

Revisão do Projeto

Disciplina: Trabalho de Conclusão de Curso I – BCC

Caro orientando,

segue abaixo a tabela de cálculo da média das notas obtidas no Pré-Projeto e Projeto, as DUAS revisões do seu projeto contendo a avaliação do professor “avaliador” e professor “TCC1”. Lembro que os ajustes indicados nestas revisões não precisam ser feitos no projeto, mas sim quando levarem o conteúdo do projeto para o artigo (se for o caso). Este material contendo todo o histórico das revisões é encaminhado para o professor de TCC2.

Atenciosamente,

Nome	PreProjeto										Projeto													
	TCC1					Avaliador					Banca			TCC1					Avaliador					Média
	A	P	N	Nota	A	P	N	Nota	Ori.	Esp.	Nota	A	P	N	Nota	A	P	N	Nota					
GianCarloGiovanela	13	3	0	16	9,38	11	4	0	15	9,11	7,00	7,50	7,25	15	1	0	16	9,79	15	0	0	15	10,00	9,4

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
() PRÉ-PROJETO	(X) PROJETO	ANO/SEMESTRE: 2021/2

GERAÇÃO AUTOMATIZADA DE RESOLUÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT UTILIZANDO MÉTODOS DE APRENDIZAGEM DE MÁQUINA

Gian Carlo Giovanella

Prof^ª Luciana Pereira de Araújo Kohler – Orientadora

Prof^ª. Andreza Sartori – Coorientadora

1 INTRODUÇÃO

A Sociedade Brasileira de Computação (SBC) considera primordial que assuntos associados à Computação sejam inseridos na Educação Básica, sendo de grande relevância para a inclusão dos indivíduos na sociedade atual (RAABE *et al.*, 2017). Pensando em viabilizar a introdução aos conceitos básicos da área, atividades lúdicas amparadas por jogos eletrônicos são utilizadas para divertir e expandir a possibilidade de aprendizagem de conceitos, conteúdos e habilidades embutidas nesses jogos, motivando os alunos e auxiliando na assimilação dos conteúdos de uma forma envolvente e criativa (BURKE, 2015).

Nesse contexto, tem-se o jogo Furbot que tem a finalidade de auxiliar no desenvolvimento do pensamento computacional em crianças de ensino fundamental (ARAÚJO; MATTOS, 2018). O jogo é dividido em várias fases, nas quais os jogadores programam um robô utilizando comandos iniciais de programação (MATTOS, 2019). O problema que se apresenta neste momento é que não há um feedback ao aluno sobre a qualidade de sua solução. Com isto, o aluno não sabe se haveria alguma forma mais eficiente de resolver os exercícios. Segundo Zeferino, Domingues e Amaral (2007, p. 176), o feedback é fundamental no processo de aprendizagem, pois ajuda os alunos a identificar áreas de aprimoramento de seus conhecimentos ou habilidades, refletindo assim, em uma maior performance no processo de aprendizagem.

Nesse contexto, essa proposta tem como objetivo fornecer uma resolução de forma automatizada para cada fase do Furbot, que poderá ser utilizada para fornecer um feedback ao jogador quando este a conclui. Para a automatização deste processo serão utilizados métodos de aprendizagem de máquina produzindo a solução ótima de acordo com os objetivos do mapa de cada fase. Considera-se uma solução ótima, a solução que apresenta o menor caminho entre a origem e o destino.

1.1 OBJETIVOS

O objetivo deste trabalho é propor uma solução de automatização da resolução das fases do jogo Furbot, que encontre a resolução mais otimizada para cada fase.

Os objetivos específicos são:

- a) gerar uma solução ótima para cada fase do jogo Furbot, sendo esta, baseada no caminho mínimo entre a origem e os objetivos da fase;
- b) fornecer feedback para o jogador comparando sua solução com a solução ótima gerada;
- c) integrar a solução gerada no jogo existente do Furbot.

A programação com Minecraft: Jornada do Herói é uma das várias ferramentas de ensino de programação desenvolvidas pela Code.org (2021a). A ferramenta introduz diversos conceitos de programação, tais como

programação sequencial, uso de laços de repetição e uso de funções pré-determinadas. A ferramenta propõe uma programação visual com o uso de blocos, sendo que seus comandos estão disponíveis em vários idiomas. O objetivo é usar os comandos disponíveis para realizar o que é proposto no início de cada fase.

A Figura 2 identifica a tela principal do jogo. Ela é dividida em alguns componentes, sendo eles: as instruções da fase (identificado pela letra A na figura); blocos para utilizar na resolução (identificado pela letra B na figura); área de trabalho que é onde o aluno deve incluir os blocos para montar o código para ser executado pelo personagem auxiliar (identificado pela letra C na figura). Ao clicar no botão executar, o personagem executa os comandos da área de trabalho, identificado pela letra F na figura, e ao finalizar, o botão de recomendar é exibido no lugar do botão executar. Caso haja um erro na sequência de passos, o personagem controlado pelo aluno não conseguirá concluir o problema. Assim, o aluno deverá pensar no que errou e recomendar ou pedir uma dica clicando na lâmpada situada no quadrado das instruções, indicado pela letra D na figura.

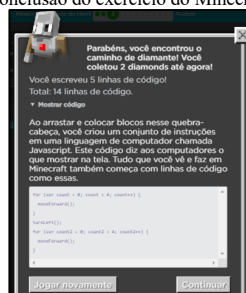
Figura 2 – Tela principal do Minecraft: Jornada do Herói



Fonte: elaborado pelo autor.

Ao pedir para executar os passos, conforme as ações da área de trabalho são executadas, elas são destacadas em amarelo, como pode ser observado na letra E da Figura 2. Caso o usuário conclua o objetivo, além de uma mensagem de parabéns, é apresentado o código gerado na linguagem JavaScript, conforme a Figura 3. As tecnologias utilizadas para o desenvolvimento incluem JavaScript e Javascript XML (JSX) que permite escrever código HTML dentro do React.

Figura 3 – Tela de conclusão do exercício do Minecraft: Jornada do Herói



Fonte: elaborado pelo autor.

2.3 CODESPARK

CodeSpark é uma plataforma de minijogos online que pode ser acessada pelo navegador ou por um aplicativo (CODESPARK, 2021b). No CodeSpark o jogador deve programar os personagens clicando e arrastando

comandos em sequência. Após finalizar a escolha e ordem dos comandos, necessita clicar no personagem para que ele execute as ações escolhidas, como pode ser observado na Figura 4.

Figura 4 – Ambiente do Jogo codeSpark



Fonte: elaborado pelo autor.

O cenário é formado por vários objetos comuns do dia a dia. O ambiente apresentado ao jogador varia de acordo com o quebra cabeça e sempre possui símbolos como os que se encontram na parte inferior da Figura 4, que se traduzem em funções tais como: seguir em frente, virar para esquerda, virar para direita ou pular em alguma direção.

O jogo proporciona a prática de vários conceitos de programação sendo eles: programação sequencial; lógica condicional e uso de funções. Uma vez executado, é possível visualizar a execução dos comandos inseridos. Para concluir o objetivo é necessário coletar todas as esmeraldas do mapa.

A plataforma não verifica o menor caminho entre o objetivo de cada fase, apenas verifica quantas esmeraldas foram coletadas e gera uma pontuação a partir disso. O número de comandos que o usuário pode utilizar em cada fase é limitado, induzindo-o a utilizar os comandos na sequência que é pré-determinada.

3 FERRAMENTA ATUAL

O Furbot teve sua origem em 2008 com o propósito de auxiliar o ensino dos conceitos iniciais de programação orientada a objetos, voltado aos alunos de graduação do departamento de Sistemas e Computação da FURB, por meio de um *framework* Java (VAHLICK; MATTOS, 2008). Em 2017, o *framework* foi utilizado para a concepção de um ambiente de programação com o objetivo de introduzir o pensamento computacional em crianças de ensino fundamental, sem a necessidade da instalação de um ambiente de programação ou destas conhecerem uma linguagem de programação (SCHLÖGL *et al.*, 2017). Por sua vez, em 2018, o Furbot passou por uma reformulação se transformando em uma plataforma composta por um jogo e um servidor (MATTOS, 2019). O jogo, desenvolvido em Unity na linguagem C#, possui um apelo mais lúdico e um gráfico mais atrativo para o público-alvo, que são as crianças de ensino fundamental, em relação ao ambiente desenvolvido em 2017 (MATTOS, 2019). Já o servidor que acompanha a plataforma é utilizado para manter os dados dos jogadores de modo que os professores possam acompanhar suas turmas e visualizar os exercícios resolvidos pelos seus alunos (MATTOS, 2019). Este servidor é desenvolvido em Java utilizando o *framework* Spring. Como o foco deste trabalho é o jogo do Furbot, este é explicado com maiores detalhes.

O jogo possui uma narrativa que tem como personagens a Professora Sam que cria o Furbot. O Furbot é um robô que os alunos precisam programar para salvar o mundo do ataque dos malvados *buggiens*. A Figura 5 apresenta uma visão da tela de uma das fases iniciais. Nessa tela, tem-se o painel de comandos, localizado à direita

da Figura 5, que permite que o aluno desenvolva os comandos necessários para movimentar o Furbot no cenário da fase.



Fonte: elaborado pelo autor.

Cada fase possui desafios próprios com o objetivo de chegar ao final do caminho coletando tesouros, evitando os *buggiens* e desviando dos obstáculos, para então continuar progredindo para as próximas fases. O jogo introduz os conceitos de programação sequencial, utilização de condicionais e o uso de laços de repetição.

Ao finalizar a fase, é exibida uma tela mostrando a pontuação do aluno. Nela, o aluno pode retornar ao jogo e tentar fazer a fase novamente a fim de aumentar sua pontuação ou seguir em frente para a próxima fase. Contudo, o aluno não tem o feedback se sua solução é a melhor possível ou não.

4 PROPOSTA

Nesta seção serão apresentadas as justificativas para o desenvolvimento deste trabalho. Além disso, serão apresentados os principais Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF), bem como suas características e a metodologia de desenvolvimento que será utilizada.

4.1 JUSTIFICATIVA

No Quadro 3 é apresentada uma comparação entre os trabalhos correlatos. Nas colunas são identificados os trabalhos e nas linhas as características.

Quadro 1 – Comparativo entre os trabalhos correlatos

	Geração automatizada de gabarito (XAVIER, 2019)	Minecraft: Jornada do Herói (CODE.ORG, 2018b)	CodeSpark (CODESPARK, 2021b)
Objetivo educacional	Sim	Sim	Sim
Conferência automatizada do resultado da resolução do exercício	Sim	Sim	Sim
Verifica se a resolução é a melhor possível	Sim	Não	Não
Plataforma	Web	Web	Web/App

Fonte: elaborado pelo autor.

Como pode ser observado no Quadro 3, todos os trabalhos são ferramentas que foram construídas para fins educacionais e verificam de forma automatizada se o objetivo foi concluído. Tanto o CodeSpark (CODESPARK, 2021a), quanto o Minecraft: Jornada do Herói (CODE.ORG, 2018b) não demonstram ao aluno se existe alguma solução melhor do que a desenvolvida por ele. Em relação ao trabalho de Xavier (2019), é utilizado o algoritmo

de busca de menor caminho Dijkstra para solucionar e encontrar a solução ótima dos exercícios. Desta forma, há uma referência que pode ser utilizada pelos alunos para chegar na melhor solução.

Analisando os correlatos apresentados, o único trabalho que se preocupa em gerar uma solução ótima é o trabalho de Xavier (2019). Contudo, o autor concluiu que essa solução não possui bom desempenho, possuindo um elevado tempo de execução, o qual o autor atribuiu ao elevado número de objetivos e obstáculos que o jogo possui. Além disso, esse tipo de algoritmo não é suficiente para a versão atual do Furbot, pois além destes itens, ainda deve ser considerado os tipos de terreno, itens coletados no mapa, gasto de energia e inimigos encontrados.

Em relação à aplicação a ser desenvolvida, podem existir várias formas de resolver um mesmo exercício no Furbot. Criar um processo automatizado para verificar a melhor solução a partir dos objetivos da fase, proporciona uma referência que os alunos podem seguir para se desafiar a aprimorar suas soluções. Para gerar a melhor solução, um estudo acerca dos principais algoritmos utilizados em aprendizagem supervisionada será realizado, observando as revisões bibliográficas mais recentes e dessa forma, será definido o que mais se encaixa no contexto dos objetivos propostos.

Tendo em vista os pontos levantados, torna-se importante a construção da aprendizagem a partir do erro, instigando e inspirando os alunos a encontrarem as melhores soluções para os desafios propostos. Com isso, é possível contribuir na formação deles e expandir o ensino de pensamento computacional desde cedo nas escolas.

Em relação à contribuição tecnológica, o trabalho proposto pode verificar a viabilidade dos métodos de aprendizagem de máquina no contexto de automatizar a busca pelos melhores resultados dado os objetivos do jogo, podendo ser utilizado para aprimorar a bonificação dos jogadores. No que se refere à contribuição pedagógica, desenvolver o recurso proposto pelo presente trabalho dentro do jogo, intensifica os objetivos educacionais do jogo do Furbot.

4.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A aplicação descrita nesse trabalho deverá:

- a) encontrar a melhor resposta para cada fase do jogo Furbot (Requisito Funcional - RF);
- b) gerar um gabarito com a solução ótima contendo os comandos sequenciais que o robô deve fazer para sair do ponto de origem e chegar até o ponto de destino (RF);
- c) comparar a resolução do aluno com a gerada pelo algoritmo (RF);
- d) ser implementado na linguagem Python (Requisito Não Funcional - RNF);
- e) ser implementado no ambiente de desenvolvimento Visual Studio (RNF).

4.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: levantar material bibliográfico pertinente a utilização de ensino de lógica e programação no ensino, pensamento computacional, aprendizagem de máquina e demais temas que forem necessários, além de aprimorar os trabalhos correlatos;
- b) elicitação de requisitos: revisar o conjunto de requisitos funcionais e não funcionais previamente identificados nesta proposta;
- c) especificação: formalizar as funcionalidades do projeto através de diagramas de casos de uso, de classe, de atividade e de componentes da Unified Modeling Language (UML) utilizando a ferramenta LucidChart;
- d) implementação do algoritmo de aprendizagem: implementar e testar os principais algoritmos de

aprendizagem de máquina na linguagem Python no ambiente de desenvolvimento Visual Studio.

