

# PENSAMENTO COMPUTACIONAL E GAMIFICATION: RELATO DE UM EXPERIMENTO NA PLATAFORMA FURBOT

Joan Giancesini Tridapalli, Mauro Marcelo Mattos – Orientador

Curso de Bacharel em Ciência da Computação  
Departamento de Sistemas e Computação  
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

joangt001@gmail.com, mattos@furb.br

**Resumo:** A necessidade de fluência digital tem levado ao incremento de pesquisas em Pensamento Computacional (PC). O projeto Furbot é uma plataforma desenvolvida há 10 anos, inicialmente voltada para apoiar ações de ensino de programação Java em turmas de graduação dos cursos de Ciências da Computação e Sistemas de Informação da FURB. A partir de 2017, uma variação do Furbot foi desenvolvida com o intuito de desenvolver habilidades de pensamento computacional em turmas do ensino fundamental de uma escola pública. A partir desta experiência, o presente artigo introduz a nova versão do Furbot que está sendo desenvolvida em Unity e descreve o contexto do desenvolvimento de minijogos (objeto deste projeto) que foram desenvolvidos para serem integrados ao Furbot na transição entre as fases do mesmo. Foram caracterizadas as alterações realizadas e os resultados preliminares obtidos.

**Palavras-chave:** Furbot. Pensamento Computacional. Minijogos. Computação desplugada. Ensino fundamental.

## 1 INTRODUÇÃO

Dados sobre situação da educação no Brasil publicados em Pisa (2015) apresentou os seguintes indicadores: em um ranking de 70 países/economias: em matemática, 65º lugar, em leitura, 59º, lugar e em ciências, 63º lugar. Nesta mesma época, uma publicação do Instituto Nacional de Estudos e Pesquisas e Anísio Teixeira (2014) reforçou o problema indicando uma taxa de reprovação no ensino médio da ordem de 13% na rede pública e de 5,5% na rede particular de ensino. (PINTO; MATTOS, 2019, p.370)

Por outro lado, a necessidade de fluência digital em função da evolução tecnológica experimentada atualmente cada vez mais tem motivado uma série de pesquisas no que convencionou chamar de Pensamento Computacional (PC). O pensamento computacional introduz uma nova abordagem para a área da Ciência Cognitiva e da Ciência da Computação, pois parte da premissa de que a inserção dos conceitos da Ciência da Computação na educação básica desenvolve uma habilidade de abstração diferente, a qual ajuda as crianças na resolução de problemas em todas as áreas da vida.

O termo introduzido por Wing (2006) representa a noção de que intuitivamente as pessoas executam atividades diárias através de alguma forma de planejamento e quando treinadas para formalizar como realizaram as tarefas conseguem refletir como as estão realizado e planejar futuras ações de forma mais organizada. Brackmann (2017) afirma que esta é uma nova abordagem que parte da premissa de que a inserção dos conceitos da Ciência da Computação na educação básica possibilita o desenvolvimento de habilidades que podem ajudar as crianças na resolução de problemas em todas as áreas da vida, seja de maneira individual ou colaborativa

Em 2006, a Sociedade Brasileira de Computação (SBC) propôs que a Computação fosse inserida na grade do ensino básico até o ensino médio, incluindo as escolas públicas em que deve-se trabalhar, na educação infantil, habilidades voltadas à compreensão de problemas, identificação de sequência de passos, representação de passos de forma relacionada e organizada e sequenciamento de passos relacionados à movimentação de corpos e trajetos espaciais (SBC, 2006). Segundo Raabe et al. (2017), no ensino fundamental, do 1º ao 5º ano, deve-se trabalhar as habilidades relacionadas ao desenvolvimento de abstração de experiências concretas descritas através de dados e listas, identificação de estruturas condicionais e de repetição, uso de linguagens lúdicas para representação de algoritmos, compreensão sobre como decompor um problema para solucioná-lo em pedaços. A partir do 6º ano ao 9º ano, a recomendação é que seja introduzida uma linguagem de programação nativa para representação de dados e processos, além da formalização dos conceitos em estruturas de dados, uso de recursões e generalizações (RAABE et al., 2017). Ampliando o espectro de possibilidades, Soria e Rivero (2019, p. 335) afirmam que:

O pensamento computacional não deve se limitar à implementação de um assunto específico dentro do currículo, deve ser visto como um eixo transversal no currículo que contribui para a construção de uma cidadania na equidade que elimina as lacunas digitais e permite a resolução de problemas a partir de compreensão e raciocínio (SORIA; RIVERO, 2019, p335).

Inserido no contexto do ensino fundamental desde 2017, o projeto Furbot busca promover a inclusão digital cidadã por meio de oficinas de programação que permitam o desenvolvimento de aptidões em pensamento computacional com uso de uma ferramenta de aprendizagem desenvolvida na Universidade Regional de Blumenau (FURB). Várias observações vêm sendo realizadas durante as intervenções em sala de aula com crianças e um dos aspectos que tem chamado a atenção é o caráter puramente formativo que as versões de Furbot possuem. Assim sendo, como parte da estratégia de aprimorar o projeto e ampliar as possibilidades de utilização para além da sala de aula, deu-se início em 2019 um piloto que pretende atingir a dois objetivos:

- a) migrar a plataforma Furbot de Java para Unity: esta tem por objetivo ampliar o leque de plataformas de hardware para além dos desktops uma vez que a ferramenta Unity permite a geração de código em html5, Android e IOS (UNITY, 2019) enquanto a versão em Java está limitada ao ambiente desktop;
- b) inserir minijogos entre as mudanças de fase: esta etapa envolve o desenvolvimento de pequenos jogos que serão inseridos entre as mudanças de fase do Furbot, e tem por objetivo explorar outros aspectos de aprendizagem que podem ser trabalhados através de atividades lúdicas.

A segunda etapa, objeto do presente trabalho, envolveu:

- a) identificar temas adequados a faixa etária das turmas de 3º e 4º anos do ensino fundamental;
- b) planejar, juntamente com a equipe de projeto que desenvolve a migração, da interface de acoplamento dos jogos ao contexto da nova versão de Furbot;
- c) validar a solução produzida com especialista na área de educação. Esta atividade foi necessária tendo em vista manter o alinhamento com o propósito educacional do projeto e que a versão Furbot Unity só deverá ser liberada para uso nas oficinas na escola a partir de agosto de 2019 (portanto, posteriormente a conclusão deste trabalho).

Um dos aspectos observados pela equipe do projeto nas intervenções na escola nos anos de 2017 e 2018 foi a dificuldade de inserção dos professores da escola junto às atividades de Furbot. Em função disto, em 2019 uma ação específica vem sendo desenvolvida, no sentido de preparar os professores das turmas que receberão o Furbot a partir do segundo semestre deste ano. As teorias e metodologias previstas para esta pesquisa são: teoria de aprendizagem socio construtivista, Problem-Based Learning (PBL) aprendizagem baseada em problemas, e programação de jogos (ou programação com temática de games) e pesquisa-ação. A união delas tem ajudado a instigar, engajar e motivar os alunos, além de contribuir para suas aprendizagens.

## 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção é dividida em quatro assuntos que fundamentam o trabalho, sendo eles: fundamentos metodológicos na subseção 2.1, pensamento computacional (subseção 2.2), o projeto Furbot na subseção 2.3 (e sua versão em Unity em subseção 2.3.1), e trabalhos correlatos (subseção 2.4).

### 2.1 FUNDAMENTOS METODOLÓGICOS

A união das teorias e metodologias citadas anteriormente tem norteado as atividades no projeto Furbot e servem para instigar, engajar e motivar os alunos, além de contribuir para suas aprendizagens. Sem a intenção de explorar de modo aprofundado estas teorias e metodologias, seguem, nos próximos parágrafos, algumas definições importantes para a pesquisa.

Conforme Pinto e Mattos (2019, p.378) citando Bissolotti, Nogueira e Pereira (2014), o socioconstrutivismo é derivado de duas teorias, o Construtivismo, de Jean Piaget, e a Teoria de Aprendizagem, de Lev Vygotski.

Para Piaget, o indivíduo constrói ativamente o conhecimento, pela interação com o meio. Ele defende que as crianças devem ter a oportunidade de descobrir e inventar as coisas por conta própria, para então, entendê-las. Enquanto Vygotski propõe que, além da interação com o meio, a interação com os semelhantes também é essencial para a aprendizagem. Em sala de aula, o socioconstrutivismo se configura como uma teoria que considera o potencial do indivíduo na construção de seu próprio conhecimento, valorizando a sua interação com o ambiente e também com outros indivíduos, no caso, professores e colegas (Bissolotti, Nogueira e Pereira (2014 apud PINTO e MATTOS, 2019, p.378).

