



**UNIVERSIDADE ESTADUAL DA PARAÍBA – UEPB
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS - CCEA
CURSO DE LICENCIATURA EM COMPUTAÇÃO**

VICTOR ANDERSON FORTUNATO DOS SANTOS

**O *SCRATCH* COMO FERRAMENTA DE APOIO AO ENSINO DE
ALGORITMOS: UMA OPINIÃO DISCENTE.**

**PATOS - PB
2017**

VICTOR ANDERSON FORTUNATO DOS SANTOS

**O SCRATCH COMO FERRAMENTA DE APOIO AO ENSINO DE
ALGORITMOS: UMA OPINIÃO DISCENTE.**

Trabalho de conclusão de curso apresentado à coordenação do curso de Licenciatura em Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Licenciado em Computação.

Área de concentração: Educação.

Orientador: Prof.^a Dr.^a Kézia de Vasconcelos Oliveira Dantas.

**PATOS
2017**

É expressamente proibida a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano da dissertação.

S237s Santos, Victor Anderson Fortunato dos
O Scratch como ferramenta de apoio ao ensino de Algoritmos
[manuscrito] : uma opinião discente / Victor Anderson Fortunato
dos Santos. - 2017.
22 p.

Digitado.
Trabalho de Conclusão de Curso (Graduação em Computação)
- Universidade Estadual da Paraíba, Centro de Ciências Exatas e
Sociais Aplicadas, 2017.
"Orientação: Prof. Dra. Kézia de Vasconcelos Oliveira
Dantas, CCEA".

1. Scratch. 2. Algoritmos. 3. Ludicidade na Informática. I.
Título.

21. ed. CDD 005.3

Victor Anderson Fortunato dos Santos

**O SCRATCH COMO FERRAMENTA DE APOIO AO ENSINO DE ALGORITMOS:
UMA OPINIÃO DISCENTE**

Trabalho de Conclusão de Curso apresentado ao
Curso de Licenciatura em Computação da
Universidade Estadual da Paraíba, em
cumprimento à exigência para obtenção do grau
de Licenciado em Computação

Aprovado em 4 de agosto de 2017

BANCA EXAMINADORA

Kézia de V. O. Dantas
Prof.^a Dra. Kézia de Vasconcelos Oliveira Dantas
(Orientadora)

Danielle Chaves de Medeiros
Prof.^a Ma. Danielle Chaves de Medeiros
(Examinadora)

Pablo Roberto Fernandes de Oliveira
Prof. Pablo Roberto Fernandes de Oliveira
(Examinador)

O SCRATCH COMO FERRAMENTA DE APOIO AO ENSINO DE ALGORITMOS: UMA OPINIÃO DISCENTE.

Victor Anderson Fortunato dos Santos*

RESUMO

Em cursos de programação é comum encontrar turmas de Algoritmos com superlotação. Na pesquisa que foi realizada para compor este trabalho, pode-se identificar que existe um grau elevado de reprovação dentre os veteranos, fato que pode subsidiar essa superlotação. Este trabalho trata exatamente isso, ele leva em consideração o fato de que a alta taxa de reprovação pode desestimular os alunos a permanecerem no curso. E para tentar reverter o quadro de reprovação, foi sugerida a inserção do *Scratch* como ferramenta de apoio aos alunos, uma vez que a utilização dessa ferramenta traz a oportunidade de uma aprendizagem de maneira lúdica, opondo-se às mais tradicionais na qual os alunos apenas interagem com uma interface pouco amigável. Para amparar essa proposta, foi realizada uma pesquisa quantitativa sobre a opinião dos alunos acerca da mesma. Para tanto, foi necessária uma primeira pesquisa para identificar os problemas existentes na opinião dos alunos. Posteriormente, foi ministrada uma oficina situando o *Scratch* na disciplina e por fim foi realizada a última pesquisa para identificar a aceitação da ferramenta. Como resultado, obteve-se aceitação de aproximadamente 96,66% para a ferramenta, ou seja, aproximadamente 96,66% dos alunos entrevistados aceitaram a proposta.

Palavras-Chave: *Scratch*. Algoritmos. Lúdico.

* Aluno de Graduação em Licenciatura em Computação na Universidade Estadual da Paraíba – Campus VII. E-mail: victor.and.fortunato@gmail.com

1. INTRODUÇÃO

O Bacharelado em Ciência da Computação aqui no Brasil é um curso superior que tem como principal objetivo formar profissionais que possam estar aptos para lidar com programação, projetando e desenvolvendo *softwares*. O profissional analisa as necessidades dos usuários e projeta as soluções que venham suprir essas necessidades. As qualidades indispensáveis aos profissionais que desejem trabalhar nessa área são raciocínio lógico, exatidão, concentração, organização e iniciativa (VERONEZZI, 2010).

A Universidade Estadual da Paraíba, Campus VII, disponibiliza esse curso. E é exatamente nele que este trabalho que foi executado no semestre 2015.2 mantém seu foco. De todas as disciplinas que compõem a grade curricular do curso, as que estão relacionadas com programação e/ou raciocínio lógico são as que mais se destacam negativamente por ter maior taxa de reprovação. Dentre as disciplinas de programação que compõem o curso, existe uma conhecida como Algoritmos. É geralmente nesta disciplina que o aluno tem seu primeiro contato com a programação na graduação, uma vez que ela é ofertada no primeiro período. Essa disciplina aborda os princípios da lógica de programação, objetivando desenvolver a capacidade de análise e resolução de problemas por meio de algoritmos.