- e) integração com o Furbot: integrar o algoritmo Python com a arquitetura do Furbot utilizando a linguagem C#;
- f) testes e validações: serão realizados testes juntamente com a construção dos algoritmos comparando as soluções geradas pelos algoritmos com a dos alunos e com a solução desenvolvida por Xavier (2019), medindo a eficiência das soluções.

As etapas de implementação e testes serão desenvolvidas de forma integrada, de modo que haja uma evolução contínua em relação ao desempenho na obtenção do melhor caminho, até se identificar o melhor algoritmo a ser utilizado. As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

etapas / quinzenas	2022									
	fev.		mar.		abr.		mai.		jun.	
	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
elicitación de requisitos										
especificação										
implementação do algoritmo de aprendizagem										
integração com o Furbot										
testes e validações										

Fonte: elaborado pelo autor.

5 REVISÃO BIBLIOGRÁFICA

Este capítulo está dividido em duas seções. A seção 5.1 aborda uma visão geral sobre a computação na educação básica. Na seção 5.2 são descritos os fundamentos principais sobre aprendizagem de máquina e métodos de aprendizagem supervisionados.

5.1 COMPUTAÇÃO NA EDUCAÇÃO BÁSICA

A inclusão de conceitos gerais de computação no ensino básico faz-se essencial como forma de aprimorar o raciocínio lógico-matemático das crianças e incentivar a criatividade por meio de tecnologias (NASCIMENTO *et al.*, 2015). Ainda, seguindo este contexto, Scaico *et al.* (2013) complementam que o ensino de conceitos de programação proporciona o desenvolvimento de várias habilidades que colaboram para aprimorar o raciocínio lógico dos estudantes. Programar abrange a capacidade de desenvolver uma solução para um problema, que caso seja grande demandará o exercício de outras habilidades (SCAICO *et al.*, 2013).

Fundamentos da computação, agregados ao pensamento crítico definem um método para solucionar problemas intitulado Pensamento Computacional (WING, 2006). Esta perspectiva argumenta que a cultura do computador ajuda a sociedade a aprender, mas especialmente, possibilita uma nova forma de aprender a aprender (LU; FLETCHER, 2009). O Pensamento Computacional é um método para solução de problemas que utiliza os princípios e técnicas da Ciência da Computação.

Segundo Ribeiro, Foss e Cavalheiro (2017), as três habilidades básicas que fundamentam o Pensamento Computacional são: abstração, automação e análise. A habilidade de abstração se associa a aptidão do indivíduo extrair somente os aspectos mais relevantes de um problema para chegar a uma solução. A habilidade de automação associa-se a aptidão de utilização um meio eletrônico que é capaz de substituir o trabalho de um ser humano e a habilidade de análise associa-se ao estudo dos resultados criados pela automação.

Como Brackmann (2017) afirma, enquanto os alunos aprendem a programar, estão também programando para aprender. Este aprendizado possibilita que eles conheçam várias outras coisas e criem oportunidades de aprendizagem.

5.2 APRENDIZAGEM DE MÁQUINA E MÉTODOS

Como afirma Bishop (2006), Sistemas de Aprendizado de Máquina são sistemas que combinam algoritmos e modelos matemáticos da área de Aprendizado de Máquina em soluções que visam a implantação do uso de tais algoritmos e modelos em ambientes reais. Aprendizado de máquina, por sua vez, se refere a área de estudos voltada ao desenvolvimento e compreensão de modelos matemáticos caracterizados pelo aprimoramento de seus resultados por meio da ingestão de dados de treino, de forma que esses modelos possam realizar previsões e decisões sem que sejam explicitamente programados para o fazerem.

A aprendizagem de máquina é dividida em três tipos de técnicas: a aprendizagem supervisionada, que treina um modelo a partir de dados conhecidos de entrada e saída para que ele possa prever resultados futuros; a aprendizagem não supervisionada, que encontra padrões ocultos ou estruturas específicas a partir de dados de entrada; e a aprendizagem por reforço, que utiliza um sistema de recompensas ou punições para se aperfeiçoar no ambiente que se encontra (NORVIG, RUSSELL, 2014).

Na aprendizagem supervisionada, Norvig e Russel (2014) descrevem que o agente analisa determinados exemplos de entrada e saída correspondentes, e aprende uma função que faz o mapeamento da entrada para a saída. Exemplificando, o agente pode aprender a reconhecer fotos de ônibus ao ver muitas imagens que lhe mostram ônibus.

Na aprendizagem não supervisionada, Norvig e Russel (2014) explicam que o agente estuda padrões na entrada, ainda que nenhum feedback explícito seja fornecido. A tarefa mais utilizada na aprendizagem não supervisionada é a detecção e agrupamento de exemplos de entrada potencialmente úteis. Exemplificando, um agente de táxi pode desenvolver gradualmente um conceito de “dia de tráfego bom” e “dia de tráfego ruim” sem necessariamente terem sido rotulados anteriormente.

Por fim, na aprendizagem por reforço, o agente aprende com base em uma sucessão de reforços, recompensas ou punições. Exemplificando, um agente de táxi entende que algo saiu errado a partir da falta de gorjeta ao final de uma corrida (NORVIG, RUSSELL, 2014).

REFERÊNCIAS

- ARAÚJO, Luciana; MATTOS, Mauro. **Furbot**: plataforma para o ensino-aprendizado de pensamento computacional e programação. In: I CONCURSO LATINOAMERICANO DE TECNOLOGIAS EDUCACIONAIS PARA APRENDIZAGEM (EDUTECH), 1., 2018, São Paulo. 1p.
- BISHOP, C. M., **Pattern Recognition and Machine Learning**, Springer, v. ISBN 978-0-387-31073-2, 2006.
- BRACKMANN, C. P., **Desenvolvimento do Pensamento Computacional Através de Atividades Desplugadas na Educação Básica**. 2017. 226 f. Tese (Doutorado) - Curso de Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BURKE, B. **Gamificar**: como a gamificação motiva as pessoas a fazerem coisas extraordinárias. São Paulo: DVS Editora, 2015.
- CODE.ORG. **Minecraft: Jornada do Herói**, [s.l.], 2021a. Disponível em: <https://studio.code.org/s/hero/lessons/1/levels/1>. Acesso em: 27 mai. 2021.
- CODE.ORG. **About Us**, [s.l.] 2021b. Disponível em: <https://code.org/about>. Acesso em: 27 mai. 2021.
- CODESPARK. **About Us**, [s.l.], 2021a. Disponível em: <https://codespark.com/about>. Acesso em: 25 mai. 2021.

CODESPARK. **Frequently Asked Questions**, [s.l.] 2021b. Disponível em: <https://codespark.com/frequently-asked-questions>. Acesso em: 25 mai. 2021.

Lu, J. J.; Fletcher, G. H. (2009). Thinking about computational thinking. In Proc. 40th Technical Symp. on Comp. Sci. Education, pages 260–264, New York, USA. ACM.

MATTOS, Mauro et al. **Furbot Móvel**: um jogo para o ensino do pensamento computacional. In: VIII CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 2019, Brasília. Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação (CBIE 2019). [S.L.]: Brazilian Computer Society (Sociedade Brasileira de Computação - Sbc), 2019. p. 1294-1301.

Nascimento, J., Xavier, D., Passos, O. e Barreto, R., (2015) **Um Relato de Experiência da Aplicação de Técnicas Interativas para Ensino da Computação na Educação Básica**. Anais do Encontro Regional de Computação e Sistemas de Informação. p. 95-104.

NORVIG, Peter; RUSSELL, Stuart. **Inteligência Artificial**: Tradução da 3a Edição. Elsevier Brasil, 2014.

RAABE, André L. A. *et al.* **Referenciais de Formação em Computação**: Educação Básica. Porto Alegre. Assembleias do WEI e da SBC. 2017. 9 f. Disponível em: <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf> Acesso em: 27 mai. 2021.

RIBEIRO, L.; FOSS, L.; CAVALHEIRO, S. A. da C. **Entendendo o Pensamento Computacional**. Jul 2017. Disponível em: <https://arxiv.org/abs/1707.00338>. Acesso em: 01 jun. 2021.

SCAICO, Pasqueline D. *et al.* **Ensino de Programação no Ensino Médio**: Uma Abordagem Orientada ao Design com a linguagem Scratch. Revista Brasileira de Informática na Educação, Porto Alegre, v. 21, n. 2, p. 92, set. 2013.

Schlögl, Lucas Eduardo; Oliveira, Gabriel Castellani de; Giovannella, Gian Carlo; Bizon, A. R.; Santos, B. F. F.; Kruger, N.; Bursoni, P.; Neumann, C. B.; Huber, E. E.; Araújo, Luciana P.; Mattos, Mauro M.; Zucco, F. D.; Cunha, K. Z.; Hein, N. **Ensino do Pensamento Computacional na Educação Básica**. Revista de Sistemas e Computação - RSC, v. 7, p. 304-322, 2017.

Vahldick, Adilson; Mattos, Mauro Marcelo. Relato de uma Experiência no Ensino de Algoritmos e Programação Utilizando um Framework Lúdico In: Simpósio Brasileiro de Informática na Educação, 2008, Fortaleza. XIX Simpósio Brasileiro de Informática na Educação. 2008.

Wing, J. M. (2006). **Computational thinking**. Communications of the ACM, v. 49, n. 3, p. 33–35.

XAVIER, Francisca Edyr. **Geração Automatizada de Gabarito e Correção de Exercícios em Ambiente Furbot Para o Ensino do Pensamento Computacional**. 2019. 26 f. TCC (Graduação) - Curso de Ciências da Computação, Universidade Regional de Blumenau, Blumenau, 2019. Disponível em: http://dsc.inf.furb.br/arquivos/tccs/monografias/2019_1_francisca-edyr-xavier_monografia.pdf. Acesso em: 01 maio 2021.

ZEFERINO, A. M. B.; DOMINGUES, R. C. L.; AMARAL, E. **Feedback como estratégia de aprendizado no ensino médico**. Rev. bras. educ. med. [online]. 2007, vol.31, n.2, pp.176-179. Disponível em: <https://www.scielo.br/j/rbem/a/yK7SFyqJBCm6h6RqNk4Szyt/abstract/?lang=pt>. Acesso em: 05 jun 2021.

FORMULÁRIO DE AVALIAÇÃO BCC – PROFESSOR AVALIADOR

Avaliador(a): Marcel Hugo

Atenção: quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

ASPECTOS AVALIADOS		Atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	X		
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?	X		
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?	X		
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?	X		
	7. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?	X		
ASPECTOS METODOLÓGICOS	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	X		
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		

O projeto de TCC será reprovado se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos **4 (quatro)** itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos **4 (quatro)** itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

PARECER: (X) APROVADO () REPROVADO

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
() PRÉ-PROJETO	(X) PROJETO	ANO/SEMESTRE: 2021/2

GERAÇÃO AUTOMATIZADA DE RESOLUÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT UTILIZANDO MÉTODOS DE APRENDIZAGEM DE MÁQUINA

Gian Carlo Giovanella

Profª Luciana Pereira de Araújo Kohler – Orientadora

Profª. Andreza Sartori – Coorientadora

1 INTRODUÇÃO

A Sociedade Brasileira de Computação (SBC) considera primordial que assuntos associados à Computação sejam inseridos na Educação Básica, sendo de grande relevância para a inclusão dos indivíduos na sociedade atual (RAABE *et al.*, 2017). Pensando em viabilizar a introdução aos conceitos básicos da área, atividades lúdicas amparadas por jogos eletrônicos são utilizadas para divertir e expandir a possibilidade de aprendizagem de conceitos, conteúdos e habilidades embutidas nesses jogos, motivando os alunos e auxiliando na assimilação dos conteúdos de uma forma envolvente e criativa (BURKE, 2015).

Nesse contexto, tem-se o jogo Furbot que tem a finalidade de auxiliar no desenvolvimento do pensamento computacional em crianças de ensino fundamental (ARAÚJO; MATTOS, 2018). O jogo é dividido em várias fases, nas quais os jogadores programam um robô utilizando comandos iniciais de programação (MATTOS, 2019). O problema que se apresenta neste momento é que não há um feedback ao aluno sobre a qualidade de sua solução. Com isto, o aluno não sabe se haveria alguma forma mais eficiente de resolver os exercícios. Segundo Zeferino, Domingues e Amaral (2007, p. 176), o feedback é fundamental no processo de aprendizagem, pois ajuda os alunos a identificar áreas de aprimoramento de seus conhecimentos ou habilidades, refletindo assim, em uma maior performance no processo de aprendizagem.

Nesse contexto, essa proposta tem como objetivo fornecer uma resolução de forma automatizada para cada fase do Furbot, que poderá ser utilizada para fornecer um feedback ao jogador quando este a conclui. Para a automatização deste processo serão utilizados métodos de aprendizagem de máquina produzindo a solução ótima de acordo com os objetivos do mapa de cada fase. Considera-se uma solução ótima, a solução que apresenta o menor caminho entre a origem e o destino.

1.1 OBJETIVOS

O objetivo deste trabalho é propor uma solução de automatização da resolução das fases do jogo Furbot, que encontre a resolução mais otimizada para cada fase.

Os objetivos específicos são:

- a) gerar uma solução ótima para cada fase do jogo Furbot, sendo esta, baseada no caminho mínimo entre a origem e os objetivos da fase;
- b) fornecer feedback para o jogador comparando sua solução com a solução ótima gerada;
- c) integrar a solução gerada no jogo existente do Furbot.

Comentado [DSdR1]: Arrumar alinhamento dos itens com a mesma margem do parágrafo.

2 TRABALHOS CORRELATOS

Nesta seção serão apresentados os trabalhos que possuem características semelhantes às dos objetivos propostos nesse trabalho. O primeiro aborda o trabalho de Xavier (2019) que é a geração automatizada de gabarito e correção de exercícios em ambiente Furbot para o ensino do pensamento computacional. O segundo é uma ferramenta para ensino introdutório de programação (CODE.ORG, 2021a). Por fim, o terceiro trabalho é um jogo de quebra cabeça chamado CodeSpark (CODESPARK, 2021a).