Dentre os trabalhos que utilizam dos conceitos de socioconstrutivismo encontra-se a abordagem da aprendizagem baseada em problemas (Problem Based Learning) sendo a estratégia instrucional está centrada no estudante, na qual há trabalho colaborativo para a resolução de problemas, o que se reflete em suas experiências e o instrutor desempenha o papel de mediador da aprendizagem (BURGUILLO, 2010). Uma estratégia que vem sendo utilizada para ensino de programação envolve a criação de jogos, conforme Prazeres e Oliveira (2018), Costa e Marchiori (2015), e Tolomei (2017) indicando que há um maior engajamento dos alunos com conseqüente aumento de interesse nos temas trabalhados. Para Pinto e Mattos (2019), “com essa abordagem, o aluno passa a ser um sujeito ativo na construção de seu conhecimento. Essa característica alude ao uso de ideias e aspectos pedagógicos construtivistas na utilização da programação de jogos”.

Os autores ainda complementam que no processo de programação para a construção dos jogos, os estudantes necessitam desempenhar algumas tarefas associadas à Ciência da Computação, para realizar suas funcionalidades. Neste sentido Kazimoglu et al. (2012) destacam:

- a) identificar e decompor o problema; definir padrões para tarefas repetitivas, na criação de uma solução algorítmica que atenda a todos os níveis do jogo, que pode se dar pela criação de funções para tarefas repetitivas;
- b) praticar o modo-debug, para verificar e detectar eventuais erros de lógica no jogo;
- c) praticar o modo run-time, para verificar movimentos durante o modo de execução;
- d) brainstorming, na atividade de examinar a estratégia de pensamento de outros jogadores, comparando a sua solução com a de outros jogos.

Alicerçado em teorias pedagógicas citadas anteriormente, o projeto Furbot vem sendo desenvolvido na forma de pesquisa-ação. Conforme dos Santos, Calíope e Barros Neto (2017), uma das definições mais difundidas de pesquisa-ação é: “[...]um tipo de pesquisa social com base empírica que é concebida e feita em estreita associação com uma ação ou com a resolução de um problema coletivo e no qual os pesquisadores e os participantes representativos da situação ou do problema estão envolvidos de modo cooperativo ou participativo” (apud THIOLENT, 2008, p.16). Tripp (2005, p. 5) afirma que “embora a pesquisa-ação tenda a ser pragmática, ela se distingue claramente da prática e, embora seja pesquisa, também se distingue claramente da pesquisa científica tradicional, principalmente porque a pesquisa-ação ao mesmo tempo altera o que está sendo pesquisado e é limitada pelo contexto e pela ética da prática”. De acordo com a equipe do projeto, em concreto, o projeto Furbot de uma forma mais geral tem o objetivo de estimular o desenvolvimento de saberes relacionados ao PC em estudantes e professores do Ensino Fundamental. Por outro lado, no presente projeto foi aplicado o método experimental para verificação do modelo. Segundo Gil (2008, p. 16), este método “[...] consiste essencialmente em submeter objetos de estudo à influência de certas variáveis, em condições controladas e conhecidas pelo investigador, para observar os resultados que a variável produz no objeto”.

## 2.2 PENSAMENTO COMPUTACIONAL

O termo Pensamento Computacional introduzido em Wing (2006) é um método para ensinar a solucionar problemas, conceber sistemas e compreender o comportamento humano inspirando-se em conceitos da Ciência da Computação ou, como colocado por Paula, Valente e Burn (2014), “é uma maneira específica de se pensar e de analisar uma situação ou um artefato, sendo independente do uso da tecnologia”. Assim, desenvolver habilidades em PC não significa necessariamente aprender a programar (ZANETTI; BORGES; RICARTE, 2016). Neste contexto, a lista de conhecimentos e habilidades necessárias para o pleno exercício da cidadania no século XXI é extensa, e inclui necessariamente o PC.

As três habilidades básicas que fundamentam o PC são: abstração, automação e análise. A habilidade de abstração está associada a capacidade de o indivíduo extrair apenas as características mais importantes de um problema para chegar a uma solução. A habilidade de automação está associada a capacidade de utilização de algum meio eletrônico que pode substituir o trabalho de um ser humano e a habilidade de análise está associada ao estudo dos resultados gerados pela automação (ANDRADE et al., 2013; VALENTE, 2016). Conforme Andrade et al. (2013) “em 2007 a Microsoft e a Carnegie Mellon University criaram o Centro de Pensamento Computacional e a Google tem se empenhado em promover esta metodologia em todo o currículo do ensino primário e secundário nos Estados Unidos”. E Brackmann (2017) complementa:

No momento que os estudantes aprendem a programar, estão também programando para aprender. Este aprendizado permite que eles aprendam muitas outras coisas e criem novas oportunidades de aprendizagem. Pode-se exemplificar isso, fazendo uma analogia entre a leitura e a escrita. Quando aprendemos a ler e escrever, criam-se novas oportunidades para que possamos aprender outras coisas. Quando aprendemos a ler, podemos então ler para aprender. O mesmo processo ocorre com a programação, ou seja, se aprendemos a programar, podemos programar para aprender ainda mais. Quando nos tornamos fluentes a ler e a escrever, não o fazemos para nos tornarmos escritores profissionais. Na realidade, poucas pessoas se tornam escritores profissionais, mas é útil para todos que aprendam a ler e escrever. Outro exemplo é o ensino da Matemática na escola, pois não se forma um matemático, mas um estudante que saiba fazer cálculos de uso rotineiro. (BRACKMANN, 2017, p. 20).

Conforme Soria e Rivero (2019) a elevação do pensamento computacional, como estratégia de aprendizagem no século XXI, remonta aos anos oitenta (século XX), no qual o desenvolvimento da aprendizagem de geometria, foi a experiência mais próxima do tema, a partir da linguagem de programação chamada "Logo". Conforme Valverde, Fernández e Garrido (2015, p. 4, tradução nossa), “nesse esquema dos anos 80, as ideias de Dewey, Piaget e Vygotsky convergiram, materializadas por Seymour Papert e geraram uma proposta disruptiva nessa década”. Essas habilidades, de acordo com Vilanova (2018) são baseadas em uma série de atitudes que são dimensões essenciais do pensamento computacional. Essas atitudes incluem:

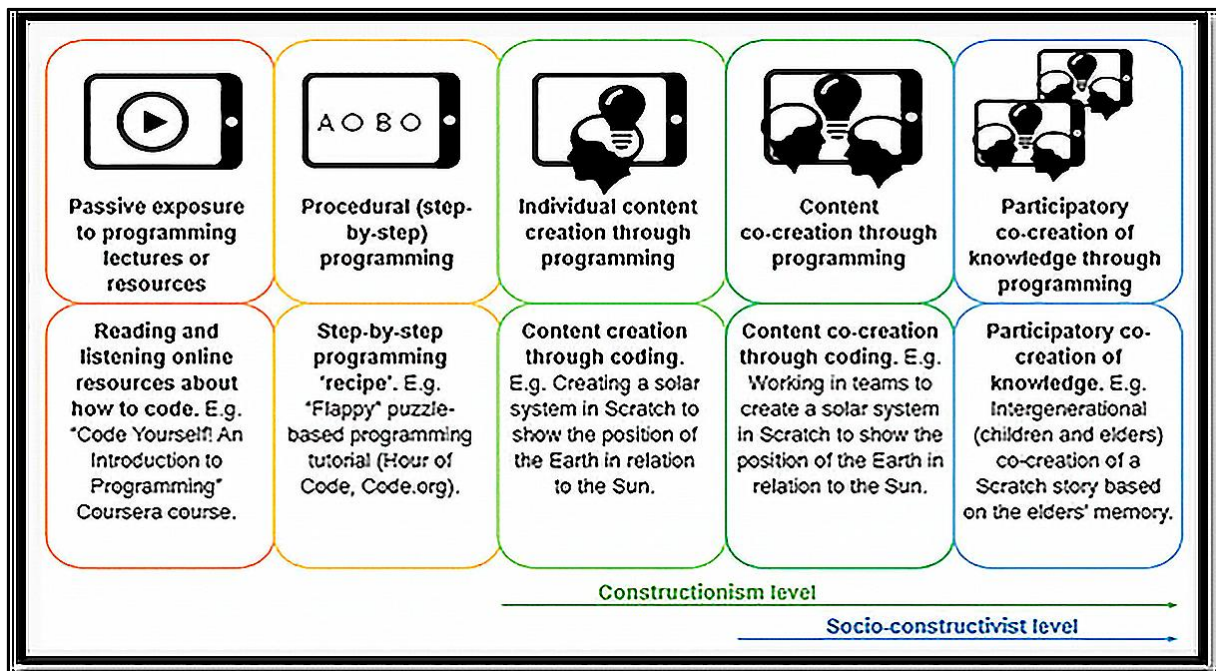
- a) confiança em si para a gestão da complexidade;

- b) perseverança em enfrentar problemas difíceis;
- c) tolerância diante de situações ambíguas;
- d) capacidade de combater problemas não estruturados.

