

Utilizando o framework 5W2H para compreender o processo de ensino e aprendizagem de programação

Ceres G. B. Morais¹, Jessica N. de F. L. Araujo¹, Rommel W. de Lima¹

¹Departamento de Informática – Universidade do Estado do Rio Grande do Norte -
(Uern)

CEP 59610-210 – Mossoró – RN – Brazil

Resumo. Nesta pesquisa, buscamos compreender como ocorre o processo de ensino e aprendizagem de programação no Ensino Superior. Para tanto, utilizamos uma abordagem mista e um estudo de caso único. Triangulamos as fontes de evidências e parametrizamos os dados de acordo com as categorias do framework 5W2H, à luz da fundamentação teórica desenvolvida nesse estudo. Os resultados obtidos podem servir de suporte para a construção de novos conhecimentos para o desenvolvimento de estratégias que possam contribuir para o ensino e aprendizagem de programação.

Abstract. In this study, our aim was to comprehensively understand the process of teaching and learning programming in Higher Education. To achieve this, we employed a mixed-methods approach and conducted a single case study. We employed triangulation of multiple sources of evidence and parameterized the data according to the categories of the 5W2H framework, guided by the theoretical foundation developed within this study. The findings of this research provide valuable support for the generation of new knowledge and the development of strategies that can enhance the teaching and learning of programming.

1. Introdução

Nos cursos superiores, especialmente na área de Informática, as Unidades Curriculares (UC) de programação desempenham um papel fundamental na formação de alunos, pois são essenciais para o entendimento dos conceitos e fundamentos necessários em UC mais avançadas [Lahtinen et al. 2005, citado por Medeiros 2019].

Morais et al. (2020) destacam que os alunos enfrentam diversas dificuldades na aprendizagem de programação, tais como interpretação de problemas computacionais e estratégias de resolução, desencorajamento diante de desafios, falta de compreensão dos conceitos-chave, habilidades matemáticas insuficientes e falta de motivação. Para minimizar esses problemas, intervenções que promovam o trabalho ativo dos alunos têm sido propostas, possibilitando o protagonismo na aprendizagem, aumentando o engajamento e o interesse pelos conteúdos abordados na UC de algoritmos [Vasconcelos et al. 2019].

Diante desse contexto, este trabalho apresenta os resultados de uma pesquisa baseada em estudo de caso sobre o processo de ensino e aprendizagem de programação no Ensino Superior. A intenção do artigo é apresentar à luz da fundamentação teórica os fatores que cercam o processo, baseados numa triangulação de dados realizada com base

em um estudo de caso desenvolvido. Os resultados obtidos foram analisados conforme as categorias: **O que, Quem, Quando, Onde, Por que, Como e Quanto**, utilizando o *framework* 5W2H [Klock et al. 2016].

Este artigo está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação teórica; a Seção 3 descreve o procedimento metodológico adotado na pesquisa; a Seção 4 apresenta o estudo de caso realizado e os resultados obtidos; a Seção 5 apresenta as considerações finais e sugestões de trabalhos futuros.

2. Fundamentação teórica

Nesta seção são apresentadas a definição e a contextualização da programação no Ensino Superior. O processo de aprendizagem de programação é descrito como uma tarefa complexa, envolvendo desafios aos docentes de Informática [Tavares 2018]. A programação vai além do domínio de uma linguagem específica e requer o desenvolvimento de habilidades e conhecimentos para solucionar problemas [Medeiros 2019; Tavares 2018].

UC de programação são fundamentais em cursos da área de Informática, permitindo o desenvolvimento lógico e algorítmico dos alunos e fornecendo conhecimentos e habilidades para conceber programas e sistemas [Ribeiro et al. 2019; Esteves et al. 2019; Stephan et al. 2020]. No entanto, os alunos enfrentam diferentes níveis de dificuldades ao longo do percurso acadêmico de aprendizagem de programação [Gomes e Mendes 2007; Raabe e Silva 2005]. O método de ensino adotado pelos professores e o método de estudo dos alunos podem influenciar nesse processo [Gomes e Mendes 2007; Santos e Menezes 2019].

A falta de motivação e interesse, a falta de familiaridade com a lógica de programação, a falta de suporte adequado dos professores e a falta de disciplina e organização nos estudos são algumas das dificuldades enfrentadas pelos alunos [Silva et al. 2015; Medeiros 2019; Santos e Menezes 2019]. Além disso, problemas de escalabilidade e infraestrutura também podem afetar o ensino e a aprendizagem de programação [Morais 2022].