Como foi previamente citado, há um grande índice de reprovação nessa matéria. Campos (2010) ao entrevistar 60 alunos de duas turmas de Instituições de Ensino Superior constatou que 65% dos entrevistados sentem dificuldades em desenvolver um algoritmo, ainda que ele possa ser classificado como de “baixa complexidade”.

Por sua vez, Lima Júnior, et al. (2015) em sua pesquisa acerca dos problemas de aprendizagem em Algoritmos identificaram que 30% dos alunos pesquisados foram direcionados para a avaliação final, 40% foram reprovados de imediato e apenas 30% conseguiram obter êxito na disciplina sem necessitar fazer a prova final. O que mais chama a atenção na pesquisa citada anteriormente, é que 40% dos alunos não conseguiram nem ao menos conceito suficiente para ter direito a prova final. Tudo isso é um problema, pois esses fatos podem fazer com que haja superlotação nas turmas de Algoritmos, o que por sua vez pode prejudicar ainda mais o aprendizado dos alunos.

Outro problema causado pela reprovação é que os alunos podem se sentir desestimulados em continuar no curso (SIMAS, 2012). Esse quesito pode ser observado em uma pesquisa feita por Rumberger e Lim (2008 apud SOUSA, et al., 2015), que ao analisar 203 estudos sobre o assunto de abandono escolar, destacaram que grande parte deles atentam para o fato de que repetidos fracassos como notas baixas e repetência são fortes aspectos para um futuro abandono escolar.

Atentos a isso, Rapkiewicz et al. (2006, p. 4), após frisarem os altos índices de reprovação na cadeira de Algoritmos e percebido que essa reprovação dificulta a permanência dos alunos no curso, trabalharam uma solução para evitar a reprovação utilizando jogos, pois eles apontam que “[...] as dificuldades na aprendizagem de Algoritmos ocorrem por falta de um método de ensino adequado para essa disciplina, de material didático inovador [...]”. Desse modo, os autores identificaram a necessidade de uma metodologia de ensino diferenciada que pudesse auxiliar os alunos. Essa ideia se encaixa nessa pesquisa, uma vez que a linguagem utilizada pela professora no momento dessa pesquisa era *Python* que é uma linguagem que se assemelha as mais tradicionais, assim sendo, não pode ser considerada uma linguagem tão inovadora e seria necessária uma remodelagem na metodologia e ferramentas utilizadas (LAGE, p. 1, 2013).

Para tentar auxiliar os alunos no aprendizado de Algoritmos, Souza, Jaeger e Cardoso (2013) trataram da aplicação de jogos para a resolução de um algoritmo para um popular problema lógico. A partir disso, foi possível concluir que através de estratégias lúdicas houve uma contribuição significativa para o aprendizado dos discentes.

Outros trabalhos como Gonçalves (2011) com jogos, Falkembach (2003) com dispositivos multimídia, Grübel e Bez (2006) utilizando jogos na construção do conhecimento, tratam o uso do ambiente lúdico como uma forma de auxiliar os alunos na disciplina de Algoritmos e na construção do conhecimento.

Desse modo, é justificável a necessidade de utilizar meios diferenciados para ajudar os alunos na busca pelo conhecimento necessário para concluir a cadeira de Algoritmos com êxito. Para tanto, foi questionado à atual professora qual ela acreditava ser a maior dificuldade que os alunos tinham em relação a sua disciplina e ela destacou a interpretação de texto como sendo o maior problema.

Partindo desse ponto, foi proposto aos alunos a utilização de uma ferramenta para auxiliá-los. Essa ferramenta denomina-se *Scratch*. O *Scratch* foi desenvolvido para trabalhar com programação em blocos (mais informações sobre o *Scratch* estão disponíveis no capítulo 2 desse documento) (SCRATCH, 2016).

Todavia, a utilização do *Scratch* por si só, não é suficiente para sanar todos os problemas que os alunos têm. Porém, Salazar, Odakura e Barvinsk (2015) apontaram que a utilização do *Scratch* é um fator motivacional para o aprendizado na disciplina de Algoritmos. Isso é um ponto positivo para o *Scratch*, pois quanto mais motivado estiver um aluno mais ele buscará melhorar seu desempenho.

Desse modo, a intenção foi fazer com que o *Scratch* ajude a sanar ou mesmo a diminuir outras dificuldades como, por exemplo, linguagem da ferramenta utilizada, timidez, falta de estímulo entre outros problemas que os alunos possam ter. Assim sendo, para os alunos seria mais fácil focar seus esforços em corrigir o problema primário, uma vez que outras dificuldades serão minimizadas com a utilização do *Scratch*.

Com base nos pontos abordados anteriormente, esse trabalho foi proposto para pesquisar e demonstrar a aceitação do *Scratch* como uma ferramenta em potencial para melhorar o desempenho dos alunos na disciplina de Algoritmos. Sendo assim, a questão que norteia esse trabalho é: Na opinião de alunos o *Scratch* pode se tornar uma ferramenta capaz de auxiliar no desenvolvimento deles e assim, aumentar as suas chances de concluírem a matéria com êxito? Ao longo de todo o projeto os esforços foram direcionados para responder tal questionamento.

