

Uma Análise Qualitativa sobre as Disciplinas de Fundamentos de Programação e Estrutura de Dados com Grounded Theory

Francisco Romes da Silva Filho, Universidade Federal do Ceará – Campus Quixadá,
romesfilho_cc@alu.ufc.br, 0000-0002-4187-9774

Emanuel Ferreira Coutinho, Universidade Federal do Ceará – Campus Quixadá,
emanuel.coutinho@ufc.br, 0000-0003-2233-7109

Resumo: O ensino de lógica de programação e codificação são tarefas complexas que requerem muito esforço por parte dos docentes. Isto se deve ao fato que os conteúdos abordados são muitas vezes novidade para os alunos, tornando-se um desafio. Uma dificuldade comum é como ensinar de maneira que o aluno se motive a aprender, ao mesmo tempo em que utilizam ferramentas de codificação mais avançadas e aplicação dos conceitos na prática. O objetivo deste trabalho é analisar qualitativamente as disciplinas de Fundamentos de Programação e Estrutura de Dados, identificando pontos fortes e fracos para a melhoria nas disciplinas. Para isto, utilizou-se Grounded Theory. Como conclusões, identificou-se a necessidade de uma didática mais adequada, adequação do conteúdo ao tempo das aulas, e maior suporte tecnológico.

Palavras Chaves: programação de computadores, análise qualitativa, grounded theory.

A Qualitative Analysis between the Disciplines of Fundamentals of Programming and Data Structure with Grounded Theory

Abstract: Teaching programming logic and coding are complex tasks requiring a lot of effort by professors. This is due to the fact that the contents covered are often new to students, becoming a challenge. A common difficulty is how to teach in a way that the student is motivated to learn, while preparing for the use of more advanced coding tools and concepts application in practice. The objective of this work is to qualitatively analyze the disciplines of Fundamentals of Programming and Data Structure, identifying strengths and weaknesses for improvement in the disciplines. For this, we used Grounded Theory. As conclusions, it was identified the need for a more adequate didactics, adaptation of the content to the time of the classes, and greater technological support.

Keywords: computer programming, qualitative analysis, grounded theory.

1. Introdução

Ensinar programação de computadores é uma tarefa que atualmente exige muito dos professores. Como o ensino de programação caminha sempre junto com a lógica de programação, que envolve conceitos matemáticos, o desafio se torna ainda maior.

O ensino de qualquer disciplina técnica constitui um conjunto de atividades desafiadoras para o docente, tanto pelo constante surgimento de novas tecnologias e ferramentas, quanto pela busca por estratégias para se ajustar às novas práticas de aquisição de conhecimento adotadas pelos aprendizes atuais (Coutinho et al., 2016). Também há o desafio de relacionar a teoria com a prática para os alunos, pois geralmente eles ingressam nos cursos de graduação com perfis e níveis de conhecimento variados. Nesse contexto, disciplinas que pretendem alinhar teoria com prática são normalmente muito difíceis de se conduzir devido a necessidade de se tratar fatores técnicos (linguagens de programação, componentes, ferramentas etc) e humanos (disponibilidade, comunicação, gestão etc), tudo ao mesmo tempo (Coutinho et al., 2016). Além disso, a tarefa de motivar os alunos e os próprios professores não é uma atividade fácil, se

verificando particularmente em disciplinas de primeiro ano de cursos de graduação. Isto se deve ao fato que os conteúdos abordados muitas vezes são novidade para os alunos, constituindo um desafio para eles.

A Educação em Computação (EC) é uma área interdisciplinar que exige competências mais específicas de seus pesquisadores (Bispo Jr et al., 2020). Suas duas principais áreas são: Educação e Ciência da Computação. Muitas atividades relacionadas à programação e codificação estão diretamente relacionadas com a EC, seja para o ensino, avaliação ou suporte, contemplando todos os níveis formais do sistema educacional.

Nesse sentido, seria possível afirmar que cada vez mais os estudantes chegam com menor conhecimento prévio sobre matemática e lógica nos cursos de graduação. Vários trabalhos na literatura discutem diferentes relatos do ensino de lógica de programação e codificação em semestres iniciais de cursos de graduação (Ferreira et al., 2016; Coutinho et al., 2017). Uma dificuldade comumente discutida e como ensinar lógica de programação de forma a motivar o aluno a aprender, ao mesmo tempo em que eles se preparam para a utilização de ferramentas de codificação mais avançadas e aplicar conceitos de lógica de programação na prática.