Romero, Lepage e Lille (2017, p. 3) (Figura 1), relacionam cinco níveis de participação criativa na programação o que evidencia que esta prática requer uma participação ativa do aluno e de seus pares, o que os envolve no processo de design e desenvolvimento da co-construção do conhecimento. Os níveis são:

- a) exposição passiva a explicações centradas no professor;
- b) procedimentos passo a passo, com suporte de tutoriais de programação;
- c) criação de conteúdo original através de programação individual;
- d) programação de equipe e finalmente;
- e) co-criação participativa de conhecimento por meio de programação.

Figura 1 - Os cinco níveis de participação criativa em atividades de programação criativa



Fonte: Romero, Lepage e Lille (2017, p. 3).

Em 2016, Zanetti, Borges e Ricarte (2016) realizaram uma revisão sistemática da literatura envolvendo o termo "Pensamento Computacional" no Brasil, no período de 2007 a 2015 e constataram que os primeiros trabalhos sobre este tema no universo dos veículos pesquisados, surgiram somente a partir de 2012. O trabalho é relevante porque procurou responder a seguinte pergunta de pesquisa: "O que se tem feito nas pesquisas realizadas no Brasil sobre Pensamento Computacional no contexto de ensino de programação" (ZANETTI; BORGES; RICARTE, 2016, p. 3). Foram encontrados 16 trabalhos neste período. Relativamente ao nível de escolaridade, o trabalho identificou atividades envolvendo os três níveis: ensino fundamental (6 trabalhos), ensino médio/técnico (10 trabalhos) e ensino superior (1 trabalho), (ZANETTI; BORGES; RICARTE, 2016). Os autores concluíram que o maior número de trabalhos destinados ao ensino médio/técnico decorre do fato de que alunos com este nível de escolaridade apresentam uma capacidade de pensamento e abstração que possibilita uma melhor assimilação de conceitos de programação e maturidade para desenvolver habilidades fundamentais para o PC, como abstração, decomposição de problemas e análise de dados. No escopo da análise foram identificadas cinco categorias, quais sejam:

- a) computação desplugada (CD): envolve métodos que procuram promover o ensino de computação sem o uso de computador, utilizando atividades lúdicas para atingir pessoas de todas as idades;
- b) jogos digitais (JD): envolve a utilização de jogos digitais para o ensino de conceitos de computação ou programação de computadores;
- c) linguagem de programação (LP): utilizam linguagens de programação de alto nível (paradigma imperativo: Pascal ou C; orientado a objetos: Java ou Python);
- d) linguagem de programação visual (LPV): envolve a utilização de linguagens de programação visual, como Scratch ou App Inventor;
- e) robótica pedagógica (RP): envolve a utilização de artefatos robóticos, reais ou virtuais, como ferramenta para o ensino de programação.

O trabalho também compilou o conjunto de habilidades trabalhadas em cada artigo, o que resultou no seguinte conjunto:

- a) coleção de dados: processo de recolher as informações adequadas;
- b) análise de dados: processo de encontrar o sentido dos dados, encontrar padrões e tirar conclusões;
- c) representação de dados: representação e organização de dados em gráficos adequados, cartas, palavras ou imagens;
- d) decomposição do problema: quebrar tarefas em partes menores e gerenciáveis;
- e) abstração: reduzir a complexidade para definir a ideia principal;
- f) algoritmos e procedimentos: série de passos ordenados e encadeados para resolver um problema ou atingir algum fim;
- g) automação: usar computadores ou máquinas para fazer tarefas repetitivas ou tediosas;
- h) simulação: representação ou modelo de um processo;
- i) paralelização: organizar recursos para, simultaneamente, realizar tarefas para alcançar um objetivo comum.

Já Bordini et al. (2016) realizaram uma análise sobre os esforços realizados entre 2010 e 2015 que têm sido feitos no sentido de disseminar e introduzir o PC no Brasil objetivando obter um panorama geral dos desdobramentos na área. O trabalho registrou que as principais iniciativas estão focadas no Ensino Fundamental e utilizam as ferramentas Scratch e práticas de computação desplugada. Ações relatadas em Bezerra e Dias (2014), Barcelos e Silveira (2012), Costa et al. (2012) e Fincher et al. (2010) têm sido realizadas, destacando as iniciativas de introdução de conceitos de programação na Educação Básica em função do seu caráter transversal às demais áreas do conhecimento (CSTA, 2011; FRANÇA et al., 2015). Contudo, no Brasil, via de regra, o ensino de tal habilidade não integra o currículo escolar, o que amplia os desafios a serem superados para a sua adoção no ensino fundamental (FRANÇA; TEDESCO, 2015).

### 2.3 O PROJETO FURBOT

Levando em consideração o atual cenário da Educação Básica no país onde há escassez de recursos de toda a ordem, ensinar habilidades computacionais pode, portanto, configurar-se como um desafio. Ao mesmo tempo, apresenta-se um cenário repleto de oportunidades aos educadores, pesquisadores e à comunidade escolar. É neste contexto que se introduz o projeto Furbot, conforme Mattos et al. (2018b, p. 2),

Uma das maneiras de motivar estudantes é o uso da técnica de gamification, ou seja, a adoção de uma estratégia de interação com base na oferta de estímulos que favoreçam o engajamento de uma maneira lúdica. Dentre várias plataformas existentes para facilitar o aprendizado de lógica de programação, a plataforma desenvolvida foi concebida para minimizar as dificuldades de aprendizagem e ensino na lógica de programação por meio de um forte apelo à área de jogos, criando assim uma atmosfera facilitadora ao aprendizado. O elemento central do ambiente é a programação de um robô que vive em um mundo bidimensional junto de outros tipos de objetos, os quais também podem ser programados. Sobre esse mundo, o aluno desenvolve atividades de movimentação em quatro direções e também de detecção de obstáculos. Esses exercícios evoluem em grau de dificuldade conforme a evolução do estudante, a ponto de possibilitar o desenvolvimento de um jogo completo.

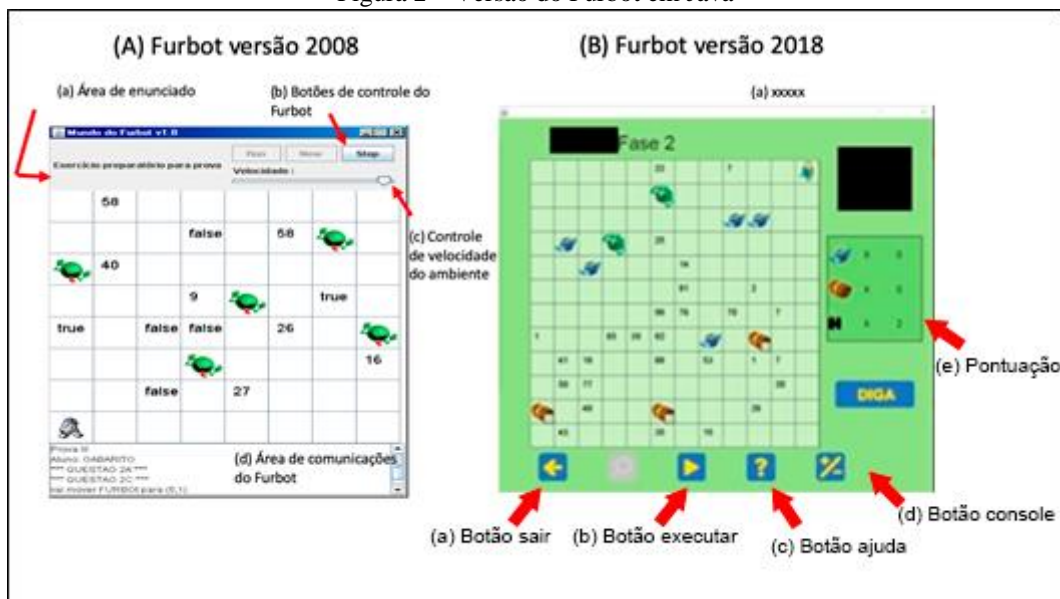
Uma das maneiras de motivar estudantes é o uso da técnica de jogos digitais (ou *gamification*), ou seja, a adoção de uma estratégia de interação com base na oferta de estímulos que favoreçam o engajamento de uma maneira lúdica, dentre vários frameworks existentes para facilitar o aprendizado de lógica de programação, o Furbot (VAHLICK; MATTOS, 2008; ARAÚJO; SCHLOGL; MATTOS, 2017; MATTOS et al., 2018) caracteriza-se por ser um micro *engine* de jogos. Esse software foi concebido para minimizar as dificuldades de aprendizagem e ensino na lógica de programação por meio de um forte apelo à área de jogos, criando assim uma atmosfera facilitadora ao aprendizado (MATTOS; VAHLICK; HUGO, 2008). A proposta do Furbot surgiu a partir das reflexões dos professores Mauro Marcelo Mattos, Adilson Vahldick e Carlos Henrique Correia sobre como minimizar a dificuldade que os acadêmicos dos cursos de Ciência da Computação e Sistemas de Informação da Universidade Regional de Blumenau (FURB) vinham enfrentando na disciplina de programação de computadores em 2007 e anos anteriores.