2.1 GERAÇÃO AUTOMATIZADA DE GABARITO E CORREÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT

O trabalho desenvolvido por Xavier (2019) teve como objetivo gerar gabaritos para as fases do projeto Furbot. Este projeto é desenvolvido em Java para o ensino do pensamento computacional voltado às crianças da rede básica de educação (SCHLÖGL *et al.*, 2017).

Xavier (2019) utilizou algoritmos de menor caminho para solucionar problemas como: encontrar o menor caminho até os objetivos do robô; gerar um gabarito dos comandos sequenciais que o robô deve fazer para sair do ponto de origem e chegar até o ponto de destino (XAVIER, 2019). Além disso, o trabalho também teve como objetivo auxiliar os professores a fazer a correção dos exercícios criados pelos alunos, visto que muitos dos professores não possuem experiência na área de programação. As tecnologias utilizadas no desenvolvimento incluem o *framework* Furbot na linguagem Java, além da biblioteca *Commons-digester-1.8* para leitura de arquivos eXtensible Markup Language (XML) e o algoritmo de Dijkstra utilizado para encontrar os caminhos mínimos entre os objetivos do mapa (XAVIER, 2019).

Na Figura 1 pode-se visualizar a comparação entre o código gerado pelo aluno e o gabarito fornecido pelo algoritmo de Xavier (2019). Ao concluir a fase, o aluno visualiza a quantidade de linhas que ele escreveu e a quantidade de linhas que poderia ter escrito. Ainda, é apresentada a pontuação com base nos objetivos do mapa.

Figura 1 – Tela de comparação resolução com o gabarito



Fonte: Xavier, 2019.

Por meio de testes em campo realizado com alunos de ensino fundamental, Xavier (2019) pôde observar como resultado uma maior motivação entre os alunos que passaram a comparar suas soluções com o gabarito gerado. Ainda, Xavier (2019) aponta que há uma demora para gerar o gabarito ao colocar muitos objetivos no mapa, sendo esta uma limitação do trabalho.

2.2 PROGRAMAÇÃO COM MINECRAFT: JORNADA DO HEROI

A programação com Minecraft: Jornada do Herói é uma das várias ferramentas de ensino de programação desenvolvidas pela Code.org (2021a). A ferramenta introduz diversos conceitos de programação, tais como

programação sequencial, uso de laços de repetição e uso de funções pré-determinadas. A ferramenta propõe uma programação visual com o uso de blocos, sendo que seus comandos estão disponíveis em vários idiomas. O objetivo é usar os comandos disponíveis para realizar o que é proposto no início de cada fase.

A Figura 2 identifica a tela principal do jogo. Ela é dividida em alguns componentes, sendo eles: as instruções da fase (identificado pela letra A na figura); blocos para utilizar na resolução (identificado pela letra B na figura); área de trabalho que é onde o aluno deve incluir os blocos para montar o código para ser executado pelo personagem auxiliar (identificado pela letra C na figura). Ao clicar no botão executar, o personagem executa os comandos da área de trabalho, identificado pela letra F na figura, e ao finalizar, o botão de recomendar é exibido no lugar do botão executar. Caso haja um erro na sequência de passos, o personagem controlado pelo aluno não conseguirá concluir o problema. Assim, o aluno deverá pensar no que errou e recomendar ou pedir uma dica clicando na lâmpada situada no quadrado das instruções, indicado pela letra D na figura.

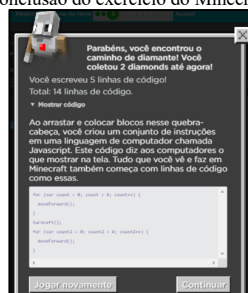
Figura 2 – Tela principal do Minecraft: Jornada do Herói



Fonte: elaborado pelo autor.

Ao pedir para executar os passos, conforme as ações da área de trabalho são executadas, elas são destacadas em amarelo, como pode ser observado na letra E da Figura 2. Caso o usuário conclua o objetivo, além de uma mensagem de parabéns, é apresentado o código gerado na linguagem JavaScript, conforme a Figura 3. As tecnologias utilizadas para o desenvolvimento incluem JavaScript e Javascript XML (JSX) que permite escrever código HTML dentro do React.

Figura 3 – Tela de conclusão do exercício do Minecraft: Jornada do Herói



Fonte: elaborado pelo autor.

2.3 CODESPARK

CodeSpark é uma plataforma de minijogos online que pode ser acessada pelo navegador ou por um aplicativo (CODESPARK, 2021b). No CodeSpark o jogador deve programar os personagens clicando e arrastando

comandos em sequência. Após finalizar a escolha e ordem dos comandos, necessita clicar no personagem para que ele execute as ações escolhidas, como pode ser observado na Figura 4.

Figura 4 – Ambiente do Jogo codeSpark



Fonte: elaborado pelo autor.

O cenário é formado por vários objetos comuns do dia a dia. O ambiente apresentado ao jogador varia de acordo com o quebra cabeça e sempre possui símbolos como os que se encontram na parte inferior da Figura 4, que se traduzem em funções tais como: seguir em frente, virar para esquerda, virar para direita ou pular em alguma direção.

O jogo proporciona a prática de vários conceitos de programação sendo eles: programação sequencial; lógica condicional e uso de funções. Uma vez executado, é possível visualizar a execução dos comandos inseridos. Para concluir o objetivo é necessário coletar todas as esmeraldas do mapa.

A plataforma não verifica o menor caminho entre o objetivo de cada fase, apenas verifica quantas esmeraldas foram coletadas e gera uma pontuação a partir disso. O número de comandos que o usuário pode utilizar em cada fase é limitado, induzindo-o a utilizar os comandos na sequência que é pré-determinada.

3 FERRAMENTA ATUAL

O Furbot teve sua origem em 2008 com o propósito de auxiliar o ensino dos conceitos iniciais de programação orientada a objetos, voltado aos alunos de graduação do departamento de Sistemas e Computação da FURB, por meio de um *framework* Java (VAHLICK; MATTOS, 2008). Em 2017, o *framework* foi utilizado para a concepção de um ambiente de programação com o objetivo de introduzir o pensamento computacional em crianças de ensino fundamental, sem a necessidade da instalação de um ambiente de programação ou destas conhecerem uma linguagem de programação (SCHLÖGL *et al.*, 2017). Por sua vez, em 2018, o Furbot passou por uma reformulação se transformando em uma plataforma composta por um jogo e um servidor (MATTOS, 2019). O jogo, desenvolvido em Unity na linguagem C#, possui um apelo mais lúdico e um gráfico mais atrativo para o público-alvo, que são as crianças de ensino fundamental, em relação ao ambiente desenvolvido em 2017 (MATTOS, 2019). Já o servidor que acompanha a plataforma é utilizado para manter os dados dos jogadores de modo que os professores possam acompanhar suas turmas e visualizar os exercícios resolvidos pelos seus alunos (MATTOS, 2019). Este servidor é desenvolvido em Java utilizando o *framework* Spring. Como o foco deste trabalho é o jogo do Furbot, este é explicado com maiores detalhes.

O jogo possui uma narrativa que tem como personagens a Professora Sam que cria o Furbot. O Furbot é um robô que os alunos precisam programar para salvar o mundo do ataque dos malvados *buggiens*. A Figura 5 apresenta uma visão da tela de uma das fases iniciais. Nessa tela, tem-se o painel de comandos, localizado à direita

da Figura 5, que permite que o aluno desenvolva os comandos necessários para movimentar o Furbot no cenário da fase.



Fonte: elaborado pelo autor.

Cada fase possui desafios próprios com o objetivo de chegar ao final do caminho coletando tesouros, evitando os *buggiens* e desviando dos obstáculos, para então continuar progredindo para as próximas fases. O jogo introduz os conceitos de programação sequencial, utilização de condicionais e o uso de laços de repetição.

Ao finalizar a fase, é exibida uma tela mostrando a pontuação do aluno. Nela, o aluno pode retornar ao jogo e tentar fazer a fase novamente a fim de aumentar sua pontuação ou seguir em frente para a próxima fase. Contudo, o aluno não tem o feedback se sua solução é a melhor possível ou não.

4 PROPOSTA

Nesta seção serão apresentadas as justificativas para o desenvolvimento deste trabalho. Além disso, serão apresentados os principais Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF), bem como suas características e a metodologia de desenvolvimento que será utilizada.

4.1 JUSTIFICATIVA

No Quadro 3 é apresentada uma comparação entre os trabalhos correlatos. Nas colunas são identificados os trabalhos e nas linhas as características.

Quadro 2 – Comparativo entre os trabalhos correlatos

	Geração automatizada de gabarito (XAVIER, 2019)	Minecraft: Jornada do Herói (CODE.ORG, 2018b)	CodeSpark (CODESPARK, 2021b)
Objetivo educacional	Sim	Sim	Sim
Conferência automatizada do resultado da resolução do exercício	Sim	Sim	Sim
Verifica se a resolução é a melhor possível	Sim	Não	Não
Plataforma	Web	Web	Web/App

Fonte: elaborado pelo autor.

Como pode ser observado no Quadro 3, todos os trabalhos são ferramentas que foram construídas para fins educacionais e verificam de forma automatizada se o objetivo foi concluído. Tanto o CodeSpark (CODESPARK, 2021a), quanto o Minecraft: Jornada do Herói (CODE.ORG, 2018b) não demonstram ao aluno se existe alguma solução melhor do que a desenvolvida por ele. Em relação ao trabalho de Xavier (2019), é utilizado o algoritmo

Comentado [DSdR2]: Arrumar o alinhamento do recuo do parágrafo.

de busca de menor caminho Dijkstra para solucionar e encontrar a solução ótima dos exercícios. Desta forma, há uma referência que pode ser utilizada pelos alunos para chegar na melhor solução.

Analisando os correlatos apresentados, o único trabalho que se preocupa em gerar uma solução ótima é o trabalho de Xavier (2019). Contudo, o autor concluiu que essa solução não possui bom desempenho, possuindo um elevado tempo de execução, o qual o autor atribuiu ao elevado número de objetivos e obstáculos que o jogo possui. Além disso, esse tipo de algoritmo não é suficiente para a versão atual do Furbot, pois além destes itens, ainda deve ser considerado os tipos de terreno, itens coletados no mapa, gasto de energia e inimigos encontrados.

Em relação à aplicação a ser desenvolvida, podem existir várias formas de resolver um mesmo exercício no Furbot. Criar um processo automatizado para verificar a melhor solução a partir dos objetivos da fase, proporciona uma referência que os alunos podem seguir para se desafiar a aprimorar suas soluções. Para gerar a melhor solução, um estudo acerca dos principais algoritmos utilizados em aprendizagem supervisionada será realizado, observando as revisões bibliográficas mais recentes e dessa forma, será definido o que mais se encaixa no contexto dos objetivos propostos.

Tendo em vista os pontos levantados, torna-se importante a construção da aprendizagem a partir do erro, instigando e inspirando os alunos a encontrarem as melhores soluções para os desafios propostos. Com isso, é possível contribuir na formação deles e expandir o ensino de pensamento computacional desde cedo nas escolas.

Em relação à contribuição tecnológica, o trabalho proposto pode verificar a viabilidade dos métodos de aprendizagem de máquina no contexto de automatizar a busca pelos melhores resultados dado os objetivos do jogo, podendo ser utilizado para aprimorar a bonificação dos jogadores. No que se refere à contribuição pedagógica, desenvolver o recurso proposto pelo presente trabalho dentro do jogo, intensifica os objetivos educacionais do jogo do Furbot.

Comentado [DSdR3]: Arrumar o alinhamento do recuo do parágrafo.

Comentado [DSdR4]: Arrumar o alinhamento do recuo do parágrafo.

Comentado [DSdR5]: Arrumar o alinhamento do recuo do parágrafo.

Comentado [DSdR6]: Arrumar o alinhamento do recuo do parágrafo.

4.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A aplicação descrita nesse trabalho deverá:

- encontrar a melhor resposta para cada fase do jogo Furbot (Requisito Funcional - RF);
- gerar um gabarito com a solução ótima contendo os comandos sequenciais que o robô deve fazer para sair do ponto de origem e chegar até o ponto de destino (RF);
- comparar a resolução do aluno com a gerada pelo algoritmo (RF);
- ser implementado na linguagem Python (Requisito Não Funcional - RNF);
- ser implementado no ambiente de desenvolvimento Visual Studio (RNF).

4.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- levantamento bibliográfico: levantar material bibliográfico pertinente a utilização de ensino de lógica e programação no ensino, pensamento computacional, aprendizagem de máquina e demais temas que forem necessários, além de aprimorar os trabalhos correlatos;
- elicitação de requisitos: revisar o conjunto de requisitos funcionais e não funcionais previamente identificados nesta proposta;
- especificação: formalizar as funcionalidades do projeto através de diagramas de casos de uso, de classe, de atividade e de componentes da Unified Modeling Language (UML) utilizando a ferramenta LucidChart;
- implementação do algoritmo de aprendizagem: implementar e testar os principais algoritmos de

aprendizagem de máquina na linguagem Python no ambiente de desenvolvimento Visual Studio.

- e) integração com o Furbot: integrar o algoritmo Python com a arquitetura do Furbot utilizando a linguagem C#;
- f) testes e validações: serão realizados testes juntamente com a construção dos algoritmos comparando as soluções geradas pelos algoritmos com a dos alunos e com a solução desenvolvida por Xavier (2019), medindo a eficiência das soluções.

As etapas de implementação e testes serão desenvolvidas de forma integrada, de modo que haja uma evolução contínua em relação ao desempenho na obtenção do melhor caminho, até se identificar o melhor algoritmo a ser utilizado. As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

etapas / quinzenas	2022							
	fev.		mar.		abr.		mai.	
	1	2	1	2	1	2	1	2
levantamento bibliográfico								
elicitação de requisitos								
especificação								
implementação do algoritmo de aprendizagem								
integração com o Furbot								
testes e validações								

Comentado [DSdR7]: maio

Fonte: elaborado pelo autor.

Comentado [DSdR8]: Diminuir espaço entre quadro e fonte.

5 REVISÃO BIBLIOGRÁFICA

Este capítulo está dividido em duas seções. A seção 5.1 aborda uma visão geral sobre a computação na educação básica. Na seção 5.2 são descritos os fundamentos principais sobre aprendizagem de máquina e métodos de aprendizagem supervisionados.

