



INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE
PERNAMBUCO
DIRETORIA DE ENSINO A DISTÂNCIA
CURSO DE ESPECIALIZAÇÃO EM EDUCAÇÃO PROFISSIONAL, CIENTÍFICA E
TECNOLÓGICA

ALLAN DIEGO SILVA LIMA

Título: ENSINAMOS A PROGRAMAR DA FORMA QUE APRENDEMOS? UMA
ANÁLISE COMPARATIVA DE COMO OS PROFESSORES DO IFPE
APRENDERAM A PROGRAMAR E COMO ELES ENSINAM

Recife
2021

ALLAN DIEGO SILVA LIMA

Título: ENSINAMOS A PROGRAMAR DA FORMA QUE APRENDEMOS? UMA
ANÁLISE COMPARATIVA DE COMO OS PROFESSORES DO IFPE
APRENDERAM A PROGRAMAR E COMO ELES ENSINAM

Área de concentração: Ensino

Orientador (a): Carlos Marques Fernandes

Recife
2021

LISTA DE QUADROS

Quadro 1.1–Cursos de TI presenciais do IFPE e seus componentes curriculares de introdução à programação.....	10
Quadro 2.1- Espaços de debate científico nacionais sobre ensino da programação de computadores onde os artigos revisados neste capítulo foram publicados.	14
Quadro 2.2 – Resumo do foco primário dos principais artigos estudados durante a revisão bibliográfica.....	20
Quadro 3.1 – Roteiro das entrevistas.....	26
Quadro 3.2– Modelo de tabela comparativa para o relatório da pesquisa.	28
Quadro 3.3 – Modelo de tabela comparativa para o relatório da pesquisa.	29
Quadro 4.1 – Programa do Componente Curricular Lógica de Programação 1 do curso Técnico de Informática para Internet ofertado pelo <i>Campus</i> Belo Jardim.	31
Quadro 4.2 – Programa do Componente Curricular Lógica de Programação do curso de TADS ofertado pelo <i>Campus</i> Jaboatão dos Guararapes.....	32
Quadro 4.3 – Programa do Componente Curricular Lógica de Programação do curso de Informática para Internet ofertado pelo <i>Campus</i> Jaboatão dos Guararapes.	34
Quadro 4.4 – Programa do Componente Curricular Lógica de Programação do curso Técnico em Informática para Internet ofertado pelo <i>Campus</i> Palmares.....	37
Quadro 4.5 – Programa do Componente Curricular Introdução à Programação do curso de TADS ofertado pelo <i>Campus</i> Recife.....	39
Quadro 4.6 – Programa do Componente Curricular Lógica de Programação do curso de TADS ofertado pelo <i>Campus</i> Garanhuns.	40
Quadro 4.7 – Programa do Componente Curricular Introdução à Programação do curso de TADS ofertado pelo <i>Campus</i> Paulista.	42
Quadro 4.8 – Programa do Componente Curricular Lógica de Programação do curso de TADS ofertado pelo <i>Campus</i> Jaboatão dos Guararapes.....	44
Quadro 4.9 – Programa do Componente Curricular Lógica de Programação do curso de Engenharia de Software ofertado pelo <i>Campus</i> Belo Jardim.	48
Quadro 4.10 – Programa do Componente Curricular Programação Imperativa do curso de TSI ofertado pelo <i>Campus</i> Igarassu.	51
Quadro 5.1 – Tabela comparativa do processo ensino-aprendizagem da graduação do(a) docente #1.	57
Quadro 5.2 – Tabela comparativa do processo ensino-aprendizagem da graduação do(a) docente #2.	61
Quadro 5.3 – Tabela comparativa do processo ensino-aprendizagem da graduação do(a) docente #3.	64
Quadro 5.4 – Tabela comparativa do processo ensino-aprendizagem da graduação do(a) docente #4.	69
Quadro 5.5 – Tabela comparativa do processo ensino-aprendizagem do curso técnico de nível médio integrado do(a) docente #4.	70
Quadro 5.6 – Tabela comparativa do processo ensino-aprendizagem da graduação do(a) docente #5.	73

LISTA DE ABREVIATURAS E SIGLAS

CC – Componente Curricular

CCs – Componentes Curriculares

IFPE – Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco

PPC – Projeto Pedagógico de Curso

TASD - Tecnólogo em Análise e Desenvolvimento de Sistemas

TI – Tecnologia da Informação

TUPI – Técnico em Informática para Internet

TSI – Tecnólogo em Sistemas para Internet

SUMÁRIO

1	INTRODUÇÃO	7
1.1	OBJETIVO GERAL.....	11
1.2	OBJETIVOS ESPECÍFICOS	12
1.3	ORGANIZAÇÃO DO DOCUMENTO	12
2	REVISÃO BIBLIOGRÁFICA.....	13
3	METODOLOGIA.....	21
3.1	CAMPO E SELEÇÃO DOS SUJEITOS.....	21
3.2	MÉTODO.....	22
3.3	TÉCNICAS	23
3.4	INSTRUMENTOS DE COLETA	25
3.5	ANÁLISE	27
3.5.1	Construção dos relatórios das entrevistas.....	27
3.5.2	Diretrizes para interpretação e análise das entrevistas	29
4	OS CURSOS DE TECNOLOGIA DA INFORMAÇÃO DO IFPE E SEUS COMPONENTES CURRICULARES DE INTRODUÇÃO A PROGRAMAÇÃO	30
4.1	CURSO TÉCNICO EM INFORMÁTICA PARA INTERNET	30
4.1.1	Campus Belo Jardim – Lógica de Programação 1	30
4.1.2	Campus Igarassu – Lógica de Programação e Estruturas de Dados	32
4.1.3	Campus Jaboatão dos Guararapes – Lógica de Programação	34
4.1.4	Campus Palmares – Lógica de Programação	36
4.2	CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS (TADS)	38
4.2.1	Campus Recife – Introdução à Programação.....	39
4.2.2	Campus Garanhuns – Lógica de Programação	40
4.2.3	Campus Paulista – Introdução à Programação	41
4.2.4	Campus Jaboatão dos Guararapes – Lógica de Programação	43
4.3	CURSO DE BACHARELADO EM ENGENHARIA DE SOFTWARE	46
4.3.1	Campus Belo Jardim – Introdução à Programação	47
4.4	CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET (TSI)	49
4.4.1	Campus Igarassu – Programação Imperativa	50
4.5	Discussão.....	53
5	RESULTADOS DAS ENTREVISTAS E DISCUSSÕES	54
5.1	ENTREVISTA #1.....	54
5.1.1	Trajetória profissional e acadêmica	54
5.1.2	Estrutura e estudos durante a graduação	55
5.1.3	Comparativo entre processo de ensino-aprendizado na graduação e no IFPE	55

5.1.4	Tabela Comparativa do Processo Ensino-Aprendizagem	57
5.1.5	Análise da Entrevista.....	57
5.2	ENTREVISTA #2.....	58
5.2.1	Trajetória profissional e acadêmica	58
5.2.2	Estrutura e estudos durante a graduação	58
5.2.3	Comparativo entre processo de ensino-aprendizado na graduação e no IFPE	58
5.2.4	Tabela Comparativa do Processo Ensino-Aprendizagem	61
5.2.5	Análise da Entrevista.....	61
5.3	ENTREVISTA #3.....	62
5.3.1	Trajetória profissional e acadêmica	62
5.3.2	Estrutura e estudos durante a graduação	62
5.3.3	Comparativo entre processo de ensino-aprendizado na graduação e no IFPE	62
5.3.4	Tabela Comparativa do Processo Ensino-Aprendizagem	64
5.3.5	Análise da Entrevista.....	65
5.4	ENTREVISTA #4.....	65
5.4.1	Trajetória profissional e acadêmica	65
5.4.2	Estrutura e estudos durante a graduação	66
5.4.3	Comparativo entre processo de ensino-aprendizado na graduação e no IFPE	66
5.4.4	Tabela Comparativa do Processo Ensino-Aprendizagem	69
5.4.5	Análise da Entrevista.....	70
5.5	ENTREVISTA #5.....	71
5.5.1	Trajetória profissional e acadêmica	71
5.5.2	Estrutura e estudos durante a graduação	71
5.5.3	Comparativo entre processo de ensino-aprendizado na graduação e no IFPE	72
5.5.4	Tabela Comparativa do Processo Ensino-Aprendizagem	73
5.5.5	Análise da Entrevista.....	73
5.6	DISCUSSÃO	74
6	CONCLUSÕES E TRABALHOS FUTUROS	77
	REFERÊNCIAS.....	80
	APÊNDICE A – ENTREVISTA #6	86

1 INTRODUÇÃO

Historicamente monopolizados pelas universidades, os cursos de Tecnologia da Informação (TI) em instituições públicas em nosso país eram poucos e concentrados nas capitais até meados dos anos 2000. Além disso, por conta da projeção de empregabilidade e salários acima da média, os cursos da área frequentemente estavam entre os mais concorridos nos processos seletivos, o que gerava como consequência uma espécie de monopólio da classe média dentro do corpo discente dos cursos TI até o começo do século.

Assim, as metodologias empregadas no ensino da programação de computadores nas universidades possuíam um público-alvo muito bem definido, utilizando prerrogativas típicas dos estudantes advindos da classe média. Entre outras coisas, os professores admitiam em suas aulas que os estudantes foram socializados para dedicar parte do tempo livre aos estudos, que as casas dos mesmos possuíam uma estrutura com espaço para estudos, computador disponível, familiaridade com os conceitos da matemática do ensino médio além do conhecimento da língua inglesa.

Entretanto, no decorrer dos últimos 20 anos diversas políticas públicas por parte do governo federal foram adotadas visando democratizar o acesso aos cursos de TI:

1. A lei de cotas (PRESIDÊNCIA DA REPÚBLICA, 2012) que permitiu aos estudantes das escolas públicas ocuparem parte relevante das vagas das instituições federais;
2. aA abertura de novos cursos na área de TI por conta da expansão dos catálogos de cursos técnicos de nível médio (REPÚBLICA FEDERATIVA DO BRASIL, 2021)¹e cursos tecnológicos (REPÚBLICA FEDERATIVA DO BRASIL, 2016)² do ministério da educação;
3. A criação dos Institutos Federais (REPÚBLICA FEDERATIVA DO BRASIL, 2008) transformou os antigos Centros Federais de Educação Tecnológica (CEFETs) levando tais instituições a ocuparem uma

¹ Disponível também no seguinte endereço: <http://cnct.mec.gov.br/>

² Disponível também no seguinte endereço: <http://portal.mec.gov.br/catalogo-nacional-dos-cursos-superiores-de-tecnologia->

posição de protagonismo na oferta vagas em cursos técnicos e superiores na área de TI.

Uma das principais consequências dessas políticas públicas foi uma completa mudança no perfil dos estudantes de TI no país. Até os anos 2000 havia pouca diversidade entre os estudantes, que eram em sua maioria tipicamente homens cisgênero, caucasianos e advindos da classe média. Entretanto, atualmente, e especialmente fora das capitais, o que se vê é uma heterogeneidade maior entre os estudantes.

Dessa forma, as premissas nas quais os pilares do ensino da programação foram construídos nos anos 2000 já não podem mais ser consideradas como universais. Sendo assim, sua replicação no contexto atual tende a agravar problemas típicos dos cursos da área de tecnologia como os altos índices de reprovação e evasão. Além disso, o modelo tradicional de ensino tende a acentuar o problema clássico da dualidade da educação brasileira, uma vez que estudantes com os privilégios típicos da classe média tendem a adaptar-se melhor ao mesmo.

Neste contexto, a revisão literária sobre como o ensino da programação é discutido nacionalmente revelou que os projetos de pesquisa publicados frequentemente são voltados para a utilização de ferramentas tecnológicas no ensino (ROCHA et al., 2010; DE SOUZA, 2009; RAPKIEWICZ et al., 2006; MOREIRA e FAVERO, 2009). Entretanto, apesar de raras na literatura nacional, também é possível encontrar discussões prioritariamente voltadas para o processo ensino-aprendizagem (AURELIANO, 2016; CAMPOS, 2010; LIMA E MEIRELES, 2014). Dada a raridade dos projetos nacionais sobre o tema, a revisão bibliográfica foi complementada com projetos desenvolvidos internacionalmente.

Esse fato pode ser uma consequência da formação acadêmica dos próprios professores que são programadores e, portanto, capazes de desenvolver e utilizar facilmente ferramentas tecnológicas. Além disso, é importante ressaltar que os professores da área raramente possuem formação pedagógica. Assim, é natural que haja entre os professores um viés para o desenvolvimento de projetos no âmbito educacional com foco no uso de ferramentas tecnológicas. Porém, esse viés abre ao mesmo tempo uma lacuna para discussões sobre o ensino da programação em um âmbito mais abstrato e voltado para a metodologia de ensino e suas práticas.

Essa discussão torna-se ainda mais importante em um ambiente, onde os cursos de TI são conhecidos por terem altas taxas de evasão e desistência comparados à cursos de outras áreas. E, especialmente, em um período aonde o perfil dos estudantes vem mudando nos últimos anos, o que naturalmente leva a considerar se as práticas consolidadas no âmbito das universidades ainda se adéquam ao novo contexto.

É justamente nesse nicho do processo de ensino-aprendizagem da programação para um novo perfil de turmas que este projeto de pesquisa se insere. Mais especificamente, pretende realizar uma análise das práticas empregadas no processo de ensino-aprendizagem da programação. Assim, o projeto visa comparar como os docentes do IFPE (Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco) aprenderam a programar e como eles estão ensinando programação em suas aulas. Ou seja, o escopo deste projeto realiza um passo, ainda que incipiente, para a reflexão sobre as práticas de ensino da programação dentro do IFPE. Nesse sentido, é importante compreender as práticas às quais os docentes do IFPE foram expostos em seu ensino formal, as práticas empregadas em suas aulas e a percepção que os docentes possuem sobre a diferença entre os perfis dos seus colegas de graduação e dos seus estudantes no IFPE.

Assim, **objeto de análise este projeto é definido como:** o conjunto de práticas utilizadas pelos professores do IFPE no processo de ensino-aprendizagem da programação de computadores nos Componentes Curriculares (CCs) de introdução à programação dos cursos presenciais de tecnologia da informação de níveis técnico subsequente e superior no IFPE.

Como forma de delimitar o escopo da pesquisa e dar ao estudo características de produção científica, **o trabalho realizado será direcionado para a averiguação da seguinte hipótese:** os processos de ensino-aprendizagem da programação de computadores utilizados pelos docentes do IFPE possuem uma relação direta com os processos adotados pelos professores desses docentes em suas formações.

Nesse sentido, o contexto do estudo terá como foco o ensino da programação dos cursos presenciais de TI do IFPE de nível técnico subsequente e superior (tecnólogos e bacharelados) que têm a programação de computadores como um eixo fundamental e que apresentem ao menos um Componente Curricular (CC)

relacionado ao tema no primeiro semestre do curso, conforme retratado no Quadro 1.1.

Quadro 1.1–Cursos de TI presenciais do IFPE e seus componentes curriculares de introdução à programação.

Campus	Nível	Curso	Componente Curricular
Belo Jardim	Técnico	Informática para Internet	Lógica de Programação 1
	Bacharelado	Engenharia de Software	Introdução à Programação
Garanhuns	Tecnólogo	Análise e Desenvolvimento de Sistemas	Lógica de Programação
Igarassu	Técnico	Informática para Internet	Lógica de Programação e Estrutura de Dados
	Tecnólogo	Sistemas para Internet	Programação Imperativa
Jaboatão dos Guararapes	Técnico	Informática para Internet	Lógica de Programação
	Tecnólogo	Análise e Desenvolvimento de Sistemas	Lógica de Programação
Palmares	Técnico	Informática para Internet	Lógica de Programação
Paulista	Tecnólogo	Análise e Desenvolvimento de Sistemas	Introdução à Programação
Recife	Tecnólogo	Análise e Desenvolvimento de Sistemas	Introdução à Programação

Como os cursos técnicos subsequentes de nível médio e os cursos superiores na área de TI possuem algumas similaridades importantes para este projeto:

- 1) Possuem em comum o requisito de o estudante ter cursado o ensino médio completo, o que acaba atraindo estudantes com faixa etária similares (quando comparados às demais modalidades);
- 2) Possuem ementa similares dos componentes curriculares de introdução à programação como será explorado no capítulo 4;
- 3) O campo de atuação dos estudantes no mercado de trabalho é majoritariamente voltado para o desenvolvimento de software.

Em contrapartida, os cursos das modalidades técnico integrado ao ensino médio, Educação de Jovens e Adultos (EJA) e licenciatura não serão contemplados no escopo deste projeto, uma vez que possuem objetivos e/ou estudantes de faixa etária distintos dos cursos listados no Quadro 1.1. Além disso, os cursos técnicos subsequentes de nível médio de computação gráfica, de manutenção e suporte em informática, e redes de computadores também não serão objeto de estudo uma vez que neles a programação é tratada como uma área complementar.

Além disso, os professores dos cursos de TI do *Campus Igarassu* não serão contemplados por este estudo, uma vez que o autor desta pesquisa era, no

momento da escrita da monografia, o professor responsável por ministrar ambos os CCs de introdução do *Campus*.

Uma vez que este trabalho propõe uma reflexão sobre a forma que programação é ensinada, e não como utilizar ferramentas tecnológicas para ensiná-la, o projeto coloca-se no sentido oposto a maioria daqueles publicados na área onde o foco é o uso de ferramentas tecnológicas. Ou seja, este projeto coloca em posição de protagonismo a discussão sobre o processo de ensino-aprendizagem da programação.

Como consequência dessa reflexão, esperam-se, no longo prazo, adequações nos modelos ensino-aprendizagem da programação, visando melhorias nos índices de evasão e reprovação nos primeiros semestres dos cursos de TI do IFPE e mais especificamente no *campus* Igarassu onde o autor da monografia está lotado.

Por colaborar com uma maior taxa de formação dos estudantes no *Campus* Igarassu, o impacto desse projeto, por sua vez, tende a influir positivamente na geração de riqueza local. Algo que é um fator importante para o desenvolvimento regional, no âmbito de uma sociedade de economia capitalista.

Assim, é possível justificar o desenvolvimento deste projeto através de contribuições nos seguintes contextos:

- **Científico** – uma vez que o projeto realiza uma rara e importante discussão sobre o ensino na programação de forma contextualizada ao contrário da maioria dos projetos que tem como foco à utilização de ferramentas tecnológicas;
- **Institucional (IFPE)** – uma vez que esta monografia oferta aos professores do IFPE uma reflexão crítica sobre as práticas utilizadas em suas aulas, o que pode ser útil na definição das ações adotadas em suas aulas de introdução à programação;
- **Local (*Campus* Igarassu)** – dado que o projeto atua para defender a adequação do ensino da programação no âmbito do IFPE - *Campus* Igarassu ao perfil dos estudantes.

1.1 OBJETIVO GERAL

Realizar uma análise comparativa entre métodos empregados no processo de ensino-aprendizagem da programação de computadores aos quais os professores

do IFPE foram expostos em suas graduações, e a forma que eles ensinam programação atualmente em suas aulas ministradas no IFPE.

1.2 OBJETIVOS ESPECÍFICOS

- 1 Identificar os métodos empregados no processo de ensino-aprendizagem da programação que os professores dos componentes curriculares de introdução à programação dos cursos TI do IFPE foram expostos durante suas graduações;
- 2 Identificar os métodos empregados no processo de ensino-aprendizagem da programação nos componentes curriculares de introdução à programação dos cursos presenciais de TI do IFPE;
- 3 Realizar uma análise comparativa sobre os dados coletados para alcançar os objetivos 1 e 2, visando identificar se a hipótese da pesquisa é verdadeira ou não.

1.3 ORGANIZAÇÃO DO DOCUMENTO

Os demais capítulos desta monografia estão divididos da seguinte forma: o capítulo 2 retrata a revisão bibliográfica realizada durante o projeto, apresentando um resumo sobre os projetos presentes na literatura científica sobre o ensino da programação; o capítulo 3 descreve a metodologia do projeto, expondo especialmente como as entrevistas foram planejadas; o capítulo 4 do projeto realiza uma contextualização do ensino da programação no IFPE trazendo os programas do CCs sobre o tema e realizando uma breve descrição sobre suas características; o capítulo 5 retrata os resultados das entrevistas e elabora uma análise visando encontrar evidências que provem ou não a hipótese que motivou esta pesquisa; por fim, o capítulo 6 apresenta as conclusões e as oportunidades encontradas para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

No âmbito do ensino da programação a maioria dos projetos encontrados na literatura possui um foco muito bem determinado para aplicações de ferramentas tecnológicas no ensino do tema. Nesta seção serão abordados exemplos de projetos que seguem essa linha, como forma de ressaltar como a discussão na academia sobre o ensino na programação está diretamente ligada à aspectos tecnológicos em detrimento de aspectos metodológicos.

Nesse contexto, o artigo “Ensino e Aprendizagem de Programação: Análise da Aplicação de Proposta Metodológica Baseada no Sistema Personalizado de Ensino” (ROCHA et al., 2010) trata do desenvolvimento de uma ferramenta para o ensino da programação inspirado em um sistema personalizado de ensino. Dessa forma, cada estudante envolvido pode seguir seu próprio ritmo de aprendizagem à medida que vai resolvendo os exercícios, do nível mais fácil ao nível mais difícil. A ferramenta foi aplicada no componente curricular Algoritmos, do primeiro semestre do Curso de Sistemas de Informação da Universidade Federal do Pará, e foi integrada ao ambiente de ensino Moodle. Os testes foram conduzidos através de uma metodologia quantitativa onde os índices de reprovação e aprovação do componente curricular foram comparados no decorrer dos semestres.