A plataforma atual é desenvolvida utilizando como base um *framework* Java existente há aproximadamente 10 anos, foi concebida para apoiar as atividades de ensino-aprendizagem da disciplina de Introdução a Programação nos cursos de graduação em Ciência da Computação e Sistemas de Informação da FURB e já foi utilizado por mais de 500 alunos. A partir de 2017 foi adaptada para ser utilizada no desenvolvimento de habilidades de Pensamento Computacional em crianças de 3o e 4o anos de uma escola pública do município de Blumenau (MATTOS et al., 2018a).

O elemento central do Furbot é a programação de um robô que vive em um mundo bidimensional junto de outros tipos de objetos, os quais também podem ser programados. Sobre esse mundo, o aluno desenvolve atividades de movimentação em quatro direções e de detecção de obstáculos. Esses exercícios evoluem em grau de dificuldade conforme a evolução do estudante, a ponto de possibilitar o desenvolvimento de um jogo completo. Como resultado do desenvolvimento, a Figura 2 apresenta as duas versões do Furbot escritas em Java. A Figura 2A contempla a versão

original desenvolvida em 2008 e usada na graduação, enquanto a Figura 2B apresenta a versão utilizada no ensino fundamental a partir de 2017.

Figura 2 – Versão do Furbot em Java



Fonte: Elaborado pelo autor.

Além disso, a versão Furbot 2018 possui um gerador de mundo (Figura 3A). O gerador de mundo permite o jogador criar a própria fase ao selecionar o objeto na barra lateral direita e arrastá-lo para a área no centro. Para realizar a movimentação do Furbot pelo mapa, utiliza-se o console de comandos (Figura 3B), que permite a inserção dos comandos para que o personagem realize ações dentro da plataforma.

Figura 3 – Gerador de mundo (A) e console de comandos (B)



Fonte: Mattos et al. (2018a).

Para complementar o pacote de recursos que o projeto Furbot possui, há também dois componentes que permitem o exercício de computação desplugada (Figura 4), quais sejam:

- um jogo de tabuleiro, em que o aluno é apresentado ao conjunto de comandos que serão utilizados na versão digital (Figura 4A);
- um jogo de tabuleiro “vivo” em que as crianças iniciam brincando com regras originais do Furbot, mas a medida em que a brincadeira fica interessante, lhes é permitido incrementar ou alterar as regras criando seus próprios jogos (Figura 4B).

A partir deste conjunto de recursos, foi utilizada a classificação de Zanetti, Borges e Ricarte (2016) para enquadrar o Furbot:

- computação desplugada (CD): Figura 4;
- jogos digitais (JD): Figura 3;
- linguagem de programação (LP): Figura 4A;
- linguagem de programação visual (LPV): Figura 4B;
- robótica pedagógica (RP): Figura 4B;

Além disso, segundo a equipe do projeto, o Furbot permite exercitar o seguinte conjunto de habilidades:



- colecção de dados: no trajeto, o robozinho precisa coletar itens no mundo (tabuleiro) e interagir com estes itens;
- análise de dados: processo de encontrar o sentido dos dados, encontrar padrões e tirar conclusões;
- representação de dados: a interface de comunicação do Furbot permite a apresentação de resultados da interação do robozinho com o tabuleiro;
- decomposição do problema: quebrar tarefas em partes menores e gerenciáveis;
- abstracção: reduzir a complexidade para definir a ideia principal;
- algoritmos e procedimentos: série de passos ordenados e encadeados para resolver um problema ou atingir algum fim.

Figura 4 – Furbot Desplugado



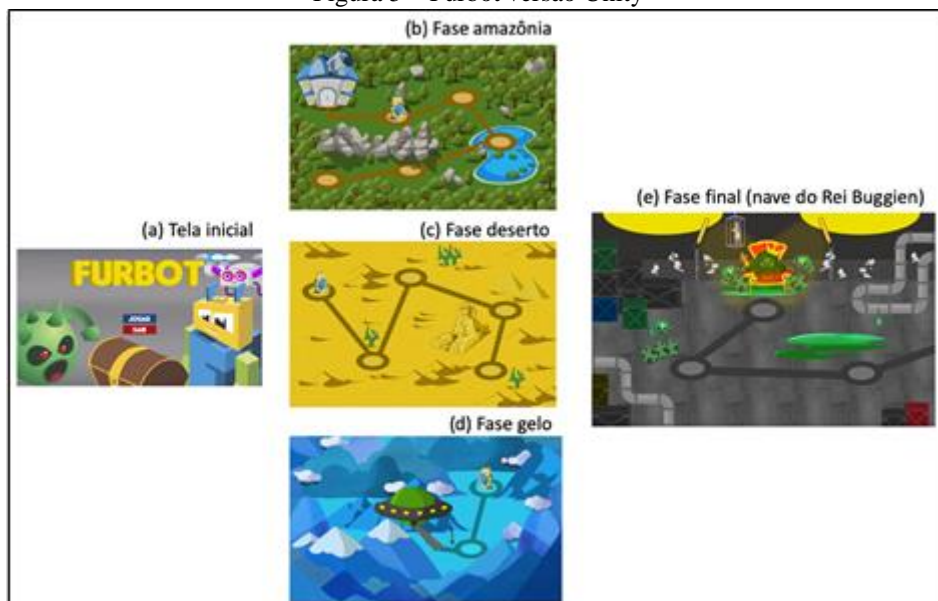
Fonte: Mattos et al. (2018a).

Segundo a equipe do projeto, um dos aspectos a destacar no contexto do projeto Furbot é que ele converge com a Base Nacional Comum Curricular (BNCC), que é um documento norteador aos currículos dos sistemas e redes de ensino, especificamente no que tange ao item 5 das competências gerais, que prevê o uso das tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais.

### 2.3.1 FURBOT VERSÃO UNITY

A concepção da versão em Unity envolveu um completo redesenho da arquitetura do projeto introduzindo-se um apelo mais forte à gamificação (MATTOS; KOHLER, 2019). A Figura 5 apresenta a tela inicial do Furbot (a) e uma sequência de fases envolvendo aspectos da Amazônia (b), do deserto (c), do gelo (d) e a fase final em que há a disputa com um personagem chamado de “Rei Buggien” (e).

Figura 5 – Furbot versão Unity

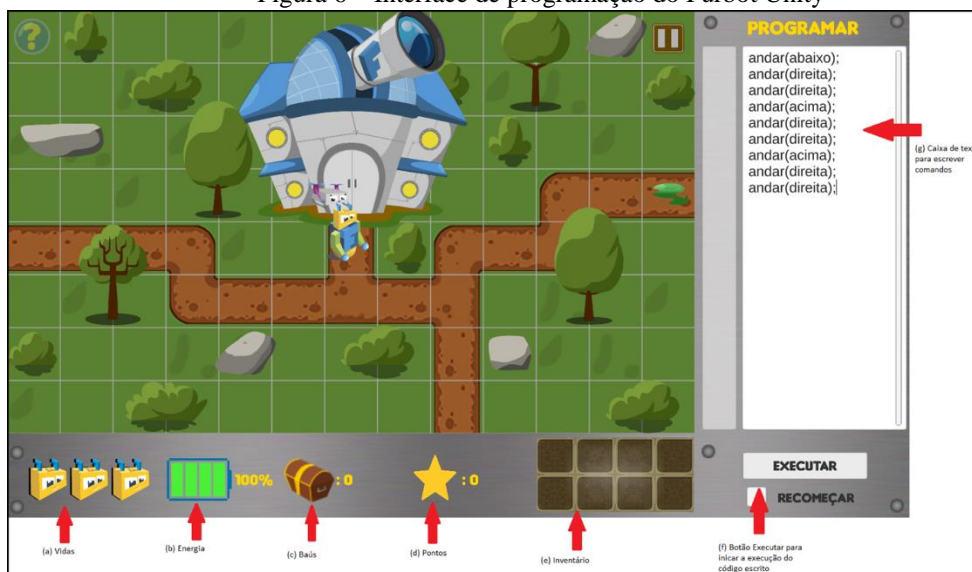


Fonte: Elaborado pelo autor.

Nesta versão, para viabilizar a movimentação do Furbot, é necessário construir uma sequência lógica de passos. A Figura 6 apresenta a interface, na qual o jogador pode introduzir (e editar) a sequência de comandos que serão

executados pelo Furbot. Na base da tela, é possível controlar o número de vidas, a quantidade de energia (o Furbot perde energia quando bate em obstáculos e a programação precisa ser alterada para capturar-se energia suficiente para concluir uma fase), o número de tesouros obtidos, vidas, e outros elementos pertencentes à história do jogo.

