experiências personalizadas, dificuldade adaptada, desafios que são perfeitamente alcançáveis, aumentar a dificuldade, à medida que as competências do aluno expandem.	(Kapp, 2012; Lee & Hammer, 2011; Zichermann & Cunningham, 2011)
Narrativa	Pressupõe a apresentação do conteúdo programático e a definição das atividades enquadradas numa história que se desenrola ao longo do curso.	(Kapp, 2012)
Liberdade de escolha	Inclui a possibilidade de os estudantes escolherem: que tipo de desafios querem completar, e.g., contribuir para um <i>blog</i> , completarem um <i>quiz</i> , criarem um vídeo educacional.	(Deterding et al., 2011; Iosup & Epema, 2014; Lee & Hammer, 2011)
Surpresas, Prémios	Pressupõe a utilização de elementos surpresa e prémios.	(Pettit, McCoy, Kinney, & Schwartz, 2015)

A gamificação na educação é aceite como uma estratégia de absorção e envolvimento do estudante com a aprendizagem (Zichermann & Cunningham, 2011). Os princípios de desenho educacional conjugados com as mecânicas de jogo, contribuem para criar esse ambiente de absorção e envolvimento. Desta forma, a absorção cognitiva será influenciada pela integração desses elementos. De acordo com Agarwal & Karahanna (2000), a **Absorção Cognitiva** é um estado de profundo envolvimento com o *software* e é exibido através das seguintes dimensões: a) dissociação temporal (ou a inabilidade de registar a passagem do tempo, enquanto se está envolvido na interação); b) focado e imerso (experiência de total envolvimento onde outras atenções, são na sua essência, ignoradas); c) prazer intensificado (captura os aspetos agradáveis da interação); d) controlo (captura a perceção do utilizador de estar a controlar a interação); e e) curiosidade (a experiência desperta a curiosidade sensorial e cognitiva do indivíduo). Estas

variáveis são representativas da forma como o estudante fica profundamente envolvido numa tarefa, até ficar num estado de alienação relativamente ao ambiente que o rodeia. De modo a avaliar esse estado, será aplicada uma dimensão, tendo por base estes cinco itens.

A **Personalidade** é definida por Phares (1988) como: “*Personality is that pattern of characteristic thoughts, feelings, and behaviors that distinguish one person from another and that persists over time and situations pp. 4*”. Cada indivíduo diferencia-se pelas suas características individuais e essas características podem assim determinar as suas decisões e ações. Desse modo e num contexto educativo, a personalidade influencia a forma como o indivíduo se posiciona na aprendizagem (De Raad & Schouwenburg, 1996). A personalidade tem sido estudada e diversas teorias têm sido desenvolvidas para caracterizar a personalidade do indivíduo, através do estudo das ações e dos comportamentos. Uma das teorias revistas na literatura, é a teoria dos traços (*big five traits*). Esta teoria estuda a personalidade e mede o grau com que determinados traços influenciam o comportamento do indivíduo em determinadas situações (Gosling, Rentfrow, & Swann Jr., 2003). Os traços de personalidade são agrupados em cinco dimensões: i) extroversão (*extroversion*); ii) agradabilidade (*agreeableness*); iii) conscienciosidade (*conscientiousness*); iv) estado emocional (*emotional stable*); e v) aberto a novas experiências (*openess to new experience*). Cada dimensão corresponde a um conjunto diverso de adjetivos que as caracteriza (Tabela 14).

Tabela 14 - Personalidade (dimensões e adjetivos associados) – adaptado de John & Srivastava (1999)

Dimensões	Adjetivos de traços associados
Extroversão	Sociável, assertivo, energético, entusiasta, assertivo, afetivo.
Agradabilidade	Confiável, altruísta, modesto, empático.
Conscienciosidade	Eficiente, organizado, disciplinado, não impulsivo.
Estado Emocional	Ansioso, irritável, depressivo, envergonhado, impulsivo, vulnerável.
Aberto a Novas Experiências	Imaginativo, curioso, originalidade, variedade de interesses.

Os sistemas informáticos podem lidar facilmente com a personalidade quando é abordada em termos de traços de personalidade (Mairesse, Walker, Mehl, & Moore, 2007), em vez do conceito mais amplo de personalidade. Num sistema educacional gamificado, a personalidade pode ser considerada um moderador do desempenho na tarefa, no estilo de aprendizagem e nas preferências de jogo (Codish & Ravid, 2014). Assim, tendo em consideração que a personalidade do indivíduo é um fator que pode influenciar a adoção do curso *online* gamificado, esta dimensão foi considerada na framework proposta.

O *flow* (em língua portuguesa, fluxo) foi inicialmente estudado por Csikszentmihalyi (1975) e centra-se numa simples questão: “Porque é que as pessoas ficavam tão comprometidas com as atividades sem terem incentivos externos?”. Csikszentmihalyi concluiu da análise efetuada a diversos estudos, que todos partilhavam um aspeto em comum, que Csikszentmihalyi designou por “*flow state*” ou “*flow experience*”. Os autores Engeser & Rheinberg (2008) caracterizaram o *flow*, como o equilíbrio entre a perceção de uma das competências e a perceção da dificuldade. Contribui para esse equilíbrio, a atividade a realizar ser coerente, ter alguma lógica associada, não ter aspetos contraditórios e fornecer um retorno claro e não ambíguo. O indivíduo sente-se assim desafiado e confiante, que tem tudo sob controlo, com um elevado grau de concentração na atividade, derivada de uma atenção profunda. Nesse sentido, o estado de *flow* tem um aspeto funcional forte, na experiência que o indivíduo experiencia altamente concentrado e com um desafio aceitável, enquanto mantém o controlo. Assim, um dos aspetos a considerar nesta investigação com a incorporação desta dimensão, é identificar o estado de *flow* experienciado no curso de programação *online* gamificado.

A aprendizagem da programação é percecionada pelos estudantes, como tendo um elevado grau de dificuldade. E muitos dos estudantes tendem a desistir na primeira dificuldade encontrada (Robins et al., 2003; Yadav et al., 2017). Tendo presente estas questões relacionadas com a dificuldade na aprendizagem da programação, Piteira & Costa (2013, 2012, 2014) conduziram diversos estudos que tiveram como objetivo identificar as dificuldades percecionadas pelos estudantes em diferentes conceitos fundamentais de programação. Com base nos resultados desses estudos e na identificação das dificuldades nos diversos conceitos de programação, integrou-se a dimensão da dificuldade percecionada, na presente *framework*. Pretende-se desta forma, que em função da dificuldade percecionada para um determinado tópico de programação, sejam consideradas estratégias educacionais gamificadas que aumentem o envolvimento, a motivação e o estímulo do estudante na aprendizagem dos conceitos de programação que apresentam um maior grau de dificuldade e que atendendo a essa dificuldade, o estudante tende a desistir. Após a identificação das diversas componentes que compõem a *framework*, procedeu-se ao desenho concetual da mesma (Figura 12). De seguida iniciou-se o processo de implementação da *framework* e respetiva validação, descrita no subcapítulo seguinte.

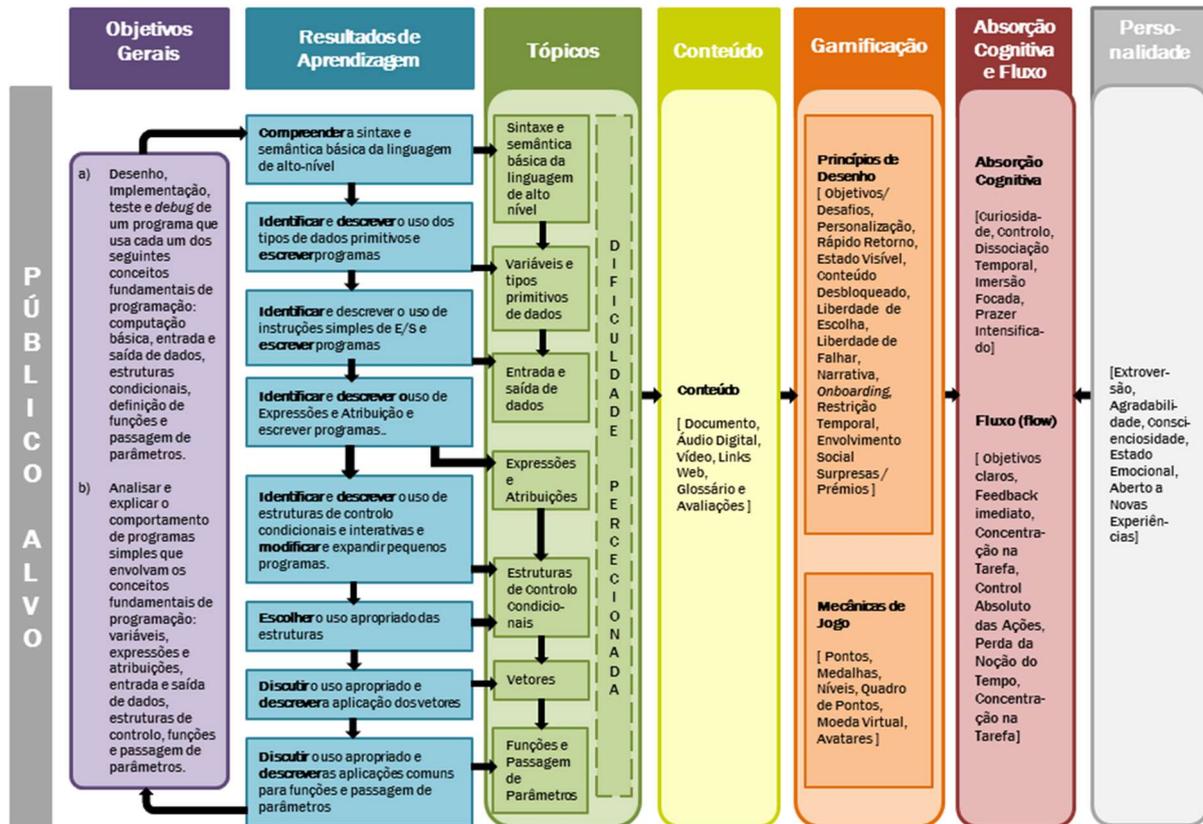


Figura 12 – *Framework* Gamificada para Cursos *Online* de Programação (Piteira & Costa, 2017)

5.6 Fase 4: Demonstração - Implementação da *framework* gamificada para cursos *online* de programação

5.6.1 Infraestrutura tecnológica de suporte à implementação do curso *online*

Para implementação e disponibilização do curso *online* de fundamentos de programação, utilizou-se um *Learning Management Systems* (LMS), nomeadamente, o LMS Moodle (<http://moodle.org>). O Moodle é atualmente, um dos LMS de código aberto mais utilizado nas instituições de ensino superior. Existe uma larga comunidade de utilizadores que contribuem para o crescimento da plataforma desde 2002, altura em que esta deu os primeiros passos e mantém-na atualizada com as inovações tecnológicas e pedagógicas que vão surgindo, disponibilizando assim, atualizações permanentes e periódicas da plataforma de *e-Learning*. Esta plataforma é gratuita e de fácil utilização e interação, permitindo aos educadores utilizá-la sem necessidade de consumir muito tempo a aprender como funciona, criando assim os seus cursos *online*. A plataforma também se encontra disponível para os estudantes, sendo de fácil interação, não acarretando grandes dificuldades na sua utilização (Piteira & Costa, 2006). O

desenho e o desenvolvimento do Moodle foram guiados por uma “pedagogia construtivista social”. Esta linha de investigação é baseada em teorias de construtivismo (Piaget, 1976) e do construcionismo (Papert & Harel, 1991). Do ponto de vista construtivista, as pessoas constroem ativamente novo conhecimento, enquanto interagem com os ambientes de aprendizagem. Tudo o que se lê, vê, ouve, sente e se toca, é testado com o conhecimento anterior e se for viável com o nosso mundo mental, pode formar novo conhecimento. O Moodle favorece a interação entre diferentes utilizadores através da utilização de fóruns, facilitando assim a criação de novo conhecimento. O construcionismo defende que a aprendizagem é particularmente efetiva, quando se constrói algo para os outros experienciarem. O Moodle permite criar conhecimento e torna-o disponível e acessível para os restantes, fazendo-nos sentir que o nosso trabalho é útil para os outros. O construtivismo social, estende o construtivismo em definições sociais, na medida em que os grupos constroem conhecimento para outros grupos, de forma colaborativa, criando uma cultura de partilha de artefactos com partilha de significado (https://docs.moodle.org/archive/pt/filosofia_do_moodle). Esta plataforma integra um conjunto de ferramentas que suportam as mais diversas atividades pedagógicas e recursos *online* de suporte à aprendizagem. Assim sendo, o *Moodle* é uma plataforma que integra, conecta e estabelece relações, com um conjunto de ferramentas que são úteis nas nossas tarefas enquanto professores, criando assim, um ecossistema tecnológico.

De seguida apresentam-se algumas das ferramentas e recursos presentes na versão atual do Moodle. Na terminologia utilizada pelo Moodle e na versão em português (Figura 13), as ferramentas são atividades. No entanto, no presente trabalho adota-se uma terminologia diferente, nomeadamente ferramentas. Ferramentas é a tecnologia que suporta as atividades pedagógicas, sendo que nesse sentido, a ferramenta fórum pode ser utilizada na atividade colaborativa de discussão de um determinado conceito ou temática ou na atividade de acompanhamento de um determinado trabalho.

5.6.1.1 Ferramenta: Teste

A ferramenta teste (*quiz*), é uma das ferramentas mais completas e robustas que integram a plataforma Moodle e que permite aos professores construir testes *online*, com um conjunto diversificado de questões: escolha múltipla, verdadeiro-falso, resposta curta, correspondência de respostas curtas aleatórias, entre outras (Figura 14). O Moodle tem por base um conjunto de tipos de questões, contudo, existem outros tipos que podem ser posteriormente adicionados e obtidos a partir da página oficial do Moodle (<http://moodle.org>).

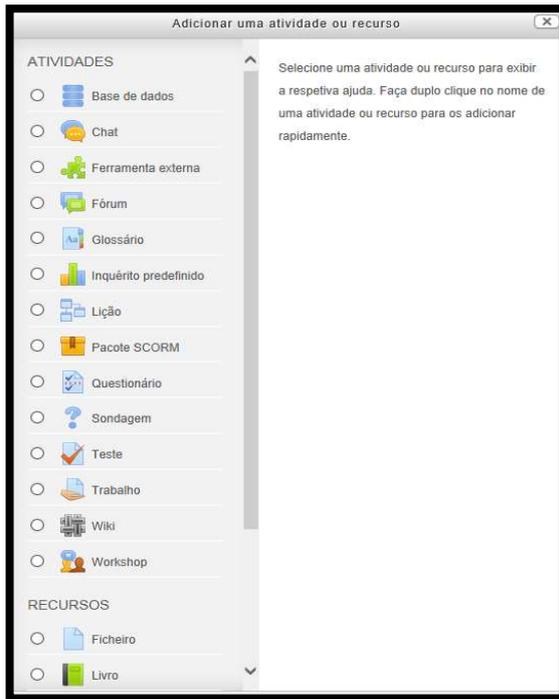


Figura 13 – Moodle: Ferramentas de suporte às atividades

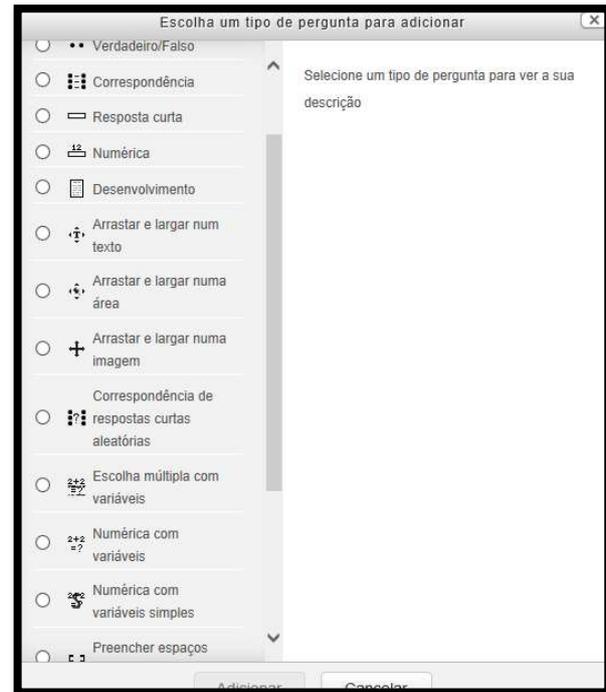


Figura 14 – Moodle: Tipos de questões (testes online)

5.6.1.2 Ferramenta: Trabalho

Esta atividade permite ao professor recolher trabalhos dos estudantes (ficheiros), revê-los e fornecer retorno, em termos de notas e de comentários.

5.6.1.3 Ferramenta: Fórum e Chat

As atividades *fórum* e *chat* são atividades que permitem aos estudantes e aos professores, uma interação entre estudantes-estudantes e entre estudantes-professores. A atividade *chat* permite uma interação síncrona, enquanto o fórum permite uma interação assíncrona. A atividade fórum permite diversos modos de parametrização e de acordo com o objetivo da atividade pedagógica suportada pela ferramenta fórum, pode-se escolher um de entre diversos tipos de fóruns, e.g., fórum de pergunta-resposta, fórum social, fórum de uso geral.

5.6.1.4 Ferramenta: Sondagem, Inquérito

As atividades sondagem e inquérito permitem, um retorno dos estudantes para os professores. Na atividade inquérito, existem cinco opções pré-definidas para seleção, não sendo possível ao professor criar os seus próprios inquéritos, apenas utilizar os que estão criados. Na atividade sondagem, é possível ao professor definir uma questão, sobre a qual pretende obter um retorno do aluno.

5.6.1.5 Ferramenta: Lição

A lição do Moodle é uma ferramenta, que permite criar conteúdo (texto, imagem e vídeo) e configurar caminhos alternativos da lição, dependente de pontos de controlo, definidos através de perguntas que são colocadas aos estudantes.

5.6.1.6 Ferramenta: Pauta

A ferramenta pauta do Moodle permite, guardar e mostrar as notas obtidas pelo estudante. Através desta ferramenta, é possível ao estudante, visualizar as notas obtidas nas diversas atividades que realiza na plataforma e visualizar também a nota final de curso.

5.6.1.7 Recurso: Etiqueta

Etiqueta é um recurso que permite ao professor introduzir informações, objetivos, entre outros, nos diversos módulos do curso.

5.6.2 Elementos de desenho educacional

Para além das ferramentas e dos recursos disponíveis na plataforma Moodle, existem elementos que permitem configurar as diversas interações com os conteúdos e com as atividades. Essas interações dependem dos objetivos pedagógicos definidos para o curso. Sumariamente, apresentam-se de seguida, os elementos de desenho educacional revistos na literatura e presentes na plataforma de *e-Learning* Moodle.

5.6.2.1 Acesso Restrito

Esta funcionalidade, recentemente integrada nas novas versões do Moodle, permite condicionar o acesso às atividades por parte dos estudantes, em função de um conjunto de requisitos:

- **Atividade completa:** requer que os estudantes tenham completado uma outra atividade. O completar a atividade, pode assumir diferentes aspetos, como visualizar um recurso (PDF, vídeo, entre outros) ou até mesmo ter efetuado uma avaliação num teste *online* e ter obtido uma determinada nota.
- **Data:** não permite ao estudante o acesso, antes ou depois de uma determinada data. Pode ser efetuado o acesso restrito entre duas datas, uma de início e outra de fim.
- **Perfil do estudante:** podem ser definidas condições tendo por base a língua ou o país definido no perfil.

5.6.2.2 Medalhas

As medalhas é uma das funcionalidades, que o Moodle passou a disponibilizar recentemente. Através desta funcionalidade, é possível parametrizar elementos específicos, que em resultado de um objetivo cumprido ou progresso efetuado, o estudante possa ser premiado através da atribuição de uma medalha.

5.6.2.3 Progresso

Esta funcionalidade permite que automaticamente, as atividades sejam marcadas como concluídas. Por exemplo, se a atividade planeada para o estudante realizar, for a resposta a uma pergunta no fórum, após a sua resposta, essa atividade é automaticamente marcada como concluída, ficando essa informação visível para o estudante.

5.6.2.4 Retorno

Nas atividades de avaliação, é possível parametrizar um retorno imediato ao estudante, da avaliação que realizou. Este retorno pode ser aplicado às atividades que tenham sido disponibilizadas através da ferramenta de testes, atividades *workshops* ou atividades fóruns.

5.6.3 Operacionalização da *framework*

A partir da *framework* teórica-concetual descrita no subcapítulo 5.5, efetuou-se a implementação da mesma (Figura 15). De forma a operacionalizar a *framework*, foi desenhado e implementado um curso *online* de fundamentos de programação. O curso foi denominado de *SmartMOOC* e foi operacionalizado com recurso à plataforma de *e-Learning* Moodle (Figura 16).

5.6.3.1 Desenho do curso *online*

O desenho do curso teve em conta as dimensões identificadas na *framework* previamente proposta, mais concretamente, os princípios de desenho educacional, os tópicos, os conteúdos, os materiais de suporte, as mecânicas e as dinâmicas de jogo. Foram estimadas no total, 20 horas de trabalho para a realização do curso, distribuídas pelos vários módulos do curso. Seguidamente, descreve-se a implementação efetuada.

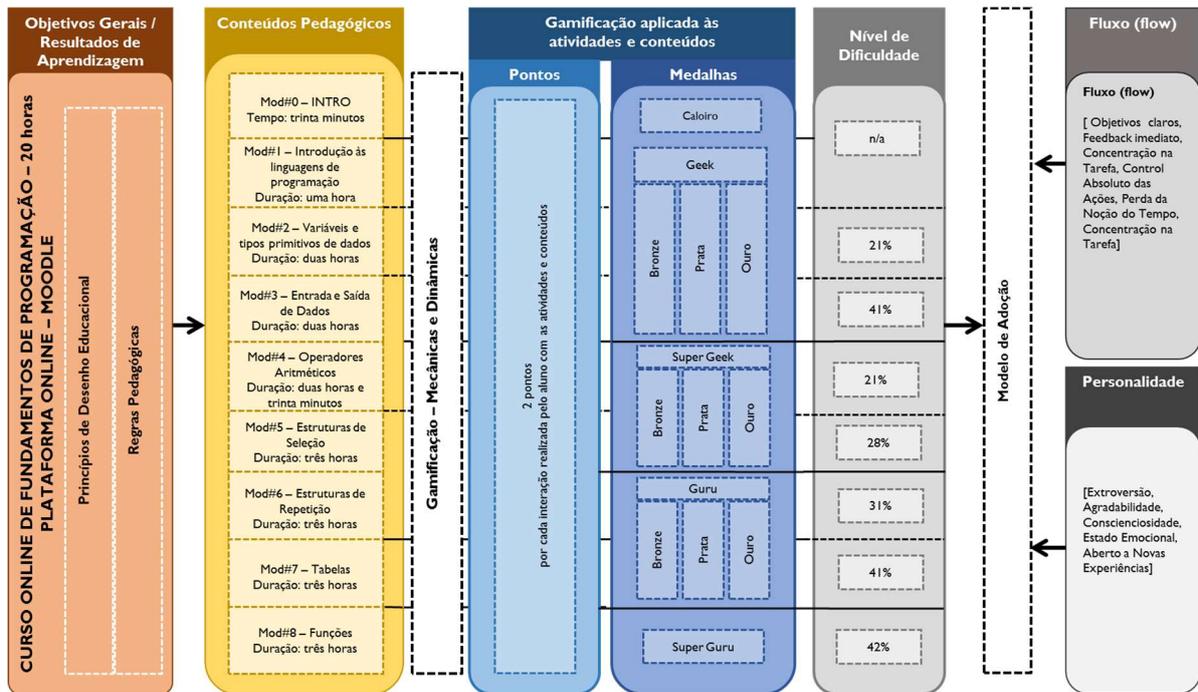


Figura 15 - Instanciação da *Framework* Teórica-Concetual (Piteira, Costa, & Aparício, 2017)



Figura 16 - Fundamentos de Programação - SmartMOOC

5.6.3.2 Roteiro do curso *online*

O roteiro do curso foi concebido de forma modular. Os módulos do curso e as atividades associadas a cada módulo foram desenhadas e implementadas na plataforma de aprendizagem *online* Moodle. O curso encontra-se acessível através do endereço web <http://iscte.acm.org/elearning>. Para aceder ao curso, os estudantes necessitam de efetuar o registo. Após validação do registo, têm acesso total ao curso.

5.6.3.3 Módulos do curso

Como referido anteriormente, o curso está estruturado em módulos, para os quais foram definidos os objetivos de aprendizagem e as atividades pedagógicas que os estudantes devem realizar de modo a poderem prosseguir no curso. De seguida, efetua-se a descrição desses módulos.

Módulo#0 - Ambientação à plataforma e apresentação geral do funcionamento do curso

Objetivos: Neste módulo, pretende-se que o estudante, através da realização de diversas atividades, tais como, a inserção da sua foto de perfil e a apresentação pessoal no fórum, se ambiente à plataforma e ao curso (Figura 17).



Figura 17 – Módulo 0 – Ambientação à Plataforma - SmartMOOC

Atividades: Aceder ao perfil: inserir uma foto, completar os campos do nome e do apelido. Aceder ao “*Café Social – Conversas à Mesa*” e fazer uma breve apresentação aos restantes colegas do curso.

Módulo#1 – Introdução às linguagens de programação

Objetivos: Pretende-se neste módulo que o estudante adquira um conhecimento geral das linguagens de programação e que compreenda a importância da aprendizagem da programação, no contexto atual em que vivemos (Figura 18).

Módulo#1 - Introdução às Linguagens de Programação

Objetivo:
Neste módulo de Introdução às linguagens de programação desafiamos-te a teres uma noção geral da história das linguagens de programação. Será que vamos conseguir??

Diagrama central: Um personagem desenhado à mão, segurando um tablet e um telefone, rodeado por várias linguagens de programação escritas à mão: C++, JAVA, C#, FLASH, HTML5, HTML, .NET, CSS, VB, PHP, e SQL.

Para alcançares este objetivo lê o conteúdo “Introdução às Linguagens de Programação” e visiona o vídeo “A importância de aprender a programar”.

Após realizares estas duas atividades colocamos-te uma questão no fórum à qual deves responder. Podes após a tua resposta, visualizar a resposta dos teus colegas. Desafiamos-te a comentar a resposta deles. Concordas? Sim? Então vamos lá...comenta a resposta dos teus colegas!

Após a visualização dos conteúdos e resposta no fórum, estarás em condições de aceders ao Módulo#2. As atividades desse módulo ficam disponíveis para iniciares a aprendizagem nesse módulo.

Bom trabalho!!!

Introdução às Linguagens de Programação

Figura 18 - Módulo 1 – Introdução às Linguagens de Programação - SmartMOOC

Atividades: Visualização de um conteúdo em formato de vídeo; Leitura de um conteúdo em formato PDF; Aceder ao fórum e responder a uma questão relacionada com as linguagens de programação. Nesta atividade, o estudante visualiza a questão e após submeter a sua resposta, terá então acesso a todas as respostas colocadas.

Módulo#2 – Variáveis

Objetivos: Neste módulo, o estudante deve adquirir conhecimentos gerais referentes a variáveis, nomeadamente, a sua definição, atribuição de valor, seleção do tipo de dado, consoante o tipo de valor a armazenar.

Atividades: Leitura do documento em formato PDF “*Variáveis e Tipos de Dados*”. Exercício prático de avaliação *online*, através do qual o estudante deverá responder a um conjunto diverso de questões relacionadas com variáveis e tipos de dados.

Na Figura 19, observam-se o desenho e o *layout* do módulo 2, na plataforma online Moodle.



Módulo#2 - Variáveis

Objetivos:
No módulo Variáveis o objetivo que te propomos é que consigas adquirir conhecimentos sobre as variáveis em programação. O que são, como as definir, como atribuir valor e como definir o tipo de dados que podem guardar. O conteúdo programático deste módulo encontra-se abaixo descrito.
Após visualizares o conteúdo que te explica e exemplifica com exemplos podes iniciar o teu teste de avaliação de conhecimentos.



PRIMEIROS PASSOS!!! PREPARADOS?!?

Contéudos Programático:

- Tipos de Dados;
- Variáveis com valor e sem valor;
- Atribuição de Valor

Tempo:
O tempo estimado de trabalho para este módulo é de duas horas.

