

Um Survey sobre a Aprendizagem de Programação no Curso de Sistemas de Informação

Gralicene Viana¹, Albert Lopes¹, Carlos Portela¹, Sandro Oliveira²

¹Faculdade de Sistemas de Informação/Universidade Federal do Pará (UFPA)

²Programa de Pós-Graduação em Ciência da Computação (ICEN/UFPA)

{gracaviana.1988, albert.richard}@gmail.com, {csp, srbo}@ufpa.br

Abstract. *Programming is the basis of the Computing area because it allows the development of problem solving skills and computational thinking. However, due to the degree of difficulty and understanding of their concepts, the disciplines that involve programming have a high failure rate. In this context, this research aims to analyze how students of the Information Systems course at Universidade Federal do Pará learned to program. Thus, the Survey method was applied to 80 students in order to collect and analyze data on programming learning. The main result obtained was the identification of the relationship between prior knowledge and performance in disciplines involving programming and the learning strategies that students use to learn how to program.*

Resumo. *A lógica de programação é a base para a área da Computação, pois permite desenvolver a habilidade de resolução de problemas e o pensamento computacional. No entanto, devido ao grau de dificuldade e compreensão dos seus conceitos, as disciplinas que envolvem a lógica de programação apresentam um alto índice de reprovação. Neste contexto, esta pesquisa objetiva investigar como os alunos do curso de Sistemas de Informação da Universidade Federal do Pará aprendem a programar. Assim, aplicou-se o método de survey com 80 alunos, a fim de coletar e analisar dados sobre a aprendizagem de programação. O principal resultado obtido foi a identificação da relação entre o conhecimento prévio e o desempenho em disciplinas que envolvam programação e as estratégias de aprendizagem que os alunos utilizam para aprender a programar.*

1. Introdução

A programação é essencial nos cursos da área da Computação, pois tem como propósito ensinar aos alunos a criar programas de software, estimulando a aprendizagem do pensamento computacional com uma abordagem para resolver problemas [Wangenheim, Nunes e Santos 2014]. Sendo assim, é importante para a formação profissional que o aluno consiga compreender o que é ensinado nela [Neto 2013]. É tão importante que o Ministério da Educação (MEC) do Brasil determina, via parecer CNE/CES nº 136/2012, aprovado em 8 de março de 2012, a obrigatoriedade do ensino dos conteúdos de lógica e de programação para todos os cursos de formação tecnológica, licenciatura e bacharelado da Ciência da Computação e afins.

No entanto, apesar de sua importância, disciplinas que envolvam programação costumam ter altos níveis de reprovação e evasão [Guedes 2014][Falcão e Junior 2015]. Deve-se a dificuldade dos alunos em assimilar os conceitos abstratos do conteúdo de lógica e programação, complicando, assim, a aplicação prática.

Historicamente, em cursos introdutórios de lógica e programação, as turmas iniciam com uma média de 50 alunos e apresentam um índice de reprovação variando em torno de 60% e 70% [Fabri 2007][Carvalho e Alves 2018][Farias, Oliveira e Silva, 2018]. Adicionalmente, [Fabri 2007] e [Carvalho e Alves 2018] relatam que nos bancos escolares existe um grande interesse em aprender computação, aumentando a preocupação em como lidar com esta demanda e a realidade da evasão escolar nos cursos de Computação.

De acordo com [Guedes 2014], as características particulares de cada contexto devem ser levadas em consideração no ensino de programação com o intuito de minimizar os altos índices de abandono e reprovação nestas disciplinas, comumente registrados na literatura. Portanto, esta pesquisa concentra-se na investigação e análise dos índices de aprovação/reprovação e os meios adotados pelos alunos para o aprendizado dos conteúdos das disciplinas referentes à programação no contexto do curso de Sistemas de Informação da Universidade Federal do Pará (UFPA), campus Tocantins/Cametá. Através do método empírico survey [Kitchenham e Pfleeger 2008], foi coletado os dados referentes aos meios de aprendizagem adotados pelos alunos, que serviram para identificar deficiências e oferecer algumas recomendações aos professores sobre alguns aspectos de ensino. Na sequência, o restante deste trabalho é organizado desta maneira: Seção 2, a apresentação da fundamentação teórica em que se baseou a pesquisa; Seção 3, descrição do survey; Seção 4, apresentação dos resultados e discussões; e Seção 5, exposição das considerações finais e os trabalhos futuros.

2. Fundamentação Teórica

2.1 Contexto

O conhecimento de lógica de programação é requisito essencial nos cursos da área da Computação, sendo importante para o desenvolvimento do raciocínio lógico e formulação de algoritmos [Rocha et al. 2010]. De acordo com [Leal 2014], desenvolver o raciocínio lógico é uma das premissas básicas para o estudante obter sucesso no aprendizado de programação. Adicionalmente, [Leal 2014] afirma que ter a capacidade de abstração de conceitos para resolução de problemas é indispensável para que o aluno tenha êxito na área da Computação.

Entretanto, o ensino e aprendizagem de programação tornou-se um grande desafio para alunos e professores, devido à grande dificuldade que os alunos enfrentam em absorver os conteúdos da mesma [Silva e Trenti 2016]. [Leal 2014] destaca que a maioria dos novatos encontra muita dificuldade, seja pela má formação no ensino básico ou por não obterem afinidade imediata com a área, resultando, segundo [Falcão e Junior 2015], em um elevado índice de reprovação e evasão de alunos do curso de Computação.

Alguns pesquisadores acreditam que uma das causas deste problema reside no fato de que o ensino de lógica e programação no Brasil esteja disponível formalmente a partir da educação superior [Oliveira et al. 2014][Medeiros, Brasil e Aranha 2014]. De acordo com [Oliveira et al. 2014], a aplicação da computação como uma ciência básica é ainda incipiente nos demais níveis de educação brasileiro. Soma-se a isto, a experimentação de tecnologias educacionais com forte apelo ao lúdico [Leite et al.

