UNIFACCAMP
Centro Universitário Campo Limpo Paulista

Uma Abordagem Semiótica para Utilização de Modelos Tridimensionais na Aprendizagem de Programação Orientada a Objetos

Gregory Acacio Seibert Oliveira

Setembro / 2018

Projeto de pesquisa de Mestrado em Ciência da Computação

Uma Abordagem Semiótica para Utilização de Modelos Tridimensionais na Aprendizagem de Programação Orientada a Objetos

Esse documento corresponde à Dissertação apresentada à Banca Examinadora para defