

**CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
MESTRADO PROFISSIONAL EM EDUCAÇÃO E NOVAS
TECNOLOGIAS**

YANKO YANEZ KELLER DA COSTA

**O DESENVOLVIMENTO DE JOGOS DIGITAIS COMO
METODOLOGIA ATIVA NO ENSINO DE PROGRAMAÇÃO DE
COMPUTADORES NO ENSINO SUPERIOR**

CURITIBA

2020

**CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
MESTRADO PROFISSIONAL EM EDUCAÇÃO E NOVAS TECNOLOGIAS**

YANKO YANEZ KELLER DA COSTA

**O DESENVOLVIMENTO DE JOGOS DIGITAIS COMO METODOLOGIA ATIVA NO
ENSINO DE PROGRAMAÇÃO DE COMPUTADORES NO ENSINO SUPERIOR**

CURITIBA

2020

YANKO YANEZ KELLER DA COSTA

**O DESENVOLVIMENTO DE JOGOS DIGITAIS COMO METODOLOGIA ATIVA NO
ENSINO DE PROGRAMAÇÃO DE COMPUTADORES NO ENSINO SUPERIOR**

Dissertação apresentada ao Programa de Pós-Graduação – Mestrado Profissional em Educação e Novas Tecnologias, como parte dos requisitos necessários para obtenção do grau de Mestre em Educação e Novas Tecnologias.

Área de Concentração: Educação

Orientador: Prof. Dr. Luciano Frontino de Medeiros

CURITIBA

2020

C837d Costa, Yanko Yanez Keller da
O desenvolvimento de jogos digitais como metodologia
ativa no ensino de programação de computadores no
ensino superior / Yanko Yanez Keller da Costa. - Curitiba,
2020.
94 f. : il. (algumas color.)

Orientador: Prof. Dr. Luciano Frontino de Medeiros
Dissertação (Mestrado Profissional em Educação e
Novas Tecnologias) – Centro Universitário Internacional
UNINTER.

1. Ensino Superior. 2. Jogos digitais. 3. Aprendizagem ativa.
4. Programação (Computadores). 5. Tecnologias educacionais.
6. Inovações educacionais. I. Título.

CDD 371.334

Catálogo na fonte: Vanda Fattori Dias - CRB-9/547



uninter.com | 0800 702 0500

CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
PRÓ-REITORIA DE PÓS-GRADUAÇÃO, PESQUISA E EXTENSÃO-PGPE
PROGRAMA DE MESTRADO E DOUTORADO PROFISSIONAL EM EDUCAÇÃO E NOVAS TECNOLOGIAS
Secretaria do Mestrado e Doutorado Profissional em Educação e Novas Tecnologias

Defesa Nº 010/2020

ATA DE DEFESA DE DISSERTAÇÃO PARA CONCESSÃO DO GRAU DE MESTRE EM
EDUCAÇÃO E NOVAS TECNOLOGIAS

No dia 06 de abril de 2020, às 14h, reuniu-se via web conferência a Banca Examinadora designada pelo Colegiado do Programa de Mestrado e Doutorado Profissional em Educação e Novas Tecnologias, composta pelos professores doutores: Luciano Frontino de Medeiros (Presidente-Orientador-PPGENT/UNINTER), Sandro Rautenberg (Integrante Externo/ UNICENTRO/ GUARAPUAVA), Elton Ivan Schneider (Integrante Interno Institucional/ UNINTER), Rodrigo Otávio dos Santos (Integrante Interno Titular-PPGENT/ UNINTER) e Germano Bruno Afonso (Integrante Interno Suplente-PPGENT/ UNINTER) para julgamento da dissertação: "O DESENVOLVIMENTO DE JOGOS DIGITAIS COMO METODOLOGIA ATIVA NO ENSINO DE PROGRAMAÇÃO DE COMPUTADORES NO ENSINO SUPERIOR", do mestrando Yanko Yanez Keller da Costa. O presidente abriu a sessão apresentando os professores membros da banca, passando a palavra em seguida ao mestrando, lembrando-lhe de que teria até vinte minutos para expor oralmente o seu trabalho. Concluída a exposição, o candidato foi arguido oralmente pelos membros da banca.

Concluída a arguição, a Banca Examinadora reuniu-se e comunicou o Parecer Final de que o mestrando foi:

- (X) APROVADO, devendo o candidato entregar a versão final no prazo máximo de 60 dias.
- () APROVADO somente após satisfazer as exigências e, ou, recomendações propostas pela banca, no prazo fixado de 60 dias.
- () REPROVADO.

O Presidente da Banca Examinadora declarou que o candidato foi aprovado e cumpriu todos os requisitos para obtenção do título de Mestre em Educação e Novas Tecnologias, devendo encaminhar à Coordenação, em até 60 dias, a contar desta data, a versão final da dissertação devidamente aprovada pelo professor orientador, no formato impresso e PDF, conforme procedimentos que serão encaminhados pela secretaria do Programa. Encerrada a sessão, lavrou-se a presente ata que vai assinada pela Banca Examinadora.

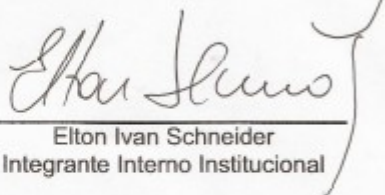
Recomendações: o aluno deverá fazer as alterações sugeridas pelos professores da banca, dentro do prazo estipulado para a entrega.



Luciano Frontino de Medeiros
Presidente da Banca



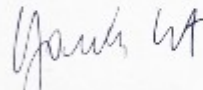
Sandro Rautenberg
Integrante Externo



Elton Ivan Schneider
Integrante Interno Institucional



Rodrigo Otávio dos Santos
Integrante Interno Titular



Yanko Yanez Keller da Costa
Mestrando

AGRADECIMENTOS

Agradeço a meus pais Adyr da Gama Costa (*in memoriam*) e Rose Liane Keller da Costa, por priorizarem a educação e o caráter como princípios direcionadores de nossa família.

Agradeço a meu orientador Prof. Dr. Luciano Frontino Medeiros, por seu incentivo e direcionamentos durante esta pesquisa.

Agradeço aos professores do programa de mestrado, em especial aos professores Luana Wunsch, Alvino Moser e João Mattar por diversas e interessantes reflexões durante esta jornada.

E agradeço, principalmente, a minha esposa Rosanni Costa, que me faz crescer e amadurecer diariamente nesta parceria de mais de 25 anos.

*"Education's purpose is to replace
an empty mind with an open one"*
Malcolm Forbes

Resumo

O ensino de programação de computadores tem crescido em importância no mundo todo, tanto em escolas quanto no ensino superior. Um dos motivos é o déficit de desenvolvedores e essa demanda aumenta à medida que a automação das atividades produtivas se dissemina na sociedade. Esta pesquisa foi desenvolvida para ser implementado em uma disciplina de programação web do curso de Gestão de Tecnologia da Informação, numa faculdade da região metropolitana de Curitiba. Trabalhando o engajamento dos alunos para aprender uma nova linguagem de programação, o objetivo deste trabalho envolveu a investigação do uso de metodologia ativa baseada em projeto com foco no desenvolvimento de jogo digital dentro da disciplina. O processo metodológico utilizado foi de natureza aplicada e de abordagem qualitativa do tipo exploratória e descritiva. Os dados para análise foram obtidos a partir da aplicação de formulários digitais, questionário semiestruturado em entrevistas e observação participante. Os resultados demonstram que o processo de desenvolvimento de um jogo a partir de uma estrutura de projeto foi um fator de engajamento dos alunos que chegaram a implementar características adicionais nos jogos além do escopo inicial.

Palavras-chave: Ensino de programação; Desenvolvimento de jogos digitais; Projetos no ensino; Metodologia ativas; Programação de computadores.

Abstract

The teaching of computer programming has grown in importance worldwide, both in schools and in higher education. One reason is the lack of developers and this demand increases as the automation of productive activities spreads in society. This research was developed to be implemented in a web programming discipline of the Information Technology Management course, at a college in the metropolitan region of Curitiba. Working on student engagement to learn a new programming language, the objective of this work involved the investigation of the use of active project-based methodology with a focus on the development of digital games within the discipline. The methodological process used was of an applied nature and of a qualitative approach, exploratory and descriptive. The data for analysis were obtained from the application of digital forms, a semi-structured questionnaire in interviews and participant observation. The results demonstrate that the process of developing a game based on a project structure was an engagement factor for students who came to implement additional features in the games beyond the initial scope.

Keywords: Programming teaching; Development of digital games; Teaching projects; Active methodologies; Computer programming.

LISTA DE ILUSTRAÇÕES

Figura 1: Demanda de profissionais de perfil tecnológico.....	17
Figura 2: Fluxograma de algoritmo de exemplo.....	34
Figura 3- Fluxo da coleta dos dados.....	42
Figura 4: Chamada inicial do projeto.....	47
Figura 5: Elementos do jogo.....	48
Figura 6: Coordenadas do objeto com "top" e "left".....	49
Figura 7: Coordenadas e colisão.....	50
Figura 8: Código JavaScript para movimento.....	51
Figura 9: Visualização do movimento do objeto.....	51
Figura 10: Código JavaScript para limite de movimento.....	52
Figura 11: Visualização do limite do movimento.....	53
Figura 12: Código JavaScript para captura de eventos de teclado.....	54
Figura 13: Código JavaScript para capturar coordenadas de objetos.....	55
Figura 14: Idade dos participantes.....	62
Figura 15: Nuvem de frequência das palavras mais utilizadas.....	70

LISTA DE TABELAS

Tabela 1: Características adicionais implementadas.....	57
Tabela 2 - Conhecimento de programação anterior a faculdade.....	62
Tabela 3 - Com relação ao nível de dificuldade no desenvolvimento de algoritmos de modo geral:.....	63
Tabela 4 - Dificuldades específicas em estruturas de linguagens de programação.....	63
Tabela 5 - Frequência do uso de jogos.....	64

LISTA DE QUADROS

Quadro 1: Exemplo de algoritmo em formato textual.....	35
Quadro 2 - Passos para coleta de dados.....	42
Quadro 3: Tecnologias e seu uso no jogo.....	44
Quadro 4: Plano do projeto.....	45
Quadro 5: Etapas obrigatórias do desenvolvimento do jogo.....	46
Quadro 6: Escopo do projeto.....	55
Quadro 7: Exemplos de jogos com elementos básicos.....	57
Quadro 8: Jogos sem cenários temáticos, mas com personagens.....	58
Quadro 9: Jogos com todas as propriedades adicionais implementadas.....	58
Quadro 10: Registro diário de acompanhamento.....	60
Quadro 11 - Idades e frequência no uso de jogos.....	65
Quadro 12 - Quem achou boa a utilização do jogo, completou o projeto?.....	66
Quadro 13 - Comparação do nível de dificuldade na execução do projeto com o nível de dificuldade geral antes do projeto.....	67
Quadro 14 - Dificuldades no desenvolvimento do jogo.....	68
Quadro 15 - Sugestões para desenvolvimento de programas.....	69
Quadro 16 - Trechos das falas dos alunos (P sendo a pergunta do pesquisador).....	71
Quadro 17: Incentivo para programar.....	73
Quadro 18: Fatores de envolvimento dos jogos e a atividade de programação.....	76
Quadro 19: Jogos com personagens e cenários.....	91

Sumário

1 INTRODUÇÃO.....	16
1.1 Relevância.....	19
1.2 Objetivo Geral.....	20
1.3 Objetivos específicos.....	20
1.4 Estruturação do trabalho.....	20
2 METODOLOGIAS ATIVAS.....	22
2.1 Uso de Projetos no ensino.....	24
2.2 Jogos no ensino.....	26
2.2.1 Jogos digitais.....	27
2.2.2 Desenvolvimento de Jogos digitais.....	29
3 PROGRAMAÇÃO DE COMPUTADORES.....	31
3.1 Conceitos usados em programação.....	33
3.2 Dificuldades em programação.....	36
3.2.1 Ensino de linguagem de programação.....	38
4 METODOLOGIA DA PESQUISA.....	40
4.1 Caracterização da pesquisa.....	40
4.2 Procedimento da pesquisa.....	41
4.3 Coleta de dados.....	42
5 DESENVOLVIMENTO DA PESQUISA.....	44
5.1 Material preparatório e elementos usados no jogo.....	46
5.2 Jogos desenvolvidos.....	56
5.3 Observação participante.....	60
5.4 Análise dos dados.....	61
5.4.1 Perfil dos participantes.....	62
5.4.2 Dificuldades com desenvolvimento de algoritmos.....	63
5.4.3 Interesse em jogos digitais.....	64
5.5 Análise do desenvolvimento do jogo.....	65
5.5.1 Entrevista com participantes do projeto.....	69

5.5.1.1 A influência do tema de jogos para o estudante.....	71
5.5.1.2 O aprendizado de programação.....	72
5.6 Reflexões sobre a pesquisa.....	74
6 CONSIDERAÇÕES FINAIS.....	78
REFERÊNCIAS.....	81
APÊNDICE.....	87
ANEXOS.....	96

1 INTRODUÇÃO

Qual ponto em comum podemos observar em situações tão diversas como uso de aplicativos bancários, jogos digitais, aulas via internet, *chats* robotizados, ambientes de compras online e carros autônomos? Em todas estas situações temos um programa de computador sendo executado e, a cada ano, mais atividades do cotidiano são auxiliadas por sistemas automatizados.

A digitalização da economia tem impulsionado a infraestrutura de redes de computadores à medida que mais pessoas iniciam sua vida online e desenvolvem suas atividades na internet. Este aumento fez com que o volume de dados passasse de 100 gigabytes por dia em 1992, para mais de 45.000 gigabytes por segundo em 2017 e com a previsão de chegar a 150.700 gigabytes por segundo em 2022 segundo as Nações Unidas (UNCTAD, 2019). No Brasil, em 2017, mais de um terço das transações bancárias já foram realizadas por dispositivos móveis (FEBRABAN, 2018, p. 22).

Esse maior uso de tecnologia tem criado um conjunto de empresas iniciantes e disruptivas (as *tech startups*) que tiveram um grande crescimento nos últimos anos e tem recebido investimentos substanciais, como por exemplo as *Fintechs* com US\$ 111 bilhões investidos (POLLARI; RUDDENKLAU, 2018) ou as *Agtechs* com US\$ 16,9 bilhões investidos (AGFUNDER, 2018) e estão revolucionando a maneira como as pessoas se relacionam com estas áreas. No Brasil, o uso de aplicativos em celular para ganhar dinheiro foi utilizado por 18 milhões de brasileiros em 2018 (EXAME, 2019).

Profissionais para a implantação de novas atividades automatizadas no Brasil tem sido cada vez mais demandados e, devido a isso, a escassez de pessoas com bons conhecimentos em tecnologia tem sido crescente, como traz a Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação (BRASSCOM) quando relaciona o total de formação de 46 mil pessoas com perfil tecnológico por ano – figura 1 - e a previsão de demanda de 70 mil profissionais anuais para o período de 2019 a 2024 (BRASSCOM, 2019).

Figura 1: Demanda de profissionais de perfil tecnológico



Fonte: BRASSCOM (2019)

Com a demanda mundial por pessoas com conhecimento em computação, o ensino de programação ou conceitos relacionados a programação de computadores tem se propagado, com alguns países Europeus integrando o ensino de programação de computadores inclusive nas séries iniciais do ensino básico (EUROPEAN SCHOOLNET, 2015). Da mesma forma, nos Estados Unidos, foram desenvolvidos objetivos de aprendizagem para estimular que crianças, desde o início do nível de ensino básico (K-12), já possam desenvolver programas com laços de repetição simples (MEDEIROS; WÜNSCH, 2019). Para permitir que estados, distritos e demais organizações implementem os conceitos de computação a seus alunos, várias entidades nos Estados Unidos colaboraram no desenvolvimento de uma proposta de diretrizes para o ensino de ciência da computação, que foram compiladas no K-12 Computer Science Framework (K12CS, 2019) e trazem como principais conceitos a serem ensinados no ensino básico:

1. Sistemas computacionais
2. Redes de computadores e Internet
3. Dados e análises
4. Algoritmos e programação
5. Impactos da computação.

Mas, enquanto aumenta a necessidade de evoluir os conhecimentos e habilidades dos jovens para prepará-los para este novo ambiente carregado de tecnologia, com desafios constantes e muita interatividade, estamos ainda às voltas com dificuldades básicas no ensino, conforme a última pesquisa do PISA em 2015, onde mostra que a

pontuação do Brasil em leitura, matemática e ciências está abaixo da média dos países que compõe a OCDE (INEP, 2015). Mesmo com esforços feitos na educação básica, onde indicadores apontam o crescimento da média em matemática nos anos iniciais do ensino fundamental (MEDEIROS; WÜNSCH, 2019).

Levantamentos feitos pelo Instituto Montenegro mostram que, mesmo no ensino superior, a proficiência dos alunos em compreensão de textos e matemática chega a apenas 34% dos estudantes, representando inclusive uma queda se comparado com os dados dos períodos de 2015 a 2018.

Para proporcionar uma melhor assimilação de conceitos e estimular o interesse do aluno na sua aprendizagem, metodologias ativas que centram o processo de ensino no próprio aluno, com experimentação, jogos, projetos, tem sido empregadas com a expectativa de gerar novas habilidades e competências que possibilitem uma melhor articulação das tecnologias com o conhecimento (MORAN, 2017, p. 24).

O uso de metodologias que enfatizam a prática e experimentação vem de encontro ao que Jenkins (2002) afirma, quando identifica que programar requer o uso de várias habilidades simultâneas, além de conceitos matemáticos e de análise de problemas e é um assunto difícil mesmo para os não iniciantes. Jenkins (2002) sustenta ainda que programar é essencialmente uma prática: para aprender a programar é necessário programar.

Segundo Ducharme (1993), o uso de atividades práticas como um artifício dentro do ensino para complemento da teoria e reforço da aprendizagem não é algo novo e vem sendo disseminado por vários educadores ao longo de décadas, citando personalidades dentro da área da educação como Friedrich Fröbel, Johann Herbart, Francis Parker e John Dewey.

A prática num curso tecnológico é uma de suas características, assim como o foco na tecnologia (SBC, 2017) e, mesmo cada curso tecnológico tendo suas especificidades, alguns desafios são comuns, o que leva a reflexões sobre a implantação dos cursos no território nacional. A Sociedade Brasileira de Computação, a partir de discussões sobre o ensino de computação no Brasil, elaborou currículos de referência para cursos de graduação no Brasil, como os de Ciência da Computação, Engenharia de Computação, Engenharia de Software, Licenciatura em Computação e Sistemas de Informação. Destes currículos surgiram as Diretrizes Curriculares Nacionais (DCNs). Devido ao esforço necessário e as especificidades, a SBC não produziu um referencial de formação para o curso que é objeto deste estudo ("Gestão da tecnologia da informação"), porém é

recomendado que sejam usados como referência as diretrizes gerais para os cursos tecnológicos nacionais, onde devem instigar a produção e a inovação científico-tecnológica que permitam aplicar seus conhecimentos e atender o mercado de trabalho (SBC, 2017).

Uma das recomendações metodológicas da SBC para o ensino nos cursos de computação está o uso de projetos que possam trazer aspectos práticos para dentro da sala de aula: "os projetos devem relacionar a teoria com a prática, contextualizando os conteúdos trabalhados nas disciplinas. Os projetos podem utilizar tutorias a fim de trazer situações do mercado de trabalho relacionado aos conhecimentos do curso, aproximando os conhecimentos com as necessidades da sociedade, das organizações" (SBC, 2017, p. 144).

Quanto mais próximo o tema fica de um uso prático, mais motivado o aluno se torna ao desenvolver a atividade (PRINCE; FELDER, 2006). Podemos perceber esta associação da relevância com a aprendizagem no estudo da NATIONAL RESEARCH COUNCIL (2000, p. 61), onde relata que:

Estudantes de todas as idades ficam mais motivados quando conseguem ver a utilidade do que estão aprendendo e quando podem usar esta informação para fazer algo que tenha impacto para outros – especialmente em sua comunidade local (NATIONAL RESEARCH COUNCIL, 2000, p. 61).

1.1 Relevância

O presente trabalho apresenta uma pesquisa empírica desenvolvida como trabalho de conclusão do mestrado profissional em educação e Novas Tecnologias no Centro Universitário Internacional – UNINTER – e tem como objetivo a aplicação de metodologia ativa baseada no desenvolvimento de jogos para o ensino de linguagem de programação em curso do nível superior.

Alinhado com as contribuições que novas tecnologias têm trazido para a sociedade, a linha de pesquisa deste programa de mestrado vem promover o uso de novas tecnologias de informação e comunicação nas investigações docentes, incentivando a criação de modelos e plataformas inovadoras em educação. Seguindo neste sentido, este trabalho pesquisa o desenvolvimento de jogos digitais como um indutor do aprendizado de linguagem de programação no ensino superior.

Consoante com um dos objetivos da linha de pesquisa que envolve a criação de ferramentas tecnológicas, como produto desta pesquisa temos o objeto de aprendizagem

constituído em um material com instruções textuais, imagens e código em linguagem de computador que permitem a sua utilização no desenvolvimento de projeto educacional para o ensino de linguagem de programação.

Este produto vem de encontro com a questão motivadora do pesquisador, que foi desenvolvida a partir do desafio de estimular os alunos a praticarem os conceitos de programação em uma disciplina onde duas linguagens de programação e duas de formatação são ensinadas no mesmo semestre. Neste sentido, a pesquisa foi direcionada para investigar se "*o uso de metodologia baseada em projeto, com o foco no desenvolvimento de jogos, pode estimular a aprendizagem de uma linguagem de programação?*".

Para resumir a finalidade desta pesquisa e permitir a delimitação do tema, seguem o objetivo geral e os objetivos específicos.

1.2 Objetivo Geral

Como objetivo geral, esta pesquisa propõe elaborar uma proposta de metodologia ativa baseada em jogos digitais, visando o ensino de programação no contexto do ensino de uma linguagem de programação no ensino superior.

1.3 Objetivos específicos

Para alcançar o objetivo principal deste trabalho, foram identificados os seguintes objetivos específicos a serem alcançados:

- a) Investigar as bases teóricas sobre o uso de jogos no ensino de programação para fundamentação do projeto de desenvolvimento de jogo.
- b) Investigar o uso de metodologias ativas baseada em projeto e em jogos digitais.
- c) Avaliar a utilização de metodologia ativa baseada em jogos e projetos para utilização como prática para o ensino de linguagem de programação.

1.4 Estruturação do trabalho

Este trabalho está estruturado em 6 capítulos. No primeiro capítulo temos o contexto onde a pesquisa se enquadra, com a justificativa do pesquisador e a questão que o levou a desenvolver a pesquisa, assim como a aderência a linha de pesquisa deste programa de mestrado.

O segundo capítulo traz a fundamentação teórica da pesquisa, com aspectos sobre metodologias ativas utilizadas no ensino, abordagens utilizando projetos e jogos digitais e como estão sendo utilizadas no ensino de programação.

O terceiro capítulo traz os conceitos básicos envolvidos na parte de programação de computadores e as dificuldades envolvidas no ensino de programação.

O quarto capítulo discorre sobre a metodologia utilizada, indicando sua caracterização, e o quinto capítulo destina-se a apresentar o desenvolvimento da pesquisa com os procedimentos utilizados em sala de aula, o material preparatório desenvolvido e a documentação dos jogos desenvolvidos pelos alunos, os resultados dos dados coletados antes, durante e após o desenvolvimento do projeto pelos alunos, apresenta o perfil dos participantes e reflexões sobre os dados.

O sexto capítulo propõe discutir sobre as considerações finais da pesquisa, avaliando seus objetivos e resultados.

2 METODOLOGIAS ATIVAS

Neste capítulo são apresentadas as principais características de metodologias ativas, o relacionamento com a proposição de atividades práticas e as abordagens de implementação como o uso de projetos e de jogos digitais na educação.

O uso de metodologias ativas tem sido proposto para que exista uma melhora no aprendizado do aluno, considerando a ênfase que estas metodologias empregam no protagonismo do aluno e seu envolvimento nas práticas, com a orientação do professor (MORAN, 2017, p. 23-24). Com o uso de métodos ativos, Barbosa e Moura (2013, p. 56) apontam que os alunos “assimilam maior volume de conteúdo” e “retem a informação por mais tempo”.

Considerando algumas variações nas definições do termo “metodologias ativas” apresentadas na literatura, este trabalho usa como referência o proposto pelos autores Mattar (2017), Barbosa e Moura (2013), Valente, de Almeida e Geraldini (2017), Bacich e Moran (2018) que, em comum, salientam o papel protagonista do aluno no processo de aquisição de conhecimento e a curadoria e acompanhamento do professor durante este processo.

Sobre o papel do aluno, Mattar (2017), indica que a “metodologia ativa” representa a inversão do papel passivo do aluno, com o professor no papel de orientador e instigador. A característica de incentivo a uma postura mais ativa dos alunos no processo de aprendizagem pode ter uma origem em muitos séculos atrás e, para exemplificar, Mattar (2017) apresenta o uso da *maieutica* por Sócrates (469-399 a.C.):

O filósofo grego utilizava-se de um método pelo qual não se propunha a ensinar diretamente, mas, indiretamente, por meio de perguntas, procurava levar as pessoas com quem conversava a reconhecer que não sabiam o que pensavam saber. Seus diálogos indicavam um caminho, não respostas, para provocar o parto de conceitos naqueles com quem debatia (MATTAR, 2017, p. 19).

Barbosa e Moura (2013), enfatizam a participação ativa do aluno quando propõem estimular o aluno a:

ler, escrever, perguntar, discutir ou estar ocupado em resolver problemas e desenvolver projetos. Além disso, o aluno deve realizar tarefas mentais de alto nível, como análise, síntese e avaliação. Nesse sentido, as estratégias que promovem aprendizagem ativa podem ser definidas como sendo atividades que ocupam o aluno em

fazer alguma coisa e, ao mesmo tempo, o leva a pensar sobre as coisas que está fazendo (BARBOSA; MOURA, 2013, p. 55).

Valente, de Almeida e Geraldini (2017) reforçam que a metodologia ativa é uma escolha, uma estratégia pedagógica de como estimular o aluno a ser mais autônomo na estruturação de seu conhecimento e na condução de sua aprendizagem e possibilita colocar seus conhecimentos em ação:

as metodologias ativas procuram criar situações de aprendizagem em que os aprendizes fazem coisas, colocam conhecimentos em ação, pensam e conceituam o que fazem, constroem conhecimentos sobre os conteúdos envolvidos nas atividades que realizam, bem como desenvolvem estratégias cognitivas, capacidade crítica e reflexão sobre suas práticas, fornecem e recebem feedback, aprendem a interagir com colegas e professor e exploram atitudes e valores pessoais e sociais (VALENTE; DE ALMEIDA; GERALDINI, 2017, p. 464).

Bacich e Moran (2018) trazem a importância da experimentação e criação desenvolvidas pelo aluno durante o seu processo de aprendizado:

As metodologias ativas dão ênfase ao papel protagonista do aluno, ao seu envolvimento direto, participativo e reflexivo em todas as etapas do processo, experimentando, desenhando, criando, com orientação do professor; [...] são estratégias de ensino centradas na participação efetiva dos estudantes na construção do processo de aprendizagem, de forma flexível, interligada e híbrida (BACICH; MORAN, 2018, p. 3).

A prática é um importante vetor para a implementação de metodologias ativas, como enfatizam Barbosa e Moura (2013): “as estratégias que promovem aprendizagem ativa podem ser definidas como sendo atividades que ocupam o aluno em fazer alguma coisa” (BARBOSA; MOURA, 2013, p. 55).

Berbel (2011, p. 37) também reforça a materialização das metodologias ativas através da prática como uma forma de causar efeito real na aprendizagem, ao indicar que “assim como ocorre com a teoria, uma metodologia, por mais promissora que seja pelas suas características, por si só, não transforma o mundo ou a educação, nem mesmo consegue promover a motivação autônoma dos alunos”.

Ao envolver o aluno em atividades práticas e associadas a temas de interesse, Bacich e Moran (2018, p. 2) indicam que “o que constatamos, cada vez mais, é que a aprendizagem por meio da transmissão é importante, mas a aprendizagem por questionamento e experimentação é mais relevante para uma compreensão mais ampla e

profunda", o que sugere que são utilizados métodos dedutivos¹ nas aulas, mas cresce em importância a utilização de métodos indutivos para reforçar a aprendizagem. A possibilidade de combinação dos dois métodos foi apontado por Bacich e Moran (2018):

Nos últimos anos, tem havido uma ênfase em combinar metodologias ativas em contextos híbridos, que unam as vantagens das metodologias indutivas e das metodologias dedutivas. Os modelos híbridos procuram equilibrar a experimentação com a dedução, invertendo a ordem tradicional: experimentamos, entendemos a teoria e voltamos para a realidade (indução-dedução, com apoio docente) (BACICH; MORAN, 2018, p. 2).

A implementação das metodologias ativas exige equilíbrio entre os conteúdos curriculares previstos, o interesse dos alunos, o nível conhecimento prévio, os recursos disponíveis e a capacidade de mediação do professor (VALENTE, 2018, p. 28). Os desafios também são apontados por Barbosa e Moura (2013) quando declaram que:

se de um lado vivenciamos com mais facilidade os métodos ativos de aprendizagem nas atividades práticas, por outro lado, resta-nos enfrentar um dos grandes desafios pedagógicos dos tempos modernos: incorporar aprendizagem ativa nos espaços e tempos atualmente ocupados pelas tradicionais aulas expositivas. (BARBOSA; MOURA, 2013, p. 56)

Para o desenvolvimento em sala de aula das metodologias ativas, Valente (2018, p. 28) indica que o uso de projetos e jogos são algumas das estratégias tradicionalmente adotadas.

2.1 Uso de Projetos no ensino

Uma das formas de cativar o aluno em sala de aula é a utilização de projetos e pesquisas tem apresentado resultados positivos ao comparar o conhecimento adquirido com o uso de projetos com aulas tradicionais (SAWYER, 2005, p. 318).

O envolvimento de projetos práticos junto a diversidade de aspectos que envolvem a aprendizagem não são recentes e já no início de 1900, quando um filósofo e pedagogo americano chamado John Dewey trouxe a valorização do "fazer", indicando que, ao resolver situações reais e participando da construção de seu conhecimento, a aprendizagem do aluno seria mais efetiva (MOURA; BARBOSA, 2013, p. 230).

1 No método dedutivo, o aluno é apresentado a diversos conceitos e teorias e depois levado a exercitar algum aspecto do que foi apresentado, levando a conclusão de que a teoria é aplicável, e ao utilizar uma metodologia indutiva, iniciamos com um ou alguns exemplos concretos e específicos que devem ser analisados, e a partir do aprofundamento e interpretação destes exemplos, vão sendo construídos fatos nos quais o aluno vai alimentando seu entendimento sobre o assunto (PRINCE; FELDER, 2006).

Conforme Barbosa e Moura (2013) o uso de projetos permite aplicar os conhecimentos em situações práticas e,

pode ser uma forma importante de compensar problemas decorrentes do uso exagerado de recursos virtuais, em detrimento de situações reais e contextuais. Essa é uma questão que se apresenta atualmente em relação à utilização de softwares educativos idealizados para simulação de situações da vida real (BARBOSA; MOURA, p. 61).

Um projeto é um conceito utilizado na sociedade para a execução de um objetivo específico e, conforme Moura e Barbosa (2013, p. 19), temos algumas características comuns nos projetos que ajudam a entender o conceito:

- São atividades orientadas para a realização de objetos específicos.
- Tem uma duração finita, com um princípio e um fim bem definidos.
- São atividades voltadas para a realização de algo único, exclusivo.
- Tem recursos limitados a disposição (pessoas, tempo, insumos, etc).
- Apresentam dimensões de complexidade e incerteza (ou risco) em sua realização.
- Surgem em função de um problema, uma necessidade, um desafio ou uma oportunidade (de uma pessoa ou instituição).

Sobre projetos em sala de aula, Bacich e Moran (2018, p. 16) definem que "é uma metodologia de aprendizagem em que os alunos se envolvem com tarefas e desafios para resolver um problema ou desenvolver um projeto que tenha ligação com a sua vida fora da sala de aula".

Ao utilizar a metodologia de projetos, é essencial que tenha uma questão ou problema que seja direcionador das atividades a serem desenvolvidas e que estas atividades resultem em artefatos ou produtos que podem ser de diferentes tipos (um relatório, um vídeo, um software, um processo), mas que respondam a questão inicial (BLUMENFELD et al, 1991).

Com relação aos tipos de projetos que podem ser desenvolvidos em sala de aula, Bacich e Moran (2018, p. 17) elencam os principais, baseados no tempo necessário e a complexidade:

Os principais modelos são:

1. Exercício-projeto, quando o projeto é aplicado no âmbito de uma única disciplina
2. Componente-projeto, quando o projeto é desenvolvido de modo independente das disciplinas, apresentando-se como uma atividade acadêmica não articulada com nenhuma disciplina específica.

3. Abordagem-projeto, quando o projeto se apresenta como uma atividade interdisciplinar, ou seja, como elo entre duas ou mais disciplinas.

4. Currículo-projeto, quando não mais é possível identificar uma estrutura formada por disciplinas, pois todas elas se dissolvem e seus conteúdos passam a estar a serviço do projeto, e vice-versa.

Na abordagem de projetos, o professor deve atuar como orientador deixando para o aluno a pesquisa, os testes e as associações do conteúdo vivenciado com suas experiências anteriores, possibilitando que a construção deste novo conhecimento seja mais bem incorporado (PRINCE; FELDER, 2006).

2.2 Jogos no ensino

O uso de jogos para estimular a prática de uma atividade não é uma ideia nova: Platão já considerava que jogos poderiam ser utilizados para reforçar comportamentos em adultos, e Piaget considerava os jogos importante para a aprendizagem das crianças (WILKINSON, 2016).

Na área militar, aliar a experiência que os jogos proporcionam com o exercício de algum conceito ou habilidade para apoiar a aprendizagem vem sendo utilizado há séculos. Um dos casos de sucesso do uso de jogos no treinamento militar auxiliou na preparação de comandantes que ganharam a Segunda Guerra Mundial (PERLA; MCGRADY, 2011).

Para incentivar a exploração de cenários estratégicos militares, foram desenvolvidos jogos específicos como o *Koenigspiel* em 1664, "*War Chess*" em 1780, *Kriegsspiel* em 1811 (SMITH, 2010), e até mesmo um complexo jogo digital como o "*Americam Army*" lançado inicialmente em 2002 para auxiliar no recrutamento e treinamento militar (NIEBORG, 2004). Desde o final da década de 1990, simulações militares vêm sendo utilizadas em computadores com o uso de ferramentas gráficas e inclusive geraram versões de jogos utilizadas em entretenimento (SMITH, 2010).

Com o sucesso destas experiências na área militar (PERLA; MCGRADY, 2011) e a força que a prática, trazida por diversos educadores, adquiriu nas últimas décadas para o reforço da aprendizagem (DUCHARME, 1993), os jogos foram percebidos como uma maneira prazerosa de treinar algumas habilidades dentro da área educacional.

2.2.1 Jogos digitais

Papert (1994) mostra a influência que os jogos digitais exercem desde a infância, destacando que é com o uso destes que muitos iniciam seu entendimento do uso da tecnologia digital (papel hoje que tem sido compartilhado com os *smartphones*):

Videogames, sendo o primeiro exemplo de tecnologia de computação aplicada à fabricação de brinquedos, foram sem dúvida a porta de entrada das crianças para o mundo da informática. Estes brinquedos, habilitando as crianças a testarem idéias sobre como trabalhar dentro de regras e estruturas preestabelecidas de um modo como poucos outros brinquedos são capazes fazer, provaram ser capazes de ensinar os estudantes, de uma forma que muitos adultos invejariam, sobre as possibilidades e empecilhos de um sistema recém-apresentado (PAPERT, 1994, p. 12).

Considerando que "os alunos de hoje não são mais as pessoas para as quais nossos sistemas educacionais foram projetados" (MATTAR, 2010, p. 10), a utilização de jogos digitais em sala de aula tem auxiliado no incentivo ao estudo e a aprendizagem e tem sido estudada por pesquisadores no mundo inteiro (CONNOLY et al., 2012). Além do estímulo na aprendizagem de conceitos em sala de aula, estudos também tem relacionado o uso de jogos digitais com a melhoria cognitiva e levam em consideração todo o esforço mental que o jovem emprega durante o jogo, como a necessidade de filtrar os diversos estímulos visuais e sonoros, a rapidez de raciocínio necessária para a tomada de decisão nos jogos e a avaliação do impacto destas decisões no resultado final (RAMOS; FRONZA; CARDOSO, 2018).

O uso de jogos também pode impactar nas habilidades perceptivo-visual que são importantes na área de tecnologia e a avaliação de resultado imediato que ocorre ao executar uma ação num jogo é que permite o acompanhamento dos erros e acertos a medida que esta ação acontece, o que também tem sido associada a eficácia da aprendizagem (CONNOLY et al., 2012). Este sistema de recompensa e *feedback* é uma das características dos jogos e é o que faz com que a pessoa que está jogando fique constantemente estimulada e busque atingir os seus objetivos (VAN ECK, 2006)

Comprovando estas situações sobre *feedback*, estudos recentes, incluindo imagens do cérebro e processamento neural, têm demonstrado que o cérebro estimula o sistema de recompensa através da liberação de dopamina, o que faz com que atividades consideradas prazerosas sejam estimuladas, repetidas e aprendidas, podendo inclusive chegar ao vício (TANG, 2018).

Segundo Prensky (2003), com os jogos digitais os jovens aprendem a usar informações de várias fontes para tomar decisões rapidamente, a deduzir regras conforme são apresentadas, a criar estratégias para transpor obstáculos e compreender sistemas complexos através da experimentação.

Deve-se avaliar como o jogo pode ser utilizado para estimular uma atividade, pois nem todos os tipos de jogos ou condições de uso dos jogos são aderentes: "games não são perfeitos para tudo, e podem não ser perfeitos para todos os públicos, assuntos e contextos. É preciso avaliar se um tema é ou não apropriado como base para um game" (MATTAR, 2010, p. 84).

Sobre o envolvimento dos alunos com jogos, foram identificados por Prensky (2001), seis fatores que são estruturantes para o envolvimento dos jogadores, que são:

1. Regras
2. Metas e objetivos
3. Resultados e *Feedback*
4. Conflito/Competição/Desafio/Oposição
5. Interação
6. Representação ou estória

Nos cursos de computação, com a dificuldade apresentada pelos alunos durante as disciplinas de programação, o uso da temática dos jogos digitais tem sido introduzida de diferentes maneiras para familiarizar os alunos com conceitos de lógica e outros aspectos da computação, sendo que a operação de jogos ou o uso de motores de jogos que não necessitam de codificação são os mais comuns (SILVA; FERNANDES; SANTOS, 2018).

Do "uso" de jogos, a área de tecnologia passou a testar o desenvolvimento de jogos como uma alternativa para as disciplinas de programação, para aumentar o envolvimento dos alunos (DE RAADT, 2010). Mas esta também não é uma proposta nova. Papert (1994), ao criar uma linguagem de computador específica para o uso de crianças, usou o desenvolvimento de um jogo como uma das primeiras opções para iniciar o trabalho com a linguagem LOGO e despertar o interesse dos jovens,:

O jogo 'vinte um' mostrou-se simples o bastante para ser jogado por programas dentro da compreensão de crianças da sétima série que eram, de fato, capazes de basear-se na experiência para a discussão de estratégias para pensar. Os estudantes ocuparam-se com prazer da criação de programas que gerariam frases em inglês aproximado e, fazendo isso, chegaram a um novo tipo de entendimento de gramática (PAPERT, 1994, p. 152).

2.2.2 Desenvolvimento de Jogos digitais

Já em 1998, Seymour Papert propunha que jovens aprendessem a desenvolver jogos como uma forma de evoluir intelectualmente, inclusive aprimorando sua habilidade de aprender de maneira autônoma (PAPERT, 1998). Mattar (2010), cita que o desenvolvimento de jogos poderia ser associado, como uma forma de cultura dos jovens e, desta forma, envolvê-los em seu próprio contexto de aprendizagem :

Com suporte e acesso às ferramentas adequadas, o entusiasmo de jogar games pode se transformar em entusiasmo para produzi-los, o que por sua vez gera pensamentos mais sofisticados sobre os aspectos dos games. O objetivo, obviamente, não é produzir games profissionais, mas é provável que num futuro próximo fazer games seja parte da cultura dos jovens, assim como desenhar, escrever ou tocar música (MATTAR, 2010, p. 58).

O desenvolvimento de jogos carrega algumas características específicas, conforme Kanode e Haddad (2009):

O videogame é uma síntese de código, imagens, música e atuação que se juntam em uma forma de entretenimento. [...] Diferentemente da maioria dos outros domínios de aplicativos de software, o desenvolvimento de jogos apresenta desafios únicos decorrentes das múltiplas disciplinas que contribuem para o videogame. O projeto combina todos esses ativos em um aplicativo (KANODE; HADDAD, 2009, p.).

O uso de objetos gráficos durante o desenvolvimento de um jogo permite que os erros que podem se apresentar dos códigos, em muitos casos, podem ser verificadas visualmente, o que facilita o diagnóstico do problema (LEUTENEGGER; EDGINGTON, 2007).

A proposta de desenvolvimento de um jogo dentro de um curso precisa ser planejado para as restrições da disciplina sendo aplicada, pois o desenvolvimento de um jogo em sala de aula assume contornos diferentes de um jogo profissional, onde o tempo de desenvolvimento pode levar anos e envolver dezenas de programadores, conforme exemplifica Kanode e Haddad (2009) sobre o relatório final do jogo BioShock, onde mais de 90 programadores, equipes distribuídas em 3 países, produziram mais de 750.000 linhas de código C++ e mais de 3.500 arquivos para o jogo em 3 anos de desenvolvimento. Sobre o tamanho e complexidade do jogo a ser desenvolvido, Mattar (2010) cita que:

Games não precisam necessariamente ser do tipo multiusuário em 3D. Muitas vezes a escolha correta é desenvolver um pequeno jogo

casual, quando esse estilo de jogo corresponde aos objetivos de aprendizagem da atividade. Nem todos os games precisam ser grandes e complicados: a escala e a complexidade devem ser escolhidas para corresponder aos objetivos de aprendizagem e ao contexto (MATTAR, 2010, p. 84).

Algumas precauções sobre o desenvolvimento de jogos são apontadas por De Raadt (2010):

O sucesso de um contexto de programação de jogos baseia-se em estudantes tendo familiaridade com jogos de computador e um interesse em produzi-los, o que não é verdade para todos os estudantes. Também podem ter barreiras criadas pelo custo de licenciamento de ambientes e de hardware apropriado para desenvolvimento de jogos (DE RAADT, 2010, p. 2).

Ao desenvolver um jogo, o aluno assume o protagonismo da atividade e, considerando que o escopo da tarefa é mais facilmente identificável devido à familiaridade do aluno com o uso de jogos, o aluno consegue perceber antecipadamente o resultado desejável do seu código e esse protagonismo é um importante foco das metodologias ativas, sendo que o uso de projetos e jogos são algumas das ferramentas para sua implantação (MORAN, 2017).

Neste capítulo foi apresentada a principal característica das metodologias ativas que é o deslocamento do foco da aprendizagem do professor para o aluno, permitindo uma maior qualidade do aprendizado. Foram feitas considerações sobre a implementação das estratégias pedagógicas com o desenvolvimento de ações práticas e o engajamento dos alunos. Em seguida, foi apresentado o uso de projetos em sala de aula como um condutor das atividades e uma forma de trazer para a aula os desafios enfrentados fora do ambiente de ensino. Também foi exposta a utilização de jogos como uma forma de maior envolvimento dos alunos nas atividades e o desenvolvimento de jogos como uma possibilidade de aprimorar suas habilidades de aprendizado autônomo.

3 PROGRAMAÇÃO DE COMPUTADORES

Neste capítulo são apresentados os principais conceitos relacionados a programação de computadores, a importância do aprendizado de programação e as dificuldades envolvidas no aprendizado de programação.