2013]. Logo, destaca-se a importância de viabilizar o conhecimento tecnológico-científico e estimular o desenvolvimento do pensamento lógico dos estudantes de ensino básico [Oliveira et al. 2014].

Assim, diversas pesquisas foram realizadas acerca do tema ensino em programação de computadores. A maioria destas é voltada para os alunos novatos em programação, que apresentam dificuldades na área de Ciências Exatas [Leal 2014][Mota et al. 2014][Amaral, Silva e Pantaleão 2015]. Somando-se a esses estudos, esta pesquisa busca discutir e identificar estratégias de aprendizagem, principalmente, para alunos novatos no tema.

2.2 O Ensino de Programação em Sistemas de Informação

O Plano Pedagógico do Curso (PPC) de Sistemas de Informação da UFPA destaca seis disciplinas obrigatórias relacionadas à área de programação: algoritmos (68h), laboratório de algoritmos (34h), estrutura de dados I (68h), estrutura de dados II (68h), programação de computadores I (68h) e programação de computadores II (68h). Estas disciplinas totalizam 374 horas (16% do total da carga horária das disciplinas obrigatórias), sendo ofertadas nos quatro primeiros períodos do curso e distribuídas segundo o Quadro 1.

Quadro 1: Disciplinas de Programação do Curso de Sistemas de Informação.

1º período	2º período	3º período	4º período
Algoritmos	Estrutura de Dados I	Estrutura de Dados II	Programação de Computadores II
Laboratório de Algoritmos		Programação de Computadores I	

As disciplinas de algoritmos e laboratório de algoritmos tratam da fundamentação da lógica básica da programação estruturada. A disciplina de algoritmos aborda métodos para solucionar problemas em nível abstrato via conjunto de regras e recursos computacionais. Os principais tópicos são: tipos de dados e operações primitivas; estruturas de controle do fluxo de informação; funções e procedimentos; e algoritmos recursivos. Já a disciplina laboratório de algoritmos visa praticar os conceitos apreendidos naquela disciplina em uma linguagem de programação específica, que, no caso da faculdade, é a linguagem C.

Enquanto as disciplinas abordadas anteriormente criam a fundação do conhecimento em programação, as disciplinas estrutura de dados I e II desenvolvem a estrutura da informação em termos computacionais. Mais estritamente, lidam com as formas de representar e manipular os dados, que mudam em função do cenário da aplicação. Acerca dos tópicos em estrutura de dados I, abordar-se: ponteiros; noções estrutura de dados; estruturas sequenciais e encadeadas; algoritmos de manipulação; noção de recursividade; busca linear e binária; e ordenação de dados. De maneira complementar, estrutura de dados II contempla: árvores e percursos em árvores; árvores de busca; grafos e percursos em grafos; tabela de dispersão.

As demais disciplinas abordam os paradigmas avançados de programação, como orientação a objeto e programação web. A disciplina de programação de computadores I aborda programação orientada a objetos através da linguagem Java.

Visando complementar os conceitos vistos anteriormente, a disciplina de programação II busca aplicar e consolidar os tópicos avançados no desenvolvimento de sistemas orientados a objetos, como desenvolvimentos em camadas, interação com banco de dados e desenvolvimento *web*.

No contexto da UFPA, campus Cametá, [Moura 2018] realizou-se o levantamento da quantidade de reprovações das turmas de Sistemas de Informação, durante os anos de 2011 até 2016, em disciplinas que possuem como base teórica algoritmos. Assim, constatou que 50,6% (334 alunos) reprovaram nas seis disciplinas obrigatórias da área de programação apresentadas no Quadro 1.

A fim de complementar estes dados, solicitou-se ao professor Albert Lopes, Faculdade de Sistemas de Informação (FASI), UFPA, campus Cametá, o levantamento dos dados de reprovação nas disciplinas ministradas por ele e que tem relevância para este estudo: algoritmos, laboratório de algoritmos e estrutura de dados I e II. O professor ministrou, durante os anos de 2016 e 2018, 8 vezes estas disciplinas. O percentual médio de reprovação obtido foi de 43,8%.

3. Survey com Alunos

Segundo [Kitchenham e Pfleeger 2008], survey é um método empírico utilizado para identificar as características de uma ampla população de indivíduos. Apesar de ser mais associado com o uso de questionários para coleta de dados, também pode ser realizado por meio de entrevistas estruturadas. A característica definidora do survey é a seleção de uma amostra representativa de uma população bem definida, e as técnicas de análise de dados usadas para generalizar a partir dessa amostra para a população, em geral para responder a perguntas de taxa básica (apresentadas em percentuais).

Uma pré-condição para a realização de um survey é uma questão de pesquisa clara que pergunta sobre a natureza de uma população-alvo específica [Easterbrook et al. 2007]. Geralmente, é inviável (e desnecessário) consultar todos os membros de uma população, então o survey primeiro identifica um subconjunto representativo como a amostra e determina como chegar a esse subconjunto para coleta de dados. Assim, identificar a unidade de análise é importante para determinar uma técnica de amostragem adequada.

3.1 Design

O survey realizado objetivou coletar informações sobre a aprendizagem dos alunos de Sistemas de Informação em programação, representados por uma amostra. Este survey foi aplicado na UFPA e seguiu os *guidelines* de [Kitchenham e Pfleeger 2008]. Portanto, foram seguidas quatro etapas, conforme ilustra a Figura 1. Inicialmente, na Etapa I, *Fundamentação*, os trabalhos de [Fabri 2007], [Neto 2013], [Wangenheim, Nunes e Santos 2014], [Falcão e Junior 2015] e [Moura 2018] foram analisados, principalmente, no que diz respeito às problemáticas encontradas na aprendizagem de programação. Na Etapa II, *Planejamento*, a população-alvo foi identificada e, também, as questões da pesquisa, o método de coleta e os critérios de análise foram definidos. Em seguida, realizou-se a coleta de dados na Etapa III, *Desenvolvimento*. Durante o período de 18 de Outubro a 19 de Novembro de 2018, o questionário foi divulgado para cerca de 146 alunos, em listas de e-mails, grupos em redes sociais, sistema acadêmico e *in loco* no campus de Cametá da UFPA.

