

Metodologias, Técnicas, Ambientes e Tecnologias Alternativas utilizadas no Ensino de Algoritmos e Programação no Ensino Superior no Brasil

Ronney Moreira de Castro, Sean Wolfgang Matsui Siqueira

Programa de Pós-Graduação em Informática (PPGI)

Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

Av. Pasteur, 456 - Urca - Rio de Janeiro - RJ - Brasil

ronney.castro@uniriotec.br, sean@uniriotec.br

***Abstract.** This paper aims to explore alternative methodologies, techniques, environments and technologies for Algorithms and Programming classes. Then, a literature mapping was performed in order to identify the alternatives that are used to teach these subjects. The paper also provides more detailed information on methodologies, techniques, environments and technologies, and illustrates their applicability, showing that it is possible to diversify the classes and allow a greater involvement of students with the content taught.*

***Resumo.** O presente trabalho tem o objetivo de explorar metodologias, técnicas, ambientes e tecnologias alternativas para aulas de Algoritmos e Programação. Para tanto, foi realizado um mapeamento da literatura com o objetivo de identificar alternativas que são usadas para ensinar essas disciplinas. O artigo traz também informações mais detalhadas das metodologias, técnicas, ambientes e tecnologias, e ilustra sua aplicabilidade, mostrando que é possível diversificar as aulas e permitir um maior envolvimento dos alunos com o conteúdo lecionado.*

1. Introdução

As disciplinas de Algoritmos e Programação estão presentes em quase todos os currículos de cursos superiores da área da Computação [Zorzo et al. 2017]. Em geral, elas são lecionadas nos primeiros períodos, mas a primeira experiência com programação pode acabar resultando em um grande índice de reprovações e, até mesmo, desistências da disciplina e/ou evasão do curso [Sirotheau et al. 2011]. Segundo pesquisas, as taxas mundiais de evasão em cursos da área de Tecnologia da Informação (TI) estão na casa dos 19% a 40% [Mooney et al. 2010] [Hüsing et al. 2013]. No Brasil, o cenário também não é diferente. Estudos mostram uma taxa média de evasão igual ou superior a 40% [Palmeira; Santos 2014] [Borba 2015] [Veronese; Lemos 2015] [Ferreira 2015]. Os dados do Censo do Ensino Superior Brasileiro¹, publicado pelo INEP/MEC em 2017, mostram que os cursos da área da Computação sofreram com cerca de 31% de evasão sendo considerados, juntamente com os cursos de Matemática, aqueles que possuem o maior índice de evasão.

O modelo de aula mais utilizado por professores de cursos de graduação é a expositiva [Gudigantala 2013]. Isso se deve muito ao fato dos próprios docentes terem aprendido com ela durante sua formação. Entretanto, diante desse cenário de

¹ <http://portal.mec.gov.br/docman/setembro-2018-pdf/97041-apresentac-a-o-censo-superior-u-ltimo/file>

reprovações, desistências e/ou evasão, um número crescente de estratégias alternativas vem ganhando destaque no que diz respeito a obter a atenção dos alunos, mudando a forma tradicional (aula expositiva) de ensino e aprendizagem, tais como: dinâmicas [Ribeiro; Manso; Borges 2016], jogos [Sousa Pires et al. 2018], ambientes de apoio [Almeida et al. 2002], ambientes colaborativos [Tobar et al. 2001], ferramentas de auxílio à construção da lógica [Hostins; Raabe 2007], modelo lúdico [Santiago; Krounbauer 2017], gamificação [Maekawa; Nagai; Izeki 2015], uso de robótica [Ávila; Cavalheiro 2017], entre outras.

Este artigo traz como contribuição um Mapeamento realizado nos principais eventos relacionados à Educação em Computação e Informática na Educação no Brasil nos últimos seis anos. O objetivo é apresentar metodologias, técnicas, ambientes e tecnologias alternativas que podem ser utilizados no ensino de Algoritmos e Programação. O trabalho mostra como está o atual cenário de pesquisa desta temática no país. Optou-se por restringir o escopo da pesquisa ao contexto Brasil pelo fato de Ensino ter um forte componente cultural. Além disso, possibilita entender o que efetivamente tem sido feito e publicado para possibilitar posteriormente o desenvolvimento de estratégias que apoiem neste contexto.