5.1 COMPUTAÇÃO NA EDUCAÇÃO BÁSICA

A inclusão de conceitos gerais de computação no ensino básico faz-se essencial como forma de aprimorar o raciocínio lógico-matemático das crianças e incentivar a criatividade por meio de tecnologias (NASCIMENTO *et al.*, 2015). Ainda, seguindo este contexto, Scaico *et al.* (2013) complementam que o ensino de conceitos de programação proporciona o desenvolvimento de várias habilidades que colaboram para aprimorar o raciocínio lógico dos estudantes. Programar abrange a capacidade de desenvolver uma solução para um problema, que caso seja grande demandará o exercício de outras habilidades (SCAICO *et al.*, 2013).

Fundamentos da computação, agregados ao pensamento crítico definem um método para solucionar problemas intitulado Pensamento Computacional (WING, 2006). Esta perspectiva argumenta que a cultura do computador ajuda a sociedade a aprender, mas especialmente, possibilita uma nova forma de aprender a aprender (LU; FLETCHER, 2009). O Pensamento Computacional é um método para solução de problemas que utiliza os princípios e técnicas da Ciência da Computação.

Segundo Ribeiro, Foss e Cavalheiro (2017), as três habilidades básicas que fundamentam o Pensamento Computacional são: abstração, automação e análise. A habilidade de abstração se associa a aptidão do indivíduo extrair somente os aspectos mais relevantes de um problema para chegar a uma solução. A habilidade de automação associa-se a aptidão de utilização um meio eletrônico que é capaz de substituir o trabalho de um ser humano e a habilidade de análise associa-se ao estudo dos resultados criados pela automação.

Como Brackmann (2017) afirma, enquanto os alunos aprendem a programar, estão também programando para aprender. Este aprendizado possibilita que eles conheçam várias outras coisas e criem oportunidades de aprendizagem.

5.2 APRENDIZAGEM DE MÁQUINA E MÉTODOS

Como afirma Bishop (2006), Sistemas de Aprendizado de Máquina são sistemas que combinam algoritmos e modelos matemáticos da área de Aprendizado de Máquina em soluções que visam a implantação do uso de tais algoritmos e modelos em ambientes reais. Aprendizado de máquina, por sua vez, se refere a área de estudos voltada ao desenvolvimento e compreensão de modelos matemáticos caracterizados pelo aprimoramento de seus resultados por meio da ingestão de dados de treino, de forma que esses modelos possam realizar predições e decisões sem que sejam explicitamente programados para o fazerem.

A aprendizagem de máquina é dividida em três tipos de técnicas: a aprendizagem supervisionada, que treina um modelo a partir de dados conhecidos de entrada e saída para que ele possa prever resultados futuros; a aprendizagem não supervisionada, que encontra padrões ocultos ou estruturas específicas a partir de dados de entrada; e a aprendizagem por reforço, que utiliza um sistema de recompensas ou punições para se aperfeiçoar no ambiente que se encontra (NORVIG, RUSSELL, 2014).

Na aprendizagem supervisionada, Norvig e Russel (2014) descrevem que o agente analisa determinados exemplos de entrada e saída correspondentes, e aprende uma função que faz o mapeamento da entrada para a saída. Exemplificando, o agente pode aprender a reconhecer fotos de ônibus ao ver muitas imagens que lhe mostram ônibus.

Na aprendizagem não supervisionada, Norvig e Russel (2014) explicam que o agente estuda padrões na entrada, ainda que nenhum feedback explícito seja fornecido. A tarefa mais utilizada na aprendizagem não supervisionada é a detecção e agrupamento de exemplos de entrada potencialmente úteis. Exemplificando, um agente de táxi pode desenvolver gradualmente um conceito de “dia de tráfego bom” e “dia de tráfego ruim” sem necessariamente terem sido rotulados anteriormente.

Por fim, na aprendizagem por reforço, o agente aprende com base em uma sucessão de reforços, recompensas ou punições. Exemplificando, um agente de táxi entende que algo saiu errado a partir da falta de gorjeta ao final de uma corrida (NORVIG, RUSSELL, 2014).

REFERÊNCIAS

- ARAÚJO, Luciana; MATTOS, Mauro. **Furbot**: plataforma para o ensino-aprendizado de pensamento computacional e programação. In: I CONCURSO LATINOAMERICANO DE TECNOLOGIAS EDUCACIONAIS PARA APRENDIZAGEM (EDUTECH), 1., 2018, São Paulo. 1p.
- BISHOP, C. M., **Pattern Recognition and Machine Learning**, Springer, v. ISBN 978-0-387-31073-2, 2006.
- BRACKMANN, C. P., **Desenvolvimento do Pensamento Computacional Através de Atividades Desplugadas na Educação Básica**. 2017. 226 f. Tese (Doutorado) - Curso de Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BURKE, B. **Gamificar**: como a gamificação motiva as pessoas a fazerem coisas extraordinárias. São Paulo: DVS Editora, 2015.
- CODE.ORG. **Minecraft: Jornada do Herói**, [s.l.], 2021a. Disponível em: <https://studio.code.org/s/hero/lessons/1/levels/1>. Acesso em: 27 mai. 2021.
- CODE.ORG. **About Us**, [s.l.] 2021b. Disponível em: <https://code.org/about>. Acesso em: 27 mai. 2021.
- CODESPARK. **About Us**, [s.l.], 2021a. Disponível em: <https://codespark.com/about>. Acesso em: 25 mai. 2021.

CODESPARK. **Frequently Asked Questions**, [s.l.] 2021b. Disponível em: <https://codespark.com/frequently-asked-questions>. Acesso em: 25 mai. 2021.

Lu, J. J.; Fletcher, G. H. (2009). Thinking about computational thinking. In Proc. 40th Technical Symp. on Comp. Sci. Education, pages 260–264, New York, USA. ACM.

MATTOS, Mauro et al. **Furbot Móvel**: um jogo para o ensino do pensamento computacional. In: VIII CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 2019, Brasília. Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação (CBIE 2019). [S.L.]: Brazilian Computer Society (Sociedade Brasileira de Computação - Sbc), 2019. p. 1294-1301.

Nascimento, J., Xavier, D., Passos, O. e Barreto, R., (2015) **Um Relato de Experiência da Aplicação de Técnicas Interativas para Ensino da Computação na Educação Básica**. Anais do Encontro Regional de Computação e Sistemas de Informação. p. 95-104.

NORVIG, Peter; RUSSELL, Stuart. **Inteligência Artificial**: Tradução da 3a Edição. Elsevier Brasil, 2014.

RAABE, André L. A. *et al.* **Referenciais de Formação em Computação**: Educação Básica. Porto Alegre. Assembleias do WEI e da SBC. 2017. 9 f. Disponível em: <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf> Acesso em: 27 mai. 2021.

RIBEIRO, L.; FOSS, L.; CAVALHEIRO, S. A. da C. **Entendendo o Pensamento Computacional**. Jul 2017. Disponível em: <https://arxiv.org/abs/1707.00338>. Acesso em: 01 jun. 2021.

SCAICO, Pasqueline D. *et al.* **Ensino de Programação no Ensino Médio**: Uma Abordagem Orientada ao Design com a linguagem Scratch. Revista Brasileira de Informática na Educação, Porto Alegre, v. 21, n. 2, p. 92, set. 2013.

Schlögl, Lucas Eduardo; Oliveira, Gabriel Castellani de; Giovannella, Gian Carlo; Bizon, A. R.; Santos, B. F. F.; Kruger, N.; Bursoni, P.; Neumann, C. B.; Huber, E. E.; Araújo, Luciana P.; Mattos, Mauro M.; Zucco, F. D.; Cunha, K. Z.; Hein, N. **Ensino do Pensamento Computacional na Educação Básica**. Revista de Sistemas e Computação - RSC, v. 7, p. 304-322, 2017.

Vahldick, Adilson; Mattos, Mauro Marcelo. Relato de uma Experiência no Ensino de Algoritmos e Programação Utilizando um Framework Lúdico In: Simpósio Brasileiro de Informática na Educação, 2008, Fortaleza. XIX Simpósio Brasileiro de Informática na Educação. 2008.

Wing, J. M. (2006). **Computational thinking**. Communications of the ACM, v. 49, n. 3, p. 33–35.

XAVIER, Francisca Edyr. **Geração Automatizada de Gabarito e Correção de Exercícios em Ambiente Furbot Para o Ensino do Pensamento Computacional**. 2019. 26 f. TCC (Graduação) - Curso de Ciências da Computação, Universidade Regional de Blumenau, Blumenau, 2019. Disponível em: http://dsc.inf.furb.br/arquivos/tccs/monografias/2019_1_francisca-edyr-xavier_monografia.pdf. Acesso em: 01 maio 2021.

ZEFERINO, A. M. B.; DOMINGUES, R. C. L.; AMARAL, E. **Feedback como estratégia de aprendizado no ensino médico**. Rev. bras. educ. med. [online]. 2007, vol.31, n.2, pp.176-179. Disponível em: <https://www.scielo.br/j/rbem/a/yK7SFyqJBCm6h6RqNk4Szyt/abstract/?lang=pt>. Acesso em: 05 jun 2021.

FORMULÁRIO DE AVALIAÇÃO BCC – PROFESSOR TCC I

Avaliador(a): Dalton Solano dos Reis

ASPECTOS AVALIADOS		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. JUSTIFICATIVA São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	4. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
ASPECTOS METODOLÓGICOS	Os métodos, recursos e o cronograma estão devidamente apresentados?	X		
	5. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?	X		
	6. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		
	7. ORGANIZAÇÃO E APRESENTAÇÃO GRÁFICA DO TEXTO A organização e apresentação dos capítulos, seções, subseções e parágrafos estão de acordo com o modelo estabelecido?		X	
	8. ILUSTRAÇÕES (figuras, quadros, tabelas) As ilustrações são legíveis e obedecem às normas da ABNT?	X		
	9. REFERÊNCIAS E CITAÇÕES As referências obedecem às normas da ABNT?	X		
	As citações obedecem às normas da ABNT?	X		
	Todos os documentos citados foram referenciados e vice-versa, isto é, as citações e referências são consistentes?	X		

O projeto de TCC será reprovado se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos 4 (quatro) itens dos **ASPECTOS TÉCNICOS** tiverem resposta ATENDE PARCIALMENTE; ou
- pelo menos 4 (quatro) itens dos **ASPECTOS METODOLÓGICOS** tiverem resposta ATENDE PARCIALMENTE.

PARECER: (X) APROVADO () REPROVADO

Revisão do Pré-projeto

Disciplina: Trabalho de Conclusão de Curso I – BCC

Caro orientando,

segue abaixo o Termo de Compromisso, as DUAS revisões do seu pré-projeto contendo a avaliação do professor “avaliador” e professor “TCC1”, junto com as avaliações da defesa na banca de qualificação. É muito importante que revise com cuidado e discuta possíveis dúvidas decorrente das revisões com o seu professor orientador, e com o professor de TCC1. Sempre procure fazer todos os ajustes solicitados, até mesmo o menores detalhes, pois todos são importantes e irão refletir na sua nota nesta disciplina.

Mas, caso o professor orientador julgue que algumas anotações das revisões não devam ser feitas, ou mesmo que sejam feitas de forma diferente a solicitada pelo revisor, anexe ao final do seu projeto a ficha “Projeto: Observações – Professor Orientador” disponível no material da disciplina, e justifique o motivo.

Lembrem que agora o limite de páginas do projeto é no máximo 12 (doze) páginas. E que a seção de “Revisão Bibliográfica” deve ser complementada.

Atenciosamente,

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

TERMO DE COMPROMISSO

I – IDENTIFICAÇÃO DO ALUNO	
Nome:	GIAN CARLO GIOVANELLA
CV Lattes:	http://lattes.cnpq.br/6327953635208241
E-mail:	giancarlo16@gmail.com
Telefone:	(47) 9 99242303
II – IDENTIFICAÇÃO DO TRABALHO	
Título provisório:	GERAÇÃO AUTOMATIZADA DE RESOLUÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT UTILIZANDO MÉTODOS DE APRENDIZAGEM DE MÁQUINA
Orientador:	Luciana Pereira de Araújo Kohler
Coorientador (se houver):	Andreza Sartori
Linha de Pesquisa:	<input checked="" type="checkbox"/> Tecnologias aplicadas à informática na educação <input type="checkbox"/> Tecnologias aplicadas ao desenvolvimento de sistemas
III – COMPROMISSO DE REALIZAÇÃO DO TCC	
Eu (aluno),	Gian Carlo Giovanella
comprometo-me a realizar o trabalho proposto no semestre 2022/1, de acordo com as normas e os prazos determinados pela FURB, conforme previsto na resolução nº.20/2016.	
Assinatura:	NÃO É NECESSÁRIO – Encaminhar por mail ao orientador
IV – COMPROMISSO DE ORIENTAÇÃO	
Eu (orientador),	Luciana Pereira de Araújo Kohler
comprometo-me a orientar o trabalho proposto no semestre 2022/1, de acordo com as normas e os prazos determinados pela FURB, conforme previsto na resolução nº.20/2016.	
Assinatura:	NÃO É NECESSÁRIO – Encaminhar por mail ao professor de TCC I

Blumenau, 18 de 08 de 2021

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
(X) PRÉ-PROJETO	() PROJETO	ANO/SEMESTRE: 2021/2

GERAÇÃO AUTOMATIZADA DE RESOLUÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT UTILIZANDO MÉTODOS DE APRENDIZAGEM DE MÁQUINA

Gian Carlo Giovanella

Profª Luciana Pereira de Araújo Kohler – Orientadora

Profª. Andreza Sartori – Coorientadora

1 INTRODUÇÃO

A Sociedade Brasileira de Computação (SBC) considera primordial que assuntos associados à Computação sejam inseridos na Educação Básica, sendo de grande relevância para a inclusão dos indivíduos na sociedade atual (RAABE *et al.*, 2017). Pensando em viabilizar a introdução aos conceitos básicos da área, atividades lúdicas amparadas por jogos eletrônicos são utilizadas para divertir e expandir a possibilidade de aprendizagem de conceitos, conteúdos e habilidades embutidas nesses jogos, motivando os alunos e auxiliando na assimilação dos conteúdos de uma forma envolvente e criativa (BURKE, 2015).

