ANEXO 1 A

Número do BANPESQ/THALES: 2014016688

Título do projeto: Explorando abordagens não-convencionais para o ensino de programação e engenharia de software no Ensino Básico e Superior

Projeto submetido para: PIBIC (X), PIBIC – Af ( ), PIBIC – EM ( ), PIBITI ( )

I. Autor:

Helio Henrique Lopes Costa Monte-Alto

II. Resumo:

O objetivo central deste projeto é explorar abordagens não-convencionais de ensino de programação para turmas iniciantes, tanto em nível básico quanto superior. O aprendizado de programação envolve pensamento lógico e abstrato. Tal habilidade é, para muitos alunos, uma habilidade difícil e que leva à desmotivação e desistência. Portanto, o ensino de programação é uma tarefa desafiadora que deve incluir elementos que despertem interesse nos alunos. Além disso, é importante notar o nível de habilidade em programação dos alunos egressos às disciplinas e cursos que envolvem programação, e a qualidade dos programas por eles desenvolvidos. Muitos acadêmicos saem despreparados para o mercado de trabalho devido à baixa qualidade do ensino de programação que tiveram, à falta de prática e à falta de conhecimento de boas práticas de programação. Este projeto de pesquisa propõe, portanto, levantar abordagens de ensino de programação diferentes das convencionais, procurando identificar falhas nos tradicionais métodos de ensino e como essas novas abordagens podem melhorar esse cenário.

III. Objetivos:

Objetivo geral: Construir um levantamento de abordagens não-convencionais de ensino e aprendizagem de programação, identificando as vantagens e desvantagens em relação aos métodos tradicionais, bem como identificar quais são as abordagens mais apropriadas para os diferentes níveis de ensino.

Como objetivos específicos:

- Fazer um levantamento sobre quais são as abordagens mais utilizadas para o ensino de programação;
- Identificar falhas nas abordagens tradicionais de ensino e aprendizado de programação;
- Fazer um levantamento sobre diferentes maneiras de ensinar programação;
- Identificar, para cada abordagem pesquisada, as vantagens e desvantagens em relação às abordagens tradicionais;
- Identificar quais abordagens melhor se aplicam ao Ensino Básico;
- Identificar quais abordagens melhor se aplicam ao Ensino Superior;
- Identificar quais abordagens melhor se aplicam ao Ensino Técnico;
- Analisar empiricamente a efetividade de algumas dessas abordagens sobre as abordagens tradicionais;
- Documentar todo o conhecimento levantado e produzido para servir de diretriz para o educador que deseje melhorar seu ensino de programação;
- Prover, para o futuro licenciado em Computação, conhecimento sobre alternativas de ensino de programação, possibilitando ao futuro educador a capacidade de decisão sobre qual estratégia de ensino utilizar em diferentes casos;
- Publicar resultados das análises dos elementos levantados e analisados em eventos e periódicos científicos.

IV. Introdução ou evidências de interesse (justificativas):

Programação é uma disciplina que envolve pensamento lógico e abstrato, e requer prática para ser dominada. Não obstante, existe uma grande dificuldade para aqueles que ingressam em cursos relacionados à Computação no que diz respeito às disciplinas que envolvem programação. Soma-se a isso o comum despreparo dos alunos que vieram de escolas que não forneceram um ensino adequado de Matemática e Lógica. Disciplinas de programação e algoritmos acabam, assim, se tornando difíceis de ensinar e ao mesmo tempo manter o interesse e motivação dos alunos, que acabam se considerando incapazes de aprender (HINTERHOLZ Jr, 2009). O índice de evasão em cursos de Computação costuma ser alto, sendo um dos principais fatores de desistência a dificuldade na compreensão de abstração e raciocínio lógico (PRIETCH, PAZETO,

2010; TOBAR, 2001). É também interessante notar a alta porcentagem de reprovações em disciplinas de programação, conforme relatos de Prietch e Pazeto (2010).