O artigo inicia com essa introdução e logo após são apresentadas mais 4 seções. A Seção 2 descreve o Mapeamento realizado, a Seção 3 traz uma descrição das metodologias e técnicas para ensino de Algoritmos e Programação, a Seção 4 discorre sobre os ambientes e tecnologias para ensino de Algoritmos e Programação e, por fim, as conclusões são apresentadas na Seção 5.

2. Mapeamento de Literatura

Para este trabalho, foi conduzido um Mapeamento da Literatura (ML) com objetivo de verificar metodologias, técnicas, ambientes e tecnologias alternativas para o ensino de Algoritmos e Programação no contexto brasileiro, mais especificamente responder a seguinte questão de pesquisa (QP): “*Que metodologias, técnicas, ambientes e tecnologias alternativas são utilizadas para o ensino de Algoritmos e Programação no ensino superior no Brasil?*”. Como a QP se restringe ao universo brasileiro, a pesquisa foi realizada através de busca nos anais dos principais eventos do país relacionados à Informática na Educação e Ensino de Computação: Workshops do CBIE (Congresso Brasileiro de Informática na Educação), incluindo o WalgProg (Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação); SBIE (Simpósio Brasileiro de Informática na Educação); WEI (Workshop sobre Educação em Computação) e WIE (Workshop de Informática na Escola). Além dos eventos, também foi incluída a RBIE (Revista Brasileira de Informática na Educação). Para que fossem encontradas as metodologias, técnicas, ambientes e tecnologias alternativas mais atuais utilizadas por docentes em todo Brasil, foram escolhidos para o mapeamento os artigos dos anos 2013, 2014, 2015, 2016, 2017 e 2018.

Para a seleção dos trabalhos adotou-se o seguinte processo: i) Nos anais dos eventos/revista feita a leitura do título, palavras-chave e resumo de cada um dos trabalhos. Foram aplicados os critérios de inclusão e exclusão do Quadro 1, com objetivo de selecionar aqueles que estavam relacionados a metodologias, técnicas, ambientes e tecnologias alternativas de ensino de Algoritmos e Programação. O Quadro 2 exhibe a quantidade de trabalhos selecionados (79 artigos); ii) Feita a leitura completa dos trabalhos e iii) Durante o processo de leitura, foi sendo feita também a identificação

e classificação das metodologias, técnicas, ambientes e tecnologias (23 itens de classificação).

Quadro 1: Critérios de exclusão e de inclusão.

Critérios de inclusão	Critérios de exclusão
<ul style="list-style-type: none"> • Uso em Ensino Superior; • Técnicas para auxiliar no entendimento de Algoritmos e Programação; • Análise do aprendizado dos alunos em Algoritmos e Programação; • Qualquer metodologia, técnica, ambiente e/ou tecnologia cujo o foco seja ensinar Algoritmos e Programação. 	<ul style="list-style-type: none"> • Artigos curtos; • Ferramentas com foco em auxiliar alunos com Deficiência; • Ferramentas que são destinadas a um único sexo (feminino e masculino); • Trabalhos que estejam relacionados a culturas específicas e/ou etnias (indígena, branco, negro etc.).

É importante destacar que o critério de exclusão “Trabalhos que estejam relacionados a culturas específicas e/ou etnias (indígena, branco, negro etc.)” foi adotado, pois existem trabalhos exclusivos contendo metodologias, técnicas, ambientes e tecnologias para estas especificidades e que não podem ser utilizados para todos os tipos de alunos.

A Tabela 1 ilustra na coluna “Tot.,” a quantidade de trabalhos em cada ano. A coluna “Sel.” exhibe a quantidade de trabalhos selecionados. Os totais podem ser verificados nas últimas colunas, onde “Geral” mostra o total de artigos (2608 trabalhos) e “T.Sel.,” o total de artigos que foram selecionados no Mapeamento (79 trabalhos).

Tabela 1: Total de trabalhos nos eventos e na revista RBIE.