Diante disso, educadores têm buscado metodologias, técnicas e ferramentas que promovam o envolvimento dos estudantes e melhorem o ensino-aprendizagem de programação. O Parecer do Conselho Nacional de Educação/Câmara de Educação Superior [CNE/CES 2012] orienta que a metodologia de ensino deve ser centrada no aluno, fortalecendo o trabalho extraclasse e estimulando habilidades de comunicação e trabalho em equipe (CNE/CES nº 136/2012), dentre essas, as metodologias ativas, como o uso de problemas, e a gamificação têm se destacado [Alves et al. 2019; Netto et al. 2017].

Em resumo, a aprendizagem de programação no Ensino Superior apresenta desafios, e diferentes fatores influenciam o processo. Compreender esses desafios e buscar estratégias inovadoras são fundamentais para melhorar o ensino-aprendizagem de programação, como destacado por Medeiros (2019) e Silva et al. (2015).

Com o estudo apresentado nessa fundamentação teórica, pudemos perceber que o

processo de ensino e aprendizagem de programação é complexo e depende de diferentes fatores, os quais refinamos a partir de categorias e parâmetros, a fim de caracterizá-lo. Para tanto, utilizamos o *framework* 5W2H (do inglês *what, who, where, when, why, how, how much*) como suporte para orientar essa pesquisa. Esse *framework* tem seu uso na área de negócios e estratégias, bem como estudos na área de gamificação [Klock et al. 2016]. Também foi utilizado para desenvolvimento da linguagem de programação Q7 [Leite et al. 2005] e adaptada para o desenvolvimento de processo de reuso de requisitos de software [Morais 2010]. Tendo em vista sua flexibilidade, o *framework* pode ser utilizado nos mais diversos contextos, inclusive no âmbito da educação, podendo ser adaptado de acordo com a especificidade de cada situação. A Figura 1 apresenta as categorias e os respectivos parâmetros de análise.

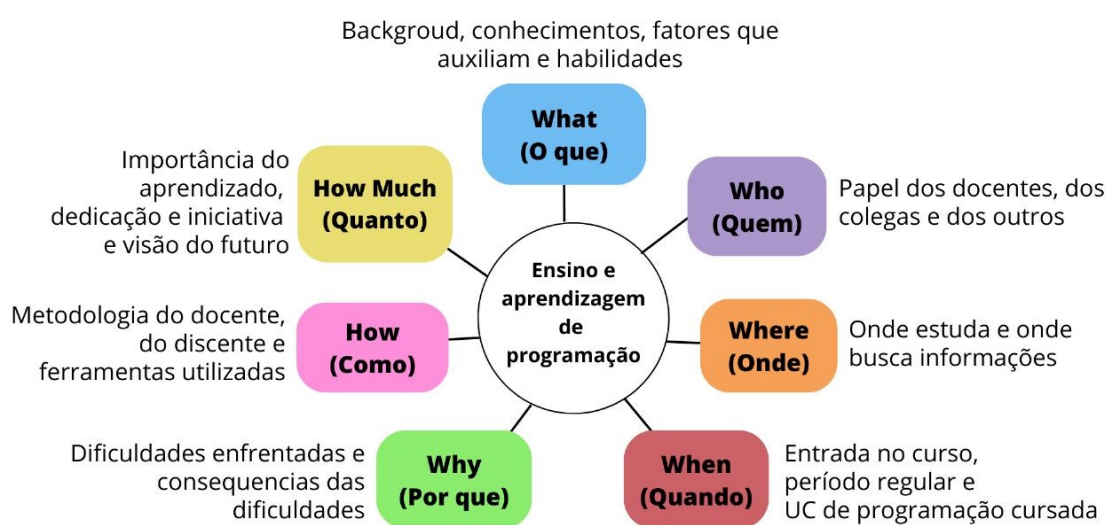


Figura 1: As categorias do 5W2H e os parâmetros definidos para esse estudo

Fonte: Autoria própria

Dessa forma, a categoria **O que** trabalha os fatores, as competências, os conhecimentos e as habilidades necessárias para a aprendizagem de programação, fatores estes importantes para que o aluno obtenha êxito. A categoria **Quem** indica os participantes do processo de ensino e aprendizagem de programação no nível superior. A categoria **Onde**, por sua vez, define o local em que pode ocorrer o processo de ensino e aprendizagem. A categoria **Quando** define em que tempo ocorre o contato dos alunos e professores com a programação. A categoria **Por que** revela as dificuldades e os desafios enfrentados pelo aluno ou turma. A categoria **Como** ajuda a conhecer as metodologias e ferramentas usadas por professores e alunos para ensinar e aprender programação. Por fim, a categoria **Quanto** aponta para o que os envolvidos no processo de aprendizagem pretendem atingir.