Nesse contexto, tem-se o jogo Furbot cuja finalidade é auxiliar no desenvolvimento do pensamento computacional em crianças de ensino fundamental (ARAÚJO; MATTOS, 2018). O jogo é dividido em várias fases, nas quais os jogadores programam um robô utilizando comandos iniciais de programação (MATTOS, 2019). O problema que se apresenta neste momento é que não há um feedback ao aluno sobre a qualidade de sua solução. Com isto, o aluno não sabe se haveria alguma forma mais eficiente de resolver os exercícios. Segundo Zeferino, Domingues e Amaral (2007, p. 176), o feedback é fundamental no processo de aprendizagem, pois ajuda os alunos a identificar áreas de aprimoramento de seus conhecimentos ou habilidades, refletindo assim, em uma maior performance no processo de aprendizagem.

Nesse contexto, essa proposta tem como objetivo fornecer uma resolução de forma automatizada para cada fase do Furbot, que poderá ser utilizada para fornecer um feedback ao jogador quando este a concluir. Para a automatização deste processo serão utilizados métodos de aprendizagem de máquina produzindo a solução ótima de acordo com os objetivos do mapa de cada fase. Considera-se uma solução ótima, a solução que apresenta o menor caminho entre a origem e o destino.

1.1 OBJETIVOS

O objetivo deste trabalho é propor uma solução de automatização da resolução das fases do jogo Furbot, que encontre a resolução mais otimizada para cada fase.

Os objetivos específicos são:

- a) gerar uma solução ótima para cada fase do jogo Furbot, sendo esta, baseada no caminho mínimo entre a origem e os objetivos da fase;
- b) fornecer feedback para o jogador comparando sua solução com a solução ótima gerada;
- c) integrar a solução gerada no jogo existente do Furbot.

Excluído: que tem a

Excluído: de

Neste capítulo serão apresentados os trabalhos que possuem características semelhantes às dos objetivos propostos nesse trabalho. O primeiro aborda o trabalho de Xavier (2019) que é a geração automatizada de gabarito e correção de exercícios em ambiente Furbot para o ensino do pensamento computacional. O segundo é uma ferramenta para ensino introdutório de programação (CODE.ORG, 2021a). Por fim, o terceiro trabalho é um jogo de quebra cabeça chamado codeSpark (CODESPARK, 2021a).

O trabalho desenvolvido por Xavier (2019) teve como objetivo gerar gabaritos para as fases do projeto Furbot. Este projeto é desenvolvido em Java para o ensino do pensamento computacional voltado às crianças da rede básica de educação (SCHLÖGL et al., 2017).

As tecnologias utilizadas no desenvolvimento incluem o *framework* Furbot na linguagem Java, além da biblioteca Commons-digester-1.8 para leitura de arquivos XMLs e o algoritmo de Dijkstra utilizado para encontrar os caminhos mínimos entre os objetivos do mapa (XAVIER, 2019).

Figura 1 – Tela de comparação resolução com o gabarito



Fonte: Xavier, 2019.

Por meio de testes em campo realizado com alunos de ensino fundamental, Xavier (2019) pode observar como resultado uma maior motivação entre os alunos que passaram a comparar suas soluções com o gabarito gerado. Ainda, Xavier (2019) aponta que há uma demora para gerar o gabarito ao colocar muitos objetivos no mapa, sendo esta uma limitação do trabalho.

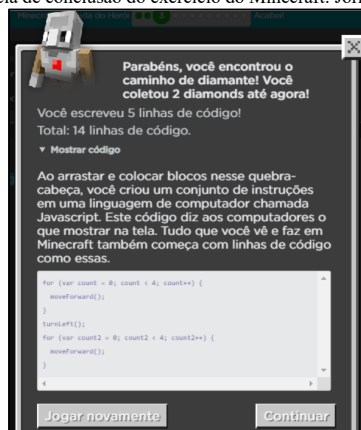
Excluído: na geração

Excluído: para gerar o gabarito

A programação com Minecraft: Jornada do Herói é uma das várias ferramentas de ensino de programação desenvolvidas pela Code.org (2021a). A ferramenta introduz diversos conceitos de programação, tais como

A programação com Minecraft: Jornada do Herói é uma das várias ferramentas de ensino de programação desenvolvidas pela Code.org (2021a). A ferramenta introduz diversos conceitos de programação, tais como

Figura 3 – Tela de conclusão do exercício do Minecraft: Jornada do Herói.



Fonte: elaborado pelo autor.

2.3 CODESPARK

CodeSpark é uma plataforma de minijogos online que pode ser acessada pelo navegador ou por um aplicativo (CODESPARK, 2021b). No CodeSpark o jogador deve programar os personagens clicando e arrastando comandos em sequência. Após finalizar a escolha e ordem dos comandos, necessita clicar no personagem para que ele execute as ações escolhidas, como pode ser observado na Figura 4.

Figura 4 – Ambiente do Jogo codeSpark



Fonte: elaborado pelo autor.

O cenário é formado por vários objetos comuns do dia a dia. O ambiente apresentado ao jogador varia de acordo com o quebra cabeça e sempre possui símbolos como os que se encontram na parte inferior da Figura 4, que se traduzem em funções tais como: seguir em frente, virar para esquerda, virar para direita ou pular em alguma direção.

O jogo proporciona a prática de vários conceitos de programação sendo eles: programação sequencial; lógica condicional e uso de funções. Uma vez executado, é possível visualizar a execução dos comandos inseridos. Para concluir o objetivo é necessário coletar todas as esmeraldas do mapa.

A plataforma não verifica o menor caminho entre o objetivo de cada fase, apenas verifica quantas esmeraldas foram coletadas e gera uma pontuação a partir disso. O número de comandos que o usuário pode utilizar em cada fase é limitado, induzindo-o a utilizar os comandos na sequência que é pré-determinada.

3 FERRAMENTA ATUAL

O Furbot teve sua origem em 2008 com o propósito de auxiliar o ensino dos conceitos iniciais de programação orientada a objetos, voltado aos alunos de graduação do departamento de Sistemas e Computação da FURB, por meio de um *framework* Java (VAHLICK; MATTOS, 2008). Em 2017, o *framework* foi utilizado para a concepção de um ambiente de programação com o objetivo de introduzir o pensamento computacional em crianças de ensino fundamental, sem a necessidade da instalação de um ambiente de programação ou destas conhecerem uma linguagem de programação (SCHLÖGL, 2017). Por sua vez, em 2018, o Furbot passou por uma reformulação se transformando em uma plataforma composta por um jogo e um servidor (MATTOS, 2019). O jogo, desenvolvido em Unity na linguagem C#, possui um apelo mais lúdico e um gráfico mais atrativo para o público-alvo, que são as crianças de ensino fundamental, em relação ao ambiente desenvolvido em 2017 (MATTOS, 2019). Já o servidor que acompanha a plataforma é utilizado para manter os dados dos jogadores de modo que os professores possam acompanhar suas turmas e visualizar os exercícios resolvidos pelos seus alunos (MATTOS, 2019). Este servidor é desenvolvido em Java utilizando o *framework* Spring. Como o foco deste trabalho é o jogo do Furbot, este é explicado com maiores detalhes.

O jogo possui uma narrativa que tem como personagens a Professora Sam que cria o Furbot. O Furbot é um robô que os alunos precisam programar para salvar o mundo do ataque dos malvados *buggiens*. A Figura 5 apresenta uma visão da tela de uma das fases iniciais. Nessa tela, tem-se o painel de comandos, localizado à direita da figura, que permite que o aluno desenvolva os comandos necessários para movimentar o Furbot no cenário da fase.



Fonte: elaborado pelo autor.

Cada fase possui desafios próprios com o objetivo de chegar ao final do caminho coletando tesouros, evitando os *buggiens* e desviando dos obstáculos, para então continuar progredindo para as próximas fases. O jogo introduz os conceitos de programação sequencial, utilização de condicionais e o uso de laços de repetição.

Ao finalizar a fase, é exibida uma tela mostrando a pontuação do aluno. Nela, o aluno pode retornar ao jogo e tentar fazer a fase novamente a fim de aumentar sua pontuação ou seguir em frente para a próxima fase. Contudo, o aluno não tem o feedback se sua solução é a melhor possível ou não.

4 PROPOSTA

Neste capítulo serão apresentadas as justificativas para o desenvolvimento deste trabalho. Além disso, serão apresentados os principais Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF), bem como suas características e a metodologia de desenvolvimento que será utilizada.

4.1 JUSTIFICATIVA

No Quadro 3 é apresentada uma comparação entre os trabalhos correlatos. Nas colunas são identificados os trabalhos e nas linhas as características.

Quadro 3 – Comparativo entre os trabalhos correlatos

	Geração automatizada de gabarito (XAVIER, 2019)	Minecraft: Jornada do Herói (CODE.ORG, 2018b)	CodeSpark (CODESPARK, 2021b)
Objetivo educacional	Sim	Sim	Sim
Conferência automatizada do resultado da resolução do exercício	Sim	Sim	Sim
Verifica se a resolução é a melhor possível	Sim	Não	Não
Plataforma	Web	Web	Web/App

Fonte: elaborado pelo autor.

Como pode ser observado no Quadro 3, todos os trabalhos são ferramentas que foram construídas para fins educacionais e verificam de forma automatizada se o objetivo foi concluído. Tanto o CodeSpark (CODESPARK, 2021a), quanto o Minecraft: Jornada do Herói (CODE.ORG, 2018B) não demonstram ao aluno se existe alguma solução melhor do que a desenvolvida por ele. Em relação ao trabalho de Xavier (2019), é utilizado o algoritmo de busca de menor caminho Dijkstra para solucionar e encontrar a solução ótima dos exercícios. Desta forma, há uma referência que pode ser utilizada pelos alunos para chegar na melhor solução.

Analisando os correlatos apresentados, o único trabalho que se preocupa em gerar uma solução ótima é o trabalho de Xavier (2019). Contudo, o autor concluiu que essa solução não possui bom desempenho, possuindo um elevado tempo de execução, o qual o autor atribuiu ao elevado número de objetivos e obstáculos que o jogo possui. Além disso, esse tipo de algoritmo não é suficiente para a versão atual do Furbot, pois além destes itens, ainda deve ser considerado os tipos de terreno, itens coletados no mapa, gasto de energia e inimigos encontrados.

Em relação à aplicação a ser desenvolvida, podem existir várias formas de resolver um mesmo exercício no Furbot. Criar um processo automatizado para verificar a melhor solução a partir dos objetivos da fase, proporciona uma referência que os alunos podem seguir para se desafiar a aprimorar suas soluções. Para gerar a melhor solução, um estudo acerca dos principais algoritmos utilizados em aprendizagem supervisionada será realizado, observando as revisões bibliográficas mais recentes e dessa forma, será definido o que mais se encaixa no contexto dos objetivos propostos.

Tendo em vista os pontos levantados, torna-se importante para o desenvolvimento social a tarefa de instigar e inspirar os alunos a encontrarem as melhores soluções para os desafios propostos. Com isso, é possível contribuir na formação deles e expandir o ensino de pensamento computacional desde cedo nas escolas.

Em relação à contribuição tecnológica, o trabalho proposto pode verificar a viabilidade dos métodos de aprendizagem de máquina supervisionados no contexto de automatizar a busca pelos melhores resultados dado os objetivos do jogo, podendo ser utilizado para aprimorar a bonificação dos jogadores e aumentar a competitividade entre eles. No que se refere à contribuição pedagógica, desenvolver o recurso proposto pelo presente trabalho dentro do jogo, intensifica os objetivos educacionais do jogo do Furbot.

4.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A aplicação descrita nesse trabalho deverá:

- encontrar a melhor resposta para cada fase do jogo Furbot (Requisito Funcional - RF);

Excluído: à cerca

Comentado [MH9]: Não entendi o que seria “desenvolvimento social”.

Comentado [MH10]: Estamos falando de educação de crianças, então cuidado na busca pela melhor resposta ou pela resposta correta. Entendo que aqui deve se utilizar uma abordagem do tipo “erro no processo de ensino-aprendizagem” ou “construção da aprendizagem a partir do erro”. Apesar de que a criança encontrou uma solução, ou seja, ela não errou, Mas pode refletir na otimização de sua solução.

Excluído: encontrar os

Comentado [MH11]: Em que contexto a competitividade é benéfica para a aprendizagem?

- b) gerar um gabarito com a solução ótima contendo os comandos sequenciais que o robô deve fazer para sair do ponto de origem e chegar até o ponto de destino (RF);
- c) comparar a resolução do aluno com a gerada pelo algoritmo (RF);
- d) ser implementado na linguagem C# (Requisito Não Funcional - RNF);
- e) ser implementado no ambiente de desenvolvimento Visual Studio (RNF);

Excluído:

4.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: levantar material bibliográfico pertinente a utilização de ensino de lógica e programação no ensino, pensamento computacional, aprendizagem supervisionada e demais temas que forem necessários, além de aprimorar os trabalhos correlatos;
- b) eliciação de requisitos: revisar o conjunto de requisitos funcionais e não funcionais previamente identificados nesta proposta;
- c) especificação: formalizar as funcionalidades do projeto através de diagramas de casos de uso, de classe, de atividade e de componentes da Unified Modeling Language (UML) utilizando a ferramenta LucidChart;
- d) implementação: implementar algoritmos para geração da melhor solução, utilizando a linguagem C# no ambiente de desenvolvimento Visual Studio;
- e) testes: elaborar e efetuar testes comparando a solução gerada pelo algoritmo com a dos alunos, medindo quantos passos de diferença existem.

Comentado [MH12]: Estas etapas estão muito tradicionais e muito sequenciais. Para os objetivos propostos e para os requisitos explicitados entendo que é preciso iterar ou subdividir as etapas.

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

etapas / quinzenas	2021							
	ago.		set.		out.		nov.	
	1	2	1	2	1	2	1	2
levantamento bibliográfico								
eliciação de requisitos								
especificação								
implementação								
testes								

Fonte: elaborado pelo autor.

Comentado [MH13]: Entendo que a comparação da solução ótima encontrada pelo algoritmo deveria ser comparada com a solução ótima encontrada pelo trabalho de Xavier, incluindo um comparativo de tempo. Desta forma você poderia afirmar que seu trabalho resolveu (ou não) a limitação apresentada por Xavier. Sobre a comparação com a solução dos alunos, penso que é um desenvolvimento a ser realizado (inclusive está explicitado como RF) e não apenas um elemento de teste.