Evento/Ano	2013		2014		2015		2016		2017		2018		Totais	
	Tot.	Sel.	Geral	T.Sel.										
WCBIE	91	1	85	2	153	12	159	3	152	1	128	8	768	27
RBIE	30	0	33	0	44	0	33	0	17	0	20	1	177	1
SBIE	114	0	162	3	140	2	133	5	172	0	171	8	892	18
WEI	54	5	44	3	45	4	47	6	25	1	52	6	267	25
WIE	52	0	71	0	67	0	104	1	133	7	77	0	504	8
TOTAL	341	6	395	8	449	18	476	15	499	9	448	23	2608	79

Após a análise dos resultados do ML, foram constatadas iniciativas para ensino de Algoritmos e Programação, categorizadas nas duas próximas seções desse trabalho. São exibidas informações para os docentes que queiram utilizá-las em um quadro e também discutidos alguns possíveis usos. A coluna “Especificação” exhibe uma descrição da iniciativa, além da quantidade de artigos e sua referência, entre parêntesis, na planilha do link <<https://is.gd/fif4WA>>. Alguns textos dessa coluna foram retirados dos próprios artigos do ML. É importante destacar também que alguns trabalhos utilizam mais de uma iniciativa.

3. Metodologias e Técnicas Alternativas para o Ensino de Algoritmos e Programação

Um método está relacionado ao caminho para se chegar a um determinado objetivo. No cotidiano, as pessoas possuem objetivos, que não se realizam por si mesmos, ou seja, é necessária uma sequência de passos para atingi-los. Um pesquisador, por exemplo, faz uso de métodos para atingir determinado objetivo de sua pesquisa. Em se tratando de ensino, o docente utiliza um conjunto de ações ou procedimentos de forma a dirigir e estimular o processo de ensino, o que é conhecido por método de ensino. Um docente pode adotar, por exemplo, o método “aula expositiva” para lecionar o conteúdo da

disciplina “Algoritmos”. A metodologia envolve o método, ou seja, é o processo que será adotado no desenvolvimento do ensino. Aliado à metodologia, existem as técnicas, que subsidiam o docente de forma a tornar o ensino mais eficaz. A técnica está relacionada à prática. Resumidamente, a metodologia está relacionada ao caminho e a técnica a forma de percorrer tal caminho [Libâneo 2017] [Furtado 2018]. O Quadro 2 ilustra metodologias e técnicas alternativas que podem ser utilizadas por docentes para o ensino de Algoritmos e Programação.

Quadro 2: Metodologias e técnicas alternativas encontradas no ML

Nº	Metodologia/Técnica	Especificação
1	Aprendizagem Baseada em Problemas	Permite desenvolvimento técnico dos participantes e também os capacita em outros atributos como comunicação, trabalho em equipe, resolução de problemas, autodidatismo, compartilhamento de conhecimento e informações, argumentação e respeito às divergências. (A08, A09, A10)
2	Aprendizagem Significativa	Nesse processo de aprendizagem, o conhecimento existente fica mais sólido, diferenciado e elaborado em termos de significados, adquirindo mais estabilidade e se internalizando por mais tempo. (A11, A12)
3	Avaliação Por Pares	Instrumento de avaliação educacional que permite alcançar objetivos pedagógicos diversos, fazendo parte de uma avaliação formadora em que aqueles que aprendem são construtores do seu próprio conhecimento. Enfatiza a aprendizagem colaborativa, ao permitir que os estudantes se avaliem mutuamente e, ao ter no retorno uma das suas dimensões estruturantes, contribui para dinamizar o diálogo entre o professor e estudantes, além de estimular o debate entre eles e melhorar suas capacidades críticas e argumentativas. Ou seja, os alunos também aprendem enquanto avaliam uns aos outros. (A17)
4	Coding Dojo	Metodologia que propõe encontros para que pessoas ganhem prática e conhecimento em alguma tecnologia previamente determinada, como programação. Existem três formas de se aplicar uma sessão de Coding Dojo (Randori, Kata e Kake), que divergem entre si quanto a forma de se organizar e de se aplicar. Entretanto o objetivo final é o mesmo: aprender e treinar um grupo de pessoas a respeito de algum conceito ou tecnologia. (A18, A19)
5	Computação Desplugada	Visa ensinar os fundamentos da computação de forma lúdica, sem o uso de computadores, sem distrações e detalhes técnicos em demasia. Pode ser aplicada nos vários tipos de ensino, com diferentes conhecimentos e experiências. É mais do que não utilizar ferramentas computacionais, mas envolve questões metodológicas de como fazê-lo. (A20, A21, A22)
6	Dinâmica de Grupo	Dinâmicas de grupo possuem o objetivo de colocar os alunos frente a uma situação ou problema e a partir disso, conseguir deles a abstração para situações mais amplas. (A23)
7	Gamificação	Uso de elementos e técnicas de projeto do jogo em contextos não-jogo. Os elementos são os atributos físicos e lógicos que ajudam a manter e informar os jogadores sobre o estado do jogo. Oferecem metas ou objetivos; pontuações: forma de recompensa; tabelas de classificação: representam as atividades que estão sendo executadas. (A44, A45, A46, A47)
8	Hackaton	Evento de programação em Computação focado em problemas. São instrumentos importantes para encorajar a experimentação e criatividade, aspectos essenciais à inovação. Consiste de um conjunto de características muito particulares, como uma interação intensa durante períodos cursos, por exemplo. (A48)
9	Jogo	A utilização de jogos permite ao aluno visualizar e experimentar