De acordo com Rodrigues Jr. (2002) *apud* Hinterholz Jr. (2009), diversos problemas podem ser citados no processo de ensino e aprendizagem de algoritmos e programação que culminam em desmotivação, reprovação ou evasão, dentre eles: dificuldade de adaptação dos alunos a desenvolverem raciocínio lógico quando estão acostumados a decorar o conteúdo; e falta de motivação do aluno gerada pelo despreparo e o desânimo quando ele acredita que a disciplina constitui um obstáculo extremamente difícil de ser superado. Além desses motivos, a própria abordagem de ensino utilizada pelo professor pode ser um fator de desmotivação. De acordo com Borges (2000), o modo tradicional não consegue facilmente motivar os alunos a se interessarem pela disciplina.

Outro problema importante, apontado por Pereira Jr e Rapkiewicz (2004), refere-se às dificuldades apresentadas por grande parte dos alunos em assimilar as abstrações envolvidas no processo de ensino e aprendizagem de programação. Além disso, muitos dos alunos não conseguem desenvolver o raciocínio lógico necessário para o posterior desenvolvimento de programas.

Este trabalho tem como proposta investigar possíveis fatores nas abordagens convencionais de ensino de programação que possam causar tal desinteresse e despreparo nos alunos, e investigar alternativas que possam substituir ou complementar tais abordagens convencionais. Procura-se, também, investigar abordagens de ensino de programação no Ensino Básico como forma de incentivo e preparo para o pensamento lógico e abstrato e para possíveis disciplinas de programação no Ensino Superior. É de interesse, ainda, observar o nível de habilidade em programação e a qualidade dos programas desenvolvidos por egressos a cursos de Computação, de modo a avaliar se o ensino de programação que tiveram durante o curso foi ou não efetivo.

V. Revisão bibliográfica:

De acordo com Graziola Jr e Haetinger (2002), “tornar o processo ensino-aprendizagem mais interessante é um desafio enfrentado pelo professor em qualquer nível de ensino”.

É de amplo conhecimento as dificuldades encontradas por alunos e professores no processo ensino-aprendizagem de Matemática no Ensino Fundamental e Médio. Muitos alunos saem do Ensino Básico com um pensamento muito negativo sobre a disciplina em si, levando ao baixo interesse por cursos de engenharia, tecnologia e ciências exatas. No Brasil, já há alguns

anos verifica-se um déficit na formação de profissionais de Computação. Em um estudo da Associação Brasileira de Empresas de Tecnologia de Informação e Comunicação (BRASSCOM) havia sido estimado que, em 2014, a demanda por profissionais de TI (Tecnologia da Informação) no mercado chegaria a 78 mil profissionais, mas apenas 33 mil concluiriam os cursos superiores da área. Soma-se a isto a diminuição na procura por cursos superiores em Computação e TI e o alto índice de evasão nesses cursos, que seria de aproximadamente 87%, sem considerar ainda a ociosidade de vagas nas universidades e faculdades brasileiras (BRASSCOM, 2011, *apud* BARCELOS, SILVEIRA, 2012).

Tal situação crítica não é exclusividade do mercado brasileiro. Muratet *et al.* (2009) afirmam que no período de 2005 a 2009 houve uma redução de cerca de 20% no número de estudantes de Ciência da Computação na França, e Crenshaw *et al.* (2008) reportam queda de 50% no interesse de estudantes americanos pela área ao longo dos anos 2000. Portanto, não é pequena a quantidade de estudantes que se encontram desmotivados pela Matemática e qualquer disciplina a ela relacionada, o que inclui a Ciência da Computação.

A falta do domínio adequado de conhecimentos matemáticos pelos alunos é, também, um possível fator explicativo para a falta de interesse e evasão em cursos da área de Computação. Uma revisão da literatura permite apontar que diversos pesquisadores (CAMPBELL, MCCABE, 1984; WILSON, SHROCK, 2001; BEAUBOUEF, 2002; SETTI, 2009) indicam possíveis correlações entre o conhecimento matemático prévio dos alunos e seu desempenho em cursos introdutórios de Computação, bem como a relevância de tópicos do conhecimento matemático para uma melhor compreensão e modelagem dos processos computacionais (BARCELOS, SILVEIRA, 2012). Coincidentemente ou não, constata-se um baixo rendimento na aprendizagem de Matemática pelos alunos no Ensino Básico. Os resultados do SARESP (Sistema de Avaliação do Rendimento Escolar do Estado de São Paulo) de 2011, por exemplo, indicam que 58,4% dos alunos do 3º ano do Ensino Médio têm desempenho considerado abaixo do básico em Matemática (TARGINO, 2012). As dificuldades com o ensino de Matemática se repetem em outros países; no Reino Unido, a RSA (*Royal Society of Arts*) recentemente identificou que algumas universidades chegam ao ponto de não divulgar os pré-requisitos de Matemática necessários para a matrícula em disciplinas, temendo com isso afugentar os alunos (PAETON, 2012).