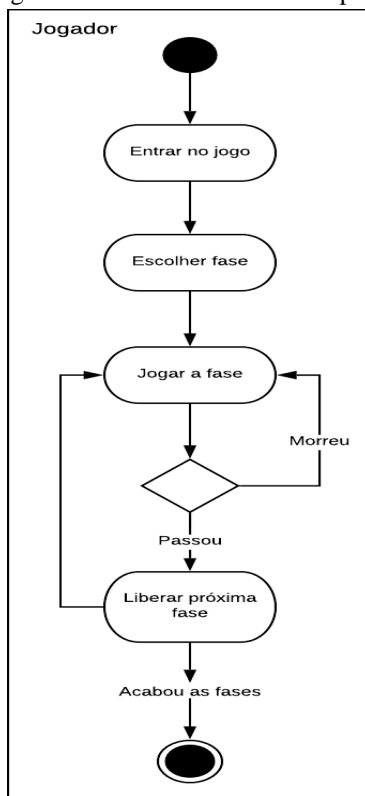
Figura 6 – Interface de programação do Furbot Unity



Fonte: elaborado pelo autor.

As fases do jogo são acessadas por meio do mapa principal (Figura 5b), onde qual o jogador pode escolher a fase que deseja jogar. O fluxo lógico principal de funcionamento do Furbot é apresentado no diagrama de atividades da Figura 7, onde o jogador ao entrar no jogo, se depara com o mapa para selecionar as fases, a ao terminar uma delas, terá a opção de reiniciá-la ou voltar ao mapa para ir à próxima.

Figura 7 – Diagrama de Atividades das fases padrão do Furbot



Fonte: elaborado pelo autor.

## 2.4 TRABALHOS CORRELATOS

São apresentados nessa seção os trabalhos correlatos ao projeto desenvolvido. São eles: Uma ambientação de jovens à jogos com o intuito de ensinar lógica computacional (MATTOS et al., 2018a), conforme o Quadro 1; o FURBOT-



WEB, tendo como objetivo construir uma plataforma web para ensino adaptativo de programação com base no Furbot (KOPSCH, 2016), como mostra o Quadro 2; e o FURBOT-C, com a finalidade de permitir que o robô possa ser movimentado com o teclado e seus movimentos registrados e exibidos em forma do pseudocódigo educacional (MAUS, 2011), sendo apresentado no Quadro 3.

Quadro 1 – Uma pesquisa-ação sobre o desenvolvimento do pensamento computacional com crianças

Referência	Mattos et al. (2018a)
Objetivos	Conduzir as crianças para um ambiente de jogos de computador no qual a coloca em contato com conceitos básicos de programação.
Principais funcionalidades	O usuário deve movimentar o robô por meio do uso dos recursos de programação em Java para implementar a lógica do método inteligência do robô que o robô chegue até o Tesouro. Para que isso aconteça, é preciso utilizar comandos Java.
Ferramentas de desenvolvimento	Eclipse.
Resultados e conclusões	Foi possível evidenciar uma evolução progressiva na capacidade cognitiva e no raciocínio lógico para a construção de algoritmos, solução de problemas e maior entendimento de lógica computacional.

Fonte: elaborado pelo autor.

Quadro 2 – FURBOT-WEB

Referência	Kopsch (2016)
Objetivos	Criar uma plataforma web para ensinar programação com blocos com base no Furbot.
Principais funcionalidades	O FURBOT-WEB é uma plataforma web de ensino personalizado de programação com blocos. Tendo base o <i>framework</i> Furbot, ele utiliza o sistema de recomendações para filtrar o conteúdo e permitir que cada aluno aprenda no próprio ritmo, automatizando a aplicação dos exercícios a ser executados.
Ferramentas de desenvolvimento	SQL Server 2014, ASP.NET MVC 5, TecEdu, Visual Studio 2015, Team Foundation Server.
Resultados e conclusões	O FURBOT-WEB atingiu os objetivos propostos que foram os de disponibilizar o Furbot em uma plataforma web de forma adaptativa com o sistema de recomendações e analisar a usabilidade da experiência do usuário. Mostrando que o sistema é amigável aos usuários mais novos, sendo mais prático para ensinar lógica de programação.

Fonte: elaborado pelo autor.

Quadro 3 – FURBOT-C

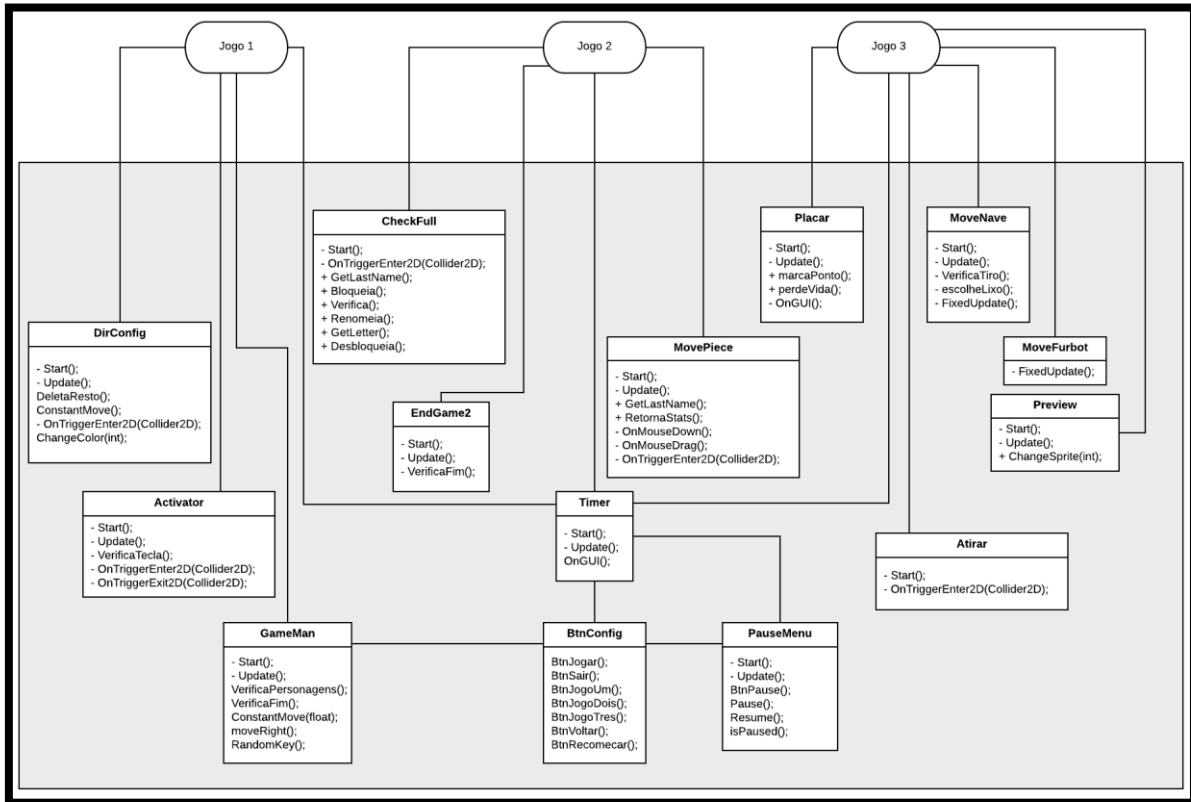
Referência	Maus (2011)
Objetivos	Permitir que o aluno possa controlar o robô pelo teclado e mouse, registrando os movimentos utilizados, para auxiliar na construção do raciocínio lógico e aumentar a jogabilidade.
Principais funcionalidades	O Furbot-C que permite que o aluno possa controlar o robô pelo teclado e mouse para caminhar, empurrar objetos e digitar algo para o robô falar. Já em <i>background</i> a ferramenta vai registrando a sequência de comandos de tal forma a apresentar ao estudante o algoritmo que produz o mesmo comportamento.
Ferramentas de desenvolvimento	NetBeans 7.0.1.
Resultados e conclusões	Por meio desta extensão, introduziu-se a possibilidade de que um estudante que tenha alguma dificuldade na resolução de algum desafio, possa experimentar conduzir o personagem fazendo uso de movimentos de mouse e teclado, bem como analisar como o algoritmo de solução poderia ser construído.

Fonte: elaborado pelo autor.