		conceitos abstratos, além de despertar a criatividade. Aqui são incluídos jogos de uso geral, diferentes daqueles desenvolvidos com propósito educacional específico. (A50, A51, A52, A53, A54, A55, A56, A57, A58, A59)
10	Jogo Sérió	Jogos desenvolvidos com princípios de design de jogo interativo e objetivo de transmitir um conteúdo educacional ou de treinamento ao usuário que vai além do entretenimento, ou seja, têm uma mensagem a transmitir. (A60, A61, A62)
11	Sala de Aula Invertida	Abordagem que possui o propósito de instituir uma mudança sistemática na sala de aula, em que o aluno tem uma postura proativa estudando os fundamentos teóricos de um conteúdo antes da aula; em sala de aula, poderá praticar o mesmo. Gera uma descentralização da atenção do professor para o estudante e, por consequência, aumenta-se o envolvimento do aluno. (A73, A74)

A **Aprendizagem Baseada em Problemas** tem como objetivo central trabalhar a compreensão de um determinado problema, em geral, relacionado à vida real. O aluno vai analisando, compreendendo, encontrando uma solução adequada para tal problema e, com isso, compreende melhor o conteúdo relacionado. O problema pode ser passado aos alunos no início de um semestre e ser trabalhado ao longo deste por eles. Também pode-se passar problemas para grupos distintos de alunos e dar um tempo para que sejam resolvidos. Logo após, solicitar que um grupo diferente analise, veja os erros e deixe sugestões de soluções de outro grupo. Na **Aprendizagem Significativa** os alunos devem possuir certo conhecimento e continuar a desenvolvê-lo, para que fique sólido e diferenciado. Um exemplo é mostrar aos alunos o comando “for”. Mesmo aqueles que não possuem conhecimento de programação, sabem o que é uma repetição e, pode-se associar o conteúdo a coisas da vida cotidiana. Em seguida, em um código, mostrar aos alunos o laço de repetição e sugerir que ele seja alterado para n repetições.

A **Avaliação por Pares** também permite que os alunos possam corrigir problemas de outros colegas. Por exemplo: verificar como o código foi indentado, se todas as variáveis foram inicializadas etc. A **Sala de Aula Invertida** consiste em inverter a ordem da aula, ou seja, passar o conteúdo antes para os alunos e discuti-lo depois na sala de aula. Essa técnica já é utilizada a tempos. Desde o ensino fundamental, os professores passavam leituras que deveriam ser realizadas em casa, antes da aula para, depois, serem discutidas por todos em sala. Essa técnica pode auxiliar muito na interação entre os alunos. Pode-se passar anteriormente para os alunos uma leitura de artigo, um vídeo etc e, no dia da aula, fazer uma discussão sobre o assunto. Pode-se também solicitar aos alunos que tragam mais informações, ou até mesmo notícias recentes, sobre o assunto para serem discutidos com todos.

A **Computação Desplugada** é uma opção para se ensinar conceitos de Algoritmos e Programação sem o uso de tecnologia. Várias dinâmicas podem ser construídas. Um exemplo é o “Contando os Pontos”, do livro “*Computer Science Unplugged*” [Bell et al. 2009]. As **Dinâmicas de Grupo** também podem auxiliar, tendo em vista que envolvem os alunos em atividades lúdicas em sala. Um exemplo é os alunos lerem *QR Codes* em sala e nos corredores de forma a resolver determinado exercício.