Outro exemplo de projeto desenvolvido no contexto do ensino da programação é o “VisuAlg-Ferramenta de apoio ao ensino de programação” (DE SOUZA, 2009). Durante o projeto foi desenvolvido um ambiente para a linguagem de programação Portugol capaz de mostrar o estado da memória do computador e a saída do programa durante sua execução. Novamente, o que se vê é uma priorização da discussão da ferramenta como mecanismo de ensino em detrimento daquilo que no âmbito da pedagogia justificaria a sua utilização. E neste caso com um agravante a mais, uma vez que o artigo publicado priva-se de apresentar informações relacionadas à aplicação ou experimentos utilizando a ferramenta, seu valor científico acaba por ser reduzido.

O terceiro exemplo selecionado para esta revisão bibliográfica, um projeto relatado através do artigo “Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais” (RAPKIEWICZ et al. 2006) consiste na utilização de jogos eletrônicos para o ensino da programação. Os autores utilizaram como principal referência teórica o trabalho de Piaget sobre a interação entre o estudante (sujeito) e os jogos (objetos). Nesse sentido, os autores

propõem que os estudantes construam coletivamente uma solução algorítmica para resolver um problema apresentado pelo jogo, e posteriormente convertê-la para um programa de computador. Apesar de a fundamentação teórica utilizar tanto Piaget, quanto a teoria dos jogos para justificar o desenvolvimento da abordagem, o artigo não apresenta resultados relacionados aos estudantes sobre o impacto da abordagem. Assim o projeto tem sua caracterização mais alinhada com o conceito de relato de experiência.

O quarto exemplo selecionado para compor esta revisão da literatura, trata do desenvolvimento e aplicação de uma ferramenta de correção automática de programas. O projeto foi descrito através do trabalho “Um ambiente para ensino de programação com feedback automático de exercícios” (MOREIRA e FAVERO, 2009). O artigo destaca-se pela discussão sobre as dificuldades no ensino da programação, propondo uma ferramenta de correção automática do código-fonte de programas como uma forma de atuar na dificuldade que os professores possuem de avaliar os estudantes rapidamente. Assim, o artigo foca nas estratégias (do ponto de vista técnico) para avaliação dos programas e sua efetividade (capacidade de corrigir corretamente os programas), privando-se, entretanto, de debater resultados concretos no âmbito pedagógico da abordagem proposta.

Diante desse contexto, e visando compreender as razões dos projetos sobre ensino da programação apresentarem uma forte relação com o uso de ferramentas tecnológicas, é importante observar que o foco tecnológico das publicações possa ter relação com os ambientes de publicação e discussão sobre o tema (Quadro 2.1) onde as mesmas foram publicadas.

Quadro 2.1- Espaços de debate científico nacionais sobre ensino da programação de computadores onde os artigos revisados neste capítulo foram publicados.

Tipo	Título	Sigla
Periódico	Revista Novas Tecnologias na Educação	RENTE
Evento	Workshop de Informática na Escola	WIE
Evento	Simpósio Brasileiro de Informática na Educação	SBIE
Periódico	Revista Brasileira de Informática na Educação	CEIE

Como exemplificado até o momento, na literatura acadêmica brasileira sobre o ensino da programação não se exime em reconhecer as dificuldades impostas ao

ensino de um tema tão complexo. Entretanto, as soluções propostas partem abordagens tecnológicas para solução de questões pontuais (correção de exercícios, criação de algoritmos, compreensão do código, entre outros). Esse tipo de projetos é relativamente comum na literatura, sendo os quatro citados neste capítulo exemplos de um conjunto de publicações e projetos bem maior que só não foi abordado em mais detalhes por conta da natureza e objetivos desta pesquisa.

Uma vez compreendido o viés para o uso de ferramentas tecnológicas, é importante relatar também exemplos de trabalhos nacionais que tivessem o processo ensino-aprendizagem como foco principal e onde tais projetos encontrariam um espaço adequado para sua publicação.

Neste contexto, o artigo “Metodologias de Ensino de Lógica de Programação” (LIMA; MEIRELES, 2014) discute três estratégias para o ensino da programação: fluxogramas, pseudocódigo e jogos educacionais. Durante o texto, as três metodologias são descritas e analisadas a respeito dos seus pontos positivos e negativos. Ao final do artigo, os autores defendem que o uso das metodologias citadas pode aumentar o número de estudantes que conseguem assimilar o conteúdo das aulas, uma vez que eles dão pluralidade à forma de ensino, ofertando aos estudantes formas distintas de compreender como programar. Entretanto, por não haver a descrição de experimentos relacionados à pesquisa que tenha levado os autores a mensurar a efetividade de cada uma delas, o artigo tem um caráter majoritariamente exploratório e descritivo.

Outro exemplo importante que possui o ensino em si como foco principal é o artigo “Metodologia ERM2C: Para melhoria do processo de ensino-aprendizagem de lógica de programação” (CAMPOS, 2010). Para o artigo foram entrevistados 60 estudantes de 2 duas turmas distintas e resultado na constatação de que a maior dificuldade dos estudantes era em construir um programa de computador do zero, capaz de resolver um problema previamente definido. Como forma de ensinar os estudantes de forma incremental a construção de programas é proposta a metodologia ERM2C (Entender, Revisar, Modificar, Complementar e Construir). A metodologia promove o ensino da programação através dos seguintes passos:

1. **Entender o código** – promove atividades para tornar o estudante capaz de compreender códigos funcionalmente corretos, escritos por terceiros;

2. **Revisar o código** – promove atividades para tornar o estudante capaz identificar erros em códigos funcionalmente incorretos, escritos por terceiros;
3. **Melhorar o código** – promove atividades para tornar o estudante capaz corrigir erros e melhorar o desempenho de códigos funcionalmente incorretos, escritos por terceiros;
4. **Complementar** – promove atividades para tornar o estudante capaz terminar a implementação de código funcionalmente incompletos e escritos por terceiros (terminar de implementar um código incompleto)
5. **Construir** – promove atividades para tornar o estudante capaz identificar e implementar códigos funcionalmente corretos a partir da especificação de um problema.

Segundo os autores do projeto, durante 4 semestres letivos, a metodologia ERM2C foi testada com um grupo de 88 estudantes voluntários em 4 IES. O resultado final dos experimentos revelou que o percentual de estudantes reprovados foi reduzido de 37% para 18% entre os estudantes que nunca haviam cursado os CCs e de 23% para 8% entre os estudantes repetentes.

Por fim, o projeto “*Stepwise self-explanation: Uma abordagem para o aprendizado de linguagens de programação através da auto-explicação de exemplos em vídeo*” (AURELIANO, 2016) foi desenvolvido durante um projeto de doutorado e propõe a utilização de um *framework* conhecido como *StepwiseImprovement*. Esta abordagem promove o ensino da programação de forma incremental e estruturada através de três dimensões: extensão, refinamento e reestruturação. Como forma de instanciar o *framework*, durante os experimentos foram utilizados vídeos como material de estudo para os estudantes, e exemplos de código com perguntas associadas a eles que ajudaram a entender os exemplos. Ao refletirem sobre as perguntas associadas aos exemplos, os estudantes realizavam um processo de auto-explicação. Neste processo o estudante explica para ele mesmo o que está aprendendo. Ao final dos experimentos realizados, apesar de não haver diferença estatística significativa entre as notas dos estudantes educados através do *framework*, e às notas dos estudantes que foram educados de forma tradicional, foi possível observar uma redução na quantidade de erros cometidos pelos estudantes educados através do *framework*.

Os três últimos exemplos relatam projetos onde o processo de ensino-aprendizagem da programação é colocado como foco principal da pesquisa. Cada um deles foi publicado em um espaço distinto. O primeiro (LIMA; MEIRELES, 2014) foi publicado em uma revista de baixo fator de impacto e que aceita artigos de várias áreas distintas, chamada Expressão Católica³. O segundo (CAMPOS, 2010) foi publicado no Workshop sobre Educação em Computação, um dos poucos eventos na área de TI que aceita artigos sobre educação na área. E, finalmente, o terceiro (AURELIANO, 2016) é uma tese de doutorado, que por definição é um trabalho único e inovador, o colabora com o que tem sido exposto nesta revisão bibliográfica de que projetos com foco no processo de ensino-aprendizagem em si é algo raro em nossa literatura. De forma geral, esses três exemplos são projetos relativamente raros, sendo encontrados após vários meses de pesquisas bibliográficas sobre o tema.

Compreendido o panorama presente na literatura brasileira, ainda resta explorar o panorama dos projetos sobre o tema da programação desenvolvidos fora do país. Assim, foi necessário expandir para um contexto internacional a revisão bibliográfica, visando desta forma, compreender como dá-se a discussão sobre o ensino da programação em Componentes Curriculares introdutórios ao tema para além dos projetos desenvolvidos nacionalmente.

No âmbito internacional, o projeto relatado pelo artigo *“Effective peer assessment for learning computer programming”* (SITTHIWORACHART e JOY, 2004) debate a importância da autoavaliação e propõe a utilização de uma ferramenta tecnológica para que os estudantes avaliem seus colegas de classe (seus pares) no ensino da programação.

Já o artigo *“Programming Pluralism: Using Learning Analytics to Detect Patterns in the Learning of Computer Programming”* (BLIKSTEIN et al., 2014) avalia a capacidade de algoritmos de inteligência artificial preverem o desempenho de estudantes no ensino da programação. Durante o projeto seus autores encontraram evidências de que é possível prever de forma relativamente precisa as notas dos estudantes nas avaliações a partir de códigos que os estudantes tenham escrito em máximo há uma semana antes da avaliação.

³<http://publicacoesacademicas.unicatolicaquixada.edu.br/index.php/rec/index>

Além disso, o artigo *“A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming”* (KAZIMOGLU et al., 2021) propõe um jogo sério como ponte para o desenvolvimento de habilidades conhecidas na literatura como *“Computational Thinking”*. Assim, as características das habilidades a serem desenvolvidas foram integradas ao gameplay do jogo, de forma que o estudante possa desenvolvê-las à medida que joga.

Em contrapartida, o artigo *“Causal attribution of success and failure made by undergraduate students in an introductory-level computer programming course”* (HAWI, 2010) foge do paradigma do uso de ferramentas para apresentar o resultado de entrevistas realizadas com estudantes de graduação, obtendo evidências indicando que os melhores estudantes de introdução à programação atribuem o próprio sucesso à estratégia de estudo.

Na mesma linha, de explorar o perfil dos estudantes, o artigo *“Attitude, Gender and Achievement in Computer Programming”* (ZUALKERNAN, 2006) apresenta os resultados de uma pesquisa que aplicou questionários com os estudantes e encontrou uma importante relação entre aqueles estudantes que têm uma atitude positiva sobre a programação e desempenho dos mesmos nos componentes curriculares introdutórios.

Ainda no âmbito do perfil dos estudantes, o artigo *“Examination of Undergraduate and Associate Degree Students' Computer Programming Attitude and Self-Efficacy According to Thinking Style, Gender and Experience”* (GUNBATAR, 2018) explora questões de estilo de pensamento sobre programação, gênero e experiência dos estudantes. Os resultados apontam para evidências de que tanto a experiência prévia sobre o tema, quanto a forma de pensar sobre o assunto têm influência positiva sobre o desempenho dos estudantes, entretanto não foi possível encontrar evidências sobre uma eventual influência relacionada ao gênero no resultado dos estudantes.

Já o artigo *“Previous Experience and the Learning of Computer Programming: The Computer Helps Those Who Help Themselves”* (KERSTEEN, 1988) aborda a questão da experiência prévia dos estudantes no uso de computadores para tarefas mais avançadas como forma de prever se o estudante terá um bom desempenho em CCs de introdução à programação. Ao final da pesquisa, os resultados apontam para a importância da experiência prévia no uso de computadores como forma de prever o sucesso dos estudantes.

Como último exemplo de projetos que exploram o perfil dos estudantes e como realizam seus estudos, o artigo *“Approaches to learning in computer programming students and their effect on success”* (DE RAADT et al., 2005) apresenta um estudo estatístico para avaliar a correlação entre o perfil dos estudantes e o sucesso dos mesmos. Os experimentos indicam que a forma que os estudantes abordam os componentes curriculares durante o semestre tem uma forte correlação com o desempenho positivo. O artigo relata especificamente, os estudantes que continuam estudando sobre o tema fora do horário de aula e se esforçam porque acham o assunto interessante possuem uma alta chance de ter sucesso durante o semestre.

Continuando a descrição da literatura estrangeira, é possível encontrar exemplos de projetos de pesquisa que abordam de forma mais direta as metodologias de ensino utilizadas durante o ensino da programação.

Por exemplo, o artigo *“Assessment frequency in introductory computer programming disciplines”* (BRITO e DE SÁ-SOARES, 2014) discute sobre como a frequência com que as avaliações são realizadas permitem aos estudantes perceberem mais rapidamente quando estão com dificuldades no aprendizado da programação.

E por fim, o artigo *“How Well Do Multiple Choice Tests Evaluate Student Understanding in Computer Programming Classes?”* (KUECHLER e SIMKIN, 2003) discute a aplicação de testes de múltipla escolha como forma de avaliar os estudantes em componentes curriculares de introdução à programação. Ao final do projeto os autores concluíram que, apesar desse tipo de avaliação ser útil em contextos de restrição de recursos para a realização de outras formas de avaliação, a aplicação de testes de múltipla escolha sozinhos não seria o ideal.

Como forma de prover uma visão mais ampla da pluralidade da revisão realizada, um resumo do tema principal dos artigos citados foi compilado no Quadro 2.2. O quadro mostra os artigos citados divididos em três categorias de acordo com seu tema predominante: Ferramentas Tecnológicas para o Ensino da Programação; Perfil dos Estudantes e Forma de Estudo; e Metodologia de Ensino.

Quadro 2.2 – Resumo do foco primário dos principais artigos estudados durante a revisão bibliográfica.

Foco primário	Autores	Desenvolvimento
Ferramentas Tecnológicas para o Ensino da Programação	Rocha et al. (2010)	Nacional
	de Souza (2009)	Nacional
	Rapkiewicz et al. (2006)	Nacional
	Moreira e Favero (2009)	Nacional
	Sitthiworachart e Joy (2004)	Internacional
	Blikstein et al. (2014)	Internacional
	Kazimoglu et al. (2021)	Internacional
Perfil dos estudantes e formas de estudo	Hawi (2010)	Internacional
	Zualkernan (2006)	Internacional
	Gunbatar (2018)	Internacional
	Kersteen (1988)	Internacional
	de Raadt et al. (2005)	Internacional
Metodologia de Ensino	Campos (2010)	Nacional
	Lima e Meireles (2014)	Nacional
	Aureliano (2016)	Nacional
	Brito e de Sá-Soares (2014)	Internacional
	Kuechler e Simkin (2003)	Internacional

A análise do Quadro 2.2 revela uma maior pluralidade dos temas dos artigos na literatura internacional, quando comparada à brasileira. Entretanto ainda é possível observar que mesmo neste contexto mais amplo, ainda há certa predominância de temas que não tenham a metodologia de ensino como foco, algo também observado em outros projetos que realizaram um processo análogo de revisão bibliográfica (AURELIANO, 2016, p. 14).

Assim, ao final desta revisão bibliográfica é possível observar que o tema de estudo deste projeto tem despertado pouco interesse de pesquisadores, especialmente os nacionais. Assim, em ambos os contextos, a revisão bibliográfica revela a incipiência do debate sobre metodologias de ensino no âmbito da programação, o que expõe o caráter inovador desta pesquisa.

3 METODOLOGIA

Este capítulo aborda a metodologia utilizada durante o desenvolvimento desta pesquisa na seguinte ordem: definição do campo e dos sujeitos de pesquisa; descrição do método utilizado para alcançar os objetivos do projeto; exposição das técnicas utilizadas dentro do método adotado; descrição dos instrumentos de coleta aplicados; e por fim, as diretrizes para análise dos resultados obtidos ao final do processo de coleta. Além disso, como a metodologia de coleta de dados e comparação entre as formas de ensino é o ponto centro deste projeto, este capítulo aborda durante todo o seu conteúdo as principais bases teóricas utilizadas para o desenvolvimento da metodologia de desenvolvimento da pesquisa.

3.1 CAMPO E SELEÇÃO DOS SUJEITOS

A delimitação exata do campo de pesquisa de um estudo é de fundamental importância para a sua adequada realização. Nesse sentido, é preciso adequar o escopo do projeto ao objetivo de compreender como é realizado o ensino da programação. Além disso, é importante observar que no contexto de uma especialização, há limitações no tempo de pesquisa e na experiência do pesquisador no âmbito das ciências humanas.

Dessa forma, visando adequar o escopo da pesquisa às limitações presentes, um conjunto de delimitações precisa ser traçado. A primeira delas vem do próprio ambiente da pesquisa, neste caso foi escolhido o próprio IFPE. Além disso, dentro do IFPE, ainda é preciso especificar de forma adequada as modalidades de ensino que serão estudadas. Assim, foram selecionados os cursos de bacharelado, de tecnologia (tecnólogos), além dos técnicos subsequentes. A seleção dessas modalidades vem da noção de que não há grandes divergências em termos de carga-horária e ementa no ensino da programação nelas. Em contrapartida, as modalidades PROEJA, médio integrado, pós-graduação e licenciaturas não serão contempladas nesse estudo, uma vez que possuem características peculiares que as diferenciam daquelas selecionadas para o estudo. Dessa forma, a inclusão delas levaria o escopo da pesquisa a estender-se para muito além daquilo que se considera adequado para uma especialização.

Assim, foi definido como espaço do campo de estudo os componentes curriculares presentes no Quadro 1.1. Porém, é importante observar mais uma vez que os professores do *Campus Igarassu* não serão incluídos na pesquisa, uma vez

que o professor que ministra os componentes curriculares de introdução a programação nos cursos do de Tecnologia da Informação do *Campus* é o autor da monografia.

Mesmo representando um potencial obstáculo (DESLANDES et al., 2002), a entrada no campo de estudo será tratada de forma natural. Isso é possível por conta da formação da equipe envolvida no projeto, tendo o primeiro autor como membro natural do campo (i.e., um professor de introdução à programação no IFPE).

Outro aspecto importante da pesquisa é a seleção dos sujeitos envolvidos na mesma. Como o foco do projeto é a compreensão de como dar-se o ensino da programação, há dois sujeitos principais envolvidos: os professores e os estudantes. Entretanto, como forma de viabilizar a pesquisa, o foco da abordagem será nos professores. Deixando assim, estudos relacionados ao processo de aprendizagem dos estudantes para futuros trabalhos na área.

Assim, dado o número relativamente pequeno de integrantes do conjunto dos sujeitos da pesquisa, para estudá-los foi adotada a técnica de coleta adotada de documentação direta intensiva através de entrevistas individuais (PRODANOV; FREITAS, 2013). Cada entrevista foi conduzida de forma estruturada nas primeiras etapas, porém também houve um espaço não estruturado ao fim. Assim, foi possível estabelecer um diálogo entre a parte estruturada e não estruturada, o que caracteriza o processo de coleta adotado como semi-estruturado (DESLANDES et al., 2002).

3.2 MÉTODO

Durante essa pesquisa o método indutivo foi aplicado, uma vez que o mesmo permite “chegar a um conhecimento mais generalizado” a partir “da constatação ou levantamento de informações particulares” (MAZUCATO, 2018).

Nesse sentido, tendo como ponto de partida da identificação dos cursos de TI do IFPE de níveis Técnico Subsequente e Superiores (Quadro 1.1), foi empregado um conjunto de técnicas predominantemente qualitativas (principalmente entrevistas) buscando identificar informações que possam ser generalizadas sobre a forma que a programação vem sendo ensinada nos componentes curriculares introdutórios dos cursos de TI do IFPE.

Mais especificamente, a pesquisa pretende comparar a forma que a programação foi ensinada aos professores do IFPE, e a forma que os mesmo a

ensinam aos seus estudantes. Nesse contexto, o método indutivo permitirá identificar os aspectos em comum e as diferenças entre ambas as formas de ensino. Assim, essa análise de caráter exploratório, também tem ambição de abrir espaços para discussões sobre os processos de ensino-aprendizagem dos professores de TI. Além disso, como o número de professores elegíveis para as entrevistas é reduzido, foi possível cobrir quase todos os membros do conjunto de professores que ensinam introdução à programação nos cursos de TI do IFPE.

3.3 TÉCNICAS

Uma vez estabelecido o método indutivo como aquele a ser utilizado durante a pesquisa, cabe agora relatar através de quais técnicas o método será desenvolvido no decorrer do projeto.

Sabendo que as técnicas de pesquisa *“representam um conjunto de procedimentos ou de processos intrínsecos a uma determinada área do conhecimento científico.”* (FONTANA, 2018). No caso dessa pesquisa, as seguintes técnicas foram adotadas: (1) documentação indireta através de pesquisa bibliográfica e revisão documental; (2) observação direta intensiva através de entrevistas:

1. Documentação indireta

- Pesquisa bibliográfica
 - Revisão da literatura acadêmica sobre as discussões realizadas no âmbito do ensino da programação (i.e., o objeto de análise deste projeto);
- Revisão Documental
 - Estudo dos projetos pedagógicos dos cursos de TI ofertados pelo IFPE, particularmente as ementas dos componentes curriculares de introdução à programação e as recomendações metodológicas para as aulas dos cursos.