### 3 DESCRIÇÃO DOS MINIJOGOS DESENVOLVIDOS

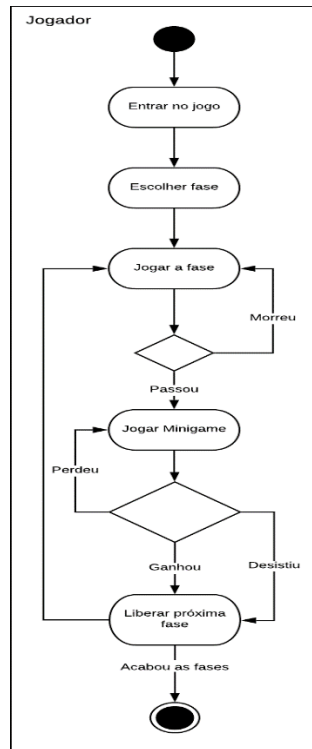
Para criação do projeto, utilizou-se uma versão do Furbot que se encontra em desenvolvimento, e está sendo produzida em Unity, facilitando a disponibilidade tanto na plataforma web como mobile. Com o intuito de inserir um contexto cujo nível de gamificação é mais acentuado, a estratégia utilizada foi o desenvolvimento de pequenos jogos visando levar a criança a brincar sem a preocupação de desenvolver uma estratégia relacionada ao FURBOT. Para isto foram desenvolvidos três jogos, os quais foram construídos de acordo com a especificação (Figura 8), no qual mostra os objetos do jogo e suas conexões, e foram inseridos ao após cada fase padrão, conforme o diagrama de atividades na 9, no qual mostra que agora, após cada fase e programação do Furbot, é inserido um minijogo para a criança jogar.

Figura 8 – Diagrama de Objetos dos jogos



Fonte: Elaborado pelo autor.

Figura 9 – Diagrama de Atividades com os jogos inseridos



Fonte: elaborado pelo autor.

### 3.1 JOGO 1

O primeiro jogo foi baseado na brincadeira de cabo de guerra, que consiste em dois times segurando um cabo, e o puxando e direções opostas, sendo que o time que fizer o outro cruzar a linha entre eles primeiro vence. Este primeiro jogo utiliza como protagonistas de um lado o personagem Furbot e um drone chamado S223 (em alusão ao número da

sala onde o Furbot está sendo desenvolvido) e, de outro, os antagonistas Buggien (personagem original do Furbot) e o Rei Buggien (personagem criado para esta fase da versão Unity).

Para fazer com que os protagonistas puxem o cabo, quadrados com setas passam na parte de cima da tela, e o jogador deve apertar o botão direcional do teclado correspondente quando o quadrado passar na marcação (Figura 10). A intenção neste jogo foi desenvolver habilidades relacionadas ao reflexo já que a velocidade das setas aumenta a medida em que o jogo se desenrola.

Figura 10 – Jogo um: cabo de guerra



Fonte: elaborado pelo autor.

Os personagens exercem força automaticamente, e caso algum deles caia no buraco central, o lado oposto ganha mais força. Os quadrados são gerados aleatoriamente, e são instanciados de acordo com a posição da marcação, facilitando em caso de reposicionamento de acordo com o formato da tela da plataforma, como mostrado no Quadro 4.

Quadro 4 – Função que instância os quadrados na tela

```

86 //1 - Esquerda / 2 - Direita / 3 - Acima / 4 - Abaixo
87 private void RandomKey()
88 {
89     int key = UnityEngine.Random.Range(1,4);
90     switch (key)
91     {
92         case 1:
93             UnityEngine.Object.Instantiate(setaEsq, pos, Quaternion.identity);
94             break;
95         case 2:
96             UnityEngine.Object.Instantiate(setaDir, pos, Quaternion.identity);
97             break;
98         case 3:
99             UnityEngine.Object.Instantiate(setaCima, pos, Quaternion.identity);
100            break;
101         case 4:
102             UnityEngine.Object.Instantiate(setaBaixo, pos, Quaternion.identity);
103            break;
104     }
105 }
106

```

Fonte: elaborado pelo autor.

### 3.2 JOGO 2

O segundo jogo (Figura 11) tem como objetivo criar um caça-palavras onde cada resposta deve ser montada como um quebra-cabeças. No protótipo foram selecionadas dez perguntas, cujas respostas são as peças. A escolha das perguntas foi realizada sob orientação de Wuo (2019) e utiliza a Base Nacional Comum Curricular (BNCC, 2019).

Figura 11 – Jogo dois: caça palavras



Fonte: elaborado pelo autor.

A principal funcionalidade do jogo está relacionada a movimentação das letras para montar a palavra alvo a qual é sorteada do conjunto de palavras disponíveis. Estas letras devem ser conectadas com os quadrados cujo tamanho corresponde ao tamanho da palavra alvo (Quadro 5). O encaixe é realizado a partir da colisão das letras com os quadros.

Quadro 5 - Função de colisão da peça com o quadrado

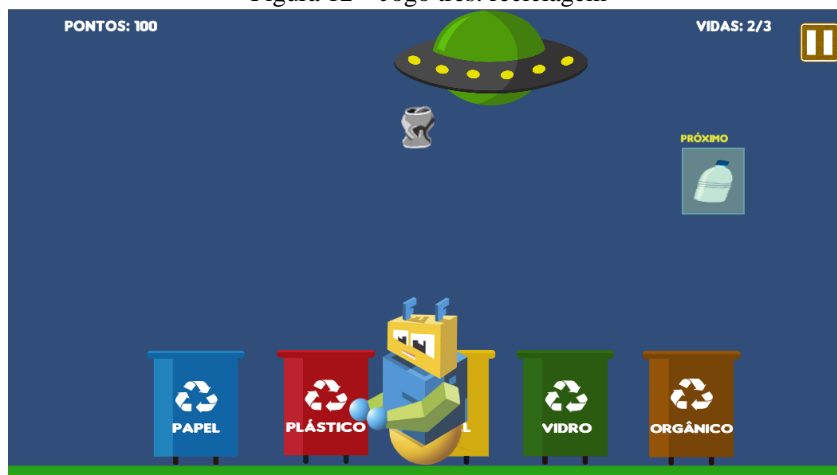
```
private void OnTriggerEnter2D(Collider2D col)
{
    if((col.gameObject.name == gameObject.name) && col.gameObject.tag == "socket")
    {
        transform.position = col.transform.position;
        pieceStats = "locked";
        Instantiate(edgePart, col.gameObject.transform.position, edgePart.rotation);
        col.gameObject.name = ""+i;
        gameObject.name = ""+i;
        i++;
    }
}
```

Fonte: elaborado pelo autor.

### 3.3 JOGO 3

O terceiro jogo busca trabalhar o aspecto de conscientização das crianças sobre a importância da reciclagem. Neste jogo o vilão, o Rei Buggien, está em sua espaçonave jogando lixo na direção do Furbot, que precisa ficar em frente a lata de lixo correspondente para ganhar pontos (Figura 12). Se o aluno errar três vezes o jogo encerra.

Figura 12 – Jogo três: reciclagem



Fonte: elaborado pelo autor.

A movimentação do Furbot ocorre com base no eixo horizontal da tela, sendo que no computador as teclas correspondentes a movimentação são as setas esquerda e direita, e as teclas A e D, e na plataforma mobile, corresponde a apertar do lado esquerdo ou direito da tela. A geração do lixo que é atirado pela espaçonave é gerada de forma aleatória (Quadro 6, linhas 46-69), sendo feita a detecção dos objetos com base no sistema de colisão da Unity. A interface contém

um indicador de qual será o próximo objeto que será lançado. Após lançado, se o lixo acertar a lata correta, o jogador ganha pontos, e caso acerte a lata errada, o jogador perde uma vida.

Quadro 6 – Código que instancia os objetos aleatoriamente.

```

36 private void VerificaTiro()
37 {
38     if(Time.time > atirarProx)
39     {
40         escolheLixo();
41     }
42     atirarProx = Time.time + atirarFreq;
43 }
44
45
46 private void escolheLixo()
47 {
48     int prox = UnityEngine.Random.Range(1, 5);
49     proxItem.ChangeSprite(prox);
50     switch (atual)
51     {
52     case 1:
53         Instantiate(metal, transform.position, Quaternion.identity);
54         break;
55     case 2:
56         Instantiate(papel, transform.position, Quaternion.identity);
57         break;
58     case 3:
59         Instantiate(plastico, transform.position, Quaternion.identity);
60         break;
61     case 4:
62         Instantiate(organico, transform.position, Quaternion.identity);
63         break;
64     case 5:
65         Instantiate(vidro, transform.position, Quaternion.identity);
66         break;
67     }
68     atual = prox;
69 }

```

Fonte: elaborado pelo autor.

#### 4 RESULTADOS

Conforme apresentado, foram desenvolvidos três minijogos os quais foram inseridos na estrutura principal da versão Unity do Furbot. Um desafio presente esteve associado ao fato de que a reescrita do Furbot em Unity está em andamento e em alguns momentos isto demandou a readequação de códigos já desenvolvidos para alinhar o funcionamento dos minijogos à nova versão da estrutura principal. Outro problema detectado ocorreu quando foram utilizadas versões diferentes do ambiente de desenvolvimento Unity. Neste caso, algumas funcionalidades pararam de funcionar devido à falta de retrocompatibilidade entre as versões, e tiveram que ser revisadas.