3. Procedimento metodológico

Nesta seção apresentamos o procedimento metodológico definido a partir dos objetivos centrais deste artigo. Dessa forma, tendo em vista a complexidade que envolve o processo de ensino e aprendizagem de programação, adotamos uma abordagem mista, combinando elementos qualitativos e quantitativos, com o objetivo de obter uma compreensão mais

abrangente do objeto estudado. A escolha dessa abordagem foi baseada na necessidade de direcionamento adequado das atividades e na busca por maior credibilidade na coleta e análise dos dados [Richardson 2017].

Nesse contexto, o estudo de caso, definido por Yin (2015) como pesquisa empírica que explora um fenômeno dentro do seu contexto real, foi considerado uma estratégia apropriada para nossa pesquisa. Optou-se por um estudo de caso único, que investigou o ensino e aprendizagem de programação em um curso de Ensino Superior de Computação. Essa abordagem permitiu uma análise mais aprofundada de um caso específico, com o intuito de compreender de forma holística o fenômeno em estudo. Para o desenvolvimento do estudo, primeiramente foi realizado um estudo aprofundado sobre o tema, que resultou na fundamentação teórica apresentada na Seção 2, a qual apoiou o desenvolvimento dos instrumentos de coleta de dados necessários.

A escolha do curso nesse estudo de caso foi baseada em critérios específicos, buscando um curso de Ensino Superior com uma matriz curricular que incluísse UC de programação em diferentes níveis de competências. O curso selecionado permitiu uma análise detalhada do processo de ensino e aprendizagem de programação, considerando metodologias adotadas, ferramentas utilizadas, conhecimentos necessários e dificuldades enfrentadas pelos alunos. Para coletar os dados foram realizadas entrevistas com professores e alunos, aplicação de questionários a alunos e análise documental do caso.

Com os dados dos questionários, desenvolvemos estatísticas descritivas que nos deram subsídios suficientes para compararmos e complementarmos as informações obtidas nas análises qualitativas. Com a aplicação das entrevistas e da análise dos documentos oficiais do curso obtivemos dados que nos permitiram realizar uma análise qualitativa, para a qual usamos a estratégia de análise de conteúdo [Bardin 1977].

A partir dos dados tratados, foi utilizada a estratégia de triangulação dos dados, a qual permite que, a partir da sobreposição de diferentes percepções, possamos esclarecer os conceitos e verificar a repetibilidade das observações [Stake 2012], para verificarmos as convergências e divergências existentes nas diferentes fontes, garantindo o rigor dos dados e obtenção dos resultados.

4. Estudo de caso

Nessa seção apresentamos o estudo de caso realizado, com a finalidade de compreendermos, a partir da triangulação dos dados, como ocorre o processo de ensino e aprendizagem de programação no Ensino Superior. Para cada uma das fontes de dados, realizamos a categorização das unidades de registro com base nas categorias definidas na fundamentação teórica. Com isso, foi gerado um arcabouço de dados organizados a partir das categorias do 5W2H que, ao serem sobrepostos, nos permitiram realizar a triangulação desses dados. A seguir, são apresentadas a caracterização do caso objeto de estudo e os resultados obtidos.

4.1. Caracterização do Caso

Criado em 1998, o Curso¹ estudado tem como objetivo formar profissionais preparados para especificar, conceber, desenvolver, implementar, adaptar, produzir, instalar e manter sistemas computacionais, bem como perfazer a integração dos recursos físicos e lógicos necessários ao atendimento das necessidades computacionais de organizações em geral, comprometidos com a ética profissional e com o desenvolvimento tecnológico [PPC 2019]². A estrutura curricular do Curso está fundamentada em quatro grandes áreas de formação conforme orientação das Diretrizes Curriculares Nacionais (DCN) da área de Computação: formação básica, formação humanística, formação complementar e formação tecnológica. O nosso estudo está inserido na grande área de formação básica, que contempla as UC de Lógica e Programação de Computadores.