Materiais:

 [Tipos de Dados e Variáveis](#) 

Quiz:

Figura 19 - Módulo 2 – Variáveis - SmartMOOC

Módulo#3 – Entrada e Saída de Dados

Objetivos: Pretende-se neste módulo que o estudante compreenda e crie corretamente sequências de instruções. Compreenda e defina corretamente a entrada, armazenamento e saída de dados para o monitor.

Atividades: Leitura do documento em formato PDF “*Entrada e Saída de Dados*”. Exercício prático de avaliação *online*, através do qual o estudante deverá responder a um conjunto diverso de questões relacionadas com entrada e saída de dados.

Conteúdos: Sequência de Instruções; Receber dados com armazenamento em variáveis;
Saída do valor armazenado em variáveis.

A Figura 20, demonstra a implementação do módulo 3 na plataforma *online*.

Módulo#3 - Entrada e Saída de Dados

Objetivos:
Agora que já sabes o que são as variáveis, neste módulo Entrada e Saída de Dados vamos explicar e exemplificar como se pode receber valores a partir do teclado, guardar esses valores em variáveis e fazer depois a saída desses valores para o écran (output dos dados e visualização dos mesmos no écran).
Os conteúdos abordados neste módulo estão identificados abaixo. **DO IT!!!!**



"Designed by www.slom.pics / Freepik"

Conteúdo Módulo Entrada e Saída de Dados:

- Sequência de Instruções;
- Receber Dados com Armazenamento em Variáveis;
- Output do valor das variáveis;

Tempo:
O tempo estimado de trabalho para este módulo é de duas horas.

Materials:

 Entrada e Saída de Dados

Conteúdo indisponível, verifique as condições de acesso:

- Deve aceder e/ou concluir a atividade **Avaliação Variáveis e Constantes (Ga)**.
- Pertencer a um grupo em **Gamification**

Quiz:

Figura 20 – Módulo 3 – Entrada e Saída de Dados - SmartMOOC

Módulo#4 – Operadores

Objetivos: Pretende-se neste módulo que o estudante compreenda e utilize corretamente os diferentes operadores aritméticos (Figura 21).

Atividades: Leitura do documento em formato PDF “Operadores e Expressões”; Exercício prático de avaliação *online*, através do qual o estudante deverá responder a um conjunto diverso de questões relacionadas com operadores aritméticos.

Conteúdos: Soma, subtração, divisão e multiplicação (particularidade entre as operações com diferentes tipos de dados); Outros operadores (incremento e decremento).

Módulo#4 - Operadores Aritméticos

Objetivos:
Agora que já sabes o que são variáveis, como guardar valores e fazer o output dos valores, queres saber como tendo duas variáveis com valores fazeres operações aritméticas entre elas. Quando isso é possível!!! Sim porque existem restrições, depende do tipo de dados das variáveis.
Assim, o objetivo deste módulo é aprenderes e treinares a utilizar os operadores aritméticos e outros tipos de operadores.



"Designed by www.slion.pics / Freepik"

Conteúdos:

- Soma, subtração, divisão e multiplicação (particularidades entre as operações com diferentes tipos de dados);
- Outros operadores aritméticos.

Tempo:
O tempo estimado de trabalho para este módulo é de duas horas e trinta minutos.

Materiais:

 Operadores e Expressões

Não disponível, a não ser que qualquer uma:

- Pontuação mínima necessária em **Avaliação Variáveis e Constantes**
- Pontuação mínima necessária em **Avaliação Variáveis e Constantes (Ga)**

Quiz:

Figura 21 - Módulo 4 – Operadores - SmartMOOC

Módulo#5 – Seleção

Objetivos: Neste módulo, o estudante deverá compreender o que são e para que servem os mecanismos de seleção. Deverá também identificar os diversos tipos de seleção e saber aplicá-los corretamente (Figura 22).

Atividades: Leitura do documento em formato PDF “*Instruções de Controlo e Seleção*”; Exercício prático de avaliação *online*, através do qual o estudante deverá responder a um conjunto diverso de questões relacionadas com mecanismos de seleção.

Conteúdos: Operadores lógicos, if e condições; if-else; if-elseif; else; switch.

Módulo#5 - Seleção

Objetivos:
Neste módulo vamos abordar os mecanismos de seleção. Os mecanismos de seleção em programação são importantes porque através deles podemos desenvolver algoritmos que em função de uma determinada condição determine fazer uma ação A ou ação B.



"Designed by www.ston.pics / Freepik"

Assim, pretendemos que neste módulo compreendas e entendas os mecanismos de seleção e a sua importância na programação. Deves saber que mecanismos de seleção existem e quando devem ser utilizados.

Conteúdos:

- Operadores Lógicos; IF e condições; IF - Else; IF-Else IF - Else; Switch.

Tempo:
O tempo estimado de trabalho para este módulo é de três horas.

Figura 22 - Módulo 5 – Estruturas de Seleção -SmartMOOC

Módulo#6 – Repetição

Objetivos: Neste módulo, o estudante deverá compreender o que são e para que servem os mecanismos de repetição. O aluno deverá conhecer os diversos mecanismos de repetição e saber aplicá-los corretamente (Figura 23).

Atividades: Leitura do documento em formato PDF “*Instruções de Repetição*”; Exercício prático de avaliação *online*, através do qual o estudante deverá responder a um conjunto diverso de questões relacionadas com mecanismos de repetição.

Conteúdos: While; Do While; For; While vs For vs Do While.

Módulo#6 - Repetição

Objetivos:
No módulo#5 abordámos os mecanismos de seleção.
Neste módulo vamos abordar os mecanismos de repetição. Também estes mecanismos à semelhança dos mecanismos de seleção têm uma importância enorme em programação.
É através da sua utilização que conseguimos desenvolver algoritmos que através de uma determinada condição executa blocos de código repetidamente (em ciclo) até que a condição de paragem seja verificada.

UPS!!! Parece que nossa tartaruga ainda não atingiu a condição de paragem!!! 😊



"Designed by www.slon.pics / Freepik"

Assim, pretendemos que neste módulo compreendas e entendas os mecanismos de repetição e a sua importância na programação. Deves saber que mecanismos de repetição existem e em que circunstâncias deves utilizar um ou outro mecanismo.

Conteúdos:

- While
- Do While
- For
- While VS Do While VS For

Tempo:
O tempo estimado de trabalho neste módulo é de três horas.

Conteúdos:

 INSTRUÇÕES REPETIÇÃO

Não disponível, a não ser que qualquer uma:

- Deve aceder e/ou concluir a atividade **Avaliação Estruturas de Seleção (Ga)**.
- Deve aceder e/ou concluir a atividade **Avaliação Estruturas de Seleção**.

Quiz:

Figura 23 - Módulo 6 – Estruturas de Repetição - SmartMOOC

Módulo#7 - Vetores

Objetivos: Pretende-se neste módulo que o estudante compreenda o que são os vetores. Deverá saber aplicá-los de forma a receber, guardar e fazer a saída de valores (Figura 24).

Atividades: Leitura do documento em formato PDF “*Vetores*”. Exercício prático de avaliação *online*, através do qual o estudante deverá responder a um conjunto diverso de questões relacionadas com vetores.

Conteúdos: Declaração com/ sem inicialização; ler valor; armazenar valores; percorrer o vetor; saída dos valores armazenados no vetor.

Módulo#7 - Vetores

Objetivos:
No presente módulo serão abordados os vetores.
Vetores também designados por arrays ou tabelas são mecanismos que nos permitem atribuir e guardar vários valores numa única variável.



Pretende-se neste módulo que compreenda e saibam utilizar estes mecanismos. Assim, no final do módulo devem saber declarar um vetor, atribuir valores, ler valores para o vetor, percorrer o vetor e mostrar valores armazenados no vetor.

Conteúdos:

- Declaração sem/com inicialização;
- Ler valor;
- Armazenar um valor;
- Percorrer o vetor;

Tempo:
O tempo estimado para a realização deste módulo é de três horas.

Conteúdos:

 Vetores

Não disponível, a não ser que qualquer uma:

- Deve aceder e/ou concluir a atividade **Avaliação Estruturas de Repetição (Ga)**.
- Deve aceder e/ou concluir a atividade **Avaliação Estruturas de Repetição**.

Quiz:

Figura 24 - Módulo 7 – Vetores - SmartMOOC

Módulo#8 – Subprogramas

Objetivos: Neste módulo, o estudante deverá compreender o que são subprogramas e qual a sua importância na criação de um programa. Deverá saber como definir um subprograma, invocar e retornar. Compreender e utilizar corretamente os argumentos (Figura 25).

Atividades: Leitura do documento em formato PDF “*Subprogramas*”. Exercício prático de avaliação *online*, através do qual o estudante deverá responder a um conjunto diverso de questões relacionadas com subprogramas.

Conteúdos: Definição, invocação e retorno. Argumentos.

Módulo#8 - SubProgramas

Finalmente o último módulo!!



Designed by www.slou.pics / Freepik

Objetivos:
Este módulo introduz os subprogramas. Os subprogramas são estruturas muito importantes em programação. Esperamos que no final compreendas o que são, como utilizar e quando utilizar.

Conteúdos:

- Definição e invocação e retorno
- Argumentos
- Override

Tempo:
O tempo previsto necessário para a realização do módulo é de três horas.

Conteúdos:

 Sub-Programas

Não disponível, a não ser que qualquer uma:

- Deve aceder e/ou concluir a atividade **Avaliação Vectores (Ga)**.
- Deve aceder e/ou concluir a atividade **Avaliação Vectores**.

Quiz:

Figura 25 - Módulo 8 – Subprogramas - SmartMOOC

5.6.3.4 Modelo do curso

Os cursos *online* abertos e massivos (MOOCs), designados na língua inglesa por *massive open online courses*, surgiram como um novo paradigma na educação, tendo recebido uma enorme atenção nos últimos anos. Diversas plataformas *online* surgiram com uma oferta diversificada de cursos nas mais diversas áreas do conhecimento (Saadatdoost, Sim, Jafarkarimi, & Hee, 2015). A distribuição de cursos através da Internet, teve início em 1974, tendo-se intensificado a partir de 2008, altura em que diversas universidades americanas começaram a distribuir, os seus cursos através da Internet. Mas o grande avanço e a mudança de paradigma, ocorreram a partir de 2012, com o aparecimento dos MOOCs. Nessa altura, os MOOCs passaram a ser um tema em discussão nas universidades e muitas delas aderiram a este novo paradigma, disponibilizando os seus próprios MOOCs. Thompsom (2011) define o conceito de MOOC como: um modelo educacional de distribuição, destinado a vários graus, massivo, teoricamente sem limite de inscrições; aberto, permitindo a participação de qualquer pessoa, habitualmente sem custos; *online*, com atividades de aprendizagem realizadas através da Internet; e um curso, estruturado à volta de um conjunto de objetivos educacionais, relacionados com uma área específica do conhecimento. Baker, Bujak e DeMillo (2012), acrescentaram ainda à definição

de MOOC: que é um modelo virtual de distribuição que permite a participação em atividades de aprendizagem em convenientes espaços e tempos, em vez de forçar os estudantes a determinados espaços e tempo; *blended learning*, que combina interações *online* e *face-to-face*, podendo facilitar uma colaboração generalizada e global com outros estudantes e com equipas especializadas de educadores; e gamificação que aplica estruturas motivacionais baseadas em jogos para incentivar a aprendizagem. Existem dois modelos comuns de MOOCs: xMOOC e cMOOC (Aparicio & Bacao, 2013; Aparicio, Bacao, & Oliveira, 2014). O modelo xMOOC segue uma abordagem centrada na aprendizagem mais tradicional, através da distribuição de conteúdos baseados em vídeos e em formatos tradicionais como o PDF e com disponibilização de *quizzes*, como forma de validar as aprendizagens (Bernhard, Bittel, Van Der Vlies, Bettoni, & Roth, 2013). O modelo cMOOC não está alinhado com o conteúdo, nem com o educador, mas com os outros estudantes participantes e com o seu conhecimento. Os cMOOCs baseiam-se na teoria de aprendizagem conectivista (Saadatdoost et al., 2015). A teoria do conectivismo foi introduzida por Siemens (2005) que a definiu como “*Connectivism is the integration of principles explored by chaos, network, and complexity and self-organization theories. Learning is a process that occurs within nebulous environments of shifting core elements – not entirely under the control of the individual. Learning (defined as actionable knowledge) can reside outside of ourselves (within an organization or a database), is focused on connecting specialized information sets, and the connections that enable us to learn more are more important than our current state of knowing pp.3*”.

Nesse sentido e tendo em consideração a literatura relacionada, o modelo seguido no curso *online* de fundamentos de programação, foi o modelo xMOOC. O desenho do curso foi orientado e baseado numa abordagem mais tradicional, com a disponibilização *online* de conteúdos em formato PDF e de vídeos, tendo também sido criados diversos *quizzes* para validar a aprendizagem dos estudantes. Os estudantes inscreveram-se livremente no curso. Não existiam requisitos prévios para a sua frequência e também não existia limite ao número de inscritos.

5.6.4 Método de avaliação das aprendizagens

As aprendizagens são avaliadas com recurso aos testes *online* (*quizzes*) presentes em todos os módulos. Os estudantes obtêm aprovação no módulo, com uma nota igual ou superior a dez valores.

5.6.5 Certificação das aprendizagens

A aprendizagem é certificada quando o estudante completa todas as atividades do curso com uma média igual ou superior a dez valores. O estudante recebe no final do curso *online*, um certificado de conclusão (Figura 26) emitido pelo *ACM Student Chapter da ACM – ISCTE* (<http://iscte.pt>).

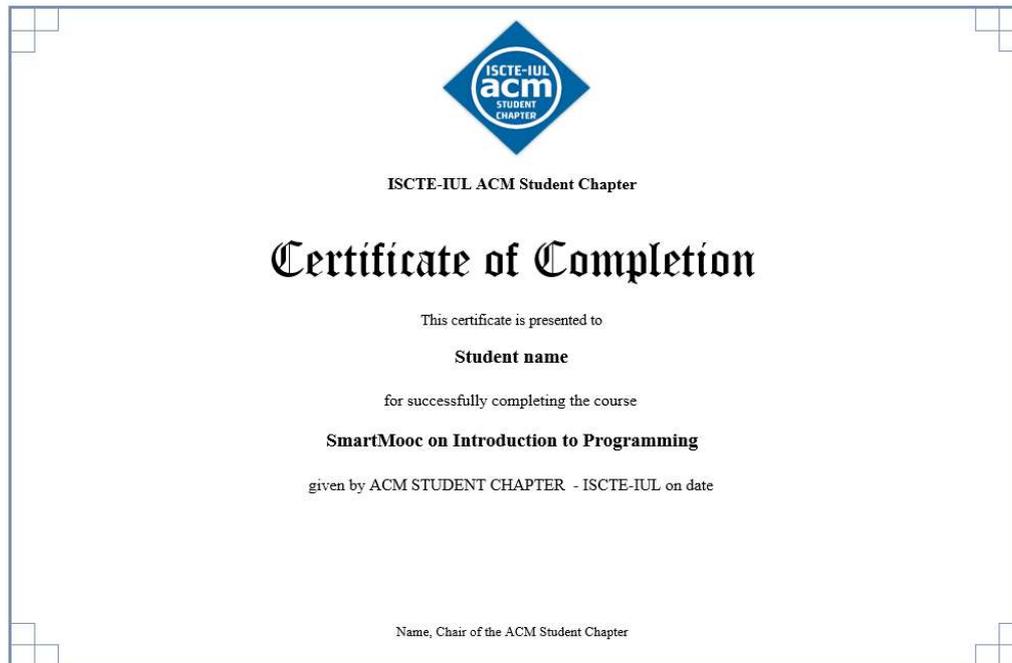


Figura 26 - Certificado de Conclusão de Curso Online - SmartMOOC

5.6.6 Princípios de desenho educacional e mecânicas de jogo aplicados ao curso *online*

Suportados nos princípios de desenho educacional revistos na revisão da literatura e tendo em consideração os objetivos definidos para o curso *online*, foram aplicados os seguintes princípios de desenho educacional: progresso, retorno, envolvimento social, estado visível, acesso bloqueado, liberdade para falhar, restrição no tempo das submissões, objetivos/ desafios. O desenho da gamificação foi igualmente orientado pela framework teórica-concetual proposta. Foram incorporados as seguintes mecânicas de jogo: pontos, níveis, medalhas, quadro de honra e barra de progresso. Foram ainda estabelecidas as respetivas dinâmicas para cada uma das mecânicas. Seguidamente, efetua-se a descrição dos princípios e o modo como foram aplicados ao curso.

5.6.6.1 Princípio educacional: Progresso

O princípio de desenho educacional progresso permite, ao estudante visualizar a progressão no curso. Para a aplicação deste princípio no curso, foi utilizada uma ferramenta gráfica que foi adicionada à plataforma (plug-in). Esta ferramenta (Figura 27) identifica todas as atividades numa barra e a cada atividade realizada e por realizar, corresponde uma cor, que pode ser parametrizada. No caso específico, foi definida a cor azul para as atividades por realizar e a verde, para as atividades realizadas. Dessa forma, o estudante tem uma visualização gráfica das atividades pedagógicas realizadas e por realizar. Esta barra encontra-se visível na primeira página do curso.



Figura 27 – Barra de Progresso

A aplicação deste princípio resulta numa visualização rápida por parte do estudante do seu próprio progresso, permitindo ainda ao professor, ter uma visão abrangente da evolução dos estudantes no curso (Figura 28).



Figura 28 - Progressão dos Estudantes nas Atividades do Curso

5.6.6.2 Princípio educacional: Retorno

O princípio de desenho educacional retorno (Figura 29), foi aplicado às avaliações nos testes *online*. No final da avaliação e após submissão do teste *online*, o estudante obtinha um retorno

imediatamente da avaliação alcançada. O estudante tinha também um retorno textual (mensagens), como forma de o estimular e motivar a continuar o curso ou a repetir a avaliação, caso não tenha sido bem-sucedido na primeira tentativa. A aplicação deste princípio na plataforma de *e-Learning Moodle*, pode assumir diferentes formas consoante a parametrização que for efetuada, por exemplo, se definirmos como abordagem pedagógica, não dar um retorno ao estudante da sua nota, podemos fazê-lo, optando por enviar apenas as mensagens aos estudantes.



Figura 29 - Submissão das Avaliações – Retorno ao Estudante

5.6.6.3 Princípio educacional: Envolvimento Social

O envolvimento social foi aplicado através da ferramenta fórum. Para esse efeito, foram definidos três fóruns no curso: fórum de apoio ao curso, fórum social, ao qual foi atribuído o nome de “*Café Social – Conversas à Mesa*”, com o objetivo de interação e partilha de conhecimentos e ainda um fórum de pergunta e de resposta (Figura 30).

No fórum de apoio ao curso, os estudantes eram convidados a colocar questões de âmbito geral sobre o curso. No fórum “*Café Social – Conversas à Mesa*”, os estudantes realizavam uma das suas primeiras atividades: apresentação aos restantes elementos do curso e poderiam, sempre que o entendessem, voltar ao fórum e colocar novas questões neste fórum, com uma temática que fosse do seu interesse discutir.

No fórum pergunta-resposta, os estudantes realizavam uma atividade, através da qual deveriam responder a uma pergunta teórica sobre paradigmas de programação. Os estudantes apenas poderiam interagir entre si nesse fórum e ver a resposta dos colegas, após responderem à pergunta.

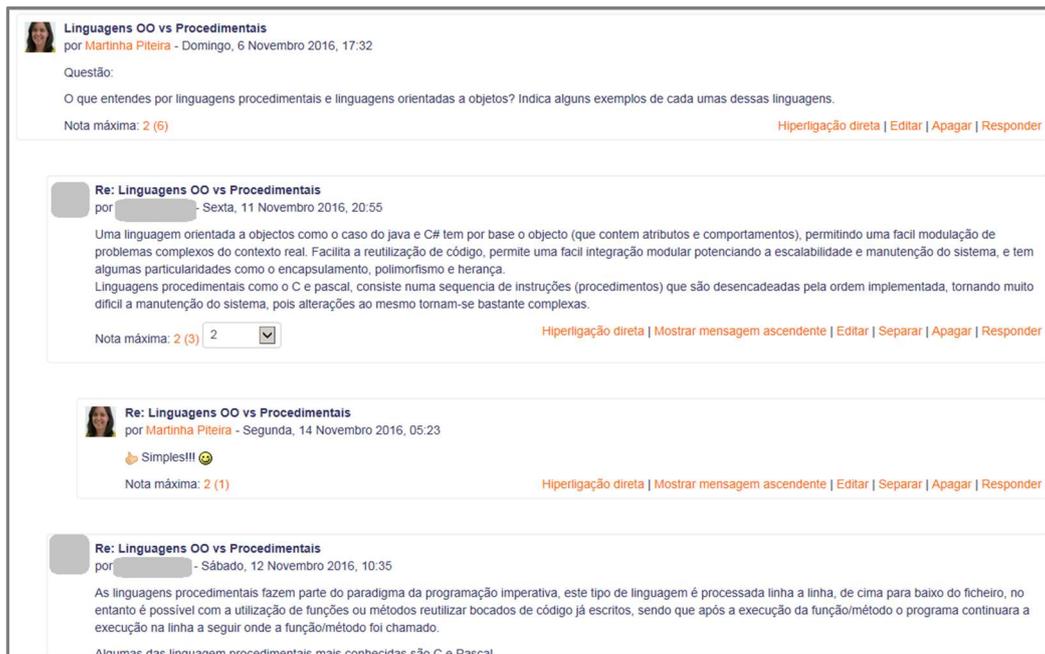


Figura 30 - Fórum Pergunta-Resposta

5.6.6.4 Princípio educacional: Estado Visível

De modo a aplicarmos o princípio estado visível, foi adicionada e configurada na plataforma uma ferramenta, que permitia aos estudantes visualizar o número de pontos que acumulavam pela realização das diversas atividades realizadas, assim como os pontos acumulados pelos colegas. Essa ferramenta efetuava a gestão dos pontos obtidos pelos estudantes e com base nos pontos totais, era contruído o Top 5.

5.6.6.5 Princípio educacional: Acesso Bloqueado

O acesso bloqueado é um princípio de desenho educacional que prevê que os conteúdos e as atividades estejam dependentes umas das outras, não permitindo o acesso sem se verificar determinadas condições nos conteúdos e nas atividades anteriores (Figura 31). No curso *online* de fundamentos de programação, foram definidos diversos módulos, apresentados e descritos anteriormente. Cada módulo continha conteúdos e atividades. O acesso aos módulos seguintes estava condicionado a determinadas condições, como por exemplo, o acesso ao conteúdo “*Tipos de Dados e Variáveis*”, estava dependente da atividade “*fórum responder a pergunta*” (Figura 30). Quando o estudante respondia à questão, essa atividade era automaticamente marcada como concluída e a atividade seguinte era desbloqueada.

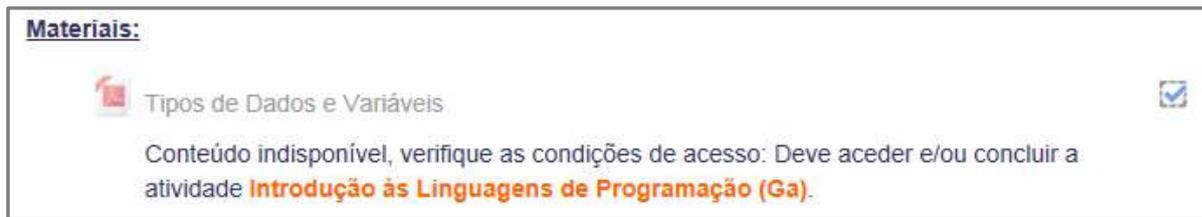


Figura 31 - Acesso Condicionado às Atividades

5.6.6.6 Princípio educacional: Acesso Restrito a Atividades

O acesso restrito às atividades e aos conteúdos (Figura 32) pode assumir diversas configurações, por exemplo, para aceder ao conteúdo “Operadores Aritméticos”, foram definidas as condições baseadas na obtenção de uma nota igual ou superior a dez e definida como condição de conclusão a visualização do conteúdo.

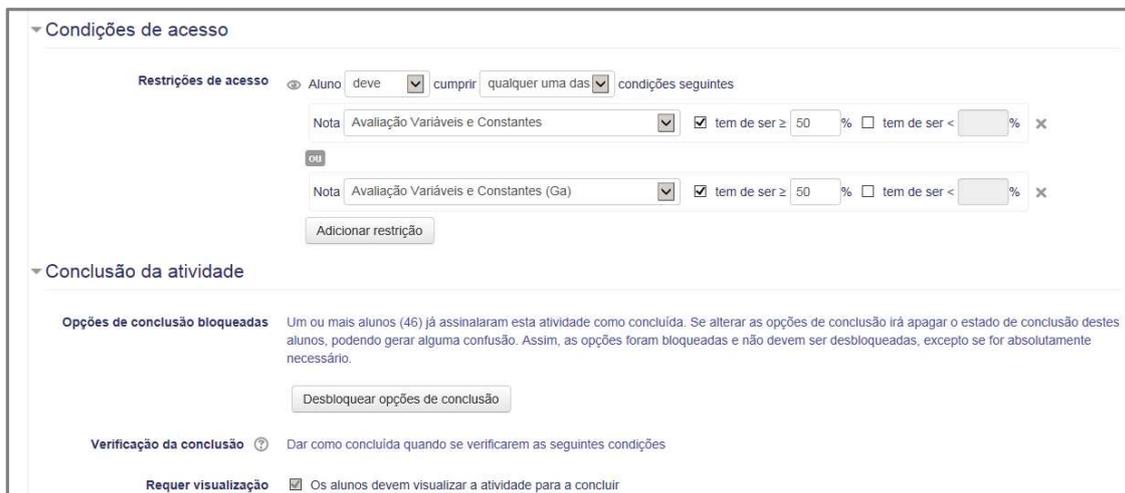


Figura 32 - Acesso Restrito às Atividades

5.6.6.7 Princípio educacional: Restrição no Tempo

O princípio educacional restrição no tempo das submissões, foi aplicado a todos os testes *online*. De acordo com o tipo de teste e com as perguntas definidas, foi estipulado o tempo necessário para que o estudante pudesse realizar o teste. Por exemplo, para realizar a atividade de avaliação *online* no módulo “Variáveis e Tipos de Dados”, foi definido como tempo limite, vinte minutos, enquanto no módulo “subprogramas”, foram definidos quarenta minutos, como tempo limite. Para definição do tempo, o docente realizou todas as avaliações e triplicou o tempo relativamente ao que demorou, a realizar cada uma das avaliações.

5.6.6.8 Mecânicas de jogo: Pontos e Quadro de Honra

Estas mecânicas foram definidas de modo a permitir ao estudante ganhar pontos e tornar visível para os colegas, os pontos que acumulou. Esperava-se que a inclusão destas mecânicas estimulasse a competição. A dinâmica associada contemplava as interações dos estudantes com os conteúdos e com as atividades. Por cada interação o estudante acumulava dois pontos. Quantas mais interações o estudante tivesse com os conteúdos e com as atividades, maior o número de pontos e conseqüentemente, a possibilidade de atingir o TOP5. Para a implementação foi utilizado o bloco *Ranking Block*, um *plug-in* adicional que foi instalado na plataforma e configurado em conformidade com o que era pretendido. O módulo adicional pode ser obtido a partir do endereço web do Moodle.

5.6.6.9 Mecânicas de jogo: Níveis

Como anteriormente revisto, os níveis permitem uma progressão e uma sequenciação, através de conteúdos e de atividades. Essa progressão e sequenciação foram implementadas através do acesso restrito às atividades e aos recursos. Os estudantes apenas tinham acesso aos módulos seguintes, quando completavam as atividades no módulo anterior e com nota mínima de aprovação na atividade de avaliação. Desse modo, os níveis e a evolução nos mesmos mostravam ao estudante a sua competência e progresso, assim como a sua posição no jogo.

5.6.6.10 Mecânicas de jogo: Medalhas

As medalhas são elementos simbólicos que são atribuídos aos estudantes por completar ou obter uma competência, um conhecimento, podendo ser visualizadas por outros, mostrando assim o seu domínio ou conhecimento. Neste estudo foram definidas cinco categorias de medalhas. O nome e o desenho das medalhas foram inspirados numa semântica associada à educação e ao tipo de utilizador de informática. De seguida descrevem-se as medalhas que foram criadas para utilização no curso *online* de programação: Caloiro, Geek, Super Geek, Guru e Super Guru.