Para auxiliar os alunos a superar as dificuldades e motivá-los no processo de aprendizagem, as estratégias de ensino utilizadas pelos professores são um fator determinante (Zanin et al., 2019). Esta importância justifica-se uma vez que, quando o professor entende a forma como o aluno aprende (lendo, escrevendo, ouvindo, ou fazendo), ele pode preparar suas aulas utilizando meios de ensino que explorem o método mais adequado para a turma. Outra estratégia seria uma discussão acerca da reflexão sobre o erro e sua interpretação em sala de aula, dado seu potencial educativo (Castro, Tedesco, 2020). Na programação de computadores, o erro é comum e é possível se aprender muito com ele. Assim a teoria e prática poderiam estar alinhadas.

No contexto desse trabalho, a articulação entre teoria e prática no âmbito de um curso de graduação não deve estar restrita aos estágios, nem tampouco aos semestres que encerram o curso. Os estímulos às práticas já podem ocorrer desde os semestres iniciais, em disciplinas como Fundamentos de Programação e Estrutura de Dados, onde a dinâmica de ensino mescla a exposição tradicional de conceito com sua prática em laboratório, promovendo e simulando ações que realizariam no mercado.

O objetivo deste trabalho é analisar qualitativamente as disciplinas de Fundamentos de Programação e Estrutura de Dados, e assim identificar relacionamentos entre pontos fortes e fracos, para possibilitar uma melhoria nas disciplinas. Para isto, um questionário aplicado a turmas da graduação foi aplicado e os resultados analisados por meio de procedimentos de *Grounded Theory* (Corbin, Strauss, 2014). Como contribuições principais deste trabalho, obtivemos recomendações para as disciplinas de programação, como uma didática mais adequada para as disciplinas e maior suporte tecnológico, que podem auxiliar na melhoria da qualidade do ensino e aprendizagem.

Este trabalho encontra-se dividido nas seguintes seções além desta introdução: na Seção 2 alguns trabalhos relacionados são discutidos; a Seção 3 apresenta os materiais e métodos aplicados ao trabalho; na Seção 4 são expostos os resultados; na Seção 5 os resultados são discutidos; e por fim, a Seção 6 apresenta as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Aprender a programar por meio de uma linguagem de programação computacional não é uma tarefa fácil para iniciantes, principalmente para quem não possui uma base matemática e lógica (Coutinho et al., 2017). Diversos fatores prejudicam o ensino de programação, tais como: falta de conhecimentos básicos do ensino médio, desmotivação,

falta de maturidade do aluno e professores despreparados. Além disso, programar requer esforço e dedicação. Coutinho et al. (2017) analisaram experiências obtidas em cinco turmas da disciplina programação de um curso de graduação. Esta disciplina tem como objetivo prover ao aluno conceitos de lógica de programação para o desenvolvimento de soluções de problemas, através do uso das estruturas básicas. Verificou-se que o desempenho acadêmico dos alunos por meio de uma análise da média final dos alunos, a quantidade de faltas, e suas relações.

Os altos índices de reprovação e evasão de estudantes universitários nas disciplinas iniciais de programação apresentam uma estatística preocupante para coordenadores de cursos de áreas tecnológicas (Barros et al., 2019). Muitas vezes, o problema da reprovação dos estudantes nessas disciplinas é apontado como um fator influenciador da evasão dos cursos. Barros et al. (2019) propuseram a utilização de técnicas de mineração de dados educacionais para tentar prever o desempenho dos alunos na disciplina de Lógica de Programação em um período inicial de um curso de graduação, por meio do desempenho nas disciplinas do primeiro período do curso. Os resultados mostraram que é possível inferir o rendimento dos estudantes com uma precisão de até 77%. Esta informação é útil para a promoção de ações para evitar reprovações e evasão, e customizar o ensino de lógica de programação.

Os elevados níveis de insucesso nas disciplinas introdutórias de programação vem sendo alvo de inúmeras questões e estudos ao longo dos anos (Holanda et al., 2019). Muitos pesquisadores vêm buscando encontrar soluções que possam contornar este cenário, propondo metodologias e técnicas. Diante desta situação, os autores conduziram uma revisão sistemática da literatura, visando compreender o panorama atual de pesquisa das publicações científicas no Brasil sobre as estratégias adotadas no ensino de programação introdutória em cursos superiores, entre os anos de 2014 a 2018. Os resultados obtidos mostram que softwares educativos, jogos digitais e metodologias específicas estão sendo desenvolvidas e utilizadas com intuito de amenizar os índices de insucesso nessas disciplinas.

Richter et al. (2019) desenvolveram uma revisão sistemática da literatura para investigar as principais tecnologias educacionais e estratégias pedagógicas utilizadas como mediadoras para o ensino de programação de computadores, com foco em Programação Orientada a Objetos. Dos 136 estudos analisados, os softwares Scratch e Alice constaram como os mais citados. O estudo também indicou que as abordagens por oficinas, aprendizagem baseada em problemas e baseada em projetos como sendo as mais exploradas.