Os **Jogos** já fazem parte da vida cotidiana dos alunos. Podem ser executados em um computador *Desktop* e também nos *Smartphones*. Além disso, existem outros tipos de jogos que não necessitam de tecnologia, como os jogos de tabuleiro. Um exemplo de utilização é o “Desafio da Serpente” [Raposo; Dantas 2016], um jogo *online* que envolve conceitos de Algoritmos e Programação e, além disso, usa a técnica de

Gamificação para “premiar” os participantes. O *Jogo Séri*o se diferencia dos demais por atender a objetivos estritamente educacionais. Os alunos podem desenvolver um jogo com essa finalidade ou, até mesmo, utilizar de outros que já existem. No jogo sério, os alunos focam mais em tarefas instrucionais e não nas mecânicas e demais elementos do jogo. A *Gamificação* usa recursos e elementos do mundo dos jogos, de forma a motivar, engajar e modificar, de forma positiva, o comportamento dos indivíduos. Seu uso com os alunos pode ser feito, por exemplo, através de um jogo *online* com perguntas e respostas que devem ser respondidas diariamente pelos discentes. Cada acerto o aluno ganha um prêmio (medalha). Ao final é possível verificar qual aluno obteve o maior número de prêmios. Além disso, também servirá de análise para cada aluno verificar onde falhou e o onde deve melhorar. Pode-se mesclar essa técnica ao AVA.

Hackatons são eventos onde os participantes têm que cumprir tarefas em um período ininterrupto, ao estilo maratona. O docente que deseja usar essa técnica pode, por exemplo, montar o evento com tarefas para os alunos resolverem em um dia inteiro. Os alunos devem formar equipes, o que também permite uma maior interação entre eles. No *Coding Dojo* o objetivo principal é o compartilhamento de experiências entre os participantes, ou seja, a aprendizagem é colaborativa. Permite também uma maior interação entre os indivíduos. Pode ser utilizado para o ensino de algoritmos já que os alunos estarão trocando experiências próprias com os demais colegas.

4. Ambientes e Tecnologias para o Ensino de Algoritmos e Programação

Um ambiente de ensino e aprendizagem deve propiciar aos discentes, formas de desenvolver sua posição no processo de ensino, ou seja, o ambiente deve proporcionar meios para favorecer a aprendizagem e também a socialização. Entre os componentes de um ambiente de ensino e aprendizagem estão os humanos (docente, orientadores educacionais, tutores etc) e os materiais (sala de aula, quadro, *Datashow* etc) [Piletti 2000]. A tecnologia vem ganhando grande importância na sociedade contemporânea, influenciando desde governos, empresas, até os próprios indivíduos de forma a alcançar seus objetivos. Na educação, o uso da tecnologia também vem crescendo e, fazer o uso desta como aliada nas atividades de ensino, pode garantir uma maior aprendizagem dos discentes [Furtado 2018]. O próprio *Smartphone*, por exemplo, pode ser utilizado como ferramenta no processo de ensino e aprendizagem. O Quadro 3 ilustra ambientes e tecnologias que foram utilizados por docentes para o ensino de Algoritmos e Programação, descritos nos trabalhos selecionados no mapeamento de literatura.

Quadro 3: Ambientes e tecnologias encontradas no ML

Nº	Ambiente/Tecnologia	Especificação
1	Ambiente Virtual de Aprendizagem	Ambiente online considerado uma extensão da sala de aula presencial. É um elemento basilar da relação com os alunos. (A01, A02, A03, A04, A05)
2	App Inventor	Ambiente visual de programação em blocos que permite o desenvolvimento de aplicativos para dispositivos móveis Android de maneira simples, principalmente se comparado às linguagens de programação tradicionais. (A06, A07)
3	Arduino	Plataforma de prototipagem de código aberto baseada em hardware e software de fácil utilização. Possui um vasto rol de aplicações que vai tanto de simples protótipos eletrônicos a grandes automações industriais. (A13, A14)
4	Áudio/Vídeo	Uso de recursos de áudio e vídeo (multimídia) para ensino. Exemplo: Vídeos, videoaulas etc. (A15, A16)
5	Ferramenta para contexto	Desenvolvimento ou uso de alguma ferramenta para um contexto