Em síntese, existe uma desmotivação em relação ao aprendizado de Matemática e uma falta de preparo nesta disciplina nos ingressantes em cursos de Computação. Muitos professores vêm tentando contornar a situação ao aplicar atividades lúdicas com materiais concretos e jogos, com o intuito de motivar os alunos à disciplina de Matemática (FIORENTINI, MIORIM, 1990). No entanto, nem todos os pesquisadores concordam com o ponto de vista de que basta acrescentar

tais elementos para se alcançar uma aprendizagem efetiva de Matemática. Carraher *et al.* (1988, *apud* FIORENTINI, MIORIM, 1990) afirmam, com base em suas pesquisas, que "não precisamos de objetos na sala de aula, mas de objetivos na sala de aula, mas de situações em que a resolução de um problema implique a utilização dos princípios lógico-matemáticos a serem ensinados". Em outras palavras, não é o objeto material em si, utilizado em sala de aula, que despertará o interesse dos alunos, mas sim o desenvolvimento de objetivos e problemas que motivem os alunos a aprenderem e utilizarem seus conhecimentos e habilidades para alcançar tais objetivos e resolver os problemas.

Tal criação de objetivos e situações interessantes para os alunos, e que impliquem na resolução de problemas utilizando a Matemática, é possível por meio de projetos que envolvam programação de computadores. A ideia de utilizar atividades lúdicas com materiais concretos durante o ensino pode, sim, ajudar a melhorar a situação (ALVES, 2001). No entanto, a programação, por lidar com objetos virtuais, permite aos alunos maior liberdade e criatividade para desenvolver soluções para problemas e lidar com situações que eles achem interessantes. Outra hipótese levantada por alguns pesquisadores é de que o pensamento lógico e computacional, essencial na prática de programação, pode contribuir para o desenvolvimento do pensamento matemático do estudante (BARCELOS, SILVEIRA, 2012).

Considerando o avanço tecnológico inerente à sociedade, disciplinas como Ciência da Computação e programação de computadores gradativamente estão sendo incorporadas nas matrizes curriculares do Ensino Médio de países europeus desde meados de 2004 (RODRIGUES, 2013). Tais conhecimentos são importantes mesmo para estudantes que não se interessam por estas áreas, pois eles podem se beneficiar desenvolvendo novas habilidades como, por exemplo, raciocínio lógico, resolução de problemas e pensamento algorítmico (GRANDELL *et al.*, 2006).

Portanto, existe um consenso entre muitos educadores de que aprender a programar melhora habilidades em Matemática e resolução de problemas (PAPERT, 1972). No entanto, evidências empíricas para tal argumento ainda são escassas, com algumas notáveis exceções (NOSS, 1987; SCHANZER *et al.*, 2015). De fato, pesquisadores versados na literatura sobre educação em Computação alertam que não se sabe se programação realmente transfere habilidades de resolução de problemas em Computação para o domínio da Matemática (SCHANZER *et al.,* 2015).

A transferência de aprendizagem entre domínios tipicamente requer tanto conexões estruturais profundas entre os domínios quanto instruções explícitas de como aplicar conceitos de uma disciplina na outra (BRANSFORD, SCHWARTZ, 1999; RICH *et al.*, 2013). Muitos projetos que tentam transferir a habilidade de resolver problemas em Computação para a habilidade de

resolver problemas de Matemática falham em lidar com pelo menos um desses dois requisitos. Isto levanta dois desafios: podemos identificar práticas específicas de resolução de problemas em Computação que possuem analogia direta aos processos na Matemática? E podemos ensiná-los de tal modo que os alunos realizem ganhos de desempenho em Matemática? O Bootstrap, uma das abordagens que deseja-se estudar neste projeto, tenta responder a esses dois desafios por meio de uma abordagem pedagógica para o ensino de Matemática e programação (SCHANZER *et al.*, 2015).

