Mapeamento sistemático sobre o uso de Pensamento Computacional no ensino de programação no ensino superior

A utilização dos computadores cresceu nas últimas décadas, desde a utilização doméstica à utilização em pesquisas científicas como o sequenciamento de DNA, passando por diversas áreas. Desta forma, o conhecimento informática e programação se tornou muito importante. Embora muito utilizada atualmente, a programação ainda é uma das principais dificuldades dos alunos da áreas de exatas, visto que o percentual de reprovações em disciplinas de programação é alto, assim como a evasão de alunos em cursos da área de informática, o que torna importante que sejam investigadas formas de melhorar o aproveitamento dos alunos e o aprendizado de programação. Neste contexto, o conceito de Pensamento Computacional se apresenta como uma alternativa ao ensino tradicional, pois ele consiste no processo pelo qual se pode chegar à solução de problemas utilizando organização e lógica. Com isso, o presente trabalho realizou um mapeamento sistemático de literatura focado na utilização do pensamento computacional no ensino de programação em cursos de graduação e os resultados mostraram que, apesar de ainda existirem poucos estudos na área, o pensamento computacional tem se mostrado uma alternativa promissora no ensino de programação.

Palavras-chave:Pensamento Computacional; Ensino de programação; Ensino superior

1. Introdução

Atualmente, vive-se em uma era digital, onde a máquina informatizada é utilizada nos mais diversos setores e a tecnologia está em constante desenvolvimento. Além disso, a programação está presente em diversas áreas e não somente nos cursos voltados para computação, as disciplinas de programação e algoritmos são componentes curriculares obrigatórios em cursos na área de exatas, comprovando que compreender lógica de programação é essencial para os estudantes que optam por essa área. Entretanto, apesar dos avanços tecnológicos e da vasta utilização da computação, há uma grande dificuldade no aprendizado de programação. De acordo com [de S. Silva et al. 2021], as principais dificuldades dos alunos estão associadas aos assuntos estudados, que são complexos, e aos métodos e didáticas utilizados. Para essa passagem de conhecimento, que pode agregar mais dificuldade. No entanto, segundo [Iepsen 2013], as maiores dificuldades dos estudantes são a grande quantidade de conceitos que precisam ser assimilados em um curto espaço de tempo, a sintaxe da linguagem de programação que deve ser aprendida, além dos conhecimentos de matemática e a habilidade de interpretação de texto, necessárias para compreender os problemas, dificultando assim, a aplicação dos conhecimentos na prática.

Considerando o exposto, se fazem necessários estudos focados em métodos não tradicionais de ensino de programação, a fim de entender se geram melhor aproveitamento, além de compreender os aspectos relacionados.

De acordo com [Valente 2016], entende-se que o Pensamento computacional é um método que consiste na resolução de problemas gerais utilizando conceitos da área de Computação. Considerando o problema apresentado e o método de Pensamento Computacional, o objetivo deste trabalho é realizar um mapeamento sistemático de estudos do

uso de Pensamento Computacional no ensino de disciplinas de programação no ensino superior.

2. Referencial teórico

2.1. O ensino de algoritmos atualmente

De acordo com [Érico Marcelo Hoff do. Amaral 2015], para ensinar lógica de programação é utilizada uma abordagem pedagógica tradicional, na qual o professor apresenta os conceitos da linguagem e os alunos os utilizam para resolver problemas específicos. Este método consiste em passar o conteúdo determinado para o grupo em um tempo específico, sem levar em consideração a individualidade de cada aluno. Embora este método seja o adequado para passar conceitos, ele transforma o aluno em um receptor de informações passivo, o que pode ser ineficiente na prática, visto que o conhecimento de algoritmos é utilizado para resolver problemas específicos, limitando seu emprego no mundo real. Essa abordagem foca na linguagem e não na solução de problemas.

De acordo com [Érico Marcelo Hoff do. Amaral 2015], o contato inicial dos alunos com a programação pode ser um pouco impactante, visto que lhes são apresentados muitos conceitos novos, precisam exercitar o raciocínio lógico, precisam dominar conceitos matemáticos e ainda aprender a sintaxe da linguagem que será utilizada, tudo isso em um curto espaço de tempo.

