

Ludicidade como abordagem pedagógica para o ensino de algoritmos de pesquisa sequencial e binária

Graziela Ferreira Guarda, Sandro Miranda de Rezende, Sérgio Crespo Coelho da Silva Pinto

grafergua@alum.us.es; {graziela guarda, sandromiranda, screspo}@id.uff.br

Escuela Técnica Superior de Ingeniería Informática – Universidad D Sevilla, Universidade Federal Fluminense

RESUMO

O presente relato de experiência apresenta os resultados da aplicação de abordagem lúdica associada ao Pensamento Computacional (PC) como estratégia pedagógica no ensino dos algoritmos de pesquisa sequencial e binária com alunos do nível superior. Nesse sentido, a atividade explorou de maneira criativa os algoritmos desenvolvendo conjuntamente habilidades do PC. A mesma foi realizada em sala de aula com 60 estudantes da disciplina de Algoritmos e Programação. A experiência foi considerada exitosa e o método estimulante por ser uma atividade atrativa e não tradicional, além de ter contribuído para a melhora de rendimento acadêmico.

CCS CONCEPTS

• **Social and professional topics** → Computing education.

PALAVRAS-CHAVE

Algoritmos de Pesquisa, Pensamento Computacional, Pesquisa Binária, Pesquisa Sequencial.

1 INTRODUÇÃO

A evasão nas disciplinas iniciais dos cursos de graduação em Computação não é novidade. De acordo com [9], no Brasil tem sido recorrente os alunos chegarem à universidade com deficiências da formação básica, o que inclui todas as áreas, desde escrita e interpretação de textos a déficits da matemática, com índices de proficiência cada vez menores no passar dos anos [9]. Estas deficiências têm sido apontadas como fatores impactantes e têm sido relacionadas ao cancelamento ou abandono da disciplina de Algoritmos e Programação.

Nesse contexto, é importante buscar novas estratégias para motivar e incentivar a permanência dos alunos nos cursos de Computação. Assim, buscou-se trabalhar de forma lúdica, explorando as habilidades dos 4 pilares do Pensamento Computacional (PC) para promover o ensino-aprendizagem dos algoritmos de pesquisa sequencial e binária. Nesse sentido, o PC pode ser compreendido como uma abordagem voltada para a resolução de problemas explorando processos cognitivos, pois

envolve a capacidade de compreender as situações propostas e criar soluções através de modelos matemáticos, científicos ou sociais para aumentar nossa produtividade, inventividade e criatividade [7].

Em relação às habilidades do PC trabalhadas, adotou-se o uso da abordagem teórica do guia difundido pela *Computer at School* [3] chamado “Quatro Pilares do Pensamento Computacional”. Essa escolha se deu pelo fato da abordagem já estar consolidada em trabalhos relevantes como o de [2] e são: decomposição: envolve identificar um problema complexo e quebrá-lo em pedaços menores e mais fáceis de gerenciar; reconhecimento de padrões: cada um desses problemas menores pode ser analisado individualmente com maior profundidade, identificando problemas parecidos que já foram solucionados anteriormente; abstração: capacidade de filtrar informações essenciais e descartar as informações desnecessárias em um determinado contexto e algoritmos: sequências lógicas para alcançar um objetivo [3], [8], [10], [11].

Desse modo, o relato de experiência visa contribuir para uma ressignificação do ensino de algoritmos de forma criativa, e a questão de pesquisa é definida da seguinte forma: Como podemos realizar atividades no ensino superior usando métodos não formais de aprendizagem na disciplina de algoritmos? O trabalho está dividido da seguinte maneira: na Seção 2, é apresentada a proposta e a metodologia utilizada. Os resultados e discussões são descritos na Seção 3 e por fim, as conclusões e perspectivas de trabalhos futuros são mostradas na Seção 4.