Todos os trabalhos citados nesta seção realizaram pesquisas com programação de computadores, discutindo metodologias, desempenho e relatos de disciplinas. Entretanto, em nenhum desses trabalhos se realizou uma análise qualitativa, muito menos utilizou-se *Grounded Theory*. Nosso trabalho pretende utilizar esta técnica para repassar à comunidade de EC uma nova perspectiva de análise da programação em computadores.

3. Material e Métodos

O objetivo desta pesquisa é analisar qualitativamente o resultado de uma consulta executada por meio de um questionário *online* aplicado aos alunos de algumas turmas de graduação, das disciplinas Fundamentos de Programação (FUP) e Estruturas de Dados (ED). Esta pesquisa é de caráter exploratório. Segundo Gil (2008), pesquisas exploratórias têm como objetivo proporcionar maior familiaridade com o problema, visando torna-lo mais explícito ou construir hipóteses. Estas pesquisas também almejam o aprimoramento de ideias ou a descoberta de intuições. Seu planejamento é, portanto,

bastante flexível, de modo que possibilite a consideração dos mais variados aspectos relativos ao fato estudado. Quanto à análise dos dados, esta será qualitativa.

A disciplina de Fundamentos de Programação possui a seguinte ementa: Algoritmos, Conceitos Fundamentais de Programação, Expressões, Controles de Fluxo, Funções e Procedimentos, Ponteiros, Vetores e Matrizes, Cadeias de Caracteres, Alocação Dinâmica, Tipos Estruturados e Arquivos. A disciplina de Estrutura de Dados possui a seguinte ementa: Noções de análise de algoritmos, Recursividade, Tipos Abstratos de Dados, Algoritmos de Ordenação, Listas Sequenciais e Encadeadas, Pilhas, Filas, Árvores. A ideia por trás deste trabalho é investigar o impacto da disciplina de Fundamentos de Programação para a disciplina Estrutura de Dados.

O questionário foi composto por questões demográficas, para se identificar o perfil do aluno, questões sobre a disciplina, para se identificar o contexto da disciplina e relação com disciplinas pré-requisitos, e questões abertas, de texto livre para o aluno expor sua opinião sobre os pontos fortes e fracos das disciplinas em questão e apontar possíveis melhorias. A Tabela 1 expõe as questões do questionário.

Tabela 1 – Questões do questionário *online* (QD = questão demográfica, QC = questão de contexto, QA = questão de opinião com texto livre).

QD01	Qual o seu curso?
QD02	Qual seu semestre de ingresso?
QD03	Qual sua idade?
QC01	Você está cursando FUP?
QC02	Você está cursando ED?
QC03	Qual foi, ou qual está sendo sua dificuldade com a disciplina de FUP?
QC04	Qual foi, ou qual está sendo sua dificuldade com a disciplina de ED?
QC05	Com qual frequência participa das monitorias da disciplina?
QA01	Pontos fortes da disciplina de FUP?
QA02	Pontos fortes da disciplina de ED?
QA03	Pontos fracos da disciplina de FUP?
QA04	Pontos fracos da disciplina de ED?
QA05	O que poderia ser feito para melhorar a base em FUP?
QA06	O que poderia ser feito para melhorar a disciplina de ED?

A análise qualitativa deste trabalho foi inspirada nos procedimentos apresentados em (Ferreira et al., 2018). Nessa análise, nós utilizamos procedimentos da metodologia *Grounded Theory* (Corbin e Strauss, 2014). A *Grounded Theory* visa criar uma teoria a partir dos dados coletados e analisados sistematicamente, sendo composta por três fases: (1) codificação aberta, (2) codificação axial e (3) codificação seletiva.

Na codificação aberta, se executa uma quebra, análise, comparação, conceituação e categorização dos dados (Corbin e Strauss, 2014). Na codificação axial, associam-se as categorias às suas subcategorias, formando categorias mais relacionadas e densas. Por fim, na codificação seletiva, se identifica a categoria ou ideia central do estudo, correspondente à teoria na qual todas as categorias estão relacionadas. O processo de codificação finaliza quando nenhum novo dado adiciona novos conhecimentos ao processo de análise da categorização. Strauss e Corbin explicam que o pesquisador pode utilizar apenas alguns passos para atingir seu objetivo de pesquisa (Corbin e Strauss, 2014). Nessa pesquisa, nós utilizamos apenas as fases 1 e 2 da *Grounded Theory* para a identificação das categorias e suas relações. Adicionalmente, para evitar tendências nas análises, outro pesquisador revisou o resultado.

Algumas citações dos participantes foram capturadas para destacar aspectos das questões de opinião. Estes participantes estão identificados na pesquisa com a seguinte nomenclatura, para preservar a anonimidade: letra maiúscula “P” seguida por um número que corresponde à sequência de resposta do questionário.