Quando a programação de computadores é entendida e assimilada, pode ser uma atividade prazerosa que pode levar ao aprendizado de outros temas como resultado do desenvolvimento da solução automatizada (JENKINS, 2002; PAPERT, 1994).

Uma forma de perceber o envolvimento que a programação de computador pode despertar em uma pessoa, de maneira que ela passe suas horas livres programando sistemas complexos por diversão, é o grande volume de software livre² disponível. Podemos observar isso em projetos de software como os da "The Apache software foundation", que hospeda diversos tipos de sistemas utilizados em empresa de vários tamanhos (por exemplo, NASA, Netflix, IBM, Facebook, Uber), assim como em universidades e por indivíduos. Uma amostra do volume e contribuição da comunidade são os indicadores da fundação (APACHE, 2019):

- Em vinte anos 1.058.321.099 de linhas de código foram liberadas para uso (*committed*).
- Mais de 3.022.836 programas liberados para uso em projetos.
- Mais de 350 projetos de desenvolvimento.
- 7.000 programadores supervisores de código voluntários, além de milhares de programadores voluntários ao redor do mundo.
- Sistemas avaliados em mais de 20 bilhões de dólares, usando estimativa baseada no modelo COCOMO II, e liberados para uso da comunidade sem custo.

Estatísticas projetam que o mercado de trabalho nos Estados Unidos vai demandar mais profissionais em ciências da informação e computação que a média de todas as demais ocupações e isto tem gerado esforços para aumentar a participação nas disciplinas desta área, com projetos que visam antecipar conhecimentos sobre computação antes de entrar na faculdade³ e capacitando professores em ciências da computação como na iniciativa da *National Science Foundation*⁴ (QIAN; LEHMAN, 2017).

2 Software livre ou *opensource*, é uma denominação usada em programas de computador que podem ser utilizados e adaptados sem custo.

3 AP Students <<https://apstudents.collegeboard.org/>>

4 CS10K <<http://cs10kcommunity.org/>>

A importância da aprendizagem de programação de computadores tem extrapolado os cursos de computação no ensino superior e o uso de conceitos vindos da computação para auxiliar as habilidades de estudantes não é uma discussão nova: desde a década de 1960 com Alan Perlis, passando por Seymour Papert na década de 1980 e recentemente, no início dos anos 2000 com Jeannette Wing, vemos a discussão sobre o uso de habilidades como abstração, decomposição de problemas, representações simbólicas, controle de fluxo e lógica condicional, entre outros, sendo propostas para o auxílio da aprendizagem e que estão diretamente associadas a conceitos usados em computação (GROVER, PEA, 2013).

Pesquisadores como Papert (1994), mostraram a importância do uso do computador como uma ferramenta de apoio importante para auxiliar o aluno em sua busca pela autonomia intelectual e o levou, inclusive, a criar uma linguagem de programação própria para isso, a linguagem LOGO, que foi aprendida e utilizada por centenas de alunos e professores em várias partes do mundo.

Enquanto o ensino de programação vem ganhando visibilidade em vários países, trazendo a possibilidade dos jovens desenvolverem sistemas, também vem sendo valorizadas outras características que acompanham o ensino de algoritmos e que não estão ligadas diretamente com a codificação, chamado de pensamento computacional (WING, 2006). Grover e Pea (2013) mostram que ainda existem diferentes definições do termo "pensamento computacional", mas as definições apontadas relacionam o diagnóstico e a resolução de problemas de uma forma que possam ser representados por um algoritmo e esta forma de estruturação do pensamento tem como seu principal fundamento a abstração.

Conforme Grover e Pea (2013, p. 39), os elementos aceitos como importantes para a implementação e identificação dos princípios do pensamento computacional, são:

- Abstrações e generalizações de padrões (incluindo modelos e simulações);
- Processamento sistemático de informações;
- Sistemas simbólicos e representações;
- Noções de fluxo de controle em algoritmos;
- Decomposição estruturada de problemas (modularização);
- Pensamento iterativo, recursivo e paralelo;
- Lógica condicional;
- Restrições em desempenho e eficiência;
- Investigação e detecção sistemática de erros;

Uma das características do pensamento computacional apontada por Wing (2006) indica que o pensamento computacional está relacionado com a forma como os humanos pensam e não é uma maneira de fazer pessoas pensarem como computadores:

Nós humanos tornamos os computadores excitantes. Equipados com dispositivos computacionais, nós usamos nossa esperteza para resolver problemas que não ousaríamos enfrentar antes da era da computação e construir sistemas com funcionalidade limitada apenas pela nossa imaginação" (WING, 2006, p. 35, tradução nossa).

Para a programação de computadores, diferentes conceitos estão envolvidos como o de algoritmos, programa de computador e linguagem de programação.

3.1 Conceitos usados em programação

Diferentes tipos de tarefas podem ser feitas de forma automatizada por computadores, de forma que são utilizados em situações tão diversas como identificar genes no DNA, acessar informações na internet, acessar serviços e prover o pagamento de produtos e, nas indústrias, administrar insumos e controlar a produção (CORMEN et al, 2009).

O computador é um aparelho eletrônico preparado para receber dados e processá-los para apresentar um resultado e, para que esse processo seja executado, ele precisa ser instruído, utilizando uma série de instruções organizadas (programa) para executar uma operação (MANZANO; OLIVEIRA, 2014).

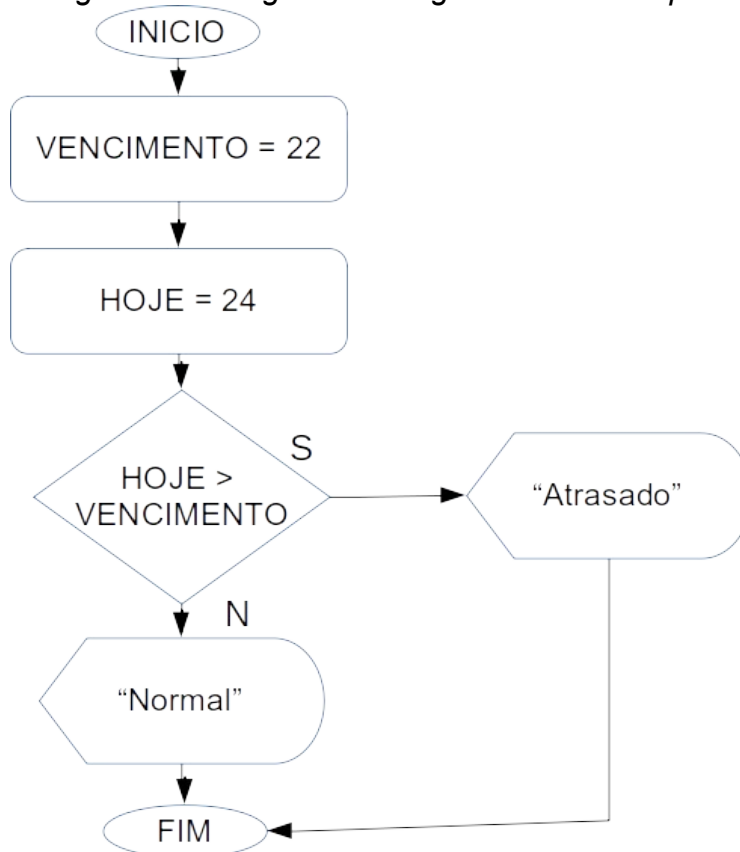
O processo de programar um computador tem várias fases, mas antes é necessário entender qual a solução que será aplicada ao problema escolhido e, com base nesta solução, criar uma sequência de passos que possam ser executados e que transformam as informações iniciais no resultado esperado para a solução do problema (CORMEN et al, 2009). Esta sequência de passos é o algoritmo e, após ele ter sido definido é que a programação inicia, podendo esta fase ser considerada uma tradução mecânica deste processo, como observa Jenkins (2002, p. 55). Manzano e Oliveira (2014, p. 24) indicam que algoritmo é empregado na matemática e em computação, sendo que na computação " está associada a um conjunto de regras e operações bem definidas e ordenadas, destinadas a solução de um problema, ou de uma classe de problemas, em um número finito de passos".

Um algoritmo pode ser expresso de forma gráfica, em um texto em inglês ou através de um projeto eletrônico, sendo que o principal requerimento é a descrição

precisa e ordenada de dos passos a serem seguidos (CORMEN et al, 2009; MANZANO; OLIVEIRA, 2014).

Um exemplo de um algoritmo expresso na forma gráfica pode ser visto na figura 2, onde é apresentado passo a passo o procedimento de verificação de vencimento de um pagamento. Nesta sequência apresentada visualmente por um diagrama de blocos⁵, temos a identificação do dia do vencimento do pagamento (22) e a data atual (24), representados por um retângulo. Depois é feito um teste se a data atual (representada pela palavra HOJE) é maior que a data do vencimento (representado pela palavra VENCIMENTO). A figura utilizada para o teste é um losango. Caso a data atual seja maior que a data do vencimento (“S”) mostra-se a mensagem de atraso (“Atrasado”). Caso não seja maior (“N”), mostra-se a mensagem de normalidade (“Normal”).

Figura 2: Fluxograma de algoritmo de exemplo



Fonte: o autor.

Considerando que a data atual é 24 e a data do vencimento é 22, o algoritmo da figura 2, mostrará a mensagem “Atrasado” como resultado da sua execução. Se for comparado a quantidade de texto necessário para explicar o funcionamento do algoritmo (no parágrafo acima) e a quantidade de símbolos com menor margem de interpretação

⁵ O diagrama de blocos é uma das formas de representar um algoritmo visualmente, sendo padronizado pela ISO 5807:1985 (MANZANO; OLIVEIRA, 2014, p. 30)

que o algoritmo terá ao ser analisado no formato visual temos uma das vantagens do formato gráfico, podendo ser entendido também por quem não conhece uma linguagem de programação (MANZANO; OLIVEIRA, 2014).

O mesmo algoritmo usado na figura 2 pode ser representado de forma textual, como podemos ver no quadro 1. Neste quadro temos duas alternativas, sendo uma com pseudocódigo⁶ e outra um trecho de programa em linguagem de programação chamada *Python*. Ao comparar as duas sequências do mesmo algoritmo, pode ser verificado a semelhança e também a característica das linguagens de programação de utilizarem palavras do idioma inglês, como “*if*”, “*print*” e “*else*”. Outra característica na representação textuais do algoritmo (e que também pode ser visto no diagrama de blocos) é o uso de palavras para armazenamento temporário das datas de vencimento e data atual. Este mecanismo é semelhante ao uso de variáveis (x, y, z) usados na Álgebra elementar.

Quadro 1: Exemplo de algoritmo em formato textual

PSEUDOCÓDIGO	PROGRAMA PYTHON
VENCIMENTO = 22	vencimento = 22
HOJE = 24	hoje = 24
SE HOJE > VENCIMENTO	if hoje > vencimento:
ENTÃO	print "Atrasado."
ESCREVA "Atrasado."	else:
SENÃO	print "Normal."
ESCREVA "Normal."	
FIM-SE	

Fonte: o autor

Para instruir o computador a executar efetivamente o algoritmo, é preciso estabelecer um processo de comunicação entre a pessoa que detém a solução a ser executada e o computador, que deve ser feito através de um tipo de linguagem especificamente criada para esta tarefa (MANZANO; OLIVEIRA, 2014). No quadro 1, temos um exemplo de linguagem de programação, mas de acordo com Manzano e

⁶ Pseudocódigo é um tipo de linguagem de programação que não tem o mesmo rigor de uma linguagem de programação real. Serve para auxiliar a descrição do algoritmo e pode ser feito usando o idioma regional (MANZANO; OLIVEIRA, 2014, p. 35)

Oliveira (2014, p. 22) existem cerca de 2.500 linguagens de programação e podem ter linguagens desenvolvidas para tarefas específicas (comercial, científica) ou de uso geral.

Segundo Manzano e Oliveira (2014), para estabelecer a comunicação com o computador foram desenvolvidas linguagens de baixo nível e alto nível, sendo que as de alto nível são mais conhecidas e utilizadas por se aproximarem da comunicação humana e ser baseadas em palavras do idioma inglês, e as de baixo nível estão mais próximas da arquitetura do computador. No quadro 1, temos um exemplo de programa em linguagem de alto nível e de uso geral chamada Python.

3.2 Dificuldades em programação

Programar computadores é difícil? Para Manzano e Oliveira (2014, p. 29), "a programação de computadores é uma prática que necessita de metodologia, disciplina e autoconhecimento do pensar. É necessário direcionar o pensamento, dentro da limitação técnica de um computador".

Quando o aluno está iniciando na programação de computadores, mesmo seu conhecimento anterior de matemática e o formato de expressões algébricas pode gerar algumas dificuldades ao desenvolver seus primeiros algoritmos, como por exemplo a necessidade de algumas linguagens de programação exigirem a declaração de variáveis antes do uso, a limitação de tamanho ou precisão em variáveis, o resultado de operações com números inteiros em algumas linguagens serem limitados e a aparência de expressões algébricas quando comparadas com a sintaxe das atribuições de valor e comparações feitas em linguagem de programação (QIAN; LEHMAN, 2017). Mesmo com a utilização de ambientes gráficos que substituem as estruturas textuais das linguagens de programação para minimizar o inconveniente com a sintaxe, como os de programação em blocos⁷, o baixo entendimento de conceitos de repetições, uso de variáveis e expressões lógicas permanece (GROVER; BASU, 2017).

Aprender a programar envolve a resolução de problemas com o uso de várias habilidades e conhecimentos simultaneamente e, principalmente, com foco na experimentação prática, conforme cita Jenkins (2002, p. 55) "programação é aprendido programando, não através de livros".

7 Ambientes de programação que utilizam modelos gráficos de estruturas de programação, que podem ser arrastadas na tela e concatenadas conforme a lógica é desenvolvida <<https://cacm.acm.org/magazines/2019/8/238340-block-based-programming-in-computer-science-education/abstract>>.

Para o aluno iniciante, as dificuldades no entendimento de programação podem ser de diferentes tipos como, por exemplo, no aprendizado de códigos de repetição (*loops*), o aluno pode ter dificuldade com a sintaxe, com a forma como o código é construído ou como utilizar a repetição para resolver um problema (QIAN; LEHMAN, 2017).

Para propor uma solução para um problema é preciso analisar a situação e criar um modelo mental que ajude a entender os principais aspectos que serão necessários para a criar um algoritmo e para isso é fundamental a capacidade de abstração (SBC, 2017). Ambrósio e Costa (2010) consideram que a complexidade da sintaxe das linguagens de programação usadas para o desenvolvimento de programas, o baixo nível de abstração dos alunos ingressantes e sua dificuldade em resolução de problemas, geram as dificuldades encontradas pelos alunos nos cursos de computação.

Sobre a dificuldade dos alunos com programação, Giraffa e Mora (2013) acrescentam também "deficiências relacionadas à expressão em língua materna (escrita), interpretação de textos e enunciados, hábitos de estudo e pesquisa e, especialmente, com formação básica deficitária no que tange a conteúdos de Matemática".

Enquanto o conhecimento de programação de computadores cresce em importância, as disciplinas de algoritmos e linguagem de computador são citadas como um dos principais desafios nos cursos de computação no Brasil (SOUZA; BATISTA; BARBOSA, 2016; BULCÃO; CAMPOS NETO; MOREIRA, 2017; RAABE; SILVA, 2005). As disciplinas iniciais de algoritmos e linguagem de computador nos cursos de computação no Brasil são, em grande parte, o principal motivo de desistência dos alunos conforme cita Giraffa e Mora (2013) em:

No que tange a evasão associada aos cursos de Computação, estudos realizados por diversos pesquisadores da área de Educação e Computação e do Ministério da Educação brasileiro (MEC) mostram que os alunos desistem dos cursos logo no primeiro ano do ensino superior. As disciplinas causadoras desta desistência são aquelas associadas ao ensino de Cálculo e de Programação (incluindo-se a disciplina de Algoritmos).

Mas o fenômeno relacionado a evasão e dificuldades dos alunos com algoritmos é uma preocupação internacional e diferentes países têm pesquisado sobre esta situação (ISA; DERUS, 2017; LAHTINEN; ALA-MUTKA; JÄRVINEN, 2005; DE RAADT et al, 2005; SCOTT; GHINEA, 2013).

Sobre as dificuldades no ensino de uma linguagem de programação temos, além do esforço da abstração envolvida na solução do problema, aspectos relacionados a uma

sintaxe rígida na montagem das instruções que devem ser seguidas pelo computador e o uso de conceitos relacionados a estrutura de dados, variáveis e arquitetura de máquina que não são facilmente compreendidos pelos alunos (AMBROSIO; COSTA, 2010).

Com base no tipo de aprendizado para o aluno, segundo Qian e Lehman (2017), o conhecimento de programação pode ser categorizado em três tipos:

1. Conhecimento sintático, que está relacionado ao conhecimento das regras da linguagem de programação e suas funcionalidades.
2. Conhecimento conceitual, que está relacionado ao entendimento de como os conceitos da programação funcionam e como serão efetivados dentro do computador.
3. Conhecimento estratégico, que se relaciona com a resolução do problema utilizando os conhecimentos de sintaxe e conceituais.

Estes desafios são mais impactantes nas disciplinas de introdução a programação no ensino superior, onde é exigido dos estudantes um maior uso de abstração e estes, percebendo suas dificuldades e limitações já no início do curso, acabam desistindo e gerando as altas taxas de evasão (ROBINS; ROUNTREE; ROUNTREE, 2003; VIZCAÍNO et al, 2000; PORTER et al., 2013). Apesar de vários estudos sobre as dificuldades das disciplinas de programação envolvendo aspectos como currículo do curso, forma pedagógica, ferramentas e linguagens, o resultado efetivo do ensino de programação em sala de aula ainda é limitado (PEARS et al., 2007).

3.2.1 Ensino de linguagem de programação

Normalmente uma disciplina de programação tem seu foco na codificação em uma linguagem específica, por exemplo, Java, C, Python, e em seu paradigma⁸ (PEARS et al, 2007).

Ao selecionar uma linguagem de programação para ser trabalhada, também é associado o tipo de paradigma de programação que será utilizado na disciplina: orientação a objetos, procedural (ou imperativa) ou funcional, apesar de que uma linguagem pode ser ensinada sem aproveitar o paradigma ao qual ela pertence (PEARS et al, 2007).

Existem diversas discussões sobre qual dos paradigmas devem ser utilizados nas disciplinas iniciais, para facilitar a compreensão dos conceitos de programação, mas,

⁸ Segundo Manzano e Oliveira (2014, p. 22), paradigma de programação é o modelo ou estilo da forma de programar.

segundo Bennedsen e Caspersen (2007), quando verificada a taxa de aprovação dos alunos, não se verificou diferenças entre os tipos de paradigmas utilizados.

Enquanto não existem sólidas evidências que indiquem qual a melhor linguagem para o ensino de programação (JENKINS, 2002), outros fatores são usados para esta escolha, como: preferência dos docentes, aspectos técnicos da linguagem, visibilidade na indústria e as ferramentas e recursos disponíveis (PEARS et al, 2007).

Uma das estratégias utilizadas para melhorar a forma de passar os conceitos fundamentais de programação e algoritmos para os estudantes e, assim, investir na retenção dos alunos, tem sido o uso de jogos digitais nas disciplinas de programação:

Acreditamos que as preocupações sobre se uma abordagem orientada a objeto ou uma abordagem procedural devem ser usadas primeiro são secundárias ao objetivo do curso e aos exemplos de conteúdo. Se os exemplos não são convincentes, os interesses dos alunos geralmente ficam lentos, tornando o argumento orientação a objeto versus procedural discutível. Acreditamos que a programação de jogos motiva a maioria dos novos programadores (LEUTENEGGER; EDGINGTON, 2007).

Neste capítulo, a programação de computadores foi apresentada como uma necessidade além do mercado de trabalho. O uso de conhecimentos advindos da programação de computadores pode auxiliar também o desenvolvimento de habilidades como abstração e a capacidade de resolução de problemas complexos. Foram apresentados os pilares do pensamento computacional e conceitos relacionados a programação de computadores como: linguagem de programação, algoritmos e computadores. Foram pesquisadas também as dificuldades da aprendizagem de programação, que podem estar relacionadas aos diferentes tipos de desafios como o maior uso da abstração, capacidade para interpretação de textos, formação básica deficitária em matemática, a complexidade da sintaxe das linguagens e conceitos de estrutura de dados e arquitetura de máquina, são alguns dos tópicos identificados.

4 METODOLOGIA DA PESQUISA

Neste capítulo temos a caracterização da pesquisa que teve como ponto central a relação entre a prática pedagógica e a utilização da tecnologia nos processos de ensino e de aprendizagem, com a utilização do desenvolvimento de um jogo digital para apoiar esta prática.

4.1 Caracterização da pesquisa

Para classificá-la foram identificados sua natureza, objetivos e metodologia, com base em critérios conceituados por Gil (2002), Moreira e Caleffe (2008), Vergara (1998), Prodanov e Freitas (2013).

Como a pesquisa trata de uma situação concreta no uso de abordagens práticas para facilitar o ensino de programação, ela tem como natureza ser aplicada, que conforme (MOREIRA; CALEFFE, 2008; PRODANOV; FREITAS, 2013; VERGARA, 1998) é um tipo de pesquisa que tem um propósito prático.

Quanto ao objetivo a atual pesquisa tem dois posicionamentos:

1. Primeiramente, temos o estudo de uma abordagem de ensino que é apoiada no uso de metodologia de projeto de um jogo digital, cuja proposta visa avaliar se o uso deste artifício pode facilitar a aprendizagem de uma nova linguagem de programação. Neste sentido, podemos classificar esta parte da pesquisa como exploratória onde, “têm como objetivo principal o aprimoramento de idéias ou a descoberta de intuições” (GIL, 2002, p. 41) e também, segundo Prodanov e Freitas (2013, p. 52) “tem como finalidade proporcionar mais informações sobre o assunto que vamos investigar, possibilitando sua definição e seu delineamento, isto é, facilitar a delimitação do tema da pesquisa; orientar a fixação dos objetivos e a formulação das hipóteses ou descobrir um novo tipo de enfoque para o assunto”.
2. Em segundo, considerando o processo de análise de dados relacionados aos questionários aplicados e as entrevistas feitas, a pesquisa também tem um caráter descritivo, onde os dados serão trabalhados para avaliar os alunos participantes dos questionários, permitindo expor seus atributos e descrevê-los (MOREIRA; CALEFFE, 2008; VERGARA, 1998), ou como define Gil (2002, p. 42) sobre a pesquisa descritiva: “têm como objetivo primordial a descrição das características

de determinada população ou fenômeno ou, então, o estabelecimento de relações entre variáveis”.

