



UNIVERSIDADE FEDERAL DO PARÁ
NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

FABRÍCIO DE SOUSA RIBEIRO

**AVALIAÇÃO DO IMPACTO DE AMBIENTES GAMIFICADOS NO
PROCESSO DE ENSINO-APRENDIZAGEM DA LÓGICA DE
PROGRAMAÇÃO DE COMPUTADORES: UMA COMPARAÇÃO ENTRE
ELEMENTOS MONOUSUÁRIO E MULTIUSUÁRIOS**

Tucuruí-Pará

2019

FABRÍCIO DE SOUSA RIBEIRO

**AVALIAÇÃO DO IMPACTO DE AMBIENTES GAMIFICADOS NO
PROCESSO DE ENSINO-APRENDIZAGEM DA LÓGICA DE
PROGRAMAÇÃO DE COMPUTADORES: UMA COMPARAÇÃO ENTRE
ELEMENTOS MONOUSUÁRIO E MULTIUSUÁRIOS**

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada do Núcleo de Desenvolvimento Amazônico em Engenharia, da Universidade Federal do Pará, como requisito para a obtenção do título de Mestre em Computação Aplicada.

Orientador: Bruno Merlin

Coorientador: Heleno Fülber

Tucuruí-Pará

2019

FABRÍCIO DE SOUSA RIBEIRO

**AVALIAÇÃO DO IMPACTO DE AMBIENTES GAMIFICADOS NO
PROCESSO DE ENSINO-APRENDIZAGEM DA LÓGICA DE
PROGRAMAÇÃO DE COMPUTADORES: UMA COMPARAÇÃO ENTRE
ELEMENTOS MONOUSUÁRIO E MULTIUSUÁRIOS**

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada do Núcleo de Desenvolvimento Amazônico em Engenharia, da Universidade Federal do Pará, como requisito para a obtenção do título de Mestre em Computação Aplicada

Orientador: Bruno Merlin

Coorientador: Heleno Fuber

Data da defesa: _____ de _____ de _____.

BANCA EXAMINADORA:

Prof. Dr. Bruno Merlin – UFPA - Orientador

Prof. Dr. Heleno Fülber – UFPA - Coorientador

Prof. Dr. Adonney Allan de Oliveira Veras – UFPA

Prof. Dr. Raphael W... – UFLA



Dedico este trabalho aos meus pais e esposa pelo apoio incondicional, pela paciência e compreensão nos momentos de minha ausência.

AGRADECIMENTOS

A realização desta dissertação só foi possível devido a colaboração de diversas pessoas, meus agradecimentos e estima.

Primeiramente a Deus, por ter-me concedido inspiração, saúde, determinação e coragem.

O meu eterno agradecimento à minha família e minha esposa, Miquela Ribeiro, pela paciência, compreensão, incentivo e apoio durante toda essa caminhada.

Ao meu orientador, Bruno Merlin, um agradecimento especial pela confiança, empenho e dedicação. Meu sincero muito obrigado pela parceria e orientação que possibilitou a conclusão dessa dissertação.

Ao meu coorientador, Heleno Fülber, pela parceria, orientação e disponibilidade, mesmo durante seu afastamento legal.

Aos professores Allan Veras e Viviane Santos pela paciência e orientação para realização desta dissertação.

Aos colaboradores da *startup* CodeCombat Inc, Nick Winter e Andrew Jakubowicz, pelo extraordinária colaboração durante a modificação da ferramenta e disponibilização das licenças para execução da pesquisa deste trabalho.

Aos meus amigos Victor da Cruz Peres e Luiz Fernando Reinoso por proporcionarem o ambiente propício para realização do experimento controlado.

Ao colegas do Instituto Federal de Educação do Pará, Campus Óbidos, pela compreensão e paciência durante as ausências para participação das atividades do programa de pós-graduação.

“Se você não puder se destacar pelo talento, vença pelo esforço.”

(Dave Weinbaum)

RESUMO

Nos cursos de Tecnologia que possuem em sua matriz curricular disciplinas relacionadas ao desenvolvimento de softwares (como Fundamentos de Programação), tem-se por parte dos discentes, dificuldades de aprendizagem do desenvolvimento de algoritmos. Conseqüentemente, faz-se necessário uma metodologia que fuja dos métodos tradicionais de ensino-aprendizagem, uma vez que os discentes precisam de estratégias diferenciadas que possam aumentar o engajamento destes. Nesse sentido, o uso de técnicas de gamificação apresenta-se como uma alternativa promissora frente aos resultados positivos de sua aplicação em várias áreas, inclusive em educação. Nesse cenário, os jogos digitais se destacam. Estes podem ser classificados em jogos monousuário e multiusuários. Diante do exposto, este trabalho tem como objetivo comparar o impacto do uso de ambientes gamificados no processo de ensino-aprendizagem da lógica de programação de computadores entre contexto de elementos monousuário e multiusuários. Os procedimentos de estudo foram divididos em três momentos: revisão sistemática da literatura, preparação do ambiente experimental e realização dos experimentos. O ambiente selecionado corresponde ao jogo Code Combat, com uma versão alterada para integrar os recursos multiusuários. Os experimentos foram realizados com 38 alunos de um curso técnico integrado ao Ensino Médio. Os dados sugerem um aumento do engajamento dos alunos que fizeram uso do ambiente gamificado que utiliza os recursos multiusuário, porém estatisticamente, não foram identificadas diferenças significativas de aprendizado entre o uso dos dois tipos de recursos.

Palavras-chave: Code Combat; Ensino-Aprendizagem; Programação de Computadores.

ABSTRACT

In Technology courses that have in their curriculum matrix disciplines related to software development (such as Fundamentals of Programming), students have learning difficulties in the development of algorithms. Consequently, a methodology that is beyond the traditional teaching-learning methods is necessary, as students need different strategies that can increase their engagement. In this sense, the use of gamification techniques is a promising alternative to the positive results of their application in various areas, including education. In this scenario, digital games stand out. These can be classified into single and multiuser games. Given the above, this paper aims to compare the impact of the use of gamified environments in the teaching-learning process of computer programming logic between single-user and multi-user context. The study procedures were divided into three moments: systematic literature review, preparation of the experimental environment and performance of the experiments. The selected environment corresponds to the Code Combat game, with a modified version to integrate the multiuser features. The experiments were carried out with 38 students from a technical course integrated with high school. The data suggest an increased engagement of students who made use of the gamified environment that uses multiuser resources, but statistically, no significant learning differences were identified between the use of both types of resources.

Keywords: Code Combat; Teaching-Learning; Computer programming.

LISTA DE ILUSTRAÇÕES

Figura 1 - Estudos classificados pelo nível de Ensino.....	32
Figura 2 - Tipos de estudos identificados.....	32
Figura 3 - Metodologia dos estudos primários.....	33
Figura 4 - Distribuição temporal dos estudos.	34
Figura 5 - Técnicas utilizadas.....	35
Figura 6 - Escolha dos mundos disponíveis.....	38
Figura 7 – Mapa Kithgard (CS1).....	39
Figura 8 - Descrição da missão.....	40
Figura 9 - Escolha do <i>avatar</i> (herói).	40
Figura 10 - Orientação para solução da missão.....	41
Figura 11 - Tela de missão do Code Combat.....	41
Figura 12 – Code Combat	50
Figura 13 - Ranking dos cursos do G2.....	59
Figura 14 - Ranking da arena do CS1.....	60
Figura 15 - Resultados sobre o tempo médio de acesso ao jogo.....	61
Figura 16 - Resultado sobre a quantidade média de missões concluídas.	62
Figura 17 - Participação nos encontros presenciais do minicurso.....	62
Figura 18 - Resultado sobre o desempenho médio dos grupos.	63
Figura 19 - Média de pontos obtidos pelo questionário motivacional IMI.	63

LISTA DE QUADROS

Quadro 1 - Padrões de mecânica de jogos	47
Quadro 2 - Elementos de jogos selecionados de acordo com os padrões relacionados	48
Quadro 3 - Elementos de jogos do Robocode e Code Combat.....	49

LISTA DE TABELAS

Tabela 1 - Questões de Pesquisa	27
Tabela 2 - String de Busca.....	28
Tabela 3 - Critérios de inclusão e exclusão.....	29
Tabela 4 - Resultado geral das buscas.	30
Tabela 5 - Ferramentas utilizadas.	34
Tabela 6 - Informações sobre os mundos/cursos do Code Combat.....	37
Tabela 7 - Valores obtidos no teste t de Student.....	64

SUMÁRIO

1	INTRODUÇÃO	14
1.1	CONTEXTUALIZAÇÃO E MOTIVAÇÃO	14
1.2	OBJETIVOS	17
1.3	ESTRUTURA DA DISSERTAÇÃO.....	18
2	FUNDAMENTAÇÃO TEÓRICA E REVISÃO SISTEMÁTICA DA LITERATURA	19
2.1	ENSINO-APRENDIZAGEM.....	19
2.2	DESAFIOS DO ENSINO-APRENDIZAGEM DA LÓGICA DE PROGRAMAÇÃO DE COMPUTADORES	20
2.2.1	Dificuldades para os docentes	21
2.2.2	Dificuldades para os alunos	22
2.2.3	Motivação	23
2.3	GAMIFICAÇÃO	24
2.4	METODOLOGIAS GAMIFICADAS APLICADAS A APRENDIZAGEM DA LÓGICA DE PROGRAMAÇÃO DE COMPUTADORES: UM REVISÃO SISTEMÁTICA DA LITERATURA	26
2.4.1	Trabalho Relacionados	26
2.4.2	Método Utilizado	27
2.4.2.1	Questões da Pesquisa	27
2.4.2.2	String de busca	28
2.4.2.3	CrITÉrios de incluso e excluso.....	28
2.4.2.4	Extrao e sntese de dados	29
2.4.3	Anlise	30
2.4.4	Resultados Gerais	30
2.4.4.1	Reposta a Questo de Pesquisa (QP1)	30
2.4.4.2	Reposta a Questo de Pesquisa (QP2)	31
2.4.4.3	Reposta a Questo de Pesquisa (QP3)	31
2.4.4.4	Reposta a Questo de Pesquisa (QP4)	32
2.4.4.5	Reposta a Questo de Pesquisa (QP5)	33
2.4.4.6	Distribuio Temporal.....	33
2.4.4.7	Ferramentas e Tcnicas utilizadas	34

2.4.5	Considerações Finais	35
2.5	CODE COMBAT.....	36
3	PROCEDIMENTOS E MÉTODOS	43
3.1	TIPO DE PESQUISA.....	43
3.2	MÉTODOS CIENTÍFICOS.....	43
3.3	ABORDAGEM DE PESQUISA.....	44
3.4	HIPÓTESES DE PESQUISA.....	44
3.5	VARIÁVEIS	44
3.6	CARACTERIZAÇÃO DA AMOSTRA.....	45
3.7	PROCEDIMENTOS.....	45
3.7.1	Revisão da Literatura	45
3.7.2	Preparação do ambiente de experimentação	46
3.7.2.1	Instrumentos para aferir o engajamento.....	46
3.7.2.2	Instrumentos para aferir o desempenho.....	46
3.7.2.3	Instrumentos para aferir a motivação	46
3.7.2.4	Definição dos elementos do ambiente gamificado	47
3.7.2.5	Criação do ambiente de experimentação	48
3.7.3	Experimento Controlado	51
3.8	AVALIAÇÃO	52
3.8.1	Experimentação	52
3.8.2	Teste de retenção de conhecimento	53
3.8.3	Teste motivacional	54
3.9	TÉCNICA DE ANÁLISE DE DADOS.....	54
3.9.1	Análise qualitativa	54
3.9.2	Análise quantitativa	54
4	RESULTADOS	56
4.1	MODIFICAÇÃO DO CODE COMBAT	56
4.1.1	Instalação do Code Combat	56
4.1.2	Execução do Code Combat	57
4.1.3	Programação do Code Combat	58
4.1.3	Parcerias consolidadas	60
4.2	ANÁLISE DOS DADOS.....	61
4.3	TESTE DAS HIPÓTESES.....	64
4.4	DISCUSSÃO DOS RESULTADOS	65

5	CONCLUSÕES.....	68
	REFERÊNCIA BIBLIOGRÁFICAS	70
	APÊNDICE A – ARTIGOS INCLUÍDOS NA REVISÃO SISTEMÁTICA	75
	APÊNDICE B – INTRINSIC MOTIVATION INVENTORY – IMI	
	(PORTUGUÊS).....	79
	APÊNDICE C – TESTE DE RETENÇÃO DE CONHECIMENTO	80
	ANEXO A – GUIA SOBRE OS CONCEITOS ABORDADOS NO	
	PRIMEIRO MAPA	83
	ANEXO B – GUIA SOBRE OS CONCEITOS ABORDADOS NO	
	SEGUNDO MAPA	87

1 INTRODUÇÃO

Matérias que tratam sobre Fundamentos de Programação são pertinentes para o desenvolvimento profissional dos acadêmicos que possuem este componente curricular em seu curso. Todavia, esta disciplina vem sendo motivo de debate sobre os desafios enfrentados por docentes e discentes durante o ensino-aprendizagem. Debates e estudos são consideravelmente enriquecedores para aprimorar a abordagem/metodologia utilizada em sala de aula com o fim de aumentar os níveis de sucesso. Neste sentido, este estudo aborda como temática a avaliação da gamificação como uma alternativa às abordagens já estudadas.

1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO

Nos cursos de Tecnologia que possuem em sua matriz curricular disciplinas relacionadas ao desenvolvimento de softwares, por exemplo Fundamentos de Programação, tem-se por parte dos alunos, dificuldades de aprendizagem no desenvolvimento de algoritmos.

Historicamente, as disciplinas introdutórias à lógica de programação de computadores são desafiadoras para a maioria dos alunos. Garlet, Bigolin e Silveira (2018) afirmam que, pelo fato do ensino da programação não ser fácil, é comum observarmos pesquisas que apontam o grande número de evasões nestes cursos, fato que tem relação com as dificuldades de aprendizagem.

Conforme Bittencourt et al. (2013), além de outras demandas, a aprendizagem de programação exige capacidade de abstração, raciocínio lógico e o domínio das noções de álgebra abstrata, sintaxe e semântica das linguagens. Por outro lado, há ainda a questão da motivação para a aprendizagem de programação. Para os autores, o uso de metodologias de ensino-aprendizagem excessivamente formais agrava essa situação.

A falta de motivação decorre também devido à conotação negativa que está associada à disciplina, passada de aluno para aluno. Adicionalmente, as disciplinas de programação adquirem a reputação de serem difíceis. Conseqüentemente isto influencia na taxa de insucesso dos estudantes.

Por esta razão, segundo Giraffa & Moraes (2013), estudos brasileiros e internacionais apontam para a necessidade de criar estratégias de incentivo à permanência e êxito dos alunos em disciplinas de algoritmos e programação, logo no início dos cursos de computação. Para os autores, a formação deficitária na formação básica dos estudantes tem sido apontada como fator impactantes e relacionado com o cancelamento ou abandono nessas disciplinas.

Desta maneira, faz-se necessário uma metodologia que fuja dos métodos tradicionais de ensino-aprendizagem, de modo que venha aumentar a produtividade do aluno durante a aprendizagem da construção de algoritmos, uma vez que o uso de estratégias diferenciadas podem promover a integração entre os discentes. Procura-se o aumento do engajamento destes em relação ao tema e, conseqüentemente, a diminuição dos índices de evasão/reprovação.

Nesse sentido, o uso de técnicas de gamificação apresentam-se como uma alternativa promissora frente aos resultados positivos de sua aplicação em várias áreas, inclusive em educação (SHELDON, 2012). A gamificação, segundo Santos (2018), é uma técnica que faz uso de mecânicas, estética e pensamentos dos jogos para envolver as pessoas; motivá-las a executar determinadas ações e promover a aprendizagem para resolução de problemas. Para Kapp (2012), a gamificação é a utilização de mecânica, estética e pensamento baseado em games para engajar pessoas, motivar a ação, promover a aprendizagem e resolver problemas.

Estar baseado em *games* implica na construção de um sistema na qual aprendizes, jogadores ou consumidores se engajarão em um desafio abstrato, definido por regras claras, interagindo e aceitando *feedback* com o alcance de resultados quantificáveis e com a presença de reações emocionais. (ALVES, 2015, p. 27).

Alves (2015) ainda relata que 97% dos jovens jogam no computador e também em *vídeo games*, e que atividades divertidas e gamificadas podem ser capazes de engajar diferentes públicos e gerações. Para a autora, a geração *Millenium* se engaja, mas “precisa perceber que o que faz é relevante, deseja ter *feedback* constante para saber se está indo bem”. O local de trabalho, a sala de aula e o modo como se trabalha mudou. Para a geração dos *gamers*, não há uma diferença tão marcante entre trabalho, aprendizado e diversão (MATTAR, 2010).

O que funcionava antes não necessariamente funciona hoje quando o assunto é aprendizagem. É este cenário que a gamificação se encaixa. Ajudando-nos a tornar a aprendizagem atrativa, engajadora, divertida e efetiva. (ALVES, 2015, p. XXII).

O engajamento é fundamental na gamificação. Afinal, segundo Alves (2015), trata-se da meta explícita dos sistemas gamificados, principalmente quando o assunto é aprendizagem em uma época em que facilitadores, professores e palestrantes disputam a atenção de seus aprendizes com a tecnologia. A autora ainda relata que atividades divertidas e gamificadas podem ser capazes de engajar diferentes públicos e gerações. A diversão é um elemento de extrema importância e faz com que o usuário tenha o interesse e prazer de jogar. Deste modo pode-se inferir que a gamificação pode ser aplicada no processo de ensino-aprendizagem da lógica de programação de computadores.

Observando os relatos de Alves (2015) e Mattar (2010), pode-se inferir que a geração atual de estudantes pode ser caracterizada pela baixa tolerância a falhas, necessidades imediatistas e não diferenciação de trabalho e diversão, a gamificação pode ser um trunfo baseado no contexto em que os sujeitos estão envolvidos.

Os jogos digitais vêm sendo cada vez mais utilizados para implementar as técnicas de gamificação. Segundo Peixoto, Santos e Resende (2015), a utilização de jogos educacionais é uma das formas de mudança de paradigma de ensino: do aprendizado ao escutar (*learning by listen*) para o aprendizado ao se fazer (*learning by doing*). Uma razão para sua adoção é que existem evidências empíricas que indicam que os jogos educacionais são ferramentas efetivas para a melhoria do aprendizado e compreensão de assuntos complexos (PEIXOTO et al., 2014). Pode-se deduzir que estes jogos permitem aumentar o interesse dos estudantes nos assuntos abordados em sala de aula.

Dentre as diversas dimensões, os jogos podem ser classificados em jogos para um jogador (monousuário), em que a execução prevê um único usuário como jogador, e os jogos para múltiplos usuários (multiusuários), em que a execução prevê mais de um usuário como participante do tema desenvolvido no jogo. A utilização desses jogos, tanto aqueles com foco educacional quanto aqueles de entretenimento, apresentam uma grande expansão no cenário mundial. Os jogos multiusuários permitem a exploração de aspectos que não poderiam ser cobertos em jogos com um

único usuário como, por exemplo, a colaboração, a sociabilidade e a competição (PEIXOTO, SANTOS e RESENDE, 2015).