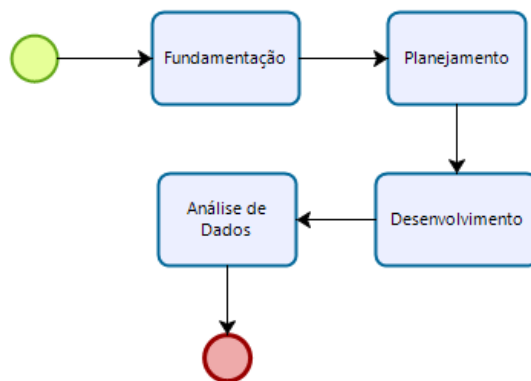


Figura 1: Etapas da Realização do Survey.

Por fim, na Etapa IV, *Análise dos Dados*, buscou-se identificar o perfil dos alunos, destacando-se a forma como eles aprendem programação.

3.2 Participantes

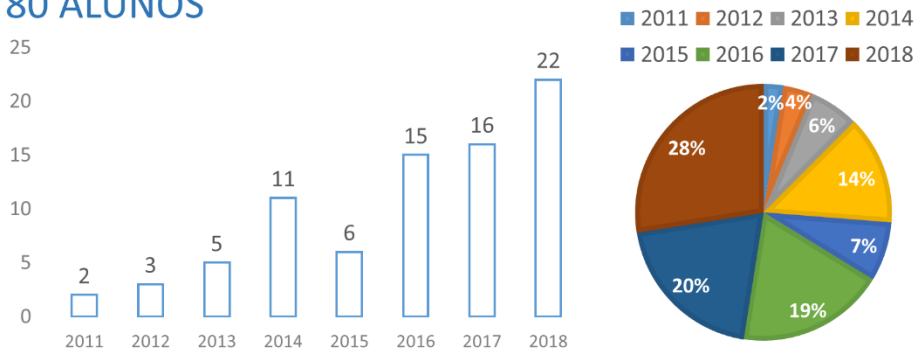
Definiu-se como participantes todas as turmas já ofertadas na UFPA Campus Cametá, contemplando os anos entre 2011 e 2018. Em relação aos egressos (turmas entre 2011 e 2013), apenas os alunos que já haviam defendido o Trabalho de Conclusão de Curso foram considerados; quanto aos graduandos (turmas entre 2014 e 2018), apenas aqueles que foram matriculados no semestre (2018.4), a fim de levantar o perfil atual do curso. Logo, a população estabelecida foi de 146 alunos, conforme mostra o Quadro 2. Os números do Quadro 2 foram levantados via sistema acadêmico e com os professores do curso de Sistemas de Informação da UFPA Campus Cametá.

Quadro 2: População e Amostra do Survey.

Turma	População	Amostra	Percentual da Amostra
2011	5	2	40%
2012	6	3	50%
2013	13	5	38%
2014	13	11	85%
2015	17	6	35%
2016	23	15	65%
2017	30	16	53%
2018	39	22	56%
TOTAL	146	80	54,79%

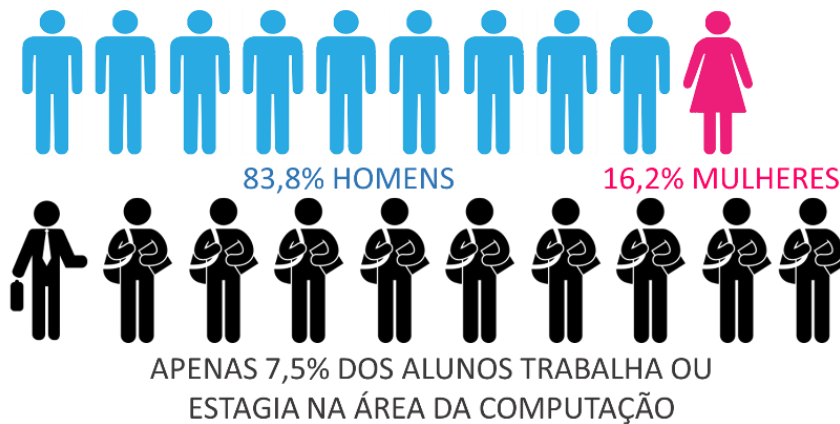
Conforme mostra o Quadro 2, conseguiu-se alcançar 54,79% da população-alvo. Considerando a amostra total, O infográfico 1 ilustra a distribuição dos alunos em relação aos semestres de ingresso. A partir da análise do perfil desta amostra, observou-se 83,8% é composta por homens e 16,2% por mulheres, sendo a média de idade igual a 23 anos. A fim de relacionar a experiência profissional com o aprendizado em programação desses alunos, perguntou-se quem estagia ou trabalha na área da Computação. O resultado mostrou que apenas 7,5% dos alunos possui esta experiência. Tais informações são apresentadas no Infográfico 2.

80 ALUNOS



Amostra de 54,79% da POPULAÇÃO

Infográfico 1: Distribuição da Amostra do Survey.



Infográfico 2: Perfil da Amostra do Survey.

Por fim, perguntou-se aos alunos se participam de grupos de pesquisa, visando identificar possível fonte de experiência na área durante o período acadêmico. Observou-se que 30% da amostra participa de grupos de pesquisa, sendo 58,3% do Laboratório de Abordagens de Ensino Focadas no Aluno (LA FocA), 37,5% do Laboratório de Programação Extrema (LABEX), ambos da UFPA Campus Cametá, e 4,2% do Laboratório de Computação Aplicada (LCA) do Instituto Federal do Pará, conforme mostra o Infográfico 3.