2 PROPOSTA E METODOLOGIA

A presente atividade é uma estratégia ou abordagem para o ensino-aprendizagem dos algoritmos de pesquisa sequencial e binária. O público-alvo foram duas turmas da disciplina introdutória de ‘Algoritmos e Programação’ de um curso de Bacharelado em Ciência da Computação. A atividade ocorreu em um dia de aula com duração de 3 horas/aulas de 50 minutos cada, de forma presencial no 2º semestre de 2019. A mesma teve por objetivo ensinar os algoritmos de pesquisa sequencial e binária de maneira lúdica, buscando melhora da compreensão dos conteúdos e, conseqüentemente, um melhor rendimento na avaliação de aprendizagem, tendo em vista o volume de reprovações da disciplina e os relatos de falta de compreensão quanto à aplicação dos algoritmos na vida prática.

A atividade desenvolvida é classificada como pesquisa quantitativa quanto à abordagem metodológica [5]. Deste modo, os objetivos foram mais centrados em analisar como os alunos estariam evoluindo quanto a exploração lúdica e de raciocínio,

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'22, Abril 24-29, 2022, Feira de Santana, Bahia, Brasil (On-line)

© 2022 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

bem como seus rendimentos nas avaliações de aprendizagem. Quanto à natureza, trata-se de uma pesquisa aplicada, tendo em vista que foi realizada uma atividade prática de caráter exploratório envolvendo ensaio com os grupos de alunos.

A mesma desenvolvida em três etapas: a primeira para explicação dos algoritmos, a segunda para a realização da atividade lúdica e a terceira para analisar o desempenho dos alunos nas avaliações de aprendizagem.

2.1 Etapa 1 – Conhecendo os Algoritmos (Pesquisa Sequencial e Binária)

A primeira etapa do processo consistiu em desenvolver junto aos alunos o conceito de pesquisa sequencial e binária. Tal etapa foi conduzida com uso do quadro da sala de aula de forma tradicional e teve duração de 50 minutos (1 hora/aula) entre as explicações do conteúdo e as respostas das dúvidas dos estudantes quanto aos aspectos teóricos, o que representa uma carga horária equivalente ao tempo gasto na sala de aula regular (laboratório).

Por pesquisa (procura ou busca) entende-se o ato de recuperar uma informação em um conjunto de dados. Os principais algoritmos de pesquisa em memória primária são: pesquisa sequencial (ou linear) e pesquisa binária. O algoritmo da pesquisa sequencial é utilizado quando os dados não estão ordenados pela chave de pesquisa. A busca se inicia pelo primeiro registro, avança sequencialmente (registro por registro) e termina com sucesso quando a chave pesquisada é encontrada, ou sem sucesso quando todos os registros são pesquisados, mas a chave não é encontrada [4].

Já no algoritmo da pesquisa binária, os dados da tabela precisam estar ordenados (ordem crescente ou decrescente) e a tabela pode ter muitos registros. Para saber se uma chave está presente na tabela, compara-se a chave com o registro que está no meio da tabela. Se a chave for menor, então o registro procurado está na primeira metade da tabela. Se a chave é maior, então o registro procurado está na segunda metade da tabela. O processo é repetido até que a chave seja encontrada ou que fique apenas um registro cuja chave é diferente da procurada, significando uma pesquisa sem sucesso. No caso da pesquisa binária, a quantidade de iterações cresce devagar e, deste modo, o algoritmo possui um desempenho muito superior em relação ao algoritmo da pesquisa sequencial. O Quadro 1 mostra a lógica dos algoritmos em português estruturado, bem como, a análise de complexidade dos mesmos [4].