2. Observação direta intensiva

- Realização de entrevistas individuais, no formato semi-estruturado, com os professores dos componentes curriculares de introdução à programação nos cursos de TI do IFPE.

A análise documental a ser desenvolvida envolve os projetos pedagógicos dos cursos de tecnologia da informação e as ementas dos componentes curriculares de introdução à programação dos cursos de tecnologia do IFPE.

Já a pesquisa bibliográfica a ser realizada tem como finalidade a identificação de projetos com escopo e temas similares, além da compreensão de como o ensino da programação é retratado e analisado dentro da literatura acadêmica. Nesse sentido, as principais fontes de consultadas envolveram os principais eventos e periódicos da área onde os projetos listados no Quadro 2.2 foram publicados, além do engenho de busca de artigos acadêmicos Google Acadêmico⁴.

Por fim, as entrevistas semi-estruturadas foram escolhidas como principal técnica de coleta de dados para pesquisa uma vez que:

“... combinam perguntas abertas e fechadas, onde o informante tem a possibilidade de discorrer sobre o tema proposto. O pesquisador deve seguir um conjunto de questões previamente definidas, mas ele o faz em um contexto muito semelhante ao de uma conversa informal.”
(BONI; QUARESMA, 2005)

Dessa forma, as entrevistas a serem realizadas durante a pesquisa permitiram compreender a relação entre a forma que os professores aprenderam a programar nas instituições de ensino que frequentaram e como eles ensinam seus estudantes a programar.

Para tal, os seguintes aspectos do ensino serão abordados durante as entrevistas:

1. Estratégias utilizadas para construção do conhecimento por parte do estudante durante as aulas, incluindo também a prática da programação;
2. Material didático produzido/utilizado;
3. Modelo de avaliação
4. Papel da monitoria;
5. Omnilateralidade (termo utilizado neste projeto no sentido de “formar o ser humano na sua integralidade física, mental, cultural, política, científico-tecnológica.” (CIAVATTA, 2014)).

⁴<http://scholar.google.com/>

3.4 INSTRUMENTOS DE COLETA

Para compreender como é realizado o ensino da programação de computadores no âmbito do IFPE, a principal técnica de coleta adotada foi a documentação direta intensiva através de entrevistas individuais.

Assim, visando caracterizar o processo como instrumento formal de coleta de dados, a abordagem será guiada pelos seguintes pontos (PRODANOV; FREITAS, 2013):

1. **Servir a um objetivo preestabelecido de pesquisa:** ao compreender o contexto onde o professor aprendeu a programar e como ele ensina programação para seus estudantes, o processo de coleta atua diretamente para colaborar com o objetivo da pesquisa;
2. **Ser planejada:** todo processo de coleta será devidamente premeditado, assim o modelo das entrevistas, as questões a serem abordadas, o processo de registro e a posterior análise foram definidos antes que se inicie qualquer forma de coleta;
3. **Ser registrada de forma sistemática:** nesse sentido, foi adotado um procedimento de registro digital das entrevistas que foram gravadas, posteriormente resumidas em relatórios, visando a organização dos dados para a etapa da análise e interpretação dos dados;
4. **Ser passível de verificação quanto ao seu grau de precisão:** como as entrevistas foram gravadas digitalmente, qualquer tipo de verificação poderá ser realizado com base nesse registro. Entretanto, por questões de ética e de privacidade, as gravações em si não serão publicadas para preservar os entrevistados. Dessa forma, qualquer tipo de acesso ao material gravado terá de contar com o consenso da equipe envolvida no projeto.

As entrevistas foram conduzidas com os professores do IFPE que ministram componentes curriculares de introdução da programação nos cursos presenciais superiores e técnicos subsequentes da área de tecnologia da informação, conforme o Quadro 1.1.

É importante relatar que os cursos do *Campus Igarassu* foram incluídos na pesquisa, uma vez que o professor que ministra os componentes curriculares de

introdução a programação nos cursos de Tecnologia da Informação do *Campus* é membro da equipe de pesquisa.

Além disso, não foram realizadas entrevistas de pré-teste uma vez que o espaço amostral é muito reduzido. Contudo, os projetos pedagógicos dos cursos, as ementas dos componentes curriculares e os currículos Lattes dos professores foram estudados antes das entrevistas.

O modelo de entrevista adotado é o semi-estruturado (DESLANDES et al., 2002), contendo perguntas pré-estabelecidas que foram utilizadas durante cada entrevista e um espaço livre de roteiro ao fim de forma que ambos os espaços possam dialogar entre si. As perguntas são predominantemente subjetivas e abertas. Assim, o roteiro retratado no Quadro 3.1 foi adotado para as entrevistas.

Quadro 3.1 – Roteiro das entrevistas.

1. Identificação do perfil do professor 1.1. Durante a graduação, qual estrutura você tinha à disposição para estudos em sua casa? 1.2. Você estudava em torno de quantas horas semanais fora do horário das aulas?
2. Aulas de programação na graduação 2.1. Como eram as aulas de introdução a programação na sua graduação? 2.2. Como era o material didático adotado? 2.3. Como era realizada a prática da programação? 2.4. Como eram feitas as avaliações? 2.5. Havia monitores ou equivalentes ajudando os professores e estudantes? 2.6. Havia algum tipo de discussão ou abordagem a respeito de aspectos como gênero, raça, religião, entre outros aspectos fora do contexto puramente técnico?
3. Aulas de programação no IFPE 3.1. Como são suas aulas de introdução a programação? 3.2. Qual o material didático adotado? 3.3. Como é realizada a prática da programação? 3.4. Como são feitas as avaliações? 3.5. Há monitores ou equivalentes ajudando os professores e estudantes? 3.6. Há algum tipo de discussão ou abordagem a respeito de aspectos como gênero, raça, religião, entre outros aspectos fora do contexto puramente técnico?
4. Reflexão sobre o perfil dos estudantes 4.1. Como você compararia o perfil dos seus colegas de classe na graduação ao perfil dos seus estudantes de introdução à programação no IFPE?
5. Parte 5 – Seção não estruturada

3.5 ANÁLISE

Uma vez estabelecida a entrevista como principal abordagem para coleta de dados do projeto, faz-se necessária a definição dos procedimentos de análise dos dados coletados.

Assim, nessa seção será definido como as informações qualitativas advindas das respostas às perguntas das entrevistas foram utilizadas para a criação dos relatórios das entrevistas; como os relatórios serão analisados; como a análise realizada levantará evidências que possam colaborar ou não com a hipótese da pesquisa (que pretende verificar se os professores estão replicando os métodos de ensino que tiveram contato no passado).

3.5.1 *Construção dos relatórios das entrevistas*

Os relatórios das entrevistas constituem o primeiro passo para a análise dos dados, nesse sentido eles atuam como agregadores das principais informações relevantes coletadas durante as entrevistas. Dessa forma, os relatórios das entrevistas foram compostos por três partes:

- 1) **Dados do perfil do professor:** contém as principais informações relacionadas ao perfil do professor como sua idade, local onde cursou a graduação, entre outras informações;
- 2) **Tabela comparativa ensino/aprendizagem:** principal etapa do relatório, feita através a construção de uma tabela contendo as informações sobre como o professor aprendeu a programar e como ele ensina;
- 3) **Observações:** observações gerais, especialmente com informações relevantes que venham da parte não estruturada da entrevista.

Tendo a tabela comparativa (como a principal etapa do relatório das entrevistas, a tabela (Quadro 3.2) como a principal etapa do relatório das entrevistas, é possível observar como suas linhas estão diretamente ligadas ao roteiro da entrevista introduzidos no Quadro 3.1, especialmente as etapas 2 e 3 do roteiro que buscaram compreender os processos de ensino-aprendizagem da graduação do entrevista e de suas aulas no IPFE. Além disso, as colunas da tabela foram preenchidas com um resumo das informações coletadas durante cada entrevista. Por exemplo, a linha referente ao “Material didático”, na coluna “Como

aprendeu” será preenchida com um resumo do que foi relatado pelo entrevistado sobre os materiais didáticos utilizados durante o seu estudo da programação em sua graduação. Enquanto isso, para a mesma linha, a coluna “Como ensina” reflete um resumo do que foi relatado durante a entrevista sobre os materiais didáticos utilizados pelo professor em suas aulas no IFPE.

Quadro 3.2– Modelo de tabela comparativa para o relatório da pesquisa.

Aspecto	Como aprendeu	Como Ensina
Aulas		
Material didático		
Processo avaliativo		
Monitoria		
Omnilateralidade		

Análise dos Dados

A análise dos dados é realizada em um contexto onde segundo Gio (2008):

“As respostas fornecidas pelos elementos pesquisados tendem a ser as mais variadas. Para que essas respostas possam ser adequadamente analisadas, torna-se necessário, portanto, organizá-las, o que é feito mediante o seu agrupamento em certo número de categorias.”

Assim, e de posse dos relatórios de entrevista, será realizada uma categorização da similaridade entre a forma que o professor aprendeu a programar e como ele ensina atualmente. Para tal, será introduzida uma codificação para a similaridade que será utilizada para classificar cada um dos aspectos do ensino estudados. Categoria será chamada de similaridade e representa uma variável quantitativa ordinal (RODRIGUES, 2011) dividida em três tipos distintos:

- 1) Baixa** – Quando não há ou há pouco em comum entre a forma que o professor aprendeu e a forma que ele ensina um determinado aspecto;
- 2) Média** – Quando há similaridades e diferenças entre a forma que os professores aprenderam e a forma que ensinam um determinado aspecto;
- 3) Alta** – Há muito em comum entre a forma que o professor aprendeu e a forma que ele ensina programação.

Uma vez estabelecidas as categorias de similaridade, é possível atualizar a tabela comparativa resultante do processo de construção dos relatórios das entrevistas incluindo uma nova coluna representando a similaridade entre as formas de aprendizado e ensino, representado assim a tabela de análise de similaridade aprendizagem-ensino (Quadro 3.3):

Quadro 3.3 – Modelo de tabela comparativa para o relatório da pesquisa.

Aspecto	Como aprendeu	Como ensina	Similaridade
Aulas			
Material didático			
Processo avaliativo			
Monitoria			
Omnilateralidade			

Uma vez compiladas as informações de similaridade como retratado na (Quadro 3.3), será possível explorar os resultados obtidos sob diversos caminhos distintos.

3.5.2 Diretrizes para interpretação e análise das entrevistas

Durante o processo de interpretação dos dados, procura-se “a obtenção de um sentido mais amplo para os dados analisado” (GIL, 2008). Neste sentido, esta seção explora as diretrizes para interpretações dos resultados das entrevistas.

A predominância de uma baixa similaridade entre os processos de ensino-aprendizado da programação é interpretada como evidência de que os professores estão utilizados processos de ensino independentes daqueles com os quais tiveram acesso durante suas graduações.

A predominância de uma média similaridade entre os processos de ensino-aprendizado da programação é interpretada como evidência de que os professores foram capazes de modificar os processos de ensino ao qual foram expostos introduzindo novas práticas que julguem serem importantes em suas aulas.

A predominância de uma alta similaridade entre os processos de ensino-aprendizado da programação é interpretada como evidência de que os professores têm replicado os métodos aos quais foram expostos durante suas graduações.

4 OS CURSOS DE TECNOLOGIA DA INFORMAÇÃO DO IFPE E SEUS COMPONENTES CURRICULARES DE INTRODUÇÃO A PROGRAMAÇÃO

Este capítulo aborda os programas dos CCs que foram estudados durante esta pesquisa, assim como exposto no Quadro 1.1. Nesse sentido, para cada um dos cursos de TI no IFPE, foram explorados: o perfil do profissional egresso de acordo com o Ministério da Educação; as recomendações metodológicas presentes nos projetos pedagógicos dos cursos; e o programa de cada CC. Ao final do capítulo também é realizada uma breve discussão sobre as similaridades entre os programas e a flexibilidade que eles permitem aos docentes do IFPE.

4.1 CURSO TÉCNICO EM INFORMÁTICA PARA INTERNET

O curso de Técnico em Informática para Internet possui uma carga horária mínima de 1000 horas e segundo o Catálogo Nacional dos Cursos Técnicos (REPÚBLICA FEDERATIVA DO BRASIL, 2021) forma um profissional habilitado para:

- Planejar e documentar aplicações para Web e dispositivos móveis.
- Desenvolver e organizar elementos estruturais e visuais de aplicações para Web e dispositivos móveis.
- Monitorar projetos de aplicações para Web e dispositivos móveis.
- Estruturar e implementar banco de dados para aplicações Web.
- Codificar aplicações para Web e dispositivos móveis.
- Publicar e testar aplicações para Web e dispositivos móveis.
- Documentar e realizar manutenção de aplicações para Web e dispositivos móveis

4.1.1 Campus Belo Jardim – Lógica de Programação 1

O projeto pedagógico do curso Técnico de Informática para Internet ofertado pelo Campus Belo Jardim (Ministério da Educação, 2014a) não possui seção dedicada exclusivamente às orientações metodológicas. Neste contexto, o seguinte programa foi elaborado para o componente curricular Lógica de Programação 1 que é responsável por introduzir o ensino da programação aos estudantes do primeiro semestre do curso:

Quadro 4.1 – Programa do Componente Curricular Lógica de Programação 1 do curso Técnico de Informática para Internet ofertado pelo *Campus* Belo Jardim.

Carga Horária 33.2 horas/relógio (não há informações sobre a divisão entre teoria e prática)
Competências • Aplicar as técnicas de programação estruturada codificando algoritmos em linguagens de programação de alto nível. • Visualizar técnicas de elaboração de algoritmo, analisando as funções de forma eficiente e otimizada. • Realizar operações usando álgebra de boole • Compreender e Aplicar estruturas de loop em soluções de problemas; • Aplicar estruturas de decisão. • Realizar manipulação de String em sistema computacional
Conteúdos Programáticos • Introdução à lógica de programação ○ Raciocínio lógico ○ Algoritmo ○ Sequências lógicas ○ Instruções lógicas ○ Álgebra booleana ○ Programas • Conceitos e implementação de algoritmos ○ Conceitos fundamentais ○ Tipos primitivos de dados ○ Memória, constantes e variáveis ○ Operadores aritméticos, lógicos e relacionais ○ Comandos básicos de atribuição e de entrada e saída de dados ○ Funções primitivas ○ Estruturas condicionais ○ Estruturas de repetição ○ Manipulação de String
Bibliografia Básica • KOTANI, Alice M.(et. al) Lógica de Programação – Os Primeiros Passos. Érica, 8ª ed. 1991. • FORBELLONE, André Luiz Villar (et. al.). Lógica de Programação – A Construção de Algoritmos e Estrutura de Dados. Makron Books, 2ªed. • FARRER, Harry. Algoritmos Estruturados. LCT, 2008. • PUGA, Sandra. Lógica de Programação e Estruturas de Dados. Pearson Prentice Hall, 2009.
Bibliografia Complementar

- CARVALHO, Sérgio E. R. de. Introdução à Programação com Pascal, Campus, Rio de Janeiro,
- 5ª ed. 1986.
- ASCENCIO, Ana Fernanda Gomes. Fundamentos da Programação de Computadores. Pearson
- Prentice Hall, 2007.
- FARRER, Harry et AL. Pascal Estruturado. LTC, 1999.
- MANZANO. José Augusto Navarro Garcia. Turbo Pascal. Érica, 1997.
- MANZANO, José Augusto Navarro Garcia. Linguagem C - Estudo Dirigido. Érica, 1997.
- FEOFILOFF, Paulo. Algoritmos em Linguagem C. Campus, Rio de Janeiro, 2009.
- ANSELMO, Fernando. Aplicando Lógica Orientada a Objeto em Java. Visual Books, 2005.
- MANZANO, José A N. G. Algoritmos – Lógica para Desenvolvimento de Programação. Érica,
- 7ªed, 1999.

4.1.2 Campus Igarassu – Lógica de Programação e Estruturas de Dados

O projeto pedagógico do curso Técnico de Informática para Internet ofertado pelo *Campus Igarassu* (Ministério da Educação, 2014b) relata no início de sua seção sobre as práticas metodológicas que:

“Os princípios pedagógicos, filosóficos e legais que subsidiam a organização pedagógica do curso são os definidos pelo MEC, nos quais a relação teoria-prática é o princípio fundamental à estrutura curricular, estando condizentes com as necessidades atuais nos diversos segmentos. O saber-pensar, o saber - fazer e o saber-ser devem ser os grandes norteadores do ensino-aprendizagem.”

Neste contexto, o seguinte programa foi elaborado para o componente curricular Lógica de Programação e Estruturas de Dados que é responsável por introduzir o ensino da programação aos estudantes do primeiro semestre do curso:

Quadro 4.2 – Programa do Componente Curricular Lógica de Programação do curso de TADS ofertado pelo *Campus Jaboatão dos Guararapes*.

Carga Horária
75 horas/relógio (não há informações sobre a divisão entre teoria e prática)
Competências
1. Utilizar uma linguagem escrita para construir algoritmos seguindo os preceitos da programação

estruturada.

2. Empregar estruturas de dados e módulos na resolução de problemas computacionais
3. Reconhecer técnicas de registro de informações em um sistema;
4. Selecionar adequadamente a técnica de algoritmos para otimização de um sistema;
5. Diferenciar as técnicas de armazenamento temporário de dados visando uma melhor adequação ao desempenho do hardware.
6. Construir a integração dos módulos (Sub-rotinas) desenvolvidos separadamente da ideia central do problema.

Conteúdos Programáticos

1. Conceitos básicos.
2. Identificadores e tipos de dados básicos.
3. Constantes e variáveis.
4. Operadores e prioridades dos operadores.
5. Comandos de atribuição, entrada e saída.
6. Estruturas de Seleção.
7. Estruturas de Repetição.
8. Estruturas homogêneas de dados (vetor e matriz)
9. Estruturas heterogênea de dados (registros)
10. Procedimentos
11. Escopo de nomes
12. Passagem de parâmetros
13. Funções
14. Biblioteca de rotinas
15. Recursividade
16. Estruturas de Dados
 - Filas
 - Pilhas
 - Listas Simples
 - Listas Ligadas
 - Algoritmos de Ordenação

Bibliografia Básica

CORMEN, Thomas H.; RIVEST, Ronald L.; STEIN, Clifford; LEISERSON, Charles E. Algoritmos: teoria e prática. 3.ed. São Paulo: Érica, 2012.

FARRER, Harry. Algoritmos Estruturados. Rio de Janeiro: LCT, 2008.

FERNANDA, Ana Gomes Ascencio. Fundamentos da Programação de Computadores. São Paulo: Pearson Prentice Hall, 2007.

FORBELLONE, André Luiz Villar. Lógica de programação: a construção de algoritmos e estrutura de dados. 3 ed. São Paulo: Brochura, 2005

Bibliografia Complementar

CARBONI, Irenice de Fátima. Lógica de Programação. São Paulo: Thomson, 2003.

SOARES, Márcio Vieira; GOMES, Marcelo Marques; Souza, Marco Antônio. Algoritmos e Lógica de Programação. 2. ed. São Paulo: Cengage Learning, 2011.

MANZANO, José Augusto N G; OLIVEIRA, Jayr Figueiredo de. Algoritmos: Lógica para Desenvolvimento de Programação de Computadores. 22 ed. São Paulo. Editora Érica, 2009

4.1.3 *Campus Jaboatão dos Guararapes – Lógica de Programação*

O projeto pedagógico do curso Técnico de Informática para Internet ofertado pelo *Campus Jaboatão dos Guararapes* (Ministério da Educação, 2018) relata no início de sua seção sobre as orientações metodológicas que:

“Da forma como é constituída a estrutura curricular do Curso de Técnico Subsequente em Informática para Internet, é exigida uma metodologia articulada com o objetivo de construir um modelo de ensino/aprendizagem que compreenda a aplicação dos princípios fundamentais ao exercício da profissão do Técnico em Informática para Internet, por meio de diversificadas práticas pedagógicas. Essa metodologia propõe ações interdisciplinares que promovem maior apreensão, por parte dos discentes, dos conteúdos ministrados bem como de sua aplicabilidade. Busca-se, assim, o desenvolvimento sociopolítico e cultural do estudante e de sua compreensão crítica da realidade, a fim de que seja capaz de aplicar seus conhecimentos e cooperar para o desenvolvimento da sociedade que o cerca e ser um profissional altamente qualificado.”

Neste contexto, o seguinte programa foi elaborado para o componente curricular Lógica de Programação que é responsável por introduzir o ensino da programação aos estudantes do primeiro semestre do curso:

Quadro 4.3 – Programa do Componente Curricular Lógica de Programação do curso de Informática para Internet ofertado pelo *Campus Jaboatão dos Guararapes*.

Carga Horária
75 horas/relógio (37.5 horas práticas, 37.5 horas práticas)
Ementa
Elementos essenciais de processamento de dados. Sistemas algébricos e relacionais. Álgebra

booleana. Conceitos de algoritmo, dado, variável, vetor, matriz, instrução e programa. Hierarquia lógica de informação (campos, registros, arquivos, organização, etc.). Algoritmo de Pesquisa e Ordenação. Estudos de caso.

Competências Desenvolvidas

1. Utilizar uma linguagem escrita para construir algoritmos seguindo os preceitos da programação estruturada.
2. Empregar estruturas de dados homogêneas e/ou heterogêneas e módulos na resolução de problemas computacionais
3. Reconhecer técnicas de registro de informações em um sistema;
4. Selecionar adequadamente a técnica de algoritmos para otimização de um sistema;
5. Diferenciar as técnicas de armazenamento temporário de dados visando uma melhor adequação ao desempenho do hardware.
6. Construir a integração dos módulos (Sub-rotinas) desenvolvidos separadamente da ideia central do problema.

