



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS DE RUSSAS**  
**CURSO DE ENGENHARIA DE SOFTWARE**

**HUDSON SILVA CUNHA**

**ANÁLISE DAS COMPETÊNCIAS TÉCNICAS DE ENGENHARIA DE SOFTWARE**  
**ENTRE O MEIO ACADÊMICO E A INDÚSTRIA DE SOFTWARE**

**RUSSAS**

**2023**

HUDSON SILVA CUNHA

ANÁLISE DAS COMPETÊNCIAS TÉCNICAS DE ENGENHARIA DE SOFTWARE ENTRE  
O MEIO ACADÊMICO E A INDÚSTRIA DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Engenharia de Software do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientador: Prof. Ms. José Osvaldo Mesquita Chaves

RUSSAS

2023

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- C978a Cunha, Hudson Silva.  
Análise das competências técnicas de engenharia de software entre o meio acadêmico e a indústria de software / Hudson Silva Cunha. – 2023.  
45 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia de Software, Russas, 2023.  
Orientação: Prof. Me. José Osvaldo Mesquita Chaves.
1. Engenharia de Software. 2. Software. 3. Competência. 4. Indústria. I. Título.

CDD 005.1

---

HUDSON SILVA CUNHA

ANÁLISE DAS COMPETÊNCIAS TÉCNICAS DE ENGENHARIA DE SOFTWARE ENTRE  
O MEIO ACADÊMICO E A INDÚSTRIA DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Engenharia de Software do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Aprovada em: 14 de Julho de 2023

BANCA EXAMINADORA

---

Prof. Ms. José Osvaldo Mesquita  
Chaves (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Alexandre Matos Arruda  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Anderson Feitoza Leitão Maia  
Universidade Federal do Ceará (UFC)

Dedico este trabalho aos meus pais João Maciel da Cunha (in memoriam) e Maria Alvaniza Silva do Carmo Cunha, que se dedicaram tanto em me dar educação.

## AGRADECIMENTOS

Em primeiro lugar, a Deus, que fez com que meus objetivos fossem alcançados, durante todos os meus anos de estudos.

Ao Prof. José Osvaldo Mesquita Chaves, pela orientação e confiança no processo de construção deste TCC. Seu apoio e ideias fizeram desta uma experiência edificante. Gostaria de agradecer também aos membros da banca examinadora pelas valiosas considerações.

À todos os professores, que com muita dedicação, passaram seu conhecimento ao longo dos anos como parte fundamental da minha trajetória de formação profissional.

À universidade, também essencial no meu processo de formação profissional, por tudo o que aprendi ao longo dos anos, e por oferecer um ambiente de respeito e oportunidades.

Aos colegas de curso que estiveram, direta ou indiretamente, me apoiando ao longo dessa jornada.

Por fim, mas não menos importante, agradeço à minha família. Às minhas irmãs Emanuelle, Evily e Letícia, que sempre me incentivaram e acreditaram que eu sou capaz. À minha mãe, Maria Alvaniza Silva do Carmo Cunha, que sempre cuidou de mim e torceu pelas minhas conquistas. E ao meu pai, João Maciel da Cunha (in memoriam), apaixonado por matemática, tinha como um de seus sonhos me ver formado. Sua reação de quando dei a notícia que eu estava aprovado para ingressar na faculdade está marcada em minha memória para sempre. Foi uma reação de, rapidamente levantar-se da cadeira, pegar na minha mão e, olhando nos meus olhos com uma expressão de quase chorar, me perguntar: “é mesmo, Hudson?”. Me viu iniciar o curso, mas infelizmente não está aqui para ver o desfecho dessa trajetória. Eu só queria poder dizer... pai, consegui! Sou muito grato aos meus pais, que desde sempre foram meus grandes incentivadores aos estudos e ensinaram-me através do exemplo os valores que tenho hoje.

“Onde há vontade, há um caminho.”

(Provérbio Inglês)

## RESUMO

Desenvolver software é uma tarefa complexa que, cada vez mais, tem exigido que os profissionais aprimorem seus conhecimentos com novas técnicas e ferramentas. A Engenharia de Software surgiu para fornecer aos profissionais o conhecimento necessário para desenvolver sistemas complexos dentro do prazo e com qualidade. Entretanto, existe uma deficiência no conhecimento aprendido nos cursos de Engenharia de Software das universidades. Este trabalho tem como objetivo investigar quais são as competências necessárias para um Engenheiro de Software atuar na indústria, relacioná-las com as disciplinas do curso e saber se há um desalinhamento do conhecimento fornecido pela universidade, para o que é demandado pela indústria. Efetuou-se um processo de coleta de dados com os participantes da pesquisa para analisar e extrair informações a respeito dos conhecimentos aprendidos na universidade, e saber se estão alinhados com as exigências das empresas em que atuam. Foram identificadas as disciplinas que mais estão alinhadas com a realidade da indústria e, também as que menos estão alinhadas, servindo como base para estudos futuros mais aprofundados acerca dos possíveis motivos da falta de concordância de algumas disciplinas com a indústria. Com esses dados, pôde-se concluir que há tanto disciplinas alinhadas, quanto desalinhadas com a realidade da indústria.

**Palavras-chave:** engenharia de software; software; competência; indústria.

## ABSTRACT

Developing software is a complex task that increasingly requires professionals to improve their knowledge with new techniques and tools. Software Engineering emerged to provide professionals with the necessary knowledge to develop complex systems on time and with quality. However, there is a deficiency in the knowledge learned in Software Engineering courses at universities. This work aims to investigate what are the necessary skills for a Software Engineer to work in the industry, relate them to the disciplines of the course and find out if there is a misalignment of the knowledge that is provided to what is demanded. A data collection process was carried out with the research participants to analyze and extract information about the knowledge learned at the university, and to know if they are aligned with the requirements of the companies in which they work. The main disciplines that are most aligned with the industry's reality were identified, as well as those that are least aligned, serving as a basis for further in-depth future studies on the possible reasons for the lack of agreement of some disciplines with the industry. With these data, it could be concluded that there are both disciplines aligned and misaligned with the reality of the industry.

**Keywords:** software engineering; software; competence; industry.

## LISTA DE FIGURAS

Figura 1 – Camadas da engenharia de software . . . . .	15
Figura 2 – Estrutura conceitual de competência. . . . .	16
Figura 3 – Processo metodológico. . . . .	23
Figura 4 – Parte do questionário enviado aos participantes da pesquisa. . . . .	27
Figura 5 – Faixa etária dos respondentes . . . . .	29
Figura 6 – Relação das disciplinas mais aproveitadas com os cargos exercidos pelos respondentes . . . . .	31
Figura 7 – Ocorrências das disciplinas mais difíceis consideradas pelos respondentes .	32
Figura 8 – Ocorrências das disciplinas optativas consideradas essenciais para exercer o cargo . . . . .	32
Figura 9 – Respostas da pergunta 13 do questionário . . . . .	36

## LISTA DE TABELAS

Tabela 1 – Áreas e Habilidades de Habilidades do Ciclo de Vida da Engenharia de Software	18
Tabela 2 – Áreas de Habilidades Transversais de Engenharia de Software . . . . .	19
Tabela 3 – Comparação com trabalhos relacionados . . . . .	21
Tabela 4 – <i>Strings</i> de busca e resultados . . . . .	24
Tabela 5 – Disciplinas do curso de Engenharia de Software . . . . .	26
Tabela 6 – Total de respondentes por sexo . . . . .	28
Tabela 7 – Faixa etária dos respondentes . . . . .	28
Tabela 8 – Cargos exercidos pelos respondentes categorizados por grupos . . . . .	29
Tabela 9 – Cargos exercidos pelos respondentes e sua relação com as disciplinas . . . . .	30
Tabela 10 – Principais disciplinas mais aproveitadas para desempenhar um papel de Desenvolvedor de Software . . . . .	31
Tabela 11 – Treinamento necessário relacionado com as disciplinas mais difíceis, segundo os respondentes da pesquisa . . . . .	33
Tabela 12 – Ocorrências das disciplinas obrigatórias que possuem uma abordagem dife- rente na prática . . . . .	34
Tabela 13 – Ocorrências das disciplinas obrigatórias que estão mais de acordo com a realidade . . . . .	35

## LISTA DE ABREVIATURAS E SIGLAS

ACM DL	ACM Digital Library
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNAE	Classificação Nacional de Atividades Econômicas
CNE	Conselho Nacional de Educação
CONCLA	Comissão Nacional de Classificação
DBLP	Digital Bibliography & Library Project
DCNs	Diretrizes Curriculares Nacionais
IBGE	Instituto Brasileiro de Geografia e Estatística
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
OTAN	Organização do Tratado do Atlântico Norte
SEEK	Software Engineering Education Knowledge
SWECOM	Software Engineering Competency Model
UFC	Universidade Federal do Ceará

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>13</b>
<b>1.1</b>	<b>Objetivo Geral e Específicos</b> . . . . .	<b>14</b>
<b>1.2</b>	<b>Estrutura do Trabalho</b> . . . . .	<b>14</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>15</b>
<b>2.1</b>	<b>Engenharia de Software</b> . . . . .	<b>15</b>
<b>2.2</b>	<b>Competência Técnica</b> . . . . .	<b>16</b>
<b>2.3</b>	<b>Competências Técnicas do Engenheiro de Software</b> . . . . .	<b>16</b>
<b>2.4</b>	<b>Indústria de Software</b> . . . . .	<b>20</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b> . . . . .	<b>21</b>
<b>4</b>	<b>METODOLOGIA</b> . . . . .	<b>23</b>
<b>4.1</b>	<b>Delimitação do Objeto de Pesquisa</b> . . . . .	<b>23</b>
<b>4.2</b>	<b>Levantamento Bibliográfico</b> . . . . .	<b>24</b>
<b>4.3</b>	<b>Coleta de Dados</b> . . . . .	<b>25</b>
<b>4.3.1</b>	<i>Elaborar Questionário</i> . . . . .	<b>25</b>
<b>4.3.2</b>	<i>Aplicar Questionário</i> . . . . .	<b>25</b>
<b>4.4</b>	<b>Análise dos Dados</b> . . . . .	<b>26</b>
<b>5</b>	<b>RESULTADOS</b> . . . . .	<b>28</b>
<b>5.1</b>	<b>Perguntas Obrigatórias</b> . . . . .	<b>28</b>
<b>5.2</b>	<b>Pergunta Opcional</b> . . . . .	<b>35</b>
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>37</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>39</b>
	<b>APÊNDICES</b> . . . . .	<b>41</b>
	<b>APÊNDICE A –QUESTIONÁRIO UTILIZADO PARA OBTER AS RES- POSTAS DA PESQUISA</b> . . . . .	<b>41</b>

## 1 INTRODUÇÃO

A engenharia de software exerce um papel fundamental no desenvolvimento de sistemas fornecendo conceitos e técnicas que ajudam a alcançar um software com qualidade e produtividade (HIRAMA, 2012). A engenharia de software é importante visto que capacita profissionais para o desenvolvimento de sistemas complexos dentro do prazo e com alta qualidade (PRESSMAN; MAXIM, 2016).