Sobre o corpo docente, o Curso é composto por 19 professores, oriundos, além de cursos de Ciência da Computação, também de cursos de áreas afins, tais como Engenharia Elétrica e Engenharia de Computação. Também existem profissionais de outras áreas, tais como Física e Matemática, mas com pós-graduação em Computação. Em relação à infraestrutura do Curso, este possui recursos próprios, tais como laboratórios, salas de professores, sala de reunião, além de alguns recursos físicos e equipamentos compartilhados pelos demais departamentos. Ao todo são seis laboratórios, dos quais três são de uso geral e três especializados para seus grupos de pesquisa.

Em relação às UC de programação do Curso, apresentadas em seu PPC (PPC, 2019, pp. 33-34), estas estão distribuídas entre o segundo e o quinto período. Cada uma possui seus pré-requisitos, carga-horária e os objetivos a serem atingidos: Construção de Algoritmos, Estrutura de Dados, Programação Estruturada, Programação Orientada a Objetos e Programação Avançada. Vendo as características de cada uma das UC, podemos entender que são diferentes os conhecimentos e habilidades exigidos para aprender a programar, as dificuldades enfrentadas e as metodologias de ensino e aprendizagem que podem ser adotadas ao longo do processo.

Para o nosso estudo, a definição dos participantes se deu de forma intencional, de acordo com a atuação dos docentes em UC de programação e os alunos que cursaram ou estavam cursando alguma UC de programação. Dos 19 professores integrantes do Curso, sete atuam na docência de alguma UC de programação. Desses, cinco participaram do estudo de caso. Em relação aos alunos, dos cerca de 60 aptos, 32 aceitaram participar respondendo ao questionário e seis participando das entrevistas.

4.2. Resultados Obtidos

A partir da estruturação dos parâmetros, realizada com base nas categorias do *framework* 5W2H, realizamos a análise dos questionários e a leitura flutuante dos depoimentos coletados e dos documentos oficiais do Curso, como forma de nos apropriar do contexto. Os dados foram analisados com base nas categorias e posteriormente identificamos as ideias centrais para então apresentarmos as análises e interpretações obtidas com a triangulação dos dados, apresentadas a seguir.

¹ Chamaremos de Curso para não identificarmos o curso objeto do estudo de caso.

² Dados coletados a partir do PPC do curso, atualizado em 2019. Não inserimos nas referências nessa versão do artigo para não identificarmos o curso objeto do estudo de caso.

4.2.1. Categoria O Que

Essa categoria nos ajuda a caracterizar os fatores que podem influenciar no processo de ensino e aprendizagem e explicitar os conhecimentos e habilidades necessários para aprender a programar. O parâmetro **background** indica que ter experiência prévia pode auxiliar, mas alunos sem contato anterior também conseguem aprender programação. Opiniões divergem sobre a importância do conhecimento em matemática. No parâmetro **conhecimentos**, foram indicados ter conhecimento em lógica e algoritmos, conhecer ferramentas de desenvolvimento de software. Embora não tenhamos obtido dados explícitos, a triangulação dos dados confirma a importância do pensamento computacional no ensino de programação.

O parâmetro **fatores que auxiliam** aponta que autonomia e proatividade são fundamentais no processo de aprendizagem de programação, assim como em outras áreas do conhecimento. Esses fatores podem ser considerados pessoais. O parâmetro **habilidade** enfatiza que resolução de problemas e pensamento lógico são essenciais para aprender a programar.

4.2.2. Categoria Quem

A categoria **Quem** nos ajuda a compreender o papel desempenhado por professores, alunos, familiares e instituições durante o processo de aprendizagem de programação. O papel crucial do professor inclui motivação, *feedback* e apoio durante o ensino e aprendizagem de programação. Através da triangulação dos dados, verificou-se que a figura do docente é considerada fundamental para o ensino. O papel dos colegas de classe inclui entretida e motivação, especialmente durante momentos de desânimo e dificuldades.

Outros atores relevantes no processo incluem apoio social da universidade, projetos de extensão e o papel da família no apoio e incentivo aos alunos. A triangulação dos dados ressalta a importância de receber ajuda de outros envolvidos no processo de ensino e aprendizagem, corroborando com a fundamentação teórica.