Metodologia

Partir-se-á do pressuposto de que o processo de aprendizagem está ligado à significação do conteúdo, e que essa, necessariamente, implica em reflexão e investigação na vinculação entre a teoria e a prática. Esta abordagem dos conhecimentos privilegia os princípios da contextualização, da interdisciplinaridade e da indissociabilidade entre ensino, pesquisa e extensão, assegurando a construção das competências gerais. Para tanto, promover-se-ão situações de aprendizagem baseadas na participação e no desenvolvimento do poder crítico e criativo. Desta forma, propõem-se: exposições dialogadas; seminários; debates competitivos; estudos de caso; resolução de situação-problema; entre outros.

Avaliação

A disciplina terá duas (02) avaliações durante o período, a intervalos previamente programados, os quais devem expressar o resultado da verificação de aprendizado em cada intervalo, e eventual exame final, o professor, a seu critério e com aprovação da coordenadoria, pode promover trabalhos, exercícios e outras atividades em classe e extraclasse, que podem ser computados nas notas ou conceitos das verificações parciais, nos limites definidos pela Diretoria.

Conteúdo Programático

1. Conceitos básicos.
2. Identificadores e tipos de dados básicos.
3. Constantes e variáveis.
4. Operadores e prioridades dos operadores.
5. Comandos de atribuição, entrada e saída.
6. Estruturas de Seleção.
7. Estruturas de Repetição.
8. Estruturas homogêneas de dados (vetor e matriz)
9. Estruturas heterogênea de dados (registros)
10. Procedimentos

11. Escopo de nomes 12. Passagem de parâmetros 13. Funções
Bibliografia Básica 1. CORMEN, Thomas H.; RIVEST, Ronald L.; STEIN, Clifford; LEISERSON, Charles E. Algoritmos: teoria e prática. 3.ed. São Paulo: Érica, 2012. 2. FARRER, Harry. Algoritmos Estruturados. Rio de Janeiro: LCT, 2008. 3. SILVEIRA Paulo, ALMEIDA, Adriano: Lógica de Programação -Crie seus primeiros programas usando Javascript e HTML. São Paulo: Casa do Código, 2014
Bibliografia Complementar 1. CARBONI, Irenice de Fátima. Lógica de Programação. São Paulo: Thomson, 2003. 2. FERNANDA, Ana Gomes Ascencio. Fundamentos da Programação de Computadores. São Paulo: Pearson Prentice Hall, 2007. 3. FORBELLONE, André Luiz Villar. Lógica de programação: a construção de algoritmos e estrutura de dados. 3 ed. São Paulo: Brochura, 2005. 4. MANZANO, José Augusto N G; OLIVEIRA, Jayr Figueiredo de. Algoritmos: Lógica para Desenvolvimento de Programação de Computadores. 22 ed. São Paulo. Érica, 2009. 5. SOARES, Márcio Vieira; GOMES, Marcelo Marques; Souza, Marco Antônio. Algoritmos e Lógica de Programação. 2. ed. São Paulo: Cengage Learning, 2011.

4.1.4 Campus Palmares – Lógica de Programação

O projeto pedagógico do curso Técnico de Informática para Internet ofertado pelo *Campus Palmares* (Ministério da Educação, 2019a) relata no início de sua seção sobre as orientações metodológicas que:

“Os princípios pedagógicos, filosóficos e legais que subsidiam a organização pedagógica do curso são os definidos pelo MEC, nos quais a relação teoria-prática é o princípio fundamental à estrutura curricular, estando condizentes com as necessidades atuais nos diversos segmentos. O saber-pensar, o saber-fazer e o saber-ser devem ser os grandes norteadores do ensino-aprendizagem”

Neste contexto, o seguinte programa foi elaborado para o componente curricular Lógica de Programação que é responsável por introduzir o ensino da programação aos estudantes do primeiro semestre do curso:

Quadro 4.4 – Programa do Componente Curricular Lógica de Programação do curso Técnico em Informática para Internet ofertado pelo *Campus* Palmares.

Carga Horária 75 horas/relógio (37.5 horas práticas, 37.5 horas práticas)
Ementa Estudo dos elementos essenciais de processamento de dados. Sistemas algébricos e relacionais. Álgebra booleana. Conceitos de algoritmo, dado, variável, vetor, matriz, instrução e programa. Hierarquia lógica de informação (campos, registros, arquivos, organização, etc.). Algoritmo de Pesquisa e Ordenação. Estudos de caso.
Competências a Serem Desenvolvidas 1. Utilizar uma linguagem escrita para construir algoritmos seguindo os preceitos da programação estruturada. 2. Empregar estruturas de dados homogêneas e/ou heterogêneas e módulos na resolução de problemas computacionais 3. Reconhecer técnicas de registro de informações em um sistema; 4. Selecionar adequadamente a técnica de algoritmos para otimização de um sistema; 5. Diferenciar as técnicas de armazenamento temporário de dados visando uma melhor adequação ao desempenho do hardware. 6. Construir a integração dos módulos (Sub-rotinas) desenvolvidos separadamente da ideia central do problema
Metodologia A disciplina será ministrada utilizando-se recursos de exposições dialogadas, grupos de discussão, seminários, debates competitivos, apresentação e discussão de filmes e casos práticos, onde os conteúdos poderão ser trabalhados mais dinamicamente, estimulando o senso crítico e científico dos estudantes.
Avaliação A disciplina terá duas (02) avaliações durante o período, a intervalos previamente programados, os quais devem expressar o resultado da verificação de aprendizado em cada intervalo, e eventual exame final, o professor, a seu critério e com aprovação da coordenadoria, pode promover trabalhos, exercícios e outras atividades em classe e extraclasse, que podem ser computados nas notas ou conceitos das verificações parciais, nos limites definidos pela Diretoria
Conteúdo Programático 1. Conceitos básicos. 2. Identificadores e tipos de dados básicos. 3. Constantes e variáveis. 4. Operadores e prioridades dos operadores. 5. Comandos de atribuição, entrada e saída. 6. Estruturas de Seleção. 7. Estruturas de Repetição. 8. Estruturas homogêneas de dados (vetor e matriz)

9. Estruturas heterogênea de dados (registros) 10. Procedimentos 11. Escopo de nomes 12. Passagem de parâmetros 13. Funções
Bibliografia Básica 4. CORMEN, Thomas H.; RIVEST, Ronald L.; STEIN, Clifford; LEISERSON, Charles E. Algoritmos: teoria e prática. 3.ed. São Paulo: Érica, 2012. 5. FARRER, Harry. Algoritmos Estruturados. Rio de Janeiro: LCT, 2008. 6. SILVEIRA Paulo, ALMEIDA, Adriano: Lógica de Programação -Crie seus primeiros programas usando Javascript e HTML. São Paulo: Casa do Código, 2014
Bibliografia Complementar 6. CARBONI, Irenice de Fátima. Lógica de Programação. São Paulo: Thomson, 2003. 7. FERNANDA, Ana Gomes Ascencio. Fundamentos da Programação de Computadores. São Paulo: Pearson Prentice Hall, 2007. 8. FORBELLONE, André Luiz Villar. Lógica de programação: a construção de algoritmos e estrutura de dados. 3 ed. São Paulo: Brochura, 2005. 9. MANZANO, José Augusto N G; OLIVEIRA, Jayr Figueiredo de. Algoritmos: Lógica para Desenvolvimento de Programação de Computadores. 22 ed. São Paulo. Érica, 2009. 10. SOARES, Márcio Vieira; GOMES, Marcelo Marques; Souza, Marco Antônio. Algoritmos e Lógica de Programação. 2. ed. São Paulo: Cengage Learning, 2011.

4.2 CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS (TADS)

O curso de Tecnologia em Análise e Desenvolvimento de Sistemas possui uma carga horária mínima de 2000 horas e segundo o Catálogo Nacional dos Cursos Superiores de Tecnologia (REPÚBLICA FEDERATIVA DO BRASIL, 2016) forma um profissional com o seguinte perfil:

“Analisa, projeta, desenvolve, testa, implanta e mantém sistemas computacionais de informação. Avalia, seleciona, especifica e utiliza metodologias, tecnologias e ferramentas da Engenharia de Software, linguagens de programação e bancos de dados. Coordena equipes de produção de softwares. Vistoria, realiza perícia, avalia, emite laudo e parecer técnico em sua área de formação.”

4.2.1 Campus Recife – Introdução à Programação

O projeto pedagógico do curso de TADS ofertado pelo *Campus Recife* (Ministério da Educação, 2012) não possui seção dedicada às orientações metodológicas para o curso. Neste contexto, o seguinte programa foi elaborado para o componente curricular Lógica de Programação que é responsável por introduzir o ensino da programação aos estudantes do primeiro semestre do curso:

Quadro 4.5 – Programa do Componente Curricular Introdução à Programação do curso de TADS ofertado pelo *Campus Recife*.

Carga Horária
90 horas/relógio (não há informações sobre divisão entre teoria e prática)
Competências
Explicar as principais características de algoritmos e da programação de computadores usando o paradigma procedural.
Habilidades
• Diferenciar compiladores e Interpretadores, código fonte e código executável. • Caracterizar programação de computadores usando o paradigma procedural. • Empregar a sintaxe e a semântica de uma linguagem de programação C na construção de algoritmos. • Descrever a sequência de ações de um algoritmo utilizando diagrama de blocos. • Distinguir erros sintáticos e semânticos.
Bases Tecnológicas
1. Conceitos básicos de algoritmos. 2. Diagrama de blocos. 3. Conceitos básicos de linguagens de programação. 4. Constantes, variáveis e tipos de dados. 5. Estruturas condicionais. 6. Estruturas de repetição. 7. Funções. 8. Vetores e matrizes. 9. Ponteiros. 10. Alocação dinâmica de memória. 11. Estruturas e enumerações. 12. Arquivos.
Bibliografia Básica
• DEITEL, Paul; DEITEL, Harvey. Como Programar. 6ª ed. São Paulo, Pearson, 2011. • MIZRAHI, Viviane. Treinamento em Linguagem C. 2ª ed. São Paulo, Prentice-Hall, 2008. • MANZANO, José; OLIVEIRA, Jayr. Algoritmos: Lógica para Desenvolvimento de Programação de Computadores. 22ª ed. São Paulo, Érica 2009.
Bibliografia Complementar

- SCHILDT, Herbert. C Completo e Total. 3ª ed, São Paulo, Makron, 1997.
- DAMAS, Luis Manoel D. Linguagem C. 10ª ed. Rio de Janeiro, LTC, 2007.

4.2.2 *Campus Garanhuns – Lógica de Programação*

O projeto pedagógico do curso de TADS ofertado pelo *Campus Garanhuns* (Ministério da Educação, 2019b) relata em sua seção sobre as orientações metodológicas que:

“A linha metodológica proposta para o curso parte do pressuposto de que o estudante deve ser o agente de sua própria aprendizagem, garantindo-lhes assim autonomia no processo educacional”

Neste contexto, o seguinte programa foi elaborado para o componente curricular Lógica de Programação que é responsável por introduzir o ensino da programação aos estudantes do primeiro semestre do curso:

Quadro 4.6 – Programa do Componente Curricular Lógica de Programação do curso de TADS ofertado pelo *Campus Garanhuns*.

Carga Horária
120 horas/relogio (75 horas práticas, 45 horas práticas)
Ementa
Lógica aplicada à informática. Importância dos programas de computador para pessoas e organizações. Definições e exemplos de algoritmos. Conceitos básicos de algoritmos. Construção de algoritmos computacionais: estruturas de seleção, estruturas de repetição, procedimentos, funções e recursividade. Entrada e saída de dados. Compreensão sobre o processo de execução de programas no computador.
Objetivo
Compreender a importância dos programas de computador e aplicar os recursos necessários para a sua construção.
Metodologia
Aulas expositivas intercaladas com práticas de laboratório; Atividades em grupos; Uso de jogos educativos; Como também a adoção de outras estratégias consideradas relevantes pelo professor. As características deste componente curricular possibilitam práticas interdisciplinares com os componentes curriculares de Fundamentos da Informática, Lógica da Matemática e Filosofia, Fundamentos da Matemática e Inglês I.
Avaliação
O desempenho individual dos estudantes será mensurado por meio de atividades, como: lista de exercícios, provas e trabalhos em grupo.

Conteúdo Programático

1. Importância dos programas de computador na sociedade atual;
2. Relação entre programas de computador e o hardware;
3. Estratégias de resolução de problemas;
4. Definição de algoritmos computacionais;
5. Variáveis e tipos de dados;
6. Operadores de atribuição, aritméticos, relacionais, lógicos e incremento e decremento;
7. Comandos de entrada e saída;
8. Estruturas de controle de fluxo – condicionais (se, se-senão, se-senão-se-senão e caso)
9. Estruturas de controle de fluxo – repetições (para, enquanto e repita-enquanto)
10. Funções e passagem de parâmetros;
11. Recursividade;
12. Vetores;
13. Compreensão sobre os tipos de erros (sintaxe, lógica e execução);
14. Depuração de código.

Bibliografia Básica

1. PUGA, S; RISSETI, G. Lógica de Programação e Estrutura de Dados com Aplicações em Java, 2ª edição. São Paulo: ed. Pearson Prentice-Hall, 2008.
2. AGUILAR, L. Fundamentos de programação, algoritmos, estruturas de dados e objetos. São Paulo: McGraw-Hill, 2008.
3. CORMEN, T; LEISERSON, C. RIVEST, R., STEIN, C. Algoritmos: Teoria e prática. Rio De Janeiro: ed. Campus, 2002.

Bibliografia Complementar

1. SCHILDT, H. C Completo e Total. São Paulo: ed. Makron Books, 1997.
2. CORMEN, T. Desmistificando algoritmos. São Paulo: ed. Campus, 2014.
3. GRONER, L. Estruturas de dados e algoritmos em JavaScript. São Paulo: ed. Novatec, 2017.
4. FORBELLONE, L.; EBERSPACHER, H. Lógica de Programação: A Construção de Algoritmos e Estruturas de Dados. São Paulo: ed. Makron Books, 2005.
5. PIVA, D.; NAKAMITI, G.; ENGELBRETCH, A.; BIANCHI, F. Algoritmos e programação de computadores. São Paulo: ed. Campus, 2012

4.2.3 Campus Paulista – Introdução à Programação

O projeto pedagógico do curso de TADS ofertado pelo *Campus Paulista* (Ministério da Educação, 2019c) não possui orientações metodológicas de caráter genético, dedicando-se assim especificamente sobre o papel dimensões específicas do curso como pesquisa, extensão e a prática profissional.

Neste contexto, o seguinte programa foi elaborado para o componente curricular Introdução à Programação que é responsável por introduzir o ensino da programação aos estudantes do primeiro semestre do curso:

Quadro 4.7 – Programa do Componente Curricular Introdução à Programação do curso de TADS ofertado pelo *Campus Paulista*.

Carga Horária
80 horas/relógio (40 horas práticas, 40 horas práticas)
Ementa
Conceitos básicos de algoritmos; Diagrama de blocos; Conceitos básicos de linguagens de programação; Constantes, variáveis e tipos de dados simples e compostos; Estruturas condicionais; Estruturas de repetição; Funções; Recursão; Exceção, Módulos; Arquivos. Ferramentas de Programação (Debug, IDE, etc)
Competências a Serem Desenvolvidas
• Ter o primeiro contato com lógica de programação; • Aprender a utilizar um ambiente de desenvolvimento integrado; • Aprender as técnicas de depuração; • Aprender a utilizar uma linguagem de programação.
Metodologia
Partir-se-á do pressuposto de que o processo de aprendizagem está ligado à significação do conteúdo, e que está, necessariamente, implica em reflexão e investigação na vinculação entre a teoria e a prática, e que se promoverá situações de aprendizagem, baseadas na participação e no desenvolvimento do poder crítico e criativo, desta forma propõe-se:
• Aulas expositivas; • Seminários (texto, tema, artigos técnico/científico); • Estudo em grupo e individual; • Desenvolvimento de projetos
Avaliação
Prova escrita e prática, apresentação de seminários, trabalhos e projetos individuais ou em grupo, etc.
Conteúdo Programático
Programa
Algoritmo e Lógica de Programação
Linguagem de Programação
Ferramentas de Programação
Variáveis, expressões e instruções
Tipos de Dados Simples e Compostos
Strings
Listas
Dicionários

Tuplas Estruturas Condicionais Estruturas de Repetição Iteração Seleção de estrutura de dados Funções Módulos Exceção Arquivos Análise de Algoritmos
Bibliografia Básica 1. MENEZES, Nilo Ney Coutinho. Introdução à Programação com Python. 2. ed. São Paulo: Novatec, 2014. 2. SOUZA, Marco Antonio Furlan de; GOMES, Marcelo Marques; SOARES, Marcio Vieira. Algoritmos e Lógica de Programação 2. ed. São Paulo: Cengage Learning, 2011. 3. GARCIA, Guto, LOPES, Anita. Introdução à Programação. 1. ed. Rio de Janeiro: Campus, 2002.
Bibliografia Complementar 1. DOWNEY, Allen B. PenseemPython .1. ed. São Paulo: Novatec/O'Reilly, 2016. 2. WAZLAWICK, Raul. Introdução aos algoritmos e programação com Pytho. 1. Rio de Janeiro: Elsevier, 2017. 3. SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. ed. Porto Alegre: Bookman, 2018. 4. MARJI, Majed. Aprenda a programar com Scratch: uma introdução visual à programação com jogos, arte, ciência e matemática. 1. São Paulo: Novatec, 2014. 5. MANZANO, José Augusto N. G. Introdução à linguagem Python. 1. São Paulo: Novatec, 2018.

4.2.4 Campus Jaboatão dos Guararapes – Lógica de Programação

O projeto pedagógico do curso de TADS ofertado pelo *Campus Jaboatão* (Ministério da Educação, 2020a) relata no início de sua seção sobre as orientações metodológicas que:

“Da forma como é constituída a estrutura curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, é exigida uma metodologia articulada com o objetivo de construir um modelo de ensino/aprendizagem que compreenda a aplicação dos princípios fundamentais ao exercício da profissão de Tecnólogo em Análise e

Desenvolvimento de Sistemas, por meio de diversificadas práticas pedagógicas. Essa metodologia propõe ações interdisciplinares que promovem maior apreensão, por parte dos discentes, dos conteúdos ministrados bem como de sua aplicabilidade. Busca-se, assim, o desenvolvimento sociopolítico e cultural do estudante e de sua compreensão crítica da realidade, a fim de que seja capaz de aplicar seus conhecimentos e cooperar para o desenvolvimento da sociedade que o cerca e ser um profissional altamente qualificado.”

Neste contexto, o seguinte programa foi elaborado para o componente curricular Lógica de Programação que é responsável por introduzir o ensino da programação aos estudantes do primeiro semestre do curso:

Quadro 4.8 – Programa do Componente Curricular Lógica de Programação do curso de TADS ofertado pelo *Campus* Jaboatão dos Guararapes.

Carga Horária
100 horas/relógio (50 horas práticas, 50 horas práticas)
Ementa
Introdução a Algoritmos; Identificadores e tipos de dados básicos. Constantes, variáveis e operadores. Estruturas de controle. Estruturas de decisão e repetição; Procedimentos e funções; recursividade.
Objetivos
• Compreender os principais conceitos da área de testes de software; • Entender os diferentes tipos de testes e os diferentes processos; • Compreender as métricas de testes e o processo de projeto e avaliação de testes; • Conhecer as principais técnicas e ferramentas para automação de testes; • Utilizar uma linguagem escrita para construir algoritmos seguindo os preceitos da programação estruturada. • Empregar estruturas de dados e períodos na resolução de problemas computacionais • Reconhecer técnicas de registro de informações em um sistema; • Selecionar adequadamente a técnica de algoritmos para otimização de um sistema; • Diferenciar as técnicas de armazenamento temporário de dados visando uma melhor adequação ao desempenho do hardware. • Construir a integração dos períodos (Sub-rotinas) desenvolvidos separadamente da idéia central do problema.

Metodologia

Partir-se-á do pressuposto de que o processo de aprendizagem está ligado à significação do conteúdo, e que essa, necessariamente, implica em reflexão e investigação na vinculação entre a teoria e a prática. Esta abordagem dos conhecimentos privilegia os princípios da contextualização, da interdisciplinaridade e da indissociabilidade entre ensino, pesquisa e extensão, assegurando a construção das competências gerais. Para tanto, promover-se-ão situações de aprendizagem baseadas na participação e no desenvolvimento do poder crítico e criativo. Desta forma, propõem-se: exposições dialogadas; seminários; debates competitivos; estudos de caso; resolução de situação-problema; entre outros

Avaliação

A disciplina terá duas (02) avaliações durante o período, a intervalos previamente programados, os quais devem expressar o resultado da verificação de aprendizado em cada intervalo, e eventual exame final, o professor, a seu critério e com aprovação da coordenadoria, pode promover trabalhos, exercícios e outras atividades em classe e extraclasse, que podem ser computados nas notas ou conceitos das verificações parciais, nos limites definidos pela Diretoria

Conteúdo Programático

1. Conceitos básicos.
2. Identificadores e tipos de dados básicos.
3. Constantes e variáveis.
4. Operadores e prioridades dos operadores.
5. Comandos de atribuição, entrada e saída.
6. Estruturas de Seleção.
7. Estruturas de Repetição.
8. Procedimentos
9. Escopo de nomes
10. Passagem de parâmetros
11. Funções
12. Biblioteca de rotinas
13. Recursividade

Bibliografia Básica

1. SILVEIRA, Paulo; ALMEIDA, Adriano. Lógica de Programação: Crie seus primeiros programas usando JavaScript e HTML. São Paulo: Casa do Código. 2016.
2. CORMEN, Thomas H.; RIVEST, Ronald L.; STEIN, Clifford; LEISERSON, Charles E. Algoritmos: teoria e prática. 1.ed. São Paulo: Elsevier, 2012.
3. FORBELLONE, Andre Luiz Vilar; EBERSPÄCHER, Henri Frederico. Lógica de Programação: Construção de Algoritmos e Estruturas de Dados. 3ª ed. São Paulo: Pearson, 2010.