Engenharia de Software é uma disciplina que possui como essência a prática dos seus princípios. Porém, software pode ser desenvolvido sem o adequado uso desses princípios (GIMENES, 2015). Ainda que existam muitas pessoas desenvolvendo software sem o uso dos princípios de engenharia de software, a maior parte do desenvolvimento de software é uma atividade profissional em que o software é desenvolvido para fins comerciais. Isto significa que um software profissional é desenvolvido por times, e não por indivíduos. É neste cenário que a engenharia de software surge para apoiar aspectos de qualidade no desenvolvimento de software (SOMMERVILLE, 2015).

Alguns autores já apontaram a existência de um grande número de alunos e egressos do curso de Engenharia de Software com dificuldade de atender às necessidades das empresas. O conhecimento acadêmico dessas pessoas não está correspondendo a realidade da indústria de software (GAROUSI *et al.*, 2019). A existência da falta de alinhamento entre o que é aprendido na universidade e o que é aplicado na indústria de software, afeta diretamente a qualidade e a produtividade.

O presente trabalho tem como propósito identificar as competências técnicas fundamentais do engenheiro de software para a sua atuação profissional no mercado. Por meio de um levantamento de informações com recém-graduados e alunos do curso de Engenharia de Software que já exercem algum papel na área, será possível estabelecer uma relação entre as competências adquiridas na universidade e as competências exigidas pelo mercado de trabalho. Assim, espera-se obter informações que respondam se a engenharia de software na comunidade acadêmica corresponde às necessidades da indústria de software. Este é também um meio de saber como as empresas estão usando o conhecimento da engenharia de software.

## 1.1 Objetivo Geral e Específicos

O objetivo geral deste trabalho é verificar se as competências técnicas imprescindíveis para a engenharia de software, estudadas na universidade, estão alinhadas às necessidades da indústria de software.

São objetivos específicos:

- Fazer o levantamento de informações para identificação das necessidades da indústria;
- Abordar as disciplinas do curso de Engenharia de Software que são entendidas como competências técnicas aos olhos da indústria;
- Analisar a aderência das competências do curso de Engenharia de Software com as necessidades da indústria.

## 1.2 Estrutura do Trabalho

O restante deste trabalho está estruturado da seguinte forma:

- *Capítulo 2*: Aborda a fundamentação teórica com os principais conceitos tratados no trabalho.
- *Capítulo 3*: Discute os principais trabalhos relacionados acerca do tema.
- *Capítulo 4*: Explica a metodologia a ser usada para obter informações necessárias sobre as competências técnicas do estudante ou egresso de Engenharia de Software e, o que é requerido pela indústria.
- *Capítulo 5*: Apresenta e analisa os resultados da pesquisa.
- *Capítulo 6*: São apresentadas as considerações finais do trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica acerca do tema. Ele define os conceitos mais importantes para a compreensão deste trabalho.

### 2.1 Engenharia de Software

O conceito de engenharia de software foi sugerido pela primeira vez em 1968, em uma conferência realizada pela Organização do Tratado do Atlântico Norte (OTAN) para discutir o que alguns participantes denominaram de *crise do software*. (NAUR; RANDELL, 1969).

O Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE) define a engenharia de software como “(1) A aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de software; ou seja, a aplicação da engenharia ao software. (2) O estudo de abordagens como definido em (1)” (IEEE, 1990, p. 67, tradução nossa).

Sommerville (2015) dá uma definição mais ampla para Engenharia de Software dizendo que se trata de uma disciplina de engenharia com ênfase nos aspectos da produção do software, desde sua idealização até sua operação e manutenção.

Nas afirmações de Pressman e Maxim (2016), a engenharia de software incorpora processo, métodos e ferramentas que propiciam aos profissionais desenvolverem software dentro do prazo e com qualidade. Afirmam ainda, que qualquer abordagem de engenharia deve estar fundamentada na qualidade. A Figura 1 ilustra essa fundamentação.

De acordo com Laplante (2017), modelagem e otimização são as atividades que definem a engenharia de software. A modelagem é uma atividade de tradução que transforma o conceito do produto de software em uma especificação de requisitos, que por sua vez, é convertido em um *design* e, por conseguinte, transformado em código. Já a otimização trata de eficiência e clareza.

Figura 1 – Camadas da engenharia de software



Fonte: Pressman e Maxim (2016, p. 16).

## 2.2 Competência Técnica

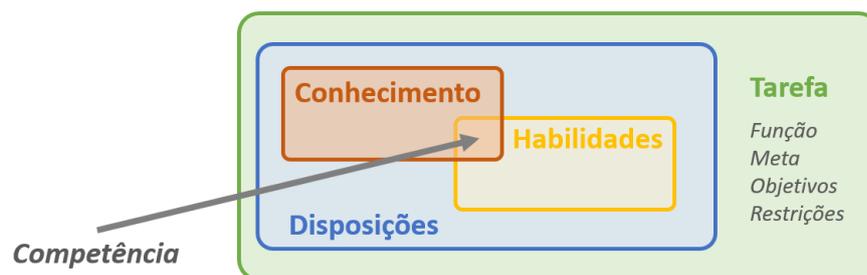
O dicionário Michaelis (2015) define *competência* como: “Aptidão que um indivíduo tem de opinar sobre um assunto e sobre o qual é versado; conjunto de conhecimentos; indivíduo com profundo conhecimento de determinado assunto”. O dicionário Merriam-Webster (2016, tradução nossa) define, na língua inglesa, *competência* como: “A qualidade ou estado de ter conhecimento, julgamento, habilidade ou força suficiente (quanto a um dever específico ou em um aspecto específico)”.

É importante destacar que as competências podem ser tratadas por duas perspectivas: do indivíduo e da organização (BOUILLON, 1996). A partir dessa classificação, pode-se instanciar – na perspectiva do indivíduo – competências técnicas, frequentemente chamadas pelo seu termo em inglês, *hard skills* e, – na perspectiva da organização – habilidades comportamentais, comumente chamadas, em inglês, de *soft skills* (PIETERSE; EEKELEN, 2016). Em uma visão ampla, habilidades comportamentais são os traços pessoais que se refletem em hábitos no local de trabalho. Já Zarifian (1999 *apud* FLEURY; FLEURY, 2001), definem competência técnica nas organizações como um conhecimento próprio do trabalho a ser realizado.

Task Force (2020) especifica que a definição de competência é composta pelas dimensões *conhecimento*, *habilidades* e *disposições*. Essa estrutura é ilustrada na Figura 2.

Competência = [Conhecimento + Habilidade + Disposições] em uma Tarefa

Figura 2 – Estrutura conceitual de competência.



Fonte: Adaptada de Task Force (2020, p. 125).

## 2.3 Competências Técnicas do Engenheiro de Software

Ao analisar a literatura, percebe-se que alguns autores abordam genericamente em suas obras as disciplinas da engenharia de software e as competências que o engenheiro deve possuir. A exemplo disso, Pieterse e Eekelen (2016), dizem que além das inúmeras competências

técnicas que um engenheiro de software deve possuir, deve-se dar atenção à algumas destas: *solucionar problemas, gerenciar configurações e usar ferramentas de desenvolvimento*.

Acerca das competências da engenharia de software, existe um certo equívoco quanto a definição do que é responsabilidade do engenheiro de software, é o que diz Wazlawick (2013). Distingui-se as atividades dos desenvolvedores e do engenheiro de software dizendo que atividades como levantamento de requisitos, modelagem, *design* e codificação, tipicamente não são de um engenheiro de software, embora ele seja capacitado a realizá-las. O desenvolvedor é um executor e é responsável por descobrir os requisitos do sistema e transformá-los em um produto executável, enquanto que o engenheiro de software não trabalha nas atividades de análise e produção de código. O engenheiro de software, por tanto, viabiliza e controla o processo, fornece e mantém as ferramentas mais adequadas a cada projeto (WAZLAWICK, 2013).

O Conselho Nacional de Educação (CNE), – por meio da Resolução nº 5, de 16 de novembro de 2016 – instituiu as Diretrizes Curriculares Nacionais (DCNs) para os cursos de graduação na área da Computação, entre eles, Engenharia de Software. Dessa Resolução, espera-se dos engenheiros de software que: “Possuam sólida formação em Ciência da Computação, Matemática e Produção, visando a criação de sistemas de software de alta qualidade de maneira sistemática, controlada, eficaz e eficiente que levem em consideração questões éticas, sociais, legais e econômicas” (BRASIL, 2016, Art. 4º. § 3º. I).

No que diz respeito as competências técnicas do engenheiro de software, pode-se destacar da Resolução: avaliar a qualidade de sistemas de software; integrar sistemas de software; gerenciar projetos de software; qualificar e quantificar seu trabalho; conceber, aplicar e validar princípios, padrões e boas práticas no desenvolvimento de software; identificar e analisar problemas avaliando as necessidades dos clientes, especificar os requisitos de software, projetar, desenvolver, implementar, verificar e documentar soluções de software baseadas no conhecimento apropriado de teorias, modelos e técnicas (BRASIL, 2016, Art. 4º. § 3º. V, VI, VII, IX, XI e XIV).

É definido por Task Force (2020) que, para Engenharia de Software, o engenheiro precisa ter competência nos seguintes aspectos: Requisitos de Software; Design de Software; Construção de Software; Teste de Software; Sustentação de Software; Processo de Software e Ciclo de Vida; Engenharia de Sistemas de Software; Qualidade de Software; Segurança de Software (*Software Security*); Proteção de Software (*Software Safety*); Gerenciamento de Configuração de Software; Medição de Software; Interação Humano-Computador; Gerenciamento de

Projetos; Atributos Comportamentais.

A IEEE (2014) fornece o Software Engineering Competency Model (SWECOM), que define o entendimento sobre as competências técnicas de um engenheiro de software. Esse modelo faz uma divisão das competências em dois grupos, que são as áreas de habilidades do ciclo de vida e áreas de habilidades transversais. Uma área de habilidade do ciclo de vida inclui as habilidades necessárias para executar atividades dentro de uma fase de desenvolvimento ou manutenção de software - por exemplo, engenharia de requisitos de software. Uma área de habilidade transversal, por sua vez, se aplica à todas as áreas de habilidade do ciclo de vida - por exemplo, garantia de qualidade.

As áreas de habilidades do ciclo de vida e habilidades transversais estão listadas nas Tabelas 1 e 2, respectivamente.