A ideia do Bootstrap é ensinar conceitos de álgebra e geometria à medida que os alunos aprendem a programar desenvolvendo seus próprios jogos eletrônicos (SCHANZER *et al.*, 2015). A hipótese levantada é que, além de ajudar a melhorar o desempenho dos alunos nessas disciplinas, também motive e prepare os alunos para o ensino de Computação no Ensino Superior, uma vez que, por meio do Bootstrap, também aprendem conceitos de Ciência da Computação e são introduzidos ao pensamento computacional. O projeto tem sido integrado por professores a aulas de Matemática e tecnologia, e a oficinas extraclasse nos EUA, atingindo milhares de estudantes desde 2006. A equipe do Bootstrap também oferece, nos EUA, *workshops* para professores que desejam utilizá-lo em suas aulas, tendo apoio voluntário e financeiro de diversas empresas da área de tecnologia, como Apple, Cisco, Facebook, Google e IBM.

VI. Materiais e métodos:

Inicialmente pretende-se realizar uma revisão bibliográfica sobre o assunto. Para isto, serão utilizados termos de busca relacionados ao tema, visando encontrar publicações relevantes que possam direcionar a pesquisa e compor um levantamento inicial de abordagens convencionais e não-convencionais de ensino de programação.

A seguir, o levantamento será analisado de modo a identificar possíveis falhas no ensino, levando em consideração conceitos pedagógicos e inerentes à programação e projeto de algoritmos. Segue-se, então, o término do levantamento sobre abordagens de ensino de programação não-convencionais e a identificação, para cada uma delas, de pontos que podem ser vantajosos em relação às abordagens tradicionais. Nesse momento podem também ser identificados possíveis práticas utilizadas em abordagens não-convencionais que podem ser complementares às abordagens convencionais, assim como pontos de abordagens não-convencionais que aparentam desvantagem em relação às abordagens convencionais. Toda essa análise deve ser fundamentada nas próprias análises empíricas apresentadas pelas referências bibliográficas e em conceitos de Pedagogia e Computação.

A análise realizada deve produzir uma relação das abordagens pesquisadas levando em consideração características qualitativas que sejam relevantes para a efetividade e fator motivador do ensino de programação. Tal quadro deve servir de insumo para a identificação de abordagens que melhor se aplicam ao Ensino Básico, Ensino Superior e Ensino Técnico, podendo também ser propostas adaptações e fusões entre diferentes abordagens. Por exemplo, métodos que incluam o uso de animações e jogos podem ser escolhidos para o Ensino Básico, enquanto que métodos que utilizem linguagens de programação mais avançadas podem ser melhor utilizados no Ensino Superior e Técnico.

O próximo passo será, com base no levantamento e análise das diversas abordagens, separar algumas para verificação empírica. Como o objetivo é verificar a efetividade de abordagens não-convencionais sobre abordagens convencionais, o ideal é escolher uma ou duas das abordagens não-convencionais levantadas e compará-las empiricamente com uma abordagem convencional. Isto pode ser feito aplicando as diferentes abordagens em sala de aula e coletando dados referentes ao progresso e interesse dos alunos.

Enfim, com base nos resultados alcançados, pretende-se montar uma documentação sobre como aplicar, em sala de aula, as abordagens que se sobressaíram dentre as demais. O objetivo de tal documentação é disponibilizá-la aos professores de Computação e aos futuros licenciados em Computação para que possam ter conhecimento dos procedimentos e, possivelmente, aplicá-los em sala de aula. Pretende-se, também, escrever artigos para eventos e periódicos científicos com o fim de divulgar a pesquisa e os resultados alcançados.