2.2. Aprendizagem significativa

De acordo com [Moreira 2010], na aprendizagem significativa uma nova informação interage de forma substantiva e não literal, com um conhecimento específico e relevante, o qual o aprendiz possua previamente e, desta forma, ao associar o novo conhecimento ao antigo o indivíduo dá a ele um significado. Este conhecimento anterior específico é denominado subsunçor ou ideia-âncora.

Conforme [Moreira 2010], a aprendizagem significativa não é aquela em que não se esquece o que foi aprendido, pois ao ficar muito tempo sem contato com determinado assunto ele naturalmente será esquecido, porém se a aprendizagem foi significativa esse esquecimento será parcial e ao rever determinado conteúdo ele será relembrado, entretanto se o esquecimento for total como se nunca tivesse aprendido o ocorreu foi uma aprendizagem mecânica, onde apenas decorou-se aquilo.

Pode-se traçar um paralelo entre aprendizagem significativa e aprendizagem mecânica com o aprendizado de lógica de programação. Por exemplo, ao aprender sobre a utilização e sintaxe de conceitos de programação, como estruturas de repetição ou seleção, apenas de forma teórica e sem a utilização para conceber soluções para situações, estes conceitos em breve serão esquecidos ou no mínimo o aluno não saberá o que fazer com tais informações fora do contexto da aula, uma vez que, para ele, isso não tem um significado. No entanto, ao utilizar estes conceitos e sintaxes aplicados ao desenvolvimento de algoritmos e resolução de problemas do mundo real o aluno pode esquecer coisas como a sintaxe ou algum outro detalhe, mas o essencial, que é a parte lógica estará presente.

2.3. Pensamento Computacional

De acordo com [Brackmann 2017], o pensamento computacional vai além da utilização de um computador e está nas mais diversas áreas. Pode-se dizer que está utilizando pensamento computacional em tarefas cotidianas como escrever a receita de algum prato.

Ainda segundo o autor, podem-se definir quatro pilares para o pensamento computacional: decomposição, abstração, algoritmos e reconhecimento de padrões.

A decomposição consiste em dividir um problema em partes menores para resolver as pequenas partes separadas, tornando mais fácil a resolução, como por exemplo dividir uma receita em passos. Depois de decompor um problema é possível comparar com problemas anteriores para identificar padrões nestas partes menores, ou seja, pode-se adaptar ou utilizar uma solução já aplicada em um problema semelhante, o que deixa o processo mais rápido e eficiente. A abstração é uma filtragem de dados que consiste em ignorar partes do problema que não são importantes em detrimento de outras com o objetivo de melhorar o foco nas coisas importantes e trabalhar com eficiência na solução e, para isso, é importante saber como fazer essa filtragem de forma que não se perca nenhuma informação importante para a solução do problema. No desenvolvimento de software a abstração é muito importante para traduzir uma ideia do mundo real em uma uma aplicação computacional.

O algoritmo é uma junção dos pilares anteriores e pode ser definido como um conjunto de instruções ordenadas para realizar uma tarefa, como uma receita de bolo dividida em passos que devem ser executados em uma ordem específica. No contexto da programação de computadores os algoritmos consistem em uma sequência de instruções em uma linguagem que a máquina entenda.

Considerando o apresentado, é possível perceber que a aplicação dos pilares de pensamento computacional vai muito além da programação, vindo a ser útil em diversas áreas como um método eficiente e organizado para solução de problemas. Além disso, no contexto do ensino de programação o método tradicional consiste em receber os conceitos e depois aplicar em soluções de problemas, geralmente matemáticos, como por exemplo implementar um algoritmo que retorne o fatorial de determinado número. Deste modo, além de o aluno ter que dominar os conceitos aprendidos é necessários também que tenha domínio de conceitos matemáticos. Entretanto, estimular o pensamento computacional pode trazer bons resultados, uma vez que pode-se utilizar analogias com o mundo real e ensinar como traduzir problemas em algoritmos promovendo uma aprendizagem significativa.