Tabela 1 – Áreas e Habilidades de Habilidades do Ciclo de Vida da Engenharia de Software

<b>Áreas de Habilidades do Ciclo de Vida</b>	<b>Habilidades</b>
<b>Habilidades de Requisitos de Software</b>	<ul style="list-style-type: none"> <li>• Elicitação de Requisitos de Software</li> <li>• Análise de Requisitos de Software</li> <li>• Especificação de Requisitos de Software</li> <li>• Verificação e Validação de Requisitos de Software</li> <li>• Processo e Gestão de Produtos de Requisitos de Software</li> </ul>
<b>Habilidades de Design de Software</b>	<ul style="list-style-type: none"> <li>• Fundamentos de Design de Software</li> <li>• Estratégias e Métodos de Design de Software</li> <li>• Projeto de Arquitetura de Software</li> <li>• Análise e Avaliação da Qualidade do Design de Software</li> </ul>
<b>Habilidades de Construção de Software</b>	<ul style="list-style-type: none"> <li>• Planejamento de Construção de Software</li> <li>• Gerenciando a Construção de Software</li> <li>• Projeto Detalhado e Codificação</li> <li>• Depuração e Teste</li> <li>• Integrando e Colaborando</li> </ul>
<b>Habilidades de Teste de Software</b>	<ul style="list-style-type: none"> <li>• Planejamento de Teste de Software</li> <li>• Infraestrutura de Teste de Software</li> <li>• Técnicas de Teste de Software</li> <li>• Medição de Teste de Software e Rastreamento de Defeitos</li> </ul>
<b>Habilidades de Sustentação de Software</b>	<ul style="list-style-type: none"> <li>• Transição de software</li> <li>• Suporte de Software</li> <li>• Manutenção de Software</li> </ul>

Fonte: IEEE (2014).

Tabela 2 – Áreas de Habilidades Transversais de Engenharia de Software

Áreas de Habilidades Transversais	Habilidades
<b>Processo e Vida Útil do Software</b>	<ul style="list-style-type: none"> <li>• Implementação do Ciclo de Vida do Desenvolvimento de Software</li> <li>• Definição e Adaptação de Processos</li> <li>• Implementação e Gestão de Processos</li> <li>• Avaliação e Melhoria de Processos</li> </ul>
<b>Engenharia de Sistemas de Software</b>	<ul style="list-style-type: none"> <li>• Modelagem do Ciclo de Vida de Desenvolvimento de Sistemas</li> <li>• Definição de Conceito</li> <li>• Engenharia de Requisitos de Sistema</li> <li>• Projeto de Sistema</li> <li>• Alocação de Requisitos</li> <li>• Engenharia de Componentes</li> <li>• Integração e Verificação de Sistemas</li> <li>• Validação e Implantação de Sistema</li> <li>• Planejamento de Sustentação do Sistema</li> </ul>
<b>Habilidades de Qualidade de Software</b>	<ul style="list-style-type: none"> <li>• Gerenciamento de Qualidade de Software</li> <li>• Revisões (revisão, passo a passo, inspeção)</li> <li>• Auditorias (concentram-se tanto no produto quanto no processo, mas são feitas por uma organização interna ou externa independente)</li> <li>• Controle Estatístico</li> </ul>
<b>Habilidades de Segurança de Software</b>	<ul style="list-style-type: none"> <li>• Requisitos</li> <li>• Projeto</li> <li>• Construção</li> <li>• Teste</li> <li>• Processo</li> <li>• Qualidade</li> </ul>
<b>Habilidades de Proteção de Software</b>	<ul style="list-style-type: none"> <li>• Requisitos</li> <li>• Projeto</li> <li>• Construção</li> <li>• Teste</li> <li>• Processo</li> <li>• Qualidade</li> </ul>
<b>Habilidades de Gerenciamento de Configuração de Software</b>	<ul style="list-style-type: none"> <li>• Planejar</li> <li>• Conduzir</li> <li>• Gerenciar Versões de Software</li> </ul>
<b>Habilidades de Medição de Software</b>	<ul style="list-style-type: none"> <li>• Planejar o Processo de Medição</li> <li>• Executar o Processo de Medição</li> </ul>
<b>Habilidades de Interação Humano-Computador</b>	<ul style="list-style-type: none"> <li>• Requisitos</li> <li>• Design de Estilo de Interação</li> <li>• Design Visual</li> <li>• Teste e Avaliação de Usabilidade</li> <li>• Acessibilidade</li> </ul>

Fonte: IEEE (2014).

## 2.4 Indústria de Software

Uma das definições de indústria é: “Um grupo de empresas que fornecem um determinado produto ou serviço” (MERRIAM-WEBSTER, 2016). Por tanto, o software sendo um produto, pode-se afirmar que o desenvolvimento de software é feito por um tipo de indústria: a indústria de software.

A maioria das pessoas têm um entendimento mais claro sobre outras indústrias e seus produtos, sejam elas químicas, de materiais de construção ou de alimentos. Essas pessoas têm ciência da indústria de software, no entanto, dificilmente entendem a classificação de indústria de software. Isso pode ocorrer devido a natureza intangível do software (KELLY, 2003).

O Instituto Brasileiro de Geografia e Estatística (IBGE), por meio da Comissão Nacional de Classificação (CONCLA), criou a Classificação Nacional de Atividades Econômicas (CNAE) para estruturar e detalhar os tipos de atividades econômicas do país, onde fica definido que a indústria de software enquadra-se no grupo “atividades dos serviços de tecnologia da informação” (CONCLA, 2007).

### 3 TRABALHOS RELACIONADOS

Muitos estudos já investigaram o problema entre o ensino acadêmico dos desenvolvedores de software e as expectativas da indústria. Em uma revisão sistemática da literatura foi possível identificar as diferentes formas de abordagem desses estudos. Lethbridge (2000) concluiu que na prática a matemática tem pouca relevância para engenheiros de software. Já Liebenberg *et al.* (2014), observaram que os graduados geralmente não dispõem de conhecimento de negócios e experiência prática em projetos reais, concluindo que as universidades precisam incluir experiências reais no ensino e se manterem no mesmo ritmo das mudanças na tecnologia.

Os resultados apresentados por Garousi *et al.* (2019) ressaltam a importância dos *requisitos, testes e exemplos do mundo real* para o ensino de Engenharia de Software. Os autores, puderam ainda, identificar problemas de conhecimento em áreas específicas, como *gerenciamento de configuração, modelos e métodos*, bem como *processos*. Já Garousi *et al.* (2020), observaram que muitos estudos universitários de Engenharia de Software evoluíram de estudos de Ciência da Computação e, por conseguinte, possuem um foco em tópicos teóricos e técnicos de Ciência da Computação, bem como fundamentos matemáticos. Nisso, puderam identificar as competências indispensáveis na indústria e reunir indícios de insuficiência de conhecimento em estudantes graduados em Engenharia de Software.

Liargkovas *et al.* (2021) investigaram as necessidades industriais dos desenvolvedores de software e avaliou a satisfação destes pelo Software Engineering Education Knowledge (SEEK), além de sugerir melhorias.

As principais características entre alguns dos trabalhos analisados e este, são vistas na Tabela 3.

Tabela 3 – Comparação com trabalhos relacionados

Característica	Liebenberg <i>et al.</i> (2014)	Garousi <i>et al.</i> (2020)	Liargkovas <i>et al.</i> (2021)	Este trabalho
Usa um guia reconhecido		✓	✓	✓
Definição clara de competência		✓		✓
Visão do aluno				✓
Visão do recém-formado	✓			✓
Visão da indústria	✓	✓	✓	✓
Visão de outros trabalhos		✓	✓	✓

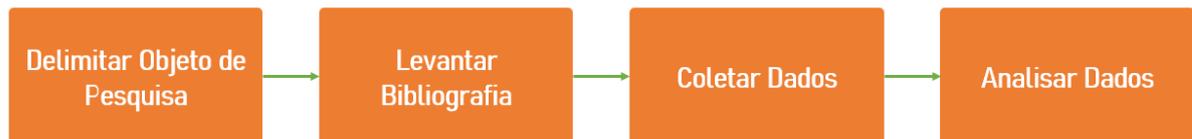
Fonte: elaborado pelo autor (2022).

Parte dos trabalhos analisados tratam as habilidades e competências do engenheiro de software de maneira mais genérica, no sentido de abordar tais características sem distinção. Por vezes, outros trabalhos tratam a Engenharia de Software como um conceito de princípios pertencentes aos Desenvolvedores de Software, levando a crer que todo Desenvolvedor de Software é um Engenheiro de Software. Assim, este trabalho busca fazer as diferenciações necessárias para se obter uma maior precisão no que tange aos princípios da Engenharia de Software.

## 4 METODOLOGIA

O método aplicado teve como objetivo relacionar as competências adquiridas na universidade com as que são requeridas pela indústria, portanto, foi feita uma pesquisa objetiva de natureza descritiva que utiliza um estudo de levantamento. Desta forma, o processo metodológico foi dividido nas seguintes etapas: **Delimitação do Objeto de Pesquisa; Levantamento Bibliográfico; Coleta de Dados; Análise dos Dados.**

Figura 3 – Processo metodológico.



Fonte: o autor (2022).

### 4.1 Delimitação do Objeto de Pesquisa

Uma forma de atingir o objetivo deste trabalho, é delimitando a pesquisa e, para isso, foi estabelecido não só o que é o objeto de pesquisa, mas também o que não está incluso como parte necessária da pesquisa e, desta forma, foi definido o escopo da pesquisa.

Como já afirmaram Botelho e Cruz (2013), para delimitar o objeto de pesquisa, deve-se responder às perguntas: **onde** e **quando**. No entanto, cabe ainda acrescentar às perguntas: **o que** e **como**. Desta forma é preciso então responder as seguintes perguntas:

- (a) *O que será pesquisado?*
- (b) *Onde será pesquisado?*
- (c) *Quando será pesquisado?*
- (d) *Como será pesquisado?*

Isto posto, a pesquisa delimitou-se em colher informações sobre o alinhamento das competências técnicas fundamentais para atuar na área de Engenheiro de Software e saber se estão de acordo com o que é requisitado na indústria de software, tendo como referência a Universidade Federal do Ceará (UFC) - Campus Russas, sendo aplicado com alunos e egressos do curso de Engenharia de Software, no ano de 2023, por meio de um questionário.

## 4.2 Levantamento Bibliográfico

Nas buscas por artigos foram utilizados alguns motores de busca, tais como, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES); Digital Bibliography & Library Project (DBLP); ACM Digital Library (ACM DL); ScienceDirect - Elsevier; IEEE. Além do que foi retornado das buscas, outros trabalhos foram encontrados por meio da verificação de referências. A Tabela 4 detalha o registro das buscas.