5 REVISÃO BIBLIOGRÁFICA

Este capítulo está dividido em duas seções. A seção 5.1 aborda uma visão geral sobre a computação na educação básica. Na seção 5.2 são descritos os fundamentos principais sobre aprendizagem de máquina e métodos de aprendizagem supervisionados.

5.1 COMPUTAÇÃO NA EDUCAÇÃO BÁSICA

A inclusão de conceitos gerais de computação no ensino básico faz-se essencial como forma de aprimorar o raciocínio lógico-matemático das crianças e incentivar a criatividade por meio de tecnologias (NASCIMENTO *et al.*, 2015). Ainda, seguindo este contexto, Scaico *et al.* (2013) complementam que o ensino de conceitos de programação proporciona o desenvolvimento de várias habilidades que colaboram para aprimorar o raciocínio lógico dos estudantes. Programar abrange a capacidade de desenvolver uma solução para um problema, que caso seja grande demandará o exercício de outras habilidades (SCAICO *et al.*, 2013).

Fundamentos da computação, agregados ao pensamento crítico, definem um método para solucionar problemas intitulado Pensamento Computacional (PC) (WING, 2006). Esta perspectiva argumenta que a cultura do computador ajuda a sociedade a aprender, mas especialmente, possibilita uma nova forma de aprender a aprender (LU; FLETCHER, 2009). O PC é um método para solução de problemas que utiliza os princípios e técnicas da Ciência da Computação.

Excluído: ,

Segundo Ribeiro, Foss e Cavalheiro (2017), as três habilidades básicas que fundamentam o PC são: abstração, automação e análise. A habilidade de abstração se associa a aptidão de o indivíduo extrair somente os aspectos mais relevantes de um problema para chegar a uma solução. A habilidade de automação associa-se a aptidão de utilização um meio eletrônico que é capaz de substituir o trabalho de um ser humano e a habilidade de análise associa-se ao estudo dos resultados criados pela automação.

Como Brackmann (2017) afirma, enquanto os alunos aprendem a programar, estão também programando para aprender. Este aprendizado possibilita que eles conheçam várias outras coisas e criem oportunidades de aprendizagem.

5.2 APRENDIZAGEM DE MÁQUINA E MÉTODOS

Como afirma Bishop (2006), Sistemas de Aprendizado de Máquina são sistemas que combinam algoritmos e modelos matemáticos da área de Aprendizado de Máquina em soluções que visam a implantação do uso de tais algoritmos e modelos em ambientes reais. Aprendizado de máquina, por sua vez, se refere a área de estudos voltada ao desenvolvimento e compreensão de modelos matemáticos caracterizados pelo aprimoramento de seus resultados por meio da ingestão de dados de treino, de forma que esses modelos possam realizar previsões e decisões sem que sejam explicitamente programados para o fazerem.

A aprendizagem de máquina é dividida em três tipos de técnicas: a aprendizagem supervisionada, que treina um modelo a partir de dados conhecidos de entrada e saída para que ele possa prever resultados futuros; a aprendizagem não supervisionada, que encontra padrões ocultos ou estruturas específicas a partir de dados de entrada; e a aprendizagem por reforço, que utiliza um sistema de recompensas ou punições para se aperfeiçoar no ambiente que se encontra (NORVIG, RUSSELL, 2014).

Como aponta Bueno (2018), o intuito da aprendizagem supervisionada é construir um modelo que faça previsões baseadas em evidências na presença de incerteza. Nesta técnica, um programa de computador pode “aprender” a partir de observações. Conforme exposto a mais observações, o programa melhora seu desempenho preditivo. Especificamente, um algoritmo de aprendizagem supervisionada recebe um conjunto conhecido de dados de entrada e um conjunto conhecido de respostas aos dados de entrada e treina um modelo para gerar previsões.

Comentado [MH14]: Por que só falou deste tipo de técnica? E os outros dois?

REFERÊNCIAS

- ARAÚJO, Luciana; MATTOS, Mauro. Furbot: plataforma para o ensino-aprendizado de pensamento computacional e programação. In: I CONCURSO LATINOAMERICANO DE TECNOLOGIAS EDUCACIONAIS PARA APRENDIZAGEM (EDUTECH), 1., 2018, São Paulo. 1p.
- BISHOP, C. M., “Pattern Recognition and Machine Learning”, Springer, v. ISBN 978-0-387-31073-2, 2006.
- BRACKMANN, Christian Puhlmann. **DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL ATRAVÉS DE ATIVIDADES DESPLUGADAS NA EDUCAÇÃO BÁSICA**. 2017. 226 f. Tese (Doutorado) - Curso de Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BUENO, Andre Luis Cavalcanti. **Relaxamento Adaptativo da Sincronização Através do Uso de Métodos de Aprendizagem Supervisionada**. 2018. 80 f. Tese (Doutorado) - Curso de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Rio de Janeiro, 2018.

BURKE, B. Gamificar: como a gamificação motiva as pessoas a fazerem coisas extraordinárias. São Paulo: DVS Editora, 2015.

CODE.ORG. **Minecraft: Jornada do Herói**, [s.l.], 2021a. Disponível em: <https://studio.code.org/s/hero/lessons/1/levels/1>. Acesso em: 27 mai. 2021.

_____. **About Us**, [s.l.] 2021b. Disponível em: <https://code.org/about>. Acesso em: 27 mai. 2021.

CODESPARK. **About Us**, [s.l.], 2021. Disponível em: <https://codespark.com/about>. Acesso em: 25 mai. 2021.

_____. **Frequently Asked Questions**, [s.l.] 2021b. Disponível em: <https://codespark.com/frequently-asked-questions>. Acesso em: 25 mai. 2021.

Lu, J. J. and Fletcher, G. H. (2009). Thinking about computational thinking. In Proc. 40th Technical Symp. on Comp. Sci. Education, pages 260–264, New York, USA. ACM.

MATTOS, Mauro et al. **Furbot Móvel: um jogo para o ensino do pensamento computacional**. In: VIII CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 2019, Brasília. Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação (CBIE 2019). [S.L.]: Brazilian Computer Society (Sociedade Brasileira de Computação - Sbc), 2019. p. 1294-1301

Nascimento, J., Xavier, D., Passos, O. e Barreto, R., (2015) “Um Relato de Experiência da Aplicação de Técnicas Interativas para Ensino da Computação na Educação Básica. ” Anais do Encontro Regional de Computação e Sistemas de Informação. p. 95-104

NORVIG, Peter; RUSSELL, Stuart. Inteligência Artificial: Tradução da 3a Edição. Elsevier Brasil, 2014.

RAABE, André L. A. *et al.* **Referenciais de Formação em Computação: Educação Básica**. Porto Alegre. Assembleias do WEI e da SBC. 2017. 9 f. Disponível em: <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf> Acesso em: 27 mai. 2021.

RIBEIRO, L.; FOSS, L.; CAVALHEIRO, S. A. da C. Entendendo o Pensamento Computacional. Jul. Disponível em: <https://arxiv.org/abs/1707.00338>. Acesso em: 01 jun. 2021

SCAICO, Pasqueline D. *et al.* Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch. Revista Brasileira de Informática na Educação, Porto Alegre, v. 21, n. 2, p. 92, set. 2013.

Schlögl, Lucas Eduardo; Oliveira, Gabriel Castellani de; Giovanella, Gian Carlo; Bizon, A. R.; Santos, B. F. F.; Kruger, N.; Bursoni, P.; Neumann, C. B.; Huber, E. E.; Araújo, Luciana P; Mattos, Mauro M.; Zucco, F. D.; Cunha, K. Z.; Hein, N. . Ensino do Pensamento Computacional na Educação Básica. Revista de Sistemas e Computação - RSC, v. 7, p. 304-322, 2017.

Vahldick, Adilson; Mattos, Mauro Marcelo. Relato de uma Experiência no Ensino de Algoritmos e Programação Utilizando um Framework Lúdico In: Simpósio Brasileiro de Informática na Educação, 2008, Fortaleza. XIX Simpósio Brasileiro de Informática na Educação. 2008.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, v. 49, n. 3, p.33–35.

XAVIER, Francisca Edyr. **GERAÇÃO AUTOMATIZADA DE GABARITO E CORREÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT PARA O ENSINO DO PENSAMENTO COMPUTACIONAL**. 2019. 26 f. TCC (Graduação) - Curso de Ciências da Computação, Universidade Regional de Blumenau, Blumenau, 2019. Disponível em: http://dsc.inf.furb.br/arquivos/tccs/monografias/2019_1_francisca-edyr-xavier_monografia.pdf. Acesso em: 01 maio 2021.

ZEFERINO, A. M. B.; DOMINGUES, R. C. L.; AMARAL, E. Feedback como estratégia de aprendizado no ensino médico. Rev. bras. educ. med. [online]. 2007, vol.31, n.2, pp.176-179. Disponível em: <https://www.scielo.br/j/rbem/a/yK7SFyqJBCm6h6RqNk4Szyt/abstract/?lang=pt>. Acesso em: 05 jun 2021.

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR AVALIADOR

Avaliador(a): **Marcel Hugo**

Atenção: quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.

ASPECTOS AVALIADOS ¹		Atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	X		
	4. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?	X		
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?		X	
	5. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?	X		
	6. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?		X	
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?		X	
	7. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?		X	
ASPECTOS METODOLÓGICOS	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	X		
	8. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		



UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
DISCIPLINA: TRABALHO DE CONCLUSÃO DE CURSO I
CURSO: CIÊNCIA DA COMPUTAÇÃO - BCC

ATA DA DEFESA: BANCA DO PRÉ-PROJETO

Venho, por meio deste, manifestar minha avaliação sobre a **apresentação** do Pré-Projeto de TCC

realizado pelo(a) acadêmico(a), Gian Carlo Giovanella no **SEGUNDO SEMESTRE DE 2021**, com o título GERAÇÃO AUTOMATIZADA DE RESOLUÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT UTILIZANDO MÉTODOS DE APRENDIZAGEM DE MÁQUINA , sob orientação do prof(a). Luciana Pereira de Araújo Kohler .

A referida apresentação obteve a seguinte nota:

Componente da Banca	Nota (de 0 a 10)
Professor(a) Avaliador(a): Marcel Hugo	7,5

ATENÇÃO. A nota acima se refere somente a apresentação do pré-projeto e vai ser repassada para o aluno (orientando). Favor preencher os campos acima e enviar por e-mail ao professor de TCC1 (dalton@furb.br). Não passar o arquivo com as anotações da revisão já enviado ao professor de TCC1 para o orientando e nem para o professor orientador. Após o professor de TCC1 receber esta ata preenchida, o professor de TCC1 vai disponibilizar para o orientando/orientador os arquivos com as revisões. Caso julgue necessário fazer mais alguma consideração relacionada ao pré-projeto ou a defesa, favor usar o espaço abaixo.

Observações da apresentação:

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
(X) PRÉ-PROJETO	() PROJETO	ANO/SEMESTRE: 2021/2

GERAÇÃO AUTOMATIZADA DE RESOLUÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT UTILIZANDO MÉTODOS DE APRENDIZAGEM DE MÁQUINA

Gian Carlo Giovanella

Profª Luciana Pereira de Araújo Kohler – Orientadora

Profª. Andreza Sartori – Coorientadora

1 INTRODUÇÃO

A Sociedade Brasileira de Computação (SBC) considera primordial que assuntos associados à Computação sejam inseridos na Educação Básica, sendo de grande relevância para a inclusão dos indivíduos na sociedade atual (RAABE *et al.*, 2017). Pensando em viabilizar a introdução aos conceitos básicos da área, atividades lúdicas amparadas por jogos eletrônicos são utilizadas para divertir e expandir a possibilidade de aprendizagem de conceitos, conteúdos e habilidades embutidas nesses jogos, motivando os alunos e auxiliando na assimilação dos conteúdos de uma forma envolvente e criativa (BURKE, 2015).

Nesse contexto, tem-se o jogo Furbot que tem a finalidade de auxiliar no desenvolvimento do pensamento computacional em crianças de ensino fundamental (ARAÚJO; MATTOS, 2018). O jogo é dividido em várias fases, nas quais os jogadores programam um robô utilizando comandos iniciais de programação (MATTOS, 2019). O problema que se apresenta neste momento é que não há um feedback ao aluno sobre a qualidade de sua solução. Com isto, o aluno não sabe se haveria alguma forma mais eficiente de resolver os exercícios. Segundo Zeferino, Domingues e Amaral (2007, p. 176), o feedback é fundamental no processo de aprendizagem, pois ajuda os alunos a identificar áreas de aprimoramento de seus conhecimentos ou habilidades, refletindo assim, em uma maior performance no processo de aprendizagem.

Nesse contexto, essa proposta tem como objetivo fornecer uma resolução de forma automatizada para cada fase do Furbot, que poderá ser utilizada para fornecer um feedback ao jogador quando este a conclui. Para a automatização deste processo serão utilizados métodos de aprendizagem de máquina produzindo a solução ótima de acordo com os objetivos do mapa de cada fase. Considera-se uma solução ótima, a solução que apresenta o menor caminho entre a origem e o destino.

1.1 OBJETIVOS

O objetivo deste trabalho é propor uma solução de automatização da resolução das fases do jogo Furbot, que encontre a resolução mais otimizada para cada fase.

Os objetivos específicos são:

- a) gerar uma solução ótima para cada fase do jogo Furbot, sendo esta, baseada no caminho mínimo entre a origem e os objetivos da fase;
- b) fornecer feedback para o jogador comparando sua solução com a solução ótima gerada;
- c) integrar a solução gerada no jogo existente do Furbot.

Comentado [DSdR15]: Alinhamento itens igual parágrafo.

2 TRABALHOS CORRELATOS

Neste capítulo serão apresentados os trabalhos que possuem características semelhantes às dos objetivos propostos nesse trabalho. O primeiro aborda o trabalho de Xavier (2019) que é a geração automatizada de gabarito e correção de exercícios em ambiente Furbot para o ensino do pensamento computacional. O segundo é uma ferramenta para ensino introdutório de programação (CODE.ORG, 2021a). Por fim, o terceiro trabalho é um jogo de quebra cabeça chamado codeSpark (CODESPARK, 2021a).