Infográfico 3: Envolvimento dos Alunos com Pesquisa.

3.3 Realização

O survey foi realizado com alunos de Sistemas de Informação das turmas de 2011 a 2018. A coleta de dados ocorreu por meio da aplicação de um questionário, desenvolvido para extrair informações de como esses alunos aprenderam programação. Os procedimentos ocorreram de forma presencial nas salas de aula das turmas de 2016 e 2018, no dia 22 de Outubro de 2018. Para as demais turmas foi disponibilizado um questionário *online* através do link goo.gl/forms/SI3fqMMwMyW901a53, entre dias 18 de Outubro a 19 de Novembro de 2018. A análise de dados ocorreu de forma paralela à coleta, através do apoio ferramental de planilhas eletrônicas.

3.4 Análise dos Dados

Inicialmente, perguntou-se a cada aluno participante se possuía conhecimento prévio de lógica de programação antes de ingressar no curso de Sistemas de Informação. Se sim, como ele havia aprendido: a) através de cursos técnicos; ou b) de forma autodidata. Se o aluno respondesse curso técnico, era solicitado o nome da instituição. Os resultados obtidos são apresentados no Infográfico 4.

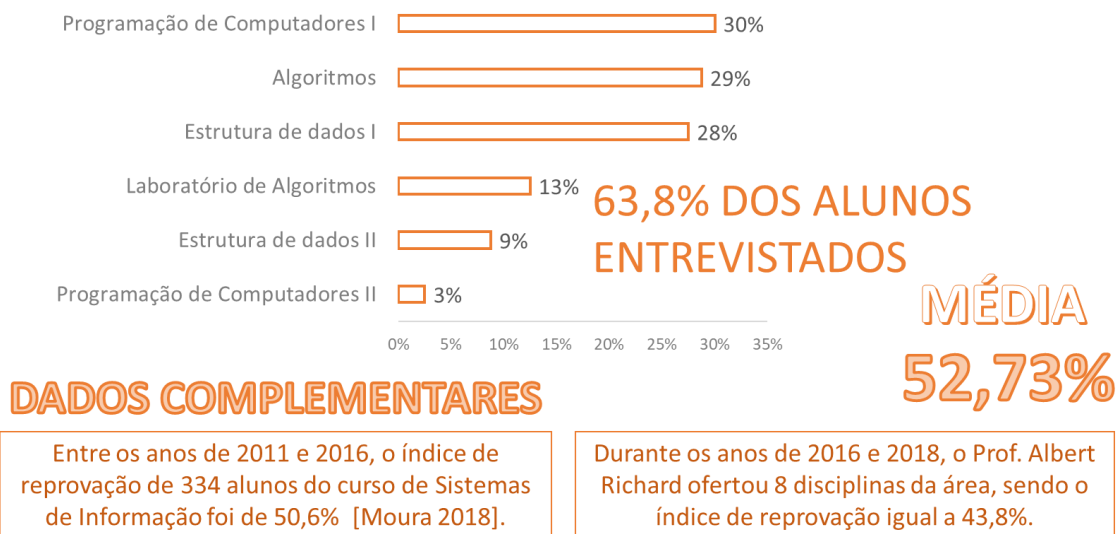


Infográfico 4: Conhecimento Prévio em Programação dos Alunos.

Observou-se que apenas 26,3% dos alunos sabiam programar antes de ingressar no curso e que 38% desses haviam aprendido de forma autodidata. Os demais 62% aprenderam em cursos técnicos, sendo 71% desses formados no IFPA.

Em seguida, questionou-se aos alunos em quais disciplinas da área de programação eles haviam sido reprovados. Os resultados da análise das respostas são apresentados no Infográfico 5. Faz-se necessário destacar que as turmas de 2017 e 2018 ainda não integralizaram as disciplinas do 3º e 4º período (ver Quadro 1).

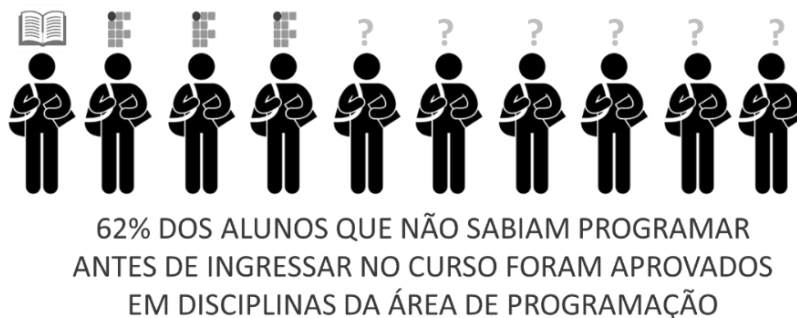
Assim, observa-se que o bloco das disciplinas que possuem maiores índices de reprovação é constituído por Programação de Computadores I, Algoritmos e Estrutura de Dados I. Já as disciplinas de Laboratório de Algoritmos, Estrutura de Dados II e Programação II possuem baixos índices se comparado com as disciplinas daquele bloco. Observa-se que as três disciplinas que mais reprovam são pré-requisitos das outras três que menos reprovam: Programação de Computadores I é pré-requisito de Programação de Computadores II; Algoritmos é a base teórica para Laboratório de Algoritmos; e Estrutura de Dados I é pré-requisito de Estrutura de Dados II.



Infográfico 5: Índice de Reprovação por Disciplina.

A média de reprovação dos alunos entrevistados foi de 63,8%. Devido ao alto valor, comparou-se este resultado com os obtidos por [Moura 2018] e com os dados levantados pelo professor Albert Lopes. Assim, a média de reprovação destes três levantamentos foi de 52,73%. Essa média é condizente com relatos na literatura sobre o índice de reprovação em disciplinas de programação.