Quadro 1: Lógica dos Algoritmos e análise de complexidade

	Pesquisa Sequencial:	Pesquisa Binária:
Algoritmos	<pre>int PesqSeq(int chave, int v[], int n) { int i; for (i=0; i<n; i++) { if (chave == v[i]) return (i); } return (-1); //Índice inválido ou não encontrado</pre>	<pre>int pesqBin(int chave, int v[], int n){ int inicio = 0; int meio; int fim = n-1; while (inicio <= fim) { meio = (inicio + fim)/2; if (chave < v[meio]) fim = meio - 1; else if (chave > v[meio]) inicio = meio + 1; else return meio; } return (-1); //Índice inválido ou não encontrado }</pre>
Análise de complexidade	<p>Para uma pesquisa com sucesso, temos: 1 iteração no melhor caso; N iterações no pior caso; $(N + 1) / 2$ iterações no caso médio. Para uma pesquisa sem sucesso, temos: N + 1 iterações.</p>	<p>A cada iteração o número de elementos a serem pesquisados é reduzido à metade: - N; N=2; N=4; N=8; ...; N=2k. Para tabela de 16 elementos! 4 iterações Para 1024 elementos, 10 iterações Para 1000000 elementos, 20 iterações.</p>

Adicionalmente, foi explicado ainda, como fazer uso dos blocos do *Scratch* ilustrados pela Figura 1 abaixo, para que a atividade lúdica pudesse ser compreendida. Essa explicação fez-se necessária pois os comandos seriam utilizados na Etapa 2 e foi essencial para que o experimento fosse realizado com sucesso.



Figura 1: Comandos do Scratch usados nos cartões

2.2 Etapa 2 – A atividade lúdica

A atividade foi projetada para acontecer entre equipes em uma matriz humana com tamanho de 8x8 (Figura 2). Cada espaço da matriz continha uma carta (desafio a ser respondido pelas equipes) para que os participantes pudessem avançar casas (Figura 3). Essa etapa ocorreu em 100 minutos (2 horas/aulas).

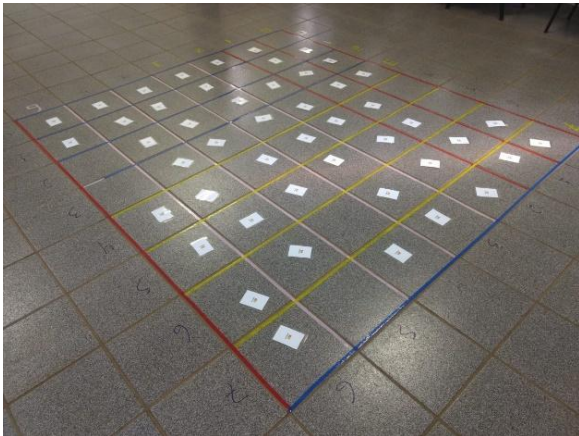


Figura 2: Matriz 8x8

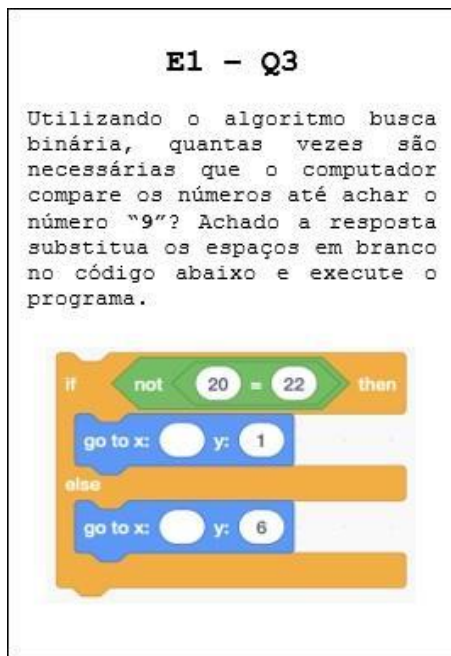


Figura 3: Cartões de um dos grupos

Quanto à operacionalização, os grupos se dividiram em 5 alunos cada, escolhidos aleatoriamente. Um deles era o líder. O líder era a pessoa responsável por se movimentar dentro da matriz utilizando para isso, as coordenadas representadas na parte azul - comandos 'go to x: y:' - este comando tinha por objetivo relembrar a posição qual o líder deveria se deslocar. Esse deslocamento ocorria na medida em que o restante da equipe respondia corretamente às questões propostas em cada uma das cartas.