Comentado [DSdR16]: Nesta seção

Comentado [DSdR17]: CodeSpark

2.1 GERAÇÃO AUTOMATIZADA DE GABARITO E CORREÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT

O trabalho desenvolvido por Xavier (2019) teve como objetivo gerar gabaritos para as fases do projeto Furbot. Este projeto é desenvolvido em Java para o ensino do pensamento computacional voltado às crianças da rede básica de educação (SCHLÖGL et al., 2017).

Xavier (2019) utilizou algoritmos de menor caminho para solucionar problemas como: encontrar o menor caminho até os objetivos do robô; gerar um gabarito dos comandos sequenciais que o robô deve fazer para sair do ponto de origem e chegar até o ponto de destino (XAVIER, 2019). Além disso, o trabalho também teve como objetivo auxiliar os professores a fazer a correção dos exercícios criados pelos alunos, visto que muitos dos professores não possuem experiência na área de programação.

Comentado [DSdR18]: Itálico.

As tecnologias utilizadas no desenvolvimento incluem o *framework* Furbot na linguagem Java, além da biblioteca Commons-digester-1.8 para leitura de arquivos XMLs e o algoritmo de Dijkstra utilizado para encontrar os caminhos mínimos entre os objetivos do mapa (XAVIER, 2019).

Comentado [DSdR19]: Juntar com parágrafo anterior. Evitar parágrafos 1 frase.

Comentado [DSdR20]: arquivos eXtensible Markup Language (XML) e o

Na Figura 1 pode-se visualizar a comparação entre o código gerado pelo aluno e o gabarito fornecido pelo algoritmo de Xavier (2019). Ao concluir a fase, o aluno visualiza a quantidade de linhas que ele escreveu e a quantidade de linhas que poderia ter escrito. Ainda, é apresentada a pontuação com base nos objetivos do mapa.

Figura 1 – Tela de comparação resolução com o gabarito



Fonte: Xavier, 2019.

Por meio de testes em campo realizado com alunos de ensino fundamental, Xavier (2019) pode observar como resultado uma maior motivação entre os alunos que passaram a comparar suas soluções com o gabarito gerado. Ainda, Xavier (2019) aponta que há uma demora na geração ao colocar muitos objetivos no mapa para gerar o gabarito, sendo esta uma limitação do trabalho.

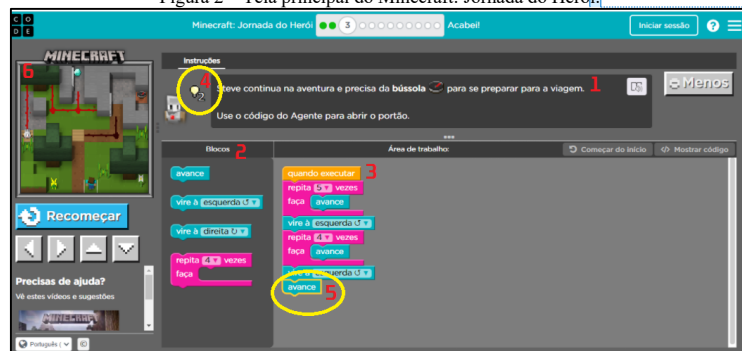
2.2 PROGRAMAÇÃO COM MINECRAFT: JORNADA DO HERÓI

A programação com Minecraft: Jornada do Herói é uma das várias ferramentas de ensino de programação desenvolvidas pela Code.org (2021a). A ferramenta introduz diversos conceitos de programação, tais como

programação sequencial, uso de laços de repetição e uso de funções pré-determinadas. A ferramenta propõe uma programação visual com o uso de blocos, sendo que seus comandos estão disponíveis em vários idiomas. O objetivo é usar os comandos disponíveis para realizar o que é proposto no início de cada fase.

A Figura 2 identifica a tela principal do jogo. Ela é dividida em alguns componentes, sendo eles: as instruções da fase (identificado pelo número 1 na figura); blocos para utilizar na resolução (identificado pelo número 2 na figura); área de trabalho que é onde o aluno deve incluir os blocos para montar o código para ser executado pelo personagem auxiliar (identificado pelo número 3 na figura). Ao clicar no botão executar, o personagem executa os comandos da área de trabalho, identificado pelo número 6 na figura, e ao finalizar, o botão de recomendar é exibido no lugar do botão executar. Caso haja um erro na sequência de passos, o personagem controlado pelo aluno não conseguirá concluir o problema. Assim, o aluno deverá pensar no que errou e recomendar ou pedir uma dica clicando na lâmpada situada no quadrado das instruções, indicado pelo item 4 na figura.

Figura 2 – Tela principal do Minecraft: Jornada do Herói.



Fonte: elaborado pelo autor.

Ao pedir para executar os passos, conforme as ações da área de trabalho são executadas, elas são destacadas em amarelo, como pode ser observado no último comando da Figura 2. Caso o usuário conclua o objetivo, além de uma mensagem de parabéns, é apresentado o código gerado na linguagem JavaScript, conforme a Figura 3. As tecnologias utilizadas para o desenvolvimento incluem JavaScript e Javascript XML (JSX) que permite escrever código HTML dentro do React.

Comentado [DSdR21]: Melhor trocar por letras A,B,C...

Comentado [DSdR22]: Remover ponto final.

Figura 3 – Tela de conclusão do exercício do Minecraft: Jornada do Herói.



Fonte: elaborado pelo autor.

2.3 CODESPARK

CodeSpark é uma plataforma de minijogos online que pode ser acessada pelo navegador ou por um aplicativo (CODESPARK, 2021b). No CodeSpark o jogador deve programar os personagens clicando e arrastando comandos em sequência. Após finalizar a escolha e ordem dos comandos, necessita clicar no personagem para que ele execute as ações escolhidas, como pode ser observado na Figura 4.

Figura 4 – Ambiente do Jogo C1 CodeSpark



Fonte: elaborado pelo autor.

O cenário é formado por vários objetos comuns do dia a dia. O ambiente apresentado ao jogador varia de acordo com o quebra cabeça e sempre possui símbolos como os que se encontram na parte inferior da Figura 4, que se traduzem em funções tais como: seguir em frente, virar para esquerda, virar para direita ou pular em alguma direção.

O jogo proporciona a prática de vários conceitos de programação sendo eles: programação sequencial; lógica condicional e uso de funções. Uma vez executado, é possível visualizar a execução dos comandos inseridos. Para concluir o objetivo é necessário coletar todas as esmeraldas do mapa.

A plataforma não verifica o menor caminho entre o objetivo de cada fase, apenas verifica quantas esmeraldas foram coletadas e gera uma pontuação a partir disso. O número de comandos que o usuário pode utilizar em cada fase é limitado, induzindo-o a utilizar os comandos na sequência que é pré-determinada.

Comentado [DSdR23]: Remover ponto final.

3 FERRAMENTA ATUAL

O Furbot teve sua origem em 2008 com o propósito de auxiliar o ensino dos conceitos iniciais de programação orientada a objetos, voltado aos alunos de graduação do departamento de Sistemas e Computação da FURB, por meio de um *framework* Java (VAHLICK; MATTOS, 2008). Em 2017, o *framework* foi utilizado para a concepção de um ambiente de programação com o objetivo de introduzir o pensamento computacional em crianças de ensino fundamental, sem a necessidade da instalação de um ambiente de programação ou destas conhecerem uma linguagem de programação (SCHLÖGL, 2017). Por sua vez, em 2018, o Furbot passou por uma reformulação se transformando em uma plataforma composta por um jogo e um servidor (MATTOS, 2019). O jogo, desenvolvido em Unity na linguagem C#, possui um apelo mais lúdico e um gráfico mais atrativo para o público-alvo, que são as crianças de ensino fundamental, em relação ao ambiente desenvolvido em 2017 (MATTOS, 2019). Já o servidor que acompanha a plataforma é utilizado para manter os dados dos jogadores de modo que os professores possam acompanhar suas turmas e visualizar os exercícios resolvidos pelos seus alunos (MATTOS, 2019). Este servidor é desenvolvido em Java utilizando o *framework* Spring. Como o foco deste trabalho é o jogo do Furbot, este é explicado com maiores detalhes.

O jogo possui uma narrativa que tem como personagens a Professora Sam que cria o Furbot. O Furbot é um robô que os alunos precisam programar para salvar o mundo do ataque dos malvados *buggiens*. A Figura 5 apresenta uma visão da tela de uma das fases iniciais. Nessa tela, tem-se o painel de comandos, localizado à direita da [figura](#), que permite que o aluno desenvolva os comandos necessários para movimentar o Furbot no cenário da fase.



Figura 5 – Tela de jogo do Furbot

Fonte: elaborado pelo autor.

Cada fase possui desafios próprios com o objetivo de chegar ao final do caminho coletando tesouros, evitando os *buggiens* e desviando dos obstáculos, para então continuar progredindo para as próximas fases. O jogo introduz os conceitos de programação sequencial, utilização de condicionais e o uso de laços de repetição.

Ao finalizar a fase, é exibida uma tela mostrando a pontuação do aluno. Nela, o aluno pode retornar ao jogo e tentar fazer a fase novamente a fim de aumentar sua pontuação ou seguir em frente para a próxima fase. Contudo, o aluno não tem o feedback se sua solução é a melhor possível ou não.

4 PROPOSTA

Neste capítulo serão apresentadas as justificativas para o desenvolvimento deste trabalho. Além disso, serão apresentados os principais Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF), bem como suas características e a metodologia de desenvolvimento que será utilizada.

Comentado [DSdR24]: SCHLÖGL et al., 2017 et al. em itálico.

Comentado [DSdR25]: Figura 5

Comentado [DSdR26]: Nesta seção

4.1 JUSTIFICATIVA

No Quadro 3 é apresentada uma comparação entre os trabalhos correlatos. Nas colunas são identificados os trabalhos e nas linhas as características.

Quadro 4 – Comparativo entre os trabalhos correlatos

	Geração automatizada de gabarito (XAVIER, 2019)	Minecraft: Jornada do Herói (CODE.ORG, 2018b)	CodeSpark (CODESPARK, 2021b)
Objetivo educacional	Sim	Sim	Sim
Conferência automatizada do resultado da resolução do exercício	Sim	Sim	Sim
Verifica se a resolução é a melhor possível	Sim	Não	Não
Plataforma	Web	Web	Web/App

Fonte: elaborado pelo autor.

Como pode ser observado no Quadro 3, todos os trabalhos são ferramentas que foram construídas para fins educacionais e verificam de forma automatizada se o objetivo foi concluído. Tanto o CodeSpark (CODESPARK, 2021a), quanto o Minecraft: Jornada do Herói (CODE.ORG, 2018b) não demonstram ao aluno se existe alguma solução melhor do que a desenvolvida por ele. Em relação ao trabalho de Xavier (2019), é utilizado o algoritmo de busca de menor caminho Dijkstra para solucionar e encontrar a solução ótima dos exercícios. Desta forma, há uma referência que pode ser utilizada pelos alunos para chegar na melhor solução.

Analisando os correlatos apresentados, o único trabalho que se preocupa em gerar uma solução ótima é o trabalho de Xavier (2019). Contudo, o autor concluiu que essa solução não possui bom desempenho, possuindo um elevado tempo de execução, o qual o autor atribuiu ao elevado número de objetivos e obstáculos que o jogo possui. Além disso, esse tipo de algoritmo não é suficiente para a versão atual do Furbot, pois além destes itens, ainda deve ser considerado os tipos de terreno, itens coletados no mapa, gasto de energia e inimigos encontrados.

Em relação à aplicação a ser desenvolvida, podem existir várias formas de resolver um mesmo exercício no Furbot. Criar um processo automatizado para verificar a melhor solução a partir dos objetivos da fase, proporciona uma referência que os alunos podem seguir para se desafiar a aprimorar suas soluções. Para gerar a melhor solução, um estudo à cerca dos principais algoritmos utilizados em aprendizagem supervisionada será realizado, observando as revisões bibliográficas mais recentes e dessa forma, será definido o que mais se encaixa no contexto dos objetivos propostos.

Tendo em vista os pontos levantados, torna-se importante para o desenvolvimento social a tarefa de instigar e inspirar os alunos a encontrarem as melhores soluções para os desafios propostos. Com isso, é possível contribuir na formação deles e expandir o ensino de pensamento computacional desde cedo nas escolas.

Em relação à contribuição tecnológica, o trabalho proposto pode verificar a viabilidade dos métodos de aprendizagem de máquina supervisionados no contexto de automatizar a encontrar os melhores resultados dado os objetivos do jogo, podendo ser utilizado para aprimorar a bonificação dos jogadores e aumentar a competitividade entre eles. No que se refere à contribuição pedagógica, desenvolver o recurso proposto pelo presente trabalho dentro do jogo, intensifica os objetivos educacionais do jogo do Furbot.

4.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A aplicação descrita nesse trabalho deverá:

- encontrar a melhor resposta para cada fase do jogo Furbot (Requisito Funcional - RF);

Comentado [DSdR27]: Centralizar.

Comentado [DSdR28]: b

Comentado [DSdR29]: Alinhamento.

Comentado [DSdR30]: Alinhamento.

Comentado [DSdR31]: acerca

Comentado [DSdR32]: Alinhamento.

Comentado [DSdR33]: Alinhamento.

- b) gerar um gabarito com a solução ótima contendo os comandos sequenciais que o robô deve fazer para sair do ponto de origem e chegar até o ponto de destino (RF);
- c) comparar a resolução do aluno com a gerada pelo algoritmo (RF);
- d) ser implementado na linguagem C# (Requisito Não Funcional - RNF);
- e) ser implementado no ambiente de desenvolvimento Visual Studio (RNF);

Comentado [DSdR34]: Ponto final.

4.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: levantar material bibliográfico pertinente a utilização de ensino de lógica e programação no ensino, pensamento computacional, aprendizagem supervisionada e demais temas que forem necessários, além de aprimorar os trabalhos correlatos;
- b) elicitação de requisitos: revisar o conjunto de requisitos funcionais e não funcionais previamente identificados nesta proposta;
- c) especificação: formalizar as funcionalidades do projeto através de diagramas de casos de uso, de classe, de atividade e de componentes da Unified Modeling Language (UML) utilizando a ferramenta LucidChart;
- d) implementação: implementar algoritmos para geração da melhor solução, utilizando a linguagem C# no ambiente de desenvolvimento Visual Studio;
- e) testes: elaborar e efetuar testes comparando a solução gerada pelo algoritmo com a dos alunos, medindo quantos passos de diferença existem.