A fim de identificar o perfil dos alunos que foram aprovados em todas as seis disciplinas da área de programação, realizou-se a análise apresentada no Infográfico 6.



Infográfico 6: Alunos que Não Sabiam Programar e Foram Aprovados.

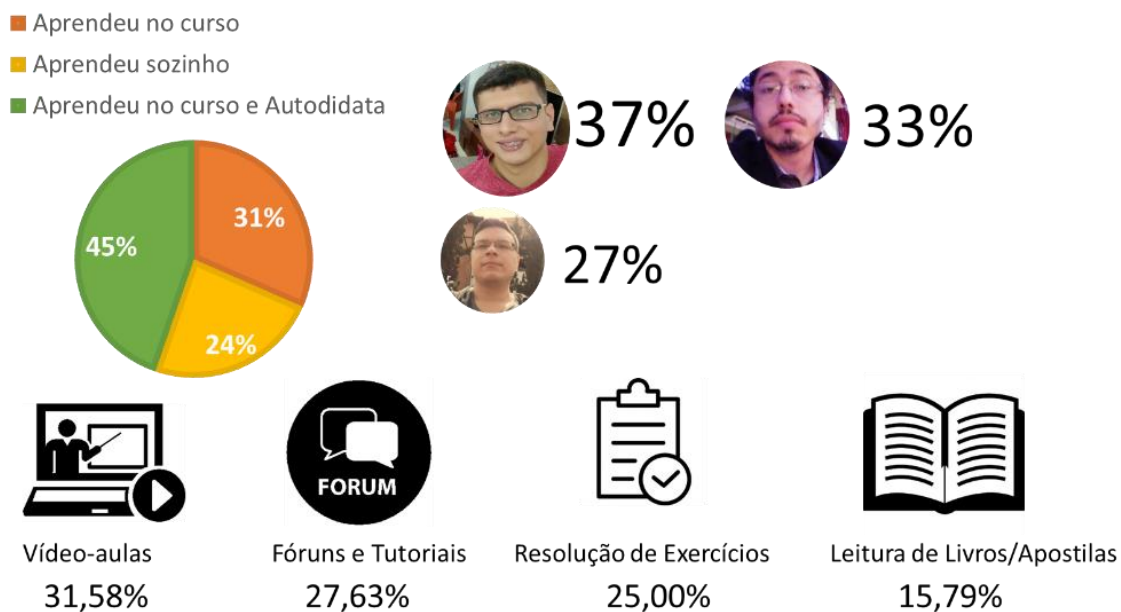
Observa-se que a maioria dos alunos aprovados (62%) não sabia programar antes de ingressar no curso, seguido por 24% de alunos que haviam feito curso técnico e 14% que sabia programação de forma autodidata.

Posteriormente, perguntou-se para os alunos se eles consideravam-se programadores. Caso respondessem sim, solicitou-se que elencassem em quais linguagens programavam. Os resultados obtidos são apresentados no Infográfico 7. A partir da análise das respostas, observou-se que apenas 37,5% dos alunos consideram-se programadores. A linguagem mais conhecida por eles é Java, abordada em programação de computadores I e II, seguida por PHP (geralmente aprendida de forma autodidata pelos alunos). A terceira linguagem mais conhecida foi C, abordada em estrutura de dados I e II. Assim como PHP, as linguagens Javascript e Android são aprendidas fora das disciplinas do curso de Sistemas de Informação da UFPA.



Infográfico 7: Alunos que se Consideram Programadores.

Por fim, questionou-se aos alunos como estes aprenderam a programar. Caso respondessem que aprenderam no curso, pediu-se que citassem quais professores contribuíram para o aprendizado. Os resultados obtidos são apresentados no Infográfico 8.



Infográfico 8: Aprendizagem dos Alunos em Programação.

Observa-se que a maioria dos alunos (45%) aprendeu nas disciplinas do curso e complementa com estudos fora da sala de aula. Quanto à aprendizagem em sala de aula, os professores mais citados foram Diovanni Araújo (37%), Fabricio Farias (33%) e Albert Lopes (27%). Em relação ao estudo autodidata, os alunos aprendem assistindo vídeo-aulas (31,58%), participando de fóruns e lendo tutoriais (27,63%) e resolvendo exercícios (25%). Uma pequena parcela dos alunos aprende a programar a partir da leitura de livros e apostilas.

4. Resultados e Discussões

O principal objetivo do survey foi investigar como os alunos do curso de Sistemas de Informação da UFPA, Campus Cametá, aprendem programação. A partir disto, obteve-se um perfil dos 80 alunos que participaram desta pesquisa. No caso do gênero, observou-se que apenas 16,2% dos alunos entrevistados são do gênero feminino. [Robe, Cadaval e Finger 2017] tentam explicar o reduzido número de mulheres nos cursos de Computação devido, principalmente, à falta de informação sobre esses cursos durante o Ensino Médio, uma vez que a mesma é pouco abordada nas escolas comparada a outros cursos. Os mesmos relatam outros fatores como: falta de motivação; desvalorização da mulher no mercado de trabalho; cultura machista; entre outros.

Na relação entre experiência obtida em estágio/emprego e a aprendizagem em programação, identificou-se que apenas 7,5% dos entrevistados trabalham ou estagiam na área. Três possíveis motivos para isto: a) a baixa média de idade (23 anos); b) falta de experiência; c) aproximadamente, metade da amostra do estudo é constituída por alunos de turmas dos períodos iniciais (do primeiro ao terceiro semestre). Portanto, acredita-se que estes alunos já saem do ensino médio com o objetivo inicial de ingressar no curso a fim de adquirir os conhecimentos e habilidades necessários e só então passam a pensar em um possível estágio/emprego.

Quanto ao conhecimento adquirido em atividades de pesquisa em programação, constatou-se que apenas 30% dos entrevistados participam de grupos de pesquisas. Isto pode estar relacionada com os conhecimentos de alguns alunos em linguagens de programação não ensinadas no curso de Sistemas de Informação e que geralmente são ofertadas através de cursos de capacitação ou projetos desenvolvidos nos grupos de pesquisa.