Caso a equipe errasse a resposta, o líder não podia se movimentar na matriz. Além disso, o líder devia identificar corretamente a posição à qual devia se encaminhar, de acordo com a resposta dada pela equipe. Caso identificasse a posição corretamente na matriz, podia concretizar o movimento e, então, recebia uma nova carta que se encontrava na nova posição. Caso errasse a posição à qual deveria se encaminhar, não era permitido concretizar o movimento (ou seja, retorna à posição em que estava) e perdia a jogada da vez. Por fim, vencida a equipe que respondesse os nove desafios mais rapidamente.

Quanto ao conteúdo das cartas, foi criado um banco de questões contendo os desafios sobre os algoritmos de pesquisa binária e sequencial para que os conceitos fossem postos em prática durante a realização da atividade. Para que as questões pudessem ser respondidas, foi disponibilizado, ainda, um vetor numérico de valores inteiros que seriam a base para a geração das respostas dos desafios (Figura 4).

```
int numeros = [2,5,7,9,12,14,16,19,24,26,
               28,30,32,35,38,42,47,49,50,51,
               52,53,57,65,67,73,77,85,86,89,93,94]
```

Figura 4: Vetor de números inteiros ordenados

Quanto ao banco de questões, o Quadro 2 mostra os desafios propostos e os respectivos gabaritos. É importante ressaltar que as questões tinham o mesmo nível de dificuldade. A observação e acompanhamento das equipes foi feita pelos três monitores da disciplina com apoio da professora e, para tal, foi utilizada uma planilha de acompanhamento. Nessa planilha foram feitos os registros das questões corretas, erradas e quanto tempo cada grupo levou para finalizar suas jogadas, além de conter um campo de observações para registro das intercorrências. O banco de questão foi disponibilizado em cartões conforme ilustrado na Figura 3 e o mesmo foi composto pelo enunciado escrito (Quadro 2) seguido da parte em Scratch (Quadro 3).

Quadro 2: Banco de questões dos cartões (cartas desafios)

Questão	Enunciado:	Gabarito:
1	Utilizando o algoritmo busca binária, quantas vezes são necessárias que o computador compare os números até achar o número "42"? Achando a resposta substitua os espaços em branco no código abaixo e execute o programa.	1
2	Utilizando o algoritmo busca binária, quantas vezes são necessárias que o computador compare os números até achar o número "19"? Achando a resposta substitua os espaços em branco no código abaixo e execute o programa.	1
3	Utilizando o algoritmo busca binária, quantas vezes são necessárias que o computador compare os números até achar o número "9"? Achando a	2

4 resposta substitua os espaços em branco no código abaixo e execute o programa. Utilizando o algoritmo busca binária, quantas vezes são necessárias que o computador compare os números até achar o número "53"? Achando a resposta substitua os espaços em branco no código abaixo e execute o programa.

5 Utilizando o algoritmo busca binária, quantas vezes são necessárias que o computador compare os números até achar o número "2"? Achando a resposta substitua os espaços em branco no código abaixo e execute o programa.

6 Utilizando o algoritmo busca binária, quantas vezes são necessárias que o computador compare os números até achar o número "89"? Achando a resposta substitua os espaços em branco no código abaixo e execute o programa.

7 Utilizando o algoritmo busca sequencial, quantas vezes são necessárias que o computador compare os números até achar o número "16"? Achando a resposta substitua os espaços em branco no código abaixo e execute o programa.

8 Em um vetor de inteiros de 1 milhão de números ordenados em ordem crescente, quantas vezes são necessárias que o computador compare os números para tentar achar um número que não está no vetor utilizando o algoritmo busca sequencial? E utilizando o algoritmo busca binária?

9 Utilizando o algoritmo busca sequencial, quantas vezes são necessárias que o computador compare os números até achar o número "2"? Achando a resposta substitua os espaços em branco no código abaixo e execute o programa.