VII. Cronograma:

Atividades	**2014**			**2015**								
	Out	**Nov**	**Dez**	**Jan**	**Fev**	**Mar**	**Abr**	**Mai**	**Jun**	**Jul**	**Ago**	**Set**
Revisão bibliográfica	x	x	x									
Levantamento das abordagens tradicionais de ensino de programação		x	x	x								
Identificação de falhas pedagógicas nas abordagens tradicionais			x	x	x							
Levantamento de abordagens não-			x	x	x	x						

convencionais de ensino de programação												
Identificação das possíveis vantagens e desvantagens das abordagens não-convencionais diante das convencionais					x	x	x	x				
Identificar as abordagens mais adequadas para diferentes níveis de ensino							x	x	x			
Elaborar verificação empírica								x	x	x		
Realizar verificação empírica											x	x

Atividades	**2015**			**2016**								
	Out	**Nov**	**Dez**	**Jan**	**Fev**	**Mar**	**Abr**	**Mai**	**Jun**	**Jul**	**Ago**	**Set**
Realizar verificação empírica	x	x	x	x	x	x	x	x	x			
Documentar resultados e métodos de ensino mais interessantes			x	x					x	x	x	x
Escrever artigos					x	x					x	x

VIII. Bibliografia:

ALVES, E. M. S. **Ludicidade e o Ensino de Matemática.** 4.ed. Campinas: Papirus Editora, 2001.

BARCELOS, T. S.; SILVEIRA, I. F. Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica. In: XX Workshop sobre Educação em Computação, Curitiba. **Anais do XXXII CSBC**, 2012.

BEAUBOUEF, T. Why computer science students need math. **SIGCSE Bulletin**, v. 34, n. 4, p. 57–59, dez 2002.

BORGES, M. A. F. **Avaliação de uma metodologia alternativa para a aprendizagem de programação.** VIII Workshop de Educação em Computação–WEI, 2000.

BRANSFORD, J., SCHWARTZ, D. Rethinking transfer: A simple proposal with multiple implications. **Review of Research in Education**, v. 24, p. 61-100. American Educational Research Association, 1999.

BRASSCOM. Tecnologia sofre com evasão universitária. Disponível em: <http://www.brasscom.org.br/brasscom/content/view/full/5155>. Acesso em: 03/02/2012.

CAMPBELL, P. F.; MCCABE, G. P. Predicting the success of freshmen in a computer science major. **Communications of the ACM**, v. 27, n. 11, p. 1108–1113, nov 1984.

CARRAHER, T. N; CARRAHER, D. W.; SCHLIEMANN, A. D. **Na vida dez, na escola zero.** São Paulo: Cortez, 1988.

CRENSHAW, T. L.; CHAMBERS, E. W.; METCALF, H.; THAKKAR, U. A case study of retention practices at the University of Illinois at Urbana-Champaign. SIGCSE '08. **Proceedings of SIGCSE 2008**. New York: ACM, 2008.

FIORENTINI, D.; MIORIM, M. A. Uma Reflexão sobre o Uso de Materiais Concretos e Jogos no Ensino da Matemática. **Boletim SBEM-SP.** São Paulo, ano 4, n.7, p. 5-10, jul./ago. 1990.

GRANDELL, L.; PELTOMAKI, M.; BACK, R. B.; SALAKOSKI, T. Why Complicate Things? Introducing Programming in High School Using Python. **Anais do 8th Australasian Computing Education Conference (ACE 2006)**. Hobart, Tasmania, Australia. 2006.

GRAZIOLA Jr, P. G.; HAETINGER, W. Utilização de um carro-robô construído com componentes de baixo custo como ferramenta de apoio ao aprendizado dos conceitos de orientação a objetos. In: VI Congresso Iberoamericano de Informática Educativa - RIBIE, 2002, Vigo - Espanha. **Anais do VI Congresso Iberoamericano de Informática Educativa - RIBIE**, 2002.

HINTERHOLZ Jr, O. Tepequém: uma nova Ferramenta para o Ensino de Algoritmos nos Cursos Superiores em Computação. **XVII-Anais do Workshop sobre Educação em Informática**. 2009.

MURATET, M.; TORGUET, P.; JESSEL, J.P.; VIALLET, F. Towards a serious game to help students learn computer programming. **International Journal of Computer Games Technology**, v. 2009, p. 3:1–3:12, jan 2009.

NOSS, R. Children's learning of geometrical concepts through Logo. **Journal for Research in Mathematics Education**, 18(5):343-362, 1987.