Tabela 4 – *Strings* de busca e resultados

Base de Dados	<i>String</i> de Busca	Resultados
ACM DL	"software engineering"AND "software industry"AND "gap"	557
ACM DL	"software engineering"AND academy AND industry NOT game* divers*	3205
ACM DL	"software engineering"AND academy AND gap AND industry NOT game* divers*	849
ACM DL	"software engineering"AND competence AND academy AND industry NOT game* divers* empirical	308
CAPES	"engenharia de software"AND indústria AND necessidades	3
DBLP	"software engineering"AND industry AND gap AND education	5
DBLP	"software engineering"AND gap AND education	12
IEEE	"software engineering"AND "software industry"AND "gap"	36
IEEE	"software engineering"AND competence AND academy AND industry	2
IEEE	"software engineering"AND university academy AND "software industry"	4
ScienceDirect	"software engineering"AND academy AND industry	1586
ScienceDirect	"software engineering"AND competence AND academy AND industry	205
ScienceDirect	"software engineering"AND competence skill AND academy AND industry	147

Fonte: o autor (2022).

As buscas feitas nas bases de dados, listadas na Tabela 4, retornaram quantidades variadas de resultados entre as bases. Foi preciso filtrar os trabalhos mais relevantes relacionados com a proposta do presente trabalho, portanto, alguns critérios foram levados em consideração na exclusão de resultados:

- (a) *Bases de dados que retornaram grande quantidade de resultados em relação a outras bases, observar no título se o artigo possui proposta semelhante a do presente trabalho;*
- (b) *Os trabalhos com data inferior a 2010 devem ser excluídos.*

Durante esse processo de exclusão foi identificado que as *strings* que retornaram uma quantidade elevada de artigos em relação a outras bases de dados tem relação com os parâmetros usados, já que são palavras utilizadas em trabalhos que possuem uma abordagem temática mais ampla.

### 4.3 Coleta de Dados

Para facilitar a compreensão, esta seção está dividida em duas subseções mais específicas.

#### 4.3.1 *Elaborar Questionário*

Lakatos e Marconi (2017) tratam o *questionário* como um instrumento de pesquisa que não requer a presença do pesquisador, e salientam que, antes da distribuição do questionário, é preciso saber com antecedência qual a informação se está buscando.

Assim, pesquisar técnicas de criação de questionários fez parte deste processo de elaboração, com o objetivo de gerar um questionário eficaz de maneira que aumentasse as chances de retorno dos participantes. Para tanto, precisou-se saber quais as disciplinas obrigatórias do curso de Engenharia de Software ofertadas na UFC - Campus Russas. Assim, para obter tal informação, a grade curricular do curso foi consultada no sistema da universidade. As disciplinas identificadas na consulta constam na Tabela 5.

A elaboração do questionário foi feita de modo a obter as informações necessárias para atender ao objetivo geral da pesquisa, sendo essas perguntas dos tipos objetivas e subjetivas. Para ilustrar algumas perguntas do questionário<sup>1</sup> enviado aos participantes, a Figura 4 mostra as perguntas 3 à 8. As perguntas 1 e 2 se referem à saber idade e sexo do participante. As perguntas 9 à 12 se assemelham a pergunta 8, onde existe um conjunto de opções oferecidas para que o respondente do questionário possa marcar. Por fim, a pergunta 13 é do tipo aberta para ser colocada as percepções acerca do tema da pesquisa.

#### 4.3.2 *Aplicar Questionário*

O questionário elaborado foi enviado por e-mail aos participantes, e conforme já dito por Lakatos e Marconi (2017), se fez pertinente enviar junto ao e-mail um texto explicando a natureza da pesquisa e a importância das respostas.

Para a aplicação do questionário, foi obtida, junto à secretaria do curso, uma listagem dos contatos de todos os egressos do curso de Engenharia de Software da UFC - Campus Russas. A lista continha um total de 95 contatos de e-mail para os quais o questionário foi enviado. Mais 15 contatos de alunos ainda ativos no curso foram adicionados à lista de contatos, e assim,

<sup>1</sup> O questionário completo encontra-se no apêndice A.

Tabela 5 – Disciplinas do curso de Engenharia de Software

<b>Código</b>	<b>Disciplina</b>	<b>Tipo</b>
RUS0077	Álgebra Linear	Optativa
RUS0300	Algoritmos em Grafos	Obrigatória
RUS0058	Análise e Projeto de Sistemas	Obrigatória
RUS0005	Arquitetura de Computadores	Obrigatória
RUS0242	Arquitetura de Software	Obrigatória
RUS0079	Compiladores	Optativa
RUS0083	Desenvolvimento de Software para Web	Optativa
RUS0060	Empreendedorismo	Obrigatória
RUS0014	Estrutura de Dados	Obrigatória
RUS0296	Ética Profissional	Obrigatória
RUS0031	Fundamentos da Economia e da Administração	Optativa
RUS0061	Fundamentos de Banco de Dados	Obrigatória
RUS0297	Fundamentos de Programação	Obrigatória
RUS0066	Gerência de Projetos de Software	Obrigatória
RUS0090	Inglês Instrumental I	Optativa
RUS0086	Inteligência Artificial	Optativa
RUS0256	Interação Humano-Computador	Obrigatória
RUS0298	Introdução à Engenharia de Software	Obrigatória
RUS0009	Introdução à Processos e Requisitos de Software	Obrigatória
RUS0299	Laboratório de Programação	Obrigatória
RUS0059	Linguagens de Programação	Obrigatória
RUS0072	Linguagens Formais e Autômatos	Optativa
RUS0062	Lógica para Computação	Obrigatória
RUS0099	Manutenção de Software	Obrigatória
RUS0004	Matemática Básica	Obrigatória
RUS0081	Matemática Computacional	Optativa
RUS0007	Matemática Discreta	Obrigatória
RUS0012	Pré-Cálculo	Optativa
RUS0008	Probabilidade e Estatística	Obrigatória
RUS0068	Processos de Software	Obrigatória
RUS0006	Programação Orientada a Objetos	Obrigatória
RUS0064	Projeto Detalhado de Software	Obrigatória
RUS0067	Projeto e Análise de Algoritmos	Obrigatória
RUS0096	Qualidade de Software	Obrigatória
RUS0082	Redes de Computadores	Obrigatória
RUS0069	Requisitos de Software	Obrigatória
RUS0085	Sistemas Distribuídos	Optativa
RUS0013	Sistemas Operacionais	Obrigatória
RUS0087	Teoria da Computação	Optativa
RUS0098	Verificação e Validação	Obrigatória

Fonte: SIGAA - UFC (2023).

totalizou-se 110 participantes para a pesquisa.

#### 4.4 Análise dos Dados

Evitar análises de dados desnecessárias implica planejar minuciosamente a análise antes da coleta dos dados (GIL, 2002). Após a coleta, foi feita a organização e classificação dos dados, depois, foram analisados e interpretados de modo a ter um correto entendimento das informações.

Figura 4 – Parte do questionário enviado aos participantes da pesquisa.

**3. Qual cargo você ocupa na empresa em que trabalha? \***

**4. Você já fez todas as disciplinas obrigatórias do curso de Engenharia de Software? \***

Sim  Não

**5. Precizou de treinamento, além da formação acadêmica, para desempenhar as funções de seu cargo? \***

Sim  Não

**6. Se a resposta anterior foi "Sim", informe qual o treinamento.**

**7. Qual das disciplinas obrigatórias do curso de Engenharia de Software você julga ser a mais difícil? \***

**8. De todas as disciplinas obrigatórias do curso, quais foram mais aproveitadas para executar seu cargo? \***

<input type="checkbox"/> Algoritmos em Grafos	<input type="checkbox"/> Análise e Projeto de Sistemas	<input type="checkbox"/> Arquitetura de Computadores
<input type="checkbox"/> Arquitetura de Software	<input type="checkbox"/> Empreendedorismo	<input type="checkbox"/> Estrutura de Dados
<input type="checkbox"/> Ética Profissional	<input type="checkbox"/> Fundamentos de Banco de Dados	<input type="checkbox"/> Fundamentos de Programação
<input type="checkbox"/> Gerência de Projetos de Software	<input type="checkbox"/> Interface Humano-Computador	<input type="checkbox"/> Introdução a Processos e Requisitos de Software
<input type="checkbox"/> Introdução à Computação e Engenharia de Software	<input type="checkbox"/> Laboratório de Programação	<input type="checkbox"/> Linguagens de Programação
<input type="checkbox"/> Lógica para Computação	<input type="checkbox"/> Manutenção de Software	<input type="checkbox"/> Matemática Básica
<input type="checkbox"/> Matemática Discreta	<input type="checkbox"/> Probabilidade e Estatística	<input type="checkbox"/> Processos de Software
<input type="checkbox"/> Programação Orientada a Objetos	<input type="checkbox"/> Projeto Detalhado de Software	<input type="checkbox"/> Projeto e Análise de Algoritmos
<input type="checkbox"/> Qualidade de Software	<input type="checkbox"/> Redes de Computadores	<input type="checkbox"/> Redes e Sistemas Distribuídos
<input type="checkbox"/> Requisitos de Software	<input type="checkbox"/> Sistemas Operacionais	<input type="checkbox"/> Verificação e Validação

Você pode marcar mais de um item.

Fonte: elaborado pelo autor (2023).

## 5 RESULTADOS

Antes de apresentar os resultados, vale observar que o perfil dos respondentes são em sua maioria, quando não em sua totalidade, de recém-formados que iniciaram suas carreiras com um perfil mais direcionado ao desenvolvimento de software, e as respostas fornecidas no questionário são um reflexo desse tipo de perfil.

Por estarem em início de carreira, recém-formados, na maioria das vezes não possuem ainda a percepção da importância que algumas disciplinas têm para a sua profissão, passando a percebê-las como tais, posteriormente com o amadurecimento profissional. Por isso, é importante ter em mente que a percepção dos recém-formados pode não refletir com exatidão as expectativas da indústria. Por tanto, suas escolhas para as respostas, tais como disciplinas essenciais, podem levar em conta afinidade ou relação da disciplina com o seu trabalho.

### 5.1 Perguntas Obrigatórias

Os resultados obtidos serão mostrados abordando inicialmente as principais perguntas do questionário, de modo a entender os grandes grupos de perguntas. Inicialmente, pode-se observar que, dos 110 participantes da pesquisa, apenas 23 responderam o questionário. Este número representa 21% do total, como mostra a Tabela 6.

Tabela 6 – Total de respondentes por sexo

<b>Público</b>	<b>Quantidade</b>	<b>Percentual</b>
Masculino	19	83%
Feminino	4	17%
<b>Total</b>	23	

Fonte: o autor (2023).

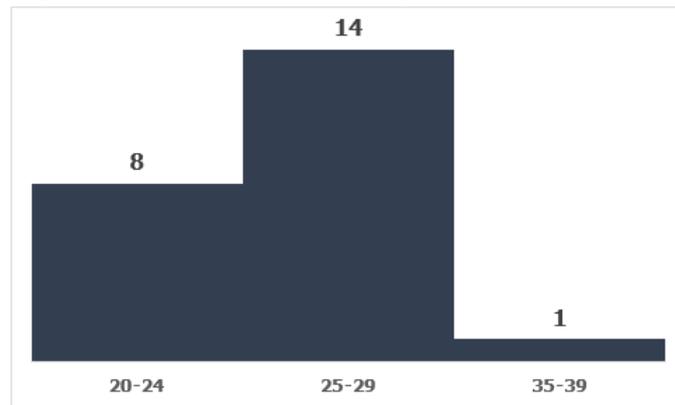
Para se ter uma noção da faixa etária dos alunos, a Tabela 7 mostra que mais de 60% dos respondentes se encontram na faixa etária de 25 à 29 anos.