3. Desenvolvimento

O desenvolvimento do presente trabalho se fundamentou em metodologia proposta por [Kitchenham 2004], que introduz e discute o conceito de Engenharia de Software baseado em Evidência, baseado nos conceitos da Medicina Baseada em Evidência, que propõe uma mudança de paradigmas na forma de condução das pesquisas da área, de forma a possibilitar organização mais efetiva.

Para melhor entendimento deste tipo de pesquisa se faz necessário a diferenciação entre os tipos de estudos, conforme apresenta [Dermeval et al. 2019]: primário é um estudo empírico que investiga uma questão de pesquisa específica, como os experimentos controlados, os estudos de caso, as pesquisas-ação, entre outros; o secundário revisa os estudos primários referentes a uma questão de pesquisa específica com o objetivo de integrar/sintetizar as evidências relacionadas à questão de pesquisa; por exemplo, uma revisão sistemática da literatura, uma meta-análise, um mapeamento sistemático da literatura e uma revisão narrativa da literatura, e; terciário é uma revisão de estudos secundários rela-

cionados à mesma questão de pesquisa.

Desta forma, como um estudo secundário, os passos do processo de desenvolvimento deste trabalho são descritos nas seções a seguir.

3.1. Questões de pesquisa

As questões de pesquisa que orientaram a execução deste trabalho são apresentadas e fundamentadas neste momento, conforme:

- QP1. Qual o curso de graduação?
- QP2. Qual a disciplina em que o Pensamento Computacional foi utilizado?
- QP3. Qual a esfera administrativa da faculdade?
- QP4. Qual linguagem de programação foi utilizada?
- QP5. Qual software foi utilizado como ambiente de desenvolvimento?
- QP6. Como os alunos foram avaliados?
- QP7. Quais os conceitos de programação estão sendo abordados nos estudos?
- QP8. Quais foram os desafios encontrados na aplicação do conceito?
- QP9. Quais métodos foram utilizados nas pesquisas?
- QP10. Como o autor classifica os resultados obtidos?

3.2. Processo de busca

Para este trabalho utilizou-se como base para a busca por estudos plataforma Google acadêmico[1]. Além disso, como forma de validação a expressão final de busca foi pesquisada no Periódicos Capes[2] para validar se os resultados retornados também estariam presentes no Google acadêmico. O processo iniciou avaliando o resultado da busca utilizando como palavras-chave apenas “Pensamento Computacional”, que é o tema central deste trabalho. Uma vez que, dentro desse tema, o presente trabalho busca entender este conceito como uma forma de melhorar a experiência dos alunos no aprendizado de lógica no ensino superior o primeiro filtro aplicado foi em relação a artigos que juntam Pensamento Computacional e lógica de programação, e, para tal, foi adicionado ao texto de busca a palavra “programação”.

Ao realizar uma busca com os novos termos, notou-se que ainda estava muito abrangente, pois retornou 3070 ocorrências, onde após uma análise superficial dos resultados, foi observado um grande número de trabalhos que não eram voltados para o ensino superior, e, por isso, adicionou-se ao texto de busca a expressão ”ensino superior”, o que reduziu o número de artigos retornados para 1210. No entanto, ao analisar os resultados, notou-se que alguns artigos relativos a ensino superior retornados anteriormente não estavam presentes na busca utilizando o novo texto, e, desta forma, foram inseridas novas expressões equivalentes a ensino superior, como “graduação” ou “tecnólogo” ou “licenciatura” e “Pensamento computacional”. Ainda assim, junto aos resultados de ensino superior vieram diversos outros de educação básica e ensino de crianças, sendo necessário alterar o texto de de busca, visando excluir os estudos desse tipo, o que retornou 324 trabalhos.