PAETON, G. University maths "too difficult" for British students. **The Telegraph**, 10 fev 2012. Disponível em: <http://bit.ly/LTozHv>. Acesso em: 20/03/2015

PAPERT, S. Teaching children to be mathematicians versus teaching about mathematics. **International Journal of Mathematical Education in Science and Technology,** 3(3):249-262, 1972.

PEREIRA Jr, C. R. J.; RAPKIEWICZ C. E. **O processo de ensino-aprendizagem de fundamentos de programação: uma visão crítica da pesquisa no Brasil.** Workshop sobre Educação em Computação - WEI. 2004.

PRIETCH, S. S.; PAZETO, T. A. "Estudo sobre a Evasão em um Curso de Licenciatura em Informática e Considerações para Melhorias." Workshop de Educação em Informática Bahia-Alagoas-Sergipe (WEIBASE) / ERBASE, 2010, Macieó/AL. **Anais da 10ª Escola Regional de**

Computação Bahia-Alagoas-Sergipe (ERBASE). Maceió/AL : SBC, CESMAC, UFAL e CEFET-AL, 2010.

RICH, P. J.; LEATHAM, K. R; WRIGHT, G. A. Convergent cognition. **Instructional Science**, 41(2):431-453, Mar. 2013.

RODRIGUES, F. S. **Estudo sobre a evasão no curso de Ciência da Computação da UFRGS.** Trabalho de Graduação (Bacharelado em Ciência da Computação). Universidade Federal do Rio Grande do Sul. Instituto de Informática. 2013.

RODRIGUES Jr, M. Como Ensinar Programação? **Informática – Boletim Informativo**, Ano I, nº 01, ULBRA, Canoas, RS. 2002.

SANTOS, R. P. dos; COSTA, H. A. X. Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos Iniciantes em Computação e Informática. In: **INFOCOMP**, Volume 5, nº.1, ISSN 1807-4545. 2006.

SCHANZER, E.; FISLER, K.; KRISHNAMURTHI, S.; FELLEISEN, M. **Transferring Skills at Solving Word Problems from Computing to Algebra Through Bootstrap.** SIGCSE' 15. 2015.

SETTI, M. DE O. G. O **Processo de Discretização do Raciocínio Matemático na Tradução para o Raciocínio Computacional: Um Estudo de Caso no Ensino/Aprendizagem de Algoritmos.** Tese (Doutorado em Educação). Curitiba: UFPR, 2009.

TARGINO, R. Quase 6 em 10 alunos do ensino médio de SP saíram da escola com desempenho ruim em matemática em 2011. **UOL Educação**, 7 mar 2012. Disponível em: <http://uol.com/blcqGH>. Acesso em: 20/03/2015

TOBAR, C. M.; ROSA, J. L. G.; COELLO, J. M. A.; PANNAIN, R.. Uma arquitetura de ambiente colaborativo para o aprendizado de programação. **Anais do Simpósio Brasileiro de Informática na Educação.** Vol. 1. No. 1. 2001.

WILSON, B. C.; SHROCK, S. Contributing to success in an introductory computer science course: a study of twelve factors. SIGCSE ’01. **Proceedings of SIGCSE '01.** New York: ACM, 2001.