Em relação ao conhecimento prévio, o percentual dos alunos que responderam que já programavam antes de ingressar no curso foi de 26,3%, cujo 71% destes alunos são de origem de cursos técnicos do IFPA. Consequentemente, constatou-se que os alunos com conhecimento prévio em programação apresentam maior rendimento no curso de Sistemas de Informação. Este resultado reforça a tese de inclusão dos conteúdos relacionados à lógica de programação no ensino básico como medida para redução da evasão nos cursos de Computação. [Farias, Oliveira e Silva 2018] ressaltam que para isto é necessário planejamento e um período de tempo significativo para absorção do conhecimento, por exemplo, os cursos técnicos dispõem em média de 18 meses para estudo, treinamento e prática.

Os resultados do survey revelam também que as disciplinas de programação de computadores I, algoritmos e estrutura de dados I são as que mais reprovam. Isto se deve ao fato de que ambas as disciplinas necessitam de lógica de programação na construção de algoritmos e são pré-requisitos para outras disciplinas mais complexas. Em seguida, constatou-se que os alunos que não reprovam nestas disciplinas, geralmente, são aqueles que possuem conhecimento prévio do assunto, seja por terem feitos cursos previamente ou por serem autodidatas. Aliás, estes buscam como fonte de conhecimento vídeos-aulas, fóruns, tutorias e, como validador, resolução de listas de exercícios.

Outro dado interessante e alarmante: apenas 37,5% dos alunos consideram-se programadores. Acredita-se que seja por causa da motivação dos alunos a cerca da

disciplina, vendo-a mais como fim do que como um ponto de partida, desejando apenas obter um conceito alto ou, na maioria dos casos, uma aprovação.

Já aqueles que continuam praticando programação fora da sala de aula, como ocorrem nos laboratórios de pesquisa ou com aqueles que praticam em casa, se desenvolvem mais como programadores. Desta forma, justificam-se os 45% dos alunos que disseram ter aprendido a programar tanto no curso quanto de forma autodidata. Quanto aos meios na aprendizagem autodidata, constatou-se que a maioria dos alunos tem preferências por vídeo-aulas, resolução de exercícios e fóruns e tutoriais.

A leitura de livros e apostilas, por sua vez, é pouco usada pelos alunos, uma vez que o aprendizado torna-se mais gradativo e pouco atrativo. Além disso, são materiais de difícil acesso por questão de custo, disponibilidade na internet e/ou em bibliotecas.

Ainda em relação aos alunos que se consideram programadores, a maioria (44%) disse saber programar em Java, que é uma linguagem que utiliza o paradigma de Programação Orientada a Objetos (POO). É a linguagem utilizada nas disciplinas de Programação de Computadores I e II, disciplinas finais do bloco de lógica e programação, justificando o percentual maior de alunos que programam nessa linguagem. Outra motivação é de que Java é a linguagem de programação orientada a objetos mais utilizada no mundo de acordo com o levantamento feito pelo site TIOBE¹.

A partir dos resultados obtidos, identificou-se algumas estratégias de aprendizagem em programação utilizadas pelos alunos. Com isto, elaborou-se um conjunto de recomendações, descritas na Subseção 4.3, que podem ser adotadas pelos professores do curso de Sistemas de Informação da UFPA, Campus Cametá.

4.1 Ameaças à Validade

Na realização de um survey, um grande desafio é controlar o viés de amostragem. O viés de amostragem causa problemas em generalizar os resultados da pesquisa, pois os entrevistados na pesquisa podem não ser uma amostra representativa da população-alvo. Neste sentido, baixas taxas de resposta aumentam o risco de viés de amostragem, impactando na validade de conclusão do estudo.

Para lidar com esse viés do survey, atingiu-se uma amostra significativa da população-alvo, correspondente a 54,79% desta. Portanto, destaca-se que esta pesquisa considerou a opinião dos alunos egressos do curso (turmas de 2011, 2012 e 2013). Adicionalmente, contemplou as turmas que atualmente cursam as disciplinas de programação, 2017 e 2018. Estas turmas não foram contempladas na pesquisa de [Moura 2018] e foram representadas por uma amostra, respectivamente, de 53% e 56%. Desta forma, considera-se a amostra obtida como representativa.

A fim de reforçar a validade de conclusão, optou-se por obter a média de reprovações entre os resultados desta pesquisa, o trabalho de [Moura 2018] e o levantamento realizado no sistema de gestão acadêmica pelo professor Albert Lopes. O resultado obtido foi uma média de 52,73%, condizente com a realidade observada no curso de Sistemas de Informação da UFPA Campus Cametá.

Quanto à validade externa, ou seja, as condições que permitem generalizar os resultados desta pesquisa para outros contextos, foi baixa, pois os resultados aqui

¹ <https://www.tiobe.com/tiobe-index/>

obtidos e analisados se aplicam à realidade da UFPA, Campus Cametá. No entanto, algumas recomendações (apresentadas na Subseção 4.3) podem auxiliar professores de outras instituições de ensino a lidar com as problemáticas tratadas por esta pesquisa.

4.2 Limitações

Uma das limitações do survey é o fato de que seus resultados recaem quase que exclusivamente num posicionamento positivista dos pesquisadores, pois o desejo de caracterizar uma população inteira através de técnicas de amostragem requer uma crença no reducionismo. Neste sentido, os resultados apresentados nesta pesquisa não devem ser generalizados para outras instituições de ensino sem a devida análise, comparação e contextualização.