4.2.3. Categoria Onde

A categoria **Onde** nos ajuda a conhecer metodologias e ferramentas usadas para o ensino e aprendizagem de programação bem como identificar onde o aluno estuda e busca seus recursos para a aprendizagem.

No parâmetro **onde estuda** verifica-se que a infraestrutura acadêmica, como salas de aula e laboratórios, é fundamental. O ambiente de estudo dos alunos (casa, laboratório, trabalho) também desempenha papel importante. No parâmetro **onde busca informação** é possível verificar que os alunos buscam informações em diversas fontes, dependendo do assunto e do grau de maturidade. Entretanto, a grande quantidade de materiais disponíveis dificulta a seleção de materiais de qualidade, o que pode prejudicar o processo de aprendizagem.

4.2.4. Categoria Quando

A categoria **Quando** aponta os fatores que podem influenciar para o aluno aprender a programar e ajuda a identificar as dificuldades enfrentadas por professores e alunos durante o processo de ensino e aprendizagem, e o que estas dificuldades podem acarretar.

O parâmetro **entrada no curso** permite caracterizar o perfil do aluno, no tocante à sua idade, o que evidencia que se trata de um público jovem, muitas vezes recém-saído do Ensino Médio e provavelmente com pouca ou nenhuma experiência no mercado de trabalho, já que a maioria dos respondentes tinha idade compreendida entre 18 e 25 anos (correspondendo a 75% dos discentes participantes do estudo de caso). Os anos de entrada (cerca de 20% dos respondentes tinham entrado há mais de 5 anos no curso) mostram que alguns alunos passaram do tempo regular do curso, que são quatro anos, o que indica que podem ter passado por dificuldades que culminam no atraso, em desistências ou reprovações. Estes fatores são evidenciados posteriormente nas falas dos alunos entrevistados.

O parâmetro **período regular** apresenta uma fotografia do percurso acadêmico de cada aluno participante da pesquisa: enquanto uns estão iniciando o curso, outros estão nos momentos finais, e outros no meio da caminhada. Esses fatores proporcionam diferentes visões e perspectivas destes em relação ao processo de ensino e aprendizagem de programação. No parâmetro **disciplinas de programação**, conhecemos as UC cursadas pelos alunos e ministradas pelos professores. Percebemos que todas as UC de programação do curso estão contempladas em nosso estudo, uma vez que todas foram sinalizadas como cursadas pelos alunos e como ministradas pelos professores. Assim, podemos ter diferentes visões de professores e alunos para uma mesma UC, tendo em vista as diferentes experiências vividas por estes.

4.2.5. Categoria Por que

A categoria **Por que** nos permite identificar as adversidades enfrentadas por professores e alunos durante o processo de ensino e aprendizagem de programação, bem como as consequências dessas dificuldades.

No parâmetro **dificuldades enfrentadas**, a triangulação dos dados revela que alunos enfrentam inconvenientes devido a problemas em sua formação básica (especialmente no tocante à Matemática e Língua Portuguesa), contratempos de adaptação às novas rotinas e ao ambiente acadêmico. As análises apontam que alunos sem experiência prévia em programação podem enfrentar mais dificuldades no início do curso, incluindo frustrações, desmotivação e até mesmo desistência. No entanto, a falta de experiência também pode ser favorável, pois esses alunos podem ser moldados com o rigor acadêmico para aprender programação.

Embora as fontes de dados não forneçam evidências de que turmas numerosas dificultam o processo de aprendizagem, elas podem prejudicar a gestão da turma, o apoio individualizado aos alunos e a avaliação formativa por parte dos professores. Estratégias como a divisão da turma ou a adoção de métodos que permitam um melhor acompanhamento do professor podem ser consideradas para superar esse desafio. Em relação à heterogeneidade da turma, alunos têm opiniões divergentes em relação à influência dos diferentes perfis e níveis de conhecimento dos alunos na dificuldade de aprendizagem. Alguns alunos não sentem dificuldades, enquanto outros relatam

experiências negativas. Professores também afirmam que a heterogeneidade da turma pode acarretar dificuldades, especialmente nos primeiros anos do curso.

Não compreender o que é solicitado ou não conseguir interpretar os problemas apresentados têm um impacto direto no processo de aprendizagem de programação. Além disso, a falta de estudo aprofundado também dificulta o processo de aprendizagem, dada a natureza complexa da programação e a necessidade de um conhecimento sólido dos conceitos básicos para avançar. Atitudes inadequadas dos alunos, tais como replicar o que é ensinado em sala de aula ou pedir a colegas para fazerem seus trabalhos mascaram a falta de aprendizado real. Alunos e professores reconhecem que a falta de compromisso ou de tempo para estudar afeta negativamente o processo de aprendizagem, o que está em consonância com a fundamentação teórica apresentada.