Todos os estudos encontrados durante a realização da revisão sistemática da literatura (seção 2.4) declararam que estes jogos foram eficazes no cumprimento dos objetivos propostos, conforme apresentado na seção 2.4.4. Os resultados dessa revisão também apontam a inexistência de estudos (nos veículos pesquisados) que tratam sobre o contexto de elementos monousuário e multiusuário voltados ao ensino-aprendizagem da lógica de programação de computadores. É importante frisar que esses estudos analisados tiveram como alicerce de suas afirmações dados extraídos de ambientes gamificados que possuem, prevalentemente, elementos monousuário. Observando estes resultados já consolidados na literatura e os aspectos comportamentais que podem ser alcançados através dos jogos multiusuários, de acordo com Peixoto, Santos e Resende (2015), se faz necessário comparar estes elementos monousuário e multiusuário com o fim investigar se algum destes produz resultados significativamente melhores no contexto do ensino-aprendizagem da lógica de programação de computadores.

1.2 OBJETIVOS

Diante do exposto, este trabalho tem como objetivo geral avaliar o impacto de ambientes gamificados no processo de ensino-aprendizagem da lógica de programação de computadores através da comparação entre os elementos monousuário e multiusuários. Para garantir que este objetivo fosse alcançado, tiveram-se os seguintes objetivos específicos:

- Realizar uma revisão sistemática da literatura sobre estudos que abordam o ensino-aprendizagem da lógica de programação;
- Desenvolver ou modificar uma ferramenta para estudo do impacto de ambientes gamificados multiusuários no ensino-aprendizagem da lógica de programação de computadores;
- Planejar e conduzir experimentos empíricos abordando o uso da ferramenta desenvolvida/modificada;
- Analisar os dados coletados.

A questão principal que norteou o trabalho é: “Quais são as evidências que o uso de ambientes gamificados multiusuários são significativamente melhores que ambientes gamificados monousuário com a finalidade de aumentar o engajamento e motivação do discente no processo de ensino-aprendizagem da lógica de programação de computadores?”. Vale observar que ambientes gamificados multiusuários além de implementarem seus próprios recursos, geralmente também fazem uso dos recursos presentes dentro dos ambientes monousuário.

1.3 ESTRUTURA DA DISSERTAÇÃO

Esta Dissertação está organizada da seguinte forma: o Capítulo 1 introduz uma síntese dos principais tópicos ao longo desta dissertação; o Capítulo 2 faz um apanhado sobre a fundamentação teórica associado ao estudo, abordando sobre conceitos com ensino-aprendizagem, desafios do ensino da lógica de programação, conceitos e informações sobre gamificação e uma revisão sistemática da literatura sobre ensino-aprendizagem da lógica de Programação de Computadores; no Capítulo 3, explica-se a metodologia utilizada; o Capítulo 4 mostra os resultados deste estudo; e Capítulo 5 contém as considerações finais sobre o trabalho.

2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO SISTEMÁTICA DA LITERATURA

Neste capítulo, faz-se uma abordagem dos conteúdos associados ao tema do trabalho: ensino-aprendizagem, trata sobre conceitos relacionados a este tópico, que por sua vez, auxilia na compreensão dos desafios de conteúdo no contexto de programação de computadores abordado na seção seguinte; desafios do ensino-aprendizagem da lógica de programação de computadores, faz uma abordagem das dificuldades no ensino da lógica de programação na perspectiva dos docentes e discentes, tratando também sobre aspectos secundário que interferem na efetividade deste processo, como motivação e infraestrutura; gamificação, versa sobre uma das soluções utilizada para superar os desafios no ensino-aprendizagem da lógica de programação de computadores; Revisão Sistemática da Literatura, aborda sobre estudos que versão sobre a gamificação aplicada ao ensino-aprendizagem da lógica de programação de computadores; Code Combat, que trata sobre um ambiente utilizado no contexto da aplicação da gamificação no processo de ensino-aprendizagem da lógica de programação de computadores.

2.1 ENSINO-APRENDIZAGEM

Segundo Altet (1999), o ensino é um processo interpessoal, intencional, que usa essencialmente a comunicação, a situação pedagógica utilizada pelo professor como meio de provocar, favorecer, fazer alcançar a aprendizagem de um saber ou de um saber-fazer. Dessa forma, aprendizagem é interpretada como um processo de aquisição e/ou mudança.

De acordo com Castro e Souza (2016), o ato de adquirir conhecimento, informação e habilidades através de ensino, estudo ou até vivência no dia a dia é denominado aprendizagem. Para Duarte et al. (2015), a aprendizagem é um processo complexo que envolve a combinação da motivação do indivíduo para estudar e as estratégias utilizadas por ele para operacionalizar este objetivo. O sucesso e a qualidade dos seus resultados da aprendizagem são dependentes de fatores como os cognitivos, afetivos, interpessoais, escolares, familiares e sociais, sendo necessário identificar e compreender tais fatores para alcançarmos a sua melhoria (FONTES, 2016).

De acordo com Santos (2018, p 8 *apud* KREPSK, 2004), o processo de aprendizagem é permanente e tem como principal objetivo o conhecimento. O conhecimento se dá enquanto o indivíduo se relaciona com o mundo, nos planos biológico e sociocultural. O aprendizado é fundamental para que se desenvolvam e amadureçam as funções cognitivas do indivíduo e, conseqüentemente, para que este se integre ao seu grupo social.

No âmbito pedagógico, a aprendizagem é abordada de diversas formas onde cada uma possui suas peculiaridades. Na aprendizagem receptiva, o indivíduo é capaz de receber uma instrução e compreendê-la; é possível reproduzir o que lhe foi passado, mas é incapaz de descobrir algo. Na aprendizagem por descoberta, exige do sujeito trabalho em troca do aprendizado; nada é fornecido facilmente. Outra aprendizagem que merece destaque é a repetitiva. Nesta há a memorização do conteúdo, embora a compreensão inicial seja um pouco confusa. Por último, a aprendizagem significativa na qual a pessoa consegue relacionar os seus conhecimentos prévios com novas informações passadas (CASTRO e SOUZA, 2016).

Pode-se pensar ainda em duas correntes em psicologia diretamente ligadas a aprendizagem, sendo uma delas o Cognitivismo e a outra Behaviorismo. O Behaviorismo fala sobre o que as pessoas fazem sob o ponto de vista externo, enquanto o Cognitivismo está relacionado com estados mentais, preocupando-se com o que acontece internamente (ALVES, 2015).

A escolha da abordagem de aprendizagem é definida pelo docente durante a elaboração e/ou execução do seu planejamento de aula. Afinal, é neste documento que o professor expressa sua estratégia “acerca dos objetivos a serem alcançados pelos alunos, conteúdo programático adequado para o alcance dos objetivos, estratégias e recursos que vai adotar para facilitar a aprendizagem, etc.” (GIL, 2012, p. 34).

2.2 DESAFIOS DO ENSINO-APRENDIZAGEM DA LÓGICA DE PROGRAMAÇÃO DE COMPUTADORES

Conforme Neto e Schuvartz (2007), acadêmicos iniciantes, ao se depararem com a disciplina, sentem-se incapazes de programar devido ao conjunto de habilidades que a programação exige.

O documento *New Horizon Report: 2015 k-12 Edition* traz, como um dos principais desafios para escolas de todo o mundo, o ensino do pensamento complexo:

É essencial para os jovens tanto entender o mundo conectado em que eles estão crescendo e também – através do pensamento complexo – aprender a usar abstração e decomposição para resolver tarefas complexas (...) Ensinar programação nas escolas está cada vez mais sendo visto como uma forma de instilar essa forma de pensamento em estudantes, uma vez que combina profundo conhecimento da ciência da computação com a criatividade e resolução de problemas (JOHNSON et al., 2015, p. 32).

Em razão disso que muitas escolas, principalmente fora do Brasil, além do conteúdo regular de ensino, possuem em sua grade curricular disciplinas que tratam sobre a aprendizagem de programação de computadores. Segundo Anjos, Duda e Silva (2016), muitas escolas já estão se adaptando a essa nova realidade, o autor cita como exemplo 160 mil escolas primárias da Inglaterra que estão tendo aulas de programação. Isso eleva ainda mais a sua importância, uma vez que se pode inferir sua importância na formação dos estudantes, de qualquer área de conhecimento, em razão das habilidades mentais que advém de tal aprendizagem.

2.2.1 Dificuldades para os docentes

Sentance (2016) lista algumas das principais dificuldades em lecionar uma disciplina de programação de computadores enfrentadas pelos professores:

- Receio de não possuir conhecimento suficiente para ajudar os alunos a resolverem a grande variedade de problemas com que podem se deparar;
- A dificuldade dos alunos em compreender o assunto;
- Problemas técnicos da escola;
- Necessidade de diferenciação para atender a diferentes níveis de habilidade;
- Falta de habilidade para resolver problemas;

Pode-se adicionar a esta lista a definição e adaptação das estratégias de ensino de acordo com as dificuldades observadas na turma durante a proposição dos problemas.

2.2.2 Dificuldades para os alunos

Segundo Gomes e Taumaturgo (2017), os desafios mais comuns enfrentados pelos discentes se encontram nas dificuldades de abstração e resolução de problemas. Contudo, isso ocorre em razão de uma série de dificuldades sentidas pelos alunos, isto é potencializado pelo fato de a disciplina que aborda o conteúdo de programação de computadores é, na maioria dos casos, uma disciplina nova, uma vez que os discentes não estão habituados a disciplinas dessa natureza, diferente de outras disciplinas como Matemática, Português, Biologia e outras que eles têm contato ainda no ensino fundamental.

Existem cinco dificuldades na aprendizagem de programação (BIGGS, 2011):

- Orientação: entender os benefícios e utilidades de saber programar;
- Máquina Notacional: entender os aspectos da máquina em que se programa que afetam o funcionamento do programa;
- Notação: entender as especificidades da sintaxe da linguagem utilizada, ou seja, como escrever os comandos desejados;
- Estruturas: entender como usar estruturas de repetição e de desvio condicional, por exemplo;
- Dominar os pragmatismos da programação: entender como usar ferramentas de escrita, execução, testes e depuração de códigos de programas;

Pea (1986) relata sobre o que chama de “*bugs*” conceituais em programadores iniciantes:

- Paralelismo: presunção de que diferentes linhas de código do programa podem estar ativas ao mesmo tempo;
- Intencionalidade: os estudantes acreditam que o programa consegue ir além das instruções dadas, como se elas fossem dicas do que deve ser feito, e o computador consegue preencher as lacunas e chegar ao resultado esperado pelo aluno;
- Egocentrismo: os estudantes acreditam que o computador consegue entender a intenção do que eles querem fazer, e que, de alguma forma,

executará a tarefa mesmo sem que tenham explicitado no código do programa.

Outra dificuldade descrita por Anjos, Duda e Silva (2016) é que, segundo eles, os alunos frequentemente conseguem encontrar uma solução mental para os problemas em questão, mas não conseguem “traduzir” a solução para um programa, na forma de instruções que o computador consiga compreender e executar. A impaciência ao analisar a solução desenvolvida também é uma dificuldade encontrada pelos discentes.

2.2.3 Motivação

Uma forma de definir a motivação é compará-la com um estado interno que ativa, orienta e mantém o comportamento, composto por cinco dimensões: a do conteúdo do comportamento, a da velocidade da iniciação deste comportamento, a da intensidade do envolvimento, a da experiência cognitiva e emocional durante o envolvimento e a do que causa o comportamento (DUARTE, 2002).

No aspecto pedagógico é possível identificar alguns fatores dos quais a motivação dos alunos possa se originar (PEA, 1986):

- Externa: baseada em uma recompensa futura, geralmente de cunho financeiro;
- Interna: o interesse pela disciplina advém do interesse pelo conteúdo em si;
- Social: desejo de agradar a um grupo de pessoas ou uma determinada pessoa;
- Conquista: a motivação vem da sensação de estar apresentando um bom desempenho;
- Inconclusivo: não se pode chegar a uma conclusão da origem da motivação;

Como cada indivíduo possui suas próprias motivações, é desafiador para o professor realizar o planejamento de sua aula adequado a todos os indivíduos. A motivação é uma característica fundamental no processo de aprendizagem, pois a intensidade e a qualidade do envolvimento necessário para aprender dependem dela.

O aprendizado sério precisa se aproximar do entretenimento para conseguir melhor engajar os alunos.

No contexto da gamificação, Alves (2015) relata que as pessoas jogam, envolvem-se e dedicam seu tempo a esta atividade em busca de emoções positivas e diversão. Desta forma, a motivação é um ponto de extrema importância quando o assunto é gamificação e aprendizagem. Para a autora, em um jogo ou brincadeira estamos totalmente engajados no momento, espaço e tempo de jogo. Jogar ou brincar tem um “gostinho de quero mais”, ou seja, traz em si o desejo de continuação. Se algo ameaçar estragar a diversão, inventamos uma maneira de continuar (ALVES, 2015, p. 93).

2.3 GAMIFICAÇÃO

A gamificação é uma metodologia ou técnica que faz uso de conceitos associados aos jogos para melhorar a experiência e o engajamento do usuário em diferentes atividades ou aplicações que não tem como propósito principal apenas o entretenimento ou diversão (DETERDING et al. 2011). Kapp (2007) define gamificação de forma semelhante, mas enfatiza o uso da mecânica, estética e raciocínio correntes nos jogos, como forma de engajar as pessoas; conduzi-las na solução de problemas e promover os processos de aprendizagem.

De acordo com Liu, Alexandrova e Nakajima (2011), para usar a gamificação em um ambiente não relacionado aos jogos, a ideia é criar-se um ciclo de gamificação. A interação inicia a partir de um objetivo ou desafio claro com uma condição de vitória específica. Toda vez que o usuário alcançar uma determinada meta, algumas recompensas são dadas nesse sentido, o que é normalmente suportado por um sistema de pontos (pontuação, moeda virtual, pontos de experiência etc.). Com base no sistema de pontos e históricos de conquistas, um quadro de líderes (globais ou parciais) e emblemas são fornecidos aos jogadores para motivar a colaboração e a competição (FRANÇA e REATEGUIM, 2013).

A maior parte das aplicações que empregam conceitos de gamificação estabelece alguma forma de recompensa pelas atividades realizadas pelo usuário. Existe uma série de elementos de design de jogos que podem ser incorporados: avatares, recursos colecionáveis, recompensas, níveis (sistema de progressão),

desbloqueio de conteúdo, combates e outros. No entanto, segundo Werbach e Hunter (2012), os elementos de jogos mais utilizados correspondem ao trio PBL (do acrônimo em inglês PBL – *points, badges, leaderboards*): Pontos (pontuação que o jogador recebe ao executar determinadas ações ou comportamentos incentivados pelo processo), Medalhas (marcos ou conquistas que são dados ao jogador como uma forma de reconhecer que ele atingiu um patamar ou status desejado), Ranking de Líderes (classificação do jogador em relação aos outros jogadores envolvidos no processo usada como métrica de comparação de sua performance).

Segundo Gomes e Tedesco (2017, p. 2, apud SCHLEMMER e LOPES, 2016), o uso dos componentes PBL torna a estruturação da gamificação mais fácil, escalável e de baixo custo. Contudo, a sua utilização de maneira descontextualizada pode ser problemática por concentrar-se em uma camada de motivação extrínseca que pode funcionar de maneira adequada para os alunos mais jovens, porém não observa-se o mesmo para os alunos mais velhos, sobretudo quando consideram-se experiências de longo prazo.

Assim, se faz necessário definir uma estratégia de gamificação baseado no público alvo, contexto, ambiente e problemas a serem resolvidos a fim de estruturar desafios e missões condizentes com os objetivos pessoais dos jogadores (SCHLEMMER e LOPES, 2016).

No contexto de jogos digitais, Santos (2018) relata em seu estudo que a gamificação não significa somente recompensar alguém pela tarefa realizada, mas utilizar o design e as técnicas próprias de jogos, que tanto cativam as pessoas, para enriquecer o processo de ensino-aprendizagem, promovendo um ambiente de aprendizagem atraente e desafiador, capaz de estimular os alunos em sua busca pelo conhecimento.

Contudo, pode-se inferir que para a aplicação de estratégias gamificadas não se faz obrigatório o uso de jogos digitais. Afinal, qualquer recurso, seja virtual ou não, que faça a implementação da PBL e tenha objetivos relacionados aos conceitos de gamificação, pode ser considerado gamificação.

2.4 METODOLOGIAS GAMIFICADAS APLICADAS A APRENDIZAGEM DA LÓGICA DE PROGRAMAÇÃO DE COMPUTADORES: UM REVISÃO SISTEMÁTICA DA LITERATURA

Esta seção trata de uma Revisão Sistemática da Literatura (RSL), com a finalidade de identificar as metodologias que fazem uso da gamificação na aprendizagem da lógica de programação de computadores, assim como também de investigar a eficácia da aplicação da gamificação. A RSL foi utilizada objetivando extrair, analisar, sintetizar e catalogar dados de diversos periódicos nacionais, especificamente portal de publicações da Comissão Especial de Informática na Educação (CEIE)¹, e internacionais, da *IEEE Xplorer Digital Library*² e *ACM Digital Library*³. O CEIE contempla as principais publicações nacionais no âmbito da informática na educação, como o Simpósio Brasileiro de Informática na Educação (SBIE), Workshop de Informática na Escola (WEI), Workshop do Congresso Brasileiro de Informática na Educação (WCBIE), Revista Brasileira de Informática na Educação (RBIE), Jornada de Atualização em Informática na Educação (JAIE) e Workshop de Desafios da Computação Aplicada à Educação (DESAFIE). Já no âmbito internacional o IEEE Xplorer e ACM são as principais fontes na área.

2.4.1 Trabalho Relacionados

Aureliano e Tedesco (2012) desenvolveram e publicaram um estudo sobre o ensino-aprendizagem da lógica de programação para iniciantes cujo objetivo foi fornecer uma análise abrangente e sistemática dos trabalhos científicos publicados no Simpósio Brasileiro de Informática na Educação (SBIE) e no Workshop de Informática na Escola (WIE). Para isso, utilizaram uma metodologia de RSL visando extrair, catalogar, analisar e sintetizar dados de todos os artigos publicados em 10 edições (2002-2011) desses eventos.

Medeiros, Silva e Aranha (2013), semelhantemente, apresentou uma RSL para o ensino e aprendizagem da lógica de programação com jogos digitais no contexto nacional. Este estudo desenvolveu um processo similar ao apresentado por Medeiros, Silva e Aranha (2013), aplicando a técnica de pesquisa secundária para

¹ CIEE - <http://www.br-ie.org/pub/index.php/>

² IEEE Xplorer - <https://www.ieee.org/>

³ ACM - <https://dl.acm.org/>

realizar uma análise abrangente e sistemática da pesquisa em jogos digitais para o ensino e aprendizagem da lógica de programação no cenário internacional.

Outro estudo que fez uso da RSL foi a pesquisa de Silva, Medeiros e Aranha (2014). No trabalho produzido, analisaram 29 estudos primários com o objetivo de investigar a eficácia dos jogos digitais para o ensino e aprendizagem da lógica de programação.