Bibliografia Complementar

1. VARELA, Helton; PEVIANI, Claudia Tinós. Scratch Um jeito divertido de aprender programação. São Paulo: Casa do Código, 2017.

2. MENEZES, Nilo Ney Coutinho. Introdução à Programação com Python. 2ª ed. São Paulo: Novatec. 2014.
3. CRUZ, Felipe. Python: Escreva seus primeiros programas. São Paulo: Casa do Código. 2015.
4. FORBELLONE, Andre Luiz Vilar; EBERSPÄCHER, Henri Frederico. Lógica de Programação: Construção de Algoritmos e Estruturas de Dados. São Paulo: Pearson, 2010.
5. NIEDERAUER, Juliano. Desenvolvendo Websites com PHP. São Paulo: Novatec. 2016.

4.3 CURSO DE BACHARELADO EM ENGENHARIA DE SOFTWARE

O curso de Bacharelado em Engenharia de Software possui uma carga horária mínima de 3200 horas, sendo normatizado pela Resolução nº 5, de 16 de Novembro de 2016 de Ministério da Educação (2016) do Ministério da Educação, que instrui as Instituições de Ensino Superior a instituírem as Diretrizes Curriculares Nacionais para os cursos de graduação na área da Computação, abrangendo os cursos de bacharelado em Ciência da Computação, em Sistemas de Informação, em Engenharia de Computação, em Engenharia de Software e de licenciatura em Computação, e dá outras providências.

Com relação ao perfil do profissional egresso o curso, a normativa relata que:

“§ 3º Levando em consideração a flexibilidade necessária para atender domínios diversificados de aplicação e as vocações institucionais, espera-se dos egressos dos cursos de Engenharia de Software que:

I - 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;

II - sejam capazes de criar soluções, individualmente ou em equipe, para problemas complexos caracterizados por relações entre domínios de conhecimento e de aplicação;

III - sejam capazes de agir de forma reflexiva na construção de software,

compreendendo o seu impacto direto ou indireto sobre as pessoas e a sociedade;

IV - entendam o contexto social no qual a construção de Software é praticada, bem como os efeitos dos projetos de software na sociedade;

V - compreendam os aspectos econômicos e financeiros, associados a novos produtos e organizações;

VI - reconheçam o caráter fundamental da inovação e da criatividade e compreendam as perspectivas de negócios e oportunidades relevantes.”

4.3.1 *Campus Belo Jardim – Introdução à Programação*

O projeto pedagógico do curso de TADS ofertado pelo *Campus Belo Jardim* (Ministério da Educação, 2020b) relata no início de sua seção sobre as orientações metodológicas que:

“Da forma como é constituída a estrutura curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, é exigida uma metodologia articulada com o objetivo de construir um modelo de ensino/aprendizagem que compreenda a aplicação dos princípios fundamentais ao exercício da profissão de Tecnólogo em Análise e Desenvolvimento de Sistemas, por meio de diversificadas práticas pedagógicas. Essa metodologia propõe ações interdisciplinares que promovem maior apreensão, por parte dos discentes, dos conteúdos ministrados bem como de sua aplicabilidade. Busca-se, assim, o desenvolvimento sociopolítico e cultural do estudante e de sua compreensão crítica da realidade, a fim de que seja capaz de aplicar seus conhecimentos e cooperar para o desenvolvimento da sociedade que o cerca e ser um profissional altamente qualificado.”

Neste contexto, o seguinte programa foi elaborado para o componente curricular Lógica de Programação que é responsável por introduzir o ensino da programação aos estudantes do primeiro semestre do curso:

Quadro 4.9 – Programa do Componente Curricular Lógica de Programação do curso de Engenharia de Software ofertado pelo *Campus* Belo Jardim.

Carga Horária
90 horas/relógio (40 horas práticas, 50 horas práticas)
Ementa
Reconhecimento de técnicas de registro de informações em arquivos. Construção e integração dos módulos (Subrotinas). Aplicação de técnicas de programação estruturada. Codificação de algoritmos em linguagens de programação de alto nível. Técnicas de elaboração de algoritmo. Operações usando álgebra de boole. Estruturas de loop em soluções de problemas. Estruturas de decisão. Manipulação de String
Objetivo Geral
Proporcionar através da lógica computacional a habilidades de desenvolver e interpretar sistemas computacionais
Objetivos Específicos:
• Desenvolver o raciocínio lógico; • Compreender operações lógicas; • Desenvolver estruturas em sequência lógica • Compreender metodologias de armazenamento em arquivo texto
Metodologia
As aulas serão ministradas de forma intercalada alternando teoria e prática. Será utilizado laboratório de informática, projetor de multimídia e linguagem de programação
Avaliação
A avaliação será realizada constantemente no dia a dia dos alunos. Utilizaremos as aulas práticas para realização das avaliações. Somada a esta avaliação ainda teremos a prova teórica e prática. Que podem ser feitas em papel de forma tradicional ou em sistema computacional.
Conteúdo Programático
• Introdução à Lógica de Programação ○ Raciocínio Lógico ○ Algoritmo ○ Diagrama de blocos ○ Sequências lógicas ○ Instruções lógicas ○ Álgebra booleana • Conceitos e implementação de algoritmos ○ Conceitos fundamentais ○ Tipos primitivos de dados

- Memória, constantes e variáveis
- Operadores aritméticos, lógicos e relacionais
- Comandos básicos de atribuição e de entrada e saída de dados
- Funções primitivas
- Estruturas condicionais
- Estruturas de repetição
- Manipulação de String
- Tipos e estrutura básica de dados
 - Estruturada de dados homogêneas I
 - Matrizes de uma dimensão ou vetores
 - Operações básicas com matrizes do tipo vetor
 - Estrutura de dados homogêneas II
 - Matriz com mais de uma dimensão
 - Operações básicas com matrizes de uma ou mais dimensão
 - Estrutura de dados heterogêneas
 - Programação Modular
 - Procedimentos e funções
 - Variáveis globais e locais
 - Método Top-Down
 - Passagem de parâmetros por valor e referência
- Algoritmos de ordenação
- Pesquisa seqüencial e binária
- Recursividade

Bibliografia Básica

ASCENCIO, Ana F. G.; CAMPOS, E. A. V. de. Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ e Java. 2. ed. São Paulo, SP: Pearson, 2010.

MANZANO, José Augusto N. G. Algoritmos: Lógica para Desenvolvimento de Programação, 1. ed. São Paulo: Érica, 1999.

MANZANO, José Augusto N. Garcia. Linguagem C: estudo Dirigido, Editora: Érica, Edição: 1997.

Bibliografia Complementar

KOTANI, Alice M. et. al. Lógica de Programação: os primeiros passos. 8. ed. São Paulo: Érica, 1991.

AVILLANO, Israel de C. Algoritmos e Pascal. 2 Ed. Rio de Janeiro: Ciência Moderna, 2006.

FARRER, Harry. Algoritmos Estruturado. 3 ed. Rio de Janeiro: LCT, 2008.

SILVA, Camila C. da. Lógica de Programação. Santa Cruz do Rio Pardo SP: Viena, 2007.

ANSELMO, F.. Aplicando Lógica Orientada a Objeto em Java. 2 ed. Florianópolis: V. Books, 2005

4.4 CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET (TSI)

O curso de Tecnologia em Sistemas para Internet possui uma carga horária mínima de 2000 horas e segundo o Catálogo Nacional dos Cursos Superiores de

Tecnologia (REPÚBLICA FEDERATIVA DO BRASIL, 2016) forma um profissional com o seguinte perfil:

“Projeta, desenvolve, testa, implanta, mantém, avalia e analisa páginas para sites de Internet e intranets, sistemas de comércio eletrônico e aplicativos para plataformas móveis para a Internet. Avalia, especifica, seleciona e utiliza metodologias e ferramentas adequadas para o desenvolvimento das aplicações.”

4.4.1 *Campus Igarassu – Programação Imperativa*

O projeto pedagógico do curso de TSI ofertado pelo *Campus Igarassu* (Ministério da Educação, 2021) relata no início de sua seção sobre as orientações metodológicas que:

“Da forma como é constituída a estrutura curricular do Curso de Tecnologia em Sistemas para Internet, é exigida uma metodologia articulada com o objetivo de construir um modelo de ensino/aprendizagem que compreenda a aplicação dos princípios fundamentais ao exercício da profissão do Tecnólogo em Sistemas para a Internet, por meio de diversificadas práticas pedagógicas. Essa metodologia propõe ações interdisciplinares que promovem maior apreensão, por parte dos discentes, dos conteúdos ministrados bem como de sua aplicabilidade. Busca-se, assim, o desenvolvimento sociopolítico e cultural do estudante e de sua compreensão crítica da realidade, a fim de que seja capaz de aplicar seus conhecimentos e cooperar para o desenvolvimento da sociedade que o cerca e ser um profissional altamente qualificado.”

Neste contexto, o seguinte programa foi elaborado para o componente curricular Programação Imperativa que é responsável por introduzir o ensino da programação aos estudantes do primeiro semestre do curso:

Quadro 4.10 – Programa do Componente Curricular Programação Imperativa do curso de TSI ofertado pelo *Campus Igarassu*.

Carga Horária
60 horas/relógio (30 horas práticas, 30 horas práticas)
Ementa
Estudo dos principais conceitos de lógica de programação no paradigma imperativo e suas aplicações para solução de problemas em uma linguagem de propósito geral, utilizada no mercado e alinhada com as necessidades do curso.
Objetivos
1. Compreender a programação de computadores; 2. Utilizar uma linguagem de programação para a construção de algoritmos seguindo o paradigma de programação imperativo; 3. Empregar os conceitos básicos da programação imperativa na solução de problemas computacionais; 4. Aplicar técnicas de registro de informações (tipos compostos) e módulos (sub-programas) na resolução de problemas computacionais
Metodologia
• Aula expositiva dialogada com uso de multimídia e/ou registro em quadro; • Aulas práticas em laboratório para implementação e testes dos algoritmos; • Exercícios em grupo durante as aulas; • Visitas técnicas; • Projetos práticos de desenvolvimento de software para web.
Avaliação
• Exercícios avaliativos individuais com questões teóricas e práticas; • Exercícios avaliativos em grupo; • Implementação e apresentação de projetos de software em grupo.
Conteúdo Programático
1. Introdução à Programação 1.1. Conceitos de Algoritmo e Lógica 1.2. Conceitos de Linguagens de programação 1.3. Variáveis Locais e Variáveis Globais 1.4. Memória 1.5. Arquitetura de von Neumann 2. Tipos de Dados 2.1. Tipos Primitivos 2.2. Tipos Estruturados

3. Expressões
 - 3.1. Operadores Lógicos
 - 3.2. Operadores Relacionais
4. Execução Condicional
 - 4.1. Decisão simples: SE-ENTÃO;
 - 4.2. Decisão composta: SE-ENTÃO-SENÃO;
 - 4.3. Decisão Múltipla: ESCOLHA-CASO;
 - 4.4. Encadeamento de estruturas de decisão.
5. Estruturas de Repetição
 - 5.1. Repetição Condicional
 - 5.2. Repetição Contada
 - 5.3. Aninhamento de estruturas de repetição
6. Subprogramas
 - 6.1. Procedimentos
 - 6.2. Funções
 - 6.3. Passagem de parâmetros por valor e por referência
7. Bibliotecas
8. Tipos Compostos
 - 8.1. Vetores
 - 8.2. Matrizes
 - 8.3. Registros
 - 8.4. Enumerações
9. Recursão

Bibliografia Básica

MANZANO, Jose Augusto N. G.; OLIVEIRA, Jayr Figueiredo de Oliveira. Algoritmos - Lógica para Desenvolvimento de Programação de Computadores. 16.ed. São Paulo: Érica, 2014.

SOUZA, Marco Antonio Furlan de; GOMES, Marcelo Marques; SOARES, Marcio Vieira. CONCÍLIO, Ricardo. Algoritmos e Lógica de Programação. Um texto introdutório para Engenharia. 2.ed. rev. e ampl. São Paulo: Cengage Learning, 2014.

FARRER, Harry et al. Algoritmos Estruturados. 3. ed, Rio de Janeiro: LCT, 2015.

Bibliografia Complementar

FLANAGAN, D. Javascript - O Guia Definitivo. 6. ed, Porto Alegre: Bookman, 2013.

SCHILDT, Herbert. Java para Iniciantes. 6.ed. Porto Alegre: Bookman, 2015.

SUMMERFIELD, Marc. Programação em Python 3: uma introdução completa à linguagem Python. Rio de Janeiro: Alta Books, 2013.

LOCKHART, Josh. PHP Moderno Novos Recursos e Boas Práticas. São Paulo: Novatec, 2015.

SCHILDT, Herbert; SKRIEN, Dale. Programação com Java: uma introdução abrangente. AMGH:, 2013.

4.5 Discussão

Inicialmente, o que se percebe ao estudar os programas dos CCs de introdução à programação dos cursos de TI do IFPE, é uma variedade de formatos entre eles. Há seções que estão presentes em alguns dos programas, mas não se fazem presentes em outros, como a seção sobre metodologia do componente. Tudo isso dificulta uma comparação mais profunda, mesmo assim ainda é possível identificar alguns padrões recorrentes nos programas.

Em relação às aulas, há uma variedade de ações propostas como aulas expositivas, aulas práticas de programação e resolução de problemas. Já no âmbito das avaliações também é possível encontrar vários diversos presentes nos programas como avaliações tradicionais e projetos de desenvolvimento de software. Além disso, em relação às ementas e conteúdo programático, comumente os programas vão desde o conceito de algoritmos até a criação e manipulação de tipos compostos como *arrays* e registros. Há planos que vão mais além introduzindo temas como estruturas de dados ou ferramentas de *debugging*. Por fim, as bibliografias contêm os livros que normalmente serão utilizados para o estudo inicial da linguagem a ser adotada pelo curso. Por exemplo, se o curso adota a linguagem de programação C como primeira linguagem, sua bibliografia é majoritariamente composta por livros sobre ela. Entretanto, também é comum haver nas ementas livros sobre temas mais avançados, como algoritmos e estruturas de dados, uma vez que eles podem ajudar os estudantes que desejem ir além do conteúdo original descrito na ementa do componente curricular.

De forma geral, os programas dos CCs estudados descrevem de forma direta e clara o que deve ser estudado durante o semestre, seus conteúdos programáticos abordam um conjunto mínimo de assuntos relacionados ao tema da introdução à programação e suas avaliações dão opções aos professores para realizar abordagens tradicionais como provas ou mais práticas como projeto. Assim, de forma geral, os programas dão flexibilidade aos professores trazerem processos de ensino-aprendizagem que julguem adequados para os CCs que irão ministrar, algo que será abordado detalhadamente no próximo capítulo.

5 RESULTADOS DAS ENTREVISTAS E DISCUSSÕES

Este capítulo explora como os professores do IFPE têm enfrentado os desafios do ensino da programação em suas aulas. Neste contexto, são apresentados os relatos de cinco entrevistas realizadas com professores do IFPE. Esses relatos contemplam assim cinco dos seis *Campi* do IFPE incluídos no escopo das entrevistas dentro deste projeto de pesquisa do projeto.

Como forma de proteger o anonimato dos docentes do IFPE entrevistados, todas as informações relacionadas ao perfil dos mesmos (como nomes, *Campus* de lotação e cursos em que ministram suas aulas) foram omitidas dos relatos das entrevistas. Além disso, o termo “docente” foi utilizado preferencialmente, por ser uma palavra desprovida de gênero.

As entrevistas seguiram o roteiro descrito no capítulo 3 - METODOLOGIA desta monografia. Desta forma, durante as entrevistas foram abordadas como foi o processo de ensino-aprendizagem utilizado durante a graduação dos professores além do processo de ensino-aprendizagem utilizado pelos mesmos em suas aulas de introdução à programação no IFPE.

O relato de cada entrevista envolve uma descrição textual dos principais aspectos explorados durante as entrevistas; uma tabela comparativa que resume as práticas adotadas quando os professores do IFPE estavam cursando componentes curriculares de introdução à programação em suas graduações e as práticas que eles adotam em suas aulas sobre o mesmo tema atualmente. Por fim, é feita uma análise da similaridade entre o processo de ensino-aprendizagem que eles foram expostos e aquele que eles desenvolvem em seus *Campi*.

Ao final do capítulo é realizada uma discussão mais ampla, envolvendo as informações obtidas nas cinco entrevistas sobre a influência de como os professores aprenderam a programar e como eles ensinam atualmente.

5.1 ENTREVISTA #1

5.1.1 *Trajetória profissional e acadêmica*

A entrevista de número 1 foi realizada com um(a) docente que iniciou sua graduação em TI no final dos anos 2000, iniciou logo em seguida um mestrado na mesma área em meados dos anos 2010 e que recentemente iniciou seu doutorado, também em TI. Além da experiência acadêmica, o(a) docente possui experiência no

mercado de TI, atuando como analista de sistemas até o ano de ingresso no IFPE como professor.

5.1.2 Estrutura e estudos durante a graduação

Durante o início da sua graduação, o(a) docente dispunha de um espaço de estudos com mesa própria, computador, e internet, que apesar de instável, era de banda larga. Por conta do suporte familiar, foi possível dedicar-se exclusivamente à graduação, como relatado: “Eu conseguia, graças ao apoio familiar, me dedicar só à graduação”.

Outro aspecto importante dos estudos do docente durante a graduação era a constante colaboração entre colegas de classe para a realização do aprendizado, algo que foi relatado da seguinte forma: “Me ajudou ter um grupo de amigos que cada um era bom em uma coisa, então a gente se reunia fora do horário das aulas para estudar”.

5.1.3 Comparativo entre processo de ensino-aprendizado na graduação e no IFPE

Durante sua graduação, o(a) docente teve seu primeiro contato com a programação através de aulas expositivas e teóricas com apresentação de slides. Os slides também continham exemplos de código, porém eles não eram executados durante as aulas. Também não havia prática no horário das aulas, ficando esse tipo de atividades reservadas aos monitores. Já em suas aulas o(a) docente também realiza aulas expositivas com slides, porém com criação e execução de programas no computador durante as mesmas. O quadro branco é um recurso utilizado frequentemente, e ao final das aulas são realizadas sessões de prática com os estudantes nos computadores do laboratório onde as aulas são realizadas.

Em relação ao material didático utilizado, em sua graduação eram utilizados majoritariamente slides e livros didáticos. Enquanto que em suas aulas, o(a) docente utiliza slides, livros e indica vídeos feitos por terceiros no YouTube relacionados aos assuntos de suas aulas.

No âmbito avaliativo, durante sua graduação foram utilizados minitestos como forma de avaliação continuada além de provas, ambos feitos em papel. Também era possível receber uma pontuação extra através da participação nas aulas de monitoria, que ocorriam fora do horário de aula. Já em suas aulas ministradas no IFPE, o(a) docente realizou experimentos de correção automática de códigos e de pontuação extra para participação nas monitorias, porém sem sucesso. Sendo

assim, o modelo de avaliação utilizado atualmente envolve múltiplos minitestes, provas ao final da primeira unidade, um projeto na segunda unidade.

No âmbito do papel dos monitores, na graduação do(a) docente eles tiveram um papel considerado como fundamental para assimilação do conteúdo do CC de introdução à programação. Havia um grupo de aproximadamente 10 monitores que realizava encontros três vezes por semana com os estudantes no contraturno das aulas. Além disso, a frequência nas aulas de monitoria resultava em pontos extras para os estudantes. Para o(a) docente, a monitoria foi descrita como um recurso importante conforme relatado na entrevista: “Uma coisa que eu usei muito na graduação era a monitoria”.

A importância dos monitores na metodologia de ensino da graduação do(a) docente era algo relatado pelo professor que ministrava a matéria no início do semestre. Na primeira aula do CC, eram apresentados dados comparativos entre o desempenho dos estudantes que participavam da monitoria e daqueles que não participavam. Não obstante, no semestre seguinte o(a) docente acabou tornando-se monitor do CC de introdução à programação. Entretanto, em suas aulas no IFPE, o(a) docente não dá mesma ênfase no recurso da monitoria, apesar de também usá-los para realização de práticas de programação complementares em horários fora das aulas.

Tendo o perfil dos estudantes em vista, o(a) docente relatou que seus colegas de classe na universidade eram muitas vezes advindos das classes sociais mais altas, porém mesmo assim “A gente não tinha nenhuma diferença de trato entre a gente”.