Desse modo, o objetivo geral desta pesquisa foi demonstrar a aceitação dos alunos acerca do uso do *Scratch* como ferramenta auxiliadora no processo de aprendizagem com relação aos conteúdos da disciplina Algoritmos. E para alcançar o objetivo geral apresentado, foram definidos os seguintes objetivos específicos: buscar na bibliografia específica embasamentos teóricos que subsidiem a pesquisa em questão; Efetuar um levantamento sobre as dificuldades encontradas pelos alunos; Executar uma oficina no intuito de apresentar a ferramenta *Scratch*, levando em consideração os problemas identificados; Realizar uma pesquisa sobre a opinião dos alunos acerca da metodologia apresentada e expor os resultados obtidos com a segunda pesquisa.

Como metodologia utilizada, segundo Silva e Menezes (2005), esta pesquisa se encaixa na categoria de pesquisa aplicada, uma vez que todo o conhecimento

abordado serviu de embasamento para a realização dos questionários e da oficina. Do ponto de vista de forma de abordagem, Silva e Menezes (2005) definem a pesquisa como quantitativa, pois as opiniões dos entrevistados foram transformadas em números para que pudessem ser analisadas.

Explorando a pesquisa do ponto de vista de seus objetivos, pode-se decompor a pesquisa em três classificações: exploratória, descritiva e explicativa (GIL, 2002). Desse modo, a presente pesquisa se define como exploratória, de modo que visa dar maior entendimento sobre o problema no intuito de torná-lo explícito. Quanto aos procedimentos técnicos, foi adotado o levantamento de dados, pois a pesquisa envolve a interrogação dos envolvidos para efetuar a averiguação acerca das dificuldades encontradas.

Logo, esta pesquisa considerou as seguintes etapas: pesquisa bibliográfica acerca dos impactos que a reprovação pode causar na graduação; Elaboração e aplicação do primeiro questionário cujo objetivo foi identificar as principais dificuldades elencadas pelos alunos; Realização da oficina utilizando a ferramenta *Scratch* de acordo com o problema identificado; Aplicação do segundo questionário, cujo objetivo é avaliar a aceitação por parte dos alunos acerca da metodologia apresentada e por fim, a demonstração dos resultados obtidos. Alcançando assim, todos os objetivos específicos previamente citados.

O restante deste trabalho está organizado da seguinte forma: seção 2 *Scratch*, na qual contém algumas informações acerca da ferramenta *Scratch* e um comparativo entre ela e a ferramenta que estava sendo utilizada em sala de aula (*Python*); Seção 3, experimento em que estão situados os passos realizados para a execução do mesmo e a observação dos dados obtidos com ele e seção 4 considerações finais, onde se encontra o que foi tido como consequência do trabalho e os trabalhos futuros,

2 SCRATCH

O *Scratch* é uma linguagem de programação desenvolvida pelo *Lifelong Kindergarten Group*, do *Media Lab* no *Massachusetts Institute of Technology* (MIT). Ele foi desenvolvido para trabalhar com programação em blocos. Através do *Scratch* é possível a criação de inúmeras atividades multimídias, tais como: músicas, jogos,

histórias interativas e várias outras coisas. O *Scratch* foi inicialmente desenvolvido para trabalhar com pessoas de oito aos dezesseis anos, porém ele vem sendo utilizado por pessoas de várias outras idades. (SCRATCH, 2016).

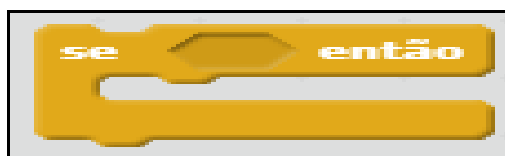
Trabalhar com o *Scratch* em sala de aula, é trazer o lúdico à metodologia de ensino aplicada, uma vez que, através dessa ferramenta, é possível pôr em prática os conceitos aprendidos de forma prazerosa utilizando os recursos de multimídias como: texto, imagens, sons, animações, etc. (SIGNIFICADOS, 2016). Nesse sentido, o *Scratch* traz a oportunidade dos alunos poderem programar e ter uma resposta mais clara e amigável por parte da ferramenta, sem as “telas pretas” das tradicionais ferramentas utilizadas para programação.

2.1 Comparando o *Scratch* com o *Python*

Esse tópico é destinado a equiparar o *Scratch* com o *Python*, que, como já citado, era a atual ferramenta utilizada para o ensino de programação na disciplina de Algoritmos.

Como já foi citado, é geralmente na disciplina de Algoritmos que os alunos têm seu primeiro contato com a programação no curso, desse modo, os conteúdos que envolvem a codificação propriamente dita (Estrutura de decisão e laços) e que foram elencados pela própria professora de Algoritmos no campo de pesquisa desse trabalho, são bastante simples e podem ser abrangidos pelo *Scratch*. Na Figura 1 está demonstrado um exemplo de uma estrutura de decisão “Se”, ou “If” como é mais comumente conhecido no meio da programação.

Figura 1 - Estrutura de decisão “Se” no *Scratch*.