2.4.2 Método Utilizado

Para Galvão e Pereira (2014), são considerados estudos secundários os que têm nos estudos primários sua fonte de dados, portanto, trata-se de um tipo de investigação focada em questão definida, que visa identificar, selecionar, avaliar e sintetizar as evidências relevantes disponíveis. Esta revisão teve como objetivo delimitar, a partir da análise de estudo primários, a eficácia da gamificação como ferramenta de auxílio na aprendizagem da lógica de programação de computadores, e também fazer uma análise do panorama experiências empíricas neste seguimento no contexto nacional entre os anos de 2013 a 2017.

2.4.2.1 Questões da Pesquisa

Para alcançar os objetivos, esta RSL tem como questões da pesquisa a seguinte pergunta: Qual o panorama atual das publicações científicas que tratam do processo de ensino-aprendizagem da lógica de programação de computadores por meio da gamificação? Para responder esta questão foram definidas as questões apresentadas na Tabela 1. A primeira coluna apresenta o código da questão da pesquisa e a segunda a descrição da mesma.

Tabela 1 - Questões de Pesquisa

Questões	Descrição
QP1	Qual é o número de participantes dos estudos que estão sendo realizados por pesquisadores?
QP2	Os estudos reportados indicam que o uso de jogos para ensinar a lógica de programação é eficaz?
QP3	Quais os níveis de escolaridade estão sendo destinados os estudos?

QP4	Que tipos de estudos estão sendo realizados por pesquisadores que investigam o ensino da lógica de programação utilizando jogos?
QP5	Que metodologias estão sendo utilizadas na aplicação da gamificação na aprendizagem da lógica de programação de computadores.

2.4.2.2 String de busca

As palavras chaves utilizadas nas *strings* de busca para obtenção dos estudos primários desta revisão são: *Gamificação, Jogos, Aprendizagem de Programação, Lógica de Programação e Algoritmos*. A partir desse conjunto de palavras, foram geradas as *strings* utilizadas da RSL (Tabela 2). Para maximizar e aprimorar os estudos encontrados na busca foi utilizado o sinônimo de algumas palavras e realizado adaptações na *string* das fontes internacionais.

Tabela 2 - String de Busca

String 1	CEIE					
	RBIE	SBIE	WCBIE	WIE	JAIE	DESAFIE
<i>(gamif* OR jogo* OR game) AND (educa* OR motiv* OR ensino OR aprendizagem) AND (lógica OR programação OR algoritmo)</i>	7	25	23	28	1	0
String 2	IEEE Xplorer			ACM		
<i>game AND teaching AND programming</i>	494			476		

2.4.2.3 Critérios de inclusão e exclusão

Para a definição dos estudos primários inicialmente foi realizada a avaliação dos 1.054 estudos encontrados por meio das *strings* de busca. Estes foram submetidos a análise das palavras-chaves e leitura do resumo, a fim de identificar aqueles relevantes para responder às questões de pesquisa. Após a realização dessa análise, o conjunto de periódicos foi reduzido para 69. Estes artigos pré-selecionados

foram submetidos às listas de Critérios de Inclusão (CI) e Critérios de Exclusão (CE) que constam na Tabela 3.

Tabela 3 - Critérios de inclusão e exclusão.

Critérios de Inclusão	Critérios de Exclusão
<ul style="list-style-type: none"> • CI1: Artigos com texto completo do estudo disponível em formato eletrônico; • CI2: Trabalhos com publicação no ano 2013 ou posterior; • CI3: Trabalhos relativos a gamificação na área de aprendizagem de programação de computadores. 	<ul style="list-style-type: none"> • CE1: Artigos Resumidos; • CE2: Trabalhos com publicação anterior ao ano 2013; • CE3: Trabalhos sobre gamificação em áreas distintas a aprendizagem da lógica de programação de computadores; • CE4: Estudos com texto completo não disponível em formato eletrônico; • CE5: Publicação cujo foco não corresponde ao uso de jogos para o ensino da lógica de programação; • CE6: Estudos que não contemplam experiência e/ou observações na perspectiva discente.

Implicitamente os estudos duplicados foram ignorados. Assim, a aplicação de apenas um dos critérios de exclusão, indicado na Tabela 3, determinou se um determinado estudo deveria ser subtraído da lista de estudos pré-selecionados. Após a triagem dos artigos pré-selecionados, foram definidos 34 estudos primários, para isso foi necessário ler na íntegra os trabalhos para submetê-los aos Critérios de Exclusão. A etapa posterior determinou a estratificação dos dados contidos nos estudos.

2.4.2.4 Extração e síntese de dados

Nesta fase, foram extraídos os seguintes dados sobre os estudos selecionados: título da obra, nomes dos autores, data de publicação, veículo de publicação, fonte, resumo, data da execução do estudo, local do estudo, tipo de estudo, nível de ensino ao qual o estudo foi destinado, descrição do estudo, hipóteses avaliadas, variáveis independentes, variáveis dependentes, metodologia de gamificação, ferramenta utilizada, técnicas utilizadas, número de participantes, material, projeto do estudo, ameaças a validade, resultados e referências relevantes. Os principais dados extraídos foram armazenados em uma planilha eletrônica.

2.4.3 Análise

Como resultado da busca e análise dos estudos, de um total de 1.054 artigos, foram pré-selecionados 34 trabalhos. A Tabela 4 apresenta os resultados gerais dos processos de pré-seleção e inclusão dos artigos.

Tabela 4 - Resultado geral das buscas.

Veículo	Total de artigos	Artigos pré-selecionados	Artigos incluídos	Artigos incluídos / Total artigos (%)
RBIE	7	3	1	14,3
SBIE	25	16	1	4
WCBIE	23	7	3	13
WEI	28	12	8	28,6
JAIE	1	0	0	0
DESAFIE	0	0	0	0
IEEE Xplorer	494	16	10	2
ACM	476	15	11	2,3
Total	1054	69	34	3,2

2.4.4 Resultados Gerais

Nesta seção, apresenta-se as respostas às questões levantadas nesta RSL. As referências sobre os estudos primários podem ser encontradas no final deste documento.

2.4.4.1 Reposta a Questão de Pesquisa (QP1)

Nos estudos analisados, pôde-se observar que a quantidade de participantes variou muito. Os estudos de escala menor continham 5, 12 e 14 participantes. Já os maiores tiveram 263, 310 e 350 participantes. Trinta e três relataram o número de participantes da pesquisa realizada. Contudo, um estudo [E5] relatou somente o número de bolsistas e voluntários, omitindo o número da amostra trabalhada para alcançar seu resultado.

2.4.4.2 Reposta a Questão de Pesquisa (QP2)

Após a análise de todos os estudos primários é possível inferir sobre a eficácia da aplicação da gamificação na aprendizagem da lógica de programação de computadores. Uma observação deve ser feita nesse ponto sobre CE6 presente na lista de Critérios de exclusão, uma vez que este foi definido em razão da possibilidade do aumento da probabilidade de eficácia apresentada nos resultados do estudo, pois este restringe os resultados a estudos de natureza empírica.

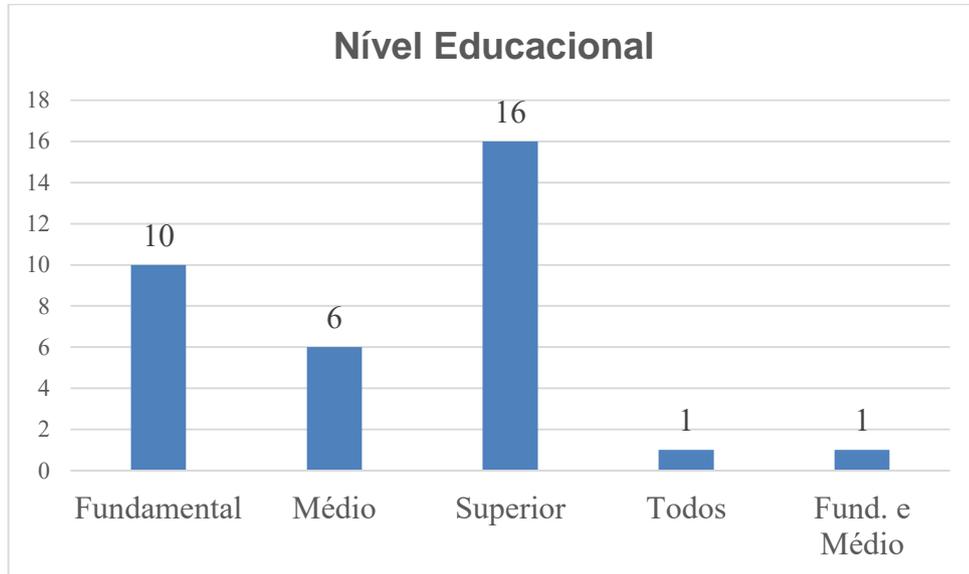
Dos 34 artigos incluídos, todos (100%) relatam que o uso gamificação na aprendizagem da lógica de programação de computadores é eficaz. Essa resposta foi obtida através da análise das seções de conclusão e discussão dos resultados presente nos estudo primários. Nestas seções, todos os autores relataram que seus objetivos foram alcançados, e em todos os casos fizeram referências as ferramentas utilizadas. Como o caso dos relatos presentes no estudo E2, quando os autores afirmam que após o uso das ferramentas houve relatos dos discentes relacionados a motivação, estímulo, diversão e competição. Os pesquisadores concluem ainda que cada uma das ferramentas utilizadas tiveram sua parcela de contribuição, todas motivaram o estudo da programação de computadores. Outro caso observado é o E5, que relata que os autores aplicaram um questionário para mensurar o nível de interesse dos participantes pelas atividades desenvolvidas, segundo eles todos os discentes avaliaram positivamente a experiência. Pode citar ainda o caso do E13, neste estudo os autores afirmam que 92,9% dos alunos que participaram da pesquisa indicaram interesse em continuar estudando programação, fornecendo assim, segundo os mesmos, indício que defende o uso da gamificação na abordagem de conceitos de programação de computadores.

2.4.4.3 Reposta a Questão de Pesquisa (QP3)

Os estudos também foram classificados pelo nível de ensino em que eles foram destinados, conforme demonstra a Figura 1. Dos trabalhos analisados, dez (29%) são destinados ao ensino fundamental [E3, E4, E6, E10, E11, E16, E21, E24, E25 e E31]. Foram identificados seis (18%) estudos que tiveram como público alvo os alunos do ensino médio [E5, E13, E27, E32 e E33]. Para o ensino superior foram verificados dezesseis (47%) estudos [E1, E2, E7, E8, E9, E12, E14, E15, E17, E18, E19, E22, E23, E26 e E29]. Por outro lado, estudos voltados para o ensino técnico

[E8], ensino fundamental e médio [E28] e para todos os níveis educacionais listados simultaneamente [E20] foram detectados somente um (3%) para cada.

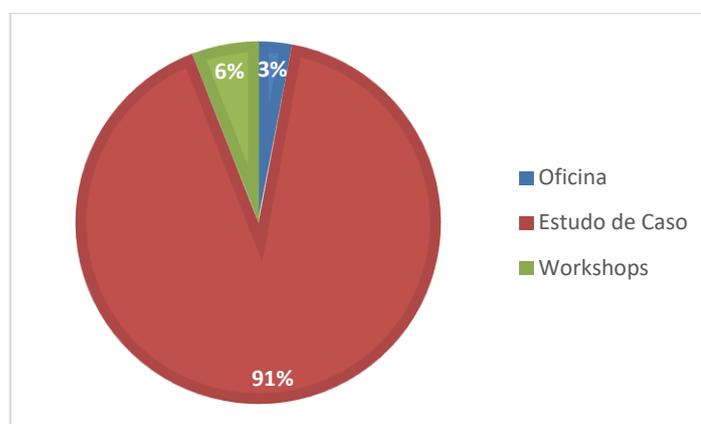
Figura 1 - Estudos classificados pelo nível de Ensino.



2.4.4.4 Reposta a Questão de Pesquisa (QP4)

Como uma das restrições definidas para esta RSL corresponde a seleção de estudos de natureza empírica, isso limitou a diversidade de trabalhos, uma vez que as pesquisas que não contemplaram a perspectiva discente foram descartados. Foi possível identificar em 31 (trinta e um) trabalhos, correspondente a 91%, o uso de Estudo de Caso para se alcançar os objetivos. Em seguida, verificou-se que 6% dos estudos fizeram uso de workshops [E23 e E28]. Por outro lado, somente em um estudo (3%), foi possível determinar o uso de oficinas [E04].

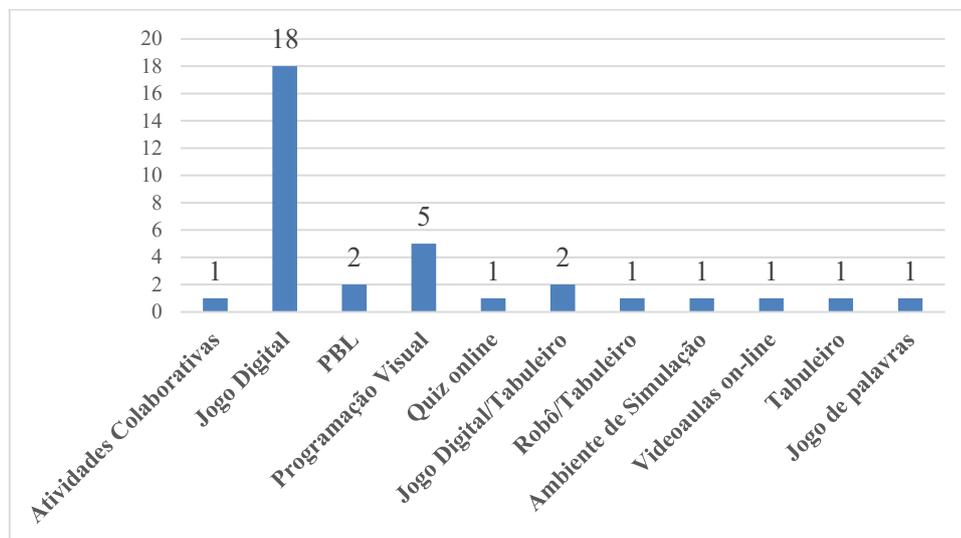
Figura 2 - Tipos de estudos identificados



2.4.4.5 Reposta a Questão de Pesquisa (QP5)

Os estudos também foram classificados quanto às metodologias adotadas para a aplicação da gamificação, conforme é demonstrado na Figura 3. O destaque é dado para o uso de Jogos Digitais, que foi relatado em dezoito trabalhos, correspondendo a 53% dos estudos primários [E1, E5, E6, E11, E13, E14, E15, E17, E18, E21, E22, E24, E26, E29, E30, E31, E32 e E33]. Há pesquisas que também fizeram o relato do uso de Programação Visual, esta metodologia foi identificada em cinco (15%) estudos [E4, E7, E10, E23 e E25]. Dois trabalhos (6%) utilizaram PBL (*Points, Badges, Leaderboards*) [E3 e E12]. A metodologia que fez uso de Jogo Digital e Tabuleiro foi utilizada por dois trabalhos (6%) [E9 e E27]. Atividades colaborativas [E8], Quiz online [E2] e uso de Robô e Tabuleiro [E16], Ambiente de Simulação [E19], Videoaulas on-line [E20], Tabuleiro [E28], e Jogos de Palavras [E34] foram utilizados por 3% cada uma.

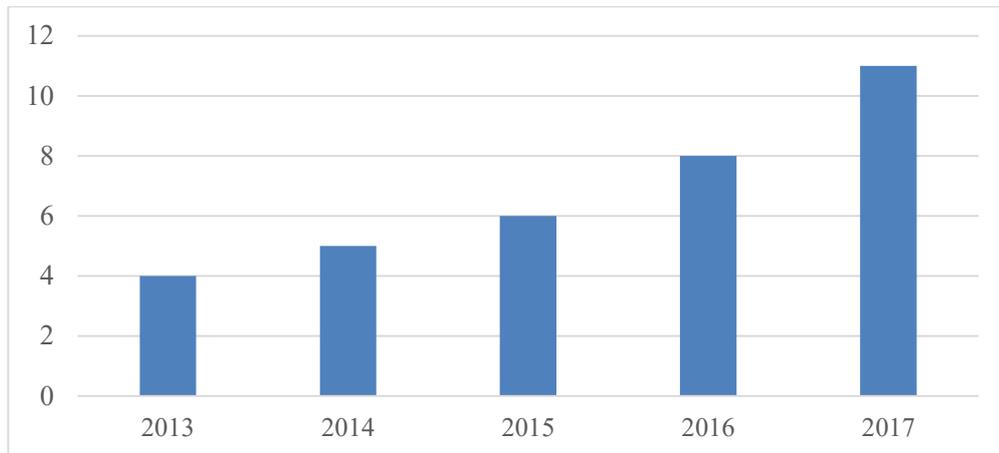
Figura 3 - Metodologia dos estudos primários.



2.4.4.6 Distribuição Temporal

O gráfico da Figura 4 ilustra a distribuição temporal de publicações nacionais relacionadas ao tema desta revisão. Observa-se que há um aumento contínuo de publicações que tratam sobre metodologias gamificadas aplicadas ao ensino da lógica de programação de computadores de natureza empírica, isso mostra que este é um tema que tem chamado atenção dos pesquisadores.

Figura 4 - Distribuição temporal dos estudos.



2.4.4.7 Ferramentas e Técnicas utilizadas

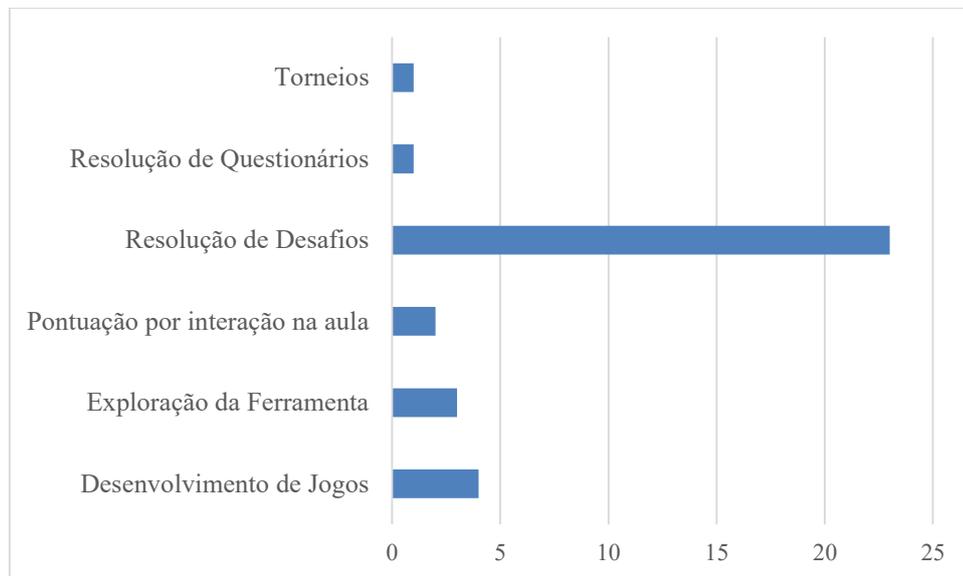
Nesta RSL, durante a extração dos dados, foram identificadas as ferramentas utilizadas para seguir o roteiro da metodologia, conforme demonstra a Tabela 5. Vale observar que alguns pesquisadores não nomearam suas ferramentas, estas não constam na Tabela 5.