Porém, a percepção do perfil dos estudantes do IFPE por parte do(a) docente é bem diferente, uma vez que foi exposto que os estudantes do IFPE são advindos de classes sociais menos favorecidas e moradores da periferia da região metropolitana do Recife, o que aparenta gerar a uma reação empatia pelo(a) docente: “O que eu fico feliz é trazer o curso para perto deles (estudantes), agora eles não precisam, como eu (precisava), pegar dois ou três ônibus para estudar”.

Em relação aos aspectos omnilaterais do ensino, tanto na sua graduação, quanto em suas aulas, o(a) docente relatou não haver discussões para além da programação. Entretanto, o(a) docente informou que ensina aos estudantes como realizar tarefas do cotidiano da programação que raramente são incluídas em ementas de CCs introdutórios como *debugging* de código e controle de versão.

5.1.4 Tabela Comparativa do Processo Ensino-Aprendizagem

Quadro 5.1 – Tabela comparativa do processo ensino-aprendizagem da graduação do(a) docente #1.

Aspecto	Graduação	IFPE	Similaridade
Aulas	Aulas expositivas, sem prática ou execução de exemplos (o professor não programava durante a aula).	Expositivas, com criação e execução de programas, além de sessões de prática ao fim das aulas (exercícios), usa o quadro.	Média.
Material Didático Adotado	Slides, livros.	Slides, adota mais de um livro, indica vídeos no YouTube para os estudantes.	Alta.
Avaliação	Minitestes e prova (no papel), presença nas monitorias.	Minitestes, prova (no computador), presença nas monitorias e projeto (em grupo), tentou realizar correção automática de exercícios, porém sem sucesso.	Média.
Monitoria	Vários monitores, prática realizadas pelos monitores, 3 dias por semana.	Sim, quando há candidatos, prática realizada pelos monitores apesar de não haver muitos monitores.	Média.
Omnilateralidade	Não.	Não, mas ensina ferramentas de <i>debug</i> e controle de versão.	Alta.

5.1.5 Análise da Entrevista

Durante a entrevista foi possível observar que há um conjunto de similaridades entre a forma que o(a) docente aprendeu a programar e a forma que ele(a) ensina atualmente no IFPE especialmente no processo de avaliação. Assim, a utilização de minitestest, provas e os experimentos com notas para frequência em aulas de monitoria são exemplos importantes dessas semelhanças. Entretanto, também foi possível observar que a forma que as aulas são ministradas foram alteradas para incluir testes com códigos durante a parte expositiva, além de práticas no laboratório ao final das aulas. Dado o exposto, o processo de ensino utilizado atualmente pelo(a) docente no IFPE pode ser considerado com uma forma, segundo as convicções do(a) próprio(a), refinada do processo utilizado durante sua graduação. Por fim, é importante ressaltar que o(a) docente compreende que há uma diferença no perfil social dos seus colegas de graduação em comparação aos estudantes do IFPE, mas ao mesmo tempo, não faz uma relação dessa diferença com o desempenho acadêmico dos estudantes, e ainda informou que tanto os

estudantes da sua graduação quando os do IFPE não possuem dificuldades para empregar-se no mercado de trabalho.

5.2 ENTREVISTA #2

5.2.1 *Trajatória profissional e acadêmica*

A entrevista de número 2 foi realizada com um(a) docente que fez a graduação em TI, iniciada no meio dos anos 2000, além de possuir aperfeiçoamento, mestrado e doutorado na mesma área realizados durante a década de 2010.

Ainda durante o mestrado, o(a) docente iniciou sua carreira profissional como professor atuando em uma escola técnica local, posteriormente trabalhou em instituições de ensino públicas até tornar-se professor do IFPE.

5.2.2 *Estrutura e estudos durante a graduação*

Durante sua graduação, o(a) docente pode contar com uma estrutura em sua casa que incluía uma sala de estudos, com biblioteca, computador e acesso à internet banda larga.

5.2.3 *Comparativo entre processo de ensino-aprendizado na graduação e no IFPE*

O primeiro contato do(a) docente com a programação, foi realizado de forma autodidata, ainda no ensino médio, quando o(a) docente inscreveu-se para participar de uma competição de programação. Mas o assunto só foi aprofundado, para além dos conceitos mais fundamentais, durante a graduação, onde foi possível estudar formalmente o tema.

Em sua graduação, as aulas eram expositivas, elas utilizavam uma lista de exercícios como ponto de partida, uma vez que o professor realizava a programação de soluções para os problemas das listas durante a aula. Os exercícios que não fossem respondidos durante a aula, eram implementados posteriormente pelos estudantes. Durante a aula, ainda havia momentos onde os estudantes eram consultados sobre as listas de exercícios e convidados a escrever as respostas que implementaram no quadro. Em contrapartida, em sua aula, o(a) docente utiliza um conjunto de ferramentas tecnológicas como o Google Colab⁵ onde guarda o material de suas aulas. Através dessa ferramenta o professor expõe brevemente a parte

⁵<https://colab.research.google.com>

teórica da aula para em ainda, ainda utilizando a ferramenta, começar a programar respostas para sua própria lista de exercícios. Além das aulas expositivas, o professor também realiza competições entre os estudantes que são separados em grupos durante a competição. As práticas distintas utilizadas na aula do(a) docente refletem uma visão sobre a pluralidade das formas de ensino onde “O professor deve apresentar o mesmo assunto de forma diferentes, fazer uns desafios.”.

Do ponto de vista do material didático adotado, durante sua graduação, o material era basicamente composto pelas listas de exercícios produzidas pelo professor, que não adotava livros ou mesmo criava slides. Porém, dentro do IFPE o(a) docente utiliza o Google Colab para armazenar a parte teórica de suas aulas e os exemplos de código, além disso também é indicado um livro-texto para os estudantes.

No âmbito da avaliação, o(a) docente foi exposto a um processo baseado em provas individuais e um projeto interdisciplinar (em conjunto com o CC de arquitetura de computadores) em pequenos grupos. Por outro lado, dentro do IFPE as práticas que o(a) docente implementa são baseadas em diversas avaliações como empenho medida através de ferramentas como o ClassDojo⁶, pontos extras são atribuídos aos estudantes que resolverem problemas no portal Code Wars⁷. Também há dinâmicas onde os estudantes que são separados em grupos e têm que inserir erros os programas de outros grupos, além de um projeto que é guiado por uma lista de funcionalidades que devem ser implementadas. Por fim, o do inglês nas atividades é utilizado como forma de atribuir pontos extras aos estudantes.

Durante a graduação do(a) docente, o papel da monitoria foi classificado como importante uma vez que os estudantes monitores, segundo o(a) próprio(a) docente: “ajudavam muito”. De forma análoga, os monitores também são utilizados no IFPE como forma de auxiliar os estudantes fora do período das aulas, uma vez por semana.

Ainda segundo o(a) docente, há uma grande diferença entre os colegas de sala da graduação e os estudantes do IFPE. Sendo dos primeiros de classes sociais mais privilegiadas, enquanto os estudantes do IFPE enfrentam graves problemas sociais, inclusive sendo membros de famílias com poucos anos de escolaridade,

⁶<https://www.classdojo.com>

⁷<https://www.codewars.com>

como retratado pelo seguinte trecho da entrevista: “No IFPE tem estudante que metade da família não lê.”.

Além disso, segundo o(a) docente, há uma importante diferença entre os colegas de graduação e os estudantes do IFPE: a evasão. No caso dos colegas de graduação, havia retenção e reprovação nos CC, porém os estudantes raramente desistiam do curso. Enquanto isso, no IFPE, além da retenção e da reprovação, a evasão nas primeiras semanas do curso ocorre frequentemente,

No âmbito da educação omnilateral, foi retratado na entrevista que durante a graduação o professor do(a) docente abordava frequentemente o tema do empreendedorismo, sendo inclusive o professor mais interessado neste tema do que na própria programação, como relatado na entrevista: “Professor João (nome fictício) tinha outra cabeça, ele era empreendedor, então a disciplina era pequena para ele... Ele não gostava muito da disciplina não”.

O tema do empreendedorismo é abordado nas aulas do(a) docente, que relatou ser “completamente a favor da escola sem partido”, porém relatou realizar diversas intervenções sobre os mais diversos temas durante sua aula, como política e religião.

Por fim, durante a fase não estruturada da entrevista o docente relatou que, parte importante dos estudantes que desiste na primeira semana do curso, por não se identificarem com a programação; e que os índices de evasão são relativamente baixos, após as desistências das primeiras semanas, mas que o elevado número de reprovações ao final do semestre representa um problema.

5.2.4 Tabela Comparativa do Processo Ensino-Aprendizagem

Quadro 5.2 – Tabela comparativa do processo ensino-aprendizagem da graduação do(a) docente #2.

Aspecto	Graduação	IFPE	Similaridade
Aulas	Aula expositiva, prática da programação no turbo pascal, o professor desenvolvia os programas de uma lista de exercícios durante a aula, e os estudantes faziam os demais exercícios fora da aula, às vezes os estudantes tinham que escrever as respostas no quadro.	Aula expositiva, prática da programação através da ferramenta Colab do Google, a parte teórica e os exemplos de código ficam no Colab, seguida de listas para estudantes fazerem individualmente ao final da aula. Em algumas aulas há competições entre os estudantes que são separados em grupos	Média.
Material Didático	Não havia slides, nem livros.	Não há slides, conteúdo teórico é armazenado no Colab, mas há a indicação de um livro sobre a linguagem adotada.	Baixa.
Avaliação	Provas e projeto interdisciplinar (colaboração com o CC Arquitetura de Computadores).	Nota de empenho (usando ferramentas como ClassDojo e Code Wars), projeto com uma <i>checklist</i> para os estudantes alcançarem, notas extras para quem usar inglês, dinâmica como estudantes trocando código e inserindo <i>bugs</i> para correção nos códigos trocados.	Baixa.
Monitoria	Sim, fora do horário das aulas, considerada de muita ajuda.	Sim, uma vez por semana, fora do horário das aulas.	Alta.
Omnilateralidade	Sim, eram abordadas questões sobre empreendedorismo.	Sim, empreendedorismo e outros temas. Incentivo ao uso do Inglês.	Média.

5.2.5 Análise da Entrevista

A entrevista foi marcada por um conjunto de similaridades pontuais entre a forma que o professor aprendeu a programar na graduação e a forma que ele ensina dentro do IFPE. Porém, essas similaridades funcionam mais como inspiração e ponto de partida para adaptação para o método empregado no IFPE. Por exemplo, nas aulas o(a) docente resolve problemas de uma lista, assim como era na sua graduação, porém suas aulas incluem outras atividades e utilizam ferramentas tecnológicas que não foram empregadas na sua graduação. Outro exemplo é a forma que o(a) docente passa o conteúdo para o estudante: abrindo o Colab e programando exercícios de sua lista. Também há aspectos que são completamente diferentes, como as avaliações. Ou seja, após a análise da entrevista, é possível

compreender que embora o método de ensino utilizado dentro IFPE tenha influência do tempo de graduação do(a) docente, essas influências representam aspectos pontuais, de forma que o método utilizado no IFPE possui sua própria originalidade

5.3 ENTREVISTA #3

5.3.1 *Trajetória profissional e acadêmica*

A entrevista de número 4 foi conduzida com um(a) docente que possui dupla graduação, em engenharia, ao fim dos anos 80, e posteriormente em um curso de tecnologia da informação, em meados da década de 90. O(a) docente também possui mestrado e doutorado, ambos ingressados durante os anos 2000. Como o docente informou ter conseguido dispensa dos CCs curriculares de introdução à programação da sua segunda graduação, as práticas abordadas durante tiveram como foco principal a primeira graduação.

A atuação profissional na área de TI do(a) docente foi iniciada próximo ao fim da primeira graduação e o começo da segunda. Já ao final da segunda graduação, foi iniciada a atuação como docente de CC componentes curriculares relacionados à programação.

5.3.2 *Estrutura e estudos durante a graduação*

Advindo de uma família de origens sociais humildes, o(a) docente não dispunha de uma estrutura de ensino apta para a programação em sua casa, especialmente por não possuir computador durante o período em que aprendeu a programar na universidade. Sendo assim, os estudos sobre a programação eram realizados majoritariamente dentro da universidade.

5.3.3 *Comparativo entre processo de ensino-aprendizado na graduação e no IFPE*

O contato com a programação do(a) docente foi realizado ainda na primeira graduação através de aulas expositivas inicialmente utilizando um pseudocódigo e posteriormente usando a linguagem FORTRAN. As aulas eram ministradas em um quadro negro com o conteúdo escrito em giz. Para testar seus programas, os estudantes tinham que reservar e utilizar um mainframe, o que significava que eles tinham que escrever o código com antecedência e realizar testes de mesa (manuais e sem nenhum auxílio computacional). Os códigos eram entregues a um funcionário da universidade, que no horário reservado, os executava no mainframe. Os resultados da execução podiam ser retirados no fim do dia ou no dia seguinte. Em

contrapartida, as aulas ministradas pelo(a) docente no IFPE são realizadas dentro de um laboratório de informática, utilizando slides, e na linguagem de programação C, sendo majoritariamente expositivas.

O material didático utilizado durante a graduação reflete as limitações do tempo, sendo raros os livros disponíveis sobre o tema, o que tornava o próprio professor com a principal fonte de conhecimento para os estudantes. Além dos livros, também eram utilizados cartões de referência, contendo resumos da sintaxe e dos principais da linguagem estudada. Já no IFPE, o(a) docente utiliza slides próprios com exemplos de código para promover o ensino da programação.

Em termos avaliativos, a metodologia aplicada durante a graduação era baseada em provas no papel que eram realizadas ao final das unidades letivas. Por outro lado, no IFPE o(a) docente emprega uma metodologia de avaliação continuada, com pequenas avaliações ao final de cada assunto (com intervalo entre uma e duas semanas). Nessas avaliações, os estudantes utilizam um modelo de código disponibilizado para resolver um problema de sua escolha, dentre aqueles disponibilizados previamente em uma lista. Além das avaliações curtas também há uma prova no laboratório, ao final da unidade, e um projeto de software que é guiado e orientado pelo próprio professor.

No contexto da monitoria, tanto na graduação do(a) docente, quanto no IFPE havia monitores, porém em ambos os casos, eles possuem um papel secundário, sendo pouco consultados pelos estudantes. E particularmente no IFPE, os monitores têm sido usados majoritariamente para auxiliar o professor e não os estudantes.

Segundo o(a) docente, há relevante diferença no perfil dos estudantes das turmas da sua graduação e aqueles do IFPE: a evasão. No caso dos estudantes da graduação, havia retenção e reprovação nos CC, porém os estudantes raramente desistem do curso. Enquanto isso, no IFPE, além da retenção e da reprovação, há uma significativa evasão nas primeiras semanas do curso, como retratado pelo seguinte trecho da entrevista: “Os estudantes entram para experimentar o curso e na primeira pressão abandonam.”.

Com relação à omnilateralidade do ensino, tanto na graduação do(a) docente, quanto no IFPE, os CCs são puramente técnicos, não sendo abordados temas que não tenham relação direta com a programação.

Por fim, durante a parte não estruturada da entrevista foi possível explorar alguns aspectos importantes da forma que o(a) docente compreende o ensino da

programação dentro do IFPE, conforme relatado na seguinte frase: “A gente tem que escolher se a gente quer ser um curso inclusivo, ou um curso de excelência.”.

Nesse sentido, foi possível observar um relato sobre as reflexões pessoais do(a) docente sobre um modelo de ensino inclusivo (na visão do(a) docente, um ensino que se esforça para reprovar menos estudantes, mas que reduz a qualidade técnica do programador) e um ensino de excelência (na visão do(a) docente, um modelo que leva a maiores índices de reprovação, retenção e evasão, mas que ao mesmo tempo é forma um profissional mais apto ao sucesso no mercado de trabalho).

5.3.4 Tabela Comparativa do Processo Ensino-Aprendizagem

Quadro 5.3 – Tabela comparativa do processo ensino-aprendizagem da graduação do(a) docente #3.

Aspecto	Primeira Graduação	IFPE	Similaridade
Aulas	Começou com pseudocódigo depois FORTRAN, giz e quadro, prática no mainframe, resultados ao final do dia, o tempo de execução no terminal era alocado. Testes eram feitos manualmente (testes de mesa).	Expositivas com código em C e dentro de laboratórios.	Baixa.
Material Didático	Poucos livros, cartão de referência, principal fonte de informações era o professor (não havia internet disponível na época).	Material próprio, slides e exemplos.	Baixa.
Avaliação	Prova no papel.	Avaliação continuada, pequenas avaliações ao final de cada capítulo (entre uma ou duas semanas). Projeto guiado/orientado pelo professor. Deixa o estudante usar modelos códigos e escolher um problema para resolver. Há uma avaliação ao final da unidade. Recuperação continuada.	Baixa.
Monitoria	Sim, mas não foi usada pelo professor, os estudantes trabalhavam em grupos para estudar.	Sim, mas os estudantes não usam frequentemente. Os monitores têm auxiliado mais o professor do que os estudantes.	Baixa.
Omnilateralidade	Não era explorado nenhum aspecto Omnilateral.	Não era explorado nenhum aspecto Omnilateral.	Alta.

5.3.5 *Análise da Entrevista*

Durante a entrevista, o(a) docente relatou que quando começou a ensinar CCs de introdução à programação, ainda na rede privada, utilizou como ponto de partida as aulas que teve durante a graduação. Porém, a análise da entrevista, demonstra uma distância temporal relevante entre a época que o(a) docente aprendeu a programar e os tempos atuais. De forma mais concreta, o primeiro contato com a programação foi realizado na era dos mainframes que tinham seus tempos de execução compartilhados entre os membros do *Campus*. Já o ensino no IFPE é realizado em tempos onde os computadores são amplamente acessíveis dentro da instituição. Assim, mesmo que no começo da sua vida como docente houvesse uma influência do processo de ensino-aprendizagem da graduação, a diferença de eras, faz com que o ensino por parte do(a) docente seja naturalmente muito distinto atualmente dentro do IFPE.

5.4 ENTREVISTA #4

Durante a entrevista #4 possível observar que o processo de aprendizagem da programação do(a) docente foi dividido em duas etapas. A primeira, dentro de um curso técnico de nível médio integrado ao ensino médio, e a segunda já em sua graduação. Dessa forma, visando preservar a metodologia da pesquisa, porém sem ignorar a importância primeira etapa do aprendizado, ambas as etapas serão descritas, porém a tabela comparativa foi dividida em duas: uma para a graduação (Quadro 5.4) e outra para o curso técnico (Quadro 5.5).

5.4.1 *Trajetória profissional e acadêmica*

A entrevista de número 4 foi conduzida com um(a) docente que teve seu primeiro contato com a programação no meio da década de noventa através de um curso técnico (integrado ao ensino médio). Sua graduação foi iniciada no começo dos anos 2000, seguida de uma especialização iniciada na mesma década e um mestrado iniciado no começo dos anos 2010.

Sua vida profissional se dividia entre o trabalho na área de TI em instituições públicas e o ensino em instituições de ensino superior privadas. As primeiras experiências com docente foram conduzidas em CCs da área de banco de dados e, posteriormente, também nos CCs de introdução à programação.

5.4.2 *Estrutura e estudos durante a graduação*

Durante sua graduação, o(a) docente teve que distanciar-se da casa dos seus pais, indo estudar em outro estado e residindo em uma pensão. Apesar do contexto, o(a) docente possuía em sua moradia um computador para realizar seus estudos. De forma geral, no início da graduação o(a) docente realizou um grande esforço para alcançar as melhores notas possíveis, uma vez que desejava reduzir a dependência dos pais, conforme relatado na entrevista da seguinte forma: “Eu queria voar (tirar notas boas) para ter minha bolsa, porque eu era estudante fora minha cidade, gastando o dinheiro do meu pai e da minha mãe”.

5.4.3 *Comparativo entre processo de ensino-aprendizado na graduação e no IFPE*

Tanto no durante o curso técnico, quanto durante a graduação, as aulas de introdução à programação ministradas pelos professores eram expositivas e foram classificadas como “tradicionais” pelo(a) docente por incluírem a realização de prática em laboratórios por parte dos estudantes, conforme relatado durante a entrevista: “Professor falando, escrevendo no quadro e você anotando no caderno”.

No curso técnico, como os estudantes eram mais novos, foram empregados exemplos classificados pelo(a) docente como “lúdicos”. Neles, os estudantes tinham que desenvolver algoritmos para soluções de problemas como, por exemplo, sair de um labirinto). Apesar dessa abordagem, o(a) docente enfrentou muitas dificuldades durante o aprendizado da programação, algo que relatou da seguinte forma durante a entrevista: “Foi bem sofrido (o primeiro contato com a programação), aulas inteiras sem entender nada, sem entender nem o que eu estava fazendo ali.”.

Já na graduação, os problemas eram mais contextualizados trabalhando temas como, por exemplo, folhas de pagamento. Além disso, na graduação além do CC de introdução à programação também havia um CC dedicado exclusivamente à prática da programação em laboratórios.

Dentro do IFPE, as aulas ministradas trazem também exemplos lúdicos, fora do computador, sem programação real e incluindo somente algoritmos). Os exemplos lúdicos são usados no primeiro encontro sobre um novo assunto, seguido de exemplos contextualizados voltados para a programação nas aulas posteriores. A aplicação de ambas as técnicas foi justificada pela própria experiência do(a) docente no aprendizado da programação: “O que não fizesse sentido, eu não tinha um viés de aprendizado” e “Eu nunca mandei um aluno computar um algoritmo para

computar a sequência de Fibonacci (uma lista abstrata de números, sem aplicação óbvia)”.