[1] https://scholar.google.com/

[2] https://www-periodicos-capes-gov-br.ezl.periodicos.capes.gov.br/

Como o objetivo deste trabalho é analisar o estado da arte, foram considerados apenas os artigos produzidos nos últimos 5 anos, e, como esta etapa da pesquisa foi realizada em 2021, o ano de corte considerado foi 2016, reduzindo a quantidade de artigos para 246. Ao iniciar a leitura dos resumos dos artigos, para aplicar os critérios de exclusão manual foi observada uma grande quantidade de artigos relacionadas ao ensino médio e ensino fundamental, o que fez com que fosse necessária uma nova atualização no texto de busca e a realização de uma nova pesquisa excluindo ensino médio e ensino fundamental. Esta nova pesquisa obteve 110 resultados, que serviram de base para a próxima etapa, que foi a leitura dos resumos e aplicação dos critérios de exclusão de forma manual.

Ao iniciar esta etapa, notou-se que a maioria dos resultados era relacionado ao ensino superior, porém sem qualquer relação com programação ou Pensamento Computacional, e, ao analisar os resumos dos trabalhos, apenas 21 eram referentes a Pensamento Computacional ou programação, e, destes, apenas 5 passariam pelos critérios de inclusão e exclusão. Desta forma, experimentou-se realizar uma nova busca, onde trocou-se as palavras referentes ao ensino superior pela palavra lógica, e removeu-se resultados relacionados a jornalismo, uma vez que vieram muitos resultados associados ao tema. O retorno para essa busca foi 219 trabalhos. O Google Acadêmico utiliza operadores lógicos em sua busca, desta forma a expressão final de busca ficou da seguinte forma: ”pensamento computacional” AND (lógica OR programação) -jornalismo -”ensino básico” -”séries iniciais” -”educação básica” -crianças -”educação infantil” -”ensino médio” -”ensino fundamental”.

Após concluir a busca foi realizado um processo de filtragem manual a partir da análise dos resumos dos trabalhos, onde o primeiro critério de exclusão aplicado foi em relação a estudos secundários, descartando revisões bibliográficas e mapeamentos de literatura, uma vez que o objetivo do presente trabalho é entender o que está sendo realizado em termos de estudos de campo.

Outros critérios de exclusão definidos foram: ser sobre inclusão, não ser sobre Pensamento Computacional, não ser sobre ensino de lógica. Ao analisar os resumos dos 219 trabalhos retornados, 63 foram considerados elegíveis para análise, dos quais em 8 o texto completo do trabalho não estava disponível, apenas o resumo, e, ao aplicar os critérios de exclusão, foram reduzidos para 8.

3.2.1. Critérios de qualidade

Para garantir a qualidade dos resultados também é necessário realizar uma análise da qualidade dos artigos, e, para isso, será utilizado como referência o método estabelecido por [Dyba and Dingsøyr 2008], o qual utiliza algumas questões que devem ser respondidas para cada artigo com o objetivo de determinar se deve ou não ser incluído na pesquisa.

Para esta análise foram selecionadas 5 das 11 perguntas propostas por [Dyba and Dingsøyr 2008], para serem aplicadas a cada trabalho restante da busca, estas são listadas a seguir:

- CQ1: As metas e objetivos foram claramente relatados?
- CQ2: Houve uma descrição adequada do contexto em qual a pesquisa foi realizada?

- CQ3: Métodos apropriados de coleta de dados foram usados e descritos?
- CQ4: A relação entre a pesquisadora e os participantes foi considerada adequada?
- CQ5: Eles forneceram valor para pesquisa ou prática?