Tabela 5 - Ferramentas utilizadas.

Metodologia	Ferramenta
Jogos Digitais	Robocode, Code Combat, Code.org, Lightbot, Code.org, NoBug's Snack Bar, Play Code Dog, Jogo da Serpente, Lost in Space, CMX, Ztech de Object-Oriented, Gidget, Lumber Jack, CodeFruits, Reduct, Cargo-Bot, The Lost Java Code, CodeSpells, Gram's House, Gram's Grocery Shop, Image Representation Game e Hidden Image Game
PBL	Classcraft
Programação Visual	Scratch
Quiz online	Kahoot e Socrative
Atividades Colaborativas	Eclipse
Ambiente de Simulação	Second Life

Além das ferramentas, foram classificadas também técnicas utilizadas para se alcançar os objetivos propostos pelos estudos. Pode-se observar na Figura 5 que a Resolução dos Desafios foi utilizada pela maioria dos estudos (68%) [E1, E6, E8, E11, E13, E14, E15, E16, E17, E18, E19, E21, E22, E24, E26, E27, E28, E29, E30, E31, E32, E33 e E34]. Em segundo lugar, está o Desenvolvimento de Jogos com quatro (12%) estudos [E7, E10, E23 e E25]. Em seguida, foi identificada em Exploração da Ferramenta com três (9%) estudos [E4, E9 e E20]. Por outro lado, a Pontuação por interação na aula foi identificada em dois (6%) estudos [E3 e E12]. E Por último, com 3% de utilização, estão a Resolução de Questionários [E2] e Torneios [E5].

Figura 5 - Técnicas utilizadas.



2.4.5 Considerações Finais

Foram apresentados os resultados de uma RSL sobre metodologias gamificadas aplicadas ao ensino da lógica de programação de computadores publicadas durante o período de 2013 a 2017 no SBIE, WEI, WCBIE, RBIE, IEEE Xplorer e ACM Digital Library. A busca realizada resultou na pré-seleção de 69 artigos, dentre os quais 34 foram incluídos para a extração de dados. Pode-se observar que a gamificação se mostrou um recurso eficaz na aprendizagem da lógica de programação de computadores em todos os estudos analisados, se mostrando um método que ajuda no desempenho dos discentes, diminuir evasão e aumenta a motivação.

Neste contexto, esta RSL identificou que as metodologias que mais se destacaram foram Jogos Digitais, Programação Visual e PBL, por meio de ferramentas como Robocode, Scratch e Classcraft. Pôde-se observar também que a metade dos estudos foram destinados ao ensino básico, confirmando a previsão de Silva et al. (2014) de que havia uma tendência de aumento de pesquisas voltadas ao ensino fundamental e médio quando se comparado com o nível superior. Vale observar também que não foram encontrados nos estudos primários abordagens que fizessem referência ou comparação entre os elementos monousuário e multiusuário presente nos jogos digitais.

Todos os estudos primários foram classificados como estudos de natureza empírica, assim pôde-se agregar maior relevância aos resultados expressados através do relato de experiência dos pesquisadores. Contudo, vale observar que se faz necessário a existência de um número maior de estudos classificados com este atributo, uma vez seria desejável que o número de estudos primários fosse maior.

No geral, os resultados da RSL indicam que a aplicação da gamificação na aprendizagem da lógica de programação de computadores é eficaz e as metodologias mais utilizadas, respectivamente, são Jogos Digitais, Programação Visual e PBL.

2.5 CODE COMBAT

O Code Combat é um jogo que aborda conceitos de programação através de uma aventura medieval, onde o jogador controla heróis e, ao longo de sua jornada, vai coletando cristais e derrotando diversos inimigos, como ogros, bandidos, entre outros (CODECOMBAT, 2019). É um jogo gratuito e é baseado na solução de desafios pelo herói controlado. Estes desafios são apresentados na forma de missões que possuem enredo que evolui de acordo com o conteúdo abordado. O usuário inicia o jogo no primeiro mapa, as primeiras missões deste são gratuitas. Cada mapa aborda um conjunto de conceitos distintos, como Introdução à Ciência da Computação (CS1), Ciência da Computação 2 (CS 2), Programação Web etc. Portanto, cada mapa representa um curso diferente, para ter acesso a todas as missões de cada mapa é necessário o professor (ou escola) adquirir a licença correspondente ao curso desejado.

O jogo disponibiliza um ambiente de programação próprio para que o jogador implemente seus códigos. Este ambiente trabalha com 3 linguagens de

programação, sendo elas: Python, JavaScript e CoffeeScript (Experimental). O jogador pode escolher entre 60 idiomas na página inicial do site do jogo.

Cada mapa fornece uma estimativa de tempo necessário para sua conclusão. Os conceitos abordados e as informações a respeito de cada mundo estão dispostas na Tabela 6.

Tabela 6 - Informações sobre os mapas/cursos do Code Combat.

Curso	Quantidade de Fases	Tempo de Conclusão Estimado	Conceitos Abordados
Introdução à Ciência da Computação	37	4 x 45-60 minutos de sessão de programação.	Sintaxe Básica, Argumentos, Strings, Loops while, Variáveis, Algoritmos.
Desenvolvimento de jogo 1	17	Sessões de programação de 5 x 45 a 60 minutos.	Sintaxe Básica, Argumentos, Usar objetos do jogo, Construir labirintos, Criar um jogo jogável e compartilhável.
Desenvolvimento Web 1	13	N/D	HTML Básico, CSS Básico, Alterar páginas web existentes, Criar uma página web compartilhável.
Ciência da Computação 2	75	6 sessões de programação de 45 a 60 minutos.	Sintaxe Básica, Argumentos, Strings, Loops while, Variáveis, Comandos if, Funções, Parâmetros, Strings avançadas.
Desenvolvimento de jogo 2	30	8 sessões de codificação de 45 a 60 minutos.	Sintaxe Básica, Funções, Strings, Comandos if, Argumentos, Manipulação básica de entrada, IA Básica do jogo, Criar um jogo jogável e compartilhável.
Desenvolvimento Web 2	22	N/D	Sintaxe Básica, Strings, Loops while, Variáveis, Argumentos, Comandos if, Funções, HTML Básico, CSS Básico, Web Script Básico, HTML Avançado, JavaScript Básico, Manipulação básica de eventos, Criar uma página web interativa compartilhável.
Ciência da Computação 3	82	11 x 45-60 minutos de sessões de codificação.	Sintaxe Básica, Argumentos, Comandos if, Variáveis, Strings, Loops while, Aritmética, Strings avançadas, Manipulação de entrada, Funções, Parâmetros, Lógica

			Booleana, Arrays, Comandos Break, Comandos continue.
Desenvolvimento de Jogos 3	11	N/D	Sintaxe Básica, Dados de Eventos.
Ciência da Computação 4	37	6 x 45-60 minutos de sessões de codificação.	Sintaxe Básica, Argumentos, Variáveis, Strings, Comandos if, Aritmética, Loops while, Arrays, Funções, Parâmetros, Objetos Literais, Laço For, Lógica Booleana.
Ciência da Computação 5	34	12 x 45-60 minutos de sessões de codificação.	Sintaxe Básica, Argumentos, Variáveis, Strings, Loops while, Arrays, Laço For, Funções, Comandos if, Lógica Booleana, Algoritmos, Comandos Break, Operações Matemáticas, Objetos Literais, Aritmética, Gráficos, Strings avançadas, Parâmetros, Comandos continue.
Ciência da Computação 6	19	N/D	Sintaxe Básica, Argumentos, Variáveis, Strings, Comandos if, Loops while, Arrays, Laço For, Funções, Lógica Booleana, Algoritmos, Comandos Break, Aritmética, Objetos Literais, Parâmetros, Vetores, Operações Matemáticas, Recursividade.

Fonte: CODECOMBAT, 2019.

Figura 6 - Escolha dos mundos disponíveis.



Fonte: www.codecombat.com

Ao acessar a página do game será exibido ao usuário o conteúdo da Figura 6. Nesta tela, o usuário deverá escolher os mapas disponíveis, em geral, para o usuário sem licença, somente o primeiro é parcialmente disponibilizado. Cada mapa aborda conceitos específicos de programação de computadores.

Figura 7 – Mapa Kithgard (CS1).



Fonte: www.codecombat.com

Após selecionar o mundo, o usuário tem acesso as missões disponíveis para o mapa escolhido. A Figura 7 exhibe a primeira missão do mapa Kithgard; conforme as missões são concluídas novas são disponibilizadas. Contudo, somente as 3 primeiras são disponíveis para o usuário que não está registrado em alguma turma virtual. As turmas virtuais só podem ser criadas por usuários com perfil de professor no site.

Figura 8 - Descrição da missão.



Fonte: www.codecombat.com

Após a escolha da missão, o usuário tem acesso a descrição dos objetivos da missão e o conteúdo de programação que será abordado, como demonstrado na Figura 8.

Figura 9 - Escolha do avatar (herói).



Fonte: www.codecombat.com

A Figura 9 demonstra o passo seguinte, a escolha do herói que irá representar o usuário ao longo das missões e a linguagem de programação que será utilizada para a escrita dos algoritmos necessários para o cumprimento das missões. As linguagens de programação disponíveis são: Python, JavaScript e CoffeeScript.

Figura 10 - Orientação para solução da missão



Fonte: www.codecombat.com

Antes de iniciar a missão, o usuário é orientado sobre os procedimentos que ele deve ter atenção para alcançar seus objetivos, conforme demonstra a Figura 10. Nesta missão, o objetivo do usuário é coletar a gema sem tocar nos espinhos.

Figura 11 - Tela de missão do Code Combat



Fonte: www.codecombat.com

Finalmente o usuário tem acesso aos recursos do ambiente que o permite cumprir a missão. Esta página é composta por vários elementos, os principais são:

- Área de exibição dos personagens e ambiente gráfico – neste local são exibidos os componentes visuais do jogo, como personagem, gemas, ogros, espinhos, armadilhas e outros. O personagem locomove-se no ambiente de acordo com o algoritmo definido no editor de código;

- Objetivos – lista o status dos objetivos a serem alcançados ao longo da missão;
- Métodos – lista os métodos disponíveis para o jogador. Métodos são responsáveis por realizar uma ação específica e, em geral, são vinculadas ao personagem. Cada método dispõe de um texto explicativo sobre seu conceito e exemplo de utilização;
- Sugestões – exhibe um conjunto de informações sobre a missão e métodos que podem auxiliar o jogador a cumprir os objetivos da missão.
- Editor de código – ferramenta que permite o usuário escrever o código correspondente ao conjunto de instruções habilitadas para a solução da missão. Caso o código escrito esteja correto, o personagem realiza o conjunto de ações esperadas e alcança os objetivos da missão, caso contrário o jogo informa ao usuário que determinado objetivo não foi alcançado.

Este jogo permite o cadastro de dois tipos de perfis: professor e aluno. O usuário não cadastrado tem acesso somente à parte do primeiro mapa. Um professor pode criar turmas. Em cada turma ele poderá adicionar um ou mais cursos, de acordo com a licença que este tenha adquirido no site. Cada curso representa um dos mapas disponíveis. As turmas podem ter um ou mais alunos. O professor consegue acompanhar o progresso de cada aluno da turma, contudo um aluno não tem acesso ao progresso dos outros discentes em cada curso. A ferramenta conta com um recurso multiusuário, chamado arena, ela está disponível no final de cada mapa. Nesse espaço, os alunos da turma podem confrontar-se. Os resultados das batalhas entre eles podem ser visualizados em um ranking.

3 PROCEDIMENTOS E MÉTODOS

Neste capítulo, é apresentada a metodologia utilizada para o desenvolvimento desta pesquisa e, ante ao conteúdo do referencial teórico, são detalhados os procedimentos realizados para alcançar os objetivos do trabalho.

3.1 TIPO DE PESQUISA

Este estudo pode ser classificado de acordo com os parâmetros das pesquisas descritiva e explicativa. Através da pesquisa descritiva será possível observar, registrar, analisar, classificar e interpretar os fatos relativos as observações quanto a realização do estudo.

Para Pedro (2016), a metodologia explicativa, que, para a autora, também pode ser denominada de causal ou experimental, ocorrerá em uma segunda etapa, objetivando identificar os fatores que contribuem para a ocorrência das variáveis que afetam o processo analisado. Pedro (2016) ainda comenta, conforme citado por Lakatos e Marconi (2010), que a pesquisa explicativa registra os fatos, analisa-os e identifica suas causas de modo a relacionar as hipóteses deduzidas por força da lógica. Por meio desta metodologia, será possível estabelecer relações de causa-efeito através da manipulação das variáveis da pesquisa, validação das hipóteses e a avaliação da questão da pesquisa através da realização de experimentos.

3.2 MÉTODOS CIENTÍFICOS

Os métodos científicos adotados para esta pesquisa são os métodos observacional, experimental e estatístico. O método observacional será utilizado com o fim de coletar dados sobre um conjunto de atitudes comportamentais típicas por meio da observação sistemática. Para Prestes (2012), esta classificação do método observacional é realizada em condições controladas, de acordo com objetivos e propósitos previamente definidos.

O método experimental (ou de experimentação) será aplicado através de um conjunto de experimentos para verificar as hipóteses por meio do controle das variáveis capazes de interferir na relação causa/efeito estudada.

Por outro lado, o método estatístico será utilizado para investigar o fenômeno estudado, uma vez que este método está relacionado com a quantificação, análise e interpretação dos dados coletados a partir das técnicas estatísticas.

3.3 ABORDAGEM DE PESQUISA

A abordagem de pesquisa adotada no projeto é qualitativa e quantitativa. A primeira será utilizada com o objetivo de emergir aspectos subjetivos ao comportamento e desempenho durante a realização dos experimentos controlados. Por outra lado, a segunda será utilizada para gerar dados para mensurar resultados, para fins de comparação, entre o uso da ferramenta gamificada com elementos monousuário e a ferramenta gamificada multiusuário. Esta comparação se faz necessária para testar a hipótese e questão de pesquisa do projeto.

3.4 HIPÓTESES DE PESQUISA

H₀: Não existem diferenças em termos de engajamento, motivação e desempenho na utilização de ambientes gamificados monousuário e ambientes gamificados multiusuários no ensino-aprendizagem da lógica de programação de computadores.

H₁: Existem diferenças em termos de engajamento, motivação e desempenho na utilização de ambientes gamificados monousuário e ambientes gamificados multiusuários no ensino-aprendizagem da lógica de programação de computadores.

3.5 VARIÁVEIS

A fim de testar a hipótese foram encontradas variáveis que estão relacionadas ao contexto desta pesquisa. A variável independente identificada são os elementos de jogos multiusuários destinados ao ensino-aprendizagem da lógica de programação de computadores. Contudo, vale ponderar a relevância das variáveis dependentes, são elas: engajamento; desempenho e motivação. Todas as variáveis estão relacionadas ao uso do ambiente gamificado que será utilizado nos experimentos.

3.6 CARACTERIZAÇÃO DA AMOSTRA

A amostra do estudo foi composta por alunos do primeiro ano do curso Técnico em Desenvolvimento de Sistemas Integrado ao Ensino Médio do Instituto Federal de Educação, Ciência e Tecnologia do Pará (FPA) Campus Óbidos. Atualmente, os estudantes nessa faixa etária geralmente estão intimamente relacionados a um ou mais jogos eletrônicos, ou são mais propícios a imergir nestes em razão do contexto em que se encontram. Logo presume-se que este tipo de jogo pode atuar como recurso motivacional no processo de ensino-aprendizagem deste público, uma vez que esta abordagem é visual.

3.7 PROCEDIMENTOS

Os procedimentos adotados para a realização dessa pesquisa de mestrado foram desenvolvidos em três momentos distintos: *i)* revisão sistemática da literatura, para identificar os trabalhos relacionados ao tema de investigação; *ii)* adaptação e preparação do ambiente experimental (Code Combat); *iii)* realização de experimentos para validação das hipóteses.

3.7.1 Revisão da Literatura

A fim de identificar os trabalhos e pesquisas relacionados a tema de estudo, foi realizada uma revisão sistemática da literatura (seção 2.4). Através desta foi possível investigar as iniciativas, práticas, abordagens e experiências do uso da gamificação no ensino-aprendizagem da lógica de programação de computadores. Contudo, apesar da contribuição com informações relevantes ao objeto de estudo, nenhum deles trouxe conteúdo conclusivo sobre a questão de pesquisa desse trabalho, a saber: *quais as evidências que o uso de ambientes gamificados multiusuários são significativamente melhores que ambientes gamificados monousuário com a finalidade de aumentar o engajamento, desempenho e motivação do discente no processo de ensino-aprendizagem da lógica de programação de computadores?*

Assim, buscando avaliar o impacto do uso de ambientes gamificados no processo de ensino-aprendizagem da lógica de programação de computadores, foi proposto um ambiente gamificado e foram definidos cenários de aprendizagem para realização de experimentos.

3.7.2 Preparação do ambiente de experimentação

Com a finalidade de alcançar os objetivos do estudo, buscar respostas para as questões dessa pesquisa e validar ou refutar a hipótese, foi necessário preparar o ambiente gamificado de modo a implementar os recursos necessários para a realização dos experimentos. Deste modo, foi necessário criar duas turmas virtuais no ambiente, ambas com acesso aos recursos gamificados, contudo uma das turmas não teve acesso aos recursos multiusuários (vide seção 3.7.2.5).

3.7.2.1 Instrumentos para aferir o engajamento

Observando o contexto acadêmico, o engajamento dos discentes foi dado em razão da avaliação da dedicação, participação e frequência de uso da ferramenta, portanto foram implementados recursos no software de experimento que registra estes dados. Ou seja, após as modificações o ambiente gamificado passou a fornecer os seguintes dados sobre o usuário: frequência de acesso à ferramenta; tempo de duração da seção; quantidade de missões cumpridas e recursos mais utilizados. Estes dados foram utilizados para aferir o engajamento do discente quanto ao uso da solução. Estes foram analisados na fase de análise de dados. Os dados foram correlacionados de acordo com grupo de experimento que forem extraídos.

3.7.2.2 Instrumentos para aferir o desempenho

Para avaliar o desempenho dos discentes quanto a aprendizagem da lógica de programação de computadores através do uso do ambiente gamificado, foram aplicados questionários através de um pós-teste. O questionário (Apêndice C) foi composto por elementos que tratam sobre conceitos relacionados à lógica de programação de computadores abordados nas missões do jogo com fim de aferir a absorção dos discentes quanto ao conteúdo tratado.

3.7.2.3 Instrumentos para aferir a motivação

Atualmente, existem várias formas de verificar se o indivíduo está motivado ou não (TAORMINA; GAO, 2013). Para Pedro (2016), a natureza da motivação intrínseca de ambientes virtuais tem sido atribuída ao seu potencial de satisfazer as necessidades psicológicas de autonomia, competência e relacionamento.