Com relação a metodologia empregada, ela pode ser classificada como qualitativa ou interpretativa (MOREIRA; CALEFFE, 2008). Também é importante ressaltar, segundo Moreira e Caleffe (2008, p. 64), que este tipo de pesquisa é “uma interação dialética contínua, análise, crítica, reiteração, reanálise e assim por diante, levando a uma construção articulada do caso”. Para Prodanov e Freitas (2013, p. 70) na pesquisa qualitativa “há uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números”.

4.2 Procedimento da pesquisa

Para o desenvolvimento da pesquisa, deve ser necessário o planejamento das atividades a serem trabalhadas para a apresentação do projeto para os alunos, para a coleta de dados para a montagem do perfil, para a coleta de informações sobre a visão do aluno após o desenvolvimento do projeto e a estrutura de perguntas a serem utilizadas nas entrevistas posteriores.

Como foco do estudo foi definida a disciplina "Programação WEB", onde é necessário ensinar duas linguagens de programação e algumas tecnologias de apoio, envolvendo também uma linguagem de marcação de texto e uma de formatação. A aplicação da metodologia proposta foi limitada ao ensino da linguagem de programação JavaScript no primeiro bimestre.

Antes do início do projeto foi aplicada uma pesquisa através de formulários, para capturar informações sobre o perfil dos alunos e a aderência que o tema sobre jogos teria. O objetivo desta pesquisa era encontrar eco do público participante da disciplina, com o levantamento feito sobre uso de jogos e a aceitação positiva do tema nos resultados das bibliografias pesquisadas. Após a coleta dos dados, os alunos foram incentivados informalmente a compartilhar que jogos usavam e onde, com o objetivo de trazer o tema para a sala de aula. Na sequência, os alunos foram informados que fariam um projeto sobre o tema de jogos. O projeto poderia ser detalhado numa próxima aula.

Para desenvolver o projeto foi feito um planejamento dos dias a serem utilizados pelos alunos para a atividade e como o projeto seria apresentado e acompanhado pelo pesquisador. O projeto foi desenvolvido para ser trabalhado durante o primeiro bimestre da disciplina de "Programação WEB" que foi ministrada no segundo semestre de 2018,

compreendendo o período letivo de 30/07/2018 a 06/10/2018. A disciplina foi ministrada duas vezes por semana, sendo duas aulas de 45 minutos por dia e, para o desenvolvimento do projeto, foram alocadas dez aulas do total.

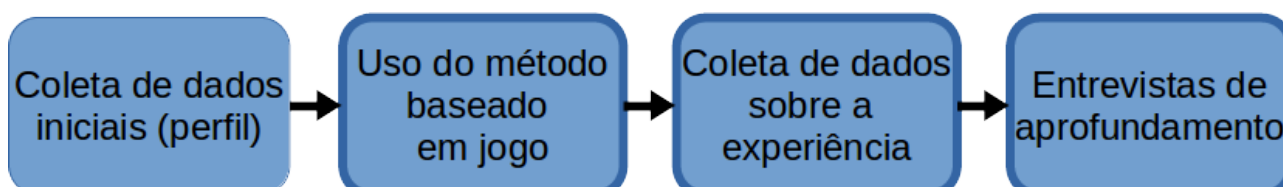
4.3 Coleta de dados

Durante a produção desta pesquisa foram desenvolvidas estratégias de coleta de dados, utilizando as seguintes ferramentas:

- a) Para a coleta do perfil dos alunos do estudo de caso, foi elaborado um questionário estruturado.
- b) Para coleta de informações sobre dificuldades e motivação no desenvolvimento do jogo, foi elaborado um questionário estruturado.
- c) Entrevistas foram feitas com os alunos para avaliar a motivação em programar computadores.

Na figura 3, temos o fluxo completo da coleta de dados considerando todas as etapas envolvidas.

Figura 3- Fluxo da coleta dos dados



Fonte: o autor.

Para o desenvolvimento da coleta das informações devem ser desenvolvidos os procedimentos, conforme Quadro 2.

Quadro 2 - Passos para coleta de dados

Para coletar dados sobre o perfil dos alunos	<p>Etapas:</p> <ol style="list-style-type: none"> 1. Criação de formulário no “Google forms” (ver APÊNDICE) 2. Aplicação do formulário em sala de aula 3. Exportação dos dados em planilha
Coleta de dados sobre as dificuldades no desenvolvimento	<p>Etapas:</p> <ol style="list-style-type: none"> 1. Criação de formulário no “Google forms” (ver APÊNDICE)

	<ol style="list-style-type: none"> 2. Aplicação do formulário em sala de aula 3. Exportação dos dados em planilha
Unificação e análise dos dados	<p>Etapas:</p> <ol style="list-style-type: none"> 1. Junção dos registros das duas coletas usando o código do aluno e e-mail 2. Aplicação de filtros para análise dos padrões e correlação de dados
Entrevistas com alunos	<p>Etapas:</p> <ol style="list-style-type: none"> 1. Elaboração de temas para as perguntas. 2. Elaboração de perguntas direcionadoras. 3. Elaboração de critérios de seleção dos participantes das entrevistas. 4. Montagem e assinatura do "Termo de consentimento livre e esclarecido". 5. Agendamento das entrevistas e gravação. 6. Transcrição das entrevistas (uso do "Google Docs" opção "Digitação por voz").

Fonte: o autor

Os alunos participantes da primeira coleta de dados, a partir do formulário sobre o perfil, foram identificados com a nomenclatura "A"+numero sequencial, conforme a ordem das respostas. Esta nomenclatura permaneceu durante toda a pesquisa, possibilitando acompanhar as demais interações destes alunos, assim como também criar associações do seu perfil com as demais coletas de dados, inclusive a fase de entrevistas.

A fase de entrevista com alunos foi a única desenvolvida após o término da disciplina de "programação WEB" e envolveu o contato dos alunos por e-mail e o convite para participação voluntária na pesquisa.

5 DESENVOLVIMENTO DA PESQUISA

Este capítulo tem como objetivo apresentar a aplicação do projeto de desenvolvimento do jogo na disciplina de programação web, assim como os resultados obtidos. Os materiais usados como referência são identificados e são apresentados jogos desenvolvidos pelos alunos durante as aulas.

Para a utilização da metodologia de projetos foi trabalhado o modelo exercício-projeto identificado por Bacich e Moran (2018) como uma proposta a ser desenvolvida dentro de uma disciplina e, em complemento a estrutura de projeto, foi adaptado o desenvolvimento de um jogo digital.

Para aplicação da abordagem de desenvolvimento de um jogo durante a disciplina foram desenvolvidos os seguintes procedimentos:

- Montagem de material com explicação básica sobre a linguagem JavaScript, com a demonstração de como um código é executado no navegador.
- Preparação do material introdutório sobre jogos no navegador web.
- Explicação dos conceitos de movimentação e colisão de objetos no navegador, utilizando trechos de código em JavaScript, HTML e CSS preparados para ser utilizados como referência.
- Apresentação da proposta de desenvolvimento do jogo com informações sobre os elementos mínimos a serem desenvolvidos e a forma de avaliação do projeto.
- Acompanhamento dos alunos durante o desenvolvimento.

A disciplina iniciou com a apresentação das tecnologias que seriam utilizadas no primeiro bimestre: *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS), *Document Object Model* (DOM) e JavaScript. No quadro 3 são apresentados as tecnologias estudadas e sua utilização no desenvolvimento do jogo.

Quadro 3: *Tecnologias e seu uso no jogo.*

TECNOLOGIA	DESCRIÇÃO
HTML	Linguagem de marcação que permite identificar a informação textual e diferentes mídias que serão apresentadas para o usuário e disponibiliza estes objetos na área de trabalho do navegador. Os personagens, o ambiente e as imagens do jogo são apresentados ao usuário por esta linguagem.
CSS	Linguagem que permite inserir estilos (cores, posicionamento, tamanho) nos objetos disponibilizados pela linguagem HTML, permitindo que os objetos do jogo possam ser posicionados e caracterizados na tela.

DOM	É uma forma padronizada de acesso dinâmico aos objetos HTML e características CSS em um documento sendo apresentado ao usuário. Desta forma é possível alterar posicionamento de um objeto já existente para promover a ilusão de movimento na tela, usado em jogos.
JAVASCRIPT	É uma linguagem de programação utilizada para interagir com o usuário do navegador e permite acesso via DOM aos objetos HTML e CSS. Com esta linguagem é possível criar os movimentos e capturar os eventos de teclado, necessários para a implementação do jogo.

Fonte: o autor

Após a apresentação aos alunos e alguns exercícios básicos com a construção de páginas HTML formatadas com CSS, foram feitos também alguns códigos básicos para entender o funcionamento e execução de um programa em JavaScript. Na sequência foi apresentado o plano do projeto do jogo, conforme plano apresentado no quadro 4.

Quadro 4: Plano do projeto

DATAS	DESCRIÇÃO
Dia 1	Aplicação do questionário inicial. Coleta de dados para perfil.
Dia 2	Apresentação da proposta do jogo e informações iniciais.
Dia 3 a 9	Desenvolvimento do jogo em aula pelos alunos.
Dia 10	Apresentação final.

Fonte: o autor

Para a apresentação do projeto de desenvolvimento do jogo, foi feito material para ser usado como base das explicações sobre o funcionamento de um jogo, sobre como podem ser feitas animações numa página WEB e como pode ser feito o controle de teclado para interação com o jogo.

Códigos de exemplos foram preparados e testados, assim como material visual para explicação de pontos de maior complexidade, como posicionamento de objetos na tela e coordenadas usadas para colisão de objetos. Estes códigos foram preparados com o objetivo de serem didáticos, sem a preocupação com desempenho, mas principalmente para serem entendidos de maneira facilitada pelos alunos. Este material foi preparado para ser um objeto de aprendizagem que possa ser reutilizado em novas aulas ou até mesmo em outras situações em que o ensino de JavaScript e sua prática sejam o objetivo.

Para permitir a avaliação do projeto, foram criadas etapas obrigatórias a serem entregues como mostrado no quadro 5.

Quadro 5: Etapas obrigatórias do desenvolvimento do jogo

ETAPAS OBRIGATÓRIAS	DESCRIÇÃO
O controle do objeto do jogador e tela principal.	Nesta etapa o aluno deve desenvolver o código do objeto a ser manipulado pelo jogador através das setas no teclado e também deve incluir os limites de tela onde o jogador e os demais elementos do jogo devem ficar confinados.
O movimento dos objetos que avançam independentemente.	Nesta etapa devem ser feitos os objetos que serão movimentados na tela e que devem provocar o movimento do jogador.
O mecanismo de colisão.	Nesta fase do jogo, deve ser desenvolvido o código que possibilite detectar quando o jogador colide com um dos objetos autônomos que passam pela tela. Nesta etapa também deve ser feita a contagem de vidas do jogador.

Fonte: o autor

Para cada uma das etapas obrigatórias foi atribuído um critério de avaliação. Foi associado um ponto caso a etapa fosse completada. Um total de 3 pontos foram distribuídos para o aluno que completou todas as etapas.

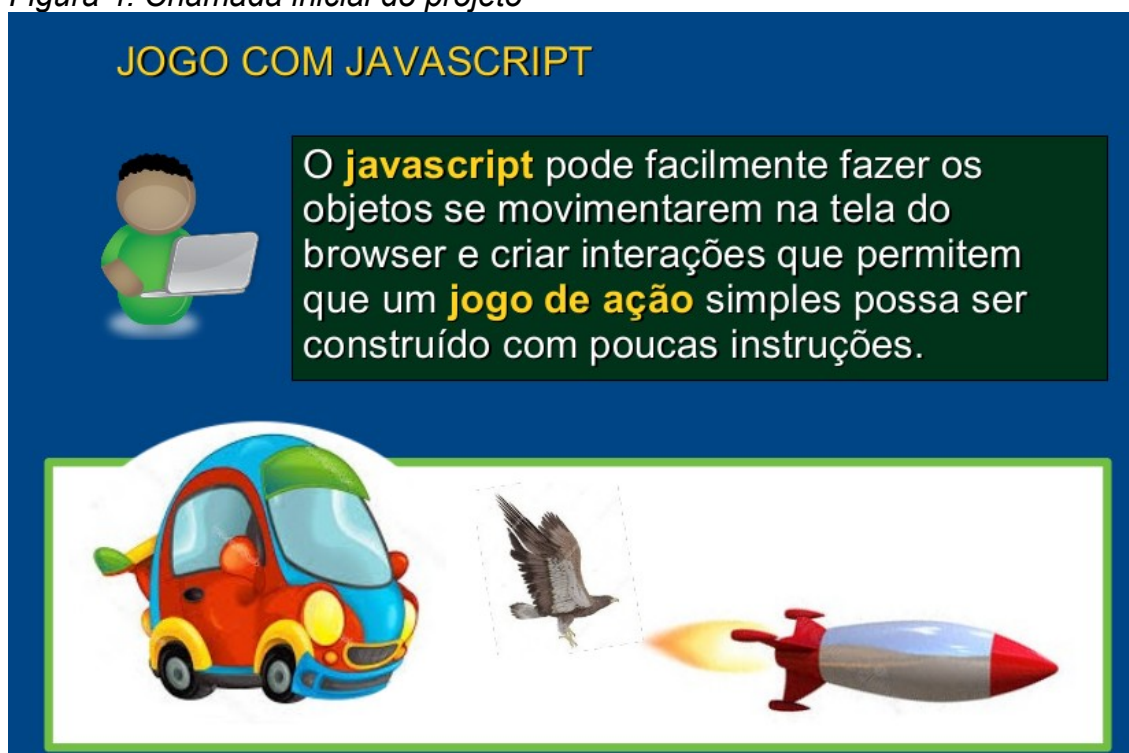
5.1 Material preparatório e elementos usados no jogo

Para esclarecer os alunos com relação a proposta do projeto e características relacionadas ao desenvolvimento de jogos, foi desenvolvido um material preparatório. Este material apresentou elementos a serem usados no desenvolvimento do jogo e trechos de códigos que exemplificaram estes elementos e foi passado aos alunos nas duas aulas iniciais do período reservado para o desenvolvimento do projeto. Na figura 4 temos a chamada inicial sobre o uso de JavaScript para desenvolvimento de jogos.

Nesta primeira parte os alunos foram apresentados ao projeto e a possibilidade do uso da linguagem JavaScript para o desenvolvimento de um jogo de ação em 2D⁹ durante as aulas.

⁹ Jogos 2D é um formato de visualização dos objetos do jogo onde são movimentados na vertical e horizontal usando duas coordenadas.

Figura 4: Chamada inicial do projeto



Fonte: o autor

Na sequência foram apresentados os elementos a serem utilizados no projeto do jogo. Foram listadas e discutidas as características dos elementos e como ele poderia fazer parte do jogo:

- a) **Movimento dos objetos.** Estava liberada a possibilidade de utilizar qualquer tipo de elemento HTML para estes objetos que iriam se movimentar na tela. Deveriam pesquisar a respeito.
- b) **Obstáculos.** Estava liberada a possibilidade de utilizar qualquer tipo de elemento HTML para estes obstáculos ou utilizar os limites da tela. Deveriam pesquisar a respeito.
- c) **Inimigos do jogador.** Poderiam utilizar quaisquer elementos de HTML. Deveriam pesquisar a respeito.
- d) **Controles a serem utilizados.** Foi apresentada a opção de utilização das setas pelo teclado. Com o mesmo mecanismo, outras teclas poderiam ser utilizadas. Os códigos das teclas a serem utilizadas deveriam ser pesquisadas. Também estava liberada a possibilidade de utilizar outro mecanismo de controle.
- e) **A colisão de objetos.** Deveria haver um mecanismo de colisão baseado nas coordenadas e no tamanho dos objetos.

Figura 5: Elementos do jogo

Elementos do nosso jogo!!

Nosso jogo será um jogo de ação 2D que deve ser desenvolvido inteiramente em *Javascript*, sem o uso de *frameworks* e bibliotecas de jogos externas.

Movimento:

Loop de repetição onde os objetos vão se movimentando sozinhos.

Obstáculos:

Objetos que devem "atrapalhar" ou provocar um desafio na movimentação do personagem.

Inimigos:

Objetos que devem ser evitados (podem também ser tiros, armadilhas).

Controles:

Utilização do teclado (setas) para movimento do personagem.

Colisão:

Forma de detectar se os objetos estão se sobrepondo.

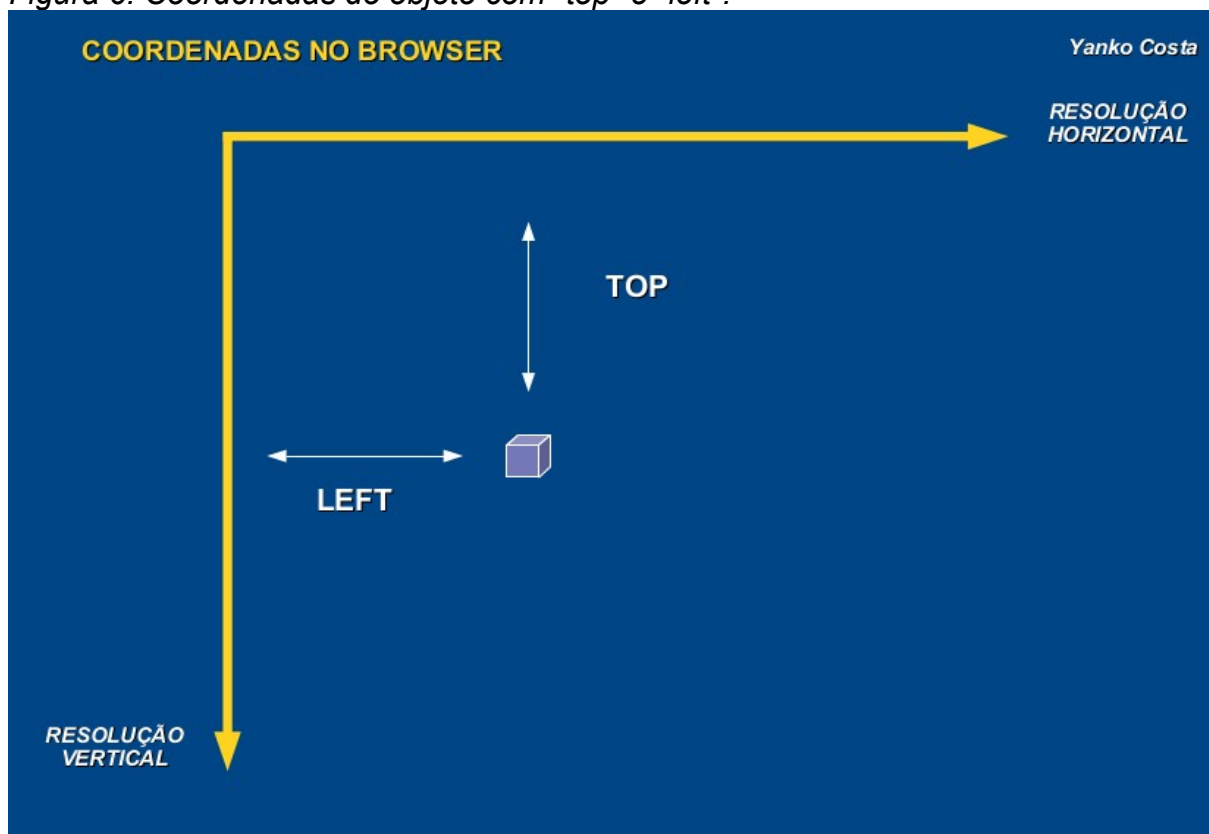
Fonte: o autor

Foi indicado que este jogo seria feito individualmente pelos alunos durante o primeiro bimestre, unindo as tecnologias de HTML, CSS, DOM e JavaScript que tinham sido apresentadas no início da disciplina.

Um dos conceitos que foi preparado para ser explicado para os alunos foi o de coordenadas usadas no navegador (*browser*). Informações sobre resolução horizontal e vertical, sobre o acesso a estes indicadores usando CSS e questões sobre posicionamento de objetos na área interna do navegador foram expostos e exemplificados. Na figura 6, um dos slides utilizados na explicação de posicionamento com as propriedades "*top*¹⁰" e "*left*" do CSS.

¹⁰ As propriedades *top* e *left* indicam a distância do objeto com relação a borda superior e esquerda do navegador web.