O caloiro é o estudante que acaba de entrar na universidade. Nesse sentido, definiu-se esta categoria para o estudante que acede pela primeira vez ao curso *online* (Figura 33).



Figura 33 - Medalha Caloiro

O *geek*, estatuto que o estudante adquire quando deixa de ser caloiro. Passa a ser reconhecido pelos pares, como um indivíduo que se destaca pelo conhecimento adquirido (Figura 34). Nesta categoria foram definidas três subcategorias: O *geek* bronze, o *geek* prata e o *geek* ouro.



Figura 34 - Medalhas da Categoria Geek

O *super geek* é uma evolução no estatuto. O estudante deixa de ser um simples *geek* e passa ao estatuto de *super geek*. O *super geek* representa um indivíduo que evolui nos seus conhecimentos e que já ultrapassou diversos níveis com um desempenho considerável. As subcategorias bronze, prata e ouro foram igualmente definidas (Figura 35).



Figura 35 - Medalhas da Categoria *Super Geek*

Com o estatuto de *guru*, os estudantes assumem-se como indivíduos conhecedores e com uma perícia nos conceitos de programação, dado que ao alcançarem a primeira medalha, a medalha

de guru bronze, os estudantes já realizaram mais de 50% do curso em fundamentos de programação (Figura 36).



Figura 36 – Medalhas da Categoria *Guru*

O *super guru* é um estado de máximo conhecimento que o estudante pode alcançar. Ser *super guru* é sinónimo de ter alcançado o final do curso em fundamentos de programação e de ter sido medalha de ouro em todas as categorias anteriores. Esta medalha é única, não tendo qualquer subcategoria associada (Figura 37).



Figura 37 - Medalha Categoria *Super Guru*

Descritas as mecânicas utilizadas no curso *online* de fundamentos de programação, descrevem-se de seguida, as regras aplicadas às mecânicas de jogo.

Após o estudante aceder ao curso e efetuar as alterações no perfil, nomeadamente, adicionar a sua fotografia e atualizar a informação referente ao nome, a medalha caloiro é automaticamente atribuída. As medalhas da categoria *Geek* são atribuídas após o estudante realizar os módulos: módulo#1 (“Introdução às Linguagens de Programação”), módulo#2 (“Variáveis e Tipos de Dados”) e módulo#3 (“Entrada e Saída de Dados”). No entanto, a atribuição das medalhas apenas poderá ser feita, se os alunos obtiverem as seguintes avaliações nos módulos 1 e 2: nota

igual ou superior a catorze (*Geek Bronze*); nota superior a dezasseis (*Geek Prata*); nota superior a dezoito nas avaliações (*Geek Ouro*).

Tabela 15 - Tabela resumo (medalhas e critérios aplicados)

Módulos/ Atividades	Mecânicas de jogo: Medalhas	CrITÉRIOS Aplicados
- Editar o Perfil - Fórum Café Social	 Categoria Caloiro	Alteração dos campos do perfil de utilizador (foto); Colocar um post no fórum social com uma breve apresentação pessoal.
Atividade de Avaliação dos Módulos: “Variáveis e tipos de Dados” e “Entrada e Saída de Dados”	 Categoria Geek	A avaliação obtida num dos módulos deverá ser: Bronze – igual ou superior a 14; Prata – igual ou superior a 16; Ouro – igual ou superior a 18
Atividade de Avaliação do Módulo: “Operadores Aritméticos e Seleção”	 Categoria Super Geek	A avaliação obtida num deverá ser: Bronze – igual ou superior a 14; Prata – igual ou superior a 16; Ouro – igual ou superior a 18
Atividade de Avaliação do Módulo: “Repetição e Vetores”	 Categoria Guru	A avaliação obtida num deverá ser: Bronze – igual ou superior a 14; Prata – igual ou superior a 16; Ouro – igual ou superior a 18
Atividade de Avaliação no Módulo: “Subprogramas”	 Categoria Super Guru	Deverá ter obtido todas as medalhas de Ouro (<i>Geek, SuperGeek e Guru</i>) nos módulos anteriores e: - Obter na avaliação do módulo <i>Subprogramas</i> uma nota igual ou superior a 16

As medalhas da categoria *Super Geek* são atribuídas após o estudante realizar os módulos: módulo#3 (“Operadores Aritméticos”) e módulo#4 (“Mecanismos de Seleção”). Para obtenção das medalhas, deve ter cumprido com os seguintes requisitos: a) ter obtido pelo menos uma das medalhas geek; b) nota igual ou superior a catorze nas avaliações do módulo#3 e do módulo#4 (*Super Geek Bronze*); c) nota superior a dezasseis nas avaliações do módulo#3 e do módulo#4 (*Super Geek Prata*); d) nota superior a dezoito nas avaliações do módulo#3 e do módulo#4 (*Super Geek Ouro*).

A categoria Guru pode ser alcançada após o estudante ter realizado o módulo#5 (“Mecanismos de Repetição”) e o módulo#6 (“Vetores”), tendo cumprido com os seguintes requisitos: i) ter obtido pelo menos uma das medalhas *Geek* e *Super Geek*; ii) nota igual ou superior a catorze nas avaliações do módulo#5 e módulo#6 (*Guru Bronze*); iii) nota superior a dezasseis nas avaliações do módulo#5 e módulo#6 (*Guru Prata*); iv) nota superior a dezoito nas avaliações do módulo#5 e módulo#6 (*Guru Ouro*).

Por último, a categoria *Super Guru* pode ser alcançada com a conclusão do módulo#8 (“Subprogramas”). Os requisitos definidos para a obtenção desta medalha, são os seguintes: i) ter obtido nas categorias anteriores, todas as medalhas de ouro: *Geek*, *Super Geek* e *Guru* e nota superior a dezasseis no teste de avaliação do módulo#8.

Tabela 16 - Tabela Resumo das Mecânicas Aplicadas no Curso

Mecânicas de jogo aplicadas no curso	
Pontos	A mecânica de jogo pontos, foi aplicada a todas as atividades presentes no curso, como por exemplo, conteúdos pedagógicos, atividades de avaliação, atividades de submissão de trabalhos, entre outros.
Quadro de Honra	A mecânica de jogo quadro de honra, foi aplicada de forma a permitir, a contabilização e a disponibilização aos estudantes dos pontos obtidos.
Medalhas	Foram criadas e aplicadas cinco medalhas, sendo que três dessas medalhas têm categorias definidas. As medalhas foram atribuídos os seguintes nomes: <i>caloiro</i> , <i>geek</i> , <i>super geek</i> , <i>guru</i> e <i>superguru</i> . Nas medalhas <i>geek</i> , <i>super geek</i> e <i>guru</i> foram definidas as categorias: bronze, prata e ouro.
Níveis	Num contexto educativo e <i>online</i> , os níveis permitem uma progressão e sequenciação através de conteúdos e atividades. Assim sendo, esta mecânica foi aplicada através da sequenciação dos módulos e pela restrição na progressão dos mesmos, tornando essa sequenciação dependente da realização das atividades e da nota mínima em cada módulo.
Barra de progresso	A barra de progresso foi aplicada no curso

5.6.6.11 Mecânicas de jogo: Barra de Progresso das Atividades

A barra de progresso permite, ao estudante uma visualização gráfica, das atividades já concluídas, e das que ainda estão por concluir. Todas as atividades que o estudante tem de realizar, sejam elas atividades de avaliação ou outras, são identificadas na barra de progresso. Caso a atividade esteja concluída, é assinalada com cor verde, caso contrário, mantém-se na cor azul. Nas atividades de avaliação, caso o estudante as tenha iniciado, mas ainda não tenha concluído, é assinalada a cor amarela. Esta mecânica foi implementada com recurso a uma ferramenta externa da versão base do Moodle. Foi instalado e configurado um *plug-in Progress Bar*, acessível para download a partir do site oficial do Moodle (<http://moodle.org>).

5.7 Fase 5: avaliação - resultados - atitudes dos estudantes face aos elementos de gamificação

Após a conclusão do curso, os estudantes foram convidados a responder a um questionário *online* (Anexo B). O questionário é composto por diversas questões revistas na literatura, entre as quais, perguntas relacionadas com os elementos de gamificação utilizados no curso, nomeadamente: barra de progresso (Tabela 17), quadro de honra (Tabela 18), medalhas (Tabela 19) e pontos (Tabela 20). Foi definida uma escala *likert* de sete valores (*1-Discordo totalmente, ..., 7-Concordo Totalmente*). Apresentam-se neste subcapítulo, os resultados obtidos.

Da análise aos dados, verifica-se que os estudantes consideram a barra de progresso útil ($M=5.593$; $SD=1.367$) e recorrem com frequência à mesma para identificar a sua evolução no curso *online* ($M=5.102$; $SD=1.522$), cf. gráfico 15.

Tabela 17 - Dimensão Barra de Progresso

Dimensão	Código	Variáveis	Autor
Barra de Progresso	PG1	Acho a barra de progresso útil	Autoria Própria
	PG2	Visualizo diversas vezes a barra de progresso para identificar as atividades realizadas e por realizar	

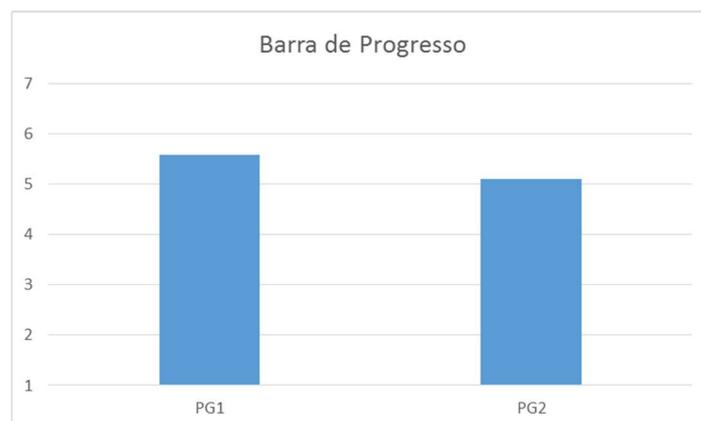


Gráfico 15 – Atitude dos Estudantes Face à Barra de Progresso

Verifica-se na análise aos dados referentes ao quadro de honra, que os estudantes o consideram útil ($M=4.898$; $SD=1.253$) e que o consultam para visualizar os seus resultados ($M=4.472$; $SD=1.683$). Não consideram que os seus resultados visíveis para os restantes estudantes seja

um fator incomodativo ($M=2.843;SD=1.607$), contudo, observa-se que para alguns dos estudantes, não ter o seu nome no quadro de honra é ligeiramente incomodativo ($M=3.083;SD=1.624$), cf. Gráfico 16.

Tabela 18 – Dimensão - Quadro de Honra (*LeaderBoard*)

Dimensão	Código	Variáveis	Autor
Quadro de Honra	L1	Acho o quadro de pontos útil	Autoria Própria
	L2	Consulta com frequência para visualizar a pontuação	
	L3	Sinto-me incomodado com os resultados listados e visíveis para todos.	
	L4	Sinto-me incomodado com os resultados listados e o meu nome não estar nessa lista.	



Gráfico 16 – Atitude dos Estudantes Face ao Quadro de Honra

Relativamente às medalhas, observa-se que são consideradas motivadoras ($M=5.519;SD=1.391$) e úteis no processo de aprendizagem ($M=5.38;SD=1.331$), tendo um efeito positivo moderado no comportamento dos estudantes no curso ($M=5.519;SD=1.264$), os quais concordam moderadamente com o aspeto visual das medalhas ($M=5.435;SD=1.27$), assim como os critérios que foram definidos para a obtenção das medalhas, são também moderadamente aceites pelos estudantes ($M=5.491;SD=1.264$). Relativamente à sua utilização em todas as disciplinas, os estudantes não têm uma posição claramente definida ($M=4.991;SD=1.507$), cf. Gráfico 17.

Tabela 19 – Dimensão - Medalhas

Dimensão	Código	Variáveis	Autor
Medalhas	B1	São motivadoras	Autoria Própria
	B2	Perturbam o meu trabalho	
	B3	Tentar obter as medalhas tem um efeito positivo no meu comportamento ao longo do curso	
	B4	O aspeto visual das medalhas é bom	
	B5	Estou satisfeito com os critérios para obter as medalhas	
	B6	Acho que as medalhas poderiam ser usadas na plataforma de <i>e-Learning</i> (Moodle) em todas as disciplinas do plano curricular	
	B7	Considero que as medalhas são úteis no processo de aprendizagem	

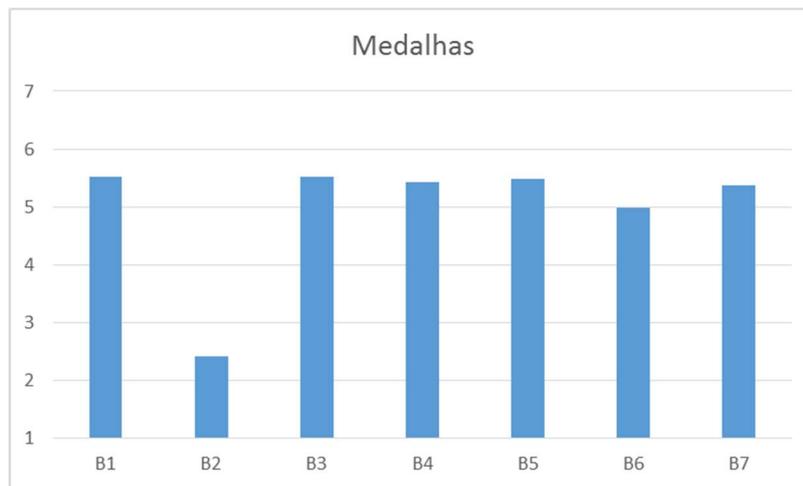


Gráfico 17 – Atitude dos Estudantes Face às Medalhas

Relativamente ao elemento de gamificação pontos, observa-se que os estudantes os consideram motivadores ($M=5.565$; $SD=1.162$), com um efeito positivo no seu comportamento ($M=5.38$; $SD=1.182$) e úteis no processo de aprendizagem ($M=5.398$; $SD=1.199$). Os pontos não perturbam o seu trabalho ($M=2.472$; $SD=1.573$) e os critérios para a atribuição dos pontos, também estão em concordância com os mesmos ($M=5.5$; $SD=1.172$). Relativamente à sua utilização em todas as disciplinas na plataforma *Moodle*, manifestam-se pela neutralidade, nem discordando, nem concordando ($M=4.861$; $SD=1.475$), cf. Gráfico 18.

Tabela 20 – Dimensão - Pontos

Dimensão	Código	Variáveis	Autor
Pontos	P1	São motivadores	Autoria Própria
	P2	Perturbam o meu trabalho	
	P3	Têm um efeito positivo no meu comportamento ao longo do curso	
	P4	Estou satisfeito com os critérios definidos para ganhar pontos	
	P5	Os pontos poderiam ser utilizados na plataforma de <i>e-Learning</i> (Moodle) em todas as disciplinas	
	P6	Considero que as medalhas são úteis no processo de aprendizagem	

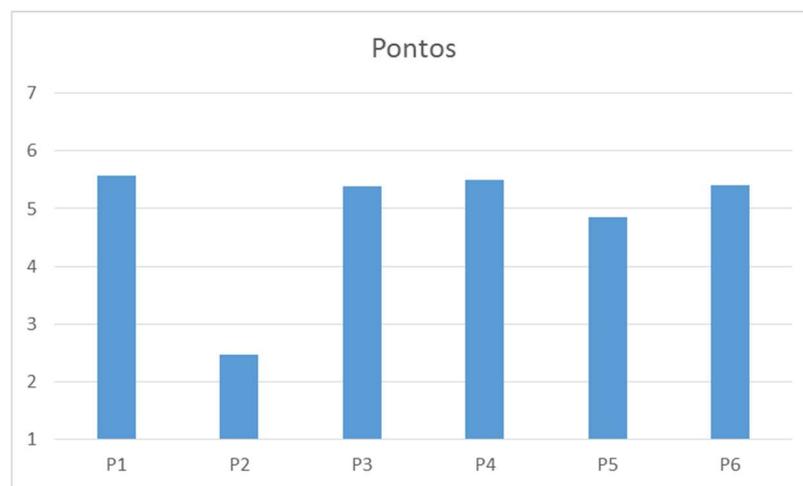


Gráfico 18 – Atitude dos Estudantes Face aos Pontos

Efetuuou-se uma comparação da atitude dos estudantes face às medalhas e aos pontos (Gráfico 19) e verificou-se que a atitude, perante estes dois elementos de jogo é muito similar. Observam-se ligeiras variações nos itens B3-P3 (efeito no comportamento ao longo do curso), onde as medalhas têm um efeito ligeiramente superior ao dos pontos. Nos itens B6-P5 (utilização dos elementos em todas as disciplinas do curso), verifica-se a mesma tendência com as medalhas, a apresentar uma ligeira preferência pela sua utilização nas disciplinas *online* da licenciatura.

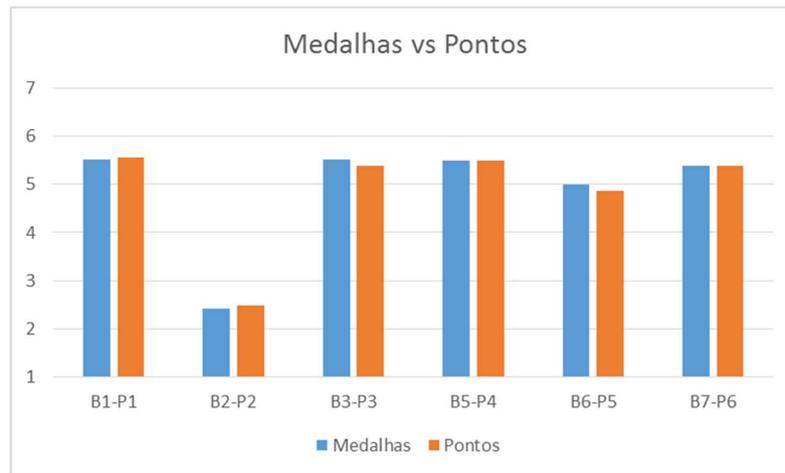


Gráfico 19 - Atitude dos Estudantes Face às Medalhas vs. Pontos

5.8 Fase 6: Comunicação

Na fase 6 da DSRM, devem-se comunicar o problema e a sua importância, o artefacto, a sua utilidade e a sua inovação, o rigor do processo de *design*, a sua eficácia para os investigadores e outras audiências relevantes. Nesse sentido, e no contexto deste estudo foram publicados dois artigos científicos (Tabela 21) e efetuadas diversas comunicações / divulgação em eventos relacionados com a temática e descritas seguidamente.

5.8.1 Comunicação em eventos:

- Apresentação do SmartMOOC na semana europeia do *CodeWeek*, em outubro de 2016. O *CodeWeek* é uma iniciativa da união europeia com a realização de diversas atividades relacionadas com a programação de computadores, e que tem como objetivo principal, promover a aprendizagem de técnicas de programação e fortalecer as competências digitais dos participantes (Figura 38).
- Sessão de entrega de certificados de conclusão aos estudantes que concluíram o SmartMOOC de Fundamentos de Programação, em abril de 2017 (Figura 39).
- Apresentação do trabalho de investigação na Pós-Graduação de Informática Aplicada às Organizações, ISCTE, abril de 2017.

- Apresentação do trabalho de investigação na Pós-Graduação de Informática Aplicada às Organizações, ISCTE, novembro de 2017.
- Participação na Noite dos Investigadores, em setembro de 2017 (Figura 40).
- Entrevista ao programa 90 segundos de ciência. (<http://www.90segundosdeciencia.pt/sobre-90-seg/>)



Figura 38 – Apresentação do SmartMOOC na CodeWeek, ISCTE, abril 2017



Figura 39 - Entrega dos Certificados de Conclusão do SmartMOOC, abril 2017



Gamificação e Robótica no Ensino da programação / Programming Learning with Gamification and Robotics | ISCTE-IUL, ISTAR

Com o objetivo motivar os estudantes para a aprendizagem de programação de computadores serão disponibilizadas duas atividades. Por um lado, os participantes vão poder realizar um conjunto de desafios que correspondem a tarefas em que é necessário programar. Realizando cada um dos desafios vão obtendo pontos, medalhas e posição na lista de melhores jogadores. Por outro lado, na segunda atividade os participantes poderão programar um robot utilizando conceitos básicos de programação ou algoritmos mais sofisticados.

Organização: ISCTE-IUL, ISTAR

Investigadores: Carlos J. Costa, Manuela Aparício e Martinha Piteira

Local: Museu Nacional de História Natural e da Ciência – ULISBOA

Figura 40 – Divulgação do SmartMOOC na Noite dos Investigadores, 2017 in Site <http://noitedosinvestigadores.org/programa/lisboa/>

5.8.2 Comunicação: Publicações Científicas

Na Tabela 21, descrevemos as publicações científicas que foram realizadas no âmbito do trabalho desenvolvido no presente Capítulo.

Tabela 21 – DSRM (Fase 6): Comunicação - Publicações Científicas

Publicações	Descrição
Piteira, M., & Costa, C. J. (2017). Gamification: Conceptual framework to online courses of learning computer programming. Em 2017 12th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1–7). ISCTE-IUL - Lisboa. https://doi.org/10.23919/CISTI.2017.7975695 .	Esta publicação contém a descrição da <i>framework</i> teórica proposta para cursos online de programação gamificados e descrita neste documento no subcapítulo 5.5
Piteira, M., Costa, C.J., & Aparício, M. (2017). A conceptual framework to implement gamification on online courses of computer programming learning: implementation. Em ICERI2017 Proceedings (pp. 7022–7031). Seville, SPAIN: IATED.	Nesta publicação foi descrito todo o processo de operacionalização da <i>framework</i> e a descrição dos resultados relativos à atitude dos estudantes face aos elementos gamificados presentes no curso online e descrito neste documento no subcapítulo 5.6

5.9 Discussão de resultados

Da análise aos dados, verifica-se que globalmente os estudantes têm uma atitude positiva face aos elementos de gamificação incorporados no curso *online*: quadro de honra (*leaderboard*), barra de progresso (*progress bar*), medalhas (*badges*) e pontos (*points*).

De acordo com Werback & Hunter (2012) o quadro de honra é um dos elementos de jogo mais utilizados na gamificação. Os quadros de honra são vistos como elementos associados à competição e à comparação do desempenho e não apenas como um mecanismo de *ranking*. O quadro de honra é visto pelos utilizadores, que utilizam sistemas com quadros de honra como um mecanismo que permite visualizar e comparar o seu desempenho com o desempenho de outros utilizadores, e são também uma forma de identificar quanto “*competidores*” são (Cheong, Filippou, & Cheong, 2014). Em certa medida estes elementos são motivadores e úteis no processo de aprendizagem, incentivando os estudantes a desafiarem-se a si próprios, e desafiados pelos resultados dos restantes colegas, o que contribui para os estudantes continuarem envolvidos com aprendizagem. Nesse sentido, os resultados obtidos estão alinhados com o descrito anteriormente. Os estudantes consideram que o quadro de honra é útil, visualizam-no com regularidade para consultar a sua pontuação e a pontuação dos restantes colegas. Os estudantes consideram não ser, um problema, os seus resultados serem visualizados pelos restantes colegas, contudo, observa-se que não terem o seu nome no quadro de honra é ligeiramente incomodativo. Os quadros de honra, são na sua maioria construídos a partir de um sistema de pontuação, e estes elementos de jogo podem também fornecer feedback e motivação, Por exemplo, se o sistema de pontos for bem desenhado para atribuir pontos, a uma atividade realizada, fornece retorno ao estudante sobre o seu desempenho e contribui para que a gamificação tenha significado (Cheong et al., 2014). Esta combinação, entre o sistema de pontos e o quadro de honra, incentiva e motiva os estudantes (Fotaris et al., 2016). Ao analisarmos a atitude dos estudantes face ao elemento de jogo pontos, verifica-se que os resultados estão em conformidade com a revisão da literatura. Os estudantes consideram que os Pontos são motivadores, e paralelamente têm um efeito positivo no seu comportamento ao longo do curso, conseqüentemente consideram-nos também úteis no processo de aprendizagem da programação. De acordo com Seaborn & Fels (2015), a gamificação pode influenciar o comportamento e atitude dos estudantes se for desenhada, neste sentido, os critérios definidos para a atribuição de pontos deve estar alinhada com os objetivos educacionais. Neste aspeto em particular, os estudantes manifestaram-se satisfeitos com os critérios definidos.

A barra de progresso é um dos elementos de jogo que permite um retorno imediato ao estudante do seu progresso no curso, facilitando a rápida visualização das atividades realizadas e por realizar. A atitude positiva demonstrada pelos estudantes relativa à sua utilidade, é reforçada com a frequência com que os estudantes o visualizam. Estes resultados demonstram que no processo de gamificação o retorno ao estudante é um fator a ter em consideração no desenho de cursos *online* gamificados. Esta observação é suportada no trabalho de Cheong, Filippou, & Cheong (2014) que segundo os autores, o interesse pela barra de progresso, pode ser um indicador de uma certa preferência pela obtenção de retorno, enquanto reforça a interação social, quando as atividades são definidas de forma a serem realizadas por equipas de trabalho. Nesse sentido, a barra de progresso é vista como um mecanismo que fornece retorno, e em certa medida fornece também algum grau de motivação. Em particular, as barras de progresso são vistas como motivacionais e aumentam o interesse através da visualização do progresso corrente. A barra de progresso está relacionada com a concretização de um objetivo e é indicadora do avanço para os próximos níveis (“*levelling up*”) (Cheong et al., 2014).

As medalhas são um dos elementos recorrentemente mais utilizados em contexto educacional (Dicheva et al., 2015). Relativamente à atitude dos estudantes face a estes elementos os resultados demonstraram que os consideram motivadores, que a sua utilização no processo de aprendizagem é útil e que influenciou o seu comportamento ao longo do curso, na tentativa, de obterem as medalhas que estavam disponíveis. Estes resultados estão em consonância com conclusões similares reportadas na literatura, que sugerem que as medalhas são motivadoras e que influenciam o comportamento dos estudantes (Hamari, 2017; Kapp, Blair, & Mesch, 2013; Werbach & Hunter, 2015). As medalhas são também um dos elementos mais estudados na literatura (Cheong et al., 2014) e diversos estudos reportam a sua utilização, por exemplo, para retorno e recompensa por completarem determinadas tarefas, pela acreditação da aquisição de uma determinada competência, entre outros (Aldemir, Celik, & Kaplan, 2018). Como referido por Hamari (2017) as medalhas podem ter elementos de narrativa com propósitos motivacionais. Nesse sentido, as medalhas que os estudantes poderiam colecionar foram criadas com semânticas familiares aos estudantes de informática, com propósito de os motivar a colecionar as medalhas, e nos resultados observados, relativamente ao aspeto visual das medalhas, os estudantes manifestaram-se positivamente agradados com o seu aspeto. Relativamente aos critérios definidos para a obtenção das medalhas, os estudantes consideraram-nos adequados e satisfeitos com os mesmos. À semelhança dos critérios

estabelecidos para a atribuição dos pontos, salienta-se a importância do alinhamento com as atividades de aprendizagem (Seaborn & Fels, 2015).

Na comparação efetuada entre atitudes face às medalhas e pontos, observou-se que ambos os elementos são considerados motivadores e úteis no processo de aprendizagem da programação, não existindo uma preferência significativa, por um dos elementos. Relativamente à influência que os elementos têm no comportamento do estudante ao longo do curso, verificou-se que as medalhas têm uma influência ligeiramente mais positiva comparativamente à influência que os pontos têm nos estudantes. Estes resultados têm suporte na literatura. Os estudantes sentem-se desafiados e motivados a colecionarem todas as medalhas, não apenas para alcançarem os objetivos educacionais, mas por querem exibir as medalhas, em certa medida agrada-os pelo aspeto visual da medalha e por aquilo que a medalha possa representar (Knutas et al., 2014).

5.10 Conclusões do Estudo 3

Neste capítulo, propôs-se uma *framework* teórica para implementação de cursos *online* de programação gamificados. A *framework* é composta pelas dimensões: público-alvo, objetivos gerais, resultados de aprendizagem, tópicos, conteúdos, gamificação (princípios de desenho educacional, mecânicas de jogo), absorção cognitiva, *flow* e personalidade. As dimensões propostas resultaram da revisão da literatura relacionada com a aprendizagem da programação, problemas e soluções, estratégias educacionais que motivem e envolvam o estudante na aprendizagem, tais como: gamificação; absorção cognitiva; *flow*; e personalidade. Descreveu-se a implementação da *framework* teórica de cursos *online* gamificados. Para a implementação da *framework*, foi desenhado e operacionalizado um curso *online* de fundamentos de programação, suportado pela plataforma de *e-Learning* Moodle.