Assim, Deci e Ryan (2011) propuseram um questionário com 22 questões curtas, denominada *Intrinsic Motivation Inventory (IMI)*. Segundo Pedro (2016), esse

questionário é um dispositivo de medição multidimensional destinado a avaliar a experiência subjetiva dos participantes com atividades desenvolvidas em experiências laboratoriais. Esse instrumento tem sido utilizado em diversas experiências relacionadas a motivação intrínseca e autorregulação e é usado para medir a motivação em 4 categorias (DECI; RYAN, 2011):

- Interesse/Aproveitamento;
- Competência;
- Escolha;
- Pressão/Tensão;

Neste sentido, o estudo fez uso do *IMI* como instrumento para aferir a motivação dos discentes quanto ao uso do ambiente gamificado adotado pelo projeto. No apêndice B, encontra-se uma versão adaptada do *IMI* ao propósito deste estudo.

3.7.2.4 Definição dos elementos do ambiente gamificado

Assim como citado no Capítulo 2 (Fundamentação Teórica e Revisão Sistemática da Literatura), existem vários padrões e elementos que se fazem presentes na gamificação. Estes, quando bem definidos para o grupo de usuários, são responsáveis pelo sucesso no alcance dos objetivos buscados. Para Vassileva (2012), os padrões mais comuns estão são:

Quadro 1 - Padrões de mecânica de jogos

Padrões	Descrição
Propriedade	Permitir que o usuário/jogador possua coisas, como pontos, símbolos, badges, pois cria lealdade ao aplicativo, jogo ou comunidade.
Conquistas	Fornecer uma representação virtual ou física de ter conquistado algo que pode ser fácil, difícil, surpreendente, divertido e conquistado sozinho ou em grupo.
Status	Cálculo e exibição de uma classificação (ranking) ou nível de um usuário/jogador

Colaboração	Colocando desafios para os usuários relacionados aos prazos ou concorrência, que podem ser resolvidos através de trabalho colaborativo.
-------------	---

A partir dos objetivos a serem alcançados, foram selecionados elementos padrões e elementos de jogos para serem implementados no ambiente educacional a ser utilizado durante os experimentos. O quadro abaixo exhibe tais elementos e padrões.

Quadro 2 - Elementos de jogos selecionados de acordo com os padrões relacionados

PADRÕES	ELEMENTOS DO JOGO
Propriedades	- Pontos - <i>Badges</i> - Itens - Gemas
Conquistas	- <i>Feedback</i> - Resultados Emocionais - Desafio - Regras
Status	- Níveis - <i>Ranking</i>

Todos os padrões e elementos relacionados no Quadro 2 fazem parte da versão atual do ambiente gamificado utilizado no experimento.

3.7.2.5 Criação do ambiente de experimentação

Para implementação dos experimentos, faz-se necessário o uso de um ambiente gamificado voltado para o ensino-aprendizagem da lógica de programação de computadores que faça uso dos elementos descritos no Quadro 2. O desenvolvimento de um jogo que possuísse estas características era o objetivo inicial, todavia três possíveis cenários foram avaliados:

1. Uso de uma ferramenta existente - este cenário contempla o uso de uma ferramenta que implemente todas as características dos elementos

desejados (recursos monousuário e multiusuários). Contudo, não foi encontrada qualquer ferramenta, no âmbito da aprendizagem da lógica de programação de computadores, que possuísse todos os recursos necessários para se alcançar os objetivos do projeto.

2. Desenvolvimento de uma ferramenta – este cenário evidenciou um potencial problema em relação ao cronograma de execução do projeto, uma vez que isto exige uma carga horária de trabalho relativamente alta.
3. Adaptar uma ferramenta existente – uma pré-condição para este cenário corresponde à escolha de uma ferramenta que atende parcialmente às características procuradas, tendo o código disponível e de livre acesso.

O último cenário foi escolhido para este projeto. Um ponto positivo que vale destacar para este é a diminuição da complexidade técnica do projeto. Deste modo, pôde-se focar na avaliação de possíveis ferramentas que se enquadram no contexto. A análise do estado da arte mostrou a existência de duas ferramentas *open-source* candidatas: Robocode e Code Combat.

Devido a problemas relacionados às dependências e necessidade de implementação e/ou mudanças em maior conjunto de elementos (Quadro 3), o Robocode não foi selecionado como ferramenta a ser adotada pelo projeto.

Quadro 3 - Elementos de jogos do Robocode e Code Combat.

Elementos	Robocode	Code Combat
Pontos	✓	✓
Badges	✗	✓
Itens	✗	✓
Gemas	✗	✓
Feedback	✗	✓
Resultados Emocionais	✓	✓
Desafio	✓	✓
Regras	✓	✓
Níveis	✗	✓
Ranking	✗	✗

O Quadro 3 faz um comparativo da implementação dos elementos do jogo, descritos no Quadro 2, pelos dois jogos. É evidente a diferença da quantidade de elementos implementados por ambos. Vale observar que o fato de o Robocode ser ambiente do estilo torneio, este faz uso de um ranking. Contudo, esse ranking é acessível somente no fim de cada torneio e a pontuação dos jogadores não influencia em futuros eventos.

Conforme apresentado na Quadro 3, o jogo digital Code Combat foi escolhido por ter vários dos recursos da gamificação que se pretendia utilizar nos experimentos. Vale observar que neste jogo *open-source* também foram encontrados problemas relacionados a dependência, contudo o apoio de uma comunidade de desenvolvedores mais consolidada foi fundamental para a solução destes.

Figura 12 – Code Combat



Fonte: www.codecombat.com

A fim de alcançar os objetivos do estudo, foram realizadas modificações no Code Combat a partir da implementação de recursos atualmente inexistentes:

- Tempo de jogo – esse recurso exibe informações sobre o total de tempo que cada usuário permaneceu *logado* no jogo. Este recurso foi essencial para ajudar a aferir o engajamento dos discentes de ambos os grupos.
- Ranking da turma – este recurso permite que os alunos possam acompanhar o progresso de seus colegas a partir da visualização de um ranking que mostrará o avanço de cada aluno. O uso deste recurso foi limitado ao segundo grupo, uma vez que este é um recurso multiusuário.

Uma outra modificação foi realizada no código do jogo a fim de restringir o acesso a uma turma virtual específica (grupo experimental) de modo que não possibilite a visualização do ranking da turma por outras turmas virtuais do jogo.

3.7.3 Experimento Controlado

Tendo como norte os objetivos do estudo, o experimento realizado foi dado a partir da aplicação do Code Combat em duas turmas virtuais distintas. A primeira turma participou do experimento destinado aos recursos monousuário. Por outra lado, a segunda turma usufruiu dos recursos monousuário e multiusuários.

Os elementos de jogos e padrões selecionados possuem a seguinte mecânica de jogos:

- **PONTOS:** Cada jogador acumula pontos ganhos através da solução de cada desafio. Estes pontos são rotulados como pontos de experiência (XP). São usados como parâmetro para o sistema de níveis, ou seja, quanto maior é a pontuação de um usuário maior é seu nível.
- **NÍVEL:** o sistema de níveis é usado para determinar o nível de experiência de um jogador. A medida que o jogador sobe de nível ele libera novos recursos para seu personagem e, conseqüentemente, conhece novas habilidades de programação.
- **BADGES:** O usuário recebe *badges* (medalhas) à medida que supera determinado grupos de missões.
- **GEMAS:** A medida que o usuário cumpre as missões, assim como pontos, ele ganha gemas, este recurso é usado para comprar novos itens.
- **ITENS:** São recursos (vestuário, escudos, espadas e outros) que o jogador pode liberá-los quando alcança determinados níveis ou compra-los através de uma determinada quantidade de gemas.
- **FEEDBACK:** O usuário conta com um editor de texto para auxiliá-lo na execução da missão, esta ferramenta possui um assistente de código que o auxilia durante o procedimento de escrita do algoritmo. Caso haja alguma dúvida, o usuário pode recorrer ao botão “sugestões”, este apresenta o conceito e exemplo de uso dos comandos de cada desafio. Quando o jogador terminar de escrever seu algoritmo, basta ele clicar no botão “pronto”, este se encarregar de executar seu código. Na tela, são exibidos os objetivos de cada missão. Quando estes são concluídos sua cor é alterado para verde, caso contrário sua cor altera para vermelho.

- **RANKING:** O *Ranking* funciona através pontos ganhos à medida que o jogador concluir as missões no modo multiusuário e pode ser visualizado por todos da turma.

3.8 AVALIAÇÃO

A avaliação foi dividida em 3 etapas: Experimentação (os alunos tiveram contato com o Code Combat através do cumprimento de suas missões); Teste de retenção de conhecimento (teste sobre programação de computadores a fim verificar o grau de assimilação do conteúdo abordado); Teste Motivacional (avaliar o nível de motivação dos discente a respeito da metodologia trabalhada).

3.8.1 Experimentação

Tendo como norte os objetivos do estudo, o experimento contou com participação de dois grupos de estudantes que fizeram uso do Code Combat. O primeiro grupo utilizou recursos do jogo correspondentes aos elementos monousuário (pontos e medalhas do PBL). O segundo grupo teve acesso a todos os recursos do jogo correspondentes aos elementos monousuário e multiusuário (ponto, medalhas e ranking de líderes do PBL).

As atividades de experimentação foram realizadas durante o período de abril a maio de 2019. A amostra foi extraída a partir da turma do primeiro ano do curso Técnico em Desenvolvimento de Sistemas integrado ao Ensino Médio do Instituto Federal de Educação, Ciência e Tecnologia do Pará (IFPA) – Campus Óbidos. Até a realização do experimento, os alunos não haviam tido contato com a programação de computadores. A amostra foi classificada em dois grupos: um grupo de controle, utilizando somente os elementos monousuário – grupo 1 (19 alunos); e um grupo experimental, usando todos os recursos, monousuário e multiusuário – grupo 2 (19 alunos). A definição dos membros de cada grupo foi dada a partir de sorteio. Os grupos (grupo de controle e grupo experimental) foram separados a fim de evitar interferência da relação destes no resultado. Para participação na atividade, os estudantes receberam um Termo de Consentimento Livre e Esclarecido (TCLE) para que pudessem solicitar a assinatura de seus responsáveis.

O experimento foi executado por meio da oferta de um minicurso de introdução a programação de computadores, com carga horária de 12 horas,

distribuídas em 3 (três) semanas. Cada grupo experimental contou com 6 (seis) encontros presenciais. Ambos os grupos receberam orientações iniciais para o jogo, material impresso de apoio e um conjunto de exercícios. O material de apoio e exercícios foram adaptados a partir do conteúdo disponível no site oficial do Code Combat (Anexos A e B). Tiveram acompanhamento de um professor e 3 monitores para auxiliá-los na solução de eventuais dúvidas. Após as orientações iniciais do primeiro encontro, cada aluno progrediu no jogo de acordo com seu ritmo. Desse modo, os estudantes não foram compelidos a progredir no jogo por influência de uma autoridade presente no laboratório de informática.

Foram disponibilizados aos grupos dois cursos disponíveis no site do jogo: Introdução à Ciência da Computação (CS1) e Ciência da Computação 2 (CS2). Cada curso possui um mapa diferente: o primeiro é composto por 38 missões, estas abordam sobre conteúdos relacionados a sintaxe básica, laços de repetição e variáveis; o segundo curso foi liberado para cada aluno após a conclusão do primeiro, é composto por 73 missões, aborda sobre condições se/senão, condições aninhadas, funções e eventos.

Os estudantes do grupo 2 (G2) foram apresentados, ainda no primeiro encontro, ao ranking da turma (Figura 13), que apresenta a classificação de todos os jogadores do grupo de acordo com a conclusão das missões. Pode-se observar a quantidade de alunos que participaram do primeiro curso é maior do que a do segundo. O G2 também foi estimulado a utilizar a “arena”, um recurso do Code Combat que permite que os jogadores da turma batalhem entre eles. Como resultado, o jogo cria um ranking específico, conforme exibido na Figura 14.

3.8.2 Teste de retenção de conhecimento

Para avaliar o desempenho dos discentes, quanto à aprendizagem da lógica de programação de computadores através do uso do ambiente gamificado, foram aplicados questionários através de um pós-teste, conforme definido na seção 3.7.2.2. O questionário aplicado é composto por 11 (onze) perguntas que tratam de conceitos abordados no primeiro mapa do jogo, com a finalidade de aferir a absorção dos discentes quanto aos conceitos sobre a sintaxe básica, laços de repetição e variáveis. Alguns alunos não alcançaram as missões do segundo mapa, em virtude disso, os conceitos relacionados ao segundo mapa (condições se/senão, condições

aninhadas, funções e eventos) não compuseram este questionário. Este instrumento está localizado no apêndice C, e foi estruturado com base nos exercícios realizados pelos discentes e a abordagem do jogo quanto ao conteúdo de lógica de programação.

3.8.3 Teste motivacional

A última etapa corresponde a aplicação de um teste motivacional, conforme definido na seção 3.7.2.3. Isto se faz necessário para corroborar ao resultado a ser encontrado quanto as hipóteses do estudo.

3.9 TÉCNICA DE ANÁLISE DE DADOS

A análise de dados foi realizada a partir dos dados coletados através dos questionários e dados extraídos do Code Combat. A análise de dados foi realizada de duas formas: análise qualitativa e análise quantitativa.

3.9.1 Análise qualitativa

A análise qualitativa foi realizada com o propósito exploratório, a fim de emergir nos aspectos subjetivos do estudo. Esta análise foi realizada a partir de observações pessoais dos pesquisadores envolvidos durante a aplicação dos experimentos.

3.9.2 Análise quantitativa

A análise quantitativa foi realizada por meio de gráficos gerados a partir de dados extraídos dos questionários (teste de retenção de conhecimento e motivacional) e banco de dados da ferramenta, a fim de obter, estatisticamente, dados para testar a hipótese e responder à questão de pesquisa. Esta análise foi realizada com base nos dados correspondentes: quantidade de missões cumpridas pelos alunos; tempo de utilização da ferramenta; e frequência de acesso ao jogo.

Além de outros fatores, para Normando, Tjäderhane e Quintão (2010), a escolha do teste estatístico apropriado requer o conhecimento sobre como os dados estão distribuídos após o término da coleta de dados (Distribuição Normal ou Distribuição Anormal). Por isso, a análise da distribuição dos dados é o primeiro passo para determinar que tipo de análise estatística deve ser utilizada (PEDRO, 2016).

Para isso, após a coleta dos dados foi efetuada a análise da curva de distribuição geral de dados utilizando medidas de tendência central (como média) e a partir do desvio-padrão foi possível estimar a variabilidade dos dados.

Neste sentido, a partir deste conjunto de dados será possível abstrair informações necessária para refutar ou confirmar a hipótese do estudo.

4 RESULTADOS

Neste capítulo, são apresentados os resultados encontrados ao longo dos experimentos, isto foi realizado a partir da exposição e análise dos dados obtidos. Vale observar que, neste capítulo, também foram adicionadas informações relacionadas à modificação do ambiente utilizado.

4.1 MODIFICAÇÃO DO CODE COMBAT

O Code Combat se mostrou um software relativamente complexo para realizar modificações em razão de vários fatores. Os problemas se mostraram presente desde a fase de instalação do ambiente até sua modificação. Para superar os obstáculos relacionados a modificação da ferramenta foi imprescindível a realização de parcerias.

4.1.1 Instalação do Code Combat

Antes de realizar as modificações planejadas foi necessário instalar e executar o código *open-source* do Code Combat. O software foi desenvolvido originalmente no ambiente Mac OS. Contudo os desenvolvedores utilizaram recursos para possibilitar a integração com outros Sistemas Operacionais (SO), entre eles os SOs baseados no Windows e Linux.

Inicialmente, tentou-se realizar a instalação do ambiente no Microsoft Windows 7. Após esta fase, o ambiente de trabalho contava com o código aberto do Code Combat e uma cópia do banco de dados, baseado no MongoDB. Contudo, em razão da necessidade do adiamento da modificação no cronograma, não seguiu-se imediatamente a fase seguinte, correspondente a modificação da ferramenta.

Alguns meses após a primeira instalação da ferramenta, observou-se que o software havia sofrido considerável modificação em sua estrutura. Na nova versão não havia mais um banco de dados disponível para download, este é fundamental para acessar os recursos do jogo, como mapas, dados do professor e estudantes, informações sobre a avatares etc. Nesta versão do ambiente, o banco de dados é acessado, em tempo de execução, no servidor do próprio Code Combat através de um *proxy reverso*.

Portanto, optou-se por realizar uma nova instalação do Code Combat. Entretanto, desta vez, as adições de novos módulos e atualização de outros, ocasionou no insucesso da instalação do ambiente no Microsoft Windows 7, mesmo após várias tentativas. Em razão disso, optou-se por utilizar outras versões do SO Windows, contudo o êxito não foi obtido.

A distribuição Linux intitulada Ubuntu foi alvo de vários testes da instalação da ferramenta e, após várias tentativas, conseguiu-se instalar com sucesso o Code Combat nesse SO. Vale observar que esta distribuição foi escolhida em razão da recomendação da comunidade de desenvolvedores do jogo, segundo esta comunidade alguns módulos do jogo se comportam bem nesta versão desta distribuição, especialmente na versão 16.04, versão utilizada.

4.1.2 Execução do Code Combat

Após as modificações estruturais sofridas pelo Code Combat a forma de execução do ambiente foi limitada ao “modo proxy”. Este modo permite que a versão do Code Combat instalada no ambiente de trabalho seja “alimentada” pelo servidor do jogo por meio de um *proxy reverso*. Isto é fundamental para realizar os testes à medida que o código do jogo venha a ser modificado.

Entretanto, encontrou-se dificuldade na execução do modo proxy, uma vez que quando executado o Code Combat do ambiente de trabalho não obtinha sucesso na conexão. Foram realizados testes em dispositivos com o Mac OS. Contudo, o sistema não demonstrou avanço quanto ao tema em questão.

Após consulta à comunidade de desenvolvedores, concluiu-se que o modo proxy do Code Combat atualmente não funciona no território do Brasil. Eventuais modificações na estrutura do código do jogo de modo que viessem a beneficiar os desenvolvedores brasileiros quanto ao problema encontrado foi cogitado pelos engenheiros do software. Entretanto, isso demandaria um tempo não disponível no cronograma do estudo. Uma solução encontrada para este problema foi o uso de uma *Virtual Private Network (VPN)*, este recurso possibilitou realizar o acesso aos servidores do Code Combat a partir de outros países. Assim contornou-se este problema potencialmente relacionado a estrutura da rede de computadores do Brasil.

4.1.3 Programação do Code Combat

O passo seguinte correspondeu à modificação do Code Combat para o desenvolvimento das páginas necessárias. Contudo, ao longo do desenvolvimento do código da página de Ranking da Turma, observou-se que esta não apresentou os resultados esperados no perfil do estudante; por outro lado, a página apresentou o conteúdo satisfatório no perfil do professor. Após nova consulta à comunidade de desenvolvedores, concluiu-se que isto ocorreu porque muitas informações dos estudantes devem ser restritas aos administradores e por professores.