	específico	específico. (A24, A25, A26, A27, A28, A29, A30, A31, A32, A33, A34, A35, A36, A37, A38, A39, A40, A41, A42, A43)
6	Interpretador Gráfico	Ambiente voltado para o ensino de Algoritmos, Programação e Estruturas de Dados. Baseado na visualização gráfica e dinâmica de como as estruturas de dados são manipuladas por um programa. (A49)
7	Juiz Online	Plataformas <i>online</i> utilizadas em competições de programação. São capazes de executar os códigos submetidos a elas e informar se o programa funcionou corretamente ou não. (A63)
8	Laboratório Remoto ou Virtual	Espaços virtuais que possibilitam capacitação de um número maior de alunos, redução dos custos da instituição com laboratórios presenciais, autonomia dos alunos na aprendizagem, maior flexibilidade de horários para experimentos e aquisição de dados com modelos reais. (A64, A65)
9	Logo	Linguagem de programação desenvolvida em 1967 por Seymour Papert e outros colaboradores. Permite que adultos e crianças usem o computador como uma ferramenta de aprendizagem. (A66)
10	Robocode	Software <i>open source</i> que possibilita simulações de batalhas com robôs virtuais. Projetado para simular uma arena de batalha com robôs, cada qual com seus próprios algoritmos de ataque e defesa. (A67, A68)
11	Robótica Educativa	Criada para uso doméstico ou escolar. Tem como principal característica, um ambiente de aprendizagem no qual alunos podem montar, programar e analisar o comportamento de um robô ou sistema robotizado. Promove a socialização e a autonomia no aprendizado, criando um ambiente que reúne ciência, tecnologia e trabalho manual. (A69, A70, A71, A72)
12	Scratch	Linguagem gráfica de programação que pretende ser mais simples, fácil de utilizar e mais intuitiva. Possui o propósito de introduzir a programação e conceitos matemáticos, ao mesmo tempo em que incentiva o pensamento criativo, o raciocínio sistemático e o trabalho colaborativo. (A75, A76, A77, A78, A79)

O uso de um **Ambiente Virtual de Aprendizagem (AVA)** permite interatividade constante entre o docente e o aluno e também pode influenciar nos hábitos e rotinas de estudo. Pode-se propor, por exemplo: i) desafio diário: objetivo é a interação diária dos discentes com a plataforma; ii) quiz semanal: para uma maior interatividade, além de permitir a autoavaliação do conteúdo lecionado; iii) chats online: permitem sanar dúvidas e melhorar o entendimento do conteúdo; iv) fóruns para discussões entre os alunos: melhoram também a interatividade entre alunos e o professor.

O uso de *Smartphones* têm aumentado significativamente nos últimos anos. O **APP Inventor** é uma opção que permite aos alunos criar seus programas e executá-los em seus próprios aparelhos, permitindo a inserção de tecnologias móveis dentro do contexto educacional. São utilizados conceitos iniciais de programação e, a forma de codificar é intuitiva, através do uso de blocos. A limitação está no fato dos programas funcionarem apenas na plataforma Android. Pode-se criar uma calculadora de combustível, por exemplo, capaz de mostrar ao usuário qual tipo combustível deve ser usado mediante seu valor. O uso de **Ferramentas** é comum para o ensino de Algoritmos e Programação, mas é necessário também o uso de um computador, por exemplo. Os alunos podem tanto utilizar uma ferramenta quanto criar a sua própria para determinado problema.

A **Robótica Educativa** e o **Arduino** vêm ganhando muito destaque devido ao Movimento Maker, onde o aprendizado é realizado através da experimentação e o conhecimento é construído através do ato de fazer algo [Martinez; Stager 2013]. O

Arduino por ser de utilizado para mostrar o uso de comando básicos como o “*for*” (um led piscar n vezes, por exemplo). Já a Robótica necessita de kits, ou até mesmo de sucatas de equipamentos eletrônicos, e desperta o interesse dos alunos permitindo criar algo que ainda pode ser programado.

O uso do *Áudio/Vídeo* já é bem difundido nas salas de aulas a algum tempo. Como o uso da Internet, essa técnica pode ser remodelada. Nos dias atuais o *Podcast*² pode ser uma boa opção. Os alunos podem criar os *Podcasts* sobre determinado conteúdo para a turma. Tais *Podcasts* devem ser ouvidos por todos para assimilação do conteúdo.

Os *Juízes Online* são muito utilizados em competições de programação. Eles executam os códigos a eles enviados e verificam se o programa funcionou adequadamente. É uma técnica que pode auxiliar os docentes na correção dos códigos feitos pelos alunos. Podem ser utilizados em conjunto com os Hackatons, dependendo das tarefas passadas aos alunos. O *Robocode* é um ambiente de simulação de batalha de robôs programados através da linguagem Java. Pode ser utilizado para aprimorar os conhecimentos dos alunos em programação, além de permitir, caso o docente deseje, criar competições entre equipes de discentes, nas quais o objetivo é combater o robô da equipe adversária.