Identificou-se também a atitude dos estudantes, perante os elementos de gamificação utilizados no curso, mais concretamente a sua atitude em relação: às medalhas, aos pontos, ao quadro de honra e à barra de progresso. Relativamente à utilidade dos elementos barra de progresso e quadro de honra, concluímos que os estudantes consideraram ligeiramente mais útil a barra de progresso, a qual visualizavam com regularidade para identificar o seu progresso no curso. No quadro de honra, não se sentem aparentemente incomodados por o seu nome não constar do quadro, como também não parecem incomodados, por o seu nome constar deste e por todos visualizarem o seu nome, nessa lista. O quadro de honra é consultado com frequência, de forma a visualizar quem nele está presente. Relativamente aos elementos medalhas e pontos, os

estudantes consideram que são motivadores, notando-se uma ligeira preferência pelas medalhas, as quais, segundo os estudantes, são elementos que não os perturbam durante a realização do seu trabalho, tendo até um efeito positivo no seu comportamento durante o curso. Denota-se que as medalhas têm um efeito ligeiramente mais positivo. As medalhas assumem também uma ligeira preferência pela sua utilização em todas as disciplinas que são disponibilizadas na plataforma, fazendo parte do plano curricular oficial do curso de informática.

No capítulo seguinte, pretende-se elaborar um estudo referente à adoção da gamificação nos cursos *online* de programação gamificados. Para esse efeito, será construído um questionário, que será lançado a uma amostra de estudantes do ensino superior politécnico e que frequentaram o curso *online* gamificado de fundamentos de programação, descrito no capítulo anterior.

6 PROPOSTA DE MODELO DE ADOÇÃO PARA CURSOS ONLINE DE PROGRAMAÇÃO GAMIFICADOS - ESTUDO 4

6.1 Introdução

No capítulo anterior, propôs-se uma framework conceitual-teórica gamificada, para cursos online. Descreveu-se também o processo de operacionalização e validação da framework teórica. No presente estudo, pretende-se investigar quais as variáveis que influenciam a adoção de cursos online gamificados. Nesse sentido, pretende-se propor um modelo teórico de adoção da gamificação para cursos online de programação. Este capítulo corresponde ao artigo de revista indexado na SCIMAGO, SCOPUS, WebScience e GoogleScholar.

6.2 Objectivos

Como objetivo para o presente estudo pretende-se propor um modelo teórico-conceitual de adoção da gamificação para cursos online, e respetiva validação .

6.3 Enquadramento teórico

6.3.1 A adoção da tecnologia

A adoção da tecnologia é uma das teorias largamente utilizadas para explicar o uso e a aceitação individual dos sistemas de informação (SI) e das tecnologias da informação (TI). Com base nesta teoria diversos estudos e.g. (Tarhini, Hone, Liu, & Tarhini, 2017) têm sido realizados com o objetivo de identificar fatores intrínsecos e extrínsecos envolvidos nas decisões, intenções e satisfação individual, na aceitação e uso dos SI e das TI, através de diversos métodos e testes (Venkatesh et al., 2002).

Diversos modelos têm sido propostos para estudar a adoção da tecnologia, e um dos mais utilizados pelos investigadores foi proposto por Davis em 1986, e tem como objetivo explicar o comportamento dos utilizadores na utilização do computador. O modelo explica o relacionamento causal entre as variáveis de aceitação do utilizador e a utilização atual do sistema, procurando o comportamento do utilizador através do conhecimento da utilidade e da facilidade percebida. Este modelo é útil não apenas para prever, mas também para descrever e

identificar a razão da rejeição de uma determinada tecnologia ou sistema e conseqüentemente implementar medidas corretivas, (Davis, 1986, 1989). Este modelo foi escolhido para ser utilizado neste estudo porque baseado na revisão da literatura o TAM é um dos modelos mais utilizados, e mais influente assim como um modelo altamente preditivo na adoção dos SI (Ferreira, Costa, Aparicio, & Aparicio, 2017; Rodrigues, Costa, & Oliveira, 2013; Rodrigues, Oliveira, et al., 2016b; Sousa, Costa, & Aparicio, 2017; Venkatesh, Brown, Maruping, & Bala, 2008; Venkatesh & Davis, 2000; Venkatesh et al., 2002)

6.3.2 Conceito de *Flow*

No âmbito do presente estudo de investigação, pretende-se estudar o nível de envolvimento e absorção do estudante com o curso *online* gamificado através da incorporação da dimensão *flow*. O conceito de *flow* foi inicialmente estudado por Csikszentmihalyi (1975), e descreve um estado de completa absorção numa determinada atividade, sem a existência de incentivos extrínsecos (Csikszentmihalyi, 1975), designado de “*flow state*” ou “*flow experience*”. Os autores Engeser & Rheinberg (2008) caracterizaram o fluxo como o equilíbrio entre a percepção das competências a atingir e a percepção da dificuldade. Contribui para esse equilíbrio a atividade a realizar ser coerente, ter alguma lógica associada, não ter aspetos contraditórios e fornecer um retorno claro e não ambíguo. O indivíduo sente-se assim desafiado e confiante que tem tudo sobre controlo com um elevado grau de concentração na atividade derivada de uma atenção profunda. Nesse sentido, o estado de *flow* tem um aspeto funcional forte, na experiência que o indivíduo experiencia altamente concentrado e com um desafio aceitável enquanto mantém o controlo. Na literatura revista são diversos os autores que conduziram estudos relacionados com o *flow* (Chang, Liang, Chou, & Lin, 2017; Hamari & Koivisto, 2014; Kiili, 2005; Liao, 2006; Lu, Zhou, & Wang, 2009; Shernoff, Csikszentmihalyi, Schneider, & Shernoff, 2014). Por exemplo, Kiili (2005) analisou o “*flow experience*” em jogos educacionais. Liao (2006) conduziu um estudo relativo às respostas emotivas e cognitivas dos estudantes na utilização de plataformas *online* de educação a distância. Lu, Zhou & Wang (2009) conduziram um estudo com o objetivo de explorar a aceitação das mensagens instantâneas por parte da população chinesa, utilizando como base teórica, o modelo de aceitação de tecnologia e a teoria do *flow*. Shernoff et al. (2014) avaliaram o efeito do fluxo, no envolvimento nos alunos com as aulas do ensino superior.

6.3.3 As cinco dimensões da personalidade

As cinco dimensões da personalidade dos indivíduos (Goldberg, 1990; Gosling et al., 2003; McCrae & Costa, 1987) são conhecidas como CANOE (*Conscientiousness, Agreeableness, Neuroticism, Openness to Experience, Extroversion*). “A personalidade é uma mistura de valores, temperamentos, estratégias, motivação, entre outros”, (Bontempo, Napier, Hayes, & Brashear, 2014), e também uma forma de pensar ou atuar em variadas situações. Gosling, Rentfrow, & Swann (2003) propuseram um modelo que é medido através de cinco fatores estruturados, composto por extroversão (sociabilidade e assertividade), agradabilidade (merecedor de confiança, cooperante), conscienciosidade, estabilidade emocional (ansiedade e depressão), abertura a novas experiências (imaginativo e pensamento independente), sendo o indivíduo avaliado de acordo com o grau que exibe em cada uma das dimensões. Vários autores têm investigado o impacto das cinco dimensões da personalidade em diversos contextos. Por exemplo, Özbek, Almaçık, Koc, Akkılıç, & Kaş (2014) conduziram um estudo para avaliação do impacto das dimensões da personalidade na aceitação da tecnologia, nomeadamente dos *smartphones*. Também Cheung (2016) conduziu um trabalho de investigação que com o objetivo de identificar o impacto das dimensões da personalidade no envolvimento dos utilizadores com ferramentas de produção gamificadas. Diversos outros autores têm investigado o impacto da personalidade e.g. (Aparicio, Bacao, & Oliveira, 2017; Buckley & Doyle, 2017; Chamorro-Premuzic & Furnham, 2009; Chamorro-Premuzic, Furnham, & Lewis, 2007; Duff, Boyle, Dunleavy, & Ferguson, 2004; Ngidi, 2013; Vasileva-Stojanovska, Malinovski, Vasileva, Jovevski, & Trajkovik, 2015), com o objetivo de identificar o impacto das dimensões da personalidade, nas várias abordagens pedagógicas utilizadas em contexto educacional.

6.4 Modelo proposto e hipóteses

O modelo é baseado na teoria de adoção da tecnologia (Davis, 1986, 1989; Davis, Bagozzi, & Warshaw, 1992); no modelo das cinco dimensões da personalidade (Gosling et al., 2003); e no conceito de *Flow* (Csikszentmihalyi, 1975). Na Tabela 22, são apresentadas as dimensões para um curso *online* gamificado de fundamentos de programação.

Tabela 22 - Dimensões do Modelo Proposto

Dimensão	Conceito	Autor
CANOE	Modelo das cinco dimensões de personalidade	
Extroversão	Mede o grau de predisposição do indivíduo para experimentar estados emocionais positivos	
Agradabilidade	Mede o grau de facilidade do indivíduo em se relacionar com os outros	
Conscienciosidade	Mede o grau em que a pessoa se define como escrupulosa, cuidadosa e perseverante.	(Gosling et al., 2003)
Estabilidade Emocional	Mede o grau de percepção relativo à tendência de vivenciar estados emocionais negativos	
Abertura a Novas Experiências	Mede o grau dos interesses do sujeito: variado campo de interesses ou antes prefere atividades limitadas	
Facilidade de Utilização	Medição do grau em que um aluno acredita que a plataforma e o curso <i>online</i> , podem ser utilizados sem esforço.	(Davis, 1993; Venkatesh & Davis, 2000)
Prazer na Utilização	Medição do prazer de utilização percebido pelo aluno, na utilização da plataforma e do curso <i>online</i> , sem ter em conta as consequências do seu desempenho.	(Davis, 1992)
Percepção da Utilidade	Medição do grau atribuído pelo utilizador à utilidade de um dado sistema e à contribuição desse sistema para aumentar o seu desempenho	(Davis, 1989)
Intenção de Utilizar	Corresponde à atitude comportamental dos estudantes face à eventual intenção de frequentar o curso <i>online</i> gamificado.	(Davis, 1989)
<i>Flow</i>	Mede o grau de completa absorção que utilizador experiência nas atividades do curso <i>online</i> gamificado.	(Csikszentmihalyi, 1975)
Uso	Corresponde à frequência de utilização para realizar as atividades de aprendizagem.	(Davis, 1992)

Dado que as dimensões de personalidade são fatores que influenciam as percepções ou causam diferentes respostas em situações similares, os estudantes perante uma tecnologia ou abordagem pedagógica podem ter diferentes respostas e percepções. Assim, com a base nos estudos anteriores formulou-se a seguinte hipótese:

(H1). *CANOE é uma dimensão refletiva de segunda ordem que se reflete em: extroversão, agradabilidade, conscienciosidade, estabilidade emocional, abertura a novas experiências.*

Baseados nas dimensões da personalidade diversos autores e.g. (O’Cass & Fenech, 2003; Shih & Fan, 2013) identificaram a existência de evidências em como as características individuais do indivíduo podem influenciar a aceitação da tecnologia. Verificando-se, assim que essas características são um bom preditor para a facilidade de utilização, percepção de prazer e utilidade percebida. Nesse sentido, formulam-se as seguintes hipóteses:

(H2a). *CANOE influencia positivamente a percepção do prazer na utilização do curso online gamificado.*

(H2b). *O CANOE influencia positivamente a percepção da facilidade de utilização do curso online gamificado.*

(H2c). *O CANOE influencia positivamente a percepção de utilidade do curso online gamificado.*

O *flow* é um estado de absoluto envolvimento e absorção com uma determinada tarefa. De acordo com vários autores (Kiili, 2005; Liao, 2006; Lu et al., 2009) o *flow* é um bom preditor para a percepção de prazer e intenção de uso de um curso *online* gamificado. Tendo em consideração que o estado de *flow* resulta numa experiência gratificante, o *flow* e o prazer de utilização deverão estar relacionados (Sherry, 2004). Deste modo, formulou-se as seguintes hipóteses:

(H3a). *O flow influencia positivamente a percepção de prazer na utilização do curso online gamificado.*

(H3b). *O flow influencia positivamente a intenção de uso do curso online gamificado.*

A facilidade de utilização, a percepção de prazer e a percepção de utilidade são fatores que influenciam a intenção de uso de uma determinada tecnologia (Childers, Carr, Peck, & Carson, 2001). A facilidade de utilização (Costa, Silva, & Aparício, 2007) refere-se à medida, que leva o estudante a acreditar que pode utilizar a tecnologia com uma baixa carga cognitiva. Consequentemente, a facilidade de utilização gera no estudante uma sensação de satisfação, concretizando-se em prazer de utilização da plataforma (Rodrigues, Oliveira, & Costa, 2016a). Diversos estudos demonstraram uma relação entre a facilidade de utilização, a percepção de prazer e a percepção de utilidade (Costa, Ferreira, Bento, & Aparicio, 2016; Davis, 1989; Teo & Noyes, 2011). Nesse sentido, formulou-se as seguintes hipóteses.

(H4a). *A facilidade de utilização influencia positivamente a percepção de prazer na utilização do curso online gamificado.*

(H4b). *A facilidade de utilização influencia positivamente a percepção de utilidade do curso online gamificado.*

O prazer na utilização encontra-se associado à motivação intrínseca, tal como os sentimentos de diversão que os estudantes podem experienciar quando estão a utilizar a aplicação (Van der Heijden, 2004). Esperamos que o prazer na utilização esteja positivamente associado à intenção

de utilizar o curso *online* gamificado, tendo em consideração que a integração de elementos de gamificação no curso adiciona “elementos diferenciadores e apelativos”, para atividades pouco apelativas. O prazer de utilização é entendido pelos autores Venkatesh & Davis (2000) como sendo o nível de agradabilidade resultante da atividade do uso do sistema independentemente do desempenho alcançado com o uso. O prazer de utilização como uma motivação intrínseca tem sido percebido como tendo um impacto significativo no utilizador e por conseguinte um impacto numa variável de motivação extrínseca como a perceção da utilidade (Davis et al., 1992; Koufaris, 2002; Van der Heijden, 2004; Venkatesh et al., 2002). Quando a utilização de uma tecnologia gera diversão e prazer, os utilizadores intrinsecamente adotam essa tecnologia. Como referido anteriormente, a gamificação introduz elementos de entretenimento e envolvimento e os utilizadores percebem como prazeroso a utilização desses sistemas de aprendizagem que incorporam esses elementos. Assim sendo, formulou-se as seguintes hipóteses:

(H5). O prazer na utilização influencia positivamente a intenção de utilização do curso online gamificado.

(H6). A perceção da utilidade influencia positivamente a intenção de utilização do curso online gamificado.

O modelo de adoção de tecnologia, nomeadamente nos cursos *online* em formato *e-Learning* (Aparicio et al., 2016), propõe que os fatores externos afetam a intenção de uso e o atual uso da tecnologia através dos efeitos mediados na perceção da facilidade de utilização e perceção de utilidade (Davis, 1986, 1989; Ferreira et al., 2017). Nesse sentido, formulou-se a seguinte hipótese:

(H7). A intenção de utilização influencia positivamente o uso do curso online gamificado.

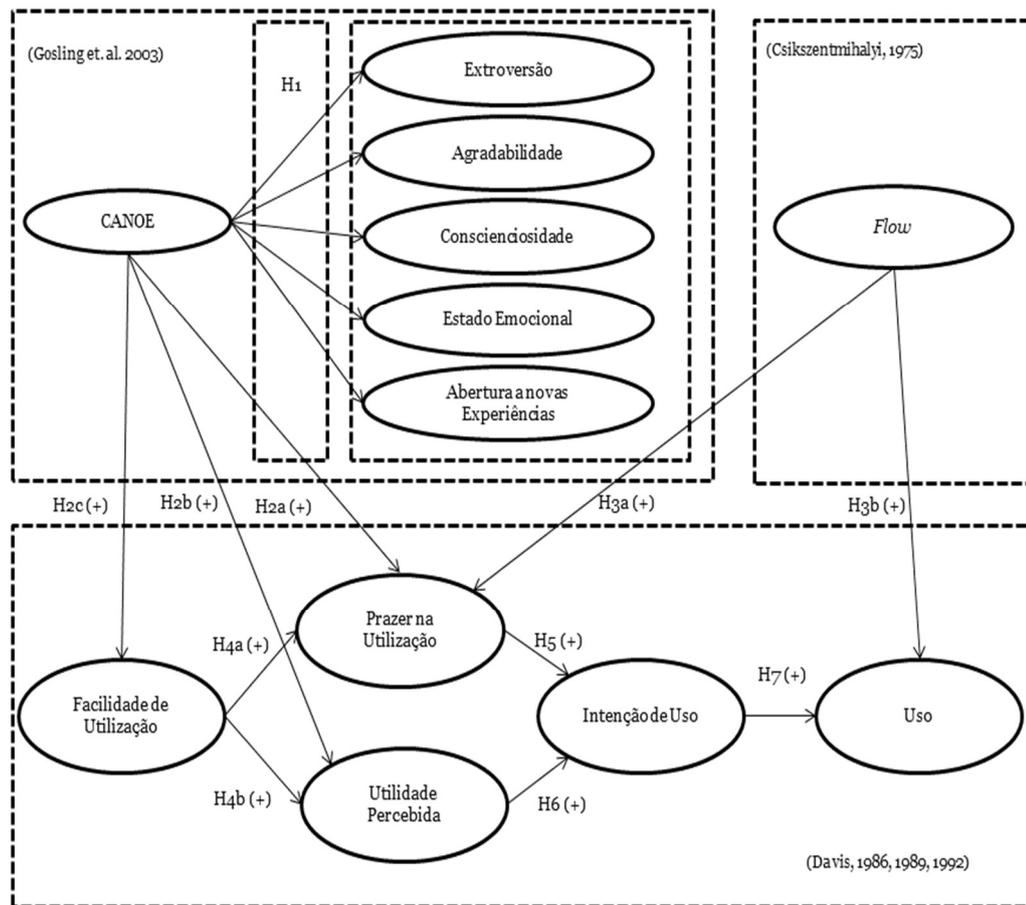


Figura 41 - Modelo de Investigação

6.4.1 Operacionalização das dimensões e recolha de dados

As dimensões foram operacionalizadas de acordo com escalas validadas e descritas na Tabela 24. Para tal foi construído um questionário, que seguiu um conjunto de fases (Ilustração 2) composto por duas partes: a primeira parte com dados gerais dos estudantes com a finalidade de caracterizar os indivíduos; e uma segunda parte composta pelos itens que correspondem à operacionalização das dimensões do modelo. De forma a avaliar o modelo teórico, os autores utilizaram o método quantitativo, lançando um instrumento de avaliação, concretamente, um questionário que foi apresentado a uma amostra de estudantes do ensino superior. De modo a proceder à recolha de dados, foi implementado um curso gamificado *online* de fundamentos de programação. A implementação do curso foi efetuada com recurso a uma framework concetual para implementação da gamificação em cursos *online* de programação, previamente desenvolvida pelos autores (Piteira & Costa, 2017; Piteira, Costa, & Aparício, 2017). O curso tem um tempo médio de conclusão de 20 horas e os estudantes frequentaram-no livremente.

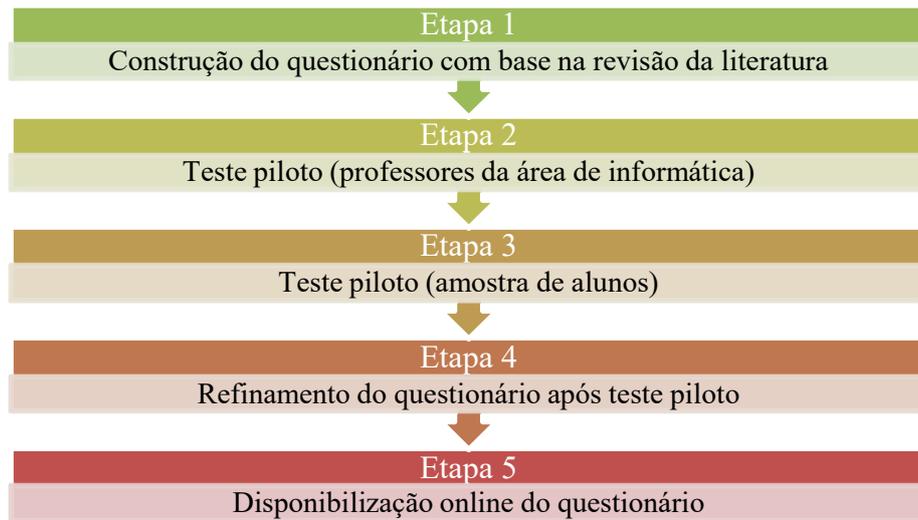


Ilustração 2 - Etapas da Concepção do Questionário de Investigação – Gamificação

Após a conclusão foram convidados a responder a um questionário. O questionário é composto por diversas questões e as respostas dos estudantes foram feitas numa escala de sete pontos (1- Discordo Fortemente, 7- Concordo Fortemente). O questionário inclui questões relacionadas com as características dos estudantes, e.g., género, curso e idade e foi disponibilizado *online* e acessível através de um endereço web. Os dados foram recolhidos entre o segundo semestre do ano letivo 2015/2016 e o primeiro semestre do ano letivo de 2016/2017.

6.4.1.1 Amostra

A amostra selecionada para o presente estudo de investigação foram os estudantes do ensino superior que já tinham frequentado pelo menos uma vez uma disciplina de programação. Responderam ao questionário, um total de cento e oito estudantes de uma instituição do ensino superior politécnico. Os estudantes do género masculino representam 72.2%%, tendo 80.5% idade compreendida entre 18 a 22 anos, 15.8% idade compreendida entre 23 a 30 anos, 0.9% com idade entre 30 e 40 anos e 2.8% com idades acima de 41 anos, cf. Tabela 23.

Tabela 23 - Variáveis Sociodemográficas - Estudo 4

Variáveis		N=108	%
Género	Masculino	78	72.2%
	Feminino	30	27.8%
Idade	< 18 anos	0	0%
	18 a 22 anos	87	80.5%
	23 a 30 anos	17	15.8%
	31 a 40 anos	1	0.9%
	mais de 41 anos	3	2.8%

Tabela 24 – Operacionalização das Dimensões

Dimensão	Cód.	Variáveis Observadas	Autor
Facilidade de Utilização	PEOU1	A utilização da interface do curso <i>online</i> não requer um grande esforço mental.	(Davis, 1989)
	PEOU2	A interação com o curso <i>online</i> é clara e compreensível.	
	PEOU3	É fácil através da interface, fazer o que pretendo fazer.	
	PEOU4	A interface do curso <i>online</i> é fácil de usar.	
Prazer na Utilização	ENJ1	A experiência de aprendizagem no curso é divertida.	(Davis et al., 1992)
	ENJ2	A experiência de aprendizagem no curso é agradável.	
	ENJ3	A experiência de aprendizagem no curso é empolgante.	
	ENJ4	A experiência de aprendizagem no curso é interessante.	
Percepção da Utilidade	PU1	O curso <i>online</i> facilita a minha aprendizagem dos conceitos fundamentais de programação.	(Davis, 1989)
	PU2	O curso <i>online</i> permite-me alcançar os conhecimentos necessários sobre os conceitos fundamentais de programação.	
	PU3	Frequentar o curso permite-me aperfeiçoar o meu conhecimento sobre os conceitos fundamentais de programação.	
	PU4	Torno-me mais eficaz a programar quando frequento o curso <i>online</i> .	
	PU5	O curso <i>online</i> é útil.	
Intenção de Utilizar	IU1	Planeio-o continuar a frequentar o curso <i>online</i> .	(Davis, 1989)
	IU2	Pretendo continuar a frequentar o curso <i>online</i> no futuro.	
	IU3	Tenciono falar sobre o curso com os meus amigos.	
Uso	U1	Durante o último mês quantas vezes acederam ao curso <i>online</i> .	(Davis et al., 1992)
	U2	Indique em média a maior duração de uma das sessões no curso <i>online</i> .	
Flow	FLO1	No curso <i>online</i> , sinto que existe a quantidade certa de desafio.	(Engeser & Rheinberg, 2008)
	FLO2	Os meus pensamentos/atividades surgem rapidamente e suavemente.	
	FLO3	Não dou pelo tempo a passar.	
	FLO4	Não tenho dificuldade em concentrar-me.	
	FLO5	A minha mente está completamente limpa.	
	FLO6	Fico totalmente absorvido naquilo que estou a fazer.	
	FLO7	Os pensamentos/ movimentos certos ocorrem de acordo com os mesmos.	
	FLO8	Sei o que tenho que fazer em cada etapa do curso.	
	FLO9	No curso <i>online</i> , tenho o controlo sobre tudo o que estou a fazer.	
	FLO10	Estou completamente perdido nos meus pensamentos.	
Personalidade	PER1	Extrovertida, entusiasta	(Gosling et al., 2003)
	PER2	Crítica, conflituosa	
	PER3	Confiável, autodisciplinada	
	PER4	Ansiosa, facilmente chateada	
	PER5	Aberta a novas experiências, criativa	
	PER6	Reservada, calada	
	PER7	Simpática, calorosa	
	PER8	Desorganizada, descuidada	
	PER9	Calma, emocionalmente estável	
	PER10	Convencional, não criativa	

6.4.2 Resultados

Através deste estudo pretendeu-se explorar as relações entre o CANOE, *flow*, facilidade de utilização, prazer na utilização, utilidade percebida, intenção de uso e uso. Para avaliar a validade das dimensões foi utilizado a modelação estrutural de equações (SEM) com o método parcial (PLS) (Henseler et al., 2014). PLS é mais adequado para modelos causais que são

teoricamente justificados (Wright, 1934). A confiabilidade e validade foram testadas de modo a examinar o modelo de medição, e posteriormente o modelo estrutural foi testado (Ringle, Wende, & Will, 2005).

6.4.3 Modelo de medida e avaliação

A confiabilidade composta para todas as dimensões está acima de 0,873, conforme Tabela 25. Também o coeficiente de Alpha de *Cronbach* assume uma confiabilidade de igualdade para todos os itens. Valores entre 0,70 e 0,90 são considerados satisfatórios (Bernstein & Nunnally, 1994; Hair, Ringle, & Sarstedt, 2011).

Tabela 25 - Resultados do Modelo de Medida

Variável Latente	Item	Loading	AVE	CC	Alpha Cronbach	Validade Discriminante?	
CANOE	Extroversão	PER1	0,740	0,555	0,862	0,799	Sim
	Conscienciosidade	PER3	0,798				
	Aberto a N. Experiências	PER5	0,731				
	Agradabilidade	PER7	0,773				
	Estabilidade Emocional	PER9	0,678				
Flow	FLO2	0,778	0,573	0,903	0,876	Sim	
	FLO4	0,786					
	FLO5	0,757					
	FLO6	0,762					
	FLO7	0,821					
	FLO8	0,680					
	FLO9	0,705					
Facilidade de Utilização	PEOU2	0,961	0,913	0,969	0,952	Sim	
	PEOU4	0,957					
Prazer na Utilização	ENJ1	0,743	0,848	0,957	0,940	Sim	
	ENJ2	0,762					
	ENJ3	0,769					
	ENJ4	0,645					
Percepção da Utilidade	PU1	0,706	0,913	0,951	0,935	Sim	
	PU2	0,765					
	PU3	0,697					
	PU4	0,843					
Intenção de Utilizar	BI1	0,895	0,827	0,935	0,895	Sim	
	BI2	0,872					
	BI3	0,959					
Uso	U2	1	1	1	1	Sim	

Nota: CC – Confiabilidade Composta

Estes resultados sugerem que todas as dimensões do modelo são confiáveis. Para medir o indicador de confiabilidade testámos o critério de que todos os *loadings* dos itens deveriam ser superiores a 0,70 e os resultados obtidos são superiores a 0,70, indicando assim que todos os indicadores são confiáveis (Henseler et al., 2014). Para analisar a validade convergente do

modelo foi calculado a variância média extraída (VME). As VME das dimensões devem ser superiores a 0,5 de modo a explicar mais de metade dos indicadores (Hair, Hult, Ringle, & Sarstedt, 2013). O primeiro compara os *loadings* dos itens com os *cross-loadings* (Anexo C) e o segundo segue o critério de (Fornell & Larcker, 1981). O modelo foi testado para a consistência interna, indicador de confiabilidade e validade discriminante. Estas avaliações permitiram avançar para a avaliação do modelo estrutural com recurso ao PLS.