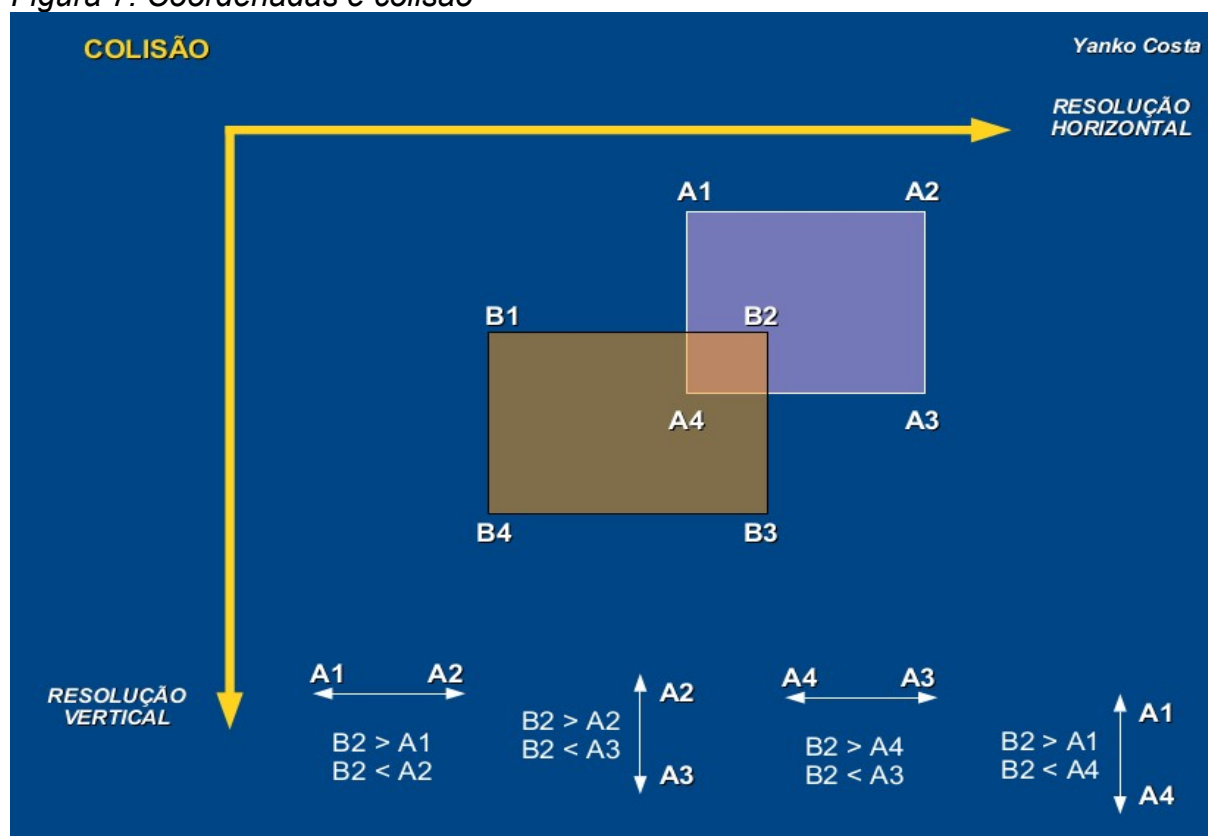
Figura 6: Coordenadas do objeto com "top" e "left".



Fonte: o autor

Outro conceito importante para o desenvolvimento do jogo está relacionado à colisão de objetos, que consiste em verificar se um objeto está dentro da área de outro objeto na tela. Com isso, é necessário compreender como os objetos se posicionam na tela e como coletar informações sobre sua dimensão. A partir destes dados pode-se identificar se um dos pontos do objeto em movimento está localizado na área de outro. Na figura 7, um exemplo do material usado para esta explicação.

Figura 7: Coordenadas e colisão



Fonte: o autor

Na figura 7, é mostrado como um objeto pode ser identificado dentro de área de outro objeto no sistema de coordenadas usado no navegador. Para simplificação, tanto da explicação quanto do desenvolvimento do código do jogo, foi estabelecido que todos os objetos fossem avaliados a partir de um formato retangular (mesmo que as figuras utilizadas tivessem formato irregular).

Após a apresentação dos conceitos, foi publicada uma série de códigos em JavaScript que demonstravam como executar a movimentação de objetos na tela e como capturar o acionamento das setas no teclado. Os trechos de código apresentados, além de facilitar o entendimento sobre elementos de animação e interatividade com JavaScript, poderiam ser utilizados como componentes iniciais do jogo a ser desenvolvido pelo aluno.

Na figura 8, temos um código feito em JavaScript, somado a uma formatação em CSS que permite que uma figura movimente-se na horizontal (da esquerda para a direita) de forma ilimitada. A função "*setInterval*" do JavaScript permite que um trecho de código seja executado a cada intervalo (neste caso, 20 milissegundos). Ela proporciona o efeito de repetição¹¹ que o jogo necessita para criar a animação dos objetos. Neste código, o

¹¹ Diferente de uma estrutura de controle de repetição tipo "*for*" e "*while*", o "*setInterval*" permite que a tela seja redesenhada a cada iteração, permitindo visualizar a movimentação dos objetos enquanto ela

elemento segue na horizontal sem parar na borda do navegador, permitindo ao aluno perceber a necessidade de controlar o elemento em movimento.

Figura 8: Código JavaScript para movimento.

Movimento


```
<style type="text/css">
  DIV { position: absolute;
        top: 80;
      }
</style>
```

```
<div id="imagem">
  
</div>
```

```
<script language="JavaScript">
  var x = 0;

  function anda() {
    var imagem = document.getElementById("imagem");
    x++;
    imagem.style.left = x;
  }

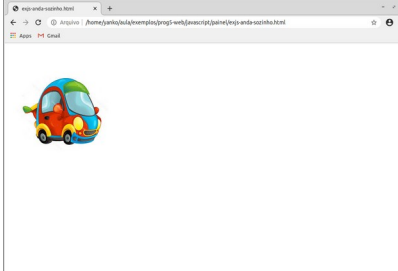
  setInterval(function() { anda() }, 20);
</script>
```



Fonte: o autor

A sequência de imagens na figura 9 representa a posição do objeto na tela, durante a execução do código da figura 3, em 3 momentos diferentes. Considerando que o código da figura 3 não tem limitação para o incremento da variável x , o objeto continua sua movimentação indefinidamente para a direita.

Figura 9: Visualização do movimento do objeto



1. Início da movimentação



2. Variável x é incrementada



3. Segue na horizontal sem parar

ocorre.

Fonte: o autor

Na figura 10, é exemplificado uma alteração no código de movimento para que a figura interrompa seu movimento ao chegar em uma distância na linha horizontal da tela. Neste caso, ao chegar na coordenada 250, o objeto volta na posição inicial. Visualmente o observador interpreta como tendo um limite no movimento do objeto, dando a impressão de um obstáculo sendo atingido.

Figura 10: Código JavaScript para limite de movimento.

```
Parede
<script language="JavaScript">
  var x = 0;

  function anda() {

    var imagem = document.getElementById("imagem");

    if (x < 250)
      x++;
    else
      x=0;

    imagem.style.left = x;
  }

  setInterval(function() { anda() }, 20);

</script>
```

Fonte: o autor

A sequência de imagens na figura 1 representa a posição do objeto na tela, durante a execução do código da figura 10, em 3 momentos diferentes. Neste caso, ao chegar no valor 250, a variável x volta a ter o valor zero, posicionando o objeto no início da tela novamente.

Figura 11: Visualização do limite do movimento



Fonte: o autor

Na figura 12, temos um exemplo de código que pode ser usado para capturar o uso da seta da direita do teclado (`tc1.keycode == 39`) e movimentar o objeto para a direita. O aluno, ao testar este exemplo, consegue apenas movimentar o objeto visualizado na tela para a sua direita de maneira ilimitada (sem parar na lateral do navegador). Por testar apenas a seta da direita, o usuário não consegue navegar em outras direções (por exemplo: para cima, para baixo e para esquerda). Este código incentiva o aluno a pesquisar as demais teclas que podem ser utilizadas e como estabelecerá os limites que o jogador terá na tela. Também poderá propor como estes limites serão percebidos visualmente pelo jogador.

Figura 12: Código JavaScript para captura de eventos de teclado.

```
Jogador

<body onkeydown="anda(event)" >

<script language="JavaScript">
var andar = 0;
function anda(tcl)  {

    var imagem = document.getElementById("jogador");

    // HORIZONTAL ->
    if (tcl.keyCode == 39) {
        andar = andar + 10;
    }

    // EXECUTA
    imagem.style.left = andar;
}
</script>

<div id="jogador">
    
</div>

</body>
```

Fonte: o autor

Para possibilitar que o aluno pudesse entender o mecanismo de colisão dentro do jogo, foi apresentado uma forma de coletar a coordenada básica do objeto, representando seu posicionamento horizontal e vertical. A partir dela, usando a padronização estabelecida de uso da forma retangular como base, pode-se inferir as demais coordenadas do objeto para identificar a área de colisão. Para isso, o aluno deveria identificar o tamanho do objeto (largura e altura) e somar estas medidas ao ponto inicial, tendo como resultado as quatro coordenadas a serem utilizadas. A partir de suas pesquisas, os alunos poderiam identificar outras formas de captura das coordenadas ou do tamanho dos objetos. Na figura 13 o trecho de código que apresenta uma das formas de coleta das coordenadas.

Figura 13: Código JavaScript para capturar coordenadas de objetos.



Pegar coordenadas

```
//coordenadas inimigo
var X1 = parseInt(document.getElementById("inimigo").style.left);
var Y1 = parseInt(document.getElementById("inimigo").style.top);
```

DICA:

```
X2 = X1 + largura_objeto
Y2 = Y1 + altura_objeto
```

Fonte: o autor

Em todos estes casos, os alunos foram estimulados a testar cada um dos códigos apresentados. Ao final da apresentação do material preparatório foi exposto o escopo do projeto do jogo a ser desenvolvido. No quadro 6, é declarado o escopo do projeto.

Quadro 6: Escopo do projeto.

O jogo baseia-se num objeto que é manipulado pelo usuário e que pode se mover para todos os lados até a borda limite do jogo. Deve se desviar de objetos que "saem" da lateral direita da tela.

Se o objeto do jogador colidir com o objeto que avança na tela, deverá haver uma sinalização e o jogador perde 1 das 3 vidas que possui. Deverá haver pelo menos 5 objetos avançando simultaneamente.

Fonte: o autor

O enunciado foi montado para incentivar os alunos a articular as informações apresentadas no material preparatório, com o desafio de lidar com mais objetos se movimentando na tela, com a criação do ambiente onde as atividades devem acontecer e a criação de uma estrutura de dados a ser utilizada para controle de cada um dos itens da tela.

5.2 Jogos desenvolvidos

Durante o período reservado para o desenvolvimento do projeto, os alunos foram encorajados a pesquisar exemplos de programas em JavaScript, além de figuras e detalhes visuais que poderiam ser utilizados nos seus jogos. Outros elementos como som também poderiam ser incluídos.

Como o objetivo principal da disciplina, ao incentivar um projeto como este, é o aprendizado da linguagem JavaScript dentro de um contexto web, uma regra estabelecida foi que o aluno deveria escrever todo o código utilizado e não deveria utilizar *frameworks*¹² prontos disponíveis na Internet. Poderiam adaptar ou não os códigos usados no material preparatório para o desenvolvimento do jogo. Também foi considerado opcional o uso de elementos gráficos e sonoros no código, sendo que não seriam pontuados caso tivessem sido implementados.

Dos participantes do projeto, 50% se dedicaram apenas ao desenvolvimento do código do jogo. Os demais investiram também em aspectos visuais, sonoros e também na jogabilidade. Para este estudo foram analisadas também as características implementadas adicionalmente ao escopo passado, o que foi considerado um esforço do aluno em tornar seu jogo mais próximo de um jogo normal ao qual está acostumado.

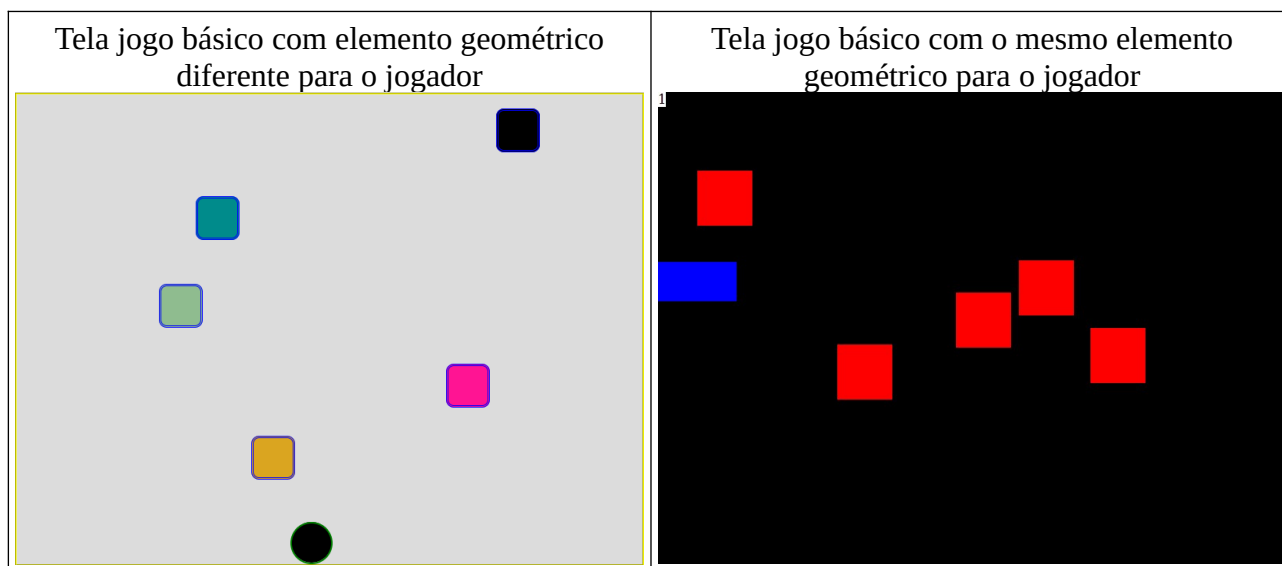
Estas características foram agrupadas em:

1. **BÁSICO** – Uso de elementos básicos de HTML (sem o desenvolvimento adicional de características como imagens, figuras ou som).
2. **PERSONAGEM** - Uso de personagens gráficos (pesquisa e utilização de figuras temáticas para os personagens).
3. **CENÁRIO** - Uso de cenário (pesquisa e utilização de elemento temático para compor o cenário e os limites da área jogável).
4. **SOM** - uso de som (pesquisa e utilização de elemento sonoro em eventos do jogo).

Como exemplo de jogo BÁSICO temos o quadro 7, onde foram colocadas duas telas demonstrando o uso dos elementos HTML usados para o jogo. Oito dos jogos desenvolvidos tinham este padrão, com pequenas variações em cor e parâmetro nos estilos CSS, como transparência, arredondamento ou bordas.

¹² Framework na área de programação de computadores envolve a utilização de códigos desenvolvidos e preparados para serem reutilizados.

Quadro 7: Exemplos de jogos com elementos básicos.



Fonte: o autor

Com relação aos projetos que incluíram elementos visuais e sonoros adicionais, temos jogos que foram desenvolvidos com uma ou mais destas características. Dos oito jogos que implementaram características adicionais, seis implementaram mais de um dos tipos identificados no projeto e três implementaram todos. A tabela 1 mostra o resumo das implementações adicionais.

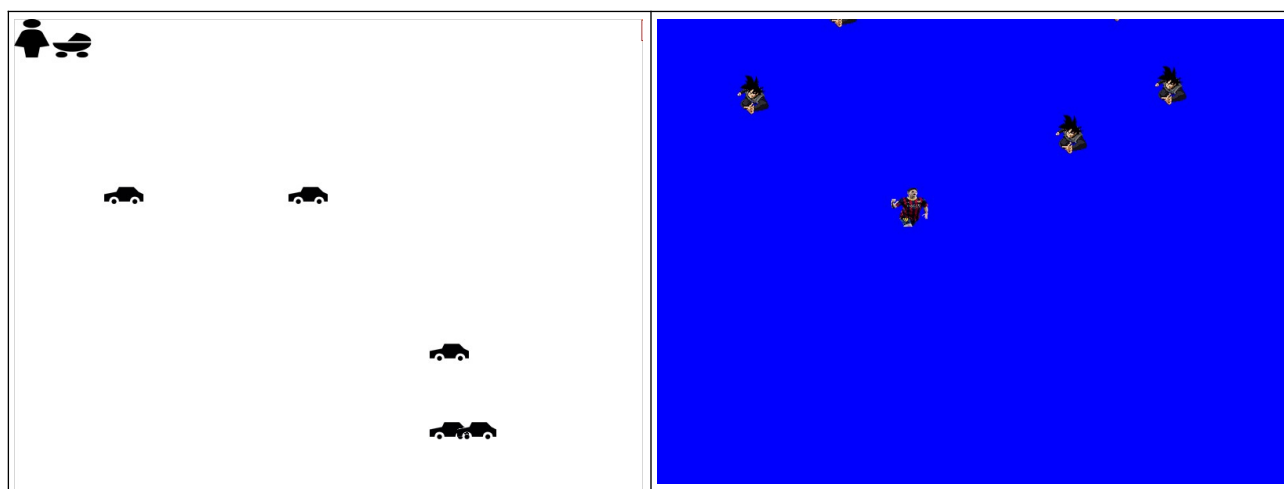
Tabela 1: Características adicionais implementadas.

TIPO DA CARACTERÍSTICA	PROJETOS QUE UTILIZARAM
PERSONAGEM	8
CENÁRIO	6
SOM	3

Fonte: o autor

Na tabela 1 temos jogos que foram incluídos em mais de uma contagem, portanto, a soma ultrapassa o total de projetos desenvolvidos. Por exemplo, os três jogos que implementaram o som também implementaram as demais propriedades. Com relação às propriedades gráficas, apenas dois projetos não acrescentaram cenários no ambiente, conforme mostrado no quadro 8.

Quadro 8: Jogos sem cenários temáticos, mas com personagens



Fonte: o autor

Dos projetos desenvolvidos, três implementaram todas as propriedades opcionais e são listados no quadro 9 com a descrição de todas as características implementadas, como cenário de fundo, figuras para os personagens, som ou formas diferenciadas de apresentação dos resultados e do início do jogo.

Quadro 9: Jogos com todas as propriedades adicionais implementadas.



Tela inicial, ambiente com fundo gráfico.

Personagem do jogador muda a visualização quando colide. Música de fundo inicia após 10 segundos de jogo.

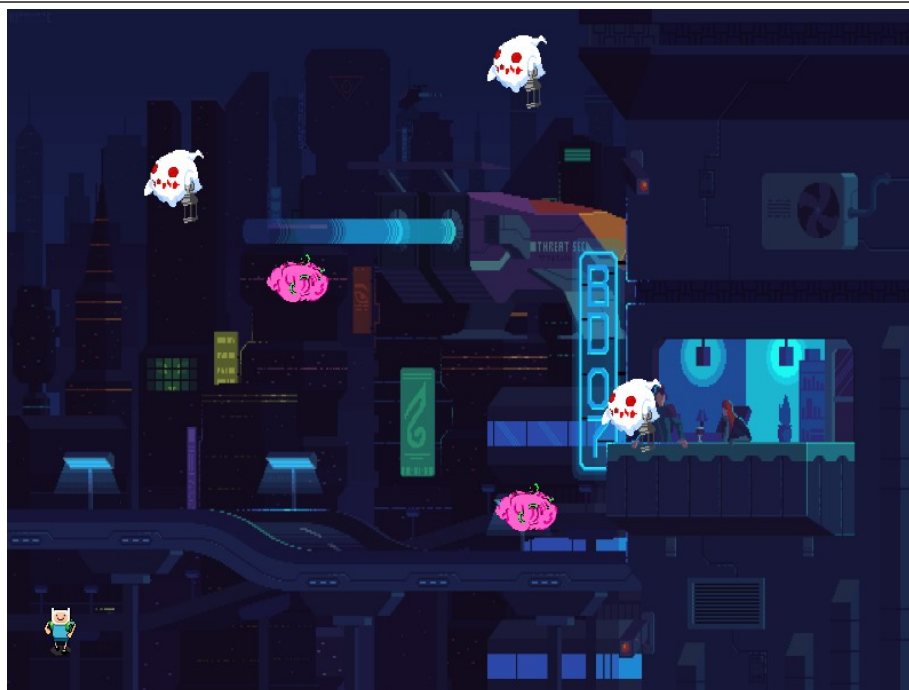
Status de vida e indicação de fim de jogo com animação.



Tela de início do jogo, com música de fundo.

Personagem do jogador muda visualização cada vez que é atingido e é gerado uma animação de explosão quando perde todas as vidas.

Dois tipos de gráficos são utilizados para os inimigos.



Dois diferentes tipos de figuras para os inimigos, fundo animado e som de fundo tocando durante todo o jogo.

Sem informações de início e status do jogador.

Fonte: o autor

No anexo D estão as telas dos demais jogos entregues que implementaram características visuais adicionais durante o seu desenvolvimento.

5.3 Observação participante

O pesquisador desenvolveu todas as etapas da pesquisa, com a construção do material explicativo, os códigos utilizados como exemplo dos conceitos de movimentação e captura de eventos, os questionários para coleta de dados antes e depois do desenvolvimento do jogo e as entrevistas feitas após o término da disciplina. O pesquisador também atuou de forma participante durante o desenvolvimento do projeto pelos alunos, tirando dúvidas sobre o código sendo desenvolvido sempre que era acionado.

Durante o acompanhamento do projeto feito pelo pesquisador, foi criado um registro diário sobre as dúvidas dos alunos, o tipo de atividade em execução e a movimentação em aula. No quadro 10, temos o resumo do registro de acompanhamento feito em aula durante o desenvolvimento do projeto.

Quadro 10: Registro diário de acompanhamento

DATA	REGISTRO
27/08/18	Questionário do perfil. Aplicado em aula. Maioria dos alunos responderam.
28/08/18	Apresentação da proposta de jogo. Receio de alguns. Dúvidas sobre uso de figuras. Alguns empolgados com a proposta.
03/09/18	Maioria pesquisando figuras para usar no jogo. Um aluno questionando sobre o código de movimentação.
04/09/18	Dúvidas específicas dos comandos JavaScript para movimento (código dos slides). 2 alunos ainda não apareceram. Alunos pesquisando sobre uso de imagens.
10/09/18	Dúvidas pontuais: erros de código. Um aluno pesquisando sobre canvas.
11/09/18	Dúvidas pontuais: erros de código.
17/09/18	Dúvida 3 alunos: dificuldades no entendimento de colisão. Explicação geral para todos da sala.
18/09/18	Dúvida colisão: dificuldade em entender as coordenadas dos objetos.
24/09/18	Dúvidas pontuais: erros de código.
25/09/18	Entrega e apresentação do jogo.

Fonte: o autor.

Durante as aulas, os alunos ficaram livres para testar os comandos e exemplos apresentados pelo professor. Também estavam liberados para pesquisar outros exemplos e formas de movimentação de objetos assim como discutir com seus colegas sobre o desenvolvimento do jogo.