O *Interpretador Gráfico* permite executar códigos e acompanhar os resultados de forma gráfica. É possível, por exemplo, mostrar como os dados são armazenados em um *Array* de forma gráfica, ou seja, o docente pode passar o algoritmo para os alunos e estes executarem tal código acompanhando como o *Array* é preenchido. É possível também visualizar operações de remoção de valores do *Array*. Outra técnica parecida é o *Laboratório Remoto ou Virtual*. Em geral é uma plataforma que simula algum ambiente real. Um exemplo é o “circuito.io” (<https://www.circuito.io>), que permite aos usuários simular aplicações com circuitos. Nesse caso específico pode-se utilizar o “Arduino” aliado ao Laboratório Virtual e executar os programas desenvolvidos. Além disso diminui o custo, tendo em vista que não será necessário adquirir placas físicas de Arduino.

A linguagem *Logo* também pode auxiliar alunos iniciantes em programação a entender como programar. Ela conta com comandos simples como o “repita” que permitem um primeiro contato com um ambiente de programação. Ao entender esse comando, por exemplo, os alunos estarão aprendendo o conceito de laço em programação. O Logo pode ser utilizado em laboratório através de exercícios preparados pelo docente. O *Scratch* também é uma excelente opção para os iniciantes em programação. A codificação é feita através de blocos de comandos e os alunos aprendem os comandos sem ter que escrevê-los. Hoje já existe um ambiente totalmente *online* permitindo uma flexibilidade de uso em várias plataformas. O professor pode solicitar também aos alunos que compartilhem suas soluções com os demais colegas pelo próprio ambiente *online*. Pode-se também usar da Avaliação em Pares para sugestões, críticas, correções etc.

5. Conclusão

Um dos maiores problemas dos cursos da área da Computação é o número excessivo de evasão dos discentes de tais cursos. Isso pode estar diretamente associado a problemas

² Gravação feita em áudio e/ou vídeo que é publicada em redes e na mídia, em forma de seriado.

de didática dos professores, indicando que a metodologia adotada não está sendo compatível com a realidade destes. Aulas tradicionais tendem a não prender a atenção dos discentes e o uso metodologias, técnicas, ambientes e tecnologias alternativas passa a ser um diferencial. Entretanto, seu uso envolve uma mudança cultural por parte dos docentes e, este trabalho pode auxiliar aqueles que queiram mudar sua forma de ensino, a introduzir estratégias alternativas em suas aulas de Algoritmos e Programação. Embora possam existir limitações na adoção dessas iniciativas, tais como, por exemplo, a necessidade de recursos específicos ou preparação dos professores, elas oferecem mecanismos adicionais para que docentes e alunos possam construir novos cenários de aprendizagem.

Este trabalho promove o esclarecimento sobre o uso de formas alternativas para o ensino de Algoritmos e Programação no contexto brasileiro e, com isto, estratégias de disseminação destas práticas podem ser elaboradas, como um Portal que foi desenvolvido exclusivamente para auxiliar os docentes na busca de tais iniciativas (ALCASYSTEM: <https://goo.gl/7HNUxj>). Espera-se também que com o uso dessa metodologia seja possível melhorar o ensino em Computação e, conseqüentemente minimizar a evasão dos cursos na área.

Referências

- Almeida, E. et al. (2002). AMBAP: Um ambiente de apoio ao aprendizado de programação. Workshop de Educação em Computação, XXII CSBC, Florianópolis.
- Avila, C., Cavalheiro, S. (2017). Robótica Educacional como Estratégia de Promoção do Pensamento Computacional-Uma Proposta de Metodologia Baseada em Taxonomias de Aprendizagem. Anais dos Workshops do CBIE (Vol. 6, No. 1, p. 1192).
- Bell, T., Alexander, J., Freeman, I., Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. The New Zealand Journal of Applied Computing and Information Technology, 13(1):20–29.
- Borba, S. F. P. (2015). Ações realizadas em um curso superior de tecnologia para reduzir sua evasão. In: XXXV CSBC. Recife, 2015.
- Cavalcante, M. M., Silva, J. D. S., Viana, E. C., Dantas, J. R. (2014). A plataforma Arduino para fins didáticos: estudo de caso com recolhimento de dados a partir do PLX-DAQ. In XXXIV CSBC, Brasília, DF.
- Ferreira, C. E. (2015). Evasão: alguns dados e muitas dúvidas. In: XXXV Congresso da Sociedade Brasileira de Computação (CSBC). Recife, 2015.
- Furtado, A. B. (2018). Elementos de Didática da Computação: 1. ed. Belém: abfurtado.com.br, 345 p.
- Gomes, A., Henriques, J., Mendes, A. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. Educação, Formação & Tecnologias 1(1), 93-103.
- Gudigantala, N. (2013). An active learning approach to teaching undergraduate introduction to MIS Course. Paper presented at the 19th AMCIS, Chicago, Illinois.
- Hostins, H., Raabe, A. (2007). Auxiliando a aprendizagem de algoritmos com a ferramenta WebPortugol. XV Workshop de Informática na Escola, Rio de Janeiro.