3

4

5

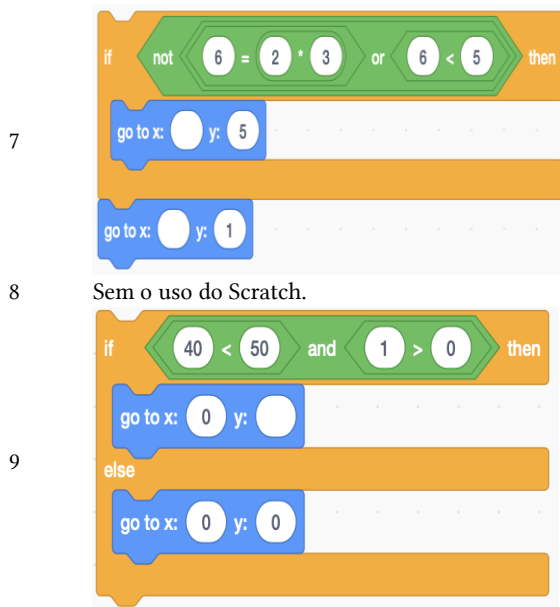
4

Sequencial: 1 milhão, Binária: 19.

7

Quadro 3: Banco de questões dos cartões (parte do Scratch)

Questão	Enunciado:
1	
2	
3	
4	
5	
6	



2.3 Etapa 3 – Análise de desempenho - avaliação de aprendizagem

Essa etapa consistiu em realizar uma comparação dos rendimentos dos estudantes na questão da avaliação de aprendizagem da disciplina que envolvia os algoritmos de pesquisa sequencial e binária, com o objetivo de identificar possível melhora na aprendizagem, tendo ocorrido em momento pós as Etapas 1 e 2.

Para tal, foi feito um comparativo entre as turmas no que se refere ao item de prova que abordava os algoritmos. Esse item foi uma questão com mesmo grau de complexidade que compunha as avaliações aplicadas no 1º semestre de 2019 (duas turmas que aprenderam os algoritmos com aulas tradicionais, com 30 estudantes em cada) e no 2º semestre de 2019 (duas turmas que aprenderam os algoritmos via atividade lúdica, com 30 estudantes em cada), objetivando-se fazer um comparativo para analisar se haveria diferença nos rendimentos dos diferentes grupos.

3 RESULTADOS E DISCUSSÕES

A atividade lúdica foi realizada com um total de 60 alunos divididos em 12 equipes de 2 turmas diferentes da disciplina de 'Algoritmos e Programação'. Ambas as turmas eram compostas por 30 alunos e cada equipe foi montada com 5 integrantes. Como a atividade foi divulgada às turmas antecipadamente e a presença foi definida como obrigatória, é importante ressaltar que todos os estudantes matriculados na disciplina participaram.

Baseado na planilha de acompanhamento, os resultados compilados da turma 1 mostraram que a equipe campeã terminou suas jogadas em 86 minutos. A média de acerto nas questões foi de 61% pontos percentuais e a equipe campeã foi a 5 (Tabela 1).

Tabela 1: Resultados da turma 1

	Eq. 1:	Eq. 2:	Eq. 3:	Eq. 4:	Eq. 5:	Eq. 6:
Qtde. Questões	6	4	8	3	9	3
Aproveitamento em porcentagem	67	44	89	33	100	33

Já na turma 2, os resultados compilados da turma mostraram que a equipe campeã terminou suas jogadas em 65 minutos. A média de acerto nas questões foi de 81.67% pontos percentuais e a equipe campeã foi a 6 (Tabela 2).

Tabela 2: Resultados da turma 2

	Eq. 1:	Eq. 2:	Eq. 3:	Eq. 4:	Eq. 5:	Eq. 6:
Qtde. Questões	7	7	6	7	8	9
Aproveitamento em porcentagem	78	78	67	78	89	100