Fonte: <https://goo.gl/ZRFQVL>

Uma estrutura de decisão “Se” diz que se uma condição ocorrer, o bloco de código deverá ser executado. No caso do *Scratch*, a condição deve ser colocada no espaço vazio entre as palavras “se” e “então”. Na Figura 2 encontra-se demonstrado a mesma condição em *Python*.

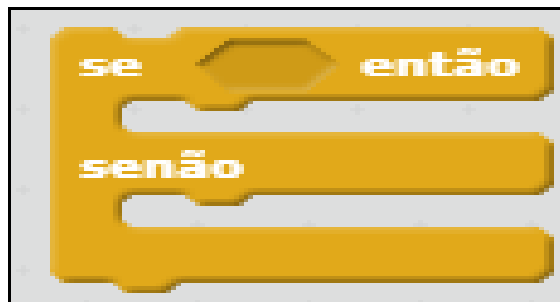
Figura 2: Estrutura de decisão “If” em *Python*

```
if <condição>:  
    <bloco de comandos indentado>
```

Fonte: <https://goo.gl/PQnpXT>

Indentação, no sentido da palavra, quer dizer que o texto tem que ser mais afastado da margem. A seguir, na Figura 3, encontra-se a imagem de um bloco referente a uma estrutura de decisão “Se - Senão” ou “If - else” no *Scratch* e na Figura 4 está demonstrado a mesma condição em *Python*:

Figura 3 – Estrutura de decisão “Se-Senão”.



Fonte: <https://goo.gl/ZRFQVL>

Figura 4 – Estrutura de decisão “If - else”.

```
if <condição>:  
    <bloco de comandos executado quando a condição é verdadeira>  
else:  
    <bloco de comandos executado quando a condição é falsa>
```

Fonte: <https://goo.gl/hGRzzf>

Esse tipo de estrutura, como já explicado na Figura 4, diz que se a condição acontecer o programa deve executar os códigos presentes dentro do “se”, caso a condição não seja satisfeita, o programa deve, necessariamente, executar os códigos presentes dentro do “senão”. Na Figura 5, encontra-se a representação das estruturas de repetição (laços) presentes no *Scratch*.

Figura 5 – Estruturas de repetição do *Scratch*.



Fonte: <https://goo.gl/ZRFQVL>

Uma estrutura de repetição diz que dada uma condição, o bloco de código deve repetir enquanto ela for satisfeita. No caso do *Scratch*, existem 3 tipos de estruturas de repetição. O “repita até que” diz que os códigos do bloco devem repetir até que tal condição seja alcançada, essa condição deve ser colocada dentro do espaço vazio logo após o “que”. O “repita vezes” repete o bloco de código de acordo com o número de vezes estipulado dentro do espaço em branco. E o “sempre” repete o bloco de código até que haja uma interrupção, como a parada do programa.

Em *Python* são utilizadas duas estruturas de repetição. O “*While*” que é usado quando não se sabe previamente quantas vezes o bloco de código deve ser repetido e o “*For*” que é usado quando já se sabe o número de vezes que o bloco de código deve ser repetido. Nas Figuras 6 e 7, respectivamente, estão ilustrados esses comandos.

Figura 6 – Estrutura de repetição “*While*”.

```
while <condição>:  
    <bloco de código indentado>
```

Fonte: <https://goo.gl/rK7CfH>

Figura 7 – Estrutura de repetição “*For*”.

```
for <var> in <lista>:  
    <bloco de código indentado>
```

Fonte: <https://goo.gl/rK7CfH>

É fácil notar que para trabalhar com o *Python* é necessário conhecer as sintaxes dos comandos e também conceitos como indentação, entre outros. Coisas que são praticamente autoexplicativas no *Scratch*, o que o torna mais fácil. Além dos conceitos apresentados, no *Scratch* ainda é possível trabalhar variáveis, listas, interação com o usuário entre muitas outras coisas que são importantes para um programador iniciante tomar conhecimento.

3 EXPERIMENTO

Até aqui, problemas significativos gerados pela reprovação em Algoritmos foram identificados. Porém, para que o trabalho possa ter seguimento, foi de substancial importância identificar quais as dificuldades os alunos sentiam para concluir a disciplina aqui trabalhada. Desse modo, para que fosse alcançado esse fim, foi realizado um questionário entre os alunos que estavam cursando a disciplina.

3.1 Primeira Etapa

Primeiramente, a coleta de dados foi feita com um público de 51 alunos da disciplina Algoritmos que se dividiam entre duas turmas nos turnos da manhã e da noite. Dentre os entrevistados, a maioria divide-se entre o primeiro e o terceiro período – o curso de Bacharelado em Computação na UEPB é constituído, na grade em vigor no momento da pesquisa, em 10 períodos -, sendo que os mais avançados se encontram no quarto período, atentando para o fato de que boa parte dos alunos já estava bem avançada no curso, mas ainda não havia concluído cadeiras do início dele. No quadro a seguir estão demonstradas as respostas dos alunos sobre qual período eles estão cursando:

Período	1º	2º	3º	4º
Qtd. de alunos nesse período.	24	3	19	5
Percentual aproximado	47,1%	5,9%	37,2%	9,8%

Quadro 1 – Quantidade de alunos por período.