Analisando-se os trabalhos correlatos (Quadro 7), pode-se avaliar que a versão atual incorpora as características das versões anteriores escritas em Java, mas introduz um aspecto relevante que é o reforço em *gamification*. O Furbot versão Unity não perdeu em recursos, mas está ganhando em potencial de atratividade para as crianças. Além disso, a ferramenta utilizada no desenvolvimento possibilita a geração de código para múltiplas plataformas de tal forma que o Furbot poderá, num futuro próximo, ser liberado para ser executado em dispositivos móveis compatíveis com Android e IOS, e para a plataforma Web.

Quadro 7 – Comparativo entre os trabalhos correlatos

Correlatos	Mattos et al. (2018)	Maus (2011)	Kopsch (2016)	Jogos Unity
Características				
Plataforma Web			X	X
Plataforma Android				X
Ferramentas	Eclipse	NetBeans 7.0.1	SQL Server 2014, TecEdu, TFS	Unity 2018.2.1
Linguagem	Java	Java	C#, JSON, JS e HTML 5	C#

Fonte: Elaborado pelo autor.

Como a plataforma não está totalmente concluída o processo de validação final em sala de aula não foi realizado no período deste trabalho. Contudo, foram realizadas duas oficinas, com crianças na faixa etária de terceiro e quarto ano, utilizando os jogos desenvolvidos: a primeira delas na feira do evento Santa Catarina (SC) Criativa ocorrido no Teatro Carlos Gomes em 22 de maio de 2019, e a segunda na Associação Criança em Primeiro Lugar no dia 04 de junho de 2019, com a participação de oito crianças. Em ambas as situações, os resultados preliminares foram positivos pois as crianças “jogaram” o Furbot e ficaram entusiasmadas com os “joguinhos”, pois “havia mais de uma forma de poder jogar dentro da plataforma Furbot além das fases normais de programação”, o que as incentivava a continuar jogando as fases com mais frequência.



## 5 CONCLUSÕES

O Furbot é um projeto que possui um grande potencial para permitir o desenvolvimento de habilidades em Pensamento Computacional. O esforço realizado durante aproximadamente 10 anos permitiu a equipe do projeto acumular um agregado de experiências que estão sendo totalizadas na atual versão do projeto. A experiência do no desenvolvimento do presente trabalho em colaborar para a consecução dos objetivos maiores do projeto foi muito interessante, particularmente pela interação que há entre as equipes do curso de Comunicação Social, por meio do laboratório Republika e do Laboratório de Desenvolvimento e Transferência de Tecnologia, onde o Furbot está sendo construído.

Por meio da pesquisa sobre desenvolvimento de jogos, foi possível identificar que características os jogos precisavam ter para abranger a faixa etária correta, juntamente com a pesquisa sobre Pensamento Computacional, para que os objetivos dos jogos possam ao mesmo tempo entreter e educar as crianças. A expectativa, a partir do início das oficinas na escola, o que deve ocorrer a partir de agosto de 2019, é que esta versão amplifique os resultados obtidos até o momento. A experiência na migração da versão em Java para Unity também possibilitou a experimentação de uma nova tecnologia, agregando conhecimentos além daqueles trabalhados durante a graduação.

Como este foi um projeto piloto, novas versões deverão ser desenvolvidas. Assim sugere-se:

- a) criação de novos cenários, animações e objetos das fases;
- b) melhorias na mecânica dos atuais minijogos e incremento na possibilidade de inclusão de novos minijogos;
- c) desenvolvimento de uma versão que possibilite jogos em grupos de alunos;
- d) possibilidade de alteração da estrutura da *engine* para permitir a atualização dos desafios de furbot a partir de um servidor remoto.

## REFERÊNCIAS

- ANDRADE, Daiane et al. Proposta de Atividades para o Desenvolvimento do Pensamento Computacional no Ensino Fundamental. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2., 2013. Campinas. **Anais...** Campinas: UNICAMP, 2013. p. 169-178.
- ARAÚJO, L. P.; SCHLÖGL, L.; MATTOS, M. M. Desenvolvimento Cognitivo com a Prática da Programação. In: SEMINÁRIO DE INFORMÁTICA E COMPUTAÇÃO, 25., 2017. Blumenau. **Anais...** Blumenau: FURB, 2017.
- BARCELOS, T. S.; SILVEIRA, I. F. Pensamento computacional e educação matemática: relações para o ensino de computação na educação básica. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 20., 2012. Curitiba. **Anais...** Curitiba: UFPR, 2012. p. 1-10.
- BEZERRA, F. L.; DIAS, K. L. Programação de computadores no ensino fundamental: experiências com logo e scratch em escola pública. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 22., 2014. Brasília. **Anais...** Brasília: SBC, 2014. p. 1515-1524.
- BORDINI, A.; AVILA, C.; CUNHA, M.; CAVALHEIRO, S. FOSS, L. Desdobramentos do pensamento computacional do Brasil. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 5., 2016. Uberlândia. **Anais...** Uberlândia, 2016. p. 200-209.
- BRACKMANN, C. P. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica.** 2017. 226 f. Tese (Doutorado em Informática na Educação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BURGUILLO, J. Using game theory and Competition-based Learning to stimulate student motivation and performance. **Computers & Education**, [S. l.], v. 15, p. 566-575, 2010.
- COELHO, A. R. **AFURBOT: Migração do Framework Furbot para Plataforma Android.** 42 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2013. Disponível em: [http://www.bc.furb.br/docs/MO/2013/353364\\_1\\_1.PDF](http://www.bc.furb.br/docs/MO/2013/353364_1_1.PDF). Acesso em: 18 jun. 2019.
- COSTA, T. et al. Trabalhando fundamentos de computação no nível fundamental: Experiência de licenciandos em computação da Universidade Federal da Paraíba. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 20., 2012. **Anais...** Curitiba: UFPR, 2012. p. 1- 10. Disponível em: [http://www2.sbc.org.br/csbc2012/anais\\_csbc/eventos/wei/artigos/Trabalhando%20Fundamentos%20de%20Computacao%20no%20Nivel%20Fundamental%20experiencia%20de%20licenciandos%20em%20Computacao%20da%20Universidade%20Federal%20da%20Paraiba.pdf](http://www2.sbc.org.br/csbc2012/anais_csbc/eventos/wei/artigos/Trabalhando%20Fundamentos%20de%20Computacao%20no%20Nivel%20Fundamental%20experiencia%20de%20licenciandos%20em%20Computacao%20da%20Universidade%20Federal%20da%20Paraiba.pdf). Acesso em: 18 jun. 2019.
- COSTA, A. C.; MARCHIORI, P. Gamificação, elementos de jogos e estratégia: uma matriz de referência. **Revista De Ciência Da Informação E Documentação**, v. 6, n.2, p. 44-65, 2015.
- CSTA - COMPUTER SCIENCE TEACHER ASSOCIATION. **CSTA K-12: computer science standards.** New York: ACM, 2011. Disponível em: <https://pt.scribd.com/document/251254516/CSTA-K-12-CSS>. Acesso em: 18 jun. 2019.
- DETERDING, S. et al. **Gamification: Using Game Design Elements in Non-Gaming Contexts.** Disponível em: <http://gamification-research.org/wp-content/uploads/2011/04/01-Deterding-Sicart-Nacke-OHara-Dixon.pdf>. Acesso em: 17 jun. 2019.