Quanto à avaliação de aprendizagem, a questão utilizada no 2º semestre de 2019 pode ser consultada no link <https://bit.ly/2UDGlfm>. A mesma foi aplicada no mesmo dia para as duas turmas, 2 semanas após a realização da atividade lúdica pois os conteúdos abordados foram os últimos da ementa da disciplina. A questão valia 5,0 pontos, de maneira que cada item do menu (consultar a prova) valia 1,0 ponto e a função principal também valia 1,0 ponto. Deste modo, foi possível analisar o rendimento dos estudantes nos itens sobre pesquisa sequencial e pesquisa binária (case 2 e case 3 da avaliação, respectivamente). A distribuição das notas foi organizada em diagramas de caixas (boxplots), ferramentas que permitem visualizar média, mediana e variabilidade da amostra, assim como identificar valores discrepantes. O Gráfico 1 estabelece um comparativo entre as pontuações nos itens mencionados dos estudantes das turmas que aprenderam os algoritmos em aulas tradicionais (Pesq. Seq. Trad. e Pesq. Bin. Trad.) e das turmas que aprenderam via metodologia lúdica (Pesq. Seq. Lud. e Pesq. Bin. Lud.).

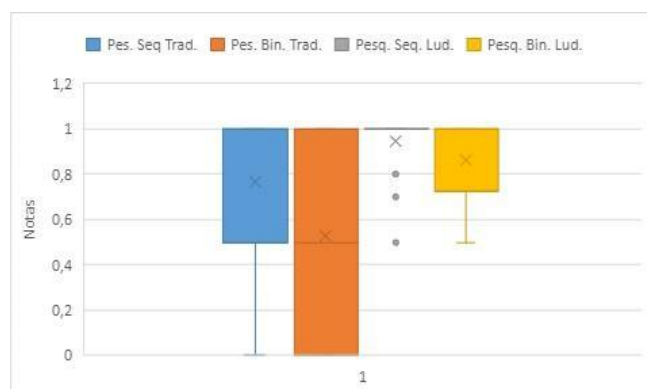


Gráfico 1: Rendimento dos estudantes nos itens da avaliação de aprendizagem.

O Gráfico 1 mostra a distribuição das pontuações dos estudantes na questão da avaliação de aprendizagem referente à implementação dos algoritmos abordados (pesquisa sequencial: case 2 e pesquisa binária: case 3). A média aritmética das pontuações é exibida através de um X em cada caso. Já a mediana é representada por uma linha horizontal que pode estar no interior ou coincidir com uma das laterais da caixa. As caixas retangulares representam as distâncias entre o quartil 1 e quartil 3, compreendendo 50% dos dados em cada distribuição. Os dados que se encontram na cauda inferior (abaixo da caixa) e na cauda superior (acima da caixa) compreendem o fundo de 25% e o topo de 25% dos valores da amostra, respectivamente.

Pode-se observar que na categoria “Pesquisa binária tradicional”, o diagrama não apresenta nem cauda inferior nem cauda superior, o que significa que, se por um lado mais de 25% dos estudantes obtiveram pontuação zero no item, por outro mais de 25% dos estudantes obtiveram pontuação integral. Tal fato indica grande variabilidade das pontuações obtidas pelos estudantes nesse caso, havendo tanto uma parcela significativa dos estudantes que teve boa compreensão do conteúdo, quanto uma parcela considerável que não compreendeu. Já nas caixas referentes às categorias “Pesquisa sequencial tradicional”, “Pesquisa sequencial lúdico” e “Pesquisa binária lúdico”, a mediana das distribuições coincide com a lateral superior da caixa, o que significa que mais da metade dos estudantes obteve a pontuação integral do item em cada caso. Tal fato, aliado aos valores das médias aritméticas observados nos diagramas, sugere que os estudantes tiveram mais facilidade com os algoritmos de pesquisa sequencial, além de indicar um melhor rendimento das turmas onde foi empregada a metodologia lúdica. Além disso, na categoria Pesquisa sequencial lúdica, o diagrama de caixa se resume a uma linha na parte superior dos dados, havendo alguns valores discrepantes abaixo (*outliers*). Isso significa que, nesse caso, o quartil 1 vale 1, o que revela que mais de 75% dos estudantes obtiveram pontuação integral no item nas turmas onde foi empregada metodologia lúdica.