4. Resultados e Análises

O questionário foi aplicado em 2019.1, na Universidade Federal do Ceará, campus Quixadá, em 5 turmas de 3 professores diferentes. O critério para a seleção destes professores foi devido ao fato deles estarem ministrando em 2019.1 a disciplina Estrutura de Dados, que possui Fundamentos de Programação como pré-requisito. Apenas 28 alunos responderam o questionário.

As questões demográficas traçaram um perfil dos alunos que responderam ao questionário. QD01 apresentava o curso de graduação de quem respondeu, sendo os seguintes: Engenharia de Computação (13), Sistemas de Informação (9), Ciência da Computação (4) e Engenharia de Software (2). Os semestres de ingresso dos alunos, informados em QD02, foram os seguintes: 2019.1 (0), 2018.1 (26), 2017.1 (0), 2016.1 (1), 2015.1 (0) e 2014.1 (1). QD03 apresentou a faixa etária dos alunos: entre 17 e 20 (24), e entre 21 e 25 (4). A Figura 1 exibe as questões demográficas.

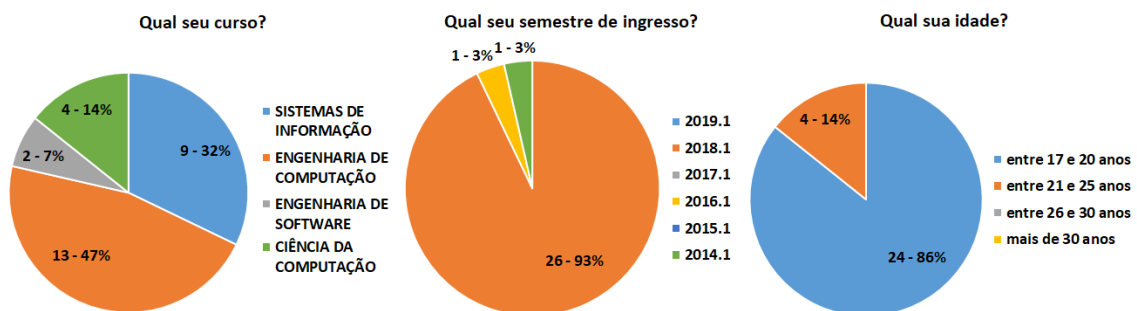


Figura 1 - Gráficos das questões demográficas.

As questões específicas sobre as disciplinas permitiram contextualizar a disciplina em relação a disciplinas pré-requisitos. Em relação estar cursando FUP (QC01), 100% dos alunos informaram que já cursaram. QC02 informa se o aluno está cursando ED, e 16 alunos responderam que reprovaram e estão cursando novamente, enquanto que 12 estão cursando sem terem reprovado. A Figura 2 exibe as demais questões de contextualização das disciplinas do ponto de vista de dificuldades e monitoria. QC03 indicou quais dificuldades na disciplina de FUP, sendo elas: não apresentaram dificuldades (13), didática (4), fatores externos (4), linguagem de programação (4), metodologia (2), lógica (1). Da mesma forma, QC04 aponta as dificuldades em ED: linguagem de programação (12), metodologia (7), fatores externos (5), não apresenta dificuldades (3) e lógica (1). Por fim, QC05 questiona a frequência na qual o aluno participa das monitorias da disciplina, obtendo como respostas: às vezes (13), nunca (11) e sempre que possível (4).

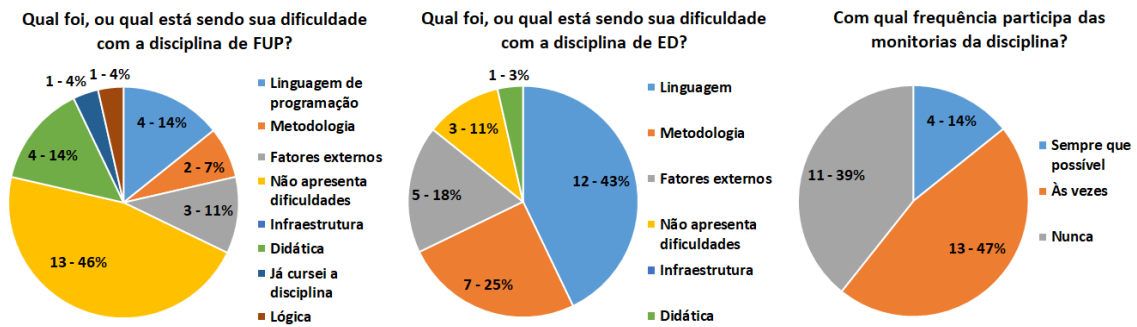


Figura 2 - Gráficos das questões de contextualização das disciplinas.