Não há evidências claras dos dados da triangulação sobre a influência da linguagem de programação adotada pelo professor. No entanto, considerando a fundamentação teórica, é possível refletir que não existe uma linguagem ideal ou método ideal de ensino de programação, pois alunos podem sentir dificuldades ou não, dependendo da linguagem utilizada. Além das dificuldades mencionadas, outros fatores que afetam o processo de ensino e aprendizagem, como a formação docente, a infraestrutura do curso, a metodologia e as ferramentas adotadas pelo professor.

O parâmetro **consequências das dificuldades** permite reconhecer os aspectos pessoais vividos pelos alunos devido às dificuldades enfrentadas. Os dados obtidos mostram que é comum que alunos desistam ou sejam reprovados em UC de programação, sendo que a desistência é mais frequente. Essas recorrências são resultado das diversas dificuldades mencionadas anteriormente. Além disso, a desmotivação e o baixo rendimento nas UC também são consequências comuns.

4.2.6. Categoria Como

A triangulação dos dados na categoria **Como** revela as metodologias e ferramentas usadas no ensino de programação e os fatores que influenciam o aprendizado dos alunos.

No parâmetro **metodologia docente**, fica evidente que a metodologia mais adequada depende do assunto, estrutura e nível da turma, além da infraestrutura do curso. Os professores devem adaptar e modificar as estratégias de ensino conforme necessário. A prática de resolução de problemas, com ênfase na articulação entre teoria e prática, é fundamental para adquirir conhecimentos de interpretação e resolução. A linguagem de programação adotada pelo professor não parece gerar dificuldades, mas sim auxiliar no processo de ensino e aprendizagem. No caso estudado, geralmente os alunos têm liberdade para utilizar a linguagem com a qual se sentem mais confortáveis.

No parâmetro **metodologia discente**, a prática de resolução de problemas continua sendo a forma mais eficaz de aprendizado, permitindo que os alunos adquiram maturidade e domínio dos conceitos da lógica de programação, independentemente da linguagem adotada. A dedicação dos alunos também influencia o processo de aprendizagem, pois mais tempo disponível para estudar possibilita a prática, a busca por recursos de qualidade e a descoberta de metodologias que beneficiem o aprendizado.

Quanto ao parâmetro **ferramentas utilizadas**, não há consenso sobre sua influência no processo de ensino e aprendizagem. No entanto, observamos que os professores influenciam a escolha inicial das ferramentas pelos alunos, mas com o tempo, os alunos se tornam capazes de selecionar aquelas que melhor se adequam ao seu contexto. Na prática, as ferramentas mais comumente utilizadas são os ambientes de desenvolvimento integrado (do inglês *Integrated Development Environment* - IDE), que são amplamente usadas no contexto profissional e possuem aplicabilidade prática no mercado de trabalho. Isso difere do que é mencionado na fundamentação teórica, levando-nos a refletir que, em cursos de Informática no Ensino Superior, os professores tendem a adotar ferramentas que os alunos usarão fora da instituição.

4.2.7. Categoria Quanto

A categoria **Quanto** revela fatores que influenciam o aprendizado de programação e apresenta aspectos emergentes não abordados na fundamentação teórica.

De acordo com o parâmetro **importância da aprendizagem de programação**, alunos se sentem motivados a aprender quando estão envolvidos em algo que realmente gostam. A aprendizagem de programação capacita-os a resolver problemas em diferentes contextos, destacando a importância da criatividade, do raciocínio lógico e da habilidade de resolução de problemas na formação do programador.

Quanto à **dedicação e iniciativa**, notamos que o comprometimento e a iniciativa dos alunos em estudar estão diretamente relacionados ao seu aprendizado. Esses fatores são fundamentais para que os alunos não desistam do curso, busquem novos conhecimentos e enfrentem as dificuldades com mais facilidade. A iniciativa está ligada à proatividade, que é uma característica importante para aprender a programar, conforme mencionado na categoria **O que** e, também evidenciado pelos professores e documentos oficiais.