Dessa forma, a partir da análise do Gráfico 1, e considerando-se os fatores mencionados anteriormente, conclui-se que houve uma melhora de rendimento nas turmas que realizaram a atividade lúdica, tanto nos algoritmos de pesquisa sequencial quanto nos de pesquisa binária. No caso de ambos os algoritmos se observou um aumento na média aritmética e na mediana da distribuição de notas dos estudantes quando se empregou a atividade lúdica. Além disso, a variabilidade da amostra diminuiu, de maneira que grande parte das pontuações passou a se concentrar em valores mais elevados.

Outro aspecto observado é que no semestre em que houve a atividade lúdica nenhum estudante deixou itens em branco. Além disso, não podemos deixar de expor que, com o aviso prévio da aula diferenciada, todos os alunos participaram, o que demonstra o interesse e motivação por aulas com abordagens didáticas diferenciadas. Esses resultados nos motivam a continuar buscando práticas diferenciadas e/ou inovadoras no ensino-aprendizagem em nível superior e mostram que é possível combater a evasão especialmente nas disciplinas de algoritmos e linguagens de

programação que são as disciplinas em que os alunos apresentam mais dificuldade.

Por fim, as equipes ficaram em observação pelos monitores da disciplina e professora e, neste sentido, foram identificados alguns elementos que foram classificados como positivos e negativos da experiência realizada.

Aspectos positivos: A atividade foi muito bem recebida pelos alunos: parte deles se motivaram a estudar os algoritmos com antecedência, bem como conhecer o *Scratch*. Além disso, os alunos solicitaram à professora que realizasse novas aulas nesse formato nos semestres seguintes pois além de ter facilitado a compressão dos algoritmos, foi um processo didático do qual eles realmente gostaram. A professora também relata que não sabia como a atividade seria recebida pelos alunos e que o retorno positivo a motivou a continuar buscando novas práticas de ensino nas suas ações pedagógicas. Já os monitores (que aprenderam os algoritmos em aulas tradicionais) relataram que o conteúdo foi reforçado de forma rica e que gostariam de ter aprendido com a abordagem pedagógica diferenciada.

Em relação ao tempo disponível para responder as questões, foi considerado suficiente: os estudantes levaram mais tempo para concluir a primeira questão por conta de uma necessidade de compreensão dos comandos. Em relação aos materiais utilizados para confecção do cenário e dos cartões, foram utilizados itens de baixo custo (papéis A4, canetinha, fita adesiva transparente, plástico para plastificar os cartões).

Aspectos negativos: Nem todas as equipes conseguiram responder mais da metade dos desafios propostos. Outro fator destacado foi o ruído: em virtude do quantitativo alto de alunos na sala e da existência de conversas paralelas, o raciocínio das equipes ficou prejudicado, gerando dispersão por parte de alguns alunos. Neste sentido, recomenda-se que em outra ocasião se realize a atividade com as turmas separadamente, o que poderia trazer também elementos novos a serem ponderados.

Quanto às habilidades do Pensamento Computacional, a atividade incluiu na sua metodologia os quatro pilares (decomposição, reconhecimento de padrões, abstração e algoritmo) e os explorou tanto no uso do *Scratch* desplugado quanto na operacionalização da atividade em si, uma vez que os estudantes tiveram que ler e interpretar os algoritmos em blocos. Além disso, também realizaram abstrações e fizeram uso da decomposição, tendo relatado, ainda, que perceberam mais eficiência na resolução de problemas quando aplicada a decomposição.

3.1 Limitações

A atividade se limitou a trabalhar os algoritmos de pesquisa sequencial e binária de forma lúdica usando o *Scratch*. A ementa da disciplina é bem mais ampla e conforme foi exposto anteriormente, esse era o último conteúdo do plano de ensino da disciplina. Além disso, o tópico da avaliação de aprendizagem que entrou na análise foi especificamente referente aos algoritmos de

pesquisa sequencial e binário não contemplando os demais conteúdos abordados na avaliação.