As questões QA01, QA02, QA03 e QA04 foram analisadas por meio da *Grounded Theory*. As Figuras 3 e 4 exibem as categorias e as relações identificadas, para os pontos fortes e para os pontos fracos das disciplinas, respectivamente. Neste trabalho, 10 categorias foram identificadas, dispostas a seguir por ordem decrescente de citações nas respostas: Didática (29), Conteúdo (16), Suporte (14), Compreensão (12), Valor (12), Experiência (8), Tecnologia (8), Opinião (7), Atividades (4) e Metodologia (2).

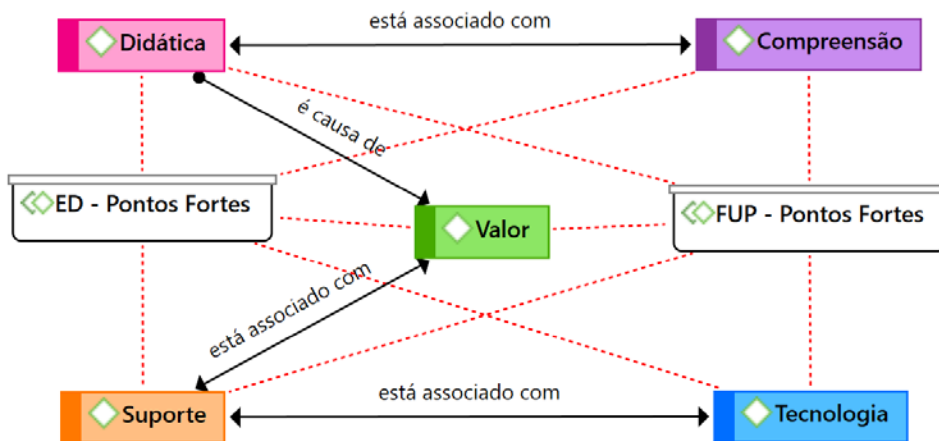


Figura 3 - Categorias e relações identificadas para os pontos fortes para FUP e ED.

Em relação aos pontos fortes, todas as categorias identificadas estavam relacionadas às duas disciplinas. As relações identificadas para os pontos fortes foram: Didática está associada com Compreensão, Didática é causa de Valor, Valor está associado com Suporte, e Suporte está associado com Tecnologia.

Didática e Compreensão estão associados, como reforçado pela citação de P23 “*Importante para iniciar o entendimento de programação. Estimula o estudante a querer fazer mais.*”. A didática sempre é um aspecto importante relacionado a pesquisas de ensino de qualquer área do conhecimento, e a maneira como o professor ministra aulas é de suma importância, como relatado por P26 “*A explicação do professor era muito clara sobre o conteúdo então ficava mais fácil o entendimento*”. Além disso, Didática também é causa de Valor, descrito no comentário de P15 “*O entendimento da necessidade e da utilidade das diversas estruturas de dados*”.

Um problema comum é o suporte tanto ferramental quanto manutenção de equipamentos em disciplinas de programação. A relação Valor está associado com Suporte ressalta justamente esse ponto, como no discurso de P6 em “*Ajuda a alunos que não tiveram contato com a programação antes da faculdade a ter noções básicas sobre o assunto*”. Da mesma forma, o Suporte está associado com Tecnologia, como no

comentário de P3 em “*Em FUP consegui apresentar o que o professor pediu. Entendimento da disciplina em C*”, que é diretamente relacionado a tecnologia.

Já para os pontos fracos, nem todas as categorias identificadas estavam relacionadas com as duas disciplinas. Para FUP, metodologia e tecnologia não estavam relacionadas. Para ED, apenas atividades não estava relacionada. As relações identificadas para os pontos fracos foram: Tecnologia é parte de Metodologia, Metodologia está associado com Didática, Didática é causa de Opinião, Conteúdo está associado com Didática, Tecnologia é parte de Conteúdo, Atividades é parte de Conteúdo, Atividades é causa de Didática, e Experiência é causa de Opinião.

Para a relação de associação entre Didática e Opinião, a forma de ministrar aulas sempre é um aspecto delicado tanto para alunos quanto para professores, como descrito por P22 em: “*Metodologia de ensino, às vezes, um pouco rápida, se tornando um tanto difícil para alunos que não possuem as noções básicas em programação de acompanhar os conteúdos*”. A relação Experiência é causa de Opinião mostrou que experiência auxilia muito tanto na didática do professor quanto no aprendizado dos alunos, destacado por P23 em “*Pode ser difícil para quem nunca teve contato com programação.*” e por P1 em “*Pouco conhecimento prévio sobre o conteúdo*”. Atividades ou práticas são essenciais em disciplinas de programação, já que não adianta apenas o professor apresentar material. nesse contexto, duas relações surgiram com a categoria Atividades.