- DOS SANTOS, J. G.; CALÍOPE, T.; BARROS NETO, J. DE P. Sessão Especial - Fast Track Semead: Tem ação nessa pesquisa? Um Levantamento da pesquisa-ação como estratégia de pesquisa qualitativa. **REGE Revista de Gestão**, v. 24, n. 4, p. 336-347, 19 dez. 2017. Disponível em: <https://www.revistas.usp.br/rege/article/view/141475>. Acesso em: 29 jun. 2019.
- ECKSTEIN, J.; MANNS, M.; MARQUARDT, K.; WALLINGFORD, E. Patterns for Experiential Learning. In: THE 6TH EUROPEAN CONFERENCE ON PATTERN LANGUAGES OF PROGRAMMS, 6., 2001, Irsee. **Anais...** Irsee: CSIS. p.477-498. Disponível em: <http://csis.pace.edu/~bergin/patterns/ExperientialLearning.html>. Acesso em: 18 jun. 2019.
- FINCHER, S.; COOPER, S.; KOLLING, M.; MALONEY, J. Comparing alice, greenfoot and scratch, 10, 2010. In: THE 41ST ACM TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION, 41., 2010, Milwaukee. **Anais...** Milwaukee: SIGCSE, 2010. p. 192-193.
- FRANÇA, R. S.; FERREIRA, V. A. S.; ALMEIDA, L. M. F.; AMARAL, H. J. C. A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação. In: XXXIV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 34., 2014, Brasília. **Anais...** Brasília: SBC, 2014. p. 1505-1514.
- FRANÇA, R. S.; TEDESCO, P. C. A. R. Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil. In: WORKSHOP DE ENSINO EM PENSAMENTO COMPUTACIONAL, 1., 2015. **Anais...** Maceió, 2015. p. 1464-1473.
- GIL, A. C. **Métodos e técnicas de pesquisa social**. 6. ed. São Paulo: Atlas, 2008.
- GLORIA, A.; BELLOTTI, F.; BERTA, R. Serious Games for education and training. **International Journal of Serious Games**. 2014. Disponível em: [https://www.researchgate.net/publication/286244094\\_Serious\\_Games\\_for\\_education\\_and\\_training](https://www.researchgate.net/publication/286244094_Serious_Games_for_education_and_training). Acesso em: 18 jun. 2019.
- INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA. **Indicadores Educacionais: taxas de rendimento**. 2014. Disponível em: <http://portal.inep.gov.br/indicadores-educacionais>. Acesso em: 30 jun. 2019.
- KAZIMOGLU, C. et al. Learning programming at the computational thinking level via digital game-play. **Procedia Computer Science**, Omaha, Nebraska, USA, v. 9, n. 0, p. 522-531, 2012.
- KOPSCH, Heloisa K. **FURBOT-WEB: Uma Plataforma Adaptativa para o Ensino de Programação**. 119 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2016. Disponível em: [http://dsc.inf.furb.br/arquivos/tccs/monografias/2016\\_1\\_heloisa-kaestner-kopsch\\_monografia.pdf](http://dsc.inf.furb.br/arquivos/tccs/monografias/2016_1_heloisa-kaestner-kopsch_monografia.pdf). Acesso em: 18 jun. 2019.
- MACHADO, L. S. et al. Serious Games Baseados em Realidade Virtual para Educação Médica. **Revista Brasileira de Educação Médica**, Rio de Janeiro, SciELO, v. 35, n. 2, p. 254-262, jun. 2011.
- MATTOS, M. M. et al. Uma pesquisa-ação sobre o desenvolvimento do pensamento computacional com crianças. In: XXIV WORKSHOP DE INFORMÁTICA NA ESCOLA, 25., 2018, Fortaleza. **Anais...** Fortaleza: SBC, 2018a. p. 421-429.
- MATTOS, M.M. et al. Ensino do pensamento computacional em escola pública por meio de uma plataforma lúdica. In: VII CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 7., 2018, Fortaleza. **Anais...** Fortaleza: SBC, 2018b, p. 589-598.
- MATTOS, M. M.; KOHLER, L. P. A. O redesenho do projeto Furbot em Unity. [Entrevista concedida a] Joan Giancesini Tridapalli. Blumenau, 25 mar. 2019.
- MATTOS, M. M.; VAHLDICK, A.; HUGO, M. **Apostila de OO e Furbot. Blumenau: Furb**, 2008. Disponível em: [http://www.inf.furb.br/poo/furbot/files/Apostila\\_FURBOT.pdf](http://www.inf.furb.br/poo/furbot/files/Apostila_FURBOT.pdf). Acesso em: 19 jun. 2019.
- MAUS, Rafael. **FURBOT-C: Uma Abordagem Construcionista para a Construção do Conhecimento em Programação**. 68 f, 2011. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2011. Disponível em: [http://www.bc.furb.br/docs/MO/2011/348523\\_1\\_1.pdf](http://www.bc.furb.br/docs/MO/2011/348523_1_1.pdf). Acesso em: 19 jun. 2019.
- OLIVEIRA, Carolina Moreira. **Serious Game como Objeto de Aprendizagem para Programação de Computadores**. 64 f, 2013. Monografia (Graduação em Tecnologia em Sistemas para Internet) – Universidade Tecnológica Federal do Paraná. Guarapuava, 2013.
- PAULA, B. H.; VALENTE, J. A.; BURN, A. O Uso de Jogos Digitais para o Desenvolvimento do Currículo para a Educação Computacional na Inglaterra. **Currículo sem Fronteiras**, v. 14, p. 46-71, 2014. Disponível em: <http://www.curriculosemfronteiras.org/vol14iss3articles/paula-valente-burn.pdf>. Acesso em: 18 jun. 2019.
- PROGRAMME FOR INTERNATIONAL STUDENT ASSESSMENT. **Pisa 2015: Pisa results in focus**. 2015. Disponível em: <http://www.oecd.org/pisa/pisa-2015-results-in-focus.pdf>. Acesso em: 30 jun. 2019.
- PINTO, S. C. C. S., MATTOS, M. S. A programação de jogos como um instrumento motivador da aprendizagem. **Espaço Pedagógico**, v. 26, n. 2, p. 370-394, maio/ago. 2019. Disponível em: [www.upf.br/seer/index.php/rep](http://www.upf.br/seer/index.php/rep). Acesso em: 30 jun. 2019.
- PRAZERES, I. M. S.; OLIVEIRA, C. A. Gamificação na Educação básica pública – Possibilidades de Aplicação: Eixo 03 - Docência, criatividade, inovação e investigação. **Anais do 9º Simpósio Internacional de Educação e Comunicação – SIMEDUC**, 17 a 19 outubro 2018. Aracaju.

- RAABE A. L. A. et al. Referenciais de Formação em Computação: Educação Básica. In: CSBC 2017, 37., Porto Alegre, 2017. **Anais...** Disponível em: <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>. Acesso em: 10 jul. 2019.
- ROMERO, M.; LEPAGE, A.; LILLE, B. Computational thinking development through creative programming in higher education. **International Journal of Educational Technology in Higher Education**, v. 14, n. 1, p. 42, 2017.
- SBC. **Grandes desafios da pesquisa em computação no Brasil 2006-2016. Relatório Sociedade Brasileira de Computação**. Porto Alegre, 2006. Disponível em: <http://www.sbc.org.br/documentos-da-sbc/send/141-grandes-desafios/802-grandesdesafiosdacomputaonobrasil>. Acesso em: 09 jul. 2019.
- SORIA V., E.; RIVERO P. C. Pensamiento computacional: una nueva exigencia para la educación del siglo XXI. **Revista Espaço Pedagógico**, v. 26, n. 2, p. 323 -337, 2019.
- TOLOMEI, B. V. A. Gamificação como Estratégia de Engajamento e Motivação na Educação. **EaD em Foco**, v.7, n. 2, p. 145–156, 2017.
- TRIPP, D. Pesquisa-ação: uma introdução metodológica. **Educação e Pesquisa**, v. 31, n. 3, p. 443-466. Disponível em: <http://dx.doi.org/10.1590/S1517-97022005000300009>. Acesso em: 29 jun. 2019.
- UNITY. Multiplatform – Publish your game to over 25 platforms, 2019. Disponível em: <https://unity3d.com/pt/unity/features/multiplatform>. Acesso em: 29 jun. 2019.
- VAHLICK, A.; MATTOS, M. M. Relato de uma Experiência no Ensino de Algoritmos e Programação Utilizando um Framework Lúdico. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 19., 2008, Fortaleza. **Anais...** Fortaleza, 2008. Disponível em: [https://www.researchgate.net/publication/280879239\\_Relato\\_de\\_uma\\_Experiencia\\_no\\_Ensino\\_de\\_Algoritmos\\_e\\_Programacao\\_Utilizando\\_um\\_Framework\\_Ludico](https://www.researchgate.net/publication/280879239_Relato_de_uma_Experiencia_no_Ensino_de_Algoritmos_e_Programacao_Utilizando_um_Framework_Ludico). Acesso em: 19 jun. 2019.
- VALENTE, J. A. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. **Revista e-Curriculum**, v.14, n. 3, p.864-897. 2016.
- VALVERDE, J.; FERNÁNDEZ, M.; GARRIDO, M. El pensamiento computacional y las nuevas ecologías del aprendizaje. **Revista de Educación a Distancia**, v. 46, p. 1-18, 2015. Disponível em: <http://uaemex.redalyc.org/articulo.oa?id=54741184003>. Acesso em: 29 jun. 2019.
- VILANOVA, G. E. Tecnología educativa para el desarrollo del pensamiento computacional. **Revista Iberoamericana de Sistemas, Cibernética e Informática**, v. 15, n. 3, p. 25-32, 2018.
- WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n.3, p. 33-35, 2006. Disponível em: <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>. Acesso em: 19 jun. 2019.
- WUO, Andrea. Avaliação pedagógica dos minijogos. [Entrevista concedida a] Joan Giancesini Tridapalli. Blumenau, 03 jun. 2019.
- ZANETTI, H.; BORGES, M.; RICARTE, I. Pensamento Computacional no Ensino de Programação: Uma Revisão Sistemática da Literatura Brasileira. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 27., 2016, Uberlândia. **Anais...** Uberlândia: SBC, 2016. p. 21-30.
- ZYDA, M. From Visual Simulation to Virtual Reality to Games. **Computer**. v. 38, n. 9, p. 25-32, 2005. Disponível em: <https://ieeexplore.ieee.org/document/1510565>. Acesso em: 19 jun. 2019.