6.4.4 Avaliação do modelo estrutural

A qualidade do modelo estrutural (Anexo E) foi avaliada da seguinte forma: primeiro foi calculado o *bootstrap*, utilizou-se 5000 subamostras para determinar a significância dos caminhos com o modelo estrutural, seguidamente foi aplicado o algoritmo PLS. As relações entre as variáveis latentes (Anexo D) foram suportadas pelos dados empíricos nos correspondentes caminhos de análise (β). Para determinar o tamanho do efeito (f^2), foram tidos em conta os seguintes limites: ($> 0,02$) com efeito pequeno; ($> 0,150$) com efeito médio; e ($> 0,35$) com efeito grande (Cohen, 1988).

Tabela 26 -Testes de Hipóteses – Resultados

Hip.	Var.Ind.	→	Var. Dep.	Resultados	Conclusão
H1	CANOE	→	Extroversão	Positivo e Estatisticamente significativo ($\beta=0,740$, $p < 0,01$)	Suportado
			Agradabilidade	Positivo e Estatisticamente significativo ($\beta=0,773$, $p < 0,01$)	
			Conscienciosidade	Positivo e Estatisticamente significativo ($\beta=0,798$, $p < 0,01$)	
			Estado Emocional	Positivo e Estatisticamente significativo ($\beta=0,678$, $p < 0,01$)	
			Aberto a Experiências	Positivo e Estatisticamente significativo ($\beta=0,731$, $p < 0,01$)	
H2a	CANOE	→	Prazer na Utilização (ENJ)	Positivo e Estatisticamente significativo ($\beta=0,152$, $p < 0,01$)	Suportado com efeito pequeno
H2b	CANOE	→	Facilidade de Utilização (PEOU)	Positivo e Estatisticamente significativo ($\beta=0,385$, $p < 0,01$)	Suportado com efeito medio
H2c	CANOE	→	Perceção de Utilidade (PU)	Positivo e Estatisticamente significativo ($\beta=0,293$, $p < 0,01$)	Suportado com efeito medio
H3a	Flow (FLO)	→	Prazer na Utilização (ENJ)	Positivo e Estatisticamente significativo ($\beta=0,332$, $p < 0,01$)	Suportado com efeito medio
H3b	Flow (FLO)	→	Uso (U)	Positivo e Estatisticamente significativo ($\beta=0,228$, $p < 0,01$)	Suportado com efeito medio
H4a	Facilidade de Utilização (PEOU)	→	Prazer na Utilização (ENJ)	Positivo e Estatisticamente significativo ($\beta=0,482$, $p < 0,01$)	Suportado com efeito grande

Hip.	Var. Ind.	→	Var. Dep.	Resultados	Conclusão
H4b	Facilidade de Utilização (PEOU)	→	Percepção de Utilidade (PU)	Positivo e Estatisticamente significativo ($\beta=0,589$, $p < 0,01$)	Suportado com efeito grande
H5	Prazer na Utilização (ENJ)	→	Intenção de Utilização (BI)	Positivo e Estatisticamente significativo ($\beta=0,419$, $p < 0,01$)	Suportado com efeito pequeno
H6	Percepção de Utilidade (PU)	→	Intenção de Utilização (BI)	Positivo e Estatisticamente significativo ($\beta=0,353$, $p < 0,01$)	Suportado com efeito pequeno
H7	Intenção de Utilização (BI)	→	Uso (U)	Positivo e Estatisticamente significativo ($\beta=0,185$, $p < 0,05$)	Suportado com efeito pequeno

Notas: * para $p < 0,10$; ** significante para $p < 0,05$; *** significante para $p < 0,01$

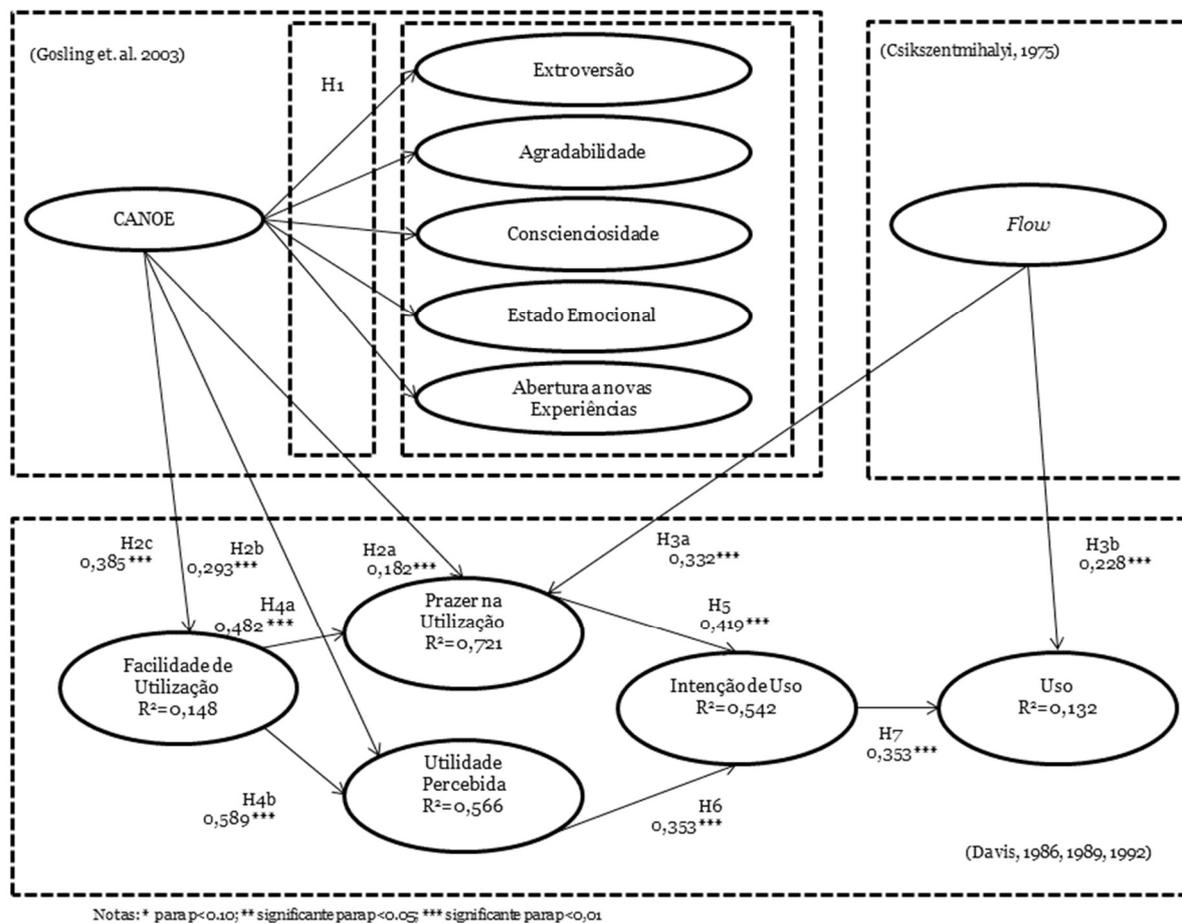


Figura 42 - Resultados do Modelo e Análise Estrutural

O CANOE explica significativamente a extroversão ($\beta=0,740$, $p < 0,01$), a agradabilidade ($\beta=0,773$, $p < 0,01$), a conscienciosidade ($\beta=0,798$, $p < 0,01$), estado emocional ($\beta=0,678$, $p < 0,01$), e aberto a novas experiências ($\beta=0,731$, $p < 0,01$), confirmando assim a H1. O modelo

estrutural explica 14,8% da variância na facilidade de utilização. O CANOE explica significativamente ($\beta=0,385$, $p<0,01$) suportando a H2b. O modelo estrutural explica 56,6% da variância na utilidade percebida. O CANOE explica significativamente ($\beta=0,293$, $p<0,01$) suportando a H2c. A facilidade de utilização explica significativamente ($\beta=0,589$, $p<0,01$) suportando H4b. O modelo estrutural explica 72,1% da variância no prazer de utilização. O CANOE explica significativamente ($\beta=0,152$, $p<0,01$) suportando a H2a. O *flow* explica significativamente ($\beta=0,332$, $p<0,01$) suportando a H3a. A facilidade de utilização explica significativamente ($\beta=0,482$, $p<0,01$) suportando a H4a. O modelo estrutural explica 54,2% da variância na intenção de utilizar. O prazer de utilização explica significativamente ($\beta=0,419$, $p<0,01$) suportando a H5. A percepção de utilidade explica significativamente ($\beta=0,352$, $p<0,01$), suportando a H6. O modelo estrutural explica 13,2% da variância no uso. O *flow* explica significativamente o uso ($\beta=0,228$, $p<0,01$), suportando a H3b. A intenção de utilização explica significativamente o uso ($\beta=0,185$, $p<0,05$) suportando a H7. A qualidade do modelo estrutural é baseada nas correlações múltiplas ao quadrado (R^2). Após a validação das medidas e confirmada a qualidade do modelo estrutural, os resultados indicam que o modelo é válido. A análise de caminhos foi analisada para testar as hipóteses de investigação.

6.4.5 Discussão de resultados

Como demonstra a Tabela 26, elaborada a partir da análise aos dados, todas as hipóteses foram suportadas. As hipóteses H2a, H2b e H2c foram suportadas pelos resultados empíricos ($p<0,01$). O CANOE tem um impacto positivo sobre o prazer na utilização, utilidade percebida e na percepção da facilidade de utilização. A hipótese H3a foi confirmada. O *flow* tem um impacto positivo no prazer na utilização ($p<0,01$). Os resultados indicam que quanto mais os estudantes se sentem no estado de *flow* mais prazer na utilização experienciam. Estes resultados têm similaridades com estudos anteriores que reportam um impacto positivo do *flow* no prazer de utilização (Voiskounsky, Mitina, & Avetisova, 2004; Weibel, Wissmath, Habegger, Steiner, & Groner, 2008). A hipótese H3b foi suportada ($p<0,01$). O *flow* tem um impacto positivo no uso. Estes resultados são corroborados por Koufaris (Koufaris, 2002). A hipótese H4a foi suportada nos resultados empíricos ($p<0,01$) e revelam que a facilidade de utilização tem um impacto positivo no prazer de utilização. Os estudantes percebem a utilização do curso *online* gamificado como fácil de usar e quanto maior for essa percepção maior será a percepção do prazer na utilização. Estudos anteriores suportam estes resultados (Davis et al., 1992). A hipótese H4b foi suportada. A análise aos resultados revelou que a facilidade de utilização tem um efeito positivo na percepção de utilidade do curso *online* gamificado. Estes resultados são

consistentes com estudos anteriores (Davis, 1989; Özbek et al., 2014). Quando a tecnologia é percebida como mais fácil de utilizar, é também considerada mais útil (Davis, 1989; Venkatesh & Davis, 2000) e mais agradável (Van der Heijden, 2004). Estes resultados sugerem que os estudantes percebem a facilidade de utilização do curso *online* gamificado e consequentemente percebem com utilidade para a sua aprendizagem. A hipótese H5 foi suportada. Os resultados empíricos ($p < 0,01$) revelam que o prazer na utilização tem um impacto positivo na intenção de uso. Estes resultados têm suporte nos estudos anteriores (Lou, Chau, & Li, 2005; Venkatesh et al., 2002; Yi & Hwang, 2003). A hipótese H6 foi suportada. A análise aos resultados revelou que a utilidade percebida tem impacto positivo na intenção de uso. A intenção positiva do aluno utilizar o curso *online* gamificado resulta da facilidade de utilização, da fácil interação com os conteúdos e atividades e do prazer experienciado enquanto o utiliza. Estes resultados são corroborados pelo autor (Davis, 1989). Por último H7 foi suportada. Os resultados empíricos ($p < 0,05$) demonstram que a intenção de uso tem impacto no uso. Estes resultados são consistentes com estudos realizados por (Moon & Kim, 2001; Venkatesh & Davis, 2000). Este estudo de investigação demonstra que o CANOE, facilidade de utilização e *flow* são determinantes no prazer na utilização. O CANOE é determinante na facilidade de utilização do curso *online* gamificado. O CANOE e a facilidade de utilização são determinantes na utilidade percebida. O prazer na utilização e a utilidade percebida são fatores determinantes na intenção de utilização. Por último, a intenção de utilização e o *flow* são determinantes que contribuem para o uso do curso *online* gamificado.

6.4.6 Conclusões

Este estudo propõe um modelo explicativo do impacto das características individuais do indivíduo (CANOE) e do *flow* na adoção de um curso *online* gamificado para aprendizagem da programação. O modelo de investigação foi testado empiricamente com 108 alunos e foi validado. Da análise ao modelo estrutural podemos concluir que o prazer de utilização é explicado em 72% pela variação do CANOE, fluxo e facilidade de utilização. A facilidade de utilização é explicada em 14,8% pelo CANOE. A utilidade percebida é explicada em 56,6% pela facilidade de utilização e pelo CANOE. A intenção de utilização é explicada em 54,2% pelo prazer na utilização e pela utilidade percebida. O uso é explicado em 13,2% pelo *fluxo* e pela intenção de uso.

7 Conclusões, limitações, contribuições e trabalhos futuros

O trabalho desenvolvido nesta tese centra-se nos problemas da aprendizagem da programação e na utilização da gamificação, como forma de motivar e envolver os estudantes na aprendizagem. Tem duas questões de investigação associadas, estando a primeira relacionada com a identificação dos problemas na aprendizagem da programação e a segunda, com a verificação do impacto da utilização da gamificação em cursos *online*. Estas duas questões de investigação conduziram-nos à elaboração de quatro estudos de investigação (cf. Figura 1). Todos os objetivos definidos no início da tese foram atingidos, assim como foram respondidas todas as questões de investigação (Tabela 27). Identificaram-se as dificuldades na aprendizagem da programação, tendo sido posteriormente proposta e validada, uma *framework* teórica para implementar a gamificação nos cursos *online* de aprendizagem da programação. Desenvolveu-se um novo modelo concetual para explicar quais os fatores determinantes na adoção da gamificação e em que medida a personalidade e o *flow* (novas dimensões), influenciam a adoção da gamificação.

Tabela 27 - Objetivos de Investigação vs Estudos de Investigação

Questões de Investigação	Objetivos de Investigação	Estudos
Q1	OBJ-1: Identificar as dificuldades percecionadas na aprendizagem da programação	Estudo 1 Estudo 2
	OBJ-2: Identificar as soluções facilitadoras na aprendizagem de programação	Revisão da literatura
Q2	OBJ-3: Propor <i>framework</i> concetual gamificada para cursos <i>online</i> de programação	Estudo 3
	OBJ-4: Implementar a <i>framework</i> concetual	
	OBJ-5: Propor modelo de adoção de curso <i>online</i> de programação gamificado	Estudo 4

7.1 Conclusões por objetivos de investigação

O primeiro objetivo (**OBJ-1**) estabelecido para esta tese, foi a identificação das dificuldades percecionadas na aprendizagem da programação. Este objetivo foi concretizado através da realização de dois estudos científicos. De modo a efetuar a identificação das dificuldades

percecionadas, foi elaborado um questionário baseado na revisão da literatura efetuada, que foi distribuído por uma amostra de estudantes e professores do ensino superior. Após a distribuição do questionário, foram recolhidos os dados e efetuada a respetiva análise. Os dados foram classificados por situações de aprendizagem, conceitos de aprendizagem, contextos de aprendizagem e materiais/recursos, sendo os dois últimos, utilizados para recolher a perceção de utilidade. Após a análise efetuada, verificou-se que globalmente, os estudantes apresentam um grau de perceção de dificuldade baixa relativamente às situações de aprendizagem e aos conceitos de programação, evidenciando um excesso de confiança. Verificou-se que os conceitos de programação e as situações de aprendizagem, que requerem do estudante um maior raciocínio abstrato, são os que eles percecionam como sendo os mais difíceis. Os conceitos identificados como sendo os mais difíceis, foram: ponteiros e referências, tipos abstratos de dados e tipos estruturados de dados. Verificou-se também que as situações de aprendizagem que necessitam de um maior raciocínio abstrato ou onde são necessárias, uma compreensão e manipulação de várias entidades, são vistas como sendo de difícil aprendizagem. Falamos das situações que envolvem o desenhar um programa para a resolução de uma determinada tarefa e dividir o programa em funcionalidades. Quando comparámos as perceções dos estudantes com as perceções dos professores, constatou-se que os professores são mais realistas nas dificuldades dos estudantes. Esta observação é confirmada e validada, quando comparamos as perceções dos estudantes e dos professores, com as provas de avaliação. As perceções dos professores encontram-se mais próximas dos resultados das provas de avaliação.

O segundo objetivo (**OBJ-2**) estabelecido nesta tese, foi a identificação das soluções facilitadoras na aprendizagem da programação. Este objetivo foi alcançado através de uma sólida e aprofundada revisão da literatura. Após a revisão da literatura, identificaram-se diversas soluções que foram propostas com o objetivo de mitigar os problemas na aprendizagem da programação. Por exemplo, foram identificadas soluções baseadas em ferramentas de visualização gráfica (e.g. Alice, Scraeth, Jeroo), soluções baseadas na utilização de artefactos tangíveis (e.g. robótica educacional), soluções que utilizam jogos educacionais digitais e soluções que recorrem à utilização de elementos característicos de jogos (gamificação). Após a identificação das soluções e tendo por base os problemas na aprendizagem da programação identificados no OBJ-1, procedeu-se à sua associação no sentido de relacionar as que soluções poderiam contribuir para minimizar esses mesmos problemas.

O terceiro objetivo estabelecido (**OBJ-3**), foi a proposta de uma *framework* teórica para implementar a gamificação em cursos de programação *online*. Este objetivo foi atingido. Para

o concretizar, numa primeira fase foi tida em consideração a revisão da literatura relacionada com os problemas na aprendizagem da programação e as respetivas soluções facilitadoras, efetuadas e concretizadas nos OBJ-1 e OBJ-2. Posteriormente, revimos na literatura aspetos relacionados com a gamificação, características individuais dos indivíduos e estados de imersão. Revimos igualmente *frameworks* relacionadas com a implementação da gamificação em cursos *online* de programação online. Após todo o processo de revisão da literatura relacionada, foram identificadas as dimensões que compõem a *framework* proposta: público-alvo; objetivos gerais; resultados de aprendizagem; tópicos; conteúdo; gamificação; absorção cognitiva; *flow*; e personalidade.

O quarto objetivo (**OBJ-4**) proposto nesta tese foi a implementação da *framework* e a respetiva validação. Este objetivo foi concretizado. De modo a cumprir com o objetivo estabelecido, foi desenhado e implementado um curso *online* de fundamentos de programação. O curso foi implementado na plataforma de aprendizagem *Moodle* e disponibilizado aos estudantes de uma instituição do ensino superior politécnico. O curso não fazia parte do currículo oficial, tendo os estudantes frequentado o curso de forma voluntária. Após frequência e conclusão do curso, os estudantes responderam a um questionário *online*, onde foram identificadas as suas atitudes face aos seguintes elementos de gamificação: medalhas, pontos, quadro de honra e barra de progresso. Globalmente, verificou-se que os estudantes têm uma atitude positiva perante os elementos que integram o curso *online* de programação.

Por último, o quinto objetivo (**OBJ-5**) estabelecido, foi a proposta de um modelo explicativo do impacto das características individuais do indivíduo (CANOE) e do *flow*, na adoção de um curso *online* gamificado para aprendizagem da programação. Este objetivo foi concretizado. Para a concretização do objetivo, a teoria de adoção da tecnologia e os modelos teóricos de adoção da tecnologia, foram revistos na literatura. Identificou-se e adotou-se o modelo TAM com as dimensões facilidade de utilização, prazer na utilização, utilidade percebida, intenção de uso e uso. Foram ainda adicionadas duas novas dimensões ao modelo: *CANOE* e *flow*. O modelo de investigação foi testado empiricamente com os estudantes que frequentaram o curso *online*, tendo sido posteriormente validado. Os resultados obtidos demonstraram que o CANOE, a facilidade de utilização e o *flow*, são determinantes no prazer da utilização. O CANOE é determinante na facilidade de utilização do curso *online* gamificado. O CANOE e a facilidade de utilização são determinantes na utilidade percebida. O prazer na utilização e a utilidade percebida são fatores determinantes na intenção de utilização. Por último, a intenção de utilização e o *flow*, são determinantes pois contribuem para o uso do curso *online* gamificado.

7.2 Limitações

A presente tese aborda aspetos relacionados com a didática da programação. Estudar a didática na educação, requer análises temporais mais alargadas, contudo, esses horizontes temporais ultrapassam o tempo de execução de uma tese, sendo assim um fator de limitação ao trabalho de investigação realizado.

7.3 Implicações Teóricas e Práticas

Como implicações teóricas, esta tese aprofunda o conhecimento científico relacionado com os problemas e soluções na aprendizagem da programação. Com a massificação dos cursos *online* e o interesse crescente da gamificação no ensino, a framework proposta, permitirá guiar os educadores na implementação de cursos online de programação gamificados. A framework é inovadora, dado que não existe atualmente nenhuma framework especificamente para cursos *online* de programação, sendo extensível a outros contextos de utilização. Expandiu-se também o modelo TAM com a inclusão de duas novas dimensões e propôs-se um modelo explicativo de adoção da gamificação em cursos *online* gamificados de programação. O modelo inclui quais as características dos estudantes e o *flow* que influenciam a adoção de cursos *online* gamificados.

Como implicações práticas, o trabalho desenvolvido nesta tese é dirigido ao sector da educação para facilitar e incentivar a adoção da gamificação nos cursos *online* de programação. Os ambientes de ensino gamificados contribuem para o envolvimento e motivação. No desenho de perfis de utilizador devemos ter em consideração as características não cognitivas do utilizador. Contribui também, para orientar o desenho e a implementação da gamificação em cursos *online* destinados à formação no sector da indústria.

7.4 Trabalhos Futuros

A partir do trabalho realizado nesta tese, outros caminhos podem ser traçados. Desta forma, sugerem-se como trabalhos futuros, estudos que avaliem o impacto do *CANOE* e do *flow* no desempenho individual do estudante. A comparação da influência do *CANOE* e do *flow* na adoção de cursos *online* gamificados em diferentes contextos de utilização, será também de grande utilidade. A realização de estudos com uma maior duração temporal que verifiquem a utilização da gamificação na aprendizagem, serão úteis para a compreensão do impacto e para

a aceitação da gamificação, na aprendizagem da programação. Por último trabalhos que verifiquem em diferentes contextos.

REFERÊNCIAS

- Agarwal, R., & Karahanna, E. (2000). Time Flies When You're Having Fun: Cognitive Absorption and Beliefs about Information Technology Usage. *MIS Quarterly*, 24(4), 665–694. <https://doi.org/10.2307/3250951>
- Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, 50(2), 179–211. [https://doi.org/10.1016/0749-5978\(91\)90020-T](https://doi.org/10.1016/0749-5978(91)90020-T)
- Ajzen, I., & Fishbein, M. (1972). Attitudes and normative beliefs as factors influencing behavioral intentions. *Journal of Personality and Social Psychology*, pp. 1–9.
- Ala-Mutka, K. M. (2005). A Survey of Automated Assessment Approaches for Programming Assignments. *Computer Science Education*, 15(2), 83–102. <https://doi.org/10.1080/08993400500150747>
- Aldemir, T., Celik, B., & Kaplan, G. (2018). A qualitative investigation of student perceptions of game elements in a gamified course. *Computers in Human Behavior*, 78(Supplement C), 235–254. <https://doi.org/10.1016/j.chb.2017.10.001>
- Álvarez, A., & Larrañaga, M. (2013). Using LEGO mindstorms to engage students on algorithm design. Em *2013 IEEE Frontiers in Education Conference (FIE)* (pp. 1346–1351). <https://doi.org/10.1109/FIE.2013.6685052>
- Aparicio, M., & Bacao, F. (2013). E-learning concept trends. Em *Proceedings of the 2013 International Conference on Information Systems and Design of Communication* (pp. 81–86). ACM.
- Aparicio, M., Bacao, F., & Oliveira, T. (2014). MOOC's business models: Turning black swans into gray swans. Em *Proceedings of the International Conference on Information Systems and Design of Communication* (pp. 45–49). ACM.
- Aparicio, M., Bacao, F., & Oliveira, T. (2016). An e-Learning Theoretical Framework. *Journal of Educational Technology & Society*, 19(1), 292–307.
- Aparicio, M., Bacao, F., & Oliveira, T. (2017). Grit in the path to e-learning success. *Computers in Human Behavior*, 66, 388–399. <https://doi.org/10.1016/j.chb.2016.10.009>
- Arlegui, J., Menegatti, E., Moro, M., & Pina, A. (2008). Robotics, Computer Science Curricula and Interdisciplinary Activities. Em *Workshop Teaching with robotics, Conference SIMPAR* (pp. 10–21). Venice.

- Attali, Y., & Arieli-Attali, M. (2015). Gamification in assessment: Do points affect test performance? *Computers & Education*, 83(Supplement C), 57–63. <https://doi.org/10.1016/j.compedu.2014.12.012>
- Baker, P. M. A., Bujak, K. R., & DeMillo, R. (2012). The Evolving University: Disruptive Change and Institutional Innovation. *Procedia Computer Science*, 14(Supplement C), 330–335. <https://doi.org/10.1016/j.procs.2012.10.037>
- Barnes, T., Powell, E., Chaffin, A., & Lipford, H. (2008). Game2Learn: Improving the Motivation of CS1 Students. Em *Proceedings of the 3rd International Conference on Game Development in Computer Science Education* (pp. 1–5). New York, NY, USA: ACM. <https://doi.org/10.1145/1463673.1463674>
- Bergin, S., & Reilly, R. (2005). The influence of motivation and confort-level on learning to program. (pp. 293–304). Apresentado na In Proceedings of the 17th Annual Workshop on the Psychology of Programming Interest Group, University of Sussex, Brighton UK.
- Berglund, A., & Lister, R. (2010). Introductory programming and the didactic triangle. Em *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103* (pp. 35–44). Darlinghurst, Australia, Australia: Australian Computer Society, Inc. Obtido de <http://dl.acm.org/citation.cfm?id=1862219.1862227>
- Bernhard, W., Bittel, N., Van Der Vlies, S., Bettoni, M., & Roth, N. (2013). The MOOCs Business Model. *Procedia - Social and Behavioral Sciences*, 106(Supplement C), 2931–2937. <https://doi.org/10.1016/j.sbspro.2013.12.339>
- Bernstein, I. H., & Nunnally, J. (1994). Psychometric theory. *New York: McGraw-Hill. Oliva, TA, Oliver, RL, & MacMillan, IC (1992). A catastrophe model for developing service satisfaction strategies. Journal of Marketing*, 56, 83–95.
- Blank, D., Kumar, D., Meeden, L., & Yanco, H. (2003). Pyro: A Python-based Versatile Programming Environment for Teaching Robotics. *J. Educ. Resour. Comput. (JERIC)*, 4(3), 3. <https://doi.org/10.1145/1083310.1047569>
- Blohm, I., & Marco, J. (2013). Gamification - Design of IT-Based Enhancing Services for Motivational Support and Behavioral Change. *Business & Information Systems Engineering*, pp. 275–278.
- Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., Krathwohl, D. R., & others. (1956). Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain. New York: David McKay Company. *Inc. (7th Edition 1972)*.
- Bontempo, K., Napier, J., Hayes, L., & Brashear, V. (2014). Does personality matter? An international study of sign language interpreter disposition. *Translation and*

Interpreting: the International Journal of Translation and Interpreting Research, 6(1), 23–46.