Observou-se que, apesar de não ser obrigatório e não pontuar na entrega do projeto, a pesquisa e teste de figuras e imagens para serem utilizadas nos jogos foi uma

atividade desenvolvida por metade dos alunos e levou a várias movimentações em sala de aula, com alunos apresentando para os colegas os temas (figuras) que tinham encontrado e como utilizariam em seu projeto.

Estas pesquisas sobre o aspecto visual dos jogos fizeram dois alunos trazerem questões sobre modelagem gráfica em 3D e sua utilização em jogos, assuntos abordados nos sites encontrados nas buscas em aula. Outro tópico questionado por mais de um aluno foi sobre o conceito da TAG HTML *<canvas>* que permitiria desenhos mais detalhados dentro do navegador. Estes dois assuntos (modelagem 3D e *canvas*) foram explanados brevemente pelo pesquisador, mas sugeriu-se deixar para pesquisas posteriores para não impactar no prazo de desenvolvimento do projeto.

O tema que gerou a maior quantidade de dúvidas foi a parte do jogo relacionada com a colisão dos itens na tela. Os alunos não encontraram facilmente explicações sobre este tipo de algoritmo na internet, o que gerou várias demandas de explicações dos alunos para o pesquisador. Os alunos que ao final não conseguiram completar o projeto, tiveram dificuldades nesta questão específica.

Uma situação presenciada pelo pesquisador nas aulas e com mais intensidade nas duas últimas aulas do projeto, foi a troca de informações feitas entre os participantes, com alunos parando sua codificação para explicar ou auxiliar outro aluno a encontrar algum erro nos seus testes. Somente quando o colega não conseguia identificar o erro, era chamado o professor pesquisador para auxiliar.

O pesquisador acompanhou todas as apresentações dos projetos até a data final, quando os alunos operavam o jogo para demonstrar o funcionamento e depois precisavam mostrar o código fonte e explicar alguns trechos do programa que tinham desenvolvido.

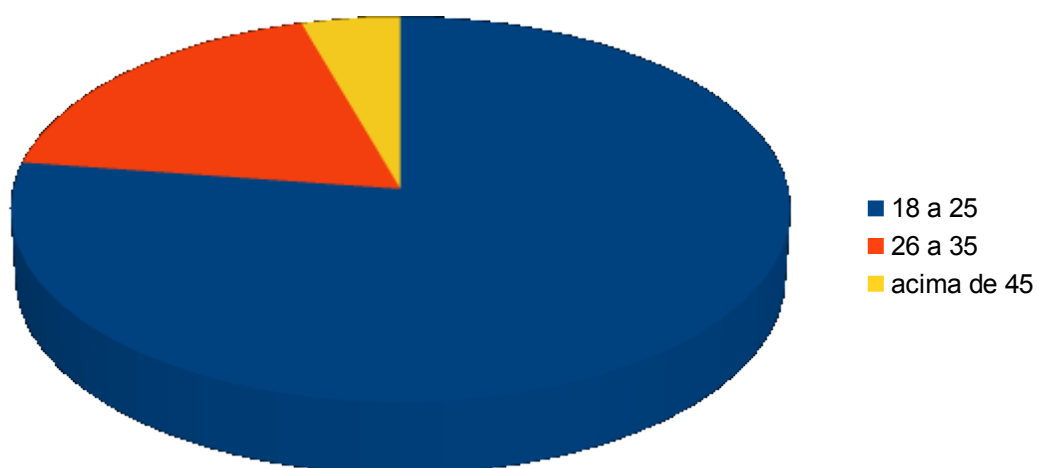
5.4 Análise dos dados

Os dados coletados utilizando os formulários foram analisados e permitiram a identificar o perfil dos participantes da disciplina e qual a avaliação eles fizeram do projeto. Sobre a amostra utilizada, no início da disciplina haviam 22 alunos identificados na lista de chamada e este foi o número de respostas ao primeiro formulário antes do início do projeto. Antes da fase de apresentação do projeto dois alunos precisaram trancar a matrícula e não participaram da disciplina.

5.4.1 Perfil dos participantes

Os primeiros dados coletados, no início da disciplina utilizada para o desenvolvimento da pesquisa, revelam algumas características do perfil desta amostragem de alunos. Com relação a idade dos participantes, a maioria (77%) tinha entre 18 e 25 anos.

Figura 14: Idade dos participantes



Fonte: o autor.

Sobre o conhecimento de programação, a maioria (59%) não tinha conhecimento de programação antes de entrar na faculdade (ver tabela 2). Um terço (33%) informou ter alguma experiência com programação antes da faculdade, sendo através de curso, em sua atividade profissional ou de forma autodidata.

Tabela 2 - Conhecimento de programação anterior a faculdade

Item	Qtd	Percentual
Não tinha nenhum conhecimento	13	59%
Já tinha tentado aprender sozinho	5	23%
Conhecia pessoas que já trabalhavam com programação	2	9%
Fiz curso de programação	1	5%
Trabalhei/trabalho com programação	1	5%
COMPLETO GERAL	22	100%

Fonte: o autor.

Nesta amostra, o primeiro contato com programação foi na faculdade, para a maioria dos alunos. Isto mostra a importância que tem as primeiras disciplinas de ensino de algoritmos e programação para este tipo de aluno.

5.4.2 Dificuldades com desenvolvimento de algoritmos

Neste curso, antes da atual disciplina de programação web, os alunos passam por uma disciplina de algoritmos e estrutura de dados e outra de linguagem de programação, onde tiveram contato com a linguagem de programação C e linguagem de programação Java (respectivamente). Quando perguntados sobre o nível de dificuldade que possuíam com relação a desenvolvimento de algoritmos, temos na tabela 3 a classificação das respostas com relação a dificuldades no desenvolvimento de algoritmos em sua forma geral, dos 22 respondentes do primeiro formulário apresentado. Mais da metade dos alunos indicaram estar num nível mediano de dificuldade (nível 3).

Tabela 3 - Com relação ao nível de dificuldade no desenvolvimento de algoritmos de modo geral:

Item	Qtd	Percentual
Nível de dificuldade 1 (pouca dificuldade)	4	18%
Nível de dificuldade 2	3	14%
Nível de dificuldade 3	12	55%
Nível de dificuldade 4	2	9%
Nível de dificuldade 5 (Muita dificuldade)	1	5%
Total respondentes	22	

Fonte: o autor.

Quando perguntados sobre o conhecimento de estruturas de controle específicas utilizadas em programas de computador, os alunos ainda não sentem segurança nos seus conhecimentos em estruturas básicas como as de decisão e repetição. Na tabela 4, foram agrupadas as respostas sobre as dificuldades específicas nas estruturas de programação de repetição, decisão e array.

Tabela 4 - Dificuldades específicas em estruturas de linguagens de programação

Item	Repetição	Decisão	Array
Nível de dificuldade 1 (pouca dificuldade)	4	4	2

Nível de dificuldade 2	3	3	1
Nível de dificuldade 3	10	13	8
Nível de dificuldade 4	5	2	10
Nível de dificuldade 5 (Muita dificuldade)	0	0	1
Total respondentes	22	22	22

Fonte: o autor.

Apesar de poucos alunos afirmarem ter dificuldade alta (nível 4 e 5) com estruturas básicas de programação (repetição e decisão), muitos alunos (50%) identificaram ter dificuldade (nível 4 e 5) com "*Listas e Arrays*".

Uma percepção do pesquisador ao acompanhar os alunos em seu início de desenvolvimento do projeto foi que pode-se perceber a dificuldade do aluno em usar os conhecimentos conceituais na programação do jogo. Este tipo de conhecimento de programação, chamado de estratégico, conforme a categorização de Qian e Lehamn (2017), pode ser difícil de coletar ao perguntar sobre o nível de dificuldade com estruturas das linguagens como decisão e repetição. Neste projeto, os alunos tinham dificuldade de aplicar as diferentes estruturas de decisão necessárias para o jogo, apesar de entenderem a sintaxe da estrutura "*if*" do JavaScript.

5.4.3 Interesse em jogos digitais

Também no primeiro questionário, foram feitas perguntas fechadas com respostas dentro da escala *Likert* para coletar o interesse dos alunos em jogos digitais. Considerando que uma das metodologias envolve o desenvolvimento de um jogo digital, esta questão permite avaliar o possível nível de engajamento.

Tabela 5 - Frequência do uso de jogos.

Item	Qtd	Percentual
Ocasionalmente	7	32%
Todo dia	7	32%
2 a 4 vezes por semana	5	23%
Raramente	3	14%
COMPLETO	22	100%

Fonte: o autor.

Nesta amostra (tabela 5), mais da metade dos alunos (54%), jogam frequentemente: todo dia ou 2 a 4 vezes por semana. Sendo que um terço da amostra joga todos os dias. Esta característica apresenta o tipo de estímulo e associação com a realidade do aluno, que um tópico sobre jogos digitais pode provocar ao ser utilizado como questão motivadora.

Quadro 11 - Idades e frequência no uso de jogos

IDADE	Tipo de frequência	Freq. uso de jogos
18 a 25	2 a 4 vezes por semana	5
	Ocasionalmente	5
	Todo dia	7
26 a 35	Ocasionalmente	2
	Raramente	2
Acima de 45	Raramente	1
Total		22

Fonte: o autor.

No quadro 11, foi associado a frequência de uso dos jogos com a idade dos alunos. Quando relacionada a idade com a frequência no uso de jogos, é possível verificar que os alunos mais jovens jogam com maior frequência do que os com maior idade.

5.5 Análise do desenvolvimento do jogo

Após o desenvolvimento e entrega da atividade de programação do jogo, os alunos foram convidados a participar de uma *survey* para avaliar sua percepção com relação ao tipo de atividade realizada. Para fins de análise das respostas deste questionário, foram considerados apenas os respondentes que efetivamente participaram do desenvolvimento do projeto (entregaram alguma das fases obrigatórias do projeto). 16 dos 20 alunos que participaram da apresentação inicial do projeto desenvolveram e entregaram um jogo, mesmo que parcialmente. 4 alunos não entregaram nenhuma parte do projeto, sendo:

- Um dos alunos da amostra inicial participou da apresentação do projeto mas não desenvolveu o projeto nem participou do restante das aulas, apesar de não ter trancado a disciplina.
- Dois alunos participaram das aulas, fizeram alguns dos testes iniciais e desistiram. Participaram das aulas acompanhando seus colegas, mas não entregaram nenhum código.

- Um aluno comentou que apesar de estar matriculado na disciplina em questão, não havia feito as disciplinas anteriores de programação e estava inseguro com a programação.

Para esta fase, dos 16 alunos que participaram do projeto, 9 alunos responderam ao segundo formulário de pesquisa.

O trabalho de desenvolvimento de jogo foi dividido em 3 entregas parciais, com dificuldade crescente:

- 1) o controle do objeto do jogador e tela principal – envolve a criação dos limites da tela onde os objetos podem se mover, a captura das teclas e o movimento do jogador.
- 2) o movimento dos objetos que avançam – envolve a movimentação simultânea dos 5 objetos que vão criar obstáculos para a movimentação do jogador e deve fazer o jogador perder vida quando implementada a colisão.
- 3) O mecanismo de colisão – deve interromper o movimento do jogador e do objeto e calcular a perda de uma das 3 vidas.

Para classificar evolução do desenvolvimento do trabalho, foram considerados os seguintes estados na entrega dos projetos:

1. PARCIAL, caso qualquer um dos 3 objetivos não for entregue completo (em funcionamento). 10 alunos entregaram o projeto de forma incompleta. Em todos os casos, ficou incompleto o mecanismo de colisão dos objetos.
2. COMPLETO, se todos os 3 objetivos forem entregues, estiverem completos e em funcionamento. 6 alunos finalizaram todos os objetivos obrigatórios.

No quadro 12, temos uma a associação da entrega com a avaliação do quanto motivador foi o desenvolvimento de um jogo na disciplina. Neste quadro, foram relacionados os respondentes do segundo questionário, sendo que o marcador 1 indicava que o projeto havia sido pouco motivador e o marcador 5 sendo muito motivador. Dos nove alunos que responderam e fizeram o projeto, seis alunos consideraram motivador/muito motivador (marcador 4 e 5) o projeto, mesmo que apenas um aluno destes seis conseguiu desenvolver todas as etapas obrigatórias na sua totalidade.

Quadro 12 - Quem achou boa a utilização do jogo, completou o projeto?

PARTICIPANTE	O quanto você considera que o desenvolvimento de um jogo foi motivador? (5 – muito motivador)	ENTREGA
A16	3	COMPLETO

A13	3	PARCIAL
A23	3	PARCIAL
A12	4	PARCIAL
A08	4	PARCIAL
A19	4	PARCIAL
A07	4	PARCIAL
A01	5	COMPLETO
A14	5	PARCIAL

Fonte: o autor.

No quadro 13, foi feita uma associação entre a questão sobre a dificuldade de programação no geral respondida no primeiro questionário com a dificuldade encontrada na implementação do jogo durante as aulas.

Quadro 13 - Comparação do nível de dificuldade na execução do projeto com o nível de dificuldade geral antes do projeto.

PARTICIPANTE	Assinale abaixo o seu nível de dificuldade na criação de algoritmos de modo geral. 5 (Muita dificuldade)	Qual o nível de dificuldade você teve na implementação do jogo proposto nas aulas? 5 (Muita dificuldade)	ENTREGA
A16	1	1	COMPLETO
A01	2	3	COMPLETO
A12	3	3	PARCIAL
A07	4	3	PARCIAL
A23	-	3	PARCIAL
A08	3	4	PARCIAL
A13	3	4	PARCIAL
A19	3	4	PARCIAL
A14	3	4	PARCIAL

Fonte: o autor.

Os dois alunos que responderam o segundo formulário e que completaram o desenvolvimento do jogo, não consideravam ter dificuldade em desenvolvimento e também indicaram que não tiveram dificuldade na implementação do código. Alunos (A08,

A13, A19 e A14) que não consideravam ter dificuldade na criação de algoritmos relataram que tiveram dificuldade na implementação do jogo, o que pode ser refletido também na entrega parcial do projeto.

No quadro 14, temos as respostas abertas, solicitadas no segundo questionário, onde o aluno poderia indicar as dificuldades encontradas. É possível ver nas falas dos alunos como: "A dificuldade em si não foram as questões /propostas do jogo e sim a programação básica" e "Sem dúvida que foi a lógica de programação", que existe dificuldade no desenvolvimento de algoritmos de modo geral.

Quadro 14 - Dificuldades no desenvolvimento do jogo.

PARTICIPANTE	Que tipo de dificuldade você teve no desenvolvimento deste jogo proposto em aula e que poderia ser melhorado em sua visão?
A01	Mais liberdade na criação do jogo.
A07	na configuração das teclas para subir e descer o jogador
A08	A dificuldade em si não foram as questões /propostas do jogo e sim a programação básica que como a maioria não consegui pegar nas matérias passadas por questão de explicação.
A12	Dificuldade no desenvolvimento de colisão, o foco maior poderia ser feito na colisão.
A13	Devia ter me esforçado um pouco mais
A14	Poucas horas de aula. Deveria ter uma disciplina especifica para JavaScript.
A16	Mudar o jogo 3 vcs pq a primeira vez fiz em Canvas ,, depois fazer em vertical e ter q mudar para horizontal
A19	Sem duvida que foi a lógica de programação. A melhora é no começo, focar um pouco mais na logica, porém dependera muito do aluno estudar.
A23	Bastante com a linguagem devido a falta de esforço de minha parte

Fonte: o autor.

No quadro 15, temos as respostas abertas, solicitadas no segundo questionário, onde o aluno poderia indicar sugestões para incentivar o desenvolvimento de programas em JavaScript. A proposta de uso de jogos para motivar a programar foi enfatizada por vários alunos, sendo 44% dos alunos respondentes, citaram especificamente o desenvolvimento de jogos como opção para motivar a programar em JavaScript.

Quadro 15 - Sugestões para desenvolvimento de programas.

PARTICIPANTE	O que você poderia sugerir, em termos de exercícios voltados a programação, que motivasse você a se envolver mais com desenvolvimento de programas em JavaScript?
A01	Mais prática de código em sala, ao invés de tanta teoria.
A07	mais jogos e criação de sites ou páginas web
A08	A proposta do jogo está excelente.
A12	Mais exercícios de colisão.
A13	Poderia ter mais aulas
A14	Exercício de jogos
A16	Mini projetos ...não só mostrar pedaços de códigos
A19	Exercícios que interajam mais na interface web, exemplo: http://www.filippobello.com/portfolio
A23	O jogo mesmo, devido a ser uma experiência legal ou algum código de criptografia

Fonte: o autor.

Mesmo nos casos em que os alunos que não conseguiram completar a atividade, o desenvolvimento do jogo teve boa aceitação. Dos alunos que entregaram parcialmente o projeto (10 alunos), a maioria 60% investiu no desenvolvimento de características adicionais como PERSONAGEM, CENÁRIO ou ambos.

5.5.1 Entrevista com participantes do projeto

Após a verificação dos comentários dos alunos com relação a boa aceitação da utilização do jogo na disciplina, iniciou a fase das entrevistas com o objetivo de buscar uma maior compreensão sobre qual a influência do tema de jogos digitais na aprendizagem da linguagem JavaScript. Desta forma, foi estruturado um processo de entrevistas com os alunos, que foram gravados e transcritos para análise posterior.

Foram enviados e-mails para os alunos que participaram da disciplina, solicitando a participação voluntária na entrevista e, mesmo sendo fora do horário e já tendo finalizado a disciplina, 9 alunos concordaram em contribuir.

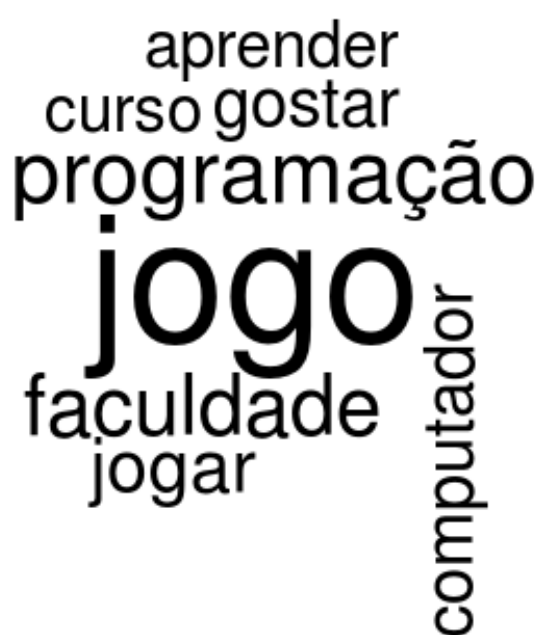
As entrevistas com os alunos foram feitas com um roteiro semiestruturado (apêndice 1), envolvendo as temáticas:

1. Experiência anterior com informática antes da faculdade.
2. Qual o interesse em jogos digitais.
3. Motivação sobre o desenvolvimento do jogo na disciplina.
4. Motivação em estudar informática
5. Perspectiva sobre a continuidade na área de informática.

Nas entrevistas, os alunos responderam as questões pré formuladas sobre os temas e que foram posteriormente analisadas. Após a entrevista, as gravações eram transcritas utilizando a opção "Gravação por voz" do editor de texto do pacote *G Suite* (Google).

O conteúdo das respostas foi analisado e também associado ao fato de ter sido aprovado ou não na disciplina. Dos 9 alunos que aceitaram participar das entrevistas, mais da metade (55%) havia sido aprovado na disciplina, mas o pesquisador também considerou de grande valia a participação de alunos que reprovaram na disciplina e, mesmo assim, se disponibilizaram a contribuir com a pesquisa. As respostas foram avaliadas e agrupadas em dois tópicos principais: a influência do tema de jogos para o estudante e o aprendizado de programação.

Figura 15: Nuvem de frequência das palavras mais utilizadas



Uma visão das frequências das palavras utilizadas pelos alunos durante as entrevistas pode ser vista na figura 10, que apresenta a imagem da nuvem de palavras capturada nas respostas dos entrevistados.

5.5.1.1 A influência do tema de jogos para o estudante

Na entrevista, procurou-se saber qual influência a atividade de jogar tinha sobre o entrevistado, como lidava com esta atividade atualmente e qual a periodicidade em jogar. Desta forma, procurou-se saber como o tema sobre jogos digitais ainda era percebido pelo entrevistado mesmo após o projeto de desenvolvimento do jogo.

Durante a primeira coleta de dados da pesquisa antes do início do projeto, o grande uso dos jogos digitais já era evidenciado como uma prática frequente para o aluno antes de entrar na faculdade. De maneira geral tanto os alunos que passaram na disciplina quanto os alunos que reprovaram jogavam bastante antes de entrar na faculdade. As entrevistas também mostraram que os alunos ainda gostam bastante de jogar, mas a periodicidade no uso de jogos atualmente varia conforme o aluno esteja empregado ou não. Mesmo assim, nas falas dos entrevistados que já não dispõe de tempo durante a semana, havia a indicação que ainda dedicam um bom tempo para jogar no fim de semana e também mostram a intensidade que o ato de jogar ainda provoca. No quadro 16 são apresentadas alguns trechos da fala dos alunos durante a entrevista, expondo a frequência com que usavam jogos digitais e como fazem atualmente.

Quadro 16 - Trechos das falas dos alunos (P sendo a pergunta do pesquisador).

PARTICIPANTE	TRECHO DA FALA
A01	P: Qual plataforma você joga ? celular <i>tablet</i> console? R: Todos. Todos que eu puder jogar eu jogo.
A02	P: Você gosta ainda de jogos digitais ? R: Gosto. Jogo bastante. Fim de semana sempre jogo.
A06	P: E você joga bastante? ou não? R: Jogo. Hoje em dia já não jogo que nem antigamente. Antigamente eu jogava muito mais, virava assim jogando. Hoje em dia com negócio de trabalho, nessa parte mais adulta, jogo menos. Mas tipo eu gosto bastante dessa área de jogos. Qualquer tipo de mídia de jogos. Qualquer coisa assim eu tô jogando. Qualquer coisa que aparecer.
A08	P: continua jogando ? R: continuo jogando... P: bastante? R: bastante... bastante não por causa da faculdade, mas ainda todo o final de semana tamo lá.
A19	P: Que tipo de jogo você joga? R: Já joguei bastante RPG, jogo de FPS. Hoje em dia deu uma parada, mas antes eu jogava bastante.