Tabela 7 – Faixa etária dos respondentes

<b>Faixa etária</b>	<b>Quantidade</b>	<b>Percentual</b>
20-24	8	34,78%
25-29	14	60,87%
35-39	1	4,35%
<b>Total</b>	23	

Fonte: o autor (2023).

Figura 5 – Faixa etária dos respondentes



Fonte: o autor (2023).

Para se fazer uma relação dos cargos que os respondentes ocupam na empresa em que trabalham, foi feita uma lista dos cargos de cada um, conforme suas respostas. Em seguida, esses cargos foram categorizados por grupos maiores. A partir desses grupos foi criada a Tabela 9, aonde é apresentada a relação direta de cada grupo de função com as disciplinas do curso. É importante relatar também que dos 23 respondentes do questionário, 21 já fizeram todas as disciplinas do curso.

Tabela 8 – Cargos exercidos pelos respondentes categorizados por grupos

<b>Respondente</b>	<b>Função</b>	<b>Grupo</b>
1	Analista de Negócios	Negócio
2	Analista de Projetos	Projeto
3	Analista de Qualidade	Qualidade
4	Analista de Qualidade	Qualidade
5	Analista de Sistemas	Desenvolvimento
6	Analista de Testes	Teste
7	Analista e Desenvolvedor de Sistemas	Desenvolvimento
8	Arquiteto de Software	Arquitetura
9	Desenvolvedor	Desenvolvimento
10	Desenvolvedor	Desenvolvimento
11	Desenvolvedor Back-end	Desenvolvimento
12	Desenvolvedor de Software	Desenvolvimento
13	Desenvolvedor FrontEnd	Desenvolvimento
14	Desenvolvedor Mobile	Desenvolvimento
15	Desenvolvedor Pleno	Desenvolvimento
16	Desenvolvedor Trainee	Desenvolvimento
17	Engenheiro de Testes	Teste
18	Gerente de Projetos	Projeto
19	Product Owner	Product Owner
20	Product Owner	Product Owner
21	Quality Assurance Jr.	Qualidade
22	Senior Software Engineer	Engenharia de Software
23	Software Engineer	Engenharia de Software

Fonte: o autor (2023).

Tabela 9 – Cargos exercidos pelos respondentes e sua relação com as disciplinas

Grupo	Relação direta com a disciplina	Ocorrências
Arquitetura	Arquitetura de Software	1
Desenvolvimento	Fundamentos de Programação	10
Engenharia de Software	<i>Todo o curso</i>	2
Negócio	Requisitos de Software	1
Product Owner	Requisitos de Software	2
Gerência de Projeto	Gerência de Projetos de Software	2
Qualidade	Qualidade de Software	3
Teste	Verificação e Validação	2

Fonte: o autor (2023).

Estes dados mostram que as principais disciplinas, que são próprias da Engenharia de Software, tem uma tendência a serem as menos seguidas por aqueles que fazem o curso. Mas vale destacar que tal informação não sugere o motivo dessa tendência, ou seja, não se pode concluir que essas disciplinas são pouco enfatizadas durante o curso apenas com essa informação. Uma vez que se pode haver razões diversas, específicas e particulares que levaram cada indivíduo a escolher determinada função, é preciso uma investigação mais profunda acerca deste assunto.

A pergunta 8 do questionário é “*De todas as disciplinas obrigatórias do curso, quais foram mais aproveitadas para executar seu cargo*”? As respostas para essa pergunta foram distribuídas em uma matriz, mostrada na Figura 6, que relaciona os grupos de cargos com as disciplinas obrigatórias do curso de Engenharia de Software. Dos 8 grupos já levantados na Tabela 8, *Desenvolvimento* corresponde a 43% das áreas de atuação dos respondentes. Destaca-se a importância de algumas disciplinas para se exercer funções da área de Desenvolvimento de Software.

Das disciplinas que se destacam para *Desenvolvimento*, lista-se as 10 mais levantadas no questionário, como mostra a Tabela 10. A partir dessa tabela, destacam-se as disciplinas **Análise e Projeto de Sistemas, Requisitos de Software, Interface Humano-Computador e Introdução a Processos e Requisitos de Software**, que mesmo não sendo disciplinas que abordam a implementação de códigos, se mostraram importantes para desempenhar a função de Desenvolvedor.

Na pergunta 5, “*Precisou de treinamento, além da formação acadêmica, para desempenhar as funções de seu cargo*?”, foram obtidas 21 respostas “Sim”, ou seja, esse é o total de pessoas que informaram na pergunta 6 qual o treinamento feito, necessário para desempenhar suas funções.

Figura 6 – Relação das disciplinas mais aproveitadas com os cargos exercidos pelos respondentes

	Arquitetura	Desenvolvimento	Engenharia de Software	Negócio	Product Owner	Gerência de Projeto	Qualidade	Teste	Totais
Algoritmos em Grafos	0	0	0	0	0	0	0	0	0
Análise e Projeto de Sistemas	0	7	1	0	1	2	3	2	16
Arquitetura de Computadores	1	0	0	0	0	1	1	0	3
Arquitetura de Software	1	5	2	0	0	2	2	1	13
Empreendedorismo	0	0	1	0	0	2	0	0	3
Estrutura de Dados	0	8	0	0	0	1	1	1	11
Ética Profissional	0	5	0	0	1	1	0	1	8
Fundamentos de Banco de Dados	1	7	1	0	1	1	2	2	15
Fundamentos de Programação	1	9	1	0	1	1	2	1	16
Gerência de Projetos de Software	0	3	1	0	1	2	2	1	10
Interface Humano-Computador	0	6	0	1	1	2	0	2	12
Introdução a Processos e Requisitos de Software	0	6	1	0	1	2	1	2	13
Introdução à Computação e Engenharia de Software	0	1	0	0	1	1	0	0	3
Laboratório de Programação	0	2	0	0	0	1	0	1	4
Linguagens de Programação	0	5	0	0	0	1	0	1	7
Lógica para Computação	1	6	1	0	0	1	0	0	9
Manutenção de Software	0	4	1	0	1	2	0	2	10
Matemática Básica	0	0	0	0	0	1	0	0	1
Matemática Discreta	0	0	0	0	0	0	0	0	0
Probabilidade e Estatística	0	0	0	0	0	0	0	0	0
Processos de Software	0	5	1	0	1	1	1	2	11
Programação Orientada a Objetos	1	8	0	0	1	1	2	2	15
Projeto Detalhado de Software	0	4	0	0	1	2	1	0	8
Projeto e Análise de Algoritmos	0	0	0	0	0	1	1	0	2
Qualidade de Software	0	5	1	0	1	2	3	2	14
Redes de Computadores	1	1	0	0	0	1	1	0	4
Redes e Sistemas Distribuídos	0	1	1	0	0	1	0	1	4
Requisitos de Software	0	7	2	0	1	2	2	2	16
Sistemas Operacionais	0	1	0	0	0	1	0	0	2
Verificação e Validação	1	8	1	0	1	1	2	2	16
Totais	8	114	16	1	15	37	27	28	

Fonte: o autor (2023)

Tabela 10 – Principais disciplinas mais aproveitadas para desempenhar um papel de Desenvolvedor de Software

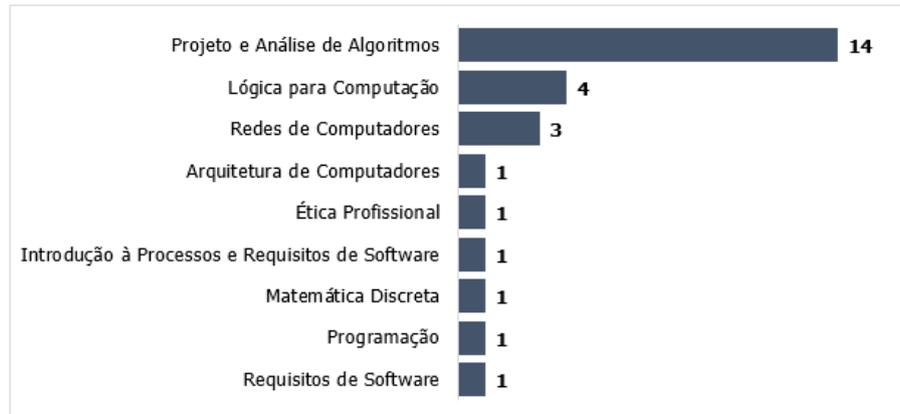
Disciplina	Ocorrências
Fundamentos de Programação	9
Estrutura de Dados	8
Programação Orientada a Objetos	8
Verificação e Validação	8
Análise e Projeto de Sistemas	7
Fundamentos de Banco de Dados	7
Requisitos de Software	7
Interface Humano-Computador	6
Introdução a Processos e Requisitos de Software	6
Lógica para Computação	6

Fonte: o autor (2023).

Com os dados apresentados pela Tabela 11, obtêm-se informações importantes que podem elucidar se as disciplinas mais difíceis, consideradas pelos alunos, têm influência ou não com o tipo de treinamento necessário para exercer o cargo na empresa. Além disso, pode-se saber quais conhecimentos as empresas desejam que seus funcionários tenham.

As disciplinas não obrigatórias para o curso de Engenharia de Software também foram colocadas em análise com a pergunta 10, “Das disciplinas que não fazem parte da grade

Figura 7 – Ocorrências das disciplinas mais difíceis consideradas pelos respondentes



Fonte: o autor (2023)

do curso, qual você julga ser a mais essencial para seu cargo”? Assim, foi possível entender se há demanda real de mercado para essas disciplinas. Então, foi feito um levantamento das disciplinas não obrigatórias do curso e que estão disponíveis para serem ofertadas<sup>1</sup>.

Figura 8 – Ocorrências das disciplinas optativas consideradas essenciais para exercer o cargo



Fonte: o autor (2023).

<sup>1</sup> De acordo com <http://www.campusrussas.ufc.br/docs/PPC-ES-Russas-Fluxograma.pdf>, acessado em 04 de abril de 2023.