- Bouvier, D., Lovellette, E., Matta, J., Alshaigy, B., Becker, B. A., Craig, M., ... Zarb, M. (2016). Novice Programmers and the Problem Description Effect. Em *Proceedings of the 2016 ITiCSE Working Group Reports* (pp. 103–118). New York, NY, USA: ACM. <https://doi.org/10.1145/3024906.3024912>
- Brought, G. (2012). dLife: A Java Library for Multiplatform Robotics, AI and Vision in Undergraduate CS and Research. Em *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (pp. 33–38). New York, NY, USA: ACM. <https://doi.org/10.1145/2157136.2157151>
- Brauner, P., Leonhardt, T., Ziefle, M., & Schroeder, U. (2010). The Effect of Tangible Artifacts, Gender and Subjective Technical Competence on Teaching Programming to Seventh Graders. Em *Teaching Fundamentals Concepts of Informatics* (pp. 61–71). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-11376-5_7
- Brazil, A. L., Conci, A., Clua, E., Bittencourt, L. K., Baruque, L. B., & Silva Conci, N. da. (2018). Haptic forces and gamification on epidural anesthesia skill gain. *Entertainment Computing*, 25(Supplement C), 1–13. <https://doi.org/10.1016/j.entcom.2017.10.002>
- Buckley, P., & Doyle, E. (2016). Gamification and student motivation. *Interactive Learning Environments*, 24(6), 1162–1175. <https://doi.org/10.1080/10494820.2014.964263>
- Buckley, P., & Doyle, E. (2017). Individualising gamification: An investigation of the impact of learning styles and personality traits on the efficacy of gamification using a prediction market. *Computers & Education*, 106(Supplement C), 43–55. <https://doi.org/10.1016/j.compedu.2016.11.009>
- Burke, B. (2016). *Gamify: How gamification motivates people to do extraordinary things*. Routledge.
- Byrne, P., & Lyons, G. (2001). The Effect of Student Attributes on Success in Programming. Em *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education* (pp. 49–52). New York, NY, USA: ACM. <https://doi.org/10.1145/377435.377467>
- Carlisle, M. C., Wilson, T. A., Humphries, J. W., & Hadfield, S. M. (2004). RAPTOR: Introducing Programming to Non-majors with Flowcharts. *J. Comput. Sci. Coll.*, 19(4), 52–60.
- Carrol, J. M., & Swatman, P. A. (2000). A Methodological framework for building theory in information systems research. *European Journal of Information Systems*, pp. 235–242.

- Chaffin, A., Doran, K., Hicks, D., & Barnes, T. (2009). Experimental Evaluation of Teaching Recursion in a Video Game. Em *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games* (pp. 79–86). New York, NY, USA: ACM. <https://doi.org/10.1145/1581073.1581086>
- Chamorro-Premuzic, T., & Furnham, A. (2009). Mainly Openness: The relationship between the Big Five personality traits and learning approaches. *Learning and Individual Differences, 19*(4), 524–529. <https://doi.org/10.1016/j.lindif.2009.06.004>
- Chamorro-Premuzic, T., Furnham, A., & Lewis, M. (2007). Personality and approaches to learning predict preference for different teaching methods. *Learning and Individual Differences, 17*(3), 241–250. <https://doi.org/10.1016/j.lindif.2006.12.001>
- Chang, C.-C., Liang, C., Chou, P.-N., & Lin, G.-Y. (2017). Is game-based learning better in flow experience and various types of cognitive load than non-game-based learning? Perspective from multimedia and media richness. *Computers in Human Behavior, 71*(Supplement C), 218–227. <https://doi.org/10.1016/j.chb.2017.01.031>
- Chao, P.-Y. (2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education, 95*(Supplement C), 202–215. <https://doi.org/10.1016/j.compedu.2016.01.010>
- Cheong, C., Filippou, J., & Cheong, F. (2014). Towards the gamification of learning: Investigating student perceptions of game elements. *Journal of Information Systems Education, 25*(3), 233.
- Cheung, L. Y. (2016). *On the influences of personality traits on employees engagement with gamified enterprise tools* (Master Thesis). Delft University of Technology, Netherlands. Obtido de <http://repository.tudelft.nl/islandora/object/uuid:9203ba22-72e0-4da5-bb50-f9e88339cc9a/datastream/OBJ/download>
- Childers, T. L., Carr, C. L., Peck, J., & Carson, S. (2001). Hedonic and utilitarian motivations for online retail shopping behavior. *Journal of Retailing, 77*(4), 511–535. [https://doi.org/10.1016/S0022-4359\(01\)00056-2](https://doi.org/10.1016/S0022-4359(01)00056-2)
- Chou, Y.-K. (2015, Março 1). Octalysis: Complete Gamification Framework - Yu-kai Chou. Obtido 1 de Novembro de 2017, de <http://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/>
- Codish, D., & Ravid, G. (2014). Personality Based Gamification: How different personalities perceive gamification. *ECIS 2014 Proceedings*. Obtido de <http://aisel.aisnet.org/ecis2014/proceedings/track12/10>
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* Lawrence Erlbaum Associates. Hillsdale, NJ, 20–26.

- Cooper, S., Dann, W., & Pausch, R. (2003). Teaching Objects-first in Introductory Computer Science. Em *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education* (pp. 191–195). New York, NY, USA: ACM.
<https://doi.org/10.1145/611892.611966>
- Cooper, S., Dann, W., & Paush, R. (2000). Alice: a 3-D tool for introductory programming concepts. *Journal of Computing in Small Colleges*, pp. 108–117.
- Costa, C. J., Aparicio, M., Aparicio, S., & Aparicio, J. T. (2017). Gamification Usage Ecology. Em *Proceedings of the 35th ACM International Conference on the Design of Communication* (p. 2:1–2:9). New York, NY, USA: ACM.
<https://doi.org/10.1145/3121113.3121205>
- Costa, C. J., Aparicio, M., & Cordeiro, C. (2012). A Solution to Support Student Learning of Programming. Em *Proceedings of the Workshop on Open Source and Design of Communication* (pp. 25–29). New York, NY, USA: ACM.
<https://doi.org/10.1145/2316936.2316942>
- Costa, C. J., Ferreira, E., Bento, F., & Aparicio, M. (2016). Enterprise resource planning adoption and satisfaction determinants. *Computers in Human Behavior*, 63, 659–671.
<https://doi.org/10.1016/j.chb.2016.05.090>
- Costa, C. J., Silva, J., & Aparicio, M. (2007). Evaluating Web Usability Using Small Display Devices. Em *Proceedings of the 25th Annual ACM International Conference on Design of Communication* (pp. 263–268). New York, NY, USA: ACM.
<https://doi.org/10.1145/1297144.1297202>
- Costa, R. L. P. de A. (2012). Ludificação: o futuro do trabalho será um jogo?: reflexão sobre o trabalho no século XXI a partir de casos de plataformas de recrutamento e seleção de jovens. (Master Thesis). *Universidade Católica Portuguesa*, Lisboa. Obtido de <http://hdl.handle.net/10400.14/17837>
- Csikszentmihalyi, M. (1975). *Beyond Boredom and Anxiety*. Jossey-Bass Publishers.
- Dale, N. B. (2006). Most Difficult Topics in CS1: Results of an Online Survey of Educators. *SIGCSE Bull.*, 38(2), 49–53. <https://doi.org/10.1145/1138403.1138432>
- Daly, T. (2009). Using introductory programming tools to teach programming concepts: A literature review. *Journal for Computing Teachers*, pp. 1–6.
- Dann, W., Cosgrove, D., Slater, D., Culyba, D., & Cooper, S. (2012). Mediated Transfer: Alice 3 to Java. Em *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (pp. 141–146). New York, NY, USA: ACM.
<https://doi.org/10.1145/2157136.2157180>
- Davis, F. (1986). *A technology acceptance model for empirically testing new end-user information systems: theory and results*. Massachusetts Institute of Technology.

- Davis, F. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3), 319–340.
<https://doi.org/10.2307/249008>
- Davis, F., Bagozzi, R. P., & Warshaw, P. R. (1992). Extrinsic and intrinsic motivation to use computers in the workplace. *Journal of Applied Social Psychology*, 22(14), 1111–1132. <https://doi.org/10.1111/j.1559-1816.1992.tb00945.x>
- De Raad, B., & Schouwenburg, H. (1996). Personality traits in learning and education. *European Journal of Personality*, 10(3), 185–200.
- De Schutter, B., & Abeele, V. (2014). Gradequest - Evaluating the impact of using games design techniques in an undergraduate course. Apresentado na 9th International Conference on the Foundations of Digital Games.
- Denning, P. J. (2004). The Field of Programmers Myth. *Commun. ACM*, 47(7), 15–20.
<https://doi.org/10.1145/1005817.1005836>
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining «Gamification». Em *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (pp. 9–15). New York, NY, USA: ACM. <https://doi.org/10.1145/2181037.2181040>
- Dichev, C., & Dicheva, D. (2017). Gamifying education: what is known, what is believed and what remains uncertain: a critical review. *International Journal of Educational Technology in Higher Education*, 14(1), 9. <https://doi.org/10.1186/s41239-017-0042-5>
- Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in Education: A Systematic Mapping Study. *Educational Technology & Society*, pp. 75–88.
- Domínguez, A., Saenz-de-Navarrete, J., de-Marcos, L., Fernández-Sanz, L., Pagés, C., & Martínez-Herráiz, J.-J. (2013). Gamifying learning experiences: Practical implications and outcomes. *Computers & Education*, 63, 380–392.
<https://doi.org/10.1016/j.compedu.2012.12.020>
- Duff, A., Boyle, E., Dunleavy, K., & Ferguson, J. (2004). The relationship between personality, approach to learning and academic performance. *Personality and Individual Differences*, 36(8), 1907–1920. <https://doi.org/10.1016/j.paid.2003.08.020>
- Eagle, M., & Barnes, T. (2009). Experimental Evaluation of an Educational Game for Improved Learning in Introductory Computing. Em *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (pp. 321–325). New York, NY, USA: ACM. <https://doi.org/10.1145/1508865.1508980>
- Eckerdal, A. (2009). *Novice Programming Students' Learning of Concepts and Practice* (PhD Dissertation). Acta Universitatis Upsaliensis Uppsala. Obtido de <http://user.it.uu.se/~annae/FullAvh-Spikenheten.pdf>

- Eckerdal, A., & Thuné, M. (2005). Novice Java Programmers' Conceptions of «Object» and «Class», and Variation Theory. Em *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (pp. 89–93). New York, NY, USA: ACM. <https://doi.org/10.1145/1067445.1067473>
- Engeser, S., & Rheinberg, F. (2008). Flow, performance and moderators of challenge-skill balance. *Motivation and Emotion*, 32(3), 158–172. <https://doi.org/10.1007/s11031-008-9102-4>
- EU. (2016). Digital Skills & Jobs. Obtido 11 de Setembro de 2017, de <https://ec.europa.eu/digital-single-market/en/policies/digital-skills>
- Ferreira, F., Costa, C. J., Aparicio, M., & Aparicio, S. (2017). Learning programming: A continuance model. Em *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–6). <https://doi.org/10.23919/CISTI.2017.7975815>
- Festinger, L. (1954). A Theory of Social Comparison Processes. *Human Relations*, 7(2), 117–140. <https://doi.org/10.1177/001872675400700202>
- Fornell, C., & Larcker, D. F. (1981). Structural Equation Models with Unobservable Variables and Measurement Error: Algebra and Statistics. *Journal of Marketing Research*, 18(3), 382–388. <https://doi.org/10.2307/3150980>
- Fotaris, P., Mastoras, T., Leinfellner, R., & Rosunally, Y. (2016). Climbing up the Leaderboard: An Empirical Study of Applying Gamification Techniques to a Computer Programming Class. *Electronic Journal of E-Learning*, 14(2), 94–110.
- Frangou, S., Papanikolaou, K., & Aravecchia, L. (2008). Representative Examples of Implementing Educational Robotics in School Based on The Constructivist Approach. Em *Workshop Proceedings of SIMPAR* (pp. 54–65). Venice.
- Franklin, D. (2015). Putting the Computer Science in Computing Education Research. *Commun. ACM*, 58(2), 34–36. <https://doi.org/10.1145/2700376>
- Fryer, L. K., Nicholas Bovee, H., & Nakao, K. (2014). E-learning: Reasons students in language learning courses don't want to. *Computers & Education*, 74, 26–36. <https://doi.org/10.1016/j.compedu.2014.01.008>
- García, F., Pedreira, O., Piattini, M., Cerdeira-Pena, A., & Penabad, M. (2017). A framework for gamification in software engineering. *Journal of Systems and Software*, 132, 21–40. <https://doi.org/10.1016/j.jss.2017.06.021>
- Gardiner, A., Aasheim, C., Rutner, P., & Williams, S. (2017). Skill Requirements in Big Data: A Content Analysis of Job Advertisements. *Journal of Computer Information Systems*, 0(0), 1–11. <https://doi.org/10.1080/08874417.2017.1289354>

- Giannetto, D., Chao, J., & Fontana, A. (2013). Gamification in a social learning environment. Em *Proceedings of the Informing Science and Information Technology Education Conference* (pp. 195–207). Informing Science Institute.
- Goldberg, L. R. (1990). An alternative ‘description of personality’: The big-five factor structure. *Journal of Personality and Social Psychology*, pp. 1216–1229.
- Gomes, A., & Mendes, A. (2007). Learning to program-difficulties and solutions. Em *International Conference on Engineering Education–ICEE* (Vol. 2007). Coimbra, Portugal.
- Goodhue, D. L., & Thompson, R. L. (1995). Task-Technology Fit and Individual Performance. *MIS Quarterly*, 19(2), 213–236. <https://doi.org/10.2307/249689>
- Gosling, S. D., Rentfrow, P. J., & Swann Jr., W. B. (2003). A very brief measure of the Big-Five personality domains. *Journal of Research in Personality*, 37(6), 504–528. [https://doi.org/10.1016/S0092-6566\(03\)00046-1](https://doi.org/10.1016/S0092-6566(03)00046-1)
- Hair, J. F., Ringle, C. M., & Sarstedt, M. (2011). PLS-SEM: Indeed a Silver Bullet. *Journal of Marketing Theory and Practice*, 19(2), 139–152. <https://doi.org/10.2753/MTP1069-6679190202>
- Hair, J., Hult, G., Ringle, C., & Sarstedt, M. (2013). *A primer on partial least squares structural equation modeling (PLS-SEM)*. Sage Publications.
- Hamari, J. (2017). Do badges increase user activity? A field experiment on the effects of gamification. *Computers in Human Behavior*, 71(Supplement C), 469–478. <https://doi.org/10.1016/j.chb.2015.03.036>
- Hamari, J., & Koivisto, J. (2014). Measuring flow in gamification: Dispositional Flow Scale-2. *Computers in Human Behavior*, 40(Supplement C), 133–143. <https://doi.org/10.1016/j.chb.2014.07.048>
- Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does Gamification Work? – A Literature Review of Empirical Studies on Gamification. Em *2014 47th Hawaii International Conference on System Sciences* (pp. 3025–3034). <https://doi.org/10.1109/HICSS.2014.377>
- Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computers & Education*, 54(4), 1127–1136. <https://doi.org/10.1016/j.compedu.2009.10.020>
- Helminen, J. (2009, Abril). *An Education-Oriented Integrated Program Visualization, Visual Debugging, and Programming Exercise Tool for Python*. (Master Thesis). Helsinki University of Technology.
- Helminen, J., & Malmi, L. (2010). Jype - a Program Visualization and Programming Exercise Tool for Python. Em *Proceedings of the 5th International Symposium on Software*

- Visualization* (pp. 153–162). New York, NY, USA: ACM.
<https://doi.org/10.1145/1879211.1879234>
- Henriksen, P., & Kölling, M. (2004). Greenfoot: Combining Object Visualisation with Interaction. Em *Companion to the 19th Annual ACM SIGPLAN Conference on Object-oriented Programming Systems, Languages, and Applications* (pp. 73–82). New York, NY, USA: ACM. <https://doi.org/10.1145/1028664.1028701>
- Henseler, J., Dijkstra, T. K., Sarstedt, M., Ringle, C. M., Diamantopoulos, A., Straub, D. W., ... Calantone, R. J. (2014). Common Beliefs and Reality About PLS: Comments on Rönkkö and Evermann (2013). *Organizational Research Methods*, 17(2), 182–209. <https://doi.org/10.1177/1094428114526928>
- Hevner, A., & Chatterjee, S. (2010). *Design Research in Information Systems - Theory and Practice* | (Vol. 22). Springer New York. Obtido de <http://www.springer.com/la/book/9781441956521>
- Hevner, A., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research, 28(1), 75–105.
- Hsu, Y.-C., & Ching, Y.-H. (2013). Mobile App Design for Teaching and Learning: Educators' Experiences in an Online Graduate Course. *International Review of Research in Open and Distance Learning*, 14(4), 117–139.
- Hu, P. J., Chau, P. Y. K., Sheng, O. R. L., & Tam, K. Y. (1999). Examining the Technology Acceptance Model Using Physician Acceptance of Telemedicine Technology. *Journal of Management Information Systems*, 16(2), 91–112. <https://doi.org/10.1080/07421222.1999.11518247>
- Ibáñez, M. B., Di-Serio, Á., & Delgado-Kloos, C. (2014). Gamification for Engaging Computer Science Students in Learning Activities: A Case Study. *IEEE Transactions on Learning Technologies*, 7(3), 291–301. <https://doi.org/10.1109/TLT.2014.2329293>
- Iosup, A., & Epema, D. (2014). An Experience Report on Using Gamification in Technical Higher Education. Em *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (pp. 27–32). New York, NY, USA: ACM. <https://doi.org/10.1145/2538862.2538899>
- Järvinen, P. (2004). Research Questions Guiding Selection of an Appropriate Research Method. Department of Computer Sciences, University of Tampere.
- Järvinen, P. (2005). Action research as an approach in design science. European Academy of Management, Munich.
- Jenkins, T. (2002). On the difficulty of learning to program (Vol. 4, pp. 53–58). Apresentado na 3rd annual Conference of LTSN-ICS Centre for Information and Computer Sciences.

- John, O. P., & Srivastava, S. (1999). The Big Five Trait taxonomy: History, measurement, and theoretical perspectives. Em *Handbook of personality: Theory and research*, 2nd ed. (pp. 102–138). New York, NY, US: Guilford Press.
- Joint Task Force on Computing Curricula, A. for C. M. (ACM), & Society, I. C. (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. New York, NY, USA: ACM.
- Kapp, K. M. (2012). *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. John Wiley & Sons.
- Kapp, K. M., Blair, L., & Mesch, R. (2013). *The gamification of learning and instruction fieldbook: Theory into practice*. New York: NY: John Wiley & Sons.
- Karakus, M., Uludag, S., Guler, E., Turner, S. W., & Ugur, A. (2012). Teaching computing and programming fundamentals via App Inventor for Android. *International Conference on Information Technology Based Higher Education and Training (ITHET)* (pp. 1–8). <https://doi.org/10.1109/ITHET.2012.6246020>
- Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming. *Procedia - Social and Behavioral Sciences*, 47, 1991–1999. <https://doi.org/10.1016/j.sbspro.2012.06.938>
- Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. *The Internet and Higher Education*, 8(1), 13–24. <https://doi.org/10.1016/j.iheduc.2004.12.001>
- Kim, J. T., & Lee, W.-H. (2015). Dynamical model for gamification of learning (DMGL). *Multimedia Tools and Applications*, 74(19), 8483–8493. <https://doi.org/10.1007/s11042-013-1612-8>
- Kim, K., & Ahn, S. J. G. (2017). The Role of Gamification in Enhancing Intrinsic Motivation to Use a Loyalty Program. *Journal of Interactive Marketing*, 40, 41–51.
- Kim, S. H., & Jeon, J. W. (2009). Introduction for freshmen to embedded systems using LEGO Mindstorms. *IEEE Transactions on Education*, 52(1), 99–108.
- Kinnunen, P., Marttila-Kontio, M., & Pesonen, E. (2013). Getting to Know Computer Science Freshmen. Em *Proceedings of the 13th Koli Calling International Conference on Computing Education Research* (pp. 59–66). New York, NY, USA: ACM. <https://doi.org/10.1145/2526968.2526975>
- Knutas, A., Ikonen, J., Nikula, U., & Porras, J. (2014). Increasing Collaborative Communications in a Programming Course with Gamification: A Case Study. Em *Proceedings of the 15th International Conference on Computer Systems and*

- Technologies* (pp. 370–377). New York, NY, USA: ACM.
<https://doi.org/10.1145/2659532.2659620>
- Kölling, M. (2010). The Greenfoot Programming Environment. *Trans. Comput. Educ.*, 10(4), 14:1–14:21. <https://doi.org/10.1145/1868358.1868361>
- Kölling, M., & Henriksen, P. (2004). Greenfoot: combining object visualisation with interaction. Em *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications* (pp. 73–82). ACM.
- Konecki, M. (2014). Problems in Programming Education and Means of Their Improvement. Em *International Scientific Book* (pp. 459–470). Vienna, Austria: DAAAM International.
- Kordaki, M. (2010). A Drawing and Multi-representational Computer Environment for Beginners' Learning of Programming Using C: Design and Pilot Formative Evaluation. *Comput. Educ.*, 54(1), 69–87.
<https://doi.org/10.1016/j.compedu.2009.07.012>
- Korkmaz, Ö., & Altun, H. (2014). A validity and reliability study of the Attitude Scale of Computer Programming Learning (ASCOPL). *Mevlana International Journal of Education*, 4(1), 30–43.
- Koufaris, M. (2002). Applying the Technology Acceptance Model and Flow Theory to Online Consumer Behavior. *Information Systems Research*, 13(2), 205–223.
<https://doi.org/10.1287/isre.13.2.205.83>
- Kumar, A. N. (2004). Three Years of Using Robots in an Artificial Intelligence Course: Lessons Learned. *J. Educ. Resour. Comput.*, 4(3), 2.
<https://doi.org/10.1145/1083310.1083311>
- Kyewski, E., & Krämer, N. C. (2018). To gamify or not to gamify? An experimental field study of the influence of badges on motivation, activity, and performance in an online learning course. *Computers & Education*, 118(Supplement C), 25–37.
<https://doi.org/10.1016/j.compedu.2017.11.006>
- Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *SIGCSE Bull.*, 37(3), 14–18. <https://doi.org/10.1145/1151954.1067453>
- Lam, P., McNaught, C., Lee, J., & Chan, M. (2014). Disciplinary difference in students' use of technology, experience in using eLearning strategies and perceptions towards eLearning. *Computers & Education*, 73, 111–120.
<https://doi.org/10.1016/j.compedu.2013.12.015>
- Landers, R. N., Bauer, K. N., & Callan, R. C. (2017). Gamification of task performance with leaderboards: A goal setting experiment. *Computers in Human Behavior*, 71, 508–515.
<https://doi.org/10.1016/j.chb.2015.08.008>

- Landers, R. N., & Callan, R. C. (2011). Casual Social Games as Serious Games: The Psychology of Gamification in Undergraduate Education and Employee Training. Em M. Ma, A. Oikonomou, & L. C. Jain (Eds.), *Serious Games and Edutainment Applications* (pp. 399–423). Springer London. https://doi.org/10.1007/978-1-4471-2161-9_20
- Lederman, N. G., & Abell, S. K. (2014). *Handbook of Research on Science Education*. Taylor & Francis. Obtido de <https://books.google.pt/books?id=jNwABAAAQBAJ>
- Lee, A. S., & Baskerville, R. L. (2003). Generalizing Generalizability in Information Systems Research. *Info. Sys. Research*, 14(3), 221–243. <https://doi.org/10.1287/isre.14.3.221.16560>
- Lee, J. J., & Hammer, J. (2011). Gamification in Education: What, How, Why Bother? *Academic Exchange Quarterly*, 15(2), 146.
- Liao, L.-F. (2006). A Flow Theory Perspective on Learner Motivation and Behavior in Distance Education. *Distance Education*, 27(1), 45–62. <https://doi.org/10.1080/01587910600653215>
- Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., Thomas, L. (2004). A Multi-national Study of Reading and Tracing Skills in Novice Programmers. Em *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education* (pp. 119–150). New York, NY, USA: ACM. <https://doi.org/10.1145/1044550.1041673>
- Liu, M., Lee, J., Kang, J., & Liu, S. (2016). What We Can Learn from the Data: A Multiple-Case Study Examining Behavior Patterns by Students with Different Characteristics in Using a Serious Game. *Technology, Knowledge and Learning*, 21(1), 33–57. <https://doi.org/10.1007/s10758-015-9263-7>
- Lou, H., Chau, P., & Li, D. (2005). Understanding Individual Adoption of Instant Messaging: An Empirical Investigation. *Journal of the Association for Information Systems*, 6(4), 5.
- Lu, Y., Zhou, T., & Wang, B. (2009). Exploring Chinese users' acceptance of instant messaging using the theory of planned behavior, the technology acceptance model, and the flow theory. *Computers in Human Behavior*, 25(1), 29–39. <https://doi.org/10.1016/j.chb.2008.06.002>
- Ma, L. (2007). *Investigating and Improving Novice Programmers' Mental Models of Programming Concepts* (PhD Dissertation). University of Strathclyde, Department of Computer & Information Sciences, Glasgow.
- Mairesse, F., Walker, M. A., Mehl, M. R., & Moore, R. K. (2007). Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. *J. Artif. Int. Res.*, 30(1), 457–500.