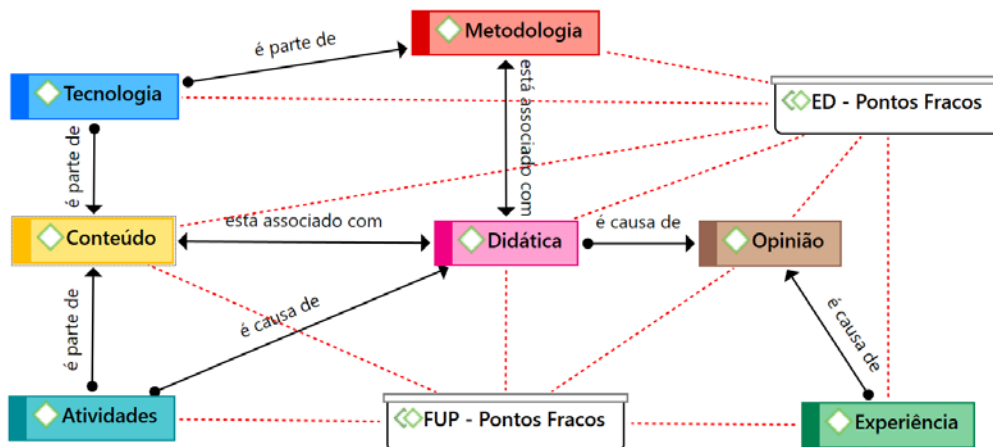


Figura 4 - Categorias e relações identificadas para os pontos fracos para FUP e ED.

A relação Atividades é causa de Didática muitas vezes se relaciona com a maneira na qual o professor ministra suas disciplinas. Isso foi ressaltado por P2 com o seguinte discurso: “*Depende do professor, aconselho fazer que nem o <anônimo>, cobrar questões no moodle cada aula e com prazo de entrega, fazendo assim o aluno estudar*”. A outra relação foi Atividades é parte de Conteúdo. Nesta relação, há a observação de ementa com o que estava sendo ministrado e como, comentado por P26 em “*Alguns conteúdos importantes pra continuidade do curso como recursão e até mesmo ponteiros eram pouco explorados. Então mais tarde senti dificuldades de aplicar esses conteúdos*”. Para as duas relações, P25 respondeu algo que se alinha com ambas: “*Falta da aplicação dos conhecimentos dinâmicos das linguagens. Alocação de espaço, ponteiros, etc.*”.

Algumas vezes a tecnologia utilizada na disciplina força abordagens e técnicas a serem utilizadas. A relação Tecnologia é parte de Metodologia relata isso, como em “*A mudança repentina de linguagem ainda mais de FUP para ED tende a ter uma difícil adaptação*”, redigido por P13. Tecnologia é parte do Conteúdo foi comentada como uma relação que causa impacto devido às mudanças que ocorrem nas disciplinas, apontado na

resposta de P13 “A mudança repentina de linguagem ainda mais de FUP para ED tende a ter uma difícil adaptação”. Por fim, a relação Conteúdo e Didática estão associados fortemente nas duas disciplinas, muitas vezes relacionados à organização, como informado por P14 em “O professor poderia ter abordado mais assuntos que seriam utilizados em ED”. Também houveram situações de duração das atividades, como apontado por P4 em “Muito conteúdo prático (obviamente), para pouco tempo”.

Metodologia e didática sempre caminham juntas, portanto a relação Metodologia e Didática estão associados era esperada surgir nos comentários. P26 relatou essa relação de maneira bastante completa com o seguinte comentário: “Na minha experiência com os conteúdos dado em ED, eles são apresentados de uma maneira equivocada, primeiro é mostrado pra que serve e os cenários que são usados, até aqui tudo bem. Depois o professor apresenta seu código, e pede pro aluno implementar, acho que é aqui que fica minha queixa, quando o professor mostra seu código, o aluno fica preso na maneira de como o professor fez e limita sua criatividade pra outros tipos de alternativas.”.

A Figura 5 apresenta o resultado consolidado da QA05 e QA06, que indicam possíveis melhorias para as disciplinas FUP e ED. A sugestão de mais contato com monitoria para FUP e melhorar o conteúdo para ED como sugestões mais citadas.