No parâmetro **visão do futuro**, os alunos percebem as possibilidades oferecidas pela aprendizagem de programação tanto no percurso acadêmico quanto no mercado de trabalho. Mesmo estando em sala de aula, eles buscam adquirir e aprimorar seus conhecimentos e habilidades. A autoconfiança elevada e a vontade de 'voar alto' são fatores motivadores que influenciam positivamente o processo de ensino e aprendizagem de programação.

5. Considerações Finais e Trabalhos Futuros

Este estudo proporcionou respostas para as questões complexas sobre o processo de ensino e aprendizagem de programação. Através da ciência, pudemos evidenciar as dificuldades enfrentadas por alunos e professores, caracterizar os fatores que ajudam os alunos a aprender a programar, identificar competências e habilidades necessárias, inventariar metodologias e ferramentas utilizadas e aplicar o *framework* 5W2H para interpretar os dados.

Os resultados obtidos, Tabela 1, fornecem suporte para a construção de novos conhecimentos e o desenvolvimento de estratégias que melhoram o ensino e aprendizagem de programação, beneficiando todos os envolvidos. Este estudo, realizado

no âmbito das Ciências da Educação, permitiu compreender o processo de ensino e aprendizagem de programação, baseando-se na literatura e nas percepções dos participantes.

Tabela 1. Resultados obtidos

Categoria	Parâmetro	Resultados
O que	background	Experiência prévia pode auxiliar
	conhecimento	Lógica, algoritmos, ferramentas de desenvolvimento e pensamento computacional
	fatores que auxiliam	Autonomia e proatividade
	habilidade	Resolução de problemas e pensamento lógico
Quem	docentes	Apoio fundamental com motivação e <i>feedback</i>
	Colegas	Enteajuda e motivação, principalmente nos momentos de desânimos e dificuldades.
	Outros	Família e apoio social da universidade.
Onde	onde estuda	Infraestrutura da universidade é fundamental.
	onde busca informações	Diversas fontes; a grande quantidade de materiais pode ser um problema.
Quando	entrada no curso	Público jovem com pouca ou nenhuma experiência no mercado de trabalho.
	periodo regular	Percurso acadêmico.
	disciplina cursada	Diferentes visões de professores e alunos para uma mesma UC.
Por que	dificuldade enfrentada	Problemas em sua formação básica, adaptação ao ambiente acadêmico, turmas numerosas, interpretação de texto, heterogeneidade da turma, falta de estudo aprofundados, falta de compromisso, formação docente e infraestrutura do curso.
	consequências das dificuldades	Desmotivação, baixo rendimento, reprovação e desistência.
Como	metodologia do docente	Depende do assunto, nível da turma e infraestrutura. Prática de resolução de problemas, com ênfase na articulação entre teoria e prática.
	metodologia do discente	Prática de resolução de problemas, dedicação e a busca por práticas de aprendizagem que o favoreça.
	ferramentas utilizadas	As comentadas utilizadas são as IDEs
Quanto	importância do aprendizado	Ficam motivados quando envolvidos com algo que gostam
	dedicação e iniciativa	Diretamente relacionado ao seu aprendizado
	visão do futuro	Percebem a importância acadêmica e profissional.

Como trabalhos futuros, sugerimos que outros estudos possam ser realizados, a fim de perceber a realização de um estudo de caso que contemple a observação participante bem como um estudo mais amplo que contemple mais de um caso, e o desenvolvimento de estratégias que possam diminuir ou eliminar as dificuldades enfrentadas por professores e alunos no ensino e aprendizagem de programação, baseadas nos resultados apresentados nesse artigo.

Referências

- Alves, G., Rebouças, A., & Scaico, P. (2019). Coding Dojo como Prática de Aprendizagem Colaborativa para Apoiar o Ensino Introdutório de Programação: Um Estudo de Caso. Anais do Workshop sobre Educação em Computação (WEI), 276–290. <https://doi.org/10.5753/wei.2019.6636>
- Bardin, L. (1977). Análise de Conteúdo. Edições 70 LDA.
- CNE/CES. (2012). Diretrizes Curriculares Nacionais para os cursos de graduação em Computação. Parecer CNE/CES No 136/2012.
- Esteves, A., Filho, A. H., & Raabe, A. (2019). Um Plugin para Aprendizagem de Lógica e Programação no Portugol Studio Baseado em Sistemas Adaptativos com Trilhas de Aprendizagem. Anais dos Workshops do Congresso Brasileiro de Informática na Educação, 8(1), Art. 1.