- Malik, S. I., & Coldwell-Neilson, J. (2017). Impact of a New Teaching and Learning Approach in an Introductory Programming Course. *Journal of Educational Computing Research*, 55(6), 789–819. <https://doi.org/10.1177/0735633116685852>
- Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014). Educational Games for Teaching Computer Programming. Em *Research on e-Learning and ICT in Education* (pp. 87–98). Springer, New York, NY. https://doi.org/10.1007/978-1-4614-6501-0_7
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251–266. [https://doi.org/10.1016/0167-9236\(94\)00041-2](https://doi.org/10.1016/0167-9236(94)00041-2)
- McCrae, R. R., & Costa, P. T. (1987). Validation of the five-factor model of personality across instruments and observers. *Journal of Personality and Social Psychology*, 52(1), 81–90. <https://doi.org/10.1037/0022-3514.52.1.81>
- McGill, M. M. (2012a). Learning to Program with Personal Robots: Influences on Student Motivation. *Trans. Comput. Educ.*, 12(1), 4:1–4:32. <https://doi.org/10.1145/2133797.2133801>
- McGill, M. M. (2012b). The Curriculum Planning Process for Undergraduate Game Degree Programs in the United Kingdom and United States. *Trans. Comput. Educ.*, 12(2), 7:1–7:47. <https://doi.org/10.1145/2160547.2160550>
- McLeod, A., Hewitt, B., Gibbs, D., & Kristof, C. (2017). Evaluating Motivation for the Use of an Electronic Health Record Simulation Game. *Perspectives in Health Information Management*, 14(Spring). Obtido de <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5430132/>
- McWhorter, W. I. (2008). *The Effectiveness of Using Lego Mindstorms Robotics Activities to Influence Self-Regulated Learning in a University Introductory Computer Programming Course* (PhD Dissertation). University of North Texas. Obtido de http://digital.library.unt.edu/ark:/67531/metadc6077/m2/1/high_res_d/dissertation.pdf
- Miljanovic, M. A., & Bradbury, J. S. (2016). Robot on!: A Serious Game for Improving Programming Comprehension. Em *Proceedings of the 5th International Workshop on Games and Software Engineering* (pp. 33–36). New York, NY, USA: ACM. <https://doi.org/10.1145/2896958.2896962>
- Milne, I., & Rowe, G. (2002). Difficulties in Learning and Teaching Programming—Views of Students and Tutors. *Education and Information Technologies*, 7(1), 55–66. <https://doi.org/10.1023/A:1015362608943>
- Moon, J.-W., & Kim, Y.-G. (2001). Extending the TAM for a World-Wide-Web context. *Information & Management*, 38(4), 217–230. [https://doi.org/10.1016/S0378-7206\(00\)00061-6](https://doi.org/10.1016/S0378-7206(00)00061-6)

- Moreno, A., Myller, N., Sutinen, E., & Ben-Ari, M. (2004). Visualizing Programs with Jeliot 3. Em *Proceedings of the Working Conference on Advanced Visual Interfaces* (pp. 373–376). New York, NY, USA: ACM. <https://doi.org/10.1145/989863.989928>
- Morrison, B. B., & DiSalvo, B. (2014). Khan Academy Gamifies Computer Science. Em *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (pp. 39–44). New York, NY, USA: ACM. <https://doi.org/10.1145/2538862.2538946>
- Mow, I. T. C. (2008). Issues and Difficulties in Teaching Novice Computer Programming. Em M. Iskander (Ed.), *Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education* (pp. 199–204). Springer Netherlands. Obtido de http://link.springer.com/chapter/10.1007/978-1-4020-8739-4_36
- Muntean, C. (2011). Raising engagement in e-learning through gamification (pp. 323–329). Apresentado na Proceedings 6th International Conference on Virtual Learning ICVL, Cluj-Napoca, Romania. Obtido de http://icvl.eu/2011/disc/icvl/documente/pdf/met/ICVL_ModelsAndMethodologies_paper42.pdf
- Ngidi, D. P. (2013). Students' Personality Traits and Learning Approaches. *Journal of Psychology in Africa*, 23(1), 149–152. <https://doi.org/10.1080/14330237.2013.10820610>
- Nikander, J., Korhonen, A., Seppälä, O., Karavirta, V., Silvasti, P., & Malmi, L. (2004). Visual Algorithm Simulation Exercise System with Automatic Assessment: TRAKLA2. *Informatics in Education - An International Journal*, 3(2), 267–288.
- O'Cass, A., & Fenech, T. (2003). Web retailing adoption: exploring the nature of internet users Web retailing behaviour. *Journal of Retailing and Consumer Services*, 10(2), 81–94. [https://doi.org/10.1016/S0969-6989\(02\)00004-8](https://doi.org/10.1016/S0969-6989(02)00004-8)
- O'Donovan, S., Gain, J., & Marais, P. (2013). A Case Study in the Gamification of a University-level Games Development Course. Em *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (pp. 242–251). New York, NY, USA: ACM. <https://doi.org/10.1145/2513456.2513469>
- Oliver, R., & Herrington, J. (2003). Exploring Technology-Mediated Learning from a Pedagogical Perspective. *Interactive Learning Environments*, 11(2), 111–126. <https://doi.org/10.1076/ilee.11.2.111.14136>
- Özbek, V., Alnıaçık, Ü., Koc, F., Akkılıç, M. E., & Kaş, E. (2014). The Impact of Personality on Technology Acceptance: A Study on Smart Phone Users. *Procedia - Social and Behavioral Sciences*, 150, 541–551. <https://doi.org/10.1016/j.sbspro.2014.09.073>
- Papert, S., & Harel, I. (1991). Situating Constructionism. Em S. Papert & I. Harel (Eds.), *Constructionism*. Norwood, NJ: Ablex Publishing Corporation. Obtido de <http://www.papert.org/articles/SituatingConstructionism.html>

- Pappas, I. O., Giannakos, M. N., & Jaccheri, L. (2016). Investigating Factors Influencing Students' Intention to Dropout Computer Science Studies. Em *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 198–203). New York, NY, USA: ACM. <https://doi.org/10.1145/2899415.2899455>
- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., ... Paterson, J. (2007). A survey of literature on the teaching of introductory programming. Em *Working group reports on ITiCSE on Innovation and technology in computer science education* (pp. 204–223). New York, NY, USA: ACM. <https://doi.org/10.1145/1345443.1345441>
- Pettit, R. K., McCoy, L., Kinney, M., & Schwartz, F. N. (2015). Student perceptions of gamified audience response system interactions in large group lectures and via lecture capture technology. *BMC Medical Education*, 15(1), 92. <https://doi.org/10.1186/s12909-015-0373-7>
- Phares, E. J. (1988). *Introduction to personality, 2nd ed.* Glenview, IL, US: Scott, Foresman & Co.
- Piaget, J. (1976). Piaget's Theory. Em *Piaget and His School* (pp. 11–23). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-46323-5_2
- Picoto, W. N., Bélanger, F., & Palma-dos-Reis, A. (2014). An organizational perspective on m-business: usage factors and value determination. *European Journal of Information Systems*, 23(5), 571–592.
- Piteira, M., & Costa, C. (2013). Learning Computer Programming: Study of Difficulties in Learning Programming. Em *Proceedings of the 2013 International Conference on Information Systems and Design of Communication* (pp. 75–80). New York, NY, USA: ACM. <https://doi.org/10.1145/2503859.2503871>
- Piteira, M., & Costa, C. J. (2006). Avaliação da usabilidade percebida: plataforma de e-learning Moodle. Em *Proceedings of the IADIS International Conference on WWW/Internet* (Vol. 2006, pp. 19–25). Murcia, Spain.
- Piteira, M., & Costa, C. J. (2012). Computer Programming and Novice Programmers. Em *Proceedings of the Workshop on Information Systems and Design of Communication* (pp. 51–53). New York, NY, USA: ACM. <https://doi.org/10.1145/2311917.2311927>
- Piteira, M., & Costa, C. J. (2014). Aprendizagem da Programação: Problemas e Soluções (pp. 46–67). Apresentado na Conferência da Associação Portuguesa de Sistemas de Informação, Santarém. <http://dx.doi.org/10.18803/capsi.v14.46-67>
- Piteira, M., & Costa, C. J. (2017). Gamification: Conceptual framework to online courses of learning computer programming. Em *2017 12th Iberian Conference on Information*

- Systems and Technologies (CISTI)* (pp. 1–7). ISCTE-IUL - Lisboa.
<https://doi.org/10.23919/CISTI.2017.7975695>
- Piteira, M., & Costa, C. J. (2017). Gamification: Conceptual Framework to Online Courses of Learning Computer Programming. Apresentado na Conferência Ibérica de Sistemas e Tecnologias de Informação, ISCTE-IUL - Lisboa.
- Piteira, M., Costa, C. J., & Aparício, M. (2017). A conceptual framework to implement gamification on online courses of computer programming learning: implementation. Em *ICERI2017 Proceedings* (pp. 7022–7031). Seville, SPAIN: IATED.
- Piteira, M., Costa, C. J., & Haddad, S. R. (2012). Educational Computer Programming Tools. Em *Proceedings of the Workshop on Open Source and Design of Communication* (pp. 57–60). New York, NY, USA: ACM. <https://doi.org/10.1145/2316936.2316947>
- Piteira, M., & Haddad, S. R. (2011). Innovate in Your Program Computer Class: An Approach Based on a Serious Game. Em *Proceedings of the 2011 Workshop on Open Source and Design of Communication* (pp. 49–54). New York, NY, USA: ACM. <https://doi.org/10.1145/2016716.2016730>
- Radošević, D., Orehovački, T., & Lovrenčić, A. (2009). Verificator: Educational Tool for Learning Programming. *Informatics in Education*, 8(2), 261–280.
- Rahmat, M., Shahrani, S., Latih, R., Yatim, N. F. M., Zainal, N. F. A., & Rahman, R. A. (2012). Major Problems in Basic Programming that Influence Student Performance. *Procedia - Social and Behavioral Sciences*, 59, 287–296. <https://doi.org/10.1016/j.sbspro.2012.09.277>
- Rahmat, M., Shahrani, S., Latih, R., Yatim, N. F. M., Zainal, N. F. A., & Rahman, R. A. (2012). Major Problems in Basic Programming that Influence Student Performance. *Procedia - Social and Behavioral Sciences*, 59, 287–296. <https://doi.org/10.1016/j.sbspro.2012.09.277>
- Rajala, T., Laakso, M.-J., Kaila, E., & Salakoski, T. (2007). VILLE: A Language-independent Program Visualization Tool. Em *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research - Volume 88* (pp. 151–159). Darlinghurst, Australia, Australia: Australian Computer Society, Inc. Obtido de <http://dl.acm.org/citation.cfm?id=2449323.2449340>
- Renumol, V. G., Janakiram, D., & Jayaprakash, S. (2010). Identification of Cognitive Processes of Effective and Ineffective Students During Computer Programming. *Trans. Comput. Educ.*, 10(3), 10:1–10:21. <https://doi.org/10.1145/1821996.1821998>
- Richards, C., Thompson, C. W., & Graham, N. (2014). Beyond Designing for Motivation: The Importance of Context in Gamification. Em *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-human Interaction in Play* (pp. 217–226). New York, NY, USA: ACM. <https://doi.org/10.1145/2658537.2658683>

- Richter, G., Raban, D. R., & Rafaeli, S. (2015). Studying Gamification: The Effect of Rewards and Incentives on Motivation. Em *Gamification in Education and Business* (pp. 21–46). Springer, Cham. https://doi.org/10.1007/978-3-319-10208-5_2
- Ringle, C., Wende, S., & Will, A. (2005). Smart-PLS Version 2.0 M3. *University of Hamburg*.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137–172. <https://doi.org/10.1076/csed.13.2.137.14200>
- Rodrigues, L. F. (2016). *A adoção de aplicações informáticas com características de jogos online: uma perspectiva no setor o e-banking em Portugal* (PhD Dissertation). ISCTE - IUL, Lisboa.
- Rodrigues, L. F., Costa, C. J., & Oliveira, A. (2013). The Adoption of Gamification in e-Banking. Em *Proceedings of the 2013 International Conference on Information Systems and Design of Communication* (pp. 47–55). New York, NY, USA: ACM. <https://doi.org/10.1145/2503859.2503867>
- Rodrigues, L. F., Costa, C. J., & Oliveira, A. (2016). Gamification: A framework for designing software in e-banking. *Computers in Human Behavior*, 62, 620–634. <https://doi.org/10.1016/j.chb.2016.04.035>
- Rodrigues, L. F., Oliveira, A., & Costa, C. J. (2016a). Does ease-of-use contributes to the perception of enjoyment? A case of gamification in e-banking. *Computers in Human Behavior*, 61(Supplement C), 114–126. <https://doi.org/10.1016/j.chb.2016.03.015>
- Rodrigues, L. F., Oliveira, A., & Costa, C. J. (2016b). Playing seriously – How gamification and social cues influence bank customers to use gamified e-business applications. *Computers in Human Behavior*, 63, 392–407. <https://doi.org/10.1016/j.chb.2016.05.063>
- Rudolph, G. (2015). *Instructional Technology for Teaching of Novice Programmers At a University of Technology* (PhD Dissertation). Cape Peninsula University of Technology - Faculty of Informatics and Design.
- Saadatdoost, R., Sim, A. T. H., Jafarkarimi, H., & Hee, J. M. (2015). Exploring MOOC from education and Information Systems perspectives: a short literature review. *Educational Review*, 67(4), 505–518. <https://doi.org/10.1080/00131911.2015.1058748>
- Salleh, S. M., Shukur, Z., & Judi, H. M. (2013). Analysis of Research in Programming Teaching Tools: An Initial Review. *Procedia - Social and Behavioral Sciences*, 103(Supplement C), 127–135. <https://doi.org/10.1016/j.sbspro.2013.10.317>
- Sanders, D., & Dorn, B. (2003). Classroom Experience with Jeroo. *J. Comput. Sci. Coll.*, 18(4), 308–316.

- Schulte, C., & Bennedsen, J. (2006). What Do Teachers Teach in Introductory Programming? Em *Proceedings of the Second International Workshop on Computing Education Research* (pp. 17–28). New York, NY, USA: ACM.
<https://doi.org/10.1145/1151588.1151593>
- Schulte, C., Clear, T., Taherkhani, A., Busjahn, T., & Paterson, J. H. (2010). An Introduction to Program Comprehension for Computer Science Educators. *Proceedings of the 2010 ITiCSE Working Group Reports* (pp. 65–86). New York, NY, USA: ACM.
<https://doi.org/10.1145/1971681.1971687>
- Seaborn, K., & Fels, D. I. (2015). Gamification in theory and action: A survey. *International Journal of Human-Computer Studies*, 74, 14–31.
<https://doi.org/10.1016/j.ijhcs.2014.09.006>
- Seixas, L. da R., Gomes, A. S., & Filho, I. J. de M. (2016). Effectiveness of gamification in the engagement of students. *Computers in Human Behavior*, 58(Supplement C), 48–63. <https://doi.org/10.1016/j.chb.2015.11.021>
- Shernoff, D. J., Csikszentmihalyi, M., Schneider, B., & Shernoff, E. S. (2014). Student Engagement in High School Classrooms from the Perspective of Flow Theory. *Applications of Flow in Human Development and Education* (pp. 475–494). Springer, Dordrecht. https://doi.org/10.1007/978-94-017-9094-9_24
- Sherry, J. L. (2004). Flow and media enjoyment. *Communication Theory*, 14(4), 328–347.
- Shichel, Y., Ziserman, E., & Sipper, M. (2005). GP-Robocode: Using Genetic Programming to Evolve Robocode Players. Em M. Keijzer, A. Tettamanzi, P. Collet, J. van Hemert, & M. Tomassini (Eds.), *Genetic Programming: 8th European Conference, EuroGP 2005, Lausanne, Switzerland, March 30 - April 1, 2005. Proceedings* (pp. 143–154). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-31989-4_13
- Shih, Y.-C., & Fan, S.-T. (2013). Adoption of Instant Messaging By Travel Agency Workers in Taiwan: Integrating Technology Readiness with the Theory of Planned Behavior. *International Journal of Business and Information; Sansia*, 8(1), 120–136.
- Siemens, G. (2005). Connectivism: A learning theory for the digital age. Em *International Journal of Instructional Technology and Distance Learning* (Vol. 2). Obtido de http://www.itdl.org/Journal/Jan_05/index.htm
- Simões, J., Redondo, R. D., & Vilas, A. F. (2013). A social gamification framework for a K-6 learning platform. *Computers in Human Behavior*, 29(2), 345–353.
<https://doi.org/10.1016/j.chb.2012.06.007>
- Simon, H. A. (1996). *The Sciences of the Artificial (3rd Ed.)*. Cambridge, MA, USA: MIT Press.

- Soares-Aguiar, A., & Palma-Dos-Reis, A. (2008). Why do firms adopt e-procurement systems. *Using logistic regression to empirically conceptual model, IEEE transactions on engineering management*, 55 (1), 120–133.
- Sorva, J., & Sirkiä, T. (2010). UUhistle: A Software Tool for Visual Program Simulation. *Proceedings of the 10th Koli Calling International Conference on Computing Education Research* (pp. 49–54). New York, NY, USA: ACM. <https://doi.org/10.1145/1930464.1930471>
- Sousa, N. M., Costa, C. J., & Aparicio, M. (2017). Ba: um fator determinante no uso de sistemas de gestão do conhecimento. *Iberian Journal of Information Systems and Technologies (RISTI)*, 2017(22), 1–19. <https://doi.org/10.17013/risti.22.1-19>
- Tarhini, A., Hone, K., Liu, X., & Tarhini, T. (2017). Examining the moderating effect of individual-level cultural values on users' acceptance of E-learning in developing countries: a structural equation modeling of an extended technology acceptance model. *Interactive Learning Environments*, 25(3), 306–328. <https://doi.org/10.1080/10494820.2015.1122635>
- Teo, T., & Noyes, J. (2011). An assessment of the influence of perceived enjoyment and attitude on the intention to use technology among pre-service teachers: A structural equation modeling approach. *Computers & Education*, 57(2), 1645–1653. <https://doi.org/10.1016/j.compedu.2011.03.002>
- Thompson, K. (2011). 7 Things You Should Know About MOOCs. Obtido 5 de Outubro de 2017, de <https://library.educause.edu/resources/2011/11/7-things-you-should-know-about-moocs>
- Ty, L., R, A. T., (head, Y. P., Burnette, T., Capehart, A. C., Cosgrove, D., ... White, K. J. (1995). *A Brief Architectural Overview of Alice, a Rapid Prototyping System for Virtual Reality*. IEEE Computer Graphics and Applications.
- US Department of Labor. (2017). Software Developers : Occupational Outlook Handbook: : U.S. Bureau of Labor Statistics. Obtido 23 de Agosto de 2017, de <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm#tab-6>
- Vaishnavi, V., & Kuechler, W. (2004). *Design Research in Information Systems*. Obtido de <http://desrist.org/design-research-in-information-systems/>
- Van der Heijden, H. (2004). User Acceptance of Hedonic Information Systems. *MIS Quarterly*, 28(4), 695–704.
- Vasileva-Stojanovska, T., Malinovski, T., Vasileva, M., Jovevski, D., & Trajkovik, V. (2015). Impact of satisfaction, personality and learning style on educational outcomes in a blended learning environment. *Learning and Individual Differences*, 38(Supplement C), 127–135. <https://doi.org/10.1016/j.lindif.2015.01.018>

- Venkatesh, V., Brown, S. A., Maruping, L. M., & Bala, H. (2008). Predicting Different Conceptualizations of System Use: The Competing Roles of Behavioral Intention, Facilitating Conditions, and Behavioral Expectation. *MIS Quarterly*, 32(3), 483–502.
- Venkatesh, V., & Davis, F. D. (2000). A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science*, 46(2), 186–204. <https://doi.org/10.1287/mnsc.46.2.186.11926>
- Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 27(3), 425–478. <https://doi.org/10.2307/30036540>
- Venkatesh, V., Speier, C., & Morris, M. G. (2002). User Acceptance Enablers in Individual Decision Making About Technology: Toward an Integrated Model. *Decision Sciences*, 33(2), 297–316. <https://doi.org/10.1111/j.1540-5915.2002.tb01646.x>
- Venkatesh, V., Thong, J. Y., & Xu, X. (2012). Consumer acceptance and use of information technology: extending the unified theory of acceptance and use of technology. *MIS Quarterly*, 36(1), 157–178.
- Vilner, T., Zur, E., & Tavor, S. (2011). Integrating Greenfoot into CS1: A Case Study. Em *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education* (pp. 350–350). New York, NY, USA: ACM. <https://doi.org/10.1145/1999747.1999864>
- Voiskounsky, A. E., Mitina, O. V., & Avetisova, A. A. (2004). Playing online games: Flow experience. *PsychNology Journal*, 2(3), 259–281.
- Watson, C., & Li, F. W. B. (2014). Failure Rates in Introductory Programming Revisited. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education* (pp. 39–44). New York, NY, USA: ACM. <https://doi.org/10.1145/2591708.2591749>
- WebSite EduScratch. (2017). eduscratch. Obtido 1 de Novembro de 2017, de <http://eduscratch.dge.mec.pt/>
- Weibel, D., Wissmath, B., Habegger, S., Steiner, Y., & Groner, R. (2008). Playing online games against computer- vs. human-controlled opponents: Effects on presence, flow, and enjoyment. *Computers in Human Behavior*, 24(5), 2274–2291. <https://doi.org/10.1016/j.chb.2007.11.002>
- Werbach, K., & Hunter, D. (2012). *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton Digital Press.
- Werbach, K., & Hunter, D. (2015). *The Gamification Toolkit: Dynamics, Mechanics, and Components for the Win*. Wharton Digital Press.

- Wilson, B. C., & Shrock, S. (2001). Contributing to Success in an Introductory Computer Science Course: A Study of Twelve Factors. Em *Proceedings of the Thirty-second SIGCSE Technical Symposium on Computer Science Education* (pp. 184–188). New York, NY, USA: ACM. <https://doi.org/10.1145/364447.364581>
- Winslow, L. E. (1996). Programming pedagogy-a psychological overview. *SIGCSE Bull.*, 28(3), 17–22. <https://doi.org/10.1145/234867.234872>
- Wolber, D. (2011). App Inventor and Real-world Motivation. Em *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education* (pp. 601–606). New York, NY, USA: ACM. <https://doi.org/10.1145/1953163.1953329>
- Wright, S. (1934). The Method of Path Coefficients. *The Annals of Mathematical Statistics*, 5(3), 161–215.
- Xinogalos, S., Satratzemi, M., & Dagdilelis, V. (2006). An introduction to object-oriented programming with a didactic microworld: objectKarel. *Computers & Education*, 47(2), 148–171. <https://doi.org/10.1016/j.compedu.2004.09.005>
- Yadav, A., Stephenson, C., & Hong, H. (2017). Computational Thinking for Teacher Education. *Commun. ACM*, 60(4), 55–62. <https://doi.org/10.1145/2994591>
- Yi, M. Y., & Hwang, Y. (2003). Predicting the use of web-based information systems: self-efficacy, enjoyment, learning goal orientation, and the technology acceptance model. *International Journal of Human-Computer Studies*, 59(4), 431–449. [https://doi.org/10.1016/S1071-5819\(03\)00114-9](https://doi.org/10.1016/S1071-5819(03)00114-9)
- Yildirim, I. (2017). The effects of gamification-based teaching practices on student achievement and students' attitudes toward lessons. *The Internet and Higher Education*, 33, 86–92. <https://doi.org/10.1016/j.iheduc.2017.02.002>
- Zichermann, G., & Cunningham, C. (2011). *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps* (1st ed.). O'Reilly Media, Inc.

ANEXOS

Anexo A – Questionário Dificuldades na Aprendizagem da Programação

Questionário - Investigação das Principais Dificuldades no Ensino / Aprendizagem da Programação de Computadores

O presente questionário insere-se num trabalho de investigação de doutoramento que tem como objetivo identificar quais as principais dificuldades sentidas pelos alunos na aprendizagem da programação de computadores. O questionário tem um tempo máximo de resposta de cerca de 15 minutos.

Os campos assinalados com * são obrigatórios responder.

*Obrigatório

Informações Gerais

1. Indique o seu nome *

2. Indique a sua idade *

3. Género *

Marcar apenas uma oval.

Feminino

Masculino

4. Indique o nome do curso que frequenta atualmente? *

5. Qual o ano que se encontra a frequentar? *

Marcar apenas uma oval.

Primeiro ano

Segundo ano

Terceiro ano

6. Qual a instituição a que pertence? *

Marcar apenas uma oval.

Instituto Politécnico de Setúbal- ESTSetúbal

Experiência Computacional

7. Quantas horas passa por semana no computador *

Marcar tudo o que for aplicável.

- Mais de 2 e menos de 5
- Entre 5 e menos de 7
- Entre 7 e menos de 10
- Mais de 10 horas

Experiência Profissional

8. Tem experiência profissional? *

Marcar apenas uma oval.

- Sim
- Não

9. Se respondeu SIM na questão anterior, indique se têm experiência profissional na área da programação ou numa outra área

Marcar tudo o que for aplicável.

- Programação
- Outra: _____

Quais as principais dificuldades sentidas na aprendizagem da programação?

10. Na utilização do ambiente de desenvolvimento (IDE) *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Fácil de Aprender	<input type="radio"/>	Difícil de Aprender						

11. Na estruturação adequada do programa *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Fácil de Aprender	<input type="radio"/>	Difícil de Aprender						

12. Aprendizagem da sintaxe da linguagem de programação *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Fácil de Aprender	<input type="radio"/>	Difícil de Aprender						

13. Conceber um programa para a resolução de uma determinado problema *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Fácil de Aprender	<input type="radio"/>	Difícil de Aprender						

14. Dividir as funcionalidades em procedimentos *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Fácil de Aprender	<input type="radio"/>	Difícil de Aprender						

15. Encontrar erros no meu próprio programa *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Fácil de Aprender	<input type="radio"/>	Difícil de Aprender						

Dos conceitos de programação abaixo listados identifique qual o grau de dificuldade na aprendizagem para cada um deles

16. Variáveis *

(tempo de vida e âmbito)

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sabe / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

17. Estruturas de Seleção *

(ex: if, if-else, switch-case)

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0- Não sei / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

18. Estruturas de Repetição *

(ex: while, do-while, for)

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0- Não sei // 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

19. Tabelas (Arrays) *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

20. Ponteiros, Referências *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

21. Parâmetros (passagem por valor) *

(passagem por valor)

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

22. Parâmetros (passagem por referência) *

(passagem por referência)

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

23. Tipos Estruturados de Dados *

(ex: struct)

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

24. Tipos Abstratos de Dados *

(ex: pilhas e filas)

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

25. Manuseamento de entrada e saída de dados *

entrada de dados (ex: tipo de dados) e saída de dados (ex: formatação do tipo de dados)
 Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

26. Tratamento de Erros *

(tratamento de exceções)
 Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

27. Utilização das Bibliotecas da Linguagem *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Fácil de Aprender	<input type="radio"/>	Difícil de Aprender							

Onde sente que aprende mais sobre programação

Seleccione entre 1 que equivale a POUCO ÚTIL e 7 que equivale a MUITO ÚTIL. No caso de não saber ou não se aplicar seleccione 0

28. Nas aulas teóricas *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco Útil	<input type="radio"/>	Muito Útil							

29. Em sessões de resolução de exercícios com os colegas *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco Útil	<input type="radio"/>	Muito Útil							

30. Nas aulas de laboratório *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco Útil	<input type="radio"/>	Muito Útil							

31. Quando estudo sozinho *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco Útil	<input type="radio"/>	Muito Útil							

32. Quando trabalho sozinho nos exercícios da unidade curricular *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
Não sei / 1- Pouco Útil	<input type="radio"/>	Muito Útil							

33. Em sessões de esclarecimento de dúvidas com o docente da unidade curricular (horário de dúvidas) *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
Não sei / 1- Pouco Útil	<input type="radio"/>	Muito Útil							

Que tipos de materiais ajudaram ou poderão ajudar na aprendizagem da programação de computadores?

Selecione entre 1 que equivale a POUCO e 7 que equivale a MUITO. No caso de não saber ou não se aplicar selecione 0

34. Livro recomendado da unidade curricular *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1- Pouco	<input type="radio"/>	Muito							

35. Apontamentos tirados nas aulas teóricas *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1- Pouco	<input type="radio"/>	Muito							

36. Cópia dos acetatos *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1- Pouco	<input type="radio"/>	Muito							

37. Exercícios resolvidos em sala de aula *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco	<input type="radio"/>	Muito							

38. Exemplos de programas completos *

(ex: código fonte)

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco	<input type="radio"/>	Muito							

39. Imagens estáticas de programas estruturados *

(ver o código do programa estático impresso em papel ou através de meio digital por exemplo num pdf)

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco	<input type="radio"/>	Muito							

40. Visualizações interativas através de ferramentas de visualização gráfica como por exemplo (Alice, BlueJ, RoboCode ou outra similar) *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco	<input type="radio"/>	Muito							

41. Tutoriais disponíveis na Internet *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco	<input type="radio"/>	Muito							

42. Foruns de discussão na Internet *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco	<input type="radio"/>	Muito							

43. Vídeos educativos disponíveis no youtube *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco	<input type="radio"/>	Muito							

44. Twitter - Páginas dedicadas à aprendizagem da programação *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco	<input type="radio"/>	Muito							

45. Facebook- Páginas dedicadas à aprendizagem da programação *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco	<input type="radio"/>	Muito							

46. Recursos disponibilizados na plataforma Moodle *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei / 1 - Pouco	<input type="radio"/>	Muito							

47. Indique outros tipos de materiais, caso utilize:

Conhecimentos Anteriores

Selecione entre 1 que equivale a Muito Baixos e 7 que equivale a Muito Elevados. No caso de não saber ou não se aplicar selecione 0

48. No ensino secundário frequentou disciplinas de programação de computadores? *

Marcar apenas uma oval.

- Sim
- Não

49. Antes de ingressar no ensino superior qual o nível de conhecimento da linguagem C *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tinha / 1 - Muito Baixos	<input type="radio"/>	Muito Elevados							

50. Antes de ingressar no ensino superior qual o nível de conhecimento da linguagem C++ *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tinha / 1 - Muito Baixos	<input type="radio"/>	Muito Elevados							

51. Antes de ingressar no ensino superior qual o nível de conhecimento da linguagem Basic *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tinha / 1 - Muito Baixos	<input type="radio"/>	Muito Elevados							

52. Antes de ingressar no ensino superior qual o nível de conhecimento da linguagem Java *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tinha / 1 - Muito Baixos	<input type="radio"/>	Muito Elevados							

53. Antes de ingressar no ensino superior qual o nível de conhecimento da linguagem Pascal *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tinha / 1 - Muito Baixos	<input type="radio"/>	Muito Elevados							

54. Antes de ingressar no ensino superior qual o nível de conhecimento da linguagem JavaScript *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tinha / 1 - Muito Baixos	<input type="radio"/>	Muito Elevados							

55. Antes de ingressar no ensino superior qual o nível de conhecimento da linguagem Python *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tinha / 1 - Muito Baixos	<input type="radio"/>	Muito Elevados							

Frequência de utilização de linguagens no curso

Selecione entre 1 que equivale a Muito Frequente e 7 que equivale a Pouco Frequente. No caso de não saber ou não se aplicar selecione 0

56. Durante o curso tenho utilizado com frequência a linguagem C *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tenho utilizado / 1 - Muito Frequente	<input type="radio"/>	Pouco Frequente							

57. Durante o curso tenho utilizado com frequência a linguagem C++ *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tenho utilizado / 1 - Muito Frequente	<input type="radio"/>	Pouco Frequente							

58. Durante o curso tenho utilizado com frequência a linguagem Pascal *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tenho utilizado / 1 - Muito Frequente	<input type="radio"/>	Pouco Frequente							

59. Durante o curso tenho utilizado com frequência a linguagem Python *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tenho utilizado / 1 - Muito Frequente	<input type="radio"/>	Pouco Frequente							

60. Durante o curso tenho utilizado com frequência a linguagem Java *

Marcar apenas uma oval.