A22	<p>P: Em qual plataforma você joga mais hoje em dia?</p> <p>R: Hoje em dia, hoje exatamente, hoje não jogo tanto. Mas mais ou menos um ano atrás eu joguei, jogava muito em console no caso. Seria no Xbox, Playstation. Jogar utilizando o computador eu não cheguei a jogar muito pois eu moro em chácara. Não há uma conexão muito boa com internet e eu acabava muitas vezes perdendo o jogo por causa da queda da internet.</p>
-----	--

Fonte: o autor.

A partir do momento que começaram a ficar limitados pelo tempo e pelas atividades profissionais, os alunos deixaram de utilizar o console ou computador, passando a jogar principalmente em smartphones.

A influência dos jogos levou um dos entrevistados a se interessar pela área de informática, procurando cursos sobre desenvolvimento de jogos: "até comprei uns cursos voltados para jogo. Só que eu me decepcionei porque eu comprei uns cursos da sobre *Unity* só que envolvia muito clicar arrastar e caminhar um boneco". Outro entrevistado comentou que procurou a faculdade de tecnologia devido à necessidade de conhecimento para o desenvolvimento de jogos: "*era mais uma base que eu queria ter. Eu queria mesmo é fazer alguma coisa relacionada à design, mas para fazer jogo*".

O tema relacionado a jogos digitais, além de estimular projetos voltados a educação conforme visto com Connolly et al. (2012), serviu de influência para a escolha de seu curso na faculdade para alguns entrevistados.

5.5.1.2 O aprendizado de programação

Os entrevistados indicaram que começaram a aprender a programar efetivamente na faculdade. Alguns entrevistados indicaram que havia interesse na área de informática antes de entrar na faculdade, mas apenas 3 entrevistados comentaram ter tido interesse em programação, procurando vídeos sobre programação, cursos de linguagem de programação ou uso de códigos para jogos, conforme pode ser verificado na sequência:

- 1 - "só tinha visto na internet uma coisa ou outra, uns vídeos. Mas eu não programava antes da faculdade. Não sabia *codar* nada (sic)"
- 2 - "fiz vários cursos na internet sobre Arduíno dai acabou cedo no java"
- 3 - "eu tinha visto antes de entrar na faculdade. Eu vi java com plugins para Minecraft para servidor"

Quando perguntado sobre o projeto de desenvolvimento do jogo na disciplina "Programação WEB", no semestre anterior e se o tema de desenvolver um jogo estimulou os alunos com relação a programar, um dos entrevistados enfatizou a questão da prática

e a importância de uma proposta descontraída para usar na programação: "sim eu acho que tudo que você tem que fazer na pratica que seja mais interativa mais descontraído ou que é um assunto que você gosta ele te incentiva mais a produzir. Que nem o jogo. O pessoal de TI, todo mundo gosta".

De modo geral, os entrevistados demonstraram que, por ter sido um projeto relacionado ao desenvolvimento de um jogo, gerou um incentivo a mais para programá-lo. No quadro 17, alguns trechos de resposta dos entrevistados durante o momento em que eram estimulados a falar sobre a influência do tema no desenvolvimento do projeto.

Quadro 17: Incentivo para programar

PARTICIPANTE	TRECHO DA FALA
A01	<i>"quis fazer o jogo quis tentar fazer pelo menos fazer mesmo se ficou estranho e não atendeu as expectativas"</i>
A02	<i>"mas eu acho interessante, tipo o mecanismo mesmo da programação aquele negócio do joguinho, movimentação é sempre um conhecimento legal a mais de adquirir"</i>
A06	<i>"eu aprendi muito mais fazendo aquela disciplina. Utilizando a parte da programação de jogos eu consegui entender bem mais de JavaScript"</i>
A08	<i>"por causa de até por causa dos games que eu desde pequeno gosto muito de jogo trouxe mais um incentivo pra programação"</i>
A16	<i>"sim ele me deu uma motivação a mais para criar um joguinho, nossa deve ser muito legal cria um joguinho"</i>
A17	<i>"sim, sim. eu não eu não tinha ideia como era feita a física de jogos aí eu fui meio que percebendo por aquilo lá"</i>
A19	<i>"creio que sim. Eu tenho uma dificuldade para programação mas creio que me ajudou um pouco assim principalmente no JavaScript"</i>
A22	<i>"sim, ele, com esse jogo eu consegui ver que é possível fazer muitas coisas. que é uma tela em branco que ali basta ter imaginação se consegue resolver muitas coisas. Mas o desenvolvimento vem com a prática. No começo ali coisas básicas você acaba tendo dificuldade, mas com esses trechos você acaba aprendendo"</i>

Fonte: o autor.

Apenas um entrevistado não demonstrou ter sido encorajado pelo tema a programar; gostou do tema por ser um jogo, mas ficou indiferente sobre a associação do tema do projeto com programação em si.

Dois alunos afirmaram que continuaram a desenvolver o jogo, mesmo depois do término da disciplina. Um dos entrevistados afirmou que reescreveu todo o código para entender melhor os conceitos de orientação a objeto: *"utilizando a parte da programação de jogos eu consegui entender bem mais de JavaScript. Alguns conceitos novos começaram a aparecer tipo que nem comecei a ver mais orientação objeto. Tipo hoje em dia eu criei um jogo novo só com orientação a objeto para ver como seria"*.

Outro entrevistado indicou que continuou desenvolvendo o jogo mesmo após o projeto para a inclusão de mais funcionalidades, conforme visto na fala "*eu fui pesquisando coisas avançadas, fora que você tinha passado*".

5.6 Reflexões sobre a pesquisa

Acompanhar e registrar a evolução dos alunos foi uma oportunidade para o pesquisador refletir sobre a metodologia e fazer associações com a bibliografia pesquisada e os autores investigados.

A primeira reflexão é a crescente importância da programação na área educacional, o que tem estimulado a busca por novas metodologias para sua implantação e incentiva o uso de estratégias para instigar o aluno na apropriação deste conhecimento. Desde Seymour Papert na década de 80, temos acompanhado iniciativas para disseminar habilidades de programação inclusive entre crianças e, sobre essa proposta, nos últimos anos, foram implementadas iniciativas de ensino regular de programação nas escolas em vários países¹³. Cresce nos últimos anos a importância de disseminar também as características do pensamento computacional, reforçando habilidades como abstração e resolução de problemas nos jovens em idade escolar (GROVER, PEA, 2013; WING, 2006). Estas habilidades também são valorizadas para um curso superior como o de "Gestão de Tecnologia da Informação", onde temos a busca de soluções com qualidade e o incentivo ao pensamento reflexivo e a autonomia intelectual, como alguns dos objetivos (SBC, 2017).

Papert (1994) propunha estimular o uso do computador e internet para dar autonomia aos jovens e para que possam ter liberdade na sua busca de conhecimentos, mas não considerando o uso de sequências de conteúdos automatizadas, e sim que estes jovens tivessem a liberdade de desenvolver seu próprio roteiro de aprendizagem. O uso de metodologias ativas vai de encontro com a proposta de tirar o aluno da posição passiva e receptiva, que acaba sendo, conforme indica Mattar (2017, p. 21), "uma educação que pressuponha a atividade (ao contrário da passividade) por parte dos alunos".

Outra reflexão é a constante dificuldade em aprender a programar, identificadas na literatura pesquisada que, por sua vez, tem sido um ponto de discussão e de partida para a investigação de novas técnicas educacionais aplicadas ao ensino de programação

13 Computing our future. Computer programming and coding - Priorities, school curricula and initiatives across Europe (http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03)

(SOUZA; BATISTA; BARBOSA, 2016; BULCÃO; CAMPOS NETO; MOREIRA, 2017; RAABE; SILVA, 2005). Diferentes educadores têm se empenhado em tornar o aprendizado mais agradável e facilitado para os novos entrantes da área (DE RAADT, 2010; GUZDIAL, 2003).

Uma das formas de estimular o aluno com programação de computadores tem sido a introdução do tema relacionado a jogos (uso ou desenvolvimento) em disciplinas tanto no ensino fundamental quanto no ensino superior (SILVA; FERNANDES; SANTOS, 2018). Desenvolver um jogo para aprender uma linguagem não é uma experiência nova para o pesquisador: em sua adolescência, foi o desafio de programar um jogo em um TK90X durante vários finais de semana que o fizeram praticar e consolidar seus conhecimentos na linguagem *Basic*, usada como padrão neste tipo de dispositivo. O método usado foi consulta de listagens de programas em revista de informática da época (Micro Sistemas) e digitando e adaptando trechos de códigos de outros programas que eram disponibilizados nas páginas da revista. Naquela época, entre 1983 e 1984, a consulta de códigos fontes em linguagem *Basic* em revistas era uma das formas de se ter acesso a códigos fontes e aprender programação.

Jenkins (2002, p. 55), apresenta que a dificuldade em programar está relacionada a algumas habilidades essenciais como resolução de problemas e alguma fundamentação em matemática e, ainda, capacidade para testar, investigar erros e corrigi-los. Estas habilidades formariam uma hierarquia onde o aluno seria conduzido por um processo:

As habilidades formam uma hierarquia e o programador vai utilizar muitas delas em a qualquer momento. Um aluno, ao se deparar com o aprendizado de uma hierarquia de habilidades geralmente, aprende a habilidade mais básica primeiro e então, vai progredindo. No caso de codificação (uma pequena parte da habilidade de programação), isto implica que os alunos aprenderão o básico da sintaxe primeiro e depois passarão gradualmente para a semântica, estrutura e, finalmente, estilo (JENKINS, 2002, p. 55).

A proposta de uso de metodologia ativa com a criação de um projeto de produção de um jogo para estimular os alunos de "programação WEB" foi uma oportunidade de convergir alguns elementos identificados acima:

- Experiência positiva do pesquisador na aprendizagem de linguagem de programação com o desenvolvimento de jogo;
- Estudos indicando o uso de jogos como forma de envolvimento dos alunos em sala de aula;

- As dificuldades que os alunos têm demonstrado no aprendizado de uma linguagem de programação de acordo com os autores pesquisados.

O que teve mais impacto para o pesquisador durante o projeto foi o envolvimento dos alunos com o desenvolvimento do jogo e, principalmente, a forma positiva que manifestaram após o término da disciplina.

Para efeito de correlação com sua experiência profissional com desenvolvimento de sistemas, o pesquisador criou um quadro identificando como o desenvolvimento de programas, em muitos casos, se assemelham com algumas características apontadas como fatores de envolvimento nos jogos citadas por Prensky (2001), e fez também uma associação destes fatores com observações feitas durante o projeto, como mostra o quadro 18:

Quadro 18: Fatores de envolvimento dos jogos e a atividade de programação

FATORES	PROGRAMAÇÃO	PROJETO
Regras	A linguagem de programação possui regras de construção (sintaxe e semântica) específicas. O programador precisa obedecê-las para seu código funcionar.	O projeto desenvolvido apresentou, adicionalmente, regras sobre o uso de códigos prontos e os elementos envolvidos.
Metas e objetivos	Um programa executa uma tarefa e deve apresentar o resultado ao usuário. O programador se orienta por este escopo para o desenvolvimento.	O projeto tinha meta específicas para cumprir (com partes de desenvolvimento obrigatório) e uma data final para a entrega do projeto completo.
Resultados e feedback	A cada teste, o programador tem um <i>feedback</i> imediato se o seu programa está funcionando conforme foi idealizado. Nem sempre a mensagem de erro é facilmente entendida o que leva o programador a realizar mais pesquisas para descobrir o novo problema. A cada teste, o resultado é apresentado de forma imediata.	A visualização do código sendo executado com a apresentação imediata do movimento ou o conteúdo correto nas variáveis era comemorado por alguns alunos. Em muitos casos, chamavam os colegas para apresentar o personagem que tinham escolhido se mover na tela.
Conflitos, competição, desafios, oposição	A implementação da tarefa através de uma linguagem de programação desafia a capacidade de abstração do programador.	A perspectiva de desenvolver algo que os alunos eram entusiastas e já conheciam, fez o desafio da empreitada mais estimulante. O uso de personagens e cenários foi uma forma de demonstrar este envolvimento por parte dos alunos.
Interação	Programadores criam comunidades e fóruns de discussão sobre programação. No ambiente de desenvolvimento (virtual ou presencial) são estabelecidas várias formas de	Durante o projeto, alunos ajudavam alunos. Em alguns momentos o corredor estava com 3 ou 4 cadeiras agrupadas ao redor de um dos alunos que mostrava (e explicava) como tinha

	colaboração entre os desenvolvedores.	implementado uma funcionalidade. Mas nem todos se sentiam confortáveis com a interação: um aluno, em duas aulas saiu mais cedo justificando que faria o desenvolvimento em casa onde conseguia se concentrar melhor.
Representação	O programa vai participar de uma solução que pode representar uma mudança na maneira que uma empresa executa uma atividade, pode ser um novo aplicativo de celular que vai auxiliar ou entreter uma criança, um sistema de controle de uma usina nuclear. Várias são as possibilidades em que um programa de computador pode representar algo, concreto ou virtual, direto ou indireto.	Prensky traz o aspecto de significado para este fator, o que para os alunos do projeto consistiu em desenvolver um jogo de ação que se assemelhava a uma atividade que já tinham grande interesse, conforme mostrou o questionário inicial e as entrevistas.

Fonte: o autor.

Podemos concluir que a união de projetos, jogos e programação de computadores teve o estímulo esperado no envolvimento dos alunos ao praticaram os conceitos e regras de uma nova linguagem de programação. Este estímulo é importante para aumentar o envolvimento do aluno com a experimentação e incentivá-lo a praticar com uma linguagem de programação, o que é considerado por diversos autores como um importante fator de aprendizagem, ainda mais considerando que programar computadores é uma habilidade essencialmente prática.

6 CONSIDERAÇÕES FINAIS

Fazer pequenos exercícios descolados da realidade dos alunos pode prejudicar seu envolvimento no aprendizado de uma linguagem de programação, como foi verificado na literatura pesquisada. Proporcionar uma metodologia que desenvolva sua autonomia na busca de informações adicionais, que esteja associada a uma atividade praticada pelo aluno e que possa ser adaptada às restrições de prazo e conteúdo de uma disciplina já existente foram os desafios identificado pelo pesquisador.

Esta pesquisa apresentou a utilização de uma metodologia ativa baseada em projetos e jogos no ensino de uma linguagem de programação de um curso de "Gestão de Tecnologia da Informação" numa faculdade da região metropolitana de Curitiba. A disciplina foi ministrada em 2018, no segundo semestre.

Como objetivo geral, esta pesquisa propôs elaborar uma proposta de metodologia ativa baseada no uso de projeto com foco no desenvolvimento de jogo digital. Esta proposta incentiva o envolvimento ativo do estudante dando ênfase ao seu papel de protagonista.

O processo de investigação desta pesquisa foi de natureza aplicada e de abordagem qualitativa. Quanto ao objetivo, a atual pesquisa foi do tipo exploratória e descritiva. Os dados foram obtidos a partir da aplicação de formulários digitais, questionário semiestruturado em entrevistas e observação participante.

No decorrer desta pesquisa, fez-se a investigação de como os jogos e as metodologias ativas estavam sendo utilizadas na educação de jovens e no ensino superior. Também foram estudadas as dificuldades que envolvem o ensino de uma linguagem de programação e a sua importância para a sociedade atual, criando uma estrutura teórica para o projeto.

Uma proposta foi planejada e aplicada em uma disciplina do ensino superior que tinha a necessidade de envolver os alunos na utilização de duas linguagens de programação no mesmo semestre. Uma das linguagens de programação (JavaScript) foi selecionada como objeto desta pesquisa e foram coletados dados dos alunos antes da aplicação da proposta para identificar o perfil e sua aderência ao tema sobre jogos. Durante o desenvolvimento foi aplicado novo questionário para identificar a percepção do aluno com a programação do jogo e as possibilidades de melhoria na proposta. Também foram coletados dados a partir da observação do pesquisador durante o trajeto do aluno

em todo o projeto. Foram coletadas informações depois de finalizarem a disciplina, por meio de entrevistas com alunos voluntários, num semestre posterior.

A partir da análise dos dados, do acompanhamento do projeto e das entrevistas, foi possível constatar que o uso de metodologia ativa baseada no desenvolvimento de jogos teve um impacto positivo na prática da linguagem de programação. Os alunos ficaram envolvidos com o projeto, buscando inclusive elementos adicionais ao escopo inicial do projeto, mesmo considerando que estes elementos adicionais não seriam pontuados na entrega.

Apesar de a pesquisa ter trazido resultados relevantes para a área, algumas limitações devem ser relacionadas. Uma limitação do trabalho foi a utilização da proposta num curto espaço da disciplina, sendo necessário diminuir a margem de implementação de objetos e eventos nos jogos. Os aspectos visuais e sonoros, assim como a jogabilidade poderiam ter sido exploradas de uma forma mais aprofundada pelos alunos, que utilizaram parte do tempo do desenvolvimento para tornar seus jogos mais sofisticados, com o uso de personagens e cenários temáticos, mas tiveram dificuldades de cumprir o prazo final.

Outra limitação foi a disciplina, onde a proposta foi aplicada, ter duas linguagens de programação, disputando a atenção dos alunos dentro do mesmo semestre. Apesar desta limitação ter motivado o uso de metodologias ativas e a incorporação do uso de jogos no projeto, a aplicação deste tipo de projeto em disciplina com foco em uma única linguagem de programação poderia aprofundar o conhecimento em JavaScript, com a criação de algoritmos mais complexos.

Como trabalhos futuros, sugere-se utilizar o material produzido em uma disciplina inicial de curso de informática onde a linguagem JavaScript seja a opção de linguagem de programação inicial dos alunos entrantes e a proposta do projeto de desenvolvimento do jogo possa envolver os alunos iniciantes no ambiente de programação. Esta perspectiva do uso de JavaScript como linguagem de programação inicial vem sendo pesquisada como uma alternativa em universidades como Stanford¹⁴ e Creighton (REED, 2001)

A facilidade com que os alunos auxiliavam e explicavam para os demais colegas os trechos de código e como tinham implementado algumas rotinas, poderia ser melhor estimulada com a utilização do conceito de "*peer instruction*" (instrução por pares). Nesta metodologia, caso o entendimento dos alunos ainda não seja suficiente, seus pares

14 CS department updates introductory courses <<https://www.stanforddaily.com/2017/02/28/cs-department-updates-introductory-courses/>>.

podem auxiliar, criando um ambiente cooperativo de aprendizagem. O material sobre os conceitos usados nos jogos e o funcionamento básico do JavaScript poderiam ser trabalhados utilizando esta metodologia.

Outra possibilidade que poderia ser melhor explorada é a apresentação dos projetos finalizados para os demais alunos do curso. Esta atividade de encerramento e apresentação é explorada em metodologia de Aprendizagem Baseada em Projeto e pode motivar o aluno a desenvolver mais características adicionais ao projeto, potencializando a prática da linguagem de programação. Este tipo de fechamento do projeto também é visto no desenvolvimento de jogos profissionais (conhecido também como o "*post mortem*"¹⁵) onde os resultados do jogo são divulgados e os aspectos do processo de desenvolvimento, como os acertos, as dificuldades e os ensinamentos são compartilhados com o público.

15 POST MORTEM SBGAMES <<https://www.sbgames.org/sbgames2018/palestras-post-mortem>>

REFERÊNCIAS

- AGFUNDER. AgFunder Agrifood Tech Investing Report 2018. Disponível em: <<https://agfunder.com/research/agrifood-tech-investing-report-2018/>>. Acesso em: 14 nov. 2019
- AMBRÓSIO , A. P.; COSTA, F. M. **O uso de PBL para o Ensino de Algoritmos e Programação de Computadores**. PBL 2010 Congresso Internacional. São Paulo, Brasil, 8-12 de fevereiro de 2010
- AP STUDENTS. Advanced Placement (AP) – The College Board. Disponível em: <<https://apstudents.collegeboard.org/>>. Acesso em: 15 set. 2019
- APACHE. The apache software foundation celebrates. Disponível em: <<https://blogs.apache.org/foundation/entry/the-apache-software-foundation-celebrates1>>. Acesso em: 02 nov. 2019.
- BACICH, Lilian; MORAN, José. **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática**. Penso Editora, 2018.
- BARBOSA, Eduardo Fernandes; DE MOURA, Dácio Guimarães. Metodologias ativas de aprendizagem na educação profissional e tecnológica. **Boletim Técnico do Senac**, v. 39, n. 2, p. 48-67, 2013.
- BENNEDSEN, Jens; CASPERSEN, Michael E. Failure rates in introductory programming. **AcM SIGcSE Bulletin**, v. 39, n. 2, p. 32-36, 2007.
- BERBEL, Neusi Aparecida Navas. As metodologias ativas e a promoção da autonomia de estudantes. **Semina: Ciências Sociais e Humanas**, v. 32, n. 1, p. 25-40, 2011.
- BLUMENFELD, Phyllis C. et al. Motivating project-based learning: Sustaining the doing, supporting the learning. **Educational psychologist**, v. 26, n. 3-4, p. 369-398, 1991
- BRASSCOM. Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação – Brasscom, **Relatório Setorial de TIC: Inteligência e Informação**. 2019. Disponível em: <<https://brasscom.org.br/relatorio-setorial-de-tic-2019/>>. Acesso em: 27 maio 2019.
- BULCÃO, Jeanne da Silva B., CAMPOS NETO , Edmilson B., MOREIRA, Keila Cruz. **Mapeamento sobre o Ensino de Algoritmo e Lógica Computacional nos Cursos de Licenciatura em Informática e Computação em Instituições de Ensino Superior no Brasil**. II Congresso sobre Tecnologias na Educação (Ctrl+E 2017).
- CONNOLLY, Thomas M. et al. A systematic literature review of empirical evidence on computer games and serious games. **Computers & education**, v. 59, n. 2, p. 661-686, 2012.
- CORMEN, Thomas H. et al. Introduction to algorithms. MIT press, 2009.

CS10K. CS10K Community. Disponível em: <<http://cs10kcommunity.org/>>. Acesso em: 10 jun. 2019

DE RAADT, Michael , et al. **Approaches to learning in computer programming students and their effect on success**. HERDSA-ACM. 2005.