Neste sentido, os estudantes não possuíam o privilégio de acesso às informações necessário para compor a página em questão. Por outro lado, a partir do código desenvolvido, eles poderiam ter acesso a outro conjunto de informações que, segundo os desenvolvedores do Code Combat (vide seção 4.1.3), são restritas aos professores e administradores, como ID, e-mail, etc. Esta falha e modo como obter as informações dos usuários foram reportadas aos engenheiros do Code Combat e será solucionada por eles.

Nessa perspectiva, uma segunda versão da página do Ranking (Figura 13) foi desenvolvida. Essa página, por sua vez, foi modificada de modo que conseguiu coletar as informações necessárias para criar o ranqueamento de alunos.

A classificação é realizada a partir do progresso individual de cada estudante, ou seja, conforme o estudante avança na conclusão das missões ele consegue 1 (um) ponto no ranking. O desempate é realizado pelo menor tempo de solução. Quanto maior é o número de missões (níveis) concluídos por um aluno e menor for o tempo de conclusão, melhor será sua classificação. A quantidade de missões concluídas é representada nesta imagem pelo termo “Níveis Completos”. O ranking foi desenvolvido para os dois mapas disponíveis para o G2. A Figura 13 ilustra o ranking de cada um dos cursos CS1 e CS2.

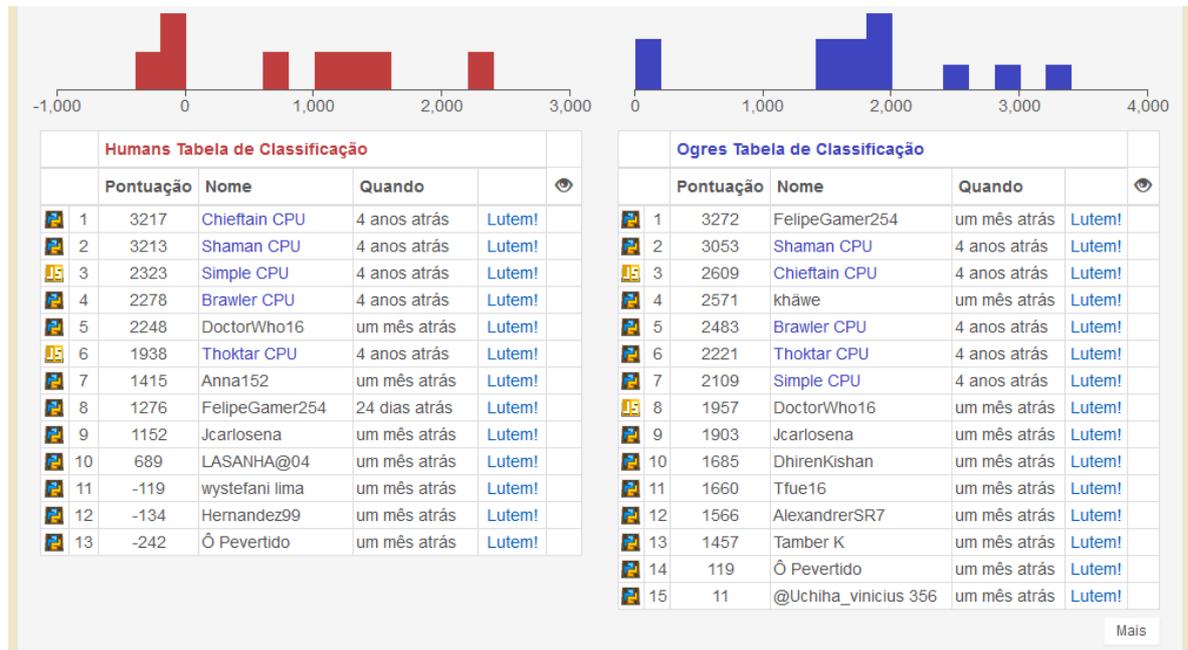
Figura 13 - Ranking dos cursos do G2.

Introdução à Ciência da Computação			Ciência da Computação 2		
Posição	Jogadores	Níveis Completos	Posição	Jogadores	Níveis Completos
1	 destruidor#0	38	1	 FelipeGamer254	73
2	 Tfuе16Legend	38	2	 Ô Pervertido	23
3	 DoctorWho16	38	3	 DhirenKishan	22
4	 LASANHA@04	38	4	 Anna152	20
5	 FelipeGamer254	38	5	 LASANHA@04	16
6	 @Uchiha_vinicius 356	37	6	 Hernandez99	15
7	 DhirenKishan	37	7	 DoctorWho16	14
8	 Ô Pervertido	37	8	 Jcarlosena	13
9	 wstefani lima	35	9	 Tamber K	12
10	 AlexandrerSR7	34	10	 AlexandrerSR7	11
11	 Tamber K	32	11	 Tfuе16Legend	10
12	 Anna152	31	12	 destruidor#0	7
13	 capitão codecombat	30	13	 Livia.R.V	5
14	 Hernandez99	29	14	 wstefani lima	5
15	 Livia.R.V	27	15	 khawe	4
16	 Jcarlosena	27	16	 capitão codecombat	2
17	 khawe	26	17	 Michel194	0
18	 Stoico	25	18	 @Uchiha_vinicius 356	0
19	 Michel194	15	19	 Stoico	0

Fonte: www.codecombat.com

Um outro recurso utilizado durante o experimento foi a arena, que disponibiliza um ranking de pontuação neste recurso (Figura 14). O acesso à arena é liberado após a conclusão da última missão de cada curso. A pontuação deste ranking é calculada a partir da solução desenvolvida por cada estudante nesta missão. Compõe este ranking os resultados de alguns bots, que, em geral, possui em seu nome a termo “CPU”.

Figura 14 - Ranking da arena do CS1.



Fonte: www.codecombat.com

4.1.4 Parcerias consolidadas

Ao longo do estudo e modificação do Code Combat, foram conquistados alguns parceiros valiosos para a realização do estudo. Entre eles pode-se citar: Nick Winter, Andrew Jakubowicz e Zolfer Figueiredo.

Zolfer Figueiredo é um desenvolvedor brasileiro que faz parte da comunidade de colaboradores voluntários do Code Combat. Junto ao jogo, ele atuou principalmente ajudando a traduzir partes do ambiente para o idioma Português do Brasil. Sua participação na colaboração deste estudo correspondeu a testar a execução Code Combat no Brasil através de um servidor local. Este desenvolvedor também não obteve êxito em executar o modo proxy do jogo, corroborando à teoria das limitações do jogo envolvendo a execução do modo proxy na rede de computadores do Brasil.

Andrew Jakubowicz é um dos engenheiros de softwares da equipe do Code Combat, ele ajudou na fase das etapas de instalação e execução do jogo, e ainda na identificação de acessos indevidos a informações sobre os estudantes.

Nick Winter é o principal parceiro deste estudo, é cofundador e *Chief Executive Officer* (CEO) da *startup* Code Combat Inc., responsável pelo jogo Code

Combat. Ele Colaborou em todas as etapas do uso e modificação do jogo, disponibilizou a hospedagem das modificações do jogo nos servidores da empresa, encaminhou dúvidas e demandas relacionadas a testes para sua equipe de engenheiros e ofertou 50 licenças para acesso aos cursos por todos os envolvidos no experimento por um prazo de 90 dias.

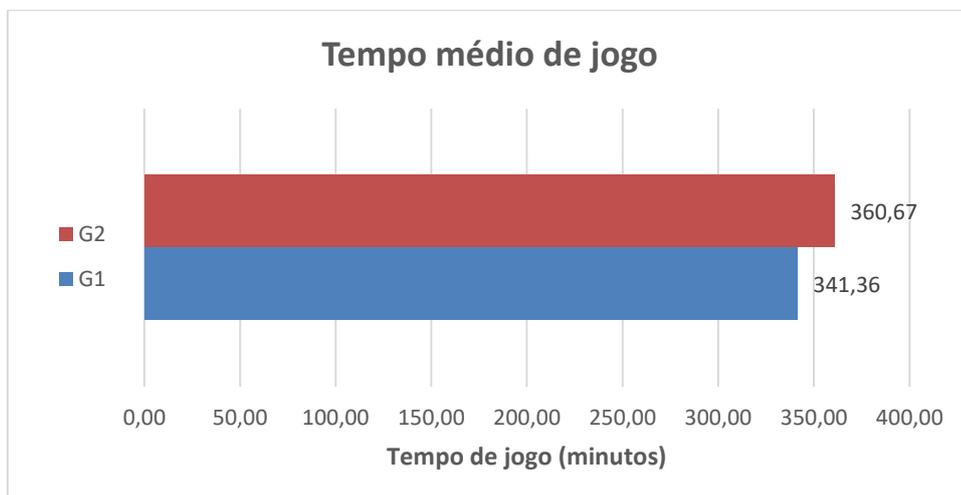
4.2 ANÁLISE DOS DADOS

Os dados desse estudo foram produzidos por 38 alunos. Contudo, somente 35 conduziram as atividades até o fim do experimento, portanto, esse é o valor final da amostra computada. Os 3 (três) estudantes que deixaram de frequentar os encontros presenciais, e não realizaram acessos posteriores ao jogo, foram classificados como “evadidos”, portanto, seus dados não foram computados em seus respectivos grupos.

Entre esses alunos que evadiram do minicurso de introdução a programação de computadores, 2 (dois) faziam parte do G1 e 1 (um) integrava o G2. Em razão disso, a amostra real computada para o G1 foi 17 (dezesete) estudantes e para o G2 foi 18 (dezoito).

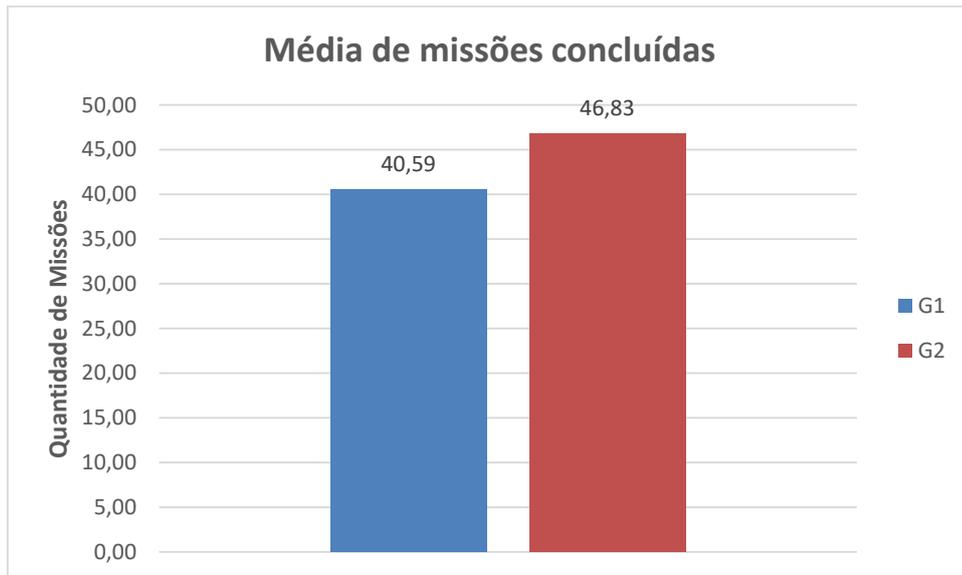
Após a coleta dos dados, foi feita uma verificação da completude dos questionários respondidos durante a realização do pós-teste. Os dados individuais dos discentes também foram extraídos dos registros feitos pelo Code Combat. Os dados foram analisados para obtenção dos resultados da avaliação, apresentada nesta seção.

Figura 15 - Resultados sobre o tempo médio de acesso ao jogo.



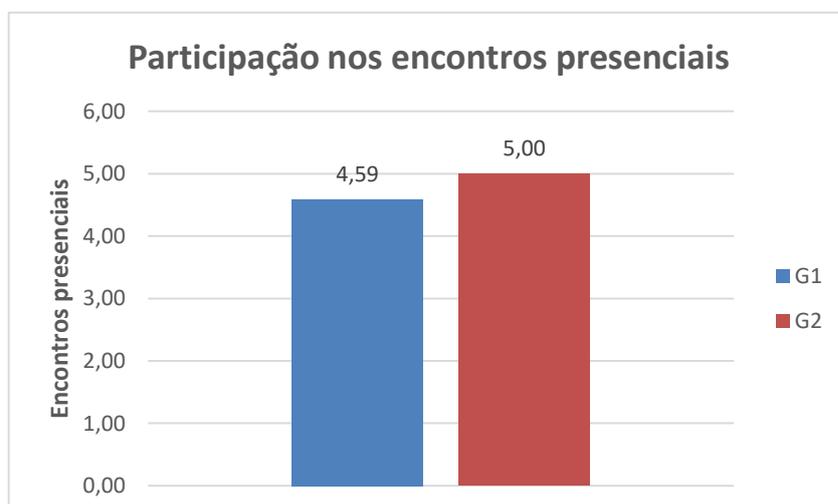
A Figura 15 apresenta o gráfico de tempo médio de jogo. Pode-se observar que os alunos do G2 permaneceram em média 360,67 minutos conectados ao jogo, enquanto os estudantes do G1 estiveram em média 341,36 minutos durante o período do experimento. Estes dados mostram 5,66% de diferença de tempo entre os grupos.

Figura 16 - Resultado sobre a quantidade média de missões concluídas.



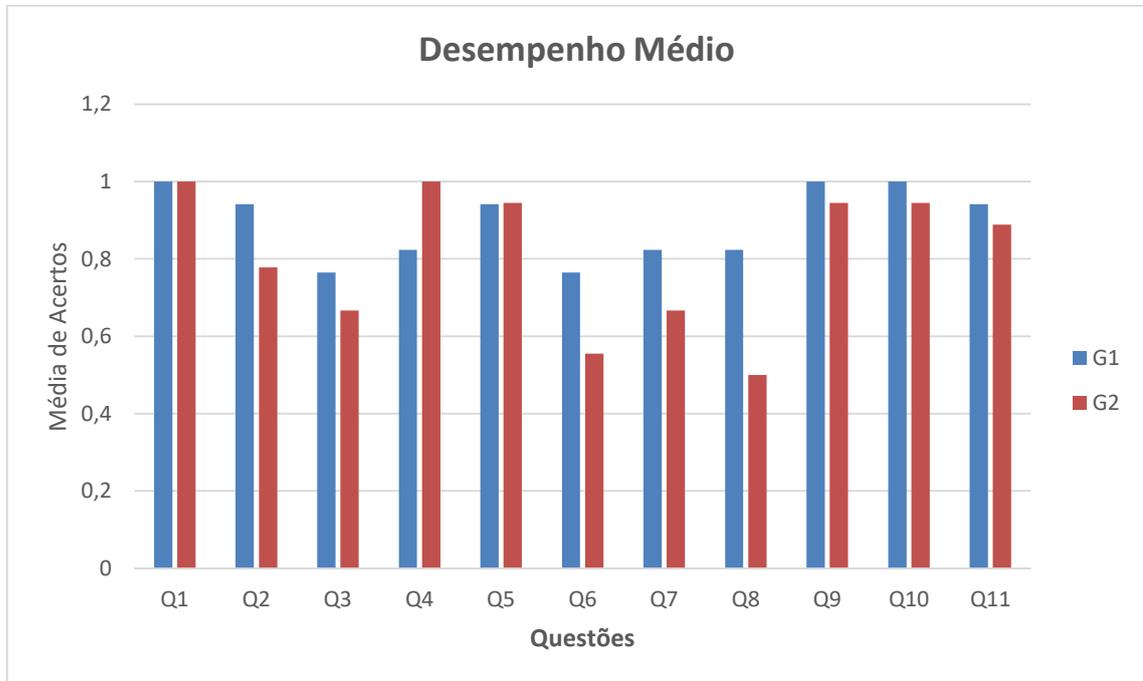
A Figura 16 deixa evidente que os alunos do G2 realizaram mais missões do Code Combat que os alunos do G1. Vale observar que a soma do quantitativo de missões entre o primeiro e o segundo mapa corresponde a 111 (cento e onze). A Figura dezesseis mostra uma discrepância de 15,4% de missões completas entre o G1 e G2.

Figura 17 - Participação nos encontros presenciais do minicurso.



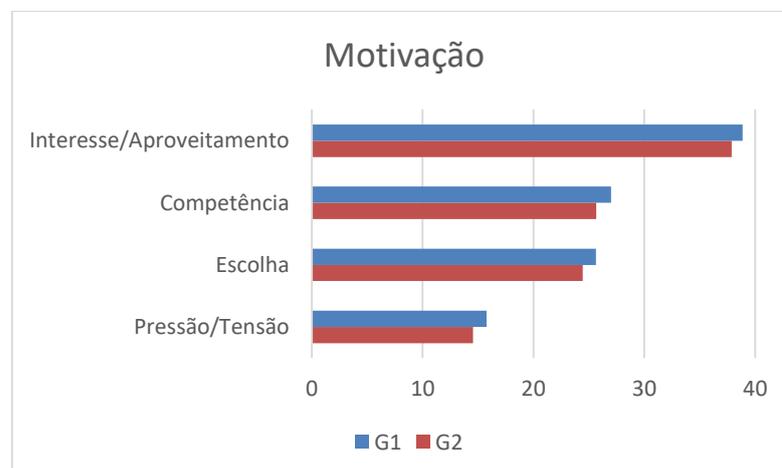
Pode observar a partir da análise da Figura 17 maior participação nos encontros presenciais pelo G2.

Figura 18 - Resultado sobre o desempenho médio dos grupos.



Em contrapartida, a análise do teste de retenção de conhecimento, ou pós-teste, revelou que o G1 obteve maior abstração do conteúdo relacionado a lógica de programação abordado pelo Code Combat. Logo, apesar dos alunos G2 terem apresentado tempo de jogo maior que os do G1, estes últimos alcançam maior média de acertos, visto que das 11 (onze) questões deste teste eles se destacaram em 8 (oito).

Figura 19 - Média de pontos obtidos pelo questionário motivacional IMI.



Na figura 19, é observado que o G1 obteve maior pontuação de Interesse/Aproveitamento em relação ao G2. Contudo, este grupo sentiu mais Pressão/Tensão que o outro. O G2 sentiu menor poder de Escolha em relação às missões realizadas. Este grupo também obteve menor pontuação de Competência em relação ao G1.

Os gráficos das figuras 15, 16 e 17 realçam a influência do ranking desenvolvido no espírito de competitividade entre os estudantes, em consequência, isto resultou no aumento do engajamento dos discentes do G2 sobre dos G1, ou seja, aumento de tempo de jogo, missões concluídas e participação nos encontros presenciais.

4.3 TESTE DAS HIPÓTESES

Para realizar os testes das hipóteses apresentadas nas subseções 3.4, foi executado o *Teste t de Student*, um teste paramétricos usado para comparar duas amostras independentes (PEDRO, 2016).

Tabela 7 - Valores obtidos no teste t de Student.

		<i>t cálculo</i>	<i>t crítico</i>	<i>p-valor</i>	<i>gl</i>
Engajamento	Tempo de jogo	-0,46991	2,03451	0,64201	33
	Missões concluídas	-1,19311	2,03451	0,24132	33
	Encontros presenciais	-1,13284	2,03451	0,26544	33
Desempenho	Acertos	0,48725	2,07387	0,63090	22
Motivação	Pressão/Tensão	0,13818	2,30600	0,89351	8
	Escolha	-0,05208	2,30600	0,95974	8
	Competência	-0,65232	2,30600	0,94949	8
	Interesse/Aproveitamento	-0,04955	2,30693	0,96078	32

No cálculo utilizado para obter os valores descritos pela Tabela 7, foi utilizado o valor 0,05 (5%) no nível de significância (α) e o *gl* (grau de liberdade) apresentado na última coluna da tabela.