O percentual mais elevado foi dos alunos que estavam cursando a disciplina pela primeira vez, porém, vale atentar para o fato de que esses alunos, em sua maioria, eram alunos que haviam entrado no curso recentemente.

Ao serem questionados sobre suas dificuldades com questões de raciocínio lógico, cerca de 60% dos entrevistados disseram sentir uma dificuldade média em relação à elas, aproximadamente 15% sentiam muita dificuldade e 25% não acreditavam sentir dificuldade. Esse dado é interessante pelo fato de que apenas um em cada quatro alunos não sentia dificuldades em resolver questões de raciocínio lógico. E todos os alunos estavam em um curso que exigia bastante desenvoltura nesse quesito.

Os alunos foram questionados sobre a área que eles julgavam sentir mais afinidade. Dentre os entrevistados aproximadamente (\approx) 79% escolheram exatas como preferencial, \approx 19% escolheram humanas e \approx 2% escolheram ambas. Mesmo a maior parte tendo escolhido exatas, \approx 65% deles sentiam de média para alta dificuldade com cálculos matemáticos. E \approx 47% sentiam de média para alta dificuldade com interpretação de texto. Este dado é interessante pelo fato de que mesmo os alunos sentindo alguma dificuldade com cálculos matemáticos, eles preferem estar em cursos relacionados com ciências exatas. Sendo assim, é importante que os professores tentem tornar suas disciplinas mais atrativas para estimular os alunos a continuar no curso.

Dos \approx 19% que escolheram humanas, \approx 90% sentiam de média para alta dificuldade com cálculos e \approx 60% deles sentiam de média para alta dificuldade com interpretação de texto. E os \approx 2% que escolheram ambas, sentiam média dificuldade com as duas. Essa parcela dos entrevistados demonstra não ter grande empatia com o curso que escolheram e isso dificulta ainda mais seu êxito na disciplina ou mesmo sua permanência no curso.

Ao serem questionados sobre se antes de ingressar no curso eles tiveram contato com algo que envolvesse programação, 76,5% responderam que não, e o restante deles tiveram contato com programação em algum curso superior, cursos *online*, 3 entrevistados trabalham com algo relacionado à programação e apenas 2 entrevistados tiveram contato com programação na escola. Isso demonstra que ainda é muito difícil encontrar escolas que incentivem o ensino da programação para seus alunos.

Quando foram perguntados se durante o curso tiveram contato com alguma disciplina de programação antes dessa, apenas 4 responderam que sim e a disciplina era P1, esse é um dado bastante peculiar, uma vez que P1 é uma disciplina ofertada após Algoritmos e necessita dos conhecimentos obtidos nessa cadeira para um bom aprendizado.

Ao serem perguntados qual o número de vezes que cursaram a disciplina, 2 não quiseram responder. Dos que sobraram $\approx 50\%$ respondeu que é a primeira vez, mas esse número coincide com a quantidade de alunos que estão cursando o primeiro período. E o restante respondeu que não é a primeira vez e suas respostas variaram entre segunda e quinta vez. Esse dado evidencia a necessidade desse trabalho, uma vez que não pode ser considerado comum um aluno cursar a mesma disciplina por mais de quatro vezes sem conseguir êxito.

Foi perguntado se havia dificuldades relacionadas à linguagem de programação utilizada na disciplina (*Python*), $\approx 35,3\%$ responderam que sim. Suas dificuldades envolviam: utilização do programa, idioma utilizado pelo programa (inglês), a quantidade de comandos, dificuldades com as estruturas de decisão (*If*, *Elif*, *Else*), praticidade da linguagem e a utilização de muitos cálculos.

Quando foi questionado sobre qual a maior dificuldade encontrada por parte dos alunos, maior parte deles respondeu que sentem dificuldade de como interpretar a questão. Uma parte equivalente sente dificuldade com a metodologia utilizada pela professora. E um ponto a se considerar foi o de alguns não terem tempo para praticar em casa e se sentir tímidos para tirar dúvidas na frente dos outros alunos.

Em resumo, de acordo com o questionário que foi aplicado, as maiores dificuldades dos alunos consistiam em: Interpretação de texto (a principal delas), linguagem de programação utilizada (seja idioma, sintaxe, entre outros) e cálculos. Essas dificuldades já eram esperadas, pois são elas as mais comentadas entre os professores que ensinam programação nesse curso.

Assim sendo, trabalhar com o *Scratch* poderia trazer bons resultados, uma vez que a linguagem utilizada é mais próxima do usuário, além de ser em português. Ao inserir o *Scratch* em sala de aula, o professor que estiver ministrando essa disciplina irá trazer uma metodologia inovadora para os padrões utilizados. Além disso, poderá trazer maior interação e motivação para os alunos em sala de aula. Sendo verificadas essas melhorias, será mais fácil para os alunos concentrar os esforços em interpretação de textos, cálculo ou qualquer dificuldade particular.

3.2 Segunda Etapa

Depois de identificados os problemas secundários (metodologia utilizada, dificuldades com a linguagem, entre outros), foi ofertada uma oficina para os voluntários que queriam seguir com a pesquisa. Essa oficina teve como intuito familiarizar o *Scratch* dentro dos problemas, ou seja, mostrar como o *Scratch* poderia auxiliar o aprendizado dos alunos acerca dos problemas encontrados.