Nas aulas do IFPE, ainda há uma separação entre as aulas semanais que ocorrem em encontros de 3 horas/aula seguido de encontros de 1 hora/aula. No encontro de 3 horas/aula é realizada uma aula tradicional, enquanto que no segundo são realizadas pequenas avaliações. Também são realizados desafios onde os estudantes devem assistir um vídeo e implementar um programa de desafio com base nele, ou utilizando ferramentas de torneio de questões como o Kahoot!⁸

Do ponto de vista do material didático, tanto no curso técnico, como na graduação, os livros eram as principais referências além do conteúdo passado pelo professor durante as aulas. Por outro lado, em suas aulas, o(a) docente além dos livros, adota materiais físicos usados nas aulas lúdicas, slides, e a indicação de materiais presentes na internet como vídeos criados por terceiros.

Em relação à avaliação dos estudantes, tanto no curso técnico, quanto na graduação, as primeiras avaliações foram realizadas no papel, com uma posterior transição para os códigos no computador. Entretanto no IFPE o modelo de avaliação envolve as seguintes atividades: prova (algo que o(a) docente relatou estar considerando não realizar nos próximos semestres, projeto em grupo e avaliações curtas (nas aulas de 1 hora/aula), especificamente sobre estas avaliações curtas o(a) docente relatou sobre a dificuldade em realizá-las: “Avaliação contínua ... é muito boa, mas ela traz um overhead muito grande (de trabalho na correção).”.

No curso técnico e na graduação do(a) docente havia estudantes monitores colaborando com o aprendizado e desenvolvendo atividades majoritariamente fora do horário de aula. Já no IFPE, os monitores são empregados em atividades extraclasse. Porém, foi relatada uma dificuldade no preenchimento das vagas para bolsas de monitoria.

No contexto do perfil dos estudantes, o(a) docente relatou que seus colegas de turma durante o período de graduação eram elitizados, sem a presença de estudantes advindos de escolas públicas uma vez que “Na minha época não existiam cotas”. Em contrapartida, o perfil dos estudantes do IFPE foi relatado como completamente diferente através de um detalhado depoimento sobre esse contexto:

⁸<https://kahoot.com/>

“87% dos nossos alunos recebem um auxílio ... que fazem a diferença para uma família que tem uma renda per capita inferior a um salário mínimo. Então assim, se eu tenho 87 pessoas que estão ali recebendo 100 reais que é a bolsa, é porque eles têm necessidades que para mim e para qualquer outro nível de alunos já teriam supridas. Então eu deixei um pouco de ter o pré-conceito literal que é simplesmente ‘não quer estudar’, pra tentar entender que o meio onde você está não te favorece a ter uma mente aberta.” Docente #4.

Além disso, quando a comparação entre os perfis foi realizada, o(a) docente fez uma crítica direta ao modelo de ensino tradicional realizado nas universidades, informando que em sua visão:

“Eu acho que o te vira da universidade federal fornece um conjunto de características que faz o aluno diferenciado em diversos aspectos, mas ele segrega demais... a gente forma muito pouco (estudantes)” Docente #4.

Por fim, no âmbito dos aspectos omnilaterais foi abordado que durante o ensino médio não havia discussões que transcendessem a programação, mas que na graduação, o professor aborda questões como empreendedorismo e orientação profissional. Nesse contexto, a atuação do(a) docente dentro do IFPE e em suas aulas de programação abordam a questão da presença das mulheres no mercado de TI, de tal forma que o(a) docente possui um projeto na instituição voltado para o tema, o que teria mudado a visão do contexto social das estudantes do *Campus*, uma vez que “O projeto (de extensão) ... me colocou muito próximo das meninas.”.

5.4.4 Tabela Comparativa do Processo Ensino-Aprendizagem

Quadro 5.4 – Tabela comparativa do processo ensino-aprendizagem da graduação do(a) docente #4.

Aspecto	Graduação	IFPE	Similaridade
Aulas	Aula expositiva sem práticas, mas com problemas contextualizados. A prática era feita em outras matérias (docente sentiu-se encantado(a)).	Aulas com práticas com problemas contextualizados, exemplos lúdicos (exemplo: formar labirintos na sala para os estudantes criarem algoritmos para a saída, uso de ferramentas como Kahoot!). Separação das aulas semanais em encontros de 3 e 1 hora/aula. As aulas de uma hora muitas vezes eram usadas para avaliações. O primeiro encontro sobre um assunto normalmente não é com uma linguagem de programação. Há desafios onde os estudantes veem um vídeo e fazem um desafio relacionado ao vídeo.	Média.
Material Didático	Livro.	Materiais lúdicos, livros, slides, indicação de cursos e vídeos de terceiros no YouTube, e sites na internet.	Baixa.
Avaliação	Início no papel, prova no papel, laboratório: prático, envio por e-mail.	Prova (não pretende fazer mais), exercícios, projeto, avaliação contínua, algumas avaliações são feitas no papel.	Baixa.
Monitoria	Sim, fora do horário da aula.	Sim, fora do horário da aula, mas há dificuldades para preencher as vagas.	Alta.
Omnilateralidade	Sim, empreendedorismo e mercado de trabalho (orientação profissional)	Sim, algumas intervenções relacionadas às questões de gênero.	Média.

Quadro 5.5 – Tabela comparativa do processo ensino-aprendizagem do curso técnico de nível médio integrado do(a) docente #4.

Aspecto	Curso Técnico	IFPE	Similaridade
Aulas	Aulas tradicionais, aulas sem computador, aulas expositivas sem prática, exemplos lúdicos.	Aulas com práticas com problemas contextualizados, exemplos lúdicos (exemplo: formar labirintos na sala para os estudantes criarem algoritmos para a saída, uso de ferramentas como Kahoot!) Separação das aulas semanais em encontros de 3 e 1 hora/aula. As aulas de uma hora muitas vezes eram usadas para avaliações. O primeiro encontro sobre um assunto normalmente não é com uma linguagem de programação. Há desafios onde os estudantes veem um vídeo e fazem um desafio relacionado ao vídeo.	Média.
Material Didático	Livro, materiais lúdicos	Materiais lúdicos, livros, slides, indicação de cursos e vídeos de terceiros no YouTube, e sites na internet.	Média.
Avaliação	Prova no papel.	Prova (não pretende fazer mais), exercícios, projeto, avaliação contínua, algumas avaliações são feitas no papel.	Baixa.
Monitoria	Sim, fora do horário da aula.	Sim, fora do horário da aula, mas há dificuldades para preencher as vagas.	Alta.
Omnilateralidade	Não	Sim, algumas intervenções relacionadas às questões de gênero.	Baixa.

5.4.5 Análise da Entrevista

A entrevista revelou que o(a) docente vem empregando práticas originárias do período de aprendizado da programação e que funcionaram para ele(a) aprender a programar. Assim, o uso de exemplos lúdicos na introdução de assuntos novos (modelo ao qual foi exposto(a) durante o curso técnico) e também na contextualização dos problemas estudados em sala de aula (modelo ao qual foi exposto(a) durante a graduação) são os dois principais exemplos dessa similaridade. Ao mesmo tempo, há diferenças importantes no modelo de ensino aplicado do IFPE. É possível observar a utilização de ferramentas tecnológicas, avaliações continuadas dos estudantes, material didático incluindo ferramentas como vídeos e abordagem da questão da presença feminina na área de TI. Dado o

exposto, é possível concluir que apesar do modelo de ensino-aprendizado utilizado nas aulas do(a) docente atualmente trazer aspectos importantes originários dos tempos de graduação e curso técnico do(a) docente, ele possui majoritariamente práticas que não se fizeram presentes durante o período de aprendizado da programação do(a) docente.

5.5 ENTREVISTA #5

5.5.1 *Trajectoria profissional e acadêmica*

A entrevista de número 5 foi conduzida com um(a) docente que entrou na graduação ao fim dos anos 90, fez mestrado logo após a graduação e voltou à academia ao final da década passada para a realização do doutorado. A experiência acadêmica no âmbito da pesquisa foi construída em paralelo à experiência no mercado de trabalho, onde atuações em papéis como o desenvolvimento de software e ensino de em instituições privadas de ensino superior fizeram parte dessa trajetória.

Diferente dos demais entrevistados, o perfil acadêmico do(a) docente revela que sua tese possui relação direta com o ensino da programação, representando assim um dos raros exemplos na literatura científica brasileira onde a discussão é feita sobre aspectos metodológicos do ensino, e não sobre ferramentas tecnológicas para o ensino da programação. Uma visão foi exemplificada através das seguintes frases: “A ferramenta não é uma salvação, ela não é uma bala de prata ... ela facilita, mas não é a solução para as grandes dificuldades” e “O pesquisador brasileiro acaba focando mais na construção de ferramentas (em contraste com os estrangeiros).”.

5.5.2 *Estrutura e estudos durante a graduação*

Um aspecto importante sobre a aprendizagem da programação do(a) docente entrevistado(a) e talvez possa ter relação com o desenvolvimento de uma tese de doutorado sobre o ensino da programação, foi do(a) docente não possui computador durante o início da graduação, o que o(a) levava a desenvolver programas no papel para posteriormente serem levados aos laboratórios da universidade para transcrição e testes.

5.5.3 *Comparativo entre processo de ensino-aprendizado na graduação e no IFPE*

Em sua graduação, as aulas de introdução à programação eram expositivas e teóricas, com exemplos de código presentes nos slides, mas sem execução destes códigos durante as aulas. Práticas da programação ocorriam tanto dentro quanto fora do horário das aulas e eram guiadas majoritariamente por monitores. Já no IFPE, segundo o(a) docente, as aulas ministradas inicialmente eram muito parecidas com as aulas da graduação, mas aos poucos foram sendo realizadas mudanças importantes construídas de forma empírica e guiadas por um processo de experimentação, erros e acertos. Esse processo resultou em um modelo implementado atualmente através de aulas invertidas, onde os estudantes recebem o material didático antes da aula. Assim, a aula é utilizada como momento de revisão, estudos de exemplo de código (através de auxo-explicação) e compartilhamento de código, mesmo sem que todos os estudantes tenham estudado o material uma vez que, segundo o(a) entrevistado(a) “Eles (os estudantes) têm uma dificuldade imensa de estudar o material antes de chegar em sala.”.

Com relação às avaliações, em sua graduação foram utilizadas listas de exercícios (individuais), provas (individuais) e projetos (em grupo). Enquanto que no IFPE, a metodologia de avaliação utilizada atualmente envolve a realização de provas onde os estudantes têm 24 horas para fazer as questões que envolvem a construção de programas de computador e um pequeno número de questões teóricas.

No âmbito da monitoria, há um contraste entre os tempos da graduação, onde havia um alto número de monitores que tinham o importante papel de guiar os estudantes na prática da programação. Enquanto isso, dentro do IFPE o modelo não foi reproduzido, havendo um número reduzido de monitores que raramente são consultados pelos estudantes.

Outro contraste importante ocorre no perfil dos estudantes, uma vez que na graduação do(a) docente os colegas de classe vinham dos colégios mais elitizados de Pernambuco, alguns chegando até chegaram a ganhar carros de presente por terem sido aprovados no vestibular de uma universidade federal. Em contrapartida, a percepção do perfil dos estudantes do IFPE por parte do(a) entrevistado(a) é de que eles são de uma classe social diferente, de menor poder aquisitivo, possuindo dificuldades financeiras graves, o que acaba refletindo no cotidiano deles.

Por fim, com relação à omnilateralidade do ensino, no componente curricular de introdução à programação da graduação não houve qualquer tipo de discussão relacionada à temas que não fossem diretamente relacionados à programação. Contudo, foi relatado que dentro do IFPE, temas como gênero e raça são tratados de forma institucional, em eventos realizados pelo *Campus*. E, apesar dos assuntos como gênero e raça não comporem a ementa do CC de introdução à programação, os temas são abordados em algumas das aulas.

5.5.4 Tabela Comparativa do Processo Ensino-Aprendizagem

Quadro 5.6 – Tabela comparativa do processo ensino-aprendizagem da graduação do(a) docente #5.

Aspecto	Graduação	IFPE	Similaridade
Aulas	Aula expositiva teórica com exemplos, aulas práticas eram feitas pelos monitores no laboratório, as aulas práticas eram feitas. dentro e fora do horário de aula.	No início era muito mais parecida como ela aprendeu: parte teórica, exemplos e depois com listas e apoio aos estudantes pelo docente e pelos monitores. Sala de aula invertida, exemplos, auto-explicação. A sala de aula é um momento de compartilhamento dos exemplos com revisão linha a linha, perguntas e troca de código entre os estudantes, os estudantes também explicam seu código aos demais.	Baixa.
Material Didático	Livro texto, slides e material da internet.	Exemplos de código, vídeos, livro-texto, material de resumo.	Média.
Avaliação	Listas de exercícios, prova e projeto.	Prova com consulta, 24 horas para fazer fora da sala de aula, construção de programas, poucas questões teóricas.	Média.
Monitoria	Havia muitos monitores	Há um monitor, mas os estudantes não costumam procurar o monitor.	Média.
Omnilateralidade	Não havia, não entra uma preocupação entre o fim da década de 90 e início da década 2000.	São abordadas em eventos extraclases, como a semana nacional de tecnologia. Apesar do assunto não ser parte da ementa, indiretamente os temas são abordados de forma retórica (discursal).	Baixa.

5.5.5 Análise da Entrevista

Após a análise da entrevista é possível observar uma relação de influência (retratada de forma direta durante a entrevista) entre a forma que o(a) docente

aprendeu a programar e como o ensino começou a ser realizado dentro do IFPE. Porém, com o tempo a metodologia foi sendo modificada para representar melhor os seguintes aspectos: (1) a forma que a(o) entrevistado(a) estudava durante a graduação (através de exemplos e auto-explicação sobre os códigos); (2) os resultados obtidos durante a pesquisa de doutorado realizada pelo(a) docente. Assim, possível afirmar, segundo as informações coletadas durante a entrevista, que embora houve inicialmente influência do processo ensino da programação ao qual o(a) docente foi exposto(a) durante a graduação, atualmente essa influência foi diluída e a metodologia de ensino-aprendizagem utilizada dentro do IFPE já pode ser considerada distinta.

5.6 DISCUSSÃO

Após todas as entrevistas serem analisadas foi possível observar um padrão de relação entre o processo de ensino-aprendizado ao qual os professores foram expostos quando receberam educação formal sobre o tema, e o processo que eles utilizam atualmente nos componentes curriculares no IFPE.

Com relação às aulas, os docentes do IFPE frequentemente utilizam as aulas dos seus professores do passado como ponto de partida para as suas. A partir do modelo ao qual foram expostos, começam a trabalhar para alterá-lo com base nas suas convicções do que funcionaria melhor nas suas aulas. Um dos exemplos mais claros dessa relação é o caso do(a) docente #2 que relatou que seu professor da graduação ministrava as aulas programando e resolvendo problemas no computador e atualmente no IFPE o(a) passa boa parte das suas aulas fazendo o mesmo, porém também incluiu em suas aulas ações de práticas da programação realizadas com os estudantes (o que não ocorria na graduação). Outro exemplo importante vem do(a) docente #5 que relatou usar as aulas da sua graduação como modelo no começo da sua carreira docente, mas decidiu testar estratégias que ele(a) utilizava para estudar em sua graduação, como a auto-explicação.

No contexto do material didático utilizado, é possível notar uma importante semelhança entre aquilo que foi utilizado na graduação dos professores, e aquilo que eles adotam em suas aulas. Nesse sentido, materiais como slides, listas de exercícios, exemplos de código e livros-texto são amplamente utilizados. Porém, atualmente os professores do IFPE também utilizam em suas aulas tipos de materiais diferentes daqueles que foram utilizados no CCs de introdução a

programação que cursaram como no caso do(a) docente #3 que utiliza um projeto guiado durante o semestre e do(a) docente #4 que indica vídeos do YouTube para seus estudantes.

Além disso, no âmbito da avaliação, ainda é possível encontrar semelhanças, como no caso do docente #1 que utiliza minitestes, assim como na sua graduação. Porém, na maioria dos casos, as semelhanças dão espaço a diferenças importantes como o uso de projetos e avaliações que em muitos casos eram feitas na graduação em papel, mas que agora são feitas no computador.

Entretanto, no aspecto do uso de monitores, as semelhanças são mais proeminentes. Assim o modelo de utilizados pelos professores dos(as) docentes, onde os monitores têm o papel de auxiliar aos estudantes, especialmente fora do horário das aulas, é replicado pelos(as) docentes #1, #2, #3 e #5. Apenas o docente #5 divergiu deste modelo, pois relatou que seus monitores acabam o ajudando mais com a matéria do que os próprios estudantes.

Por fim, na dimensão da omnilateralidade, similaridades também foram relatadas, como no caso do(a) docente #2 que discute sobre empreendedorismo com seus estudantes, assim como seu professor da graduação o fazia. Mas forma geral, tanto na graduação dos(as) docentes, quanto em suas aulas no IFPE aspectos omnilaterais são discutidos de forma muito tímida apesar de haver esforços preliminares nesse sentido, no caso dos(as) docentes #4 e #5 que chegam a abordar a questão gênero em suas aulas, apesar de isto não está presente nas ementas de seus componentes curriculares.

Dado o exposto durante o presente capítulo, é possível voltar à hipótese que originou este trabalho: os processos de ensino-aprendizagem da programação de computadores utilizados pelos docentes do IFPE possuem uma relação direta com os processos adotados pelos professores desses docentes em suas formações. Ao revisarmos a hipótese da pesquisa, após as entrevistas é possível afirmar que há evidências da existência uma relação direta entre ambos os processos. Ou seja, o processo de pesquisa desenvolvido no curso deste projeto levou ao encontro de evidência que sugerem a veracidade da hipótese da pesquisa. Além disso, também é possível afirmar que há evidências de que a relação, por parte do(a) docente do IFPE, entre os processos de ensino-aprendizagem de suas educações formais, é de inspiração e utilização como pontos de partida para a criação de um processo

próprio. Também é possível afirmar que a relação não é de cópia, muito menos de réplica.

Por fim, outro fator importante que foi possível explorar durante as entrevistas foi a comparação entre o perfil dos colegas de graduação dos professores e dos seus estudantes do IFPE. Neste contexto, foi possível observar que de forma unânime, todos os professores do IFPE que há diferenças importantes. Quatro dos cinco entrevistados classificaram seus colegas de graduação como advindos de classes sociais mais altas, e relatam os estudantes do IFPE como majoritariamente originários de classes sociais menos privilegiadas. Entretanto, apesar dessa percepção, nenhum dos professores durante as entrevistas relatou que seus processos de ensino-aprendizagem foram concebidos com o perfil social dos estudantes em mente, o que teoricamente poderia gerar um conjunto de problemas que são dignos de outro projeto de pesquisa. Por exemplo: qual seria a chance de um processo de ensino-aprendizagem fortemente baseado no uso de ferramentas tecnológicas funcionar adequadamente para estudantes que tiveram restrição de acesso computadores em suas vidas, sendo inclusive parte dos estudantes advindos de famílias que não tiveram sequer acesso à alfabetização?

6 CONCLUSÕES E TRABALHOS FUTUROS

Após o longo processo de análise desenvolvido durante esta monografia, foi possível encontrar um nicho de pesquisa pouco investigado (especialmente no âmbito nacional): o processo de ensino-aprendizagem da programação de computadores. Em seguida, foi possível realizar uma definição do escopo do projeto para que o mesmo fosse relevante ao contexto do curso de Especialização em Educação ao qual esta monografia encontra-se incluída. Assim, o tema do projeto foi delineado como o estudo dos processos de ensino-aprendizagem da programação de computadores dentro do IFPE, e mais especificamente a relação entre a forma que os professores do IFPE aprenderam a programar e como eles ensinam atualmente no instituto. Só então foi possível estabelecer a metodologia de coleta dos dados a serem analisados: o estudo dos projetos pedagógicos dos cursos TI do IFPE e entrevistas com os professores dos CCs de introdução à programação dos cursos de TI do IFPE.

Após análise dos dados coletados, foi possível encontrar evidências de que os processos de ensino-aprendizagem utilizados pelos professores IFPE nos CCs de introdução à programação possuem uma relação direta com aqueles aos quais eles foram expostos durante suas graduações, o que aponta para a validade da hipótese da pesquisa. Nesse sentido, a maioria dos processos de ensino-aprendizagem utilizados no IFPE tiveram como inspiração ou ponto de partida a forma que os próprios professores foram ensinados a programar durante suas graduações.

Nesse sentido, o projeto deixar como legado para a comunidade científica duas contribuições:

1. O retrato histórico da forma que o ensino da programação de computadores é realizado dentro do IFPE;
2. Uma análise comparativa da relação entre a forma que os professores do IFPE aprenderam a programar durante suas graduações, e como eles ensinam programação em seus *Campi*;

Além das contribuições desenvolvidas durante a pesquisa, este projeto também foi capaz de revelar ao menos os seguintes espaços em aberto que poderiam ser abordados em futuros trabalhos:

- 1. Adaptação do processo de ensino-aprendizagem da programação ao perfil dos estudantes:** Apesar de reconhecerem as diferenças entre o perfil

dos estudantes do IFPE e o perfil dos seus colegas de turma, os professores revelaram em suas entrevistas que utilizam processos que foram inspirados em outros que tinham como público-alvo estudantes de classes sociais mais privilegiadas. Porém os estudantes do IFPE são majoritariamente de classes sociais menos privilegiadas, sendo muitos inclusive considerados em estado de vulnerabilidade social. Nesse sentido, nenhum dos professores relatou que seu processo de ensino-aprendizagem leva em conta o perfil dos estudantes, na verdade, o que se viu nas entrevistas foi que há professores incluindo em seus processos práticas que funcionaram com eles próprios. De certa forma, essa posição dos professores pode ter relação com o fato de que, segundo as próprias entrevistas, a maioria deles não possuía o mesmo perfil social dos estudantes durante suas graduações. Assim, há um espaço importante aberto para uma discussão científica sobre como criar um processo de ensino-aprendizagem que seja voltado para o perfil dos estudantes do IFPE.