A partir da análise dos valores descritos na Tabela 7, pode-se observar que todos os resultados *t cálculo* são inferiores que o *t crítico*, e o *p-valor* é maior que nível de significância. Este resultado revela que a hipótese H_0 foi aceita e a H_1 foi rejeitada. Ou seja, estatisticamente, segundo cálculo do Teste t de Student, não foram identificadas diferenças significativas, no experimento realizado neste estudo, em termos de engajamento, motivação e desempenho na utilização de ambientes

gamificados que fazem uso de elementos monousuário em ambientes gamificados que façam uso de elementos multiusuários no ensino-aprendizagem da lógica de programação de computadores.

4.4 DISCUSSÃO DOS RESULTADOS

A análise quantitativa apresentada na subseção 4.2 mostra que o G2 obteve resultados melhores que o G1 no engajamento, ou seja, os alunos deste grupo permaneceram mais tempo no jogo, realizaram maior quantidade de missões e tiveram maior participação nos encontros presenciais. Mesmo que o teste de hipóteses, presente na subseção 4.3, tenha apontado que, estatisticamente, não houve diferenças significativas no engajamento dos dois grupos. A partir da observação ao longo do experimento, pôde-se constatar maior entusiasmo no G2, isto geralmente estava relacionado a assuntos relacionados ao ranking do grupo. Um dos alunos desse grupo ficou demasiadamente interessado na dinâmica do jogo, o que o levou a solucionar a maioria das missões do primeiro mapa ainda no dia do primeiro encontro. Este resultado se deu em razão do estudante possuir computador e acesso à Internet em sua residência. Mas vale observar que a maioria dos estudantes, de ambos os grupos, durante o período de realização do experimento, não possuíam acesso à Internet em casa.

Ainda sobre os alunos do G2, foi possível presenciar, em vários momentos distintos, estudantes perguntando aos seus colegas sobre sua atual posição no ranking, na expectativa de saber se conseguiu superá-los ou distanciar em número de conclusão de missões. Isso demonstrou as mudanças comportamentais desse grupo em relação ao G1, conforme afirmado por Peixoto, Santos e Resende (2015) sobre os recursos multiusuários realçarem comportamentos relacionados à sociabilidade e competição.

O resultado correspondente ao desempenho médio, apresentado na Figura 18, revela que o G1 obteve maior quantidade de acertos que o G2 no pós-teste. Especula-se que esse cenário é resultado do nível de competitividade maior no G2, em razão da concorrência por melhores posições no ranking. Isso pôde ser observado durante o experimento, alguns dos estudantes estavam mais concentrados em finalizar rapidamente cada nova missão apresentada pelo jogo do que da aprendizagem dos conceitos abordados por este.

A Figura 19 também apresenta o destaque para o G1. Este grupo obteve melhor desempenho no Interesse/Aproveitamento, Competência e Escolha, contudo o G2 sentiu-se menos pressionado. Pode-se especular que a atmosfera de competição, presente no G2, foi favorável para a criação de um ambiente mais propício a socialização dos estudantes, e, conseqüentemente, diminuir a tensão entre estes. O poder de escolha do G1 foi superior ao do G2, possivelmente em razão da percepção dos alunos sobre a necessidade de fazer mais missões para subir nas posições do ranking. A competência inferior dos alunos do G2 em relação ao G1 pode sugerir a possibilidade dos alunos do grupo estarem mais concentrados em subir de classificação do que necessariamente dominar o conteúdo abordado pelo jogo. Entretanto o menor interesse dos alunos do G2 pelo jogo, constatado pelo teste motivacional (Figura 19), se opõe ao que se espera do observado nos outros aspectos já citados, talvez isso seja consequência de uma necessidade de realização de mapeamento do perfil de cada estudante para aprofundar em suas preferências quanto a abordagem dos elementos do jogo. Para a confirmação dessas hipóteses é necessário a realização de novos experimentos. Entretanto, o teste de hipóteses também não apontou diferenças significativas nestes nos valores encontrados.

Ao longo dos encontros presenciais pôde-se observar que os discentes sentiam-se mais confortáveis em terem suas dúvidas sanadas pelos monitores (alunos de turmas dos anos seguintes) e colegas da turma do que pelo professor. Possivelmente em razão deles se identificarem mais pelos indivíduos da sua faixa etária de idade e que tenham experiências semelhantes. Em alguns alunos observou-se a responsabilidade e preocupação em definir os comandos corretos para cumprir cada missão sem causar eventuais danos ao personagem que se estava controlando, demonstrando sua atenção pelo estado de saúde deste.

Todos os resultados encontrados levaram a rejeição da hipótese H_1 . Portanto, para questão levantada neste estudo (quais as evidências que o uso de ambientes gamificados multiusuários são significativamente melhores que ambientes gamificados monousuário com a finalidade de aumentar o engajamento e motivação do discente no processo de ensino-aprendizagem da lógica de programação de computadores?) é possível concluir que, através do experimento realizado, não foram encontradas evidências substanciais para afirmar ambientes gamificados

multiusuários são significativamente melhores que ambientes gamificados monousuários no contexto estudado.

Mesmo com este resultado é possível afirmar que ambientes gamificados monousuário, bem como multiusuários, são eficazes no ensino aprendizagem da lógica de programação de computadores, assim como relatados nos trabalhos relacionados na subseção 2.4.1.

5 CONCLUSÕES

Através do desenvolvimento deste trabalho, pode-se observar, na prática, os benefícios descritos nos resultados de estudos que examinaram a aplicação da gamificação no ensino-aprendizagem da lógica de programação de computadores, como no relato de Zanchett, Vahdick e Raabe (2015), quando descrevem que os objetivos educacionais podem ser atingidos enquanto os estudantes se divertem. É importante frisar que esses estudos tiveram como alicerce de suas afirmações dados extraídos de um ambiente gamificado que possui, prevalentemente, elementos monousuário.

O experimento também se alinha com o estudo de Penteado e Damasceno (2014), uma vez que se pode concluir que de fato a gamificação traz benefícios e contribuições no ensino-aprendizagem da lógica de programação de computadores. Um dos benefícios observados que se pode citar é o entusiasmo dos discentes ao fazer uso de um recurso metodológico e lúdico que está fortemente relacionado ao contexto extraclasse que muitos participam.

Diante dos resultados apresentados, pode-se concluir que após a adição do elemento multiusuário, ranking de líderes, no Code Combat, não houve significativo aumento no engajamento, desempenho e motivação dos discentes em relação a sua ausência no ambiente gamificado. Isto sugere que ambas as abordagens são apropriadas para o ensino-aprendizagem da lógica de programação de computadores. Ainda assim, através da análise dos dados obtidos, pôde-se observar que o G2 apresentou um menor número de evasão de alunos em relação ao G1. Portanto, não é plausível afirmar, mas especular, que, apesar de estatisticamente não haver diferenças nos resultados dos grupos, os alunos do G2 se mostraram mais atraídos pelo minicurso do que o G1.

Foi observado também que o aumento do engajamento dos discentes não implica necessariamente no aumento significativo da abstração dos conceitos abordados pelo ambiente gamificado, uma vez que os aspectos relacionados a competição podem estimular a necessidade do estudante de se sobressair sobre seus adversários de ranking através da tentativa em resolver os desafios mais rapidamente.

As parcerias formadas ao longo do estudo foram fundamentais para a realização do mesmo, uma vez que estas influenciaram diretamente desde a instalação e desenvolvimento da página de ranking até a hospedagem dos recursos desenvolvidos e disponibilização das licenças para executar o experimento. O Code Combat se mostrou um ambiente muito atrativo para os discentes, em razão de sua mecânica, enredo, design e feedback ao usuário.

Sugestões de trabalhos futuros incluem o desenvolvimento de um outro ambiente, para deste modo não haver limitações quanto ao uso de licenças. Assim, pretende-se realizar novos estudos de casos, agregando novos recursos multiusuários, como elementos de colaboração (participação de clãs ou alianças), eventos de competição (torneios) e promover a participação de maior quantidade de grupos. O foco estará em elucidar mais profundamente as características dos elementos multiusuários e seu impacto no engajamento e abstração dos conceitos abordados, escalabilidade das técnicas gamificadas utilizadas e análise do público alvo em potencial.

REFERÊNCIA BIBLIOGRÁFICAS

ALVES, Flora. **Gamification como Criar Experiências de Aprendizagem Engajadoras um Guia Completo: Do Conceito à Prática**. 2ª ed. São Paulo. DVS Editora, 2015.

ALTET, M. **As pedagogias da aprendizagem**. Instituto Piaget, Lisboa. 1999.

ANJOS, Cleverson S. dos; DUDA, Rodrigo; SILVA, Sani de C. R. da. Desafios para o ensino da programação no ensino fundamental. In: Simpósio Nacional de Ensino de Ciência e Tecnologia, V SINECT, 2016. **Anais...** Ponta Grossa-PR. 2016

AURELIANO, V. C. O.; TEDESCO, P. C. A. R. “Ensino-aprendizagem de Programação para Iniciantes: uma Revisão Sistemática da Literatura focada no SBIE e WIE”. In: Simpósio Brasileiro de Informática na Educação, 2012, Curitiba, **Anais...** Curitiba. 2012.

BIGGS, John. **Teaching for Quality Learning at University**. 4ª ed., Open University Press, 2011.

BITTENCOURT, Roberto A.; ROCHA, Anderson S.; SANTANA, Bianca L.; SANTANA, Carla S.; CARNEIRO, Douglas A.; BORGES, Gutemberg A.; CHALEGRE, Henderson S.; SILVA, Jadson. SANTOS, Jéssica; SILVA, Lucas A.; ANDRADE, Pedro. Aprendizagem de Programação Através de Ambientes Lúdicos em um Curso de Engenharia da Computação: Uma Primeira Incursão. In : XXI Workshop sobre de Educação em Computação (WEI). 2013. Maceió-AL. **Anais...** Maceió. 2013.

CASTRO, Ronney M.; SOUZA; Gleison. O Uso de Recursos Lúdicos Para o Ensino de Processos em Engenharia de Software. In: Workshop sobre Educação em Computação, 24., 2016, Porto Alegre. **Anais...** Porto Alegre. 2016.

CODECOMBAT. **Sobre o Code Combat**. Disponível em: <<https://codecombat.com/about>>. Acesso em: 02 abril 2019.

DETERDING, S.; DIXON, D.; KHALED, R.; e NACKE, L. **From game design elements to gamefulness: defining "gamification"**. Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, p. 9-15. 2011.

DUARTE, A. M. **Aprendizagem, Ensino e Aconselhamento Educacional**. Porto: Porto Editora, 2002.

DUARTE, A. M., CABRITO, B., FIGUEIRA, A. I.; MONGE, J. **Teaching practices for passive and active learning in rural and urban elementary teachers**. Sisyphus: Educational Sciences Journal, v. 3, n. 2, p. 134-154, 2015.

FONTES, Marisa A. Motivação e estratégias de aprendizagem segundo a teoria das abordagens à aprendizagem: implicações para a prática de ensino-aprendizagem. In: Revista Ibero-Americana de Estudos em Educação, **Anais...** V.11, 2016, p 1727-1744.

FRANÇA, Rômulo M.; REATEGUI, Eliseo B. SMILE-BR: aplicação de conceitos de gamificação em ambiente de aprendizagem baseado em questionamento. In: Simpósio Brasileiro de Informática na Educação, **Anais**. V 23, 2013, p 366-375.

GARLET, Daniela; BIGOLIN, Nara; SILVEIRA, Sidnei. **Ensino de Programação de Computadores na Educação Básica: um estudo de caso**. Resiget. Vol 2. 2018.

GALVÃO, T. F.; PEREIRA, M. G. **Revisões sistemáticas da literatura: passos para sua elaboração**. Epidemiol. Serv. Saúde, p. 183-184, 2014, Brasília.

GIL, Antônio Carlos. **Metodologia do ensino superior**. 4 Ed. São Paulo: Atlas, 2012.

GIRAFFA, L. M., MORAES. M. C. (2013) Evasão na disciplina de Algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. **Anais CLABES III**. Disponível em: http://www.clabes2013-alfaguia.org.pa/docs/Libro_de_Actas_III_CLABES.pdf. Acesso em: 04 de julho de 2018.

GOMES, T.; TEDESCO, P. Gamificando a sala de aula: desafios e possibilidades em uma disciplina experimental de Pensamento Computacional no ensino fundamental. In.: Workshop de Informática na Escola, 2017, São Paulo. **Anais...** São Paulo. 2017.

GOMES, Danielle; TAUMATURGO, Maurício. **Quais as melhores práticas para ensinar programação e como inovar no processo de ensino e aprendizagem**. Set. 2017. Disponível em <https://medium.com/cesar-reports/quais-as-melhores->

pr%C3%A1ticas-para-ensinar-programa%C3%A7%C3%A3o-e-como-inovar-no-processo-de-ensino-e-42f65694e7e3. Acesso em 21/05/2018.

JOHNSON, L. et al. **NMC Horizon Report: Edição Educação Básica 2015**. Austin, Texas: The New Media Consortium, 2015.

KAPP, K. M. **Gadgets, games, and gizmos for learning: tools and techniques for transferring know-how from boomers to gamers**. [S.I.]: John Wiley and Sons. 2007.

KAPP, K. M. **The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education**. Pfeiffer & Company. 2012.

LAKATOS, E.; MARCONI, M. D. A. **Fundamentos de metodologia científica**. Atlas, 2010. Disponível em: <<https://books.google.com.br/books?id=Y2WFRAAACAAJ>>.

MATTAR, J. **Games em educação: como os nativos digitais aprendem**. Pearson Prentice Hall, São Paulo. 2010.

MEDEIROS, T. J.; SILVA, T. R.; Aranha, E. H. S. “Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura”. In: Revista Novas Tecnologias na Educação - RENOTE, v.11, n.3. **Anais...** Porto Alegre. 2013.

NETO, W. C. B.; SCHUVARTZ, Aguinaldo. Ferramenta computacional de apoio ao processo de ensino-aprendizagem dos fundamentos de programação de computadores. In: Simpósio Brasileiro de Informática Na Educação, v. 18, 2007, São Paulo. **Anais...** São Paulo. 2007.

NORMANDO, David; TJÄDERHANE, Leo; QUINTÃO, Cátia. A escolha do teste estatístico: um tutorial em forma de apresentação de Power Point. In: Dental Press J. Orthod, v. 15, no. 1, p. 101-106, 2010. **Anais...** Maringá. 2010.

PEA, Roy D. **Language-Independent Conceptual “Bugs” in Novice Programming**. J. Educational Computing Research, Vol. 2. 1986.

PEDRO, Laís Z. **Uso da Gamificação em ambientes virtuais de aprendizagem para reduzir o problema da externalização de comportamentos indesejáveis**. Dissertação (Dissertação em Ciência da Computação e Matemática Computacional) – USP. São Paulo, p 152. 2016.

PEIXOTO, Daniela C. C.; SANTOS, Silas T.; RESENDE, Rodolfo F. Avaliação de Jogos Digitais Educacionais Multiusuários: Uma Revisão Sistemática da Literatura. In: Simpósio Brasileiro de Informática na Educação – SBIE, v. 26, 2015. **Anais...** Maceió-AL. 2015.

PENTEADO, C.; DAMASCENO, E. Contribuições de jogos digitais no ensino da lógica de programação para o ensino médio integrado em informática. Revista E.T.C. Logomarca. Edição 01. 2014.

PRESTES, Maria. **A pesquisa e a Construção do Conhecimento Científico: Do planejamento aos textos, da escola à academia.** 4ª ed. São Paulo: Rêspel Editora, 2012, p. 310.

SANTOS, Ribamar A. dos. **Gamificação no Ensino-Aprendizagem de Algoritmos e Lógica Aplicada a Computação.** Dissertação (Dissertação em Ciência da Computação) – FACCAMP. São Paulo, p 160. 2018.

SENTANCE, Sue; CSIZMADIA, Andrew. **Computing in the curriculum: Challenges and strategies from a teacher's perspective.** Education and Information Technologies, p. 1-27, 2016.

SCHLEMMER, E. e LOPES, D. Q. **Avaliação da aprendizagem em processos gamificados: desafios para apropriação do método cartográfico.** In: Jogos digitais e aprendizagem: fundamentos para uma prática baseada em evidências. Alves, L. e Coutinho, I. J (Orgs.). 2016. Campinas: Editora Papirus.

SILVA, T.; MEDEIROS, T.; ARANHA, E. Jogos Digitais e Aprendizagem de Programação: uma revisão sistemática da literatura. In.: Simpósio Brasileiro de Informática na Educação, p. 692-701, 2014, Brasília. **Anais...** Brasília, 2014.

SHELDON, L. **The Multiplayer Classroom: Designing Coursework as a Game.** Cengage Learning, Boston, MA., 2012.

TAORMINA, R. J. GAO, J. H. **Maslow and the motivation hierarchy: measuring satisfaction of the needs.** The American journal of psychology, v. 126, n. 2, jan 2013. Disponível em www.ncbi.nlm.nih.gov/pubmed/23858951. Acesso 28/07/2018.

VASSILEVA, J. **Motivating participation in social computing applications: a user modeling perspective.** User Model User-Adapted Interaction, v. 22, n. 1, p. 177-201, 2012. Disponível em: <https://dx.doi.org/10.1007/s11257-011-9109-5>. Acesso em: 27/07/2018.

WERBACH, K.; HUNTER, D. **For the win: how game thinking can revolutionize your business.** Philadelphia: Wharton Digital Press, 2012.

ZANCHETT, G.; VAHLDICK, A.; RAABE, A. Jogos de Programar como uma Abordagem para os Primeiros Contatos dos Estudantes com a Programação. In.: Workshop do Congresso Brasileiro de Informática na Educação, 2015, Recife. **Anais...** Recife, 2015.