4. Resultados

Ao final das buscas restou o total 8 estudos que se encaixam nos parâmetros da pesquisa e respondem às perguntas de pesquisa. Ao final restaram os seguintes estudos: [Mota and Neves 2020], [Rezende and Junior 2018], [Amaral et al. 2016], [Figueiredo and García-Peñalvo 2021], [Vahldick et al. 2018], [Souza et al. 2020], [Coutinho et al. 2018] e [Tozzi et al. 2019]. Em relação ao público ao público alvo dos estudos, dos oito estudos, apenas três foram realizados com cursos de computação, enquanto que os demais foram realizados em outras áreas como por exemplo física e engenharia de produção. Além disso, cinco estudos foram realizados em disciplinas iniciais de programação, como por exemplo introdução a programação e algoritmos e dois foram realizados em cursos de extensão. Bem como, cinco estudos foram realizados em universidades federais. A análise acerca da forma como os estudos foram realizados mostrou que foram utilizados diversos softwares, inclusive alguns estudos se tratavam de testes de ferramentas, desta forma alguns dos softwares citados foram: Open Simulator, Autodesk, Tinkercad™, App Inventor, HTProgramming, Code Studio3, NoBug's, entre outros. A respeito da linguagem utilizada, alguns estudos não informaram linguagem, mas a maioria utilizou Scratch, enquanto que apenas um utilizou Java e outro utilizou GNU C. Ademais, a maioria dos estudos abordou temas de disciplinas iniciais de programação focados em desenvolvimento e apenas um abordou conceitos abstratos no desenvolvimento de Pensamento Computacional. A avaliação dos alunos foi realizada por meio de questionário, análise estatística do desempenho dos alunos, análise do código fonte produzido pelos alunos, com a utilização de jogos e por meio da ferramenta que estava sendo testada. Em relação à forma como os estudos foram conduzidos,a maioria foi realizada com a mesma turma dividindo o curso em tópicos ou momentos distintos, mas também foram realizados questionários sobre um teste de ferramenta, testes de metodologias diferentes com grupos distintos, análise de códigos fontes de uma amostra de alunos e atividades em papel com o mesmo grupo. Além disso, os estudos apontaram que os maiores desafios na aplicação dos conceitos foram: falta de contato prévio dos alunos com programação, falta de afinidade dos alunos com a tarefa que estava sendo realizada e dificuldade dos alunos na solução dos problemas. Ademais, todos os estudos obtiveram resultados positivos.

5. Considerações finais

A pesquisa foi iniciada com o termo "Pensamento Computacional" e, devido ao grande número de resultados iniciais, esperava-se que o número de artigos resultantes ao final do processo seria alto, no entanto, ao iniciar a leitura dos resumos foi observado que haviam muitos resultados referentes a diversas áreas, e que seriam necessários filtros para chegar a resultados que fossem relevantes ao objetivo da pesquisa. O processo de definir um texto de pesquisa e filtrar os resultados foi a parte mais extensa no trabalho, uma vez que a cada novo texto de busca foi necessário ler no mínimo os resumos dos primeiros estudos resultantes da busca e com base nestes resultados definir os próximos passos. Ao ler os textos completos e excluir outros estudos que também não se encaixavam no

proposto pelo trabalho, observou-se que haviam poucos resultados. Além disso, ao ler os estudos resultantes da busca observou-se que o Google acadêmico inclui muitos trabalhos que nem ao menos citam os termos da pesquisa ao longo do texto, estando presentes, por exemplo apenas nas referências.

Durante o desenvolvimento do presente trabalho, as pesquisas, tanto para o referencial teórico quanto no mapeamento sistemático, deixaram evidente a grande dificuldade dos alunos no aprendizado de lógica de programação. Além disso, ao longo da pesquisa foi possível perceber que, atualmente, buscam-se diversas alternativas para este problema, como a utilização de robótica, gamificação ou alguma outra forma lúdica para ensinar programação e que, mesmo de forma implícita, o desenvolvimento do Pensamento Computacional é um componente chave em todas as alternativas.

Em relação à utilização do Pensamento Computacional, observou-se que há diversos estudos em áreas diferentes da informática, como por exemplo na arquitetura, jornalismo, entre outras. Além disso, existem muitos estudos com relação ao ensino de lógica, porém não são voltados ao ensino superior e devido a não se encaixarem no contexto do presente trabalho não foram estudados de forma mais aprofundada. Além disso, os estudos voltados ao ensino superior, inclusive os selecionados ao final do mapeamento, não são exclusivamente sobre utilização de Pensamento Computacional, uma vez que em geral buscam testar algum método ou ferramenta para o desenvolvimento de Pensamento Computacional.