2. **A integração da monitoria ao processo de ensino-aprendizagem:** o relato das entrevistas colocou o papel da monitoria dentro do IFPE em um contexto secundário, onde os mesmos trabalham majoritariamente fora do horário das aulas e realizando atividades complementares àquela realizadas durante as aulas. Por serem ex-estudantes do CC, os monitores possuem um ponto de vista distinto do CC, quando comparado ao professor. E é justamente essa visão que os coloca em uma condição de colaborar de forma direta no ensino da programação dos seus colegas de curso. Nesse sentido, há um espaço aberto para o uso da monitoria de forma integrada e fundamental no processo de ensino-aprendizagem.
3. **A integração de aspectos omnilaterais no ensino da programação:** O desenvolvimento de software frequentemente é realizado em um ambiente plural (com pessoas de distintos gêneros, raças, classe sociais e religiões) e interdisciplinar (com profissionais de área distintas como design, recursos humanos e administração). Nesse contexto, o programador profissional não pode limitar-se única e exclusivamente a ser competente tecnicamente, especialmente porque a área de TI já é marcada por exemplos de preconceito recorrentes, como o machismo. Mesmo assim, as entrevistas revelaram que dentro do IFPE o ensino da programação explora poucos aspectos omnilaterais, sendo inclusive as ementas dos CCs puramente técnicas. Assim

sendo, há espaço para a construção de um processo de ensino-aprendizagem que seja capaz de integrar aspectos omnilaterais aos aspectos técnicos.

4. **É possível generalizar as descobertas deste projeto para outras áreas ou outras instituições?** Em outros cursos do IFPE, especialmente onde os professores majoritariamente não possuem formação em educação, também é possível traçar uma relação entre a forma que eles aprenderam e a forma que eles ensinam? Em outras instituições, públicas ou privadas, também é possível traçar uma relação entre a forma que eles aprenderam e a forma que eles ensinam?
5. **Como realizar um julgamento de valor sobre a forma que o IFPE ensina seus estudantes a programar?** À medida que o projeto privou-se de realizar qualquer tipo de avaliação dos processos de ensino-aprendizagem aplicados dentro IFPE, ele também deixa em aberto um espaço para refletir sobre como avaliá-las. Por exemplo, seguinte pergunta fica em aberto: as semelhanças e diferenças entre as formas de aprendizagem e ensino expostas nesta monografia são benéficas aos estudantes?

REFERÊNCIAS

AURELIANO, Viviane Cristina Oliveira. **Stepwise self-explanation**: Uma abordagem para o aprendizado de linguagens de programação através da auto-explicação de exemplos em vídeo. Orientador: Patrícia Cabral de Azevedo Restelli Tedesco. 2016. Tese (Doutorado) - Universidade Federal de Pernambuco, Recife, 2016.

BASER, Mustafa. Attitude, Gender and Achievement in Computer Programming. **Middle-East Journal of Scientific Research**, [s. l.], v. 14, ed. 2, p. 248-255, 2013.

BLIKSTEIN, Paulo; WORSLEY, Marcelo; PIECH, Chris; SAHAMI, Mehran; COOPER, Steven; KOLLER, Daphne. Programming Pluralism: Using Learning Analytics to Detect Patterns in the Learning of Computer Programming. **Journal of the Learning Sciences**, [s. l.], v. 24, ed. 4, p. 561-599, 24 out. 2014.

BONI, Valdete; QUARESMA, Sílvia Jurema. Aprendendo a entrevistar: como fazer entrevistas em Ciências Sociais. **Em Tese**, [s. l.], v. 2, ed. 1, p. 68-80, 2005.

BRITO, Miguel A.; DE SÁ-SOARES, Filipe. Assessment frequency in introductory computer programming disciplines. **Computers in Human Behavior**, [s. l.], v. 30, p. 623-628, 2014.

CAMPOS, Ricardo Luiz B. L. **Metodologia ERM2C**: Para melhoria do processo de ensino-aprendizagem de lógica de programação. Anais do XVIII Workshop sobre Educação em Computação, Belo Horizonte, v. 2, p. 961-970, 2010.

CIAVATTA, Maria. O ENSINO INTEGRADO, A POLITECNIA E A EDUCAÇÃO OMNILATERAL. POR QUE LUTAMOS?. **Trabalho & Educação**, [s. l.], ed. 23, p. 187-205, 2014.

DE SOUZA, Cláudio Morgado. **VisuAlg - Ferramenta de apoio ao ensino de programação**. Revista Eletrônica TECCEN, [s. l.], v. 2, n. 2, p. 01-09, 2009.

DE RAADT, Michael; HAMILTON, Margaret; LISTER, Raymond; TUTTY, Jodi. Approaches to learning in computer programming students and their effect on success. **Proceedings of the 28th HERDSA Annual Conference**, [s. l.], p. 407-414, 2005.

DESLANDES, Suely Ferreira; NETO, Otaio Cruz; GOMES, Romeu; MINAYO, Maria Cecília de Souza. **Pesquisa Social: Teoria, Método e Criatividade**. 21. ed. Petrópolis: Vozes, 2002. 96 p. ISBN 978-8532652027.

FONTANA, Felipe. Técnicas de pesquisa. In: MAZUCATO, Thiago; ZAMBELLO, Aline Vanessa; SOARES, Alessandra Guimarães; TAUIL, Carlos Eduardo; DONZELLI, Cleivaldo Aparecido; FONTANA, Felipe; CHOTOLLI, Wesley Piante. **Metodologia da Pesquisa e do Trabalho Científico**. 1. ed. Penápolis: FUNEPE, 2018. cap. 8, p. 59-78. ISBN 97885-93683-03-9.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. 6ª. ed. São Paulo: Atlas SA, 2008. ISBN 978-85-224-5142-5.

GUNBATAR, Mustafa Serkan. Examination of Undergraduate and Associate Degree Students' Computer Programming Attitude and Self-Efficacy According to Thinking Style, Gender and Experience. **Contemporary Educational Technology**, [s. l.], v. 9, ed. 4, p. 354-373, 2018.

HAWI, Nazir. Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. **Computers & Education**, [s. l.], v. 54, ed. 4, p. 1127-1136, 2010.

LIMA, José Weliff Ferreira; MEIRELES, Alandson Mendonça R. **Metodologias de Ensino de Lógica de Programação**. Revista Expressão Católica, [s. l.], v. 3, ed. 2, p. 167-171, 2014.

KAZIMOGLU, Cagin; KIERNAN, Mary; BACON, Liz; MACKINNON, Lachlan. A Serious Game for Developing Computational Thinking and Learning Introductory

Computer Programming. **Procedia: Social and Behavioral Sciences**, [s. l.], v. 47, p. 1991-1999, 2012.

KERSTEEN, Zoe A.; LINN, Marcia C.; CLANCY, Michael; HARDYCK, Curtis. Previous Experience and the Learning of Computer Programming: The Computer Helps Those Who Help Themselves. **Journal of Educational Computing Research**, [S. l.], v. 4, n. 3, p. 321-333, 1 ago. 1988.

KUECHLER, William L.; SIMKIN, Mark G. How Well Do Multiple Choice Tests Evaluate Student Understanding in Computer Programming Classes?. **Journal of Information Systems Education**, [s. l.], v. 14, n. 4, 2003.

KUENZER, Acácia Zeneida. **EDUCAÇÃO E TRABALHO NO BRASIL: o estado da questão**. 1ª. ed. Brasília: INEP, 1991. 125 p.

MAZUCATO, Thiago. Métodos. In: MAZUCATO, Thiago; ZAMBELLO, Aline Vanessa; SOARES, Alessandra Guimarães; TAUIL, Carlos Eduardo; DONZELLI, Cledivaldo Aparecido; FONTANA, Felipe; CHOTOLLI, Wesley Piante. **Metodologia da Pesquisa e do Trabalho Científico**. 1. ed. Penápolis: FUNEPE, 2018. cap. 7, p. 53-58. ISBN 97885-93683-03-9.

MINISTÉRIO DA EDUCAÇÃO. Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Projeto Pedagógico de Curso. **Projeto Pedagógico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**, Recife, 2012.

MINISTÉRIO DA EDUCAÇÃO. Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Projeto Pedagógico de Curso. **Projeto Pedagógico do Curso Técnico de Informática para Internet**, Belo Jardim, 2014a.

MINISTÉRIO DA EDUCAÇÃO. Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Projeto Pedagógico de Curso **Projeto Pedagógico do Curso Técnico de Informática para Internet Subsequente**, Igarassu, 2014b.

MINISTÉRIO DA EDUCAÇÃO. Câmara de Educação Superior. Resolução. **RESOLUÇÃO Nº 5, DE 16 DE NOVEMBRO DE 2016**, Brasília, 2016.

MINISTÉRIO DA EDUCAÇÃO. Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Projeto Pedagógico de Curso. **Projeto Pedagógico do Curso Técnico de Informática para Internet Subsequente**, Jaboatão dos Guararapes, 2018.

MINISTÉRIO DA EDUCAÇÃO. Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Projeto Pedagógico de Curso. **Projeto Pedagógico do Curso Técnico de Informática para Internet Subsequente**, Palmares, 2019a.

MINISTÉRIO DA EDUCAÇÃO. Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Projeto Pedagógico de Curso. **Projeto Pedagógico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**, Garanhuns, 2019b.

MINISTÉRIO DA EDUCAÇÃO. Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Projeto Pedagógico de Curso. **Projeto Pedagógico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**, Paulista, 2019c.

MINISTÉRIO DA EDUCAÇÃO. Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Projeto Pedagógico de Curso. **Projeto Pedagógico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**, Jaboatão, 2020a.

MINISTÉRIO DA EDUCAÇÃO. Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Projeto Pedagógico de Curso. **Projeto Pedagógico do Curso de Bacharelado em Engenharia de Software**, Belo Jardim, 2020b.

MINISTÉRIO DA EDUCAÇÃO. Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. Projeto Pedagógico de Curso. **Projeto Pedagógico do Curso de Tecnologia em Sistemas para Internet**, Igarassu, 2021.

MOREIRA, Mireille Pinheiro; FAVERO, Eloi Luiz. **Um ambiente para ensino de programação com feedback automático de exercícios.** Anais do XVII Workshop sobre Educação em Computação, [s. l.], v. 17, 2009.

PRESIDÊNCIA DA REPÚBLICA. Lei nº 11.892, de 29 de dezembro de 2008. Institui a Rede Federal de Educação Profissional, Científica e Tecnológica, cria os Institutos Federais de Educação, Ciência e Tecnologia, e dá outras providências. **Lei nº 11.892, de 29 de Dezembro de 2008**, Brasília, 2008.

PRESIDÊNCIA DA REPÚBLICA. **Lei nº 12.711, de 29 de agosto de 2012.** Dispõe sobre o ingresso nas universidades federais e nas instituições federais de ensino técnico de nível médio e dá outras providências. LEI Nº 12.711, DE 29 DE AGOSTO DE 2012, Brasília, 29 ago. 2012.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico.** 2. ed. Novo Hamburgo: Feevale, 2013. 276 p. ISBN 978-85-7717-158-3.

RAPKIEWICZ, Cleli Elena; FALKEMBACH, Gilse; SEIXAS, Louise; ROSA, Núbia dos Santos; DA CUNHA, Vanildes Vieira; KLEMMANN, Miriam. **Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais.** RENOTE: Revista Novas Tecnologias na Educação, Porto Alegre, v. 4, ed. 2, 2006.

REPÚBLICA FEDERATIVA DO BRASIL. Ministério da Educação. **Catálogo Nacional dos Cursos Superiores de Tecnologia.** 3. ed. Brasília: [s. n.], 2016. 194 p.

REPÚBLICA FEDERATIVA DO BRASIL. Ministério da Educação. **Catálogo Nacional dos Cursos Técnicos.** 4. ed. Brasília: [s. n.], 2021. 510 p.

ROCHA, Paulo Santana; FERREIRA, Benedito; MONTEIRO, Dionne; NUNES, Danielle da Silva Costa; GÓES, Hugo Cezar do Nascimento. **Ensino e Aprendizagem de Programação: Análise da Aplicação de Proposta Metodológica Baseada no Sistema Personalizado de Ensino.** RENOTE: Revista Novas Tecnologias na Educação, Porto Alegre, v. 8, ed. 3, 2010.

RODRIGUES, Miguel Ângelo Vilela. O Tratamento e Análise de Dados. In: METODOLOGIA para a Investigação Social. 1. ed. Lisboa: Escolar, 2011. cap. 9, p. 171-210. ISBN 9789725923290.

SAVIANI, Dermeval. **A pedagogia histórico-crítica e a educação escolar.** Pensando a educação, São Paulo, p. 23-33, 1989.

SITTHIWORACHART, Jirarat; JOY, Mike. Effective peer assessment for learning computer programming. **ACM SIGCSE Bulletin**, [s. l.], v. 36, ed. 3, p. 122-126, 2004.

ZUALKERNAN, Imran A.; ALLERT, James; QADAH, Ghassan Z. Learning Styles of Computer Programming Students: A Middle Eastern and American Comparison. **IEEE Transactions on Education**, [s. l.], v. 49, ed. 4, p. 443-450, 13 nov. 2006.

APÊNDICE A – ENTREVISTA #6

Esta entrevista foi realizada com um(a) docente do IFPE que ministra aulas de introdução à programação em um curso de engenharia (porém não de engenharia de software). Por tanto, como o(a) docente não ministra aulas introdução à programação em nenhum dos cursos inclusos no Quadro 1.1, a entrevista não foi incluída no capítulo 5 - RESULTADOS DAS ENTREVISTAS E DISCUSSÕES desta monografia.

Trajetória profissional e acadêmica

O(a) docente entrevistado iniciou sua graduação em um curso de engenharia no meio na segunda metade da década de 90 e possui especialização, mestrado e doutorado, sendo todos também em áreas técnicas e iniciados durante a década 2010.

Por conta de um curso técnico iniciado antes da graduação, o(a) docente já trabalhava quando iniciou seus estudos na universidade. E foi justamente no trabalho que ele teve os primeiros contatos com a programação de computadores, mesmo que de forma incipiente. Além disso, após tornar-se professor(a) do IFPE, o(a) docente ministrou aulas de programação em cursos de licenciatura, até finalmente tornar-se o responsável pelas aulas de introdução à programação em um curso de engenharia em seu *Campus*.

Estrutura e estudos durante a graduação

Como o(a) docente trabalhava durante sua graduação, dispunha de poucas horas para estudar fora do período de suas aulas (que eram no turno da noite). Porém também o(a) docente relatou que “Com o tempo fui me acostumando com a rotina de tempo restrito e a gente foi conseguindo vencer as rotinas de adversidades do curso”. Além disso, em sua casa, o mesmo possuía à sua disposição computador com acesso à internet para escrever, testar e executar seus programas.

Comparativo entre processo de ensino-aprendizado na graduação e no IFPE

Durante a entrevista o(a) docente relatou sua experiência na graduação foi típica de um curso de engenharia da época, onde a programação não era

considerada uma qualidade essencial para um engenheiro, sendo inclusive o(a) professor(a) responsável pelo CC no curso uma pessoa com pouca vivência na área.

Nesse sentido, as aulas em sua graduação eram majoritariamente expositivas, com períodos de prática uma vez que as aulas eram ministradas em um laboratório. Porém, suas aulas no IFPE são iniciadas com um período de motivação voltado para a importância da programação no mundo atual. Já as demais aulas, apesar conterem partes expositivas, têm um foco maior na prática da programação, sendo isto relatado da seguinte forma: “Eu só parava de fazer exercícios quando percebia que a grande maioria estava entendendo.” Nestas práticas, são trabalhados problemas do cotidiano dos estudantes e da engenharia. Além disso, os estudantes são estimulados a criarem, testarem e corrigirem os problemas de seus programas. Inclusive, a questão da criação dos programas em si foi relatada como uma das principais dificuldades dos estudantes:

“A grande dificuldade da disciplina de introdução (à programação nas engenharias) é que você não vai aprender a fazer um método, você não vai compreender como fazer uma integral, você não vai aprender as funções disso, as funções daquilo, você tem que aprender a criar, você tem que desenvolver.”

Ainda sobre a prática da programação, o(a) docente relatou que inicialmente utiliza ferramentas como o Scratch⁹ para desenvolver o raciocínio lógico para a criação de algoritmos. Posteriormente, os estudantes fazem a transição para a programação em código-fonte através de ferramentas como o Scilab¹⁰.

No âmbito da sua graduação, havia pouco materiais didáticos adotados, sendo o quadro da sala a principal fonte de transmissão do material que a(o) professor(a) utilizava para material de desenvolvia em suas aulas. Em contraste, no IFPE, o(a) docente utiliza majoritariamente slides para a parte expositiva das suas

⁹<https://scratch.mit.edu/>

¹⁰<https://www.scilab.org/>

aulas, além de adotar livros por conta dos problemas e exercícios que os mesmos possuem.

Em termos avaliativos, em sua graduação eram utilizados exercícios e um programa (como um projeto) para compor as notas dos estudantes. No IFPE, o modelo adotado envolve listas de exercícios e uma prova, ambas no computador. Ainda sobre a prova, foi relatada uma mudança nos últimos semestres onde o(a) docente relatou que em suas primeiras turmas cobrava mais questões e com soluções relativamente simples. Porém, atualmente ele(a) reduziu o número de questões para cobrar soluções mais complexas das mesmas.

O uso da monitoria no CC de Introdução à Programação na graduação foi algo que o(a) docente informou não se recordar. Enquanto que em suas aulas no IFPE, monitores já foram utilizados no passado, mas não foram sendo utilizados recentemente.

Quando consultado sobre aspectos omnilaterais no ensino da programação em sua graduação, o(a) docente informou que as aulas eram puramente técnicas e nenhum outro aspecto, além da programação, era abordado. Contudo, o(a) docente relatou que em suas aulas aborda a contextualização da programação especialmente para as engenharias, de forma que seus estudantes possam compreender a mesma como um conhecimento estratégico para o futuro de suas carreiras.

No contexto do perfil dos seus colegas de graduação estudantes, foi relatado que eles, em sua maioria, trabalhavam, mas que entraram na graduação sabendo o que era um curso de engenharia e que sabiam o que iriam estudar no curso. Enquanto isso, no IFPE os estudantes advindos dos cursos de técnicos integrados ao ensino médio do *Campus* possuem uma compressão sobre o que é o curso e o que eles irão estudar. Porém, os demais estudantes, de forma geral, desconhecem o curso. Não houve relatos relacionados ao perfil socioeconômico tanto dos colegas de classe da graduação, quanto dos estudantes do IFPE.

Tabela Comparativa do Processo Ensino-Aprendizagem

Quando A.1 – Tabela Comparativa do Processo Ensino-Aprendizagem do(a) Docente#6.

Aspecto	Graduação	IFPE	Similaridade
Aulas	Expositivas no laboratório, com práticas através de exercícios, o quadro era muito utilizado, algumas vezes também era usado um datashow.	Expositiva com exemplos nomeio seguidas das práticas. Uma aula típica segue a ordem: problema inicial, exposição, exercícios e prática. Realiza muitos exercícios práticos. Sempre voltada a estimular os estudantes a resolverem problemas. Trabalha problemas práticos para resolver problemas do cotidiano e da engenharia elétrica.	Baixa.
Material Didático	Slides (porém poucos usados)	Slides. Livros (por conta dos problemas e exercícios).	Baixa.
Avaliação	Lista de exercícios, um trabalho (programa que valia nota)	Listas e Provas no computador. Provas inicialmente: problemas (em torno de 3) para implementar soluções simples. Provas atualmente: menos problemas (1 ou 2) e respostas mais detalhadas.	Baixa.
Monitoria	Não lembra	Não	Não se aplica
Omnilateralidade	Aulas puramente técnicas	Aspectos profissionais da programação, exemplo: onde está a programação na nossa vida? A contextualização é feita durante todo o curso para justificar porque um engenheiro precisa saber programar.	Baixa

Análise da Entrevista

Ao realizar a entrevista foi possível constatar que, de forma geral, há uma baixa semelhança entre o processo de ensino na graduação do(a) docente e a forma que o(a) mesma ensina dentro do IFPE. Uma possível explicação para isso, é que na sua graduação a programação era tratada como uma área de pouca importância, enquanto que no curso de engenharia do seu *Campus* no IFPE ela é tratada como uma área estratégica. Outra possibilidade para explicar essa diferença seria o fato de que o(a) professor(a) da graduação não possuía muita experiência na área.

Independente disso, o CC curricular no IFPE possui contornos muito distintos e mais relevantes, quando comparado do CC da graduação do(a) docente. Assim, no IFPE os conhecimentos adquiridos no CC são utilizados em outros CCs dentro do curso de engenharia. Dado o contexto, o(a) docente relatou utilizar processo próprio de ensino-aprendizagem, refinado através de tentativas e testes com os estudantes, algo que foi relatado na entrevista como um processo ainda em desenvolvimento (como por exemplo, nas mudanças realizadas no método avaliativo).