<https://doi.org/10.5753/cbie.wcbie.2019.1442>

- Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. *International Conference on Engineering Education–ICEE*, 7.
- Klock, A. C. T., Gasparini, I., & Pimenta, M. S. (2016). 5W2H Framework: A guide to design, develop and evaluate the user-centered gamification. *Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems*, 1–10.
- Leite, J. C. S., Yu, Y., Liu, L., Eric, S., & Mylopoulos, J. (2005). Quality-based software reuse. *International Conference on Advanced Information Systems Engineering*, 535–550.
- Medeiros, R. P. (2019). Hello, world: Uma análise sobre dificuldades no ensino e na aprendizagem de introdução à programação nas universidades. Tese de Doutorado. Universidade Federal de Pernambuco. Recife, Pernambuco.
- Morais, C. G. B. (2010). *Cognitio: Um processo para reuso de requisitos* [Dissertação de Mestrado, Universidade do Estado do Rio Grande do Norte]. http://www.dominiopublico.gov.br/pesquisa/DetalheObraForm.do?select_action=&co_obra=184485
- Morais, C. G. B. (2022). *Ensino e aprendizagem de programação: Estudo de caso no Ensino Superior* [DoctoralThesis, Universidade do Minho]. <https://repositorium.sdum.uminho.pt/handle/1822/78752>
- Morais, C. G. B., Mendes Neto, F. M., & Osório, A. J. M. (2020). Dificuldades e desafios do processo de aprendizagem de algoritmos e programação no ensino superior: Uma revisão sistemática de literatura. *Research, Society and Development*, 9(10), e9429109287. <https://doi.org/10.33448/rsd-v9i10.9287>
- Netto, D., Medeiros, L. M., Pontes, D. de, & Moraes, E. de. (2017). Game Logic: Um jogo para auxiliar na aprendizagem de lógica de programação. *Anais Do Workshop Sobre Educação Em Computação (WEI). Anais do XXV Workshop sobre Educação em Computação*. <https://doi.org/10.5753/wei.2017.3546>
- Raabe, A. L. A., & Silva, J. da. (2005). Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. *XIII Workshop de Educação em Computação (WEI'2005)*. São Leopoldo, RS, Brasil, 3, 5.
- Ribeiro, F., Merlin, B., & Fülber, H. (2019). Avaliação do impacto de ambientes gamificados no processo de ensino-aprendizagem de programação de computadores: Uma comparação entre elementos monousuário e multiusuário. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 30(1), Art. 1. <https://doi.org/10.5753/cbie.sbie.2019.803>
- Richardson, R. J. (2017). *Pesquisa social: Métodos e técnicas*. Atlas São Paulo.
- Santos, R., & Menezes, C. de. (2019). Ambiente para Aprendizagem de Programação com apoio Dialogado por Assistentes Inteligentes. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 30(1), Art. 1. <https://doi.org/10.5753/cbie.sbie.2019.1511>
- Silva, T. R. da, Medeiros, T., Medeiros, H., Lopes, R., & Aranha, E. (2015). Ensino-aprendizagem de programação: Uma revisão sistemática da literatura. *Revista*

Brasileira de Informática na Educação, 23(01), Art. 01.
<https://doi.org/10.5753/rbie.2015.23.01.182>

Stake, R. (2012). *A arte da investigação com estudos de caso*. 3a ed. Lisboa: Fundação Calouste Gulbenkian.

Stephan, J., Oliveira, A., & Renhe, M. C. (2020). O Uso de Jogos para Apoiar o Ensino e Aprendizagem de Programação. *Anais do Simpósio Brasileiro de Informática na Educação*, 381–390. <https://doi.org/10.5753/cbie.sbie.2020.381>

Tavares, P. C. (2018). *O impacto da animação e da avaliação automática na motivação para o ensino da programação*. Tese de Doutoramento. Universidade do Minho. Braga, Portugal. <http://repositorium.sdum.uminho.pt/>

Vasconcellos, I. L. B., Tamariz, A. D. R., & Batista, S. C. F. (2019). Planejamento, desenvolvimento e avaliação de um ambiente virtual de aprendizagem gamificado. *RENOTE*, 17(1), Art. 1. <https://doi.org/10.22456/1679-1916.95663>

Yin, R. K. (2015). *Estudo de Caso-: Planejamento e métodos*. Bookman editora.