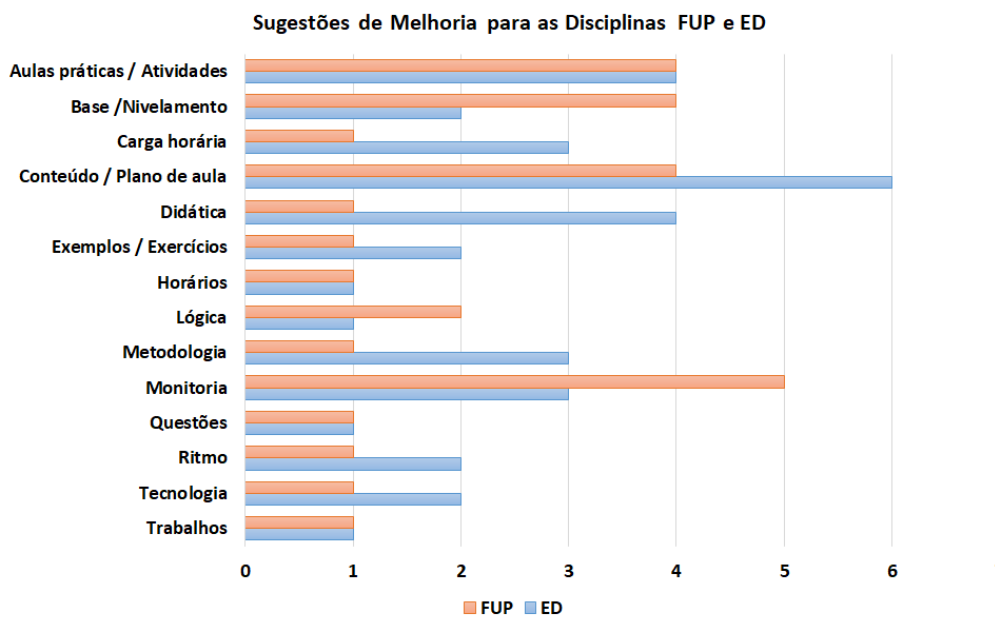


Figura 5 - Sugestões de melhoria para as disciplinas FUP e ED.

4. Discussão

Naturalmente se tende a apontar mais problemas do que pontos fortes, o que se comprovou pela quantidade de códigos e relações: 5 categorias para os pontos fortes, com 4 relações, e 7 categorias para os pontos fracos com 8 relações. Assim, nem todas as categorias estavam presentes ao mesmo tempo nos pontos fortes e fracos.

As tecnologias utilizadas em algumas disciplinas forçaram certas metodologias, apontado por alguns alunos. Isso pode ser considerado um problema se a disciplina possuir a característica de ser fundamental (conceitos básicos e iniciais), ou seja, conceitos que devem ser aplicados e atendidos para praticamente todas as tecnologias.

Do ponto de vista de metodologia e didática, percebe-se uma dificuldade conceitual na sua diferenciação por parte dos alunos. Por exemplo, na citação de P22 “Metodologia de ensino, às vezes, um pouco rápida”. Muitas vezes a metodologia é

confundida com didática. Percebe-se também a necessidade de adequação do ritmo de ensino, volume de conteúdo e atividades, para uma melhoria do aprendizado.

QC03 indicou que metade das respostas de dificuldade em FUP foi didática. Conforme QC04, metodologia e linguagem de programação são responsáveis por mais da metade das respostas de dificuldade em ED. Há uma relação entre didática e metodologia, que impacta diretamente no aprendizado dos alunos. Essas questões reforçam as relações de pontos fracos que envolvem didática e metodologia. Uma das sugestões apontadas pelos alunos, tanto para FUP quanto para ED, foi melhorar a metodologia e didática, também destacado pela relação identificada pelo *Grounded Theory*. Entretanto, como um ponto forte, a relação de associação entre Didática e Compreensão é um indício que os alunos detêm consciência do esforço dos professores nas disciplinas. O ideal é que muitos comentários fossem nessa linha, para fins de motivação. Isso também pode ser reforçado por QC03, que indicou quase a metade dos alunos não possuíram dificuldade em FUP.

QC05 informou a frequência de participação dos alunos nas monitorias. A resposta que 11 alunos nunca participaram e 13 participam apenas às vezes pode ser um caminho a ser investigado. As categorias tecnologia e atividades poderiam ter seus problemas minimizados se a atitude de participação das monitorias fosse maior. Suporte e tecnologia podem ser melhorados com mais contato dos alunos com monitores. Isso também refletiu pela pouca frequência apontada em QC05. Além disso, exemplos, aulas práticas e atividades foram apontadas como sugestões de melhorias. Porém, se os alunos tivessem mais contato com monitores, é possível que problemas fossem minimizados.

Por fim, QC02 indicou que 16 alunos reprovaram FUP. Esse número é alto, e infelizmente é comum ocorrer um alto número de reprovações em disciplinas iniciais de programação. Um aspecto que pode ser considerado para a redução desse número nesse contexto é avaliar a qualidade da didática e as tecnologias empregadas nas disciplinas, revelados pelas respostas do questionário.

5. Conclusões

O ensino de lógica de programação e codificação é uma atividade complexa e que requer muito esforço por parte dos docentes e discentes. Este trabalho buscou apresentar resultados de uma análise qualitativa por meio de *Grounded Theory* para identificar melhorias nas disciplinas de FUP e ED, e seus pontos fortes e fracos. Como grandes resultados, percebeu-se a necessidade de uma didática mais adequada para tais disciplinas, adequação do conteúdo ao tempo das aulas, e maior suporte tecnológico. Também notou-se que a necessidade por mais atividades ou práticas para proporcionar um melhor desempenho na disciplina. Todos esses itens apontados pela análise qualitativa também foram reforçados pelos alunos na questão relacionada às sugestões de melhorias.