	0	1	2	3	4	5	6	7	
0 - Não sei ou não tenho utilizado / 1 - Muito Frequente	<input type="radio"/>	Pouco Frequente							

ATITUDE FACE À PROGRAMAÇÃO

61. Gosto de aprender programação! *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo completamente	<input type="radio"/>	Concordo inteiramente						

62. Gosto de programar! *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo completamente	<input type="radio"/>	Concordo inteiramente						

63. Acho que programar vai ser útil para a minha vida profissional! *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo completamente	<input type="radio"/>	Concordo inteiramente						

64. Programar é fácil! *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo completamente	<input type="radio"/>	Concordo inteiramente						

65. Acho que programar é útil na minha formação! *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo completamente	<input type="radio"/>	Concordo inteiramente						

66. Invisto muito tempo na aprendizagem da programação! *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo completamente	<input type="radio"/>	Concordo inteiramente						

67. Sei programar muito bem! *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo completamente	<input type="radio"/>	Concordo inteiramente						

Com tecnologia
 Google Forms

Anexo B - Questionário Percepções dos Estudantes – Gamificação

A Gamificação na Aprendizagem Online da Programação

***Obrigatório**

1. Selecione a sua idade. *

Marcar apenas uma oval.

- Menos de 18 anos
- Entre 18 e 22 anos
- Entre 23 e 30 anos
- Entre 31 e 40 anos
- Mais de 41 anos

2. Selecione o seu Género. *

Marcar apenas uma oval.

- Masculino
- Feminino

3. Selecione a formação que se encontra atualmente a realizar. *

Marcar apenas uma oval.

- Curso Técnico Superior Profissional (CTEsP)
- Licenciatura

4. Quantas horas passa por semana no computador? *

Marcar apenas uma oval.

- Entre 1 e menos de 4 horas
- Entre 4 e menos de 8 horas
- Entre 8 e menos de 12 horas
- Mais de 12 horas

5. Caso possua um código previamente atribuído, insira-o abaixo. Em alternativa, insira o seu número de aluno.

6. Selecione a plataforma que utilizou para fazer o curso online de programação. *

Marcar apenas uma oval.

- Plataforma Online Moodle - Curso SMARTMOOC *Passe para a pergunta 8.*
- Plataforma Online CODEACADEMY

7. Indique qual ou quais os cursos que frequentou na plataforma CodeAcademy. *

Marcar tudo o que for aplicável.

- Java
- Python
- JavaScript
- PHP
- AngularJs
- HTML/CSS
- JQuery
- SQL
- Outra

Quando estou a frequentar o curso online de programação e a realizar as atividades de aprendizagem, sinto que:

8. TD1: O tempo parece passar mais rapidamente. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

9. TD2: Perco por vezes a noção do tempo a passar. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

10. TD3: O tempo voa. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

11. TD4: A maior parte das vezes, acabo por ficar mais tempo para além do tempo que tinha planeado ficar. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

12. TD5: Costumo ficar mais tempo do que inicialmente pretendia. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

13. FLO3: Nunca dou pelo tempo a passar. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

Quando estou a frequentar o curso online de programação e a realizar as atividades de aprendizagem:

14. FI1: Consigo desviar a minha atenção da maioria das distrações. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

15. FI2: Fico absorvido no que estou a fazer. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

16. FI3: Fico imerso na tarefa que estou a desempenhar. (considere imerso um estado total de absorção) *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

17. FI4: Distraio-me facilmente com outros assuntos. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

18. FI5: A minha atenção não é desviada facilmente para outros assuntos. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

19. FLO2: Os meus pensamentos/atividades surgem rapidamente e suavemente. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

20. FLO4: Não tenho dificuldades em concentrar-me. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

21. FLO5: A minha mente está completamente limpa. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

22. FLO6: Fico totalmente absorvido naquilo que estou a fazer. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

23. FLO7: Os pensamentos / movimentos certos, ocorrem livremente. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

24. FLO10: Estou completamente perdido nos meus pensamentos. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

Quando estou a frequentar o curso online de programação e a realizar as atividades de aprendizagem sinto que:

25. HE1: Divirto-me a interagir com as diversas atividades, conteúdos e ferramentas disponíveis no curso. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

26. HE2: Frequentar o curso online de programação proporciona-me muitos momentos divertidos. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

27. HE3: Gosto de frequentar o curso online de programação. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

28. HE4: Frequentar o curso online de programação aborrece-me. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

29. CO1: Quando estou a frequentar o curso online, tenho o controlo sobre o que estou a fazer. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

30. CO2: Não tenho o controlo na minha interação com o curso online. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

31. CO3: O curso online permite-me controlar a minha interação com o computador. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

32. CU1: Frequentar o curso online estimula a minha curiosidade. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

33. CU2: A interação com curso online torna-me mais curioso. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

34. CU3: Frequentar o curso estimula a minha imaginação. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

35. FU1: A utilização da interface do curso online não requer um grande esforço mental. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

36. FU2: A interação com o curso online é clara e compreensível. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

37. FU3: É fácil, através da interface, fazer o que pretendo. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

38. FU4: A interface do curso online é fácil de usar. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

39. PU1: A experiência de aprendizagem no curso, é divertida. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

40. PU2: A experiência de aprendizagem no curso, é agradável. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

41. PU3: A experiência de aprendizagem no curso, é empolgante. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

42. PU4: A experiência de aprendizagem no curso, é interessante. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

43. UT1: O curso online facilita a minha aprendizagem dos conceitos fundamentais de programação. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

44. UT2: O curso online permite-me alcançar os conhecimentos necessários sobre os conceitos fundamentais de programação. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

45. UT3: Frequentar o curso permite-me aperfeiçoar o meu conhecimento sobre os conceitos fundamentais de programação. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

46. UT4: Torno-me mais eficaz a programar quando frequento o curso online. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

47. UT5: O curso online é útil. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

48. IP4: O curso online é útil para a minha formação. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

49. IP1: Frequentar o curso online ajuda-me a melhorar o meu desempenho na programação. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

50. IP2: Frequentar o curso online ajuda-me a melhorar a eficiência na programação. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

51. IP3: O curso online facilita-me a realização das tarefas e atividades de aprendizagem. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
 Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

52. FLO1: No curso online, sinto que existe a quantidade certa de desafio. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
 Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

53. FLO8: Sei o que tenho que fazer em cada etapa do curso. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
 Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

54. FLO9: No curso online, tenho o controlo sobre tudo o que estou a fazer. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
 Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

55. Comparada com todas as outras atividades de aprendizagem online em que participei, esta em particular é: *

1 - Fácil 2 - Moderadamente Fácil 3 - Um Pouco Fácil 4 - Nem Fácil, Nem Difícil 5 - Um Pouco Difícil 6 - Moderadamente Difícil 7 - Difícil
 Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Fácil	<input type="radio"/>	Difícil						

56. Considero que a minha competência, nesta área de programação é: *

1 - Baixa 2 - Moderadamente Baixa 3 - Um Pouco Baixa 4 - Nem Baixa, Nem Alta 5 - Um Pouco Alta 6 - Moderadamente Alta 7 - Alta
 Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Baixa	<input type="radio"/>	Alta						

57. O atual volume de trabalho e a realização de tarefas associadas são: *

1 - Muito Baixos 2 - Moderadamente Baixos 3 - Um Pouco Baixos 4 - Na Medida Certa 5 - Um Pouco Altos 6 - Moderadamente Altos 7 - Muitos Altos

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Muito Baixos	<input type="radio"/>	Muito Altos						

58. BI1: Planeio-o continuar a frequentar o curso online. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

59. BI3: Pretendo continuar a frequentar o curso online no futuro. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

60. BI2: Tenciono falar sobre o curso com os meus amigos. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

61. U1: Durante o último mês, quantas vezes acedeu ao curso online? *

Marcar apenas uma oval.

- Apenas uma
- Duas
- Três
- Quatro
- Cinco
- Mais de cinco

62. U2: Indique em média a maior duração de uma das sessões no curso online. (sessão - tempo decorrido entre o login e logout da aplicação) *

Marcar apenas uma oval.

- Menos de uma hora
- Entre uma hora e menos de duas horas
- Entre duas horas e menos de três horas
- Entre três horas e menos de 4 horas
- Entre quatro horas e cinco horas
- Mais de cinco horas

Considere as medalhas (badges) associadas às atividades de aprendizagem no curso online de programação. Como vê a sua utilização nas atividades de aprendizagem online?

63. B1: São motivadoras. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

64. B2: Perturbam o meu trabalho. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

65. B3: Tentar obter as medalhas tem um efeito positivo no meu comportamento ao longo do curso. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

66. B4: O aspeto visual das medalhas é bom. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

67. B5: Estou satisfeito com os critérios para obter as medalhas. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

68. B7: Considero que as medalhas são úteis no processo de aprendizagem. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

69. B6: Acho que as medalhas poderiam ser usadas na plataforma de eLearning (moodle) em todas as disciplinas do plano curricular. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

Considere os pontos associados às atividades de aprendizagem no curso online de programação. Como vê a sua utilização nas atividades de aprendizagem?

70. P1: São motivadores. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

71. P2: Perturbam o meu trabalho. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

72. P3: Têm um efeito positivo no meu comportamento ao longo do curso. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

73. P4: Estou satisfeito com os critérios definidos para ganhar pontos. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

74. P6: Considero que os pontos são úteis no processo de aprendizagem. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

75. P5: Os pontos poderiam ser utilizados na plataforma de eLearning (moodle) em todas as disciplinas do plano curricular. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

Relativamente ao Quadro de Pontos (LeardBoard) disponível no curso online de programação indique:

76. L2: Consulta com frequência para visualizar a pontuação. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

77. L3: Sinto-me incomodado com os resultados listados e visíveis para todos. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

78. L4: Sinto-me incomodado com os resultados listados e o meu nome não estar nessa lista. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

79. L1: Acho o quadro de pontos (LeardBoard) útil. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

Relativamente à barra de progresso disponível no curso online de programação indique:

80. PG2: Visualizo diversas vezes a barra de progresso para identificar as atividades realizadas e por realizar. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

81. PG1: Acho a barra de progresso útil. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

Pessoalmente, vejo-me como sendo uma pessoa:

82. PER1: Extrovertida, entusiasta *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

83. PER2: Crítica, conflituosa *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

84. PER3: Confiável, auto-disciplinada *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

85. PER4: Ansiosa, facilmente aborrecida *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

86. PER5: Aberta a novas experiências, criativa *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

87. PER6: Reservada, calada *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

88. PER7: Simpática, calorosa *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

89. PER8: Desorganizada, descuidada. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

90. PER9: Calma, emocionalmente estável. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

91. PER10: Convencional, não criativa. *

1 - Discordo Totalmente 2 - Discordo Moderadamente 3 - Discordo um Pouco 4 - Nem Concordo, Nem Discordo, 5 - Concordo um Pouco 6 - Concordo Moderadamente 7 - Concordo Totalmente
Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Discordo Totalmente	<input type="radio"/>	Concordo Totalmente						

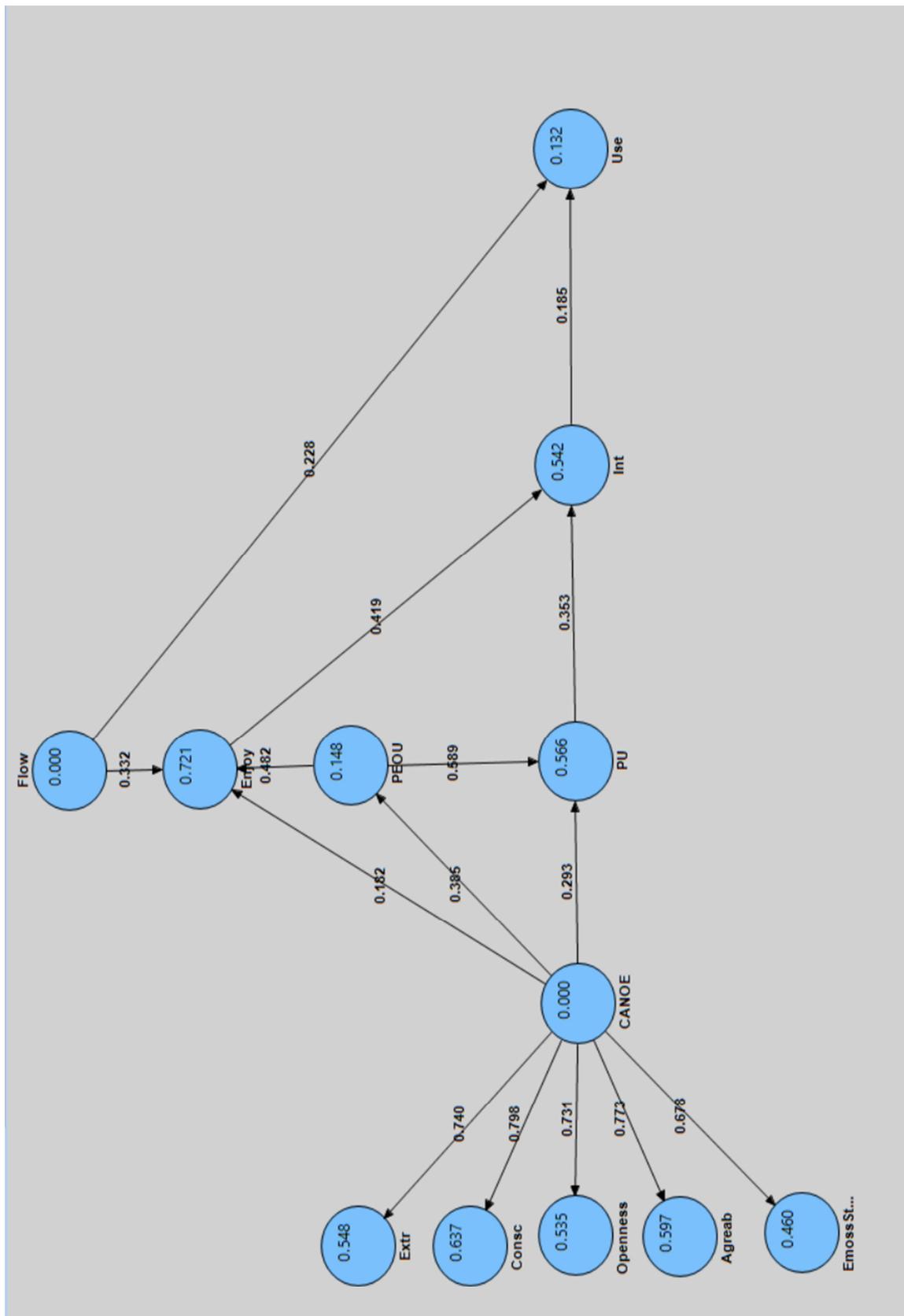
Anexo C – Cross-Loadings

	Agreab	CANOE	Consc	EmossStab	Enjoy	Extr	Flow	Int	Openness	PEOU	PU	Use
BI1	0.348	0.463	0.440	0.313	0.593	0.274	0.442	0.895	0.338	0.474	0.594	0.208
BI2	0.359	0.480	0.445	0.261	0.647	0.248	0.547	0.872	0.450	0.547	0.655	0.322
BI3	0.390	0.462	0.396	0.251	0.684	0.233	0.510	0.959	0.426	0.545	0.644	0.306
FLO2	0.412	0.432	0.343	0.237	0.550	0.168	0.778	0.393	0.420	0.357	0.437	0.350
FLO4	0.347	0.454	0.389	0.263	0.505	0.211	0.786	0.367	0.453	0.434	0.473	0.263
FLO5	0.437	0.531	0.464	0.316	0.454	0.356	0.757	0.302	0.389	0.305	0.339	0.128
FLO6	0.376	0.453	0.368	0.297	0.546	0.180	0.762	0.396	0.443	0.482	0.502	0.262
FLO7	0.339	0.449	0.309	0.427	0.542	0.133	0.821	0.415	0.454	0.447	0.421	0.270
FLO8	0.339	0.434	0.253	0.267	0.606	0.213	0.680	0.460	0.523	0.626	0.547	0.256
FLO9	0.351	0.518	0.433	0.363	0.700	0.272	0.705	0.527	0.492	0.627	0.626	0.188
FU2	0.225	0.326	0.273	0.216	0.724	0.052	0.599	0.550	0.419	0.961	0.676	0.144
FU3	0.229	0.326	0.236	0.259	0.690	0.036	0.619	0.482	0.429	0.948	0.604	0.130
FU4	0.348	0.441	0.354	0.267	0.768	0.118	0.607	0.611	0.520	0.957	0.724	0.121
PER1	0.479	0.740	0.547	0.317	0.259	1.000	0.286	0.276	0.448	0.074	0.219	-0.086
PER1	0.479	0.740	0.547	0.317	0.259	1.000	0.286	0.276	0.448	0.074	0.219	-0.086
PER3	0.531	0.798	1.000	0.525	0.435	0.547	0.481	0.468	0.355	0.304	0.372	0.153
PER3	0.531	0.798	1.000	0.525	0.435	0.547	0.481	0.468	0.355	0.304	0.372	0.153
PER5	0.494	0.731	0.355	0.395	0.554	0.448	0.606	0.448	1.000	0.480	0.548	0.157
PER5	0.494	0.731	0.355	0.395	0.554	0.448	0.606	0.448	1.000	0.480	0.548	0.157
PER7	1.000	0.773	0.531	0.336	0.467	0.479	0.490	0.403	0.494	0.283	0.439	0.152
PER7	1.000	0.773	0.531	0.336	0.467	0.479	0.490	0.403	0.494	0.283	0.439	0.152
PER9	0.336	0.678	0.525	1.000	0.394	0.317	0.412	0.300	0.395	0.259	0.330	0.095
PER9	0.336	0.678	0.525	1.000	0.394	0.317	0.412	0.300	0.395	0.259	0.330	0.095
PU1	0.452	0.511	0.403	0.346	0.937	0.220	0.678	0.678	0.459	0.692	0.706	0.308
PU2	0.362	0.489	0.414	0.348	0.929	0.216	0.677	0.670	0.461	0.742	0.765	0.277
PU3	0.469	0.531	0.350	0.349	0.904	0.244	0.717	0.589	0.543	0.658	0.697	0.275
PU4	0.441	0.579	0.431	0.405	0.912	0.276	0.696	0.665	0.579	0.715	0.843	0.227
UT1	0.467	0.527	0.384	0.249	0.743	0.272	0.604	0.617	0.554	0.689	0.894	0.190
UT2	0.361	0.441	0.307	0.297	0.762	0.141	0.593	0.594	0.509	0.699	0.933	0.197
UT3	0.418	0.483	0.314	0.334	0.729	0.213	0.559	0.637	0.499	0.592	0.911	0.252
UT4	0.327	0.411	0.275	0.335	0.645	0.171	0.560	0.563	0.409	0.461	0.816	0.226
UT5	0.376	0.450	0.366	0.268	0.759	0.176	0.565	0.683	0.459	0.660	0.897	0.216
use2	0.152	0.134	0.153	0.095	0.295	-0.086	0.330	0.310	0.157	0.137	0.242	1.000

Anexo D – Correlação entre Variáveis Latentes

	Agreab	CANOE	Consc	EmissStab	Enjoy	Extr	Flow	Int	Openness	PEOU	PU	Use
Agreab	1	0	0	0	0	0	0	0	0	0	0	0
CANOE	0.7726	1	0	0	0	0	0	0	0	0	0	0
Consc	0.5312	0.7981	1	0	0	0	0	0	0	0	0	0
EmissStab	0.3356	0.6782	0.5245	1	0	0	0	0	0	0	0	0
Enjoy	0.4671	0.573	0.4348	0.3935	1	0	0	0	0	0	0	0
Extr	0.4789	0.7403	0.5474	0.3171	0.2594	1	0	0	0	0	0	0
Flow	0.4899	0.6186	0.481	0.4121	0.7511	0.2863	1	0	0	0	0	0
Int	0.4029	0.5155	0.4684	0.3002	0.7077	0.2756	0.5523	1	0	0	0	0
Openness	0.4935	0.7313	0.3554	0.3954	0.5542	0.4477	0.6064	0.4482	1	0	0	0
PEOU	0.2831	0.3846	0.3041	0.2589	0.763	0.0742	0.6359	0.5768	0.4798	1	0	0
PU	0.4393	0.5201	0.3721	0.33	0.8186	0.219	0.6463	0.6958	0.5478	0.7021	1	0
Use	0.1524	0.1336	0.1533	0.0946	0.2947	-0.0859	0.3296	0.3104	0.1572	0.1371	0.2415	1

Anexo E – Modelo Estrutural



Anexo F – Total Effects

Total Effects (Mean, STDEV, T-Values)

	Original Sample (O)	Sample Mean (M)	Standard Deviation (STDEV)	Standard Error (STERR)
CANOE -> Agreeab	0.772615	0.771495	0.057018	0.057018
CANOE -> Consc	0.798081	0.796587	0.039811	0.039811
CANOE -> EmissStab	0.678157	0.674625	0.071719	0.071719
CANOE -> Enjoy	0.367670	0.360978	0.101283	0.101283
CANOE -> Extr	0.740252	0.738028	0.068822	0.068822
CANOE -> Int	0.337552	0.333819	0.089621	0.089621
CANOE -> Openness	0.731299	0.728036	0.055325	0.055325
CANOE -> PEOU	0.384640	0.381742	0.098904	0.098904
CANOE -> PU	0.520109	0.511863	0.103448	0.103448
CANOE -> Use	0.062329	0.060588	0.037120	0.037120
Enjoy -> Int	0.418756	0.414555	0.109191	0.109191
Enjoy -> Use	0.077323	0.074993	0.044819	0.044819
Flow -> Enjoy	0.331931	0.338036	0.082683	0.082683
Flow -> Int	0.138998	0.139356	0.048806	0.048806
Flow -> Use	0.253278	0.255391	0.101039	0.101039
Int -> Use	0.184650	0.180645	0.093632	0.093632
PEOU -> Enjoy	0.481755	0.476213	0.066666	0.066666
PEOU -> Int	0.409700	0.404488	0.054585	0.054585
PEOU -> PU	0.589161	0.588711	0.067607	0.067607
PEOU -> Use	0.075651	0.072721	0.038917	0.038917
PU -> Int	0.352981	0.352277	0.106112	0.106112
PU -> Use	0.065178	0.063385	0.039268	0.039268

Anexo G – Path Coefficients

Path Coefficients (Mean, STDEV, T-Values)

	Original Sample (O)	Sample Mean (M)	Standard Deviation (STDEV)	Standard Error (STERR)
CANOE -> Agreeab	0.772615	0.771495	0.057018	0.057018
CANOE -> Consc	0.798081	0.796587	0.039811	0.039811
CANOE -> EmissStab	0.678157	0.674625	0.071719	0.071719
CANOE -> Enjoy	0.182368	0.179824	0.082108	0.082108
CANOE -> Extr	0.740252	0.738028	0.068822	0.068822
CANOE -> Openness	0.731299	0.728036	0.055325	0.055325
CANOE -> PEOU	0.384640	0.381742	0.098904	0.098904
CANOE -> PU	0.293494	0.288127	0.086808	0.086808
Enjoy -> Int	0.418756	0.414555	0.109191	0.109191
Flow -> Enjoy	0.331931	0.338036	0.082683	0.082683
Flow -> Use	0.227612	0.230394	0.105300	0.105300
Int -> Use	0.184650	0.180645	0.093632	0.093632
PEOU -> Enjoy	0.481755	0.476213	0.066666	0.066666
PEOU -> PU	0.589161	0.588711	0.067607	0.067607
PU -> Int	0.352981	0.352277	0.106112	0.106112

Anexo H – Outer Weights

Outer Weights (Mean, STDEV, T-Values)

	Original Sample (O)	Sample Mean (M)	Standard Deviation (STDEV)	Standard Error (STERR)
BI1 <- Int	0.331512	0.330586	0.017367	0.017367
BI2 <- Int	0.382102	0.383347	0.020079	0.020079
BI3 <- Int	0.386026	0.387273	0.015447	0.015447
FLO2 <- Flow	0.200453	0.201354	0.017890	0.017890
FLO4 <- Flow	0.176990	0.177337	0.015365	0.015365
FLO5 <- Flow	0.145508	0.145310	0.015582	0.015582
FLO6 <- Flow	0.188428	0.188788	0.017106	0.017106
FLO7 <- Flow	0.188290	0.188783	0.016170	0.016170
FLO8 <- Flow	0.204825	0.203340	0.023189	0.023189
FLO9 <- Flow	0.223225	0.222909	0.022521	0.022521
FU2 <- PEOU	0.345390	0.345971	0.010354	0.010354
FU3 <- PEOU	0.322384	0.323958	0.015117	0.015117
FU4 <- PEOU	0.378715	0.376451	0.015503	0.015503
PER1 <- Extr	1.000000	1.000000	0.000000	
PER1 <- CANOE	0.244041	0.245153	0.028195	0.028195
PER3 <- Consc	1.000000	1.000000	0.000000	
PER3 <- CANOE	0.285885	0.287240	0.020741	0.020741
PER5 <- Openness	1.000000	1.000000	0.000000	
PER5 <- CANOE	0.285257	0.284132	0.019958	0.019958
PER7 <- Agreeab	1.000000	1.000000	0.000000	
PER7 <- CANOE	0.281341	0.281523	0.021631	0.021631
PER9 <- EmossStab	1.000000	1.000000	0.000000	
PER9 <- CANOE	0.243620	0.242847	0.021693	0.021693
PU1 <- Enjoy	0.273246	0.273986	0.010273	0.010273
PU2 <- Enjoy	0.276743	0.277435	0.009149	0.009149
PU3 <- Enjoy	0.258105	0.257763	0.008409	0.008409
PU4 <- Enjoy	0.277982	0.277996	0.010263	0.010263
UT1 <- PU	0.237925	0.238456	0.012891	0.012891
UT2 <- PU	0.229560	0.229971	0.011893	0.011893
UT3 <- PU	0.224566	0.225234	0.010538	0.010538
UT4 <- PU	0.188470	0.188574	0.019370	0.019370
UT5 <- PU	0.239429	0.240060	0.013597	0.013597
use2 <- Use	1.000000	1.000000	0.000000	

Anexo I – Outer Loadings

Outer Loadings (Mean, STDEV, T-Values)

	Original Sample (O)	Sample Mean (M)	Standard Deviation (STDEV)	Standard Error (STERR)
BI1 <- Int	0.894790	0.892463	0.031278	0.031278
BI2 <- Int	0.871876	0.870431	0.028174	0.028174
BI3 <- Int	0.959054	0.958432	0.010627	0.010627
FLO2 <- Flow	0.777785	0.776207	0.049493	0.049493
FLO4 <- Flow	0.786277	0.785073	0.039887	0.039887
FLO5 <- Flow	0.756532	0.751565	0.044287	0.044287
FLO6 <- Flow	0.762348	0.760529	0.050669	0.050669
FLO7 <- Flow	0.821278	0.819633	0.035361	0.035361
FLO8 <- Flow	0.679784	0.678688	0.058868	0.058868
FLO9 <- Flow	0.704774	0.706488	0.048312	0.048312
FU2 <- PEOU	0.961001	0.960688	0.009199	0.009199
FU3 <- PEOU	0.947585	0.947923	0.015524	0.015524
FU4 <- PEOU	0.957432	0.957714	0.009760	0.009760
PER1 <- Extr	1.000000	1.000000	0.000000	
PER1 <- CANOE	0.740252	0.738028	0.068822	0.068822
PER3 <- Consc	1.000000	1.000000	0.000000	
PER3 <- CANOE	0.798081	0.796587	0.039811	0.039811
PER5 <- Openness	1.000000	1.000000	0.000000	
PER5 <- CANOE	0.731299	0.728036	0.055325	0.055325
PER7 <- Agreeab	1.000000	1.000000	0.000000	
PER7 <- CANOE	0.772615	0.771495	0.057018	0.057018
PER9 <- EmissStab	1.000000	1.000000	0.000000	
PER9 <- CANOE	0.678157	0.674625	0.071719	0.071719
PU1 <- Enjoy	0.937196	0.936479	0.013356	0.013356
PU2 <- Enjoy	0.929058	0.929466	0.015341	0.015341
PU3 <- Enjoy	0.904413	0.901270	0.027763	0.027763
PU4 <- Enjoy	0.911466	0.910539	0.020698	0.020698
UT1 <- PU	0.893675	0.891327	0.026285	0.026285
UT2 <- PU	0.932850	0.931707	0.018678	0.018678
UT3 <- PU	0.911060	0.909492	0.031009	0.031009
UT4 <- PU	0.816151	0.812731	0.053939	0.053939
UT5 <- PU	0.897195	0.894029	0.026202	0.026202
use2 <- Use	1.000000	1.000000	0.000000	