APÊNDICE A – ARTIGOS INCLUÍDOS NA REVISÃO SISTEMÁTICA

- E1:**RAPOSO, E. H. S.; DANTAS, V. O Desafio da Serpente - Usando gamification para motivar alunos em uma disciplina introdutória de programação In.: Simpósio Brasileiro de Informática na Educação, 2016, Porto Alegre.
- E2:**NAGAI, W.; IZEKI, C.; DIAS, R. Experiência no Uso de Ferramentas Online Gamificadas na Introdução à Programação de Computadores. In.: Workshop de Informática na Escola, 2016, Porto Alegre.
- E3:**GOMES, T.; TEDESCO, P. Gamificando a sala de aula: desafios e possibilidades em uma disciplina experimental de Pensamento Computacional no ensino fundamental. In.: Workshop de Informática na Escola, 2017, São Paulo.
- E4:**FRANÇA, R.; AMARAL, H. Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch. In.: Workshop de Informática na Escola, 2013, Maceió.
- E5:**PANTELÃO, E.; AMARAL, L.; SILVA, G. Uma abordagem baseada no ambiente Robocode para ensino de programação no Ensino Médio. In.: Revista Brasileira de Informática na Educação, 2017, São Paulo.
- E6:**MARTINS, R.; MARQUES, A. Inserção da programação no ensino fundamental Uma análise do jogo Labirinto Clássico da Code.org através de um modelo de avaliação de jogos educacionais. In.: Workshop de Informática na Escola, 2016, Porto Alegre.
- E7:**RAFALSKI, J.; SANTOS, O. Uma experiência com a Linguagem Scratch no Ensino de Programação com Alunos do Curso de Engenharia Elétrica. In.: Workshop de Informática na Escola, 2016, Porto Alegre.
- E8:**AGUIAR, J. Experiência baseada em Gamificação no Ensino sobre Herança em Programação Orientada a Objetos. In.: Workshop do Congresso Brasileiro de Informática na Educação, 2015, Recife.
- E9:**RODRIGUES, L; NOGUEIRA, G; QUEIROGA, A. Experiências no ensino de Programação Orientada a Objetos: RoboCode, Greenfoot e Jogos de Tabuleiro no Ensino Superior. In.: Workshop de Informática na Escola, 2017, São Paulo.

- E10:** DUARTE, K.; SILVEIRA, T.; BORGES, M. Abordagem para o Ensino da Lógica de Programação em Escolas do Ensino Fundamental II através da Ferramenta Scratch 2.0. In.: Workshop de Informática na Escola, 2017, São Paulo.
- E11:** SILVA, J.; GENS, C.; SANTOS, C. Lógica de Programação: Iniciação Lúdica com Play Code Dog. In.: Workshop de Informática na Escola, 2017, São Paulo.
- E12:** BRITO, A.; MADEIRA, C. XP & Skills: gamificando o processo de ensino de introdução a programação. In.: Workshop do Congresso Brasileiro de Informática na Educação, 2015, Recife.
- E13:** ZANCHETT, G.; VAHLDICK, A.; RAABE, A.. Jogos de Programar como uma Abordagem para os Primeiros Contatos dos Estudantes com a Programação. In.: Workshop do Congresso Brasileiro de Informática na Educação, 2015, Recife.
- E14:** LAGUNA, A.; TORRENTE, J.; IGLESIAS, B.; MANJÓN, B. Building a Scalable Game Engine to Teach Computer Science Languages. In: IEEE Revista Iberoamericana de Tecnologías Del Aprendizaje, Vol. 10, N. 4, 2015.
- E15:** MALLIARAKIS, C.; SATRATZEMI, M.; XINO GALOS, S. CMX: The Effects of an Educational MMORPG on Learning and Teaching Computer Programming. In: IEEE Transactions on Learning Technologies, Vol. 10, N. 2, 2017.
- E16:** SHIN, J.; KWON, D.; LEE, W. The Effects of a Robot Game Environment on Computer Programming Education for Elementary School Students. In: IEEE Transactions on Education, Vol. 60, N. 2, 2017.
- E17:** WONG, Y.; YATIM, M.; TAN, W. Use computer game to learn Object-Oriented programming in computer science courses. In: IEEE Global Engineering Education Conference (EDUCON), Istanbul. 2014.
- E18:** LEE, M.; BAHMANI, F.; KWAN, I.; LAFERTE, J.; CHARTERS, P.; HORVATH, A.; LUOR, F.; CAO, J.; LAW, C.; BESWETHERICK, M.; LONG, S.; BURNETT, M.; KO, A. Principles of a debugging-first puzzle game for computing education. In: IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2014.
- E19:** STIGALL, J.; SHAMA, S. Virtual reality instructional modules for introductory programming courses. In: IEEE Integrated STEM Conference (ISEC), 2017.

- E20:** REGALADO, M.; ARANHA, E.; SILVA, T. Gamifying an online approach for promoting game development Learning and contest: An experience report. In: IEEE Frontiers in Education Conference (FIE), 2016.
- E21:** DORELO, A.; ZIPITRÍA, S. Studying novice learners knowledge about program execution. In: XLIII Latin American Computer Conference (CLEI), Cordoba-Argentina, 2017.
- E22:** CABEZAS, I. On combining gamification theory and ABET criteria for teaching and learning engineering. In: IEEE Frontiers in Education Conference (FIE), El Paso, TX, USA, 2015.
- E23:** BITTENCOURT, R.; SANTOS, D.; RODRIGUES, C.; BATISTA, W.; CHALEGRE, H. Learning programming with peer support, games, challenges and scratch. In: IEEE Frontiers in Education Conference (FIE), El Paso, TX, USA, 2015.
- E24:** GOYAL, S.; CHOPRA, S.; MOHANAN, D. CodeFruits Teaching Computational Thinking Skills Through Hand Gestures. In: CHI PLAY'17, Amsterdam, NL. 2017.
- E25:** HERMANS, F.; AIVALOGLOU, E. Teaching software engineering principles to K-12 students a MOOC on scratch. In: 39th International Conference on Software Engineering: Software Engineering and Education Track. Buenos Aires, Argentina. 2017.
- E26:** ARAWJO, I.; WANG, C.; MYERS, A.; ANDERSEN, E.; GUIMBRETIERE, F. Teaching Programming with Gamified Semantics. In: Conference on Human Factors in Computing Systems. Denver, USA. 2017.
- E27:** EORDANIDIS, S.; GEE, E.; CARMICHAEL, G. The effectiveness of pairing analog and digital games to teach computer science principles to female youth. In: Journal of Computing Sciences in Colleges. 2017.
- E28:** UNNIKRISHAN, R.; AMRITA, N.; MUIR, A.; RAO, B. Of Elephants and Nested Loops: How to Introduce Computing to Youth in Rural India. In: The 15th International Conference on Interaction Design and Children. Manchester, United Kingdom. 2016.
- E29:** LEE, E.; SHAN, V.; BETH, B.; LIN, C. A structured approach to teaching recursion using cargo-bot. In: Proceedings of the tenth annual conference on International computing education research, Scotland, United Kingdom. 2014.

- E30:** ZHANG, J.; SIMITH, E.; CALDWELL, E.; PERKINS, M. Learning and practicing decision structures in a game. In: Journal of Computing Sciences in Colleges, Vol. 29. 2014.
- E31:** ESPER, S.; WOOD, S.; LERNER, S.; GRISWOLD, W. Codespells how to design quests to teach java concepts. In: Journal of Computing Sciences in Colleges, Vol. 29. 2014.
- E32:** TESSLER, J.; BETH, B.; LIN, C. Using cargo-bot to provide contextualized learning of recursion. In: International computing education research. San Diego, USA. 2013.
- E33:** GARDINER, C.; CARMICHAEL, G.; LATHAM, J.; LOZANO, N.; GREENE, J. Influencing middle school girls to study computer science through educational computer games. In: Journal of Computing Sciences in Colleges, Vol. 28. 2013.
- E34:** ROGERS, M.; WESSEL, R. The joy of text word games in CS I. In: Journal of Computing Sciences in Colleges, Vol. 28. 2013.

APENDICE B – INTRINSIC MOTIVATION INVENTORY – IMI (PORTUGUÊS)

Para cada frase, por favor indique o quão verdadeira é para você, utilizando a seguinte escala:

1	2	3	4	5	6	7
não é verdade		um pouco verdade			Verdade	

1. Enquanto eu estava jogando, eu estava pensando o quanto estava aproveitando.
2. Eu não me senti nervoso(a) enquanto jogava.
3. Eu senti que foi minha escolha cumprir as missões.
4. Eu acho que sou muito bom com esse jogo.
5. Eu achei esse jogo muito interessante.
6. Eu me senti tenso enquanto jogava esse jogo.
7. Eu acho que me sai bem nesse jogo, comparado com meus colegas.
8. Jogar esse jogo foi divertido.
9. Eu me senti tranquilo(a) jogando esse jogo.
10. Eu gostei muito de jogar esse jogo.
11. Eu não tive muita escolha em cumprir as missões do jogo.
12. Eu estou satisfeito com o meu desempenho nesse jogo.
13. Eu estava ansioso(a) durante o jogo.
14. Eu achei esse jogo muito chato.
15. Eu senti que eu estava fazendo o que eu queria, enquanto eu estava jogando.
16. Me senti muito competente nesse jogo.
17. Eu achei o jogo muito divertido.
18. Eu me senti pressionado enquanto jogava o jogo.
19. Eu senti obrigação de cumprir as missões.
20. Eu despreveria essa atividade como muito agradável.
21. Eu joguei esse jogo porque eu não tive escolha.
22. Depois de jogar esse jogo por um tempo, me senti muito eficiente nisso.

APÊNDICE C – TESTE DE RETENÇÃO DE CONHECIMENTO



REFLEXÃO SOBRE O JOGO

Nome: _____ Grupo: _____

1. A ordem em que os comandos são escritos em um código é importante? Porque?

2. Um humano pode entender as instruções mesmo se houver um erro no código?

3. Um computador consegue entender as instruções caso houver um erro no código?

4. Defina o conceito da estrutura chamada “loop”.

5. Observe o código abaixo e explique o que ele faz.

```
while true:
    hero.moveRight ()
```

```
hero.moveUp()
```

6. O que é uma variável?

7. Como você cria uma variável?

8. Escreva um exemplo de utilização de uma variável.

9. Você pode criar uma variável antes de utilizá-la?

10. O valor de uma variável pode ser alterado?

11. Observe o código abaixo e explique o que ele faz.

```
hero.moveRight()  
hero.moveUp()
```


ANEXO A – GUIA SOBRE OS CONCEITOS ABORDADOS NO PRIMEIRO MAPA



Introdução à Ciência da Computação

Sintaxe Básica (níveis 1-6)

Metas

- Use a sintaxe do Python
- Chamar funções
- Entenda que a ordem é importante

Sintaxe é como escrevemos código. Assim como a ortografia e a gramática são importantes para escrever a prosa, a sintaxe é importante ao escrever código. Os seres humanos são bons em descobrir o que algo significa, mesmo que não seja exatamente correto, mas os computadores não são tão inteligentes e precisam que você escreva sem erros.

- Exemplo de código: `hero.moveRight()`
- Vocabulário: (objeto) (função)
- Leia em voz alta: “hero ponto move right”

Objetos são os blocos de construção do Python. São coisas ou personagens que podem executar ações. Seu herói é um objeto. Pode executar as ações de movimento.

Funções são ações que um objeto pode fazer. `moveRight()` é uma função. Os nomes das funções são sempre seguidos por parênteses. A ordem das funções é importante!

Reflexão

Por que a sintaxe é importante?? (Ela permite que você seja específico sobre exatamente o que você quer que aconteça.)

A ordem importa? (sim)

Um humano pode entender as instruções mesmo se houver um erro na sintaxe? (às vezes)

Um computador consegue entender? (não)

Loops (níveis 7-14)

Metas

- Escreva um loop infinito
- Quebre um problema em pedaços menores
- Decidir quais partes de uma ação se repetem

Um **loop** é uma maneira de repetir o código. Uma maneira de escrever loops é usar a palavra-chave `_while_`, seguida por uma **expressão**, que pode ser avaliada como `True` ou `False`. `_while_` é uma palavra especial que diz ao computador para avaliar (ou resolver) o que vem depois dele e, em seguida, executar as ações abaixo até que a expressão se torne Falsa.

Estes níveis no CodeCombat requerem um **loop infinito**, ou um loop que repete para sempre. Para isso, precisamos de uma expressão que seja sempre verdadeira. Por sorte, `True` é um atalho do Python que sempre é avaliado como Verdadeiro!

Abaixo, `while` é a palavra chave, e `True` é uma expressão

```
while True:
    hero.moveRight() # ação
    hero.moveUp()   # outra ação
```

Você pode colocar quantas linhas de código quiser dentro do loop. Todos eles precisam ser indentados com quatro espaços. É assim que o Python sabe que eles fazem parte do loop. A indentação é uma parte importante do Python! Sempre que você tiver um problema com seu código, verifique primeiro a indentação.

O que é um loop? (uma maneira de repetir ações)

O que é uma expressão? (algo que é verdadeiro ou falso, geralmente usando `=`, `<`, ou `>`)

Como você escreve um loop que nunca termina? (Use `while True`)

Variáveis (níveis 15-20)

Metas

- Criar uma variável
- Usar uma variável como argumento
- Escolher o nome apropriado para a variável

Uma **variável** mantém seus dados para serem usados mais tarde. Você faz uma variável dando-lhe um nome, depois dizendo qual **valor** ela deve manter.

```
enemy = "Kratt"
```

A variável `enemy` mantém (=) o valor "Kratt"

Agora você pode usar sua variável em vez do próprio valor!

```
hero.attack("Kratt") é o mesmo que hero.attack(enemy)
```

Assim, uma variável pode representar um valor.

Variáveis também podem ser alteradas e verificadas. Você poderia dizer "pontuação = 0" e depois "pontuação = 1". Ou você poderia usar sua variável em uma expressão em loop, exemplo `while score < 10`:

Reflexão

Como você cria uma variável? (variável = alguma coisa)

Como você pode usar uma variável? (Representando um valor, verificando em um loop)

Você pode usar uma variável antes de criá-la? (Não, ainda não existe!)

ANEXO B – GUIA SOBRE OS CONCEITOS ABORDADOS NO SEGUNDO MAPA



Ciência da Computação 2

Condicionais (If)

O curso 2 introduz conceitos de programação mais avançados, portanto o progresso nos níveis deve ser mais lento. Preste muita atenção às instruções, para que você saiba qual é o objetivo do nível e para os comentários em linha (denotados com um #) para saber qual código está faltando.

Metas

- Construa uma condicional
- Escolha expressões apropriadas
- Avaliar expressões

Condicionais executam código dependendo do estado do jogo. Eles começam avaliando uma declaração como `True` ou `False`, então eles executam o código somente se a declaração for `True`. Observe que a sintaxe é semelhante a um loop, pois precisa de dois pontos e um recuo de quatro

espaços.

`if` é a palavra-chave e `==` é a expressão

```
if enemy == "Kratt":
    attack(enemy) # Está é a ação
```

A palavra-chave para uma condição é `if`. Um condicional irá funcionar apenas uma vez, mas se você quiser continuar checando, você tem que colocá-lo dentro de um loop. Observe como o recuo funciona.

```
while true:
    if enemy == "Kratt":
        attack(enemy)
```

O código é chamado de código porque estamos codificando nossas ideias em uma linguagem que o computador possa entender. Você pode usar esse processo de três etapas para reformular suas ideias sempre que estiver escrevendo um código. Contanto que você conheça a sintaxe da linguagem de programação, você sabe como deve ser a ideia codificada!

Por que precisamos de condicionais? (nem todas as ações acontecem o tempo todo) **Qual é a parte que vem entre o `if` e os dois pontos?** (uma expressão) **O que é importante saber sobre as expressões?** (Eles precisam ser verdadeiros ou falsos)

Condicionais (Else)

Metas

- Construa uma condicional `if-else`.
- Identificar diferentes ações que ocorrem em diferentes circunstâncias.
- Defina `else` como o oposto de `if`.

Estamos acostumados a usar condicionais para fazer algo se a expressão for Verdadeira , mas e se for Falso ? É aí que `else` entra. `Else` significa se não OU caso contrário OU o oposto .

Observe que `else` deve ser indentado com o mesmo número de espaços que o `if`. E também precisa de dois pontos: `:` como `if` .

Abaixo, `if` e `else` são palavras-chave, e `==` é a expressão

```
if today == weekday:
    goToSchool() # action
else: # keyword
    watchCartoons() # action
```

O que `else` significa? (senão) **Por que não vem com outra expressão??** (a expressão está implícita - é o oposto do `if`, ou quando o `if` é falso) **Você sempre precisa de um `else`?** (não, depende da situação)

Condicionais Aninhadas

Metas

- Construir uma condicional aninhada
- Leia e compreenda uma condicional aninhada
- Entenda a indentação

Nós começamos com condicionais verificando uma coisa antes de tomarmos uma ação. Então nós tivemos duas ações para fazer e tivemos que decidir qual delas fazer quando. Mas às vezes você tem mais do que duas coisas que você quer fazer. É quando você quer colocar um condicional dentro de outro condicional.

primeira condicional abaixo é "se for um final de semana", a segunda condicional (aninhada) é "se eu tiver um jogo de futebol".

```
if it's a weekend:
    if I have a soccer game:
        Wake up at 6
    else:
        Sleep in
else:
    Wake up at 7
```

A indentação começa a importar muito agora. Recuamos quatro espaços para colocar o código dentro de um loop ou condicional, e isso inclui outros condicionais. O código dentro da segunda condicional é recuado em um total de oito espaços.

Por que precisamos de condicionais aninhadas? (Porque às vezes são possíveis mais de duas ações diferentes **Por que indentamos a segunda condicional por 4 espaços?** (Para mostrar que está dentro da primeira condicional.) **O que significa quando uma ação é indentada com 8 espaços?** (Depende de duas expressões sendo True ou False)

Funções

Metas

- Identifique funções.
- Construa uma definição de função.
- Chame uma função.

ocê já usou funções! Quando você digita `hero.cleave()`, `cleave()` é uma função. Até agora você só usou funções internas, mas também pode escrever suas próprias. Primeiro, você precisa definir a função usando `def`

```
def getReady():
    hero.wash(face)
    hero.brush(teeth)
    hero.putOn(armor)
```

Então você precisa chamar a função.

```
getReady()
```

Qual é a diferença entre definir e chamar? (Para definir precisa de um `def` antes e dois pontos depois. Então tem algum código indentado sob ele. Ambos têm parênteses.)

Os programadores usam funções para tornar seu código fácil de ler e rápido de escrever. É como uma jogada de basquete: você sabe lançar, driblar e passar, então você pode criar uma função que combine essas partes e dê um nome a ela.

```
def out-over-up():
    p1.driblar()
    p1.passar(p2)
    p2.lançar()
```

Então, quando o treinador quer que essa sequência de ações aconteça, ela apenas chama o nome da peça: “Out-over-up!”

Por que as funções facilitam a codificação? (Porque você não precisa dizer as etapas complicadas toda vez; basta usar o nome da função.) **Por que é importante dar bons nomes às suas funções?** (Então você pode lembrar para o que eles são para mais tarde.) **O que a palavra-chave `def` significa?** (definir)

Eventos

Metas

- Ouça eventos e execute código
- Use o tratamento de eventos para controlar um animal de estimação
- Escrever código simultâneo, misturando execução direta e manipulação de eventos

Até agora, você tem escrito código que é executado uma vez, de cima para baixo: primeiro faça isso, então faça isso, então faça isso . *Você também aprendeu a escrever loops while, onde você pode dizer, então faça isso para sempre* . Usando a manipulação de eventos, agora você tem uma maneira de dizer, **quando** isso acontece, **então** faça isso *. É como se fosse uma declaração if, exceto que os eventos podem acontecer a qualquer momento, não apenas quando você está verificando-os.

O que você usa para manipulação de eventos? (Para executar uma função quando algo acontece a função.) **Que tipo de dados é um nome de evento?** (O nome do evento que você ouve é uma string.) **Por que você não usa parênteses de função quando começa a ouvir um evento?** (Os parênteses tornariam a função executada agora e você deseja executá-la mais tarde quando o evento acontecer.)