Tabela 11 – Treinamento necessário relacionado com as disciplinas mais difíceis, segundo os respondentes da pesquisa

Respondente	Treinamento(s)	Disciplina(s) mais difícil(eis)
1	Treinamento de Liderança, certificação voltadas para Gestão	Projeto e Análise de Algoritmos
2	Cursos especializados, específicos	Projeto e Análise de Algoritmos
3	Fiz diversos cursos on-line sobre novas tecnologias: jenkins, spring boot, splunk, teste, etc.	Projeto e Análise de Algoritmos
4	Linguagem de programação Clojure	Projeto e Análise de Algoritmos
5	Testes Automatizados, Integração Contínua e Devops	Projeto e Análise de Algoritmos
6	Fiz alguns cursos para me aprofundar sobre React	Arquitetura de Computadores
7	Treinamento relacionado as tecnologias utilizadas, como Spring, por exemplo. Além de treinamento sobre o sistema no qual desempenharia o desenvolvimento.	Ética; IPRS; Requisitos de Software
8	Cursos de programação entre outros	Programação; Projeto e Análise de Algoritmos
9	Cursos de testes	Lógica para Computação
10	Treinamento dos processos que a empresa usa, Já tendo o conhecimento prévio dos processos vistos na graduação.	Projeto e Análise de Algoritmos
11	Habilidade técnicas, como aprendizado de outras linguagens de programação e frameworks.	Projeto e Análise de Algoritmos
12	Treinamentos sobre ferramentas, padrões e metodologias	Projeto e Análise de Algoritmos
13	Se não tivesse feito meio mundo de curso e aprimorado minhas competências não tinha nem entrado na empresa. Fora isso treinamento no sistema da empresa para saber o que e como faz;	Projeto e Análise de Algoritmos; Lógica para Computação
14	Ferramentas de testes automatizados, ferramenta de testes de api, tipos de testes, conceitos entre outros	Redes de Computadores
15	Treinamento para testes com cypress, Banco de dados na prática, qualidade de software na prática	Projeto e Análise de Algoritmos; Lógica para Computação; Redes de Computadores
16	Adaptação ao cenário do mercado de trabalho na qual estou envolvido, tive que me adaptar e entender como funcionaria as coisas.	Lógica para Computação
17	Em frameworks específicos como react native e react	Projeto e Análise de Algoritmos
18	Curso de scrum	Projeto e Análise de Algoritmos
19	Treinamento de adaptação ao processo da empresa. Mas a base teórica foi construída fortemente na faculdade e por estudo próprio	Projeto e Análise de Algoritmos; Redes de Computadores
20	Foi necessário realizar alguns cursos voltados para a gerência de produtos para compreender melhor quais as ferramentas e técnicas mais utilizadas pelo mercado.	Matemática Discreta

Fonte: o autor (2023).

A Tabela 12 apresenta uma das principais questões sobre o objetivo deste trabalho, pois trata da pergunta 9 do questionário, que é “*Dentre as disciplinas obrigatórias do curso, quais possuem uma abordagem diferente na prática na empresa que trabalha?*”? Os resultados obtidos mostram que a disciplina **Gerência de Projetos de Software** possui 7 ocorrências, isso corresponde à 30% das pessoas que responderam o questionário.

Como um complemento da pergunta 9, a pergunta 12, “*Dentre as disciplinas obrigatórias listadas abaixo, quais estão mais de acordo com a realidade da empresa em que*

*trabalha”?*, investiga quais disciplinas obrigatórias estão mais alinhadas com o que é posto em prática com os problemas reais de uma empresa que desenvolve software. Isso é mostrado nas Tabelas 12 e 13. Com esses dados pode-se destacar algumas disciplinas que foram percebidas tendo uma quantidade parecida de ocorrências entre as duas tabelas. Neste caso, destaca-se **Gerência de Projetos de Software**, aonde 9 pessoas disseram que a disciplina está alinhada com a prática real da empresa. Seguindo na mesma análise, a disciplina **Arquitetura de Software** obteve 7 respostas afirmando que existe alinhamento com a realidade da empresa, contra 6 que disseram não haver alinhamento.

Tabela 12 – Ocorrências das disciplinas obrigatórias que possuem uma abordagem diferente na prática

Disciplina	Ocorrências
Gerência de Projetos de Software	7
Arquitetura de Software	6
Qualidade de Software	5
Requisitos de Software	5
Análise e Projeto de Sistemas	4
Algoritmos em Grafos	4
Verificação e Validação	4
Linguagens de Programação	4
Redes de Computadores	4
Redes e Sistemas Distribuídos	4
Matemática Básica	4
Matemática Discreta	4
Probabilidade e Estatística	4
Projeto e Análise de Algoritmos	4
Arquitetura de Computadores	3
Processos de Software	3
Interface Humano-Computador	3
Fundamentos de Programação	2
Estrutura de Dados	2
Fundamentos de Banco de Dados	2
Manutenção de Software	2
Lógica para Computação	2
Sistemas Operacionais	2
Empreendedorismo	1
Projeto Detalhado de Software	1
Introdução à Computação e Engenharia de Software	1
Laboratório de Programação	1

Fonte: o autor (2023).

Tabela 13 – Ocorrências das disciplinas obrigatórias que estão mais de acordo com a realidade

Disciplina	Ocorrências
Qualidade de Software	14
Verificação e Validação	14
Requisitos de Software	12
Fundamentos de Banco de Dados	11
Programação Orientada a Objetos	11
Fundamentos de Programação	11
Manutenção de Software	10
Gerência de Projetos de Software	9
Ética Profissional	9
Análise e Projeto de Sistemas	8
Projeto Detalhado de Software	8
Interface Humano-Computador	7
Arquitetura de Software	7
Estrutura de Dados	7
Linguagens de Programação	7
Redes de Computadores	6
Lógica para Computação	6
Introdução a Processos e Requisitos de Software	6
Laboratório de Programação	6
Empreendedorismo	5
Processos de Software	5
Projeto e Análise de Algoritmos	4
Redes e Sistemas Distribuídos	4
Arquitetura de Computadores	3
Sistemas Operacionais	3
Introdução à Computação e Engenharia de Software	3
Matemática Básica	3
Algoritmos em Grafos	2
Matemática Discreta	2
Probabilidade e Estatística	2

Fonte: o autor (2023).

## 5.2 Pergunta Opcional

Para finalizar a análise, esta subseção apresenta os resultados obtidos pela pergunta 13, “*Escreva outras observações acerca de possíveis diferenças entre o curso de Engenharia de Software na universidade e as necessidades reais na indústria de software*”. A pergunta 13 era de caráter opcional e, dos 23 respondentes, 12 quiseram respondê-la. A Figura 9 mostra as respostas.

Com esses relatos, identificou-se que disciplinas optativas podem estar mais próximas da realidade que as demais, e há um relato de que **Gerência de Projetos de Software** e **Requisitos de Software** deveriam ser abordadas de maneira mais aprofundada. No entanto, não ficou claro quanto ao nível de profundidade adequado. Ainda assim, para aprofundar mais uma

disciplina, deve-se levar em consideração que alguns assuntos são muito extensos, e aprofundar demais ocasionaria a geração de conteúdo que não caberia em um semestre letivo.

Figura 9 – Respostas da pergunta 13 do questionário

Nº	Observações do respondente do questionário
1	As disciplinas optativas estão muito mais próximas da realidade do que as demais. Não que as obrigatórias não tenham seu devido valor, porém foram muito mal dadas. Um exemplo claro foram as disciplinas de Verificação e Validação e arquitetura de software, que é requisito quase que obrigatório nas empresas que trabalham com o mínimo de qualidade, e que os professores nem sequer tem experiência para ministrar. Manutenção por exemplo, os alunos não estão preparados para tocar em um produto em produção, as aulas são muito teóricas. Quer desafiar os alunos ? pega os alunos do 6-7 semestre e coloca pra desenvolver features com requisitos complexos em um software desenvolvido pelos alunos do 2 semestre.
2	Processos de acompanhamento de projetos e quais técnicas mais utilizadas no mercado para resolver os problemas mais recorrentes dos projetos, ex: atrasos, troca de escopo..
3	Nas disciplinas relacionadas ao desenvolvimento de software, gerência e requisitos deveriam ter um aprofundamento maior, e tentar simular mais a utilização prática das metodologias, utilização de ferramentas de gerenciamento de desenvolvimento em geral
4	Acho que falta um pouco de dev ops e computação em nuvem no curso (pelo menos eu não fiz nada relacionado a isso). Trabalho bastante com QA e testes. Acho que é interessante ter uma abordagem mais prática dos testes, além dos testes unitários.
5	No ponto de vista do desenvolvedor, quanto mais você trabalha com programação você adquire conhecimento. Acho que necessitaria de mais cadeira opcionais de programação um pouco mais avançada pra quem quer se aprimorar, uma disciplina que não foi citada que eu acho muito importante é Banco de Dados Avançados, tem muita coisa que precisa ser vista e acredito que só fundamentos não daria tempo.
6	Estou indo para meu 3 emprego na área e até agora aproximadamente 60% do que eu aprendi na faculdade nunca utilizei.
7	As disciplinas trazem um conteúdo muito desatualizado do que é utilizado no mercado.
8	Muitas tecnologias que o mercado atual exige, o curso não está suprindo. O essencial seria o coordenador de curso verificar como o mercado está funcionando e aplicar medidas entre a mercado e a instituição de ensino.
9	tudo que vi na graduação me tornou um profissional excelente
10	Muitos conteúdos vistos na graduação não serão utilizados pelo profissional, independente do cargo, porém alguns são muito importantes para firmar uma base de conhecimento.
11	O embasamento do curso é ótimo, talvez necessite de mais imersões no cotidiano, mas isso é apoiado por meio de projetos e a empresa jr.  Considerações Questionário: - As perguntas 8 e 11 são muito parecidas. - Pensando em uma empresa que vivencia todo um processo de ciclo de vida, a empresa conforme a pergunta 12 vivenciará toda a teoria de todas as disciplinas.
12	As diferenças entre a visão teórica com a prática na indústria, muitas vezes ocorre devido as empresas não aplicarem corretamente alguns conceitos no desenvolvimento como na gerência. Na minha opinião isso ocorre devido alguns pontos fundamentais: cultura da empresa que é difícil de mudar e; profissionais que não possui uma formação abrangente como tem o engenheiro de software e; muitos profissionais de outras áreas que estão fazendo transição de carreira para áreas de tecnologia sem ter o conhecimento adequado. O conteúdo do curso por ser amplo se torna fundamental para o profissional aplicar corretamente na indústria.

Fonte: o autor (2023).

## 6 CONSIDERAÇÕES FINAIS

Este trabalho pretendeu analisar as competências técnicas de engenharia de software entre o meio acadêmico e a indústria de software, a fim de identificar se há um desequilíbrio nos níveis de conhecimentos oferecidos na universidade em relação ao que é exigido na indústria de software. Essa análise partiu de uma pesquisa feita por questionário aplicado à alunos e recém-formados do curso de Engenharia de Software.

O presente trabalho contribui dando o passo inicial ao tratar do tema dentro da UFC – Campus Russas, levantando informações de dentro e fora da universidade, permitindo alinhar cada vez mais o curso de Engenharia de Software com as expectativas da indústria de software.