4.3 Recomendações para o Curso de Sistemas de Informação

A partir da análise dos resultados obtidos, algumas implicações e recomendações podem ser relatadas para o contexto do curso de Sistemas de Informação da UFPA, Campus Cametá. Em relação à predominância do gênero masculino no curso, recomenda-se aos professores a criação de um projeto de extensão que replique experiências de sucesso no ensino de programação para mulheres. Destaca-se as iniciativas Technovation², Women Who Code Recife³ e Rails Girls Belém⁴. Estas iniciativas buscam tanto incentivar mulheres a ingressar na área de tecnologia quanto ensinar programação a essas.

Observou-se que alunos que trabalham, realizam estágio ou participam de grupo de pesquisa na área possuem um maior conhecimento em programação e dominam mais linguagens. Neste sentido, a FASI poderia buscar aumentar as oportunidades de estágios, fomentar a criação de empresas (através da Incubadora de Sistemas de Informação) e criar mais grupos de pesquisa a fim de permitir que mais alunos obtenham tais benefícios. Quanto ao desempenho positivo de alunos advindos de cursos técnicos, poderia ser feito um esforço de divulgação do vestibular em Sistemas de Informação em instituições regionais que oferecem esses cursos na área da Computação.

Dado o índice de reprovação em disciplinas de programação, o problema pode estar associado à didática docente e/ou linguagem adotada como base. Neste sentido, o PPC do curso sugere a realização de dupla docência. Assim, recomenda-se que dois professores possam atuar nas disciplinas de programação, a fim de beneficiar os alunos com diferentes didáticas. Adicionalmente, recomenda-se adotar mais de uma linguagem de programação para que os alunos conheçam diferentes paradigmas e sintaxes que, talvez, sejam mais fáceis de assimilar.

O baixo percentual de alunos que se consideram programadores pode representar uma falta de segurança destes em suas habilidades técnicas. Neste contexto, recomenda-se a realização de desafios técnicos, como a Maratona de Programação⁵ e Hackatons⁶. Estas iniciativas tendem a incentivar a capacitação e

² <http://www.technovationbrasil.org/>

³ <https://www.womenwhocode.com/recife>

⁴ <http://railsgirls.com/belem.html>

⁵ <http://maratona.ime.usp.br/>

competitividade dos alunos, fazendo com que melhorem continuamente suas habilidades em programação.

A maioria dos alunos aprendem a programar nas disciplinas do curso e complementam este estudo fora de sala de aula, principalmente através de vídeo-aulas. De acordo com [Raabe, Bernades e Junior 2014], vídeo-aulas para conteúdos ligados à programação são facilmente encontradas, pois muitos professores têm criado materiais de apoio à aprendizagem dos estudantes e divulgado na internet. O fato dos alunos assistirem vídeo-aulas de outros professores pode reforçar a necessidade dos alunos vivenciarem outras didáticas de ensino. Outro ponto observado diz respeito à consulta de conteúdos em fóruns e tutorias, estratégia bastante usada no mercado de trabalho, onde se busca diretamente soluções para uma demanda técnica em programação.

Por fim, destaca-se que os alunos consideram efetiva a realização de exercícios de fixação para aprender a programar. No entanto, a leitura de livros e apostilas não é praticada pela maioria desses alunos. Destaca-se que o professor deve incentivar a leitura desse material, pois esta estratégia de aprendizagem é essencial para que os alunos fundamentem os conceitos básicos, necessários para o desenvolvimento da lógica de programação e conhecimento das sintaxes das linguagens de programação. Se os alunos somente buscarem uma resposta pronta, em fóruns, tutoriais ou vídeo-aulas, ficarão dependentes destes e não desenvolverão habilidades básicas de resolução de problemas e pensamento computacional.

5. Considerações Finais

Este trabalho investigou de que forma os alunos do curso de Sistemas de Informação da UFPA Cametá aprendem programação. Através de um survey, realizado com 80 alunos, observou-se que 26,3% já programavam antes de iniciar o curso e que estes apresentam maior rendimento no curso. Os resultados do survey revelam também que as disciplinas que mais reprovam são programação de computadores I, algoritmos e estrutura de dados I. Estes resultados apontam que os alunos que não reprovam nessas disciplinas, geralmente, são aqueles que possuem conhecimento prévio do assunto e os autodidatas, que adotam como estratégias de aprendizagem assistir vídeo-aulas, resolver lista de exercícios, consultar fóruns e tutoriais.

Tendo em vista que a computação é uma área que cresce a cada dia e carece de profissionais bem qualificados para atender as demandas do mercado de trabalho, esta pesquisa pode contribuir positivamente com este cenário, tendo em vista seu foco na formação de profissionais. Assim, espera-se contribuir cientificamente com a área da Computação, uma vez que, a mesma, oferece dados relevantes para o ensino-aprendizagem de programação, possibilitando que outras pesquisas sejam realizadas a partir das informações aqui descritas. Neste contexto, destaca-se as seguintes questões de pesquisa: Qual o impacto da leitura de livros e apostilas para a aprendizagem de programação? Qual a influência do conhecimento prévio em lógica de programação para o desempenho dos calouros do curso? Quais os benefícios da dupla docência para o ensino de programação?

Adicionalmente, espera-se contribuir socialmente com o ensino de

⁶ <http://brasil.campus-party.org/hackathon/>

programação, sendo que esse apresenta um alto índice de reprovações e evasões. Com os resultados apresentados neste trabalho, é possível traçar estratégias de ensino voltadas para potencializar o ensino-aprendizagem de programação e, conseqüentemente, melhorar o quadro atual do mesmo. Neste sentido, destaca-se que os professores devem oferecer atenção especial para o ensino de mulheres, buscando valorizá-las, motivá-las e inseri-las, através de ações concretas, em um curso frequentado predominantemente por homens. Ressalta-se, também, a importância da realização de estágios e pesquisas na área, a fim de possibilitar que os alunos tenham uma vivência prática dos conteúdos vistos em sala de aula e adquiram conhecimento de linguagens e tecnologias que não são abordadas no curso de Sistemas de Informação. Por fim, destaca-se a necessidade de incentivar os alunos a participarem de competições, como maratonas de programação, que fomentem a sua motivação e competitividade. Estas competições fazem com que os alunos busquem aprender extraclasse e de forma autodidata.