A oficina se estendeu por três encontros, no primeiro momento foi apresentada para os alunos a programação em blocos, suas funções e os objetivos de trabalhar ela em sala de aula. Posteriormente, foi falado um pouco sobre a ferramenta *Scratch*.

No segundo encontro, foi apresentada a ferramenta *Scratch* para os alunos. Para começar a trabalhar com a ferramenta, foram requisitadas à que era atual professora de Algoritmos questões que ela havia trabalhado ou pretendia trabalhar em sala para que fossem postas em prática na oficina.

As questões foram selecionadas de modo que pudesse abranger os conteúdos considerados mais importantes pela professora. Os conteúdos em questão eram: criação de variáveis, estrutura de decisão e laços de repetição. Como já era de se esperar os alunos acharam muito fácil trabalhar com a ferramenta e a programação em blocos.

O terceiro encontro serviu para inserir o fluxograma como auxílio para a estruturação de um código no *Scratch* e dar continuidade aos exercícios. Primeiramente o código era estruturado no fluxograma e depois elaborado no *Scratch*.

3.3 Terceira Etapa

A terceira etapa consistiu em aplicar o segundo questionário com os alunos que continuaram participando da pesquisa, configurando 30 voluntários. A partir de agora a pesquisa teve seu foco voltada para identificar o que os alunos tinham de conhecimento prévio da ferramenta, da metodologia utilizada para ensinar a disciplina e qual a opinião deles sobre a inserção da ferramenta como auxílio.

Ao serem questionados se eles já tinham conhecimento acerca do *Scratch* apenas cinco pessoas responderam que sim. Para os alunos que responderam que

sim foi feita uma pergunta extra, sobre onde e quando eles haviam tido contato com o *Scratch*, um disse que foi na escola e o restante não soube informar onde haviam tido contato com a ferramenta.

Mais uma vez foi explicado que a ferramenta havia sido proposta no intuito de ajudar os alunos em concluir com êxito a disciplina, após isso foi perguntado se eles concordavam com a proposta e se acreditavam que ela realmente poderia ajudá-los, com exceção de um aluno que se absteve de responder a pergunta, todos concordaram e suas repostas variavam entre dizer que a ferramenta facilitava no entendimento (maior índice de respostas), achavam a ferramenta simples e também frisaram que a ferramenta deveria ser inserida na disciplina. Ou seja, a proposta de inserir o *Scratch* como ferramenta auxiliadora teve uma aceitação de aproximadamente 96,66%.

Sendo assim, ao final desse segundo questionário pode-se dizer que os alunos estão dispostos a aceitar o *Scratch* como uma ferramenta capaz de auxiliá-los na obtenção dos conhecimentos necessários para obter êxito na disciplina de Algoritmos. Não como a solução de todos os problemas, mas como uma ferramenta capaz de facilitar sua trajetória na disciplina.

4 CONSIDERAÇÕES FINAIS

A pesquisa nasceu da necessidade de encontrar soluções para amenizar o índice de reprovação na disciplina. Fato comprovado por pesquisas bibliográficas e nessa pesquisa, na qual aproximadamente 50% dos voluntários eram de alunos que já reprovaram na disciplina, pois como constatado, além desse alto índice de reprovação sobrecarregar a disciplina (no que se diz respeito à quantidade de alunos), ele ainda pode influenciar negativamente o índice de evasão no curso.

Então, a ferramenta *Scratch* foi proposta como uma solução para esse problema. Contudo, ao realizar a pesquisa bibliográfica foi identificado que a utilização da ferramenta por si só não era o bastante para fazer com que os alunos conseguissem conceito suficiente para conseguir êxito na disciplina. Porém a utilização da ferramenta é considerada uma ótima estratégia no que se diz respeito à motivação, metodologia diversificada e interação em grupo, e esses quesitos foram

identificados como problemas dentre os voluntários, ou seja, estavam em falta segundo os entrevistados.

Desse modo, foi aplicado aos alunos um questionário para saber a aceitação do *Scratch* como uma ferramenta capaz de auxiliá-los na aquisição dos conhecimentos necessários para obter aprovação na disciplina, não como solução, mas como uma ferramenta capaz facilitar o processo de aprendizado, uma vez que ela é mais fácil de utilizar do que as tradicionais e seus esforços estariam mais voltados aos problemas individuais, como resolução de cálculos, interpretação de texto, etc.

O resultado obtido com a pesquisa apontou alto índice de aceitação da ferramenta por parte dos alunos (96,66%). Desse modo, como trabalho futuro almeja-se a produção de um material instrucional para auxiliar o professor em suas aulas e os alunos na compreensão e utilização do *Scratch*. Posteriormente, verificar com a professora se utilizando a ferramenta os alunos foram capazes de melhorar seu desempenho e buscar estratégias para combater diretamente as outras deficiências elencadas nessa pesquisa tais como, interpretação de texto e resolução de cálculos.