Para se atingir o objetivo geral de que as competências da engenharia de Software, estudadas na universidade, estão alinhadas ou não às necessidades da indústria de software, levantou-se informações que apontam as necessidades da indústria de software. Essas informações foram obtidas com o relato dos participantes da pesquisa sobre os treinamentos que precisaram fazer para exercerem seus cargos nas empresas. Além disso, foram identificadas no SWECOM quais são as disciplinas específicas da Engenharia de Software.

Com isso, a análise permitiu identificar que há uma divergência na percepção das pessoas quanto a noção do que é uma disciplina da universidade estar alinhada à realidade da indústria, visto que fazer tal avaliação, muitas vezes implica assumir que os alunos passaram por diferentes experiências na sua formação.

Com as informações levantadas por esta pesquisa foi possível saber que as disciplinas específicas da Engenharia de Software, estudadas na universidade, estão cumprindo seu papel de serem fundamentais para o profissional que está na indústria, e que as disciplinas que não são específicas ainda se mostram tão importantes quanto as específicas. Foi possível também obter o dado de que uma mesma disciplina foi entendida por uma quantidade de participantes como alinhada com a realidade da empresa, mas também uma quantidade bem próxima de participantes julgaram essa disciplina como desalinhada com a realidade da indústria. Os treinamentos informados pelos participantes não indicaram relação com as competências da engenharia de software, mostrando que não houve a necessidade de aprimorar tais competências.

O instrumento de coleta dos dados trouxe algumas facilidades durante a aplicação devido a sua natureza digital, possibilitando monitorar quem abriu o e-mail e até quem abriu o link do questionário, permitindo assim, que fosse feito um novo envio do questionário apenas para aqueles que ainda não haviam aberto o e-mail até certo tempo.

No entanto, foi percebido um problema quanto a concepção do questionário. Este problema refere-se à quantidade de alternativas que foram dadas nas perguntas 8 à 12. Uma vez que as disciplinas diretamente ligas às competências técnicas foram identificadas, poderia ter sido disponibilizadas como opções de respostas apenas as disciplinas identificadas. Houve ainda uma baixa quantidade de respostas ao questionário, enquanto esperava-se uma participação acima de 50% de respondentes, pelo menos.

Em pesquisas futuras, pode-se ajustar a delimitação das opções de respostas do questionário, buscando tratar apenas das disciplinas que são relevantes na investigação. Pode-se também expandir o escopo da pesquisa incluindo as habilidades comportamentais (*soft skills*) do engenheiro de software, buscando compreender melhor todos os aspectos que envolvem a sua formação.

Quanto às disciplinas optativas apontadas como essenciais na Engenharia de Software, recomenda-se estudar a possibilidade de torná-las obrigatórias. Uma alternativa para tornar obrigatória uma disciplina optativa sem aumentar a quantidade de disciplinas da grade curricular, é diminuir uma disciplina optativa para aumentar uma obrigatória. Sugere-se ainda, que haja uma intensificação na parceria da universidade com a indústria de software, para que os alunos possam experienciar projetos reais, além de a universidade apresentar para a indústria as soluções de problemas e inovações criadas pelo alunos.

## REFERÊNCIAS

- BOTELHO, J. M.; CRUZ, V. A. G. da. **Metodologia Científica**. São Paulo: Pearson Education do Brasil, 2013. ISBN 978-85-430-0006-0.
- BOUILLON, J.-L. Guy le boterf, de la compétence : Essai sur un attracteur étrange. 1995. 1996. ISSN 1168-1446. Disponível em: [https://www.persee.fr/doc/sciso\\_1168-1446\\_1996\\_num\\_37\\_1\\_1502\\_t1\\_0214\\_0000\\_2](https://www.persee.fr/doc/sciso_1168-1446_1996_num_37_1_1502_t1_0214_0000_2).
- BRASIL. Conselho Nacional de Educação (CNE). Resolução nº 5, de 2016. **Diário Oficial da União**, Brasília, DF, p. 22–24, nov 2016.
- CONCLA. **Classificação Nacional de Atividades Econômicas – CNAE versão 2.0**. Rio de Janeiro: IBGE, 2007. ISBN 978-85-240-3971-3. Disponível em: <https://concla.ibge.gov.br/documentacao/documentacao-cnae-2-0.html>. Acesso em: 12 jan. 2022.
- FLEURY, M. T. L.; FLEURY, A. Construindo o conceito de competência. **Revista de Administração Contemporânea**, FapUNIFESP (SciELO), v. 5, n. spe, p. 183–196, 2001. Disponível em: <https://doi.org/10.1590/s1415-65552001000500010>.
- GAROUSI, V.; GIRAY, G.; TUZUN, E.; CATAL, C.; FELDERER, M. Closing the gap between software engineering education and industrial needs. **IEEE Software**, v. 37, n. 2, p. 68–77, 2020.
- GAROUSI, V.; GIRAY, G.; TüzÜN, E.; CATAL, C.; FELDERER, M. Aligning software engineering education with industrial needs: A meta-analysis. **Journal of Systems and Software**, v. 156, p. 65–83, 2019. ISSN 0164-1212. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0164121219301347>.
- GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. ISBN 85-224-3169-8.
- GIMENES, I. Os dilemas didáticos da engenharia de software. **Computação Brasil**, Sociedade Brasileira de Computação - SBC, n. 28, p. 21–25, 2015. Disponível em: <https://www.sbc.org.br/component/flippingbook/book/22>.
- HIRAMA, K. **Engenharia de Software: Qualidade e Produtividade com Tecnologia**. Rio de Janeiro: Elsevier Brasil, 2012. ISBN 9788535248814.
- IEEE. IEEE standard glossary of software engineering terminology. **IEEE Std 610.12-1990**, IEEE, p. 1–84, 1990. Disponível em: <https://doi.org/10.1109/ieeestd.1990.101064>.
- IEEE (Ed.). **Software Engineering Competency Model (SWECOM)**. Version 1.0. IEEE Computer Society, 2014. ISBN 978-0-7695-5373-3. Disponível em: <https://www.computer.org/volunteering/boards-and-committees/professional-educational-activities/software-engineering-competency-model>.
- KELLY, M. **From airline reservations to Sonic the Hedgehog: a history of the software industry**. Cambridge, Mass: The MIT Press, 2003. ISBN 0-262-03303-8.
- LAKATOS, E. M.; MARCONI, M. de A. **Fundamentos de Metodologia Científica**. 8. ed. São Paulo: GEN – Atlas S.A., 2017. ISBN 978-85-970-1076-3.

LAPLANTE, P. A. **What Every Engineer Should Know about Software Engineering**. 1st. ed. USA: CRC Press, Inc., 2017. ISBN 1138468533.

LETHBRIDGE, T. C. Priorities for the education and training of software engineers. **Journal of Systems and Software**, v. 53, n. 1, p. 53–71, 2000. ISSN 0164-1212. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0164121200000091>.

LIARGKOVAS, G.; PAPADOPOULOU, A.; KOTTI, Z.; SPINELLIS, D. Software engineering education knowledge versus industrial needs. **IEEE Transactions on Education**, p. 1–9, 2021.

LIEBENBERG, J.; HUISMAN, M.; MENTZ, E. Knowledge and skills requirements for software developer students. **International Journal of Computer and Information Engineering**, World Academy of Science, Engineering and Technology, v. 8, n. 8, p. 2612 – 2617, 2014. ISSN eISSN: 1307-6892. Disponível em: <https://publications.waset.org/vol/92>.

MERRIAM-WEBSTER. **The Merriam-Webster Dictionary**. Merriam-Webster, Inc, 2016. Disponível em: <https://www.merriam-webster.com/dictionary/competence>. Acesso em: 6 jan. 2022.

MICHAELIS. **Michaelis Moderno Dicionário da Língua Portuguesa**. Melhoramentos Ltda., 2015. ISBN 978-85-06-04024-9. Disponível em: <https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/competência/>. Acesso em: 6 jan. 2022.

NAUR, P.; RANDELL, B. (Ed.). **Software Engineering: Report of a Conference Sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO**. Scientific Affairs Division, NATO, 1969. Disponível em: <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/NATOREports/index.html>.

PIETERSE, V.; EEKELEN, M. V. Which are harder? soft skills or hard skills? p. 1–8, 2016. Disponível em: <http://hdl.handle.net/2263/60615>.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software: Uma Abordagem Profissional**. 8ª. ed. São Paulo: McGraw Hill Brasil, 2016. ISBN 978-8580555332.

SOMMERVILLE, I. **Engenharia de Software**. 10ª. ed. São Paulo: Pearson, 2015. ISBN 978-85-430-2497-4.

Task Force, C. **Computing Curricula 2020: Paradigms for Global Computing Education**. New York, NY, USA: Association for Computing Machinery (ACM), 2020. ISBN 9781450390590.

WAZLAWICK, R. S. **Engenharia de Software: Conceitos e Práticas**. 1ª. ed. Rio de Janeiro: Elsevier, 2013. ISBN 978-85-352-6120-2.

ZARIFIAN, P. **Objectif Compétence. Pour une nouvelle logique**. Editions Liaisons, 1999. 229 p. Disponível em: <https://halshs.archives-ouvertes.fr/halshs-00438955>.

## APÊNDICE A – QUESTIONÁRIO UTILIZADO PARA OBTER AS RESPOSTAS DA PESQUISA

**Questão 1.** Qual a sua idade?

**Questão 2.** Qual o seu sexo?

- (a) Feminino
- (b) Masculino

**Questão 3.** Qual cargo você ocupa na empresa em que trabalha?

**Questão 4.** Você já fez todas as disciplinas obrigatórias do curso de Engenharia de Software?

- (a) Sim
- (b) Não

**Questão 5.** Precizou de treinamento, além da formação acadêmica, para desempenhar as funções de seu cargo?

- (a) Sim
- (b) Não

**Questão 6.** Se a resposta anterior foi “Sim”, informe qual o treinamento.

**Questão 7.** Qual das disciplinas obrigatórias do curso de Engenharia de Software você julga ser a mais difícil?

**Questão 8.** De todas as disciplinas obrigatórias do curso, quais foram mais aproveitadas para executar seu cargo?

*(Você pode marcar mais de um item.)*

- |                                                        |                                                           |                                                                           |
|--------------------------------------------------------|-----------------------------------------------------------|---------------------------------------------------------------------------|
| <input type="checkbox"/> Algoritmos em Grafos          | <input type="checkbox"/> Fundamentos de Banco de Dados    | <input type="checkbox"/> Requisitos de Software                           |
| <input type="checkbox"/> Análise e Projeto de Sistemas | <input type="checkbox"/> Fundamentos de Programação       | <input type="checkbox"/> Introdução à Computação e Engenharia de Software |
| <input type="checkbox"/> Arquitetura de Computadores   | <input type="checkbox"/> Gerência de Projetos de Software | <input type="checkbox"/> Laboratório de Programação                       |
| <input type="checkbox"/> Arquitetura de Software       | <input type="checkbox"/> Interface Humano-Computador      | <input type="checkbox"/> Linguagens de Programação                        |
| <input type="checkbox"/> Empreendedorismo              | <input type="checkbox"/> Introdução a Processos e         | <input type="checkbox"/> Lógica para Computação                           |
| <input type="checkbox"/> Estrutura de Dados            |                                                           |                                                                           |
| <input type="checkbox"/> Ética Profissional            |                                                           |                                                                           |