Considerando o exposto, pôde-se perceber que os estudos em português relacionados à aplicação de Pensamento Computacional no ensino de programação no ensino superior ainda são poucos, e, não há uma descrição detalhada dos conceitos abordados ou estudos mais aprofundados na aplicação do tema. Além disso, a partir da descrição, é possível perceber que não há uma forte aplicação dos pilares do Pensamento Computacional, há apenas uma descrição genérica como, por exemplo, o desenvolvimento do Pensamento Computacional, etc.

Apesar da descrição rasa da aplicação dos pilares de Pensamento Computacional é possível perceber que todos os estudos realizados obtiveram bons resultados. Além disso, mostram que a utilização do Pensamento Computacional no ensino de programação na graduação pode ser um excelente método, visto que pode permitir aos alunos que compreendam a utilização dos pilares supracitados para a resolução de problemas do mundo real e assim relacioná-los com conhecimentos prévios.

Por fim, a partir do que foi aprendido com essa pesquisa, pontua-se como alternativa promissora para trabalhos futuros a realização de um estudo de campo aplicando os pilares de Pensamento Computacional como ferramenta no ensino de lógica de programação no ensino superior. Adicionalmente, considerando a quantidade de estudos com ensino de lógica de programação nos níveis médio e fundamental, também se considera interessante a realização de um mapeamento destes estudos em busca de entender como o conceito vem sendo utilizado.

References

Amaral, E., Medina, R., and Tarouco, L. M. R. (2016). Processo de ensino e aprendizagem de algoritmos integrando ambientes imersivos e o paradigma de blocos de programação

visual. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação.*, page 20.

Brackmann, C. P. (2017). Desenvolvimento do pensamento computaciona através de atividades desplugadas na educação básica.

Coutinho, E., Bonates, M., and Moreira, L. O. (2018). Relato sobre o uso de uma ferramenta de desenvolvimento de jogos para o ensino introdutório de lógica de programação. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, page 689.

de S. Silva, R. A., de Fátima P. Ferreira, M., de Sousa. Santos, I., Andrade, R. M. C., and F, B. B. A. (2021). Evasão em computação na ufc sob a perspectiva dos alunos. *Anais do XXIX Workshop sobre Educação em Computação*, pages 338–347.

Dermeval, D., de Miranda. Coelho, J. A. P., and Bittencourt, I. I. (2019). Mapeamento sistemático e revisão sistemática da literatura em informática na educação.

Dyba, T. and Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Science Direct*, (50):833–859.

Figueiredo, J. and García-Peñalvo, F. J. (2021). Help to programming: uma ferramenta para o ensino e aprendizagem da programação. *Grupo GRIAL*.

Iepsen, E. F. (2013). Ensino de algoritmos: detecção do estado afetivo de frustração para apoio ao processo de aprendizagem.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Nome do periódico*.

Moreira, M. A. (2010). O que é afinal aprendizagem significativa?

Mota, L. P. and Neves, I. (2020). Robótica como ferramenta para o desenvolvimento do pensamento computacional e introdução a lógica de programação. *ANAIS DO WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI)*, pages 141–145.

Rezende, C. M. and Junior, E. L. B. (2018). Uso da linguagem scratch no ensino de programação para licenciandos em física. *Ciclo Revista (ISSN 2526-8082)*, 3(1).

Souza, R. L. D., Ferreira, F. Z., and da Costa Botelho, S. S. (2020). Proposta para avaliação de códigos fonte com tf-idf. *Anais do XXXI Simpósio Brasileiro de Informática na Educação, SBC*, pages 112–121.

Tozzi, Y. L., Vieira, P. H., Altomani, R. S., Piovezan, T., and Venturini, P. C. (2019). Scratch na universidade. *Brazilian Applied Science Review*, 3(6):2643–2648.

Vahldick, A., Schoeffel, P., Liz, F. B., Wazlawick, R., and Ramos, V. (2018). Maior frequência na aplicação de instrumentos de avaliação em uma disciplina introdutória de programação: Impactos no desempenho e motivação. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, page 739.

Valente, J. A. (2016). Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. *Revista E-curriculum*, 14(3):864–897.

Érico Marcelo Hoff do. Amaral (2015). Processo de ensino e aprendizagem de algoritmos integrando ambientes imersivos e o paradigma de blocos de programação visual.