Apesar da existência de muitos comentários negativos, algumas citações foram na direção oposta, como o comentário de P3 citando na questão de melhorias o seguinte: “*Nada a declarar, meu professor teve uma boa metodologia*”. Isso reforça a ideia que existem boas práticas e boa didática. Uma ação possível seria mapear tais ações e disseminar nas futuras turmas das disciplinas de FUP e ED.

Como trabalhos futuros, pretende-se avaliar a qualidade da didática nas disciplinas de FUP e ED, ampliar a análise qualitativa para outras disciplinas de programação, como Programação Orientada a Objetos, e incentivar a utilização das monitorias para apoio às disciplinas. Uma análise da metodologia aplicada pelos professores também pode ser uma boa fonte de pesquisa, podendo ser adotada uma metodologia mais genérica, mas que seja mais adequada ao ensino de programação.

Espera-se que este trabalho possa também colaborar com a comunidade de Educação em Computação a partir das experiências e perspectivas relatadas no contexto do ensino de programação, motivando trabalhos semelhantes de análise qualitativa com *Grounded Theory*, possibilitando assim uma evolução da área.

Agradecimentos

Este trabalho foi realizado com recursos do projeto NÚCLEO DE ANÁLISE QUALITATIVA PARA ESTUDOS DE COMPUTAÇÃO, EDITAL Nº 6/2020 - PREX, da Universidade Federal do Ceará (UFC).

Referências Bibliográficas

- BARROS, R. P.; JUNIOR, O. V. S.; SILVA, I. R. M.; SANTOS, L. F.; NETO, V. R. C. Predição do rendimento dos alunos em lógica de programação com base no desempenho das disciplinas do primeiro período do curso de ciências e tecnologia utilizando técnicas de mineração de dados. In: **VIII Congresso Brasileiro de Informática na Educação - XXX Simpósio Brasileiro de Informática na Educação (SBIE 2019)**. 2019.
- BISPO JR, E.; RAABE, A.; MATOS, E.; MASCHIO, E. BARBOSA, E.; CARVALHO, L.; BITTENCOURT, R.; DURAN, R.; e FALCÃO, T. Tecnologias na educação em computação: Primeiros referenciais. **Brasileira de Informática na Educação - RBIE**. v.28. 2020.
- CASTRO, F.; TEDESCO, F. Promovendo a Reflexão sobre o Erro em Disciplinas Introdutórias de Programação no Ensino Superior. **Revista Brasileira de Informática na Educação - RBIE**. v.28. 2020.
- CORBIN, J.; STRAUSS, A. **Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory**. Thousand Oaks: SAGE Publications, 4th edition. 2014.
- COUTINHO, E. F.; GOMES, G. A. M.; JOSE, M. L. A. Applying design thinking in disciplines of systems development. In: **8th Euro American Conference on Telematics and Information Systems (EATIS)**. 2016.
- COUTINHO, E. F.; LIMA, E. T.; SANTOS, C. C. Um panorama sobre o desempenho de uma disciplina inicial de programação em um curso de graduação. **Revista Tecnologias na Educação**. v.19, n.9. 2017.
- FERREIRA, A. C.; SANTOS, J.; SILVA, R.; OLIVEIRA, A. T. R.; ZABOT, D.; ABDALLA, D.; MATOS, E. Hello world: relato de experiência de um curso de iniciação à programação. In: **II Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg)**. 2016.
- FERREIRA, T.; VIANA, D.; FERNANDES, J.; SANTOS, R. Identifying emerging topics and difficulties in software engineering education in brazil. In: **XXXII Brazilian Symposium on Software Engineering, SBES '18**. 2018.
- GIL, A. C. Como elaborar projetos de pesquisa. 4. ed. São Paulo: Atlas, 2008.
- HOLANDA, W. D.; FREIRE, L. P.; COUTINHO, J. C. S. Estratégias de ensino-aprendizagem de programação introdutória no ensino superior: uma Revisão Sistemática da Literatura. **RENOTE - Revista Novas Tecnologias na Educação**. v.17, n.1. 2019.
- RICHTER, C.J.; BERNARDI, G.; CORDENONSI, A. Z. O Ensino de Programação Mediado por Tecnologias Educacionais: uma Revisão Sistemática de Literatura. **RENOTE - Revista Novas Tecnologias na Educação**. v.17, n.1. 2019.
- ZANIN, A.; SPARREMBERGER, A. S.; BARBOSA, J. L. V. Uma Proposta de Boas Práticas para o Ensino Colaborativo de Programação de Computadores. **RENOTE - Revista Novas Tecnologias na Educação**. v.17, n.3. 2019.