- |                                                           |                                                               |                                                             |
|-----------------------------------------------------------|---------------------------------------------------------------|-------------------------------------------------------------|
| <input type="checkbox"/> Manutenção de Software           | <input type="checkbox"/> Objetos                              | <input type="checkbox"/> Redes e Sistemas Distribu-<br>ídos |
| <input type="checkbox"/> Matemática Básica                | <input type="checkbox"/> Projeto Detalhado de<br>Software     | <input type="checkbox"/> Requisitos de Software             |
| <input type="checkbox"/> Matemática Discreta              | <input type="checkbox"/> Projeto e Análise de Al-<br>goritmos | <input type="checkbox"/> Sistemas Operacionais              |
| <input type="checkbox"/> Probabilidade e Estatís-<br>tica | <input type="checkbox"/> Qualidade de Software                | <input type="checkbox"/> Verificação e Validação            |
| <input type="checkbox"/> Processos de Software            | <input type="checkbox"/> Redes de Computadores                |                                                             |
| <input type="checkbox"/> Programação Orientada a          |                                                               |                                                             |

**Questão 9.** Dentre as disciplinas obrigatórias do curso, quais possuem uma abordagem diferente na prática na empresa que trabalha?

*(Você pode marcar mais de um item.)*

- |                                                              |                                                                                     |                                                               |
|--------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------|
| <input type="checkbox"/> Algoritmos em Grafos                | <input type="checkbox"/> Computador                                                 | <input type="checkbox"/> Processos de Software                |
| <input type="checkbox"/> Análise e Projeto de Sis-<br>temas  | <input type="checkbox"/> Introdução a Processos e<br>Requisitos de Software         | <input type="checkbox"/> Programação Orientada a<br>Objetos   |
| <input type="checkbox"/> Arquitetura de Computa-<br>dores    | <input type="checkbox"/> Introdução à Computa-<br>ção e Engenharia de Soft-<br>ware | <input type="checkbox"/> Projeto Detalhado de<br>Software     |
| <input type="checkbox"/> Arquitetura de Software             | <input type="checkbox"/> Laboratório de Programa-<br>ção                            | <input type="checkbox"/> Projeto e Análise de Al-<br>goritmos |
| <input type="checkbox"/> Empreendedorismo                    | <input type="checkbox"/> Linguagens de Programa-<br>ção                             | <input type="checkbox"/> Qualidade de Software                |
| <input type="checkbox"/> Estrutura de Dados                  | <input type="checkbox"/> Lógica para Computação                                     | <input type="checkbox"/> Redes de Computadores                |
| <input type="checkbox"/> Ética Profissional                  | <input type="checkbox"/> Manutenção de Software                                     | <input type="checkbox"/> Redes e Sistemas Distri-<br>buídos   |
| <input type="checkbox"/> Fundamentos de Banco<br>de Dados    | <input type="checkbox"/> Matemática Básica                                          | <input type="checkbox"/> Requisitos de Software               |
| <input type="checkbox"/> Fundamentos de Progra-<br>mação     | <input type="checkbox"/> Matemática Discreta                                        | <input type="checkbox"/> Sistemas Operacionais                |
| <input type="checkbox"/> Gerência de Projetos de<br>Software | <input type="checkbox"/> Probabilidade e Estatís-<br>tica                           | <input type="checkbox"/> Verificação e Validação              |
| <input type="checkbox"/> Interface Humano-                   |                                                                                     |                                                               |

**Questão 10.** Das disciplinas que não fazem parte da grade do curso, qual você julga ser a mais essencial para seu cargo?

*(Você pode marcar mais de um item.)*

- |                                                                               |                                                                     |                                                                          |
|-------------------------------------------------------------------------------|---------------------------------------------------------------------|--------------------------------------------------------------------------|
| <input type="checkbox"/> Álgebra Linear                                       | <input type="checkbox"/> Estimativa De Custos Em                    | <input type="checkbox"/> Autômatos                                       |
| <input type="checkbox"/> Cálculo Diferencial E Integral I                     | <input type="checkbox"/> Projetos De Software                       | <input type="checkbox"/> Matemática Computacional                        |
| <input type="checkbox"/> Compiladores                                         | <input type="checkbox"/> Estruturas De Dados Avançadas              | <input type="checkbox"/> Métodos E Ferramentas Da Engenharia De Software |
| <input type="checkbox"/> Computação Em Nuvem                                  | <input type="checkbox"/> Experimentação Em Engenharia De Software   | <input type="checkbox"/> Pré-Cálculo                                     |
| <input type="checkbox"/> Desenvolvimento De Software Concorrente              | <input type="checkbox"/> Fundamentos Da Economia E Da Administração | <input type="checkbox"/> Redes Sociais                                   |
| <input type="checkbox"/> Desenvolvimento De Software Para Dispositivos Móveis | <input type="checkbox"/> Gerência De Configuração                   | <input type="checkbox"/> Reuso De Software                               |
| <input type="checkbox"/> Desenvolvimento De Software Para Persistência        | <input type="checkbox"/> Integração De Aplicações                   | <input type="checkbox"/> Segurança                                       |
| <input type="checkbox"/> Desenvolvimento De Software Para Web                 | <input type="checkbox"/> Inteligência Artificial                    | <input type="checkbox"/> Sistemas Distribuídos                           |
| <input type="checkbox"/> Especificação Formal De Software                     | <input type="checkbox"/> Introdução Ao Desenvolvimento De Jogos     | <input type="checkbox"/> Sistemas Multiagentes                           |
|                                                                               | <input type="checkbox"/> Leitura De Software                        | <input type="checkbox"/> Teoria Da Computação                            |
|                                                                               | <input type="checkbox"/> Linguagens Formais E                       | <input type="checkbox"/> Trabalho Cooperativo Baseado Em Computadores    |

**Questão 11.** Das disciplinas que não fazem parte da grade do curso, qual você julga ser a mais essencial para seu cargo?

*(Você pode marcar mais de um item.)*

- |                                                        |                                                                           |                                                      |
|--------------------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------|
| <input type="checkbox"/> Algoritmos em Grafos          | <input type="checkbox"/> Fundamentos de Programação                       | <input type="checkbox"/> Laboratório de Programação  |
| <input type="checkbox"/> Análise e Projeto de Sistemas | <input type="checkbox"/> Gerência de Projetos de Software                 | <input type="checkbox"/> Linguagens de Programação   |
| <input type="checkbox"/> Arquitetura de Computadores   | <input type="checkbox"/> Interface Humano-Computador                      | <input type="checkbox"/> Lógica para Computação      |
| <input type="checkbox"/> Arquitetura de Software       | <input type="checkbox"/> Introdução a Processos e Requisitos de Software  | <input type="checkbox"/> Manutenção de Software      |
| <input type="checkbox"/> Empreendedorismo              | <input type="checkbox"/> Introdução à Computação e Engenharia de Software | <input type="checkbox"/> Matemática Básica           |
| <input type="checkbox"/> Estrutura de Dados            |                                                                           | <input type="checkbox"/> Matemática Discreta         |
| <input type="checkbox"/> Ética Profissional            |                                                                           | <input type="checkbox"/> Probabilidade e Estatística |
| <input type="checkbox"/> Fundamentos de Banco de Dados |                                                                           | <input type="checkbox"/> Processos de Software       |

- |                                                          |                                                   |                                                  |
|----------------------------------------------------------|---------------------------------------------------|--------------------------------------------------|
| <input type="checkbox"/> Programação Orientada a Objetos | <input type="checkbox"/> Algoritmos               | <input type="checkbox"/> Requisitos de Software  |
| <input type="checkbox"/> Projeto Detalhado de Software   | <input type="checkbox"/> Qualidade de Software    | <input type="checkbox"/> Sistemas Operacionais   |
| <input type="checkbox"/> Projeto e Análise de Al-        | <input type="checkbox"/> Redes de Computadores    | <input type="checkbox"/> Verificação e Validação |
|                                                          | <input type="checkbox"/> Redes e Sistemas Distri- |                                                  |
|                                                          | <input type="checkbox"/> buídos                   |                                                  |

**Questão 12.** Dentre as disciplinas obrigatórias listadas abaixo, quais estão mais de acordo com a realidade da empresa em que trabalha?

*(Você pode marcar mais de um item.)*

- |                                                           |                                                                           |                                                          |
|-----------------------------------------------------------|---------------------------------------------------------------------------|----------------------------------------------------------|
| <input type="checkbox"/> Algoritmos em Grafos             | <input type="checkbox"/> Computador                                       | <input type="checkbox"/> Processos de Software           |
| <input type="checkbox"/> Análise e Projeto de Sistemas    | <input type="checkbox"/> Introdução a Processos e Requisitos de Software  | <input type="checkbox"/> Programação Orientada a Objetos |
| <input type="checkbox"/> Arquitetura de Computadores      | <input type="checkbox"/> Introdução à Computação e Engenharia de Software | <input type="checkbox"/> Projeto Detalhado de Software   |
| <input type="checkbox"/> Arquitetura de Software          | <input type="checkbox"/> Laboratório de Programação                       | <input type="checkbox"/> Projeto e Análise de Algoritmos |
| <input type="checkbox"/> Empreendedorismo                 | <input type="checkbox"/> Linguagens de Programação                        | <input type="checkbox"/> Qualidade de Software           |
| <input type="checkbox"/> Estrutura de Dados               | <input type="checkbox"/> Lógica para Computação                           | <input type="checkbox"/> Redes de Computadores           |
| <input type="checkbox"/> Ética Profissional               | <input type="checkbox"/> Manutenção de Software                           | <input type="checkbox"/> Redes e Sistemas Distribuídos   |
| <input type="checkbox"/> Fundamentos de Banco de Dados    | <input type="checkbox"/> Matemática Básica                                | <input type="checkbox"/> Requisitos de Software          |
| <input type="checkbox"/> Fundamentos de Programação       | <input type="checkbox"/> Matemática Discreta                              | <input type="checkbox"/> Sistemas Operacionais           |
| <input type="checkbox"/> Gerência de Projetos de Software | <input type="checkbox"/> Probabilidade e Estatística                      | <input type="checkbox"/> Verificação e Validação         |
| <input type="checkbox"/> Interface Humano-                |                                                                           |                                                          |
|                                                           |                                                                           |                                                          |

**Questão 13.** Escreva outras observações acerca de possíveis diferenças entre o curso de Engenharia de Software na universidade e as necessidades reais na indústria de software.

*(Opcional.)*