4 CONSIDERAÇÕES FINAIS

O presente artigo apresentou um relato de experiência do ensino de algoritmos de pesquisa de forma lúdica que foi desenvolvida com turmas de nível superior. Os resultados mostraram melhora de rendimento acadêmico nas avaliações de aprendizagem, engajamento e maior motivação pelo aprender. De forma complementar, o intuito foi também desenvolver as habilidades do PC para superar obstáculos e dificuldades, em especial os da aprendizagem de linguagens de programação com apoio do *Scratch* desplugado.

Atividades lúdicas são ricas, estimulam a vida social e representam uma contribuição significativa na aprendizagem. Através destas, alunos desenvolvem capacidades, conhecimentos, atitudes e habilidades, destacando-se entre elas: o favorecimento da mobilidade, a imaginação, a diversão, a aceitação de regras, o desenvolvimento do raciocínio lógico, entre outros. O uso abordagem lúdica foi uma maneira interessante para trabalhar as temáticas com o público-alvo. Neste sentido, o objetivo foi alcançado e a experiência considerada exitosa pelos alunos, que foi recebida com entusiasmo e também pela professora e monitores.

4.1 Trabalhos Futuros

Como trabalhos futuros, pretende-se explorar novas abordagens lúdicas, análises qualitativas e mais aprofundadas em relação as atividades em grupos buscando compreender melhor o desempenho individual de um aluno e o quanto isso pode afetar o desempenho geral do trabalho em equipe e também possíveis adequações para o atual ensino remoto e híbrido. Além disso, estima-se buscar identificar o impacto na capacidade de aprendizado desses alunos em disciplinas posteriores do curso.

REFERÊNCIAS

- [1] Barr, V. & Stephenson, C. (2011). Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1):48-54.
- [2] Brackmann, C. P. (2017) Desenvolvimento do Pensamento Computacional através de atividades desplugadas na Educação Básica. Tese (Doutorado). Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Informática na Educação, 226 f.
- [3] Computer at School (CAS). Developing Computational Thinking. Teaching London Computing, 2014. Disponível em: <http://teachinglondoncomputing.org/resources/developing-computational-thinking/>. Acesso em: 10/06/2019.
- [4] Cormen, H. Thomas; Leiserson, E. Charles; Ronald L. Rivest, Stein Clifford. (2012). Algoritmos: Teoria e Prática, 3ª edição, Editora GEN LTC. ISBN: 978-8535236996.
- [5] Gerhardt, T. E.; Silveira, D. T. Métodos de Pesquisa. Universidade Federal do Rio Grande do Sul, 2009. 1a edição. ISBN 978-85-386-0071-8.
- [6] Giraffa, L. M. M.; Mora, M. C. Evasão na disciplina de algoritmos e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. III Clabes, 2013, México.
- [7] Guarda, Graziela Ferreira; Pinto, Sérgio Crespo C. S. Dimensões do Pensamento Computacional: conceitos, práticas e novas perspectivas. In: Simpósio Brasileiro de Informática na Educação, 31. 2020, Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 1463-1472. DOI: <https://doi.org/10.5753/cbie.sbie.2020.1463>.
- [8] Moreno-León, J., Robles, G., Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *Revista de Educación a Distancia*, (46), 1-23.
- [9] Souza, Daniella; Goulart, Maria; Guarda, Graziela; Goulart, Ione. Lightbot Logicamente: um game lúdico amparado pelo Pensamento Computacional e a Matemática. In: Workshop de Informática na Escola, XIV, 2018, Fortaleza. Anais: Sociedade Brasileira de Computação – SBC, 2018. p. 61 - 69.
- [10] Wing, J. M. (2006, March). Computational thinking. *Communications of the ACM*, 49(3), 33-35. DOI: 10.1145/1118178.1118215.
- [11] Wing, J. Computational thinking. What and why? Carnegie-Mellon School of Computer Science Research Notebook (Mar. 2011). <https://www.cs.cmu.edu/link/research-notebookcomputational-thinkingwhat-and-why>.