As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 - Cronograma

	2021							
	ago.		set.		out.		nov.	
etapas / quinzenas	1	2	1	2	1	2	1	2
levantamento bibliográfico								
elicitação de requisitos								
especificação								
implementação								
testes								

Fonte: elaborado pelo autor.

Comentado [DSdR35]: Arrumar ano/meses. Semestre TCC2.

Comentado [DSdR36]: Centralizar.

5 REVISÃO BIBLIOGRÁFICA

Este capítulo está dividido em duas seções. A seção 5.1 aborda uma visão geral sobre a computação na educação básica. Na seção 5.2 são descritos os fundamentos principais sobre aprendizagem de máquina e métodos de aprendizagem supervisionados.

5.1 COMPUTAÇÃO NA EDUCAÇÃO BÁSICA

A inclusão de conceitos gerais de computação no ensino básico faz-se essencial como forma de aprimorar o raciocínio lógico-matemático das crianças e incentivar a criatividade por meio de tecnologias (NASCIMENTO *et al.*, 2015). Ainda, seguindo este contexto, Scaico *et al.* (2013) complementam que o ensino de conceitos de programação proporciona o desenvolvimento de várias habilidades que colaboram para aprimorar o raciocínio lógico dos estudantes. Programar abrange a capacidade de desenvolver uma solução para um problema, que caso seja grande demandará o exercício de outras habilidades (SCAICO *et al.*, 2013).

Fundamentos da computação, agregados ao pensamento crítico definem um método para solucionar problemas, intitulado Pensamento Computacional (PC) (WING, 2006). Esta perspectiva argumenta que a cultura do computador ajuda a sociedade a aprender, mas especialmente, possibilita uma nova forma de aprender a aprender (LU; FLETCHER, 2009). O PC é um método para solução de problemas que utiliza os princípios e técnicas da Ciência da Computação.

Segundo Ribeiro, Foss e Cavalheiro (2017), as três habilidades básicas que fundamentam o PC são: abstração, automação e análise. A habilidade de abstração se associa a aptidão de o indivíduo extrair somente os aspectos mais relevantes de um problema para chegar a uma solução. A habilidade de automação associa-se a aptidão de utilização um meio eletrônico que é capaz de substituir o trabalho de um ser humano e a habilidade de análise associa-se ao estudo dos resultados criados pela automação.

Como Brackmann (2017) afirma, enquanto os alunos aprendem a programar, estão também programando para aprender. Este aprendizado possibilita que eles conheçam várias outras coisas e criem oportunidades de aprendizagem.

5.2 APRENDIZAGEM DE MÁQUINA E MÉTODOS

Como afirma Bishop (2006), Sistemas de Aprendizado de Máquina são sistemas que combinam algoritmos e modelos matemáticos da área de Aprendizado de Máquina em soluções que visam a implantação do uso de tais algoritmos e modelos em ambientes reais. Aprendizado de máquina, por sua vez, se refere a área de estudos voltada ao desenvolvimento e compreensão de modelos matemáticos caracterizados pelo aprimoramento de seus resultados por meio da ingestão de dados de treino, de forma que esses modelos possam realizar previsões e decisões sem que sejam explicitamente programados para o fazerem.

A aprendizagem de máquina é dividida em três tipos de técnicas: a aprendizagem supervisionada, que treina um modelo a partir de dados conhecidos de entrada e saída para que ele possa prever resultados futuros; a aprendizagem não supervisionada, que encontra padrões ocultos ou estruturas específicas a partir de dados de entrada; e a aprendizagem por reforço, que utiliza um sistema de recompensas ou punições para se aperfeiçoar no ambiente que se encontra (NORVIG, RUSSELL, 2014).

Como aponta Bueno (2018), o intuito da aprendizagem supervisionada é construir um modelo que faça previsões baseadas em evidências na presença de incerteza. Nesta técnica, um programa de computador pode “aprender” a partir de observações. Conforme exposto a mais observações, o programa melhora seu desempenho preditivo. Especificamente, um algoritmo de aprendizagem supervisionada recebe um conjunto conhecido de dados de entrada e um conjunto conhecido de respostas aos dados de entrada e treina um modelo para gerar previsões.

REFERÊNCIAS

- ARAÚJO, Luciana; MATTOS, Mauro. Furbot: plataforma para o ensino-aprendizado de pensamento computacional e programação. In: I CONCURSO LATINOAMERICANO DE TECNOLOGIAS EDUCACIONAIS PARA APRENDIZAGEM (EDUTECH), 1., 2018, São Paulo. 1p.
- BISHOP, C. M., “Pattern Recognition and Machine Learning”, Springer, v. ISBN 978-0-387-31073-2, 2006.
- BRACKMANN, Christian Puhlmann. **DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL ATRAVÉS DE ATIVIDADES DESPLUGADAS NA EDUCAÇÃO BÁSICA**. 2017. 226 f. Tese (Doutorado) - Curso de Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BUENO, Andre Luis Cavalcanti. **Relaxamento Adaptativo da Sincronização Através do Uso de Métodos de Aprendizagem Supervisionada**. 2018. 80 f. Tese (Doutorado) - Curso de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Rio de Janeiro, 2018.

Comentado [DSdR37]: Remover vírgula

Comentado [DSdR38]: Remover.

Comentado [DSdR39]: Pensamento Computacional

Comentado [DSdR40]: Pensamento Computacional

Comentado [DSdR41]: do

Comentado [DSdR42]: Formato ABNT para eventos.

Comentado [DSdR43]: Remover aspas duplas. Negrito.

Comentado [DSdR44]: Abreviar.

Comentado [DSdR45]: Só as letras iniciais em maiúsculo.

Comentado [DSdR46]: Abreviar.

BURKE, B. **Gamificar**, como a gamificação motiva as pessoas a fazerem coisas extraordinárias. São Paulo: DVS Editora, 2015.

CODE.ORG. **Minecraft: Jornada do Herói**, [s.l.], 2021a. Disponível em: <https://studio.code.org/s/hero/lessons/1/levels/1>. Acesso em: 27 mai. 2021.

CODE.ORG. **About Us**, [s.l.] 2021b. Disponível em: <https://code.org/about>. Acesso em: 27 mai. 2021.

CODESPARK. **About Us**, [s.l.], 2021. Disponível em: <https://codespark.com/about>. Acesso em: 25 mai. 2021.

CODESPARK. **Frequently Asked Questions**, [s.l.] 2021b. Disponível em: <https://codespark.com/frequently-asked-questions>. Acesso em: 25 mai. 2021.

Lu, J. J. and Fletcher, G. H. (2009). Thinking about computational thinking. In Proc. 40th Technical Symp. on Comp. Sci. Education, pages 260–264, New York, USA. ACM.

MATTOS, Mauro et al. **Furbot Móvel**: um jogo para o ensino do pensamento computacional. In: VIII CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 2019, Brasília. Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação (CBIE 2019). [S.L.]: Brazilian Computer Society (Sociedade Brasileira de Computação - Sbc), 2019. p. 1294-1301

Nascimento, J., Xavier, D., Passos, O. e Barreto, R., (2015) **Um Relato de Experiência da Aplicação de Técnicas Interativas para Ensino da Computação na Educação Básica**. Anais do Encontro Regional de Computação e Sistemas de Informação. p. 95-104

NORVIG, Peter; RUSSELL, Stuart. **Inteligência Artificial**. Tradução da 3a Edição. Elsevier Brasil, 2014.

RAABE, André L. A. et al. **Referenciais de Formação em Computação**: Educação Básica. Porto Alegre. Assembleias do WEI e da SBC. 2017. 9 f. Disponível em: <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf> Acesso em: 27 mai. 2021.

RIBEIRO, L.; FOSS, L.; CAVALHEIRO, S. A. da C. **Entendendo o Pensamento Computacional**. Jul. Disponível em: <https://arxiv.org/abs/1707.00338>. Acesso em: 01 jun. 2021

SCAICO, Pasqueline D. et al. **Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch**. Revista Brasileira de Informática na Educação, Porto Alegre, v. 21, n. 2, p. 92, set. 2013.

Schlögl, Lucas Eduardo; Oliveira, Gabriel Castellani de; Giovanella, Gian Carlo; Bizon, A. R.; Santos, B. F. F.; Kruger, N.; Bursoni, P.; Neumann, C. B.; Huber, E. E.; Araújo, Luciana P; Mattos, Mauro M.; Zucco, F. D.; Cunha, K. Z.; Hein, N. **Ensino do Pensamento Computacional na Educação Básica**. Revista de Sistemas e Computação - RSC, v. 7, p. 304-322, 2017.

Vahldick, Adilson; Mattos, Mauro Marcelo. **Relato de uma Experiência no Ensino de Algoritmos e Programação Utilizando um Framework Lúdico** In: Simpósio Brasileiro de Informática na Educação, 2008, Fortaleza. XIX Simpósio Brasileiro de Informática na Educação. 2008.

Wing, J. M. (2006). **Computational thinking**. Communications of the ACM, v. 49, n. 3, p.33–35.

XAVIER, Francisca Edyr. **GERAÇÃO AUTOMATIZADA DE GABARITO E CORREÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT PARA O ENSINO DO PENSAMENTO COMPUTACIONAL**. 2019. 26 f. TCC (Graduação) - Curso de Ciências da Computação, Universidade Regional de Blumenau, Blumenau, 2019. Disponível em: http://dsc.inf.furb.br/arquivos/tccs/monografias/2019_1_francisca-edyr-xavier_monografia.pdf. Acesso em: 01 maio 2021.

ZEFERINO, A. M. B.; DOMINGUES, R. C. L.; AMARAL, E. **Feedback como estratégia de aprendizado no ensino médico**. Rev. bras. educ. med. [online]. 2007, vol.31, n.2, pp.176-179. Disponível em: <https://www.scielo.br/j/rbem/a/yK7SFyqJBCm6hRqNk4Szyt/abstract/?lang=pt>. Acesso em: 05 jun 2021.

Comentado [DSdR47]: Negrito.

Comentado [DSdR48]: Juntar com a frase anterior.

Comentado [DSdR49]: Remover sublinhas. CODE.ORG.

Comentado [DSdR50]: 2021a

Comentado [DSdR51]: Remover sublinhas. CODESPARK.

Comentado [DSdR52]: ;

Comentado [DSdR53]: Formato ABNT para eventos.

Comentado [DSdR54]: Formato ABNT para eventos.

Comentado [DSdR55]: Ponto final.

Comentado [DSdR56]: Arrumar espaçamento entre parágrafos.

Comentado [DSdR57]: Remover aspas duplas.

Comentado [DSdR58]: Negrito.

Comentado [DSdR59]: Ponto final.

Comentado [DSdR60]: Negrito.

Comentado [DSdR61]: Falta ano 2017.

Comentado [DSdR62]: Negrito.

Comentado [DSdR63]: Ponto final.

Comentado [DSdR64]: Negrito.

Comentado [DSdR65]: Negrito.

Comentado [DSdR66]: Espaço em branco.

Comentado [DSdR67]: Formato ABNT para eventos.

Comentado [DSdR68]: Negrito.

Comentado [DSdR69]: Juntar com a frase anterior.

Comentado [DSdR70]: Só as letras iniciais em maiúsculo.

Comentado [DSdR71]: Negrito.

FORMULÁRIO DE AVALIAÇÃO BCC – PROFESSOR TCC I

Avaliador(a): Dalton Solano dos Reis

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	X		
	O problema está claramente formulado?	X		
	2. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	X		
	Os objetivos específicos são coerentes com o objetivo principal?	X		
	3. JUSTIFICATIVA São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?	X		
ASPECTOS METODOLÓGICOS	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	X		
	4. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	X		
	Os métodos, recursos e o cronograma estão devidamente apresentados?	X		
	5. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?	X		
	6. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?	X		
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	X		
	7. ORGANIZAÇÃO E APRESENTAÇÃO GRÁFICA DO TEXTO A organização e apresentação dos capítulos, seções, subseções e parágrafos estão de acordo com o modelo estabelecido?	X		
	8. ILUSTRAÇÕES (figuras, quadros, tabelas) As ilustrações são legíveis e obedecem às normas da ABNT?		X	
	9. REFERÊNCIAS E CITAÇÕES As referências obedecem às normas da ABNT?		X	
	As citações obedecem às normas da ABNT?		X	
	Todos os documentos citados foram referenciados e vice-versa, isto é, as citações e referências são consistentes?	X		

Comentado [DSdR72]: Indicados no texto.

Comentado [DSdR73]: Indicados no texto.

Comentado [DSdR74]: Indicados no texto.



UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
DISCIPLINA: TRABALHO DE CONCLUSÃO DE CURSO I
CURSO: CIÊNCIA DA COMPUTAÇÃO - BCC

ATA DA DEFESA: BANCA DO PRÉ-PROJETO

Venho, por meio deste, manifestar minha avaliação sobre a **apresentação** do Pré-Projeto de TCC realizado pelo(a) acadêmico(a), Gian Carlo Giovanella no **SEGUNDO SEMESTRE DE 2021**, com o título GERAÇÃO AUTOMATIZADA DE RESOLUÇÃO DE EXERCÍCIOS EM AMBIENTE FURBOT UTILIZANDO MÉTODOS DE APRENDIZAGEM DE MÁQUINA.

A referida apresentação obteve a seguinte nota:

Componente da Banca	Nota (de 0 a 10)
Professor(a) Orientador(a): Luciana Pereira de Araújo Kohler	7,0

A apresentação aconteceu em 21/ 10 / 2021 na sala de reunião virtual do MS-Teams, tendo início às 17:00 hs e foi encerrada às 17:30 hs.

ATENÇÃO. A nota acima se refere somente a apresentação do pré-projeto e vai ser repassada para o aluno (orientando). Favor preencher os campos acima e enviar por e-mail ao professor de TCC1 (dalton@furb.br). Lembro que os arquivos com as anotações das revisões do professor de TCC1 e Avaliador serão enviados para o orientando e professor orientador após o professor de TCC1 receber esta ata preenchida. Caso julgue necessário fazer mais alguma consideração relacionada ao pré-projeto ou a defesa, favor usar o espaço abaixo.

Observações da apresentação:

Minha internet estava um pouco travada, mas consegui acompanhar, só não pude ligar a câmera.