ABSTRACT

In programming courses it is common to find classes of Algorithms with overcrowding. In the research that was carried out to compose this work, one can identify that there is a high degree of failure among the veterans, a fact that can subsidize this overcrowding. This work deals exactly that, it takes into account either the fact that the high rate of disapproval may discourage students from remaining in the course. And to try to reverse the reproach, it was suggested to insert Scratch as a tool to support the students, since the use of this tool brings the opportunity of learning in a playful way, opposing the more traditional ones in which students only interact with an unfriendly interface. To support this proposal, a quantitative research was carried out on the students' opinion about it. In order to do so, an initial research was necessary to identify the problems in the students' opinion. Subsequently, a workshop was held situating Scratch in the discipline and finally the last research was carried out to identify the acceptance of the tool. As a result, it reached an

acceptance of approximately 96.66% for the tool, that is, approximately 96.66% of the students interviewed accepted the proposal.

Keywords: Scratch. Algorithms. Ludic.

REFERÊNCIAS:

CAMPOS, Ricardo L. B. L. Metodologia ERM2C: Para melhoria do processo de ensino-aprendizagem de lógica de programação. In: XVII *Workshop Sobre Educação em Computação*, 17., 2010, Minas Gerais. **Anais**. Minas Gerais: PUCMINAS, 2010.

FALKEMBACH, Gilse A. M.; et al. Aprendizagem De Algoritmos: Uso Da Estratégia Ascendente De Resolução De problema. In: Taller Internacional de Software Educativo, 8, 2003, Santiago. **Anais**. Santiago: Universidad de Chile. Disponível em: <<http://docplayer.com.br/3981338-Aprendizagem-de-algoritmos-uso-da-estrategia-ascendente-de-resolucao-de-problemas.html>>. Acesso em 06 de mai. de 2016.

GIL, Antônio C.. **Como Elaborar Projetos de Pesquisa**. 4. Ed. São Paulo: Atlas. 2002. Disponível em: <http://www.academia.edu/4405328/GIL_Antonio_Carlos_COMO_ELABORAR_PROJETOS_DE_PESQUISA_Copia>. Acesso em: 10 de nov. 2014.

GONÇALVES, Ricardo E. **Jogo Digital para o Ensino dos Fundamentos da Programação**. 2011. 73 f. Dissertação (Mestrado) - Faculdade de Engenharia, Universidade do Porto. Portugal. 2011. Disponível em: < <https://repositorio-aberto.up.pt/bitstream/10216/62088/1/000149236.pdf> >. Acesso em: 06 de mai. de 2016.

GRÜBEL, Joceline M.; BEZ, Marta R. Jogos educativos. **Revista Renote**, Rio Grande do Sul. v.4 , n. 2, dez. 2006. Disponível em: <<http://seer.ufrgs.br/renote/article/view/14270>>. Acesso em: 06 mai. de 2016.

LAGE, Bruno. **Aprendendo a Programar em Python – Introdução**. 2013. Disponível em: < <http://www.devmedia.com.br/aprendendo-a-programar-em-python-introducao/17093>>. Acesso em: 26 de abr. 2017.

LIMA JÚNIOR, José A. T. L.; et al. Dificuldades no processo de aprendizagem de Algoritmos: uma análise dos resultados na disciplina de AL1 do Curso de sistemas de Informação da FAETERJ - Campus Paracambi. **CADERNOS UniFOA**. Rio de Janeiro, n. 27, mai. 2015. Disponível em: <<http://web.unifoa.edu.br/cadernos/edicao/27/5-15.pdf> >. Acesso em: 26 de abr. 2017.

RAPKIEWICZ, Clevi Elena; et al. Estratégias pedagógicas no ensino de algoritmos e Programação associadas ao uso de jogos educacionais. **CINTED**, Rio Grande Do Sul, v.4, n. 2, dez. 2006. Disponível em:

<<http://seer.ufrgs.br/renote/article/download/14284/8203> >. Acesso em: 26 de abr. 2017.

SALAZAR, Rafael; ODAKURA, Valguima; BARVINSK, Carla. Scratch no ensino superior: motivação. In: XXVI Simpósio Brasileiro de Informática na Educação, 26., 2015, Alagoas. **Anais do XXVI Simpósio Brasileiro de Informática na Educação**. Maceió, 2015.

SCRATCH. **Perguntas Frequentes**. 2016. Disponível em: <<https://scratch.mit.edu/info/faq/>>. Acesso em: 06 de mai. de 2016

SIGNIFICADOS. **O que é Lúdico**. 2016. Disponível em: <<http://www.significados.com.br/ludico/>>. Acesso em: 06 de mai. de 2016.

SILVA, Edna L.; MENEZES, Estera M.. **Metodologia da pesquisa e elaboração de dissertação**. 4. Ed. Florianópolis: UFSC, 2005. Disponível em: <https://projetos.inf.ufsc.br/arquivos/Metodologia_de_pesquisa_e_elaboracao_de_teses_e_dissertacoes_4ed.pdf>. Acesso em: 24 de abr. 2016.

SIMAS, Anna. **As graduações Campeãs de desistência**. 2012. Disponível em: <<http://www.gazetadopovo.com.br/educacao/vida-na-universidade/ufpr/as-graduacoes-campeas-de-desistencia-26khijqy1gurtas1veawhyz2>>. Acesso em: 30 de abr. de 2016.