Apesar da relevância dos resultados obtidos, a amostra da população foi baixa (80 alunos). Sendo assim, a maioria das análises se aplica à realidade da UFPA, Campus Cametá. No entanto, as recomendações apresentadas na Subseção 4.3 podem auxiliar outros professores a lidarem com problemas semelhantes aos tratados por esta pesquisa. Como trabalhos futuros, pretende-se investigar outros aspectos que influenciam a aprendizagem de programação, buscando enriquecer a análise e fortalecer as conclusões, como o perfil de aprendizagens dos alunos e suas características psicológicas, cognitivas, sociais e econômicas. Essa pesquisa futura deverá ser realizada em parceria com outros cursos da UFPA Campus Cametá, como Pedagogia e Ciências Sociais.

Referências

- Amaral, L. Silva, G. e Pantaleão, E. (2015). Plataforma Robocode como Ferramenta Lúdica de Ensino de Programação de Computadores-Extensão Universitária em Escolas Públicas de Minas Gerais. Em Simpósio Brasileiro de Informática na Educação (SBIE), páginas 200-208.
- Carvalho, E. Alves, F. (2018). A Eficiência do Ensino de Lógica de Programação na Modalidade a Distância. Em Anais do XXIV Workshop de Informática na Escola (WIE). Natal, Brasil.
- Easterbrook, S. Singer, J. Storey, M-A. Damian, D. *Selecting Empirical Methods for Software Engineering Research*. In: Guide to Advanced Empirical Software Engineering. Springer, 2007. Cap. 11.
- Fabri, J. (2007). O Ensino de Lógica de Programação e o Desenvolvimento de Jogos Educacionais: Um Caso Aplicado aos Alunos do Curso de Licenciatura Plena em Matemática. Revista Tecnologias, Sociedade e Conhecimento.
- Falcão, E. Junior, J. (2015). Desenvolvimento de jogos eletrônicos como metodologia de ensino de Programação para alunos do curso de informática do Instituto Federal Catarinense – Campus Camboriú. Em VIII Mostra Nacional de Iniciação Científica e Tecnológica Interdisciplinar.
- Farias, C. Oliveira, A. Silva, E. (2018). Uso do Scratch na Introdução de Conceitos de Lógica de Programação: relato de experiência. Em Anais do XXIV Workshop de

Informática na Escola (WIE). Natal, Brasil.

- Guedes, E. (2014). Um Estudo Observacional sobre a Disciplina Introdutória de Programação. Em Anais da 20ª Workshop de Informática na Escola (WIE). Dourados, Brasil.
- Kitchenham, B. Pflieger, S. *Personal Opinion Surveys. In: Guide to Advanced Empirical Software Engineering*. Springer, 2008. Cap. 3, p. 63-92.
- Leal, A. (2014). Ensino de Programação no Ensino Médio Integrado: Uma Abordagem Utilizando Padrões e Jogos com Materiais Concretos. Goiânia, 2014. 105p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.
- Leite, V. Senefonte, H. Barbosa, C. e Seabra, R. (2013). VisuAlg: Estudo de Caso e Análise de Compilador destinado ao ensino de Programação. Em *Nuevas Ideas en Informática Educativa*, páginas 637-640.
- Medeiros, T. Brasil, P. Aranha, E. (2014). Um framework para criação de jogos voltados para o ensino de lógica de programação. Em Anais do III Congresso Brasileiro de Informática na Educação (CBIE). Campo Grande, Brasil.
- Mota, F. Ribeiro, N. Emmendorfer, L. Butzen, P. Machado, K. e Adamatti, T. (2014). Desenvolvendo o Raciocínio Lógico no Ensino Médio: uma proposta utilizando a ferramenta Scratch. Em Simpósio Brasileiro de Informática na Educação (SBIE), páginas 377-381.
- Moura, D. (2018). Suporte ao Aprendizado de Lógica Computacional com a Plataforma Algorácil. Trabalho de Conclusão de Curso da Faculdade de Sistemas de Informação, Universidade Federal do Pará.
- Neto, V. (2013). A Utilização da Ferramenta Scratch como Auxílio na Aprendizagem de Lógica de Programação. Em Congresso Brasileiro de Informática na Educação (CBIE). São Paulo, Brasil.
- Oliveira, M. Souza, A. Barbosa, A. e Barreiros, E. (2014). Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. Em Workshop sobre Educação em Computação (WEI), páginas 1525-1534.
- Raabe, A. Bernades, A. Junior, R. (2014). Produção e Avaliação de Videoaulas: Um Estudo de Caso no Ensino de Programação. Em Anais da 20ª Workshop de Informática na Escola (WIE). Dourados, Brasil.
- Robe, R. Cadaval, L. e Finger, A. (2017). Análise da Participação Feminina nos Cursos de Computação. Em Anais do 8º Salão Internacional de Ensino, Pesquisa e Extensão – Universidade Federal do Pampa.
- Rocha, P. Ferreira, B. Monteiro, D. Nunes, D. Goés, H.C. (2010). Ensino e Aprendizagem de Programação: Análise da Aplicação de Proposta Metodológica Baseada no Sistema Personalizado de Ensino. Em *Novas Tecnologias na Educação*. V.8 N°3, Dezembro, 2010.
- Silva, B. Trenti M. (2016). Dificuldades no Ensino-Aprendizagem de Programação de Computadores: Contribuições para sua Compreensão e Resolução. Em Anais do V Simpósio Nacional de Ensino de Ciência e Tecnologia (SINECT).
- Wangenheim, C. Nunes, V. Santos, G. (2014). Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. Em *Revista Brasileira de Informática na Educação*, Volume 22, Número 3, 2014.