ANEXO 2

PLANO DE TRABALHO DO ALUNO

NOME DO ORIENTADOR: Helio Henrique Lopes Costa Monte-Alto
TÍTULO DO PROJETO: Explorando abordagens não-convencionais para o ensino de programação e engenharia de software no ensino básico e superior
REGISTRO NO BANPESQ/THALES: 2014016688 DATA DE INÍCIO: 14/10/2014 DATA DE TÉRMINO: 13/10/2016 (previsto)
NOME DO ALUNO: Daniel Antonio Karling DATA DE INÍCIO: 01/08/2015 DATA DE TÉRMINO: 31/07/2016
TÍTULO DO PLANO DE TRABALHO: Investigação da abordagem pedagógica Bootstrap no ensino de computação e matemática em escolas brasileiras
OBJETIVO: Objetivo geral: investigar as metodologias de ensino utilizadas no Bootstrap e sua efetividade, procurando destacar as características que podem levar sua aplicação ao sucesso ou ao fracasso. Como objetivos específicos: • Identificar as metodologias de ensino utilizadas no Bootstrap com base na literatura, procurando identificar seus pontos fortes e fracos. • Realizar um levantamento de abordagens com propostas semelhantes ao Bootstrap e procurar montar um quadro comparativo com base nas características investigadas. • Aplicar aulas experimentais utilizando a abordagem Bootstrap para verificar sua efetividade.
METODOLOGIA/SÍNTESE DAS ATIVIDADES: Será realizada uma revisão bibliográfica sobre metodologias de ensino que abordem o ensino de Computação para o Ensino Básico e, preferencialmente, abordagens que busquem englobar o ensino de Matemática em combinação com o ensino de programação, ou que se utilize de programação para se alcançar a aprendizagem de Matemática. A pesquisa deverá incluir, ainda, uma revisão bibliográfica sobre transferência de aprendizagem entre domínios ou disciplinas diferentes para que se possa avaliar o quanto o Bootstrap é eficaz em sua proposta de transferir a

aprendizagem de resolução de problemas com programação para a álgebra e geometria.

Tendo sido concluída a revisão bibliográfica, um comparativo entre as metodologias encontradas e um confronto com a literatura da área de ensino de Computação, pretende-se avaliar a efetividade da aplicação do Bootstrap. Para isso, pretende-se realizar uma replicação, ou uma adaptação, do experimento realizado por Schanzer *et al.* (2015) com alunos do Ensino Básico brasileiro. Pretende-se, com isso, coletar dados referentes à satisfação dos alunos, a melhoria em suas notas de Matemática na escola, e se as aulas tiveram influência em seu interesse por Computação e Matemática. Como uma replicação do experimento de Schanzer *et al.* (2015), será necessária a aplicação de um teste de álgebra antes e depois do curso aos alunos.

Portanto, a avaliação da hipótese de que o Bootstrap, de fato, ajuda no desempenho dos alunos com a disciplina de Matemática e aumenta o interesse e motivação por Computação e Matemática, será feita por meio de um estudo experimental. A experimentação pode ajudar a construir uma base de conhecimento confiável e, assim, reduzir a incerteza sobre quais teorias, métodos e ferramentas são adequados (CONRADI *et al.*, 2002).

Em síntese, para o desenvolvimento do projeto, o aluno deverá aprimorar seus conhecimentos na área de ensino de Computação e Matemática, transferência de aprendizagem, e sobre a abordagem Bootstrap. Para tanto, as atividades previstas são:

- Revisão bibliográfica sobre metodologias de ensino de Matemática e Computação para o Ensino Básico;
- Revisão bibliográfica sobre transferência de aprendizagem;
- Comparativo entre abordagens e metodologias de ensino;
- Planejamento das aulas experimentais e testes;
- Aplicação das aulas experimentais e testes;
- Análise dos dados experimentais;
- Escrita de artigos e relatórios.

CRONOGRAMA:

**Cronograma de execução												
Descrição das Atividades	Assinalar o mês em que a atividade será executada											
	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	11º	12º
Revisão bibliográfica sobre metodologias de ensino de	x	x	x									

Matemática e Computação para o Ensino Básico												
Revisão bibliográfica sobre transferência de aprendizagem			x	x	x							
Comparativo entre abordagens e metodologias de ensino				x	x	x						
Planejamento das aulas experimentais e testes						x	x	x				
Aplicação das aulas experimentais e testes								x	x	x		
Análise dos dados experimentais										x	x	
Escrita de artigos e relatórios					x	x					x	x

REFERÊNCIAS:

CONRADI, R.; BASILI, V.; CARVER, J.; SHULL, F.; TRAVASSOS, G., A Pragmatic Document Standards for an Experience Library: Roles, Documents, Contents and Structure. **Technical Report CS-TR-4235, University of Maryland**, April 2001.

SCHANZER, E.; FISLER, K.; KRISHNAMURTHI, S.; FELLEISEN, M. **Transferring Skills at Solving Word Problems from Computing to Algebra Through Bootstrap.** SIGCSE' 15. 2015.