SOUSA, José N. M.; et al. **Principais fatores que impactam na reprovação e evasão dos alunos dos cursos tecnológicos no instituto federal de ciência e tecnologia do ceará**. 2015. Disponível em: <<http://www2.ipece.ce.gov.br/encontro/2015/trabalhos/PRINCIPAIS%20FATORES%20QUE%20IMPACTAM.pdf>>. Acesso em: 06 de mai. de 2016.

SOUZA, Marcelo; JAEGER, Eliana V.; CARDOSO, Brigiane M. S. Ensino De Algoritmos Apoiado Pelo Uso De Jogos Digitais Educativos. **Novas Tecnologias na Educação**, Rio Grande do Sul, v. 11, n. 3, dez. 2013.

VERONEZZI, Felipe. **Ciência da Computação: curso, carreira e mercado**. 2010. Disponível em <<http://www.guiadacarreira.com.br/profissao/ciencia-da-computacao/>>. Acesso em 03 de mai. de 2016.

APÊNDICE A – PRIMEIRO QUESTIONÁRIO



UEPB

Este Questionário é de cunho acadêmico, cujo objetivo é entender um pouco mais sobre as dificuldades dos alunos na disciplina de Algoritmos.

Aluno: Victor Anderson Fortunato dos Santos

Orientador: Kézia De Vasconcelos Oliveira

Prezado aluno(a), sou graduando do curso Licenciatura em Computação, 8º período, da Universidade Estadual da Paraíba (UEPB), estou fazendo uma pesquisa que será fonte de dados no projeto de conclusão de curso. Preciso da sua colaboração para o mesmo, preenchendo esse questionário. Com base nesses questionamentos será possível identificar características relevantes para compreender as dificuldades em concluir com mérito a disciplina de Algoritmos.

-Nome (opcional): _____

-Em que ano concluiu o ensino médio _____

-Cursou o ensino fundamental em escola:

Pública () Privada () Ambas ()

-Cursou o ensino Médio em escola:

Pública () Privada () Ambas ()

-Ano e período que entrou no curso (ex.: 2014.2): _____

-Período em que se encontra no curso: _____

-Você se considera hábil em raciocínio lógico?

Muito () Bastante () Razoável () Pouco () Muito pouco () Não ()

-Com qual área você se identifica melhor?

() Humanas () Exatas. Por qual motivo?

-Você tem dificuldade com disciplinas que envolvem cálculos?

Muito () Bastante () Razoável () Pouco () Muito pouco () Não ()

-Você tem dificuldade com interpretação de texto?

Muito () Bastante () Razoável () Pouco () Muito pouco () Não ()

-Você teve contato com algo que envolvesse programação antes de ingressar no curso?

Sim () Não () se sim, onde? _____

-Durante o curso você já teve contato com alguma disciplina de programação antes dessa?

Sim () Não () se sim, qual(is)?

-Essa é a primeira vez que você tenta pagar essa disciplina?

Sim () Não () se não, então qual é? _____

-Você sente dificuldade com a sintaxe da linguagem utilizada na disciplina?

Sim () Não () por qual motivo?

-Qual a maior dificuldade que você encontra nessa disciplina?

Obs.: Leve em consideração quesitos como: semântica, detalhamento no entendimento das questões, abstração, entre outros.

APÊNDICE B – SEGUNDO QUESTIONÁRIO



UEPB

Este Questionário é de cunho acadêmico, cujo objetivo é apurar os resultados e opiniões sobre as ferramentas apresentadas para procurar diminuir as dificuldades dos alunos na disciplina de Algoritmos.

Aluno: Victor Anderson Fortunato dos Santos

Orientador: Kézia De Vasconcelos Oliveira

Prezado aluno(a), sou graduando do curso Licenciatura em Computação, 8º período, da Universidade Estadual da Paraíba (UEPB), estou fazendo uma pesquisa que será fonte de dados no projeto de conclusão de curso. Necessito da sua colaboração para o mesmo, preenchendo esse questionário. Com base nesses questionamentos será possível identificar se as ferramentas propostas poderão ajudar no andamento e conclusão dessa e das próximas disciplinas de Algoritmos.

-Nome (opcional): _____

-Essa é a primeira vez que você tenta pagar essa disciplina?

Sim () Não () se não, então qual é? _____

Você já tinha conhecimento das ferramentas propostas?

() Sim, conhecia as duas () Sim, mas apenas Scratch () Sim, mas apenas o Fluxograma () Não

Caso tenha respondido “Sim” na questão anterior, responda essa. Caso tenha respondido “Não”, desconsidere essa questão. Quando você usou essa(s) ferramenta(s)? _____

As ferramentas foram propostas no intuito de ajudar os alunos em interpretação de texto e em concluir com êxito a disciplina. Você acha que a proposta é válida, ou seja, ela realmente pode ajudar os alunos nesse quesito?

Sim. Não. Mas por quê?

Você acredita que o Fluxograma é uma ferramenta que pode auxiliar na compreensão de uma questão e na elaboração de um código?

Sim. Não, por quê?

Você acha que a utilização do Scratch e da programação em blocos pode ser um aliado no quesito que diz respeito à interpretação de texto, uma vez que, utilizando a programação em blocos você se preocupa menos com a linguagem e mais com o que a questão está pedindo?

Sim Não

Você acha que, a partir de um Fluxograma, é mais fácil implementar uma questão no Scratch?

Sim. Não, por quê?