- Hüsing, T., Korte, W. B., Fonstad, N., Lanvin, B., Cattaneo, G., Kolding, M., Van Welsum, D. (2013). e-Leadership: e-Skills for competitiveness and innovation. Vision, roadmap and foresight scenarios. prepared for European Commission.
- Libâneo, J. C. (2017). Didática. Cortez Editora.
- Maekawa, C. H., Nagai, W. A., Izeke, C. A. (2015). Relato de Gamificação da disciplina Projeto e Análise de Algoritmos do curso de Engenharia de Computação. In: Anais dos Workshops do CBIE (Vol. 4, No. 1, p. 1425).
- Martinez, S. L., Stager, G. S. (2013). Invent to learn: Making, tinkering, and engineering in the classroom. Constructing modern knowledge press.
- Melchiorretto, A. F., Kraemer, C. (2015). O celular na sala de aula. Artefactum - Revista de estudos em Linguagens e Tecnologia, v. 11, n. 2, 2015.
- Mooney, O., Patterson, V., O'connor, M., Chantler, A. (2010). A study of progression in Irish higher education. Higher Education Authority, Dublin.
- Palmeira, L. B., Santos, M. P. (2014). Evasão no Bacharelado em Ciência da Computação da Universidade de Brasília: análise e mineração de dados. Monografia do curso de Ciência da Computação da UnB – Universidade de Brasília. Brasília: UnB, 2014. 287 p.
- Piletti, C. (2000). Didática Geral. São Paulo: Ática (Série Educação).
- Pereira Júnior, J.C.R., Rapkiewicz, C. O. (2014). Processo de ensino-aprendizagem de fundamentos de programação: Uma visão crítica da pesquisa no Brasil. XII Workshop de Educação em Computação, Salvador.
- Raposo, E. H. S., Dantas, V. (2016). O Desafio da Serpente-Usando gamification para motivar alunos em uma disciplina introdutória de programação. In XXVII SBIE (Vol. 27, No. 1, p. 577).
- Ribeiro, J. P., Manso, M. A., Borges, M. (2016). Dinâmicas com App Inventor no Apoio ao Aprendizado e no Ensino de Programação. XXII Workshop de Informática na Escola, Uberlândia.
- Santiago A. D. V., Krounbauer, A. H. (2017). Um Modelo Lúdico para o Ensino de Conceitos de Programação de Computadores. Revista Brasileira de Informática na Educação. 25. 1. 10.5753.
- Sirotheau, S. et al. (2011). Aprendizagem de iniciantes em algoritmos e programação: foco nas competências de auto avaliação. XXII SBIE, Aracaju.
- Sousa Pires, F. G., Melo, R., Machado, J., Silva, M. S., Franzoia, F., Freitas, R. (2018, October). EcoLogic: um jogo de estratégia para o desenvolvimento do pensamento computacional e da consciência ambiental. In Anais dos Workshops do CBIE (Vol. 7, No. 1, p. 629).
- Tobar, C. M. et al. (2001). Uma arquitetura de ambiente colaborativo para o aprendizado de programação. XII SBIE, Vitória, 2001.
- Veronese, T. B., Lemos, R. M. (2015). SAPERE - Sistema de Apoio ao Ensino para Redução da Evasão. In: XXXV CSBC. Recife, 2015.
- Zorzo, A. F., Nunes, D., Matos, E., Steinmacher, I., Leite, J., Araujo, R. M., Correia, R., Martins, S. (2017). Referenciais de Formação para os Cursos de Graduação em Computação. Sociedade Brasileira de Computação (SBC). 153p, 2017.