DE RAADT, Michael. Introductory programming in a web context. In: **Conferences in Research and Practice in Information Technology (CRPIT)**. Australian Computer Society Inc., 2010. p. 79-86.

DUCHARME, C. Historical Roots of the Project Approach in the United States. **Annual Conventoin of the National ASSociation for the Education of Young Children, Anaheim, CA**, 1993.

EUROPEAN SCHOOLNET. Computing our future. Computer programming and coding - Priorities, school curricula and initiatives across Europe. Disponível em: <http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03>. Acesso em: 5 out. 2019

EXAME. **A economia dos apps**. São Paulo: Abril, edição 1187, ano 53, .n. 7, 17 abril 2019, p. 18-29

FEBRABAN. Pesquisa FEBRABAN de Tecnologia Bancária. Disponível em: <https://cmsportal.febraban.org.br/Arquivos/documentos/PDF/febraban_2018_Final.pdf>. Acesso em: 15 nov. 2019.

GIL, Antonio Carlos. Como elaborar projetos de pesquisa. - 4. ed. - São Paulo : Atlas, 2002

GIRAFFA, Maria Martins; MORA, Michael Costa. Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. In: **Congressos CLABES**. 2013.

GROVER, Shuchi; BASU, Satabdi. Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic. In: **Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education**. 2017. p. 267-272.

GROVER, Shuchi; PEA, Roy. Computational thinking in K–12: A review of the state of the field. **Educational researcher**, v. 42, n. 1, p. 38-43, 2013.

GUZDIAL, Mark. A media computation course for non-majors. In: **Proceedings of the 8th annual conference on Innovation and technology in computer science education**. 2003. p. 104-108.

INEP. **Brasil no Pisa 2015 – Relatório Nacional**. Disponível em: <http://download.inep.gov.br/acoes_internacionais/pisa/resultados/2015/pisa_2015_brazil_prt.pdf>. Acesso em: 17 jan. 2019

ISA, Noor Azizah Mat; DERUS, Siti Rosminah. **Students experience in learning Fundamental Programming: An analysis by gender perception**. Advanced Journal of Technical and Vocational Education, 2017

JENKINS, Tony. On the difficulty of learning to program. In: **Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences**. 2002. p. 53-58.

K12CS. K–12 Computer Science Framework . Disponível em: <<https://k12cs.org/>>. Acesso em: 10 out. 2019

KANODE, Christopher M.; HADDAD, Hisham M. Software engineering challenges in game development. In: **2009 Sixth International Conference on Information Technology: New Generations**. IEEE, 2009. p. 260-265.

LAHTINEN, Essi; ALA-MUTKA , Kirsti; JÄRVINEN, Hannu-Matti. **A Study of the Difficulties of Novice Programmers**. ITiCSE'05- ACM.

LEUTENEGGER, Scott; EDGINGTON, Jeffrey. A games first approach to teaching introductory programming. In: **Proceedings of the 38th SIGCSE technical symposium on Computer science education**. 2007. p. 115-118.

MANZANO, J.A.N.G.; OLIVEIRA, J. F. Algoritmos: Lógica para Desenvolvimento de Programação de Computadores. **Editora Érica, edição**, v. 27, 2014.

MATTAR, João. GAMES EM EDUCAÇÃO - COMO OS NATIVOS DIGITAIS APRENDEM. São paulo: pearson prentice hall, 2010.

MATTAR, João. METODOLOGIAS ATIVAS: para a educação presencial, blended e a distância. **São Paulo: Artesanato Educacional**, p. 28-29, 2017.

MEDEIROS, L. F. ; WÜNSCH, L. P. Ensino de programação em robótica com Arduino para alunos do ensino fundamental: relato de experiência. **Revista Espaço Pedagógico**, v. 26, n. 2, p. 456-480, 2019.

MORAN, J. M. Metodologias Ativas e Modelos Híbridos na Educação. In: YAEHASHI, S. et al. (Orgs). **Novas Tecnologias Digitais: Reflexões sobre mediação, aprendizagem e desenvolvimento**. Curitiba: CRV, 2017, p.23-35

MOREIRA, H.; CALEFFE, L. G. Metodologia da pesquisa para o professor pesquisador. - Rio de Janeiro: Lamparina. 2008.

MOURA, D. G. De.; BARBOSA, E. F. Trabalhando com projetos: planejamento e gestão de projetos educacionais. 8a edição. Petrópolis: Vozes, 2013.

NATIONAL RESEARCH COUNCIL et al. **How people learn: Brain, mind, experience, and school: Expanded edition**. National Academies Press, 2000.

NIEBORG, D. B. (2004). **America's Army: More Than a Game**. Paper presented at the Transforming Knowledge into Action through Gaming and Simulation, Munchen: SAGSAGA.

PAPERT, Seymour M. **A Máquina das Crianças: Repensando a Escola na Era da Informática** . Porto Alegre, Artes Médicas, 1994.

PAPERT, Seymour. Does easy do it? Children, games, and learning. **Game Developer**, v. 5, n. 6, p. 88, 1998.

PEARS, Arnold et al. **A survey of literature on the teaching of introductory programming**. In: ACM sigcse bulletin. ACM, 2007. p. 204-223.

PERLA, Peter P.; MCGRADY, E. D. Why wargaming works. **Naval War College Review**, v. 64, n. 3, p. 111-130, 2011.

POLLARI, Ian; RUDDENKLAU, Anton. The Pulse of fintech 2018. **Biannual global analysis of investment in fintech**. Disponível em: <<https://assets.kpmg/content/dam/kpmg/xx/pdf/2019/02/the-pulse-of-fintech-2018.pdf>>. Acesso em: 15 nov. 2019.

PORTER, Leo et al. **Success in introductory programming: what works?**. Communications of the ACM, v. 56, n. 8, p. 34-36, 2013.

PRENSKY, Marc. Digital game-based learning. **Computers in Entertainment (CIE)**, v. 1, n. 1, p. 21-21, 2003.

PRENSKY, Marc. Fun, play and games: What makes games engaging. **Digital game-based learning**, v. 5, n. 1, p. 5-31, 2001.

PRINCE, M.J. ; FELDER, R.M. (2006). **Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases**, Journal of Engineering Education. 95(2) 123–138. Disponível em: <http://champs.cecs.ucf.edu/Library/Journal_Articles/pdfs/Inductive%20teaching%20and%20learning%20methods.pdf>. Acesso em: 25 jun. 2019.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar. Metodologia do trabalho científico [recurso eletrônico] : métodos e técnicas da pesquisa e do trabalho acadêmico – 2. ed. – Novo Hamburgo: Feevale, 2013.

QIAN, Yizhou; LEHMAN, James. Students' misconceptions and other difficulties in introductory programming: A literature review. **ACM Transactions on Computing Education (TOCE)**, v. 18, n. 1, p. 1-24, 2017.

RAABE, André Luís Alice; SILVA, J. M. C. da. Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. In: **XIII Workshop de Educação em Computação (WEI'2005)**. São Leopoldo, RS, Brasil. 2005.

RAMOS, D. K.; FRONZA, F. C. A.; CARDOSO, F. L. (2018). **Jogos eletrônicos e funções executivas de universitários**. Estudos de Psicologia (Campinas), 35(2), 217-228. <http://dx.doi.org/10.1590/1982-02752018000200010>

REED, David. Rethinking CS0 with javascript. **ACM SIGCSE Bulletin**, v. 33, n. 1, p. 100-104, 2001.

ROBINS, Anthony; ROUNTREE, Janet; ROUNTREE, Nathan. **Learning and teaching programming: A review and discussion**. Computer science education, v. 13, n. 2, p. 137-172, 2003.

SAWYER, R. Keith (Ed.). **The Cambridge handbook of the learning sciences**. Cambridge University Press, 2005.

SBC. Referenciais de Formação para os Cursos de Graduação em Computação 2017 – Disponível em: <<https://www.sbc.org.br/educacao/diretrizes-para-ensino-de-computacao-na-educacao-basica>> . Acesso em: 21 out. 2019.

SCOTT, Michael James; GHINEA, Gheorghita. **Educating Programmers: A Reflection on Barriers to Deliberate Practice**. The Higher Education Academy. HEA STEM Conf. (2013)

SILVA, Rodrigo Ribeiro; FERNANDES, Juliana; SANTOS, Rodrigo. Panorama da Utilização de Jogos Digitais no Ensino de Programação no Nível Superior na Última Década: Uma Revisão Sistemática da Literatura. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2018. p. 535.

SMITH, Roger. The long history of gaming in military training. **Simulation & Gaming**, v. 41, n. 1, p. 6-19, 2010.

SOUZA, Draylson Micael, BATISTA, Marisa Helena da Silva, BARBOSA, Ellen Francine. **Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático**. Revista Brasileira de Informática na Educação, Volume 24, Número 1, 2016

TANG, Paul Kiet. What do our brains really want?. **The Lancet Neurology**, v. 17, n. 4, p. 308, 2018.

UNCTAD. Digital Economy Report 2019: Value creation and capture—Implications for developing countries. Disponível em: <https://unctad.org/en/PublicationsLibrary/der2019_overview_en.pdf> . Acesso em: 15 out. 2019.

VALENTE, José Armando; DE ALMEIDA, Maria Elizabeth Bianconcini; GERALDINI, Alexandra Fogli Serpa. Metodologias ativas: das concepções às práticas em distintos níveis de ensino. **Revista Diálogo Educacional**, v. 17, n. 52, p. 455-478, 2017.

VALENTE, José Armando. A sala de aula invertida e a possibilidade do ensino personalizado: uma experiência com a graduação em midialogia. **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática**. Porto Alegre: **Penso**, p. 26-44, 2018.

VAN ECK, Richard. Digital game-based learning: It's not just the digital natives who are restless. **EDUCAUSE review**, v. 41, n. 2, p. 16, 2006.

VERGARA, Sylvia Constant. Projetos e relatórios de pesquisa em administração. São Paulo: Atlas, 1998.

VIZCAÍNO, Aurora et al. An adaptive, collaborative environment to develop good habits in programming. In: **International Conference on Intelligent Tutoring Systems**. Springer, Berlin, Heidelberg, 2000. p. 262-271.

WILKINSON, Phil. A brief history of serious games. In: **Entertainment computing and serious games**. Springer, Cham, 2016. p. 17-41.

WING, Jeannette M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006.

APÊNDICE

APÊNDICE A - PRIMEIRO FORMULÁRIO DISCIPLINA WEB

Perguntas utilizadas no formulário "*Google Forms*".

1. Endereço de e-mail.
2. Qual sua faixa de idade?
 - abaixo de 18
 - 18 a 25
 - 26 a 35
 - 36 a 45
 - acima de 45
3. Você já possuía conhecimento de alguma linguagem de programação antes de entrar na faculdade?
 - Não tinha nenhum conhecimento
 - conhecia pessoas que já trabalhavam com programação
 - já tinha tentado aprender sozinho
 - fiz curso de programação
 - trabalhei/trabalho com programação
4. Você já participou e/ou concluiu outras disciplinas de programação na faculdade?
 - Não fiz nenhuma disciplina de programação
 - Iniciei mas não concluí
 - Iniciei e concluí (aprovado)
5. Assinale abaixo o seu nível de dificuldade na criação de algoritmos de modo geral. Sendo 1 (Pouca dificuldade) e 5 (Muita dificuldade).

(1) (2) (3) (4) (5)
6. Assinale abaixo o seu nível de dificuldade na criação de algoritmos que envolvem repetição (for, while). Sendo 1 (Pouca dificuldade) e 5 (Muita dificuldade).

(1) (2) (3) (4) (5)
7. Assinale abaixo o seu nível de dificuldade na criação de algoritmos que envolvem estruturas de decisão (if). Sendo 1 (Pouca dificuldade) e 5 (Muita dificuldade).

(1) (2) (3) (4) (5)
8. Assinale abaixo o seu nível de dificuldade na criação de algoritmos que envolvem listas e arrays. Sendo 1 (Pouca dificuldade) e 5 (Muita dificuldade).

(1) (2) (3) (4) (5)
9. Você utiliza com alguma frequência algum tipo de jogo digital?
 - raramente
 - 1 vez por mês
 - ocasionalmente
 - 2 a 4 vezes por semana
 - todo dia
10. Qual plataforma digital é a que mais você utiliza para jogar (mesmo que ocasionalmente)?
 - celular/smartphone
 - computador
 - tablet
 - console de videogame
 - outro dispositivo eletrônico

11. Você já se interessou em pesquisar como um jogo digital é feito?
- Nunca tive interesse
 - Já fiz um programa de um jogo simples (sem interface gráfica)
 - Já desenvolvi um jogo gráfico (2D/3D)
 - Já pesquisei trechos de códigos de jogos
 - Acompanho sites sobre desenvolvimento de jogos
12. Você acredita que conhecer programação é importante dentro da área de informática? Sendo 1 - Pouco importante e 5 – Muito importante.
- (1) (2) (3) (4) (5)
13. Você acredita que conhecer programação pode ser importante fora da área de informática? Sendo 1 - Pouco importante e 5 – Muito importante.
- (1) (2) (3) (4) (5)
14. Qual seu grau de interesse em seguir carreira na área de programação? Sendo 1 - Pouco interesse e 5 – Muito interesse.
- (1) (2) (3) (4) (5)

APÊNDICE B - SEGUNDO FORMULÁRIO DISCIPLINA WEB

Perguntas utilizadas no formulário "*Google Forms*".

1. Endereço de e-mail
2. O quanto você considera que o desenvolvimento de um jogo foi motivador? Sendo 1 - Pouco importante e 5 – Muito importante.
(1) (2) (3) (4) (5)
3. Qual o nível de dificuldade você teve na implementação do jogo proposto nas aulas? Sendo 1 - Fácil e 5 – Difícil.
(1) (2) (3) (4) (5)
4. Que tipo de dificuldade você teve no desenvolvimento deste jogo proposto em aula e que poderia ser melhorado em sua visão?
5. O que você poderia sugerir, em termos de exercícios voltados a programação, que motivasse você a se envolver mais com desenvolvimento de programas em javascript?

APÊNDICE C - QUESTIONÁRIO SEMIESTRUTURADO ALUNOS

ALUNOS

1. EXPERIÊNCIA EM INFORMÁTICA

Que tipo de experiência já tinha sobre a área de informática antes da faculdade ?

2. JOGOS

Você gosta de jogos digitais? Que tipo de jogos? Em qual plataforma você mais joga: celular/tablet, console, computador ?

3. EXPERIÊNCIA EM PROGRAMAÇÃO

Já tinha alguma experiência ou já havia lido sobre programação de computadores?
O quanto o desenvolvimento de um jogo digital na disciplina trouxe uma maior motivação para programar?

4. MOTIVAÇÃO

O que **motivou** a estudar na área de informática?

5. PERSPECTIVA NA INFORMÁTICA

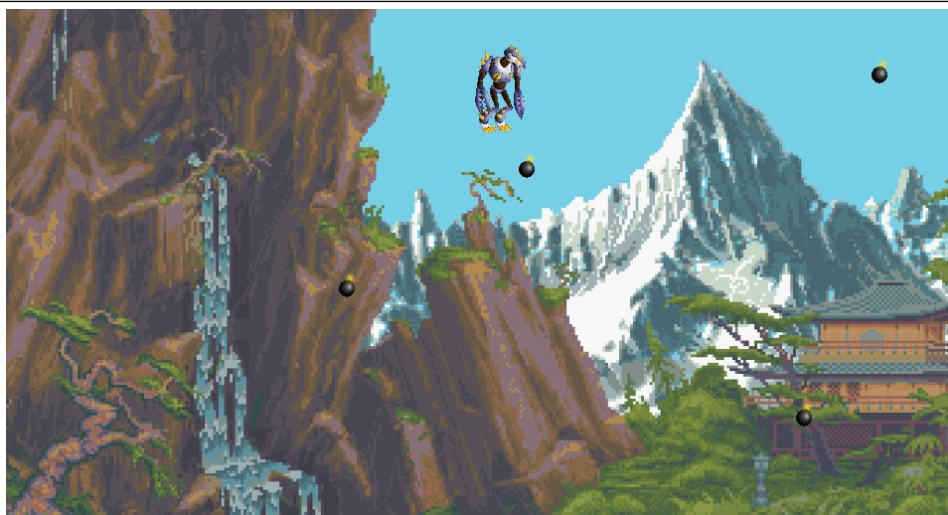
Você atualmente trabalha na área de informática? Em que tipo de função ?
Qual sua perspectiva sobre a área de informática? Pretende seguir carreira? Pretende se aprimorar sobre o assunto ?

APÊNDICE D - TELAS DOS JOGOS IMPLEMENTADOS COM USO DE PERSONAGENS E CENÁRIOS

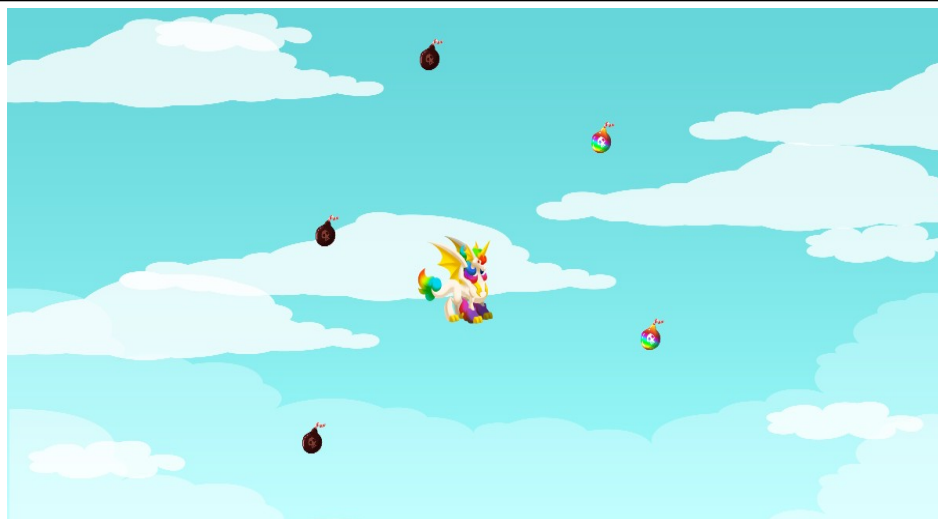
Quadro 19: Jogos com personagens e cenários



Personagem, figuras diferentes para cada um dos obstáculos e cenário de fundo.



Personagem , cenário de fundo e figura personalizada para os obstáculos



Personagem, dois tipos de figura para os obstáculos e cenário de fundo.

APÊNDICE E - CÓDIGOS *JAVASCRIPT* DE REFERÊNCIA

EXEMPLO DE MOVIMENTO

```
<html>
<head>
<style type="text/css">
DIV {
    position: absolute;
    color: yellow;
    top: 80;
}
#imagem {
    background: white;
    width: 80px;
    height: 50px;
}
img {
    max-width: 100%;
    height: auto;
}
</style>
</head>

<body>
<div id="imagem">
    '
</div>
<script language="JavaScript">
    var x = 0;

    function anda() {
        var imagem = document.getElementById("imagem");

        if (x >= 250)
            decrementa=1;

        if (x <= 0)
            decrementa=0;
```

```

        if (decrementa ==0)
            x++;
        else
            x--;

        imagem.style.left = x;
    }

    setInterval(function() { anda() }, 20);

</script>
</body>
</html>

```

EXEMPLO DE USO DO TECLADO

```

<html>
<head>
<style type="text/css">
DIV {
    position: absolute;
    top:80
}
</style>
</head>

<body onkeydown="anda(event)">
<h3>Usar setas!!!
<script language="JavaScript">
    var andar = 0;
    var altura = 80;
    function anda(tcl) {
        var imagem = document.getElementById("imagem");

        // HORIZONTAL
        if (tcl.keyCode == 39) {
            andar = andar + 10;
        }
    }

```

```
// EXECUTA
imagem.style.left  = andar;
imagem.style.top   = altura;

}
</script>
<div id="imagem">

</div>
</body>
</html>
```

ANEXOS

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Declaro, por meio deste termo, que concordei em ser entrevistado (a) e/ou participar na pesquisa de campo referente ao projeto/pesquisa intitulado (a) **PROPOSTA DE FRAMEWORK PARA USO DE JOGOS E PROJETO NO ENSINO DE PROGRAMAÇÃO DE COMPUTADORES NO ENSINO SUPERIOR** desenvolvida (o) por **YANKO COSTA**. Fui informado (a), ainda, de que a pesquisa é (coordenada / orientada) por DR LUCIANO FRONTINO DE MEDEIROS, a quem poderei contatar / consultar a qualquer momento que julgar necessário via telefone nº 55 41 2102 4987 ou e-mail LUCIANO.ME@UNINTER.COM.

Afirmo que aceitei participar por minha própria vontade, sem receber qualquer incentivo financeiro ou ter qualquer ônus e com a finalidade exclusiva de colaborar para o sucesso da pesquisa. Fui informado (a) dos objetivos estritamente acadêmicos do estudo, que, em linhas gerais é coletar informações sobre *motivação em programar computadores e o uso de jogos eletrônicos*.

Fui também esclarecido (a) de que os usos das informações por mim oferecidas estão submetidos às normas éticas destinadas à pesquisa envolvendo seres humanos, da Comissão Nacional de Ética em Pesquisa (CONEP) do Conselho Nacional de Saúde, do Ministério da Saúde, conforme a Resolução 466/2012.

Minha colaboração se fará de forma anônima, por meio de entrevista semi estruturada (a ser gravada a partir da assinatura desta autorização). O acesso e a análise dos dados coletados se farão apenas pelo (a) pesquisador (a) e/ou seu (s) orientador (es) / coordenador (es).

Fui ainda informado (a) de que posso me retirar desse (a) estudo / pesquisa / programa a qualquer momento, sem prejuízo para meu acompanhamento ou sofrer quaisquer sanções ou constrangimentos.

Atesto recebimento de uma cópia assinada deste Termo de Consentimento Livre e Esclarecido, conforme recomendações da Comissão Nacional de Ética em Pesquisa (CONEP).

___ de _____ de 2019.

Assinatura do (a) participante: _____

Assinatura do (a) pesquisador (a): _____

Assinatura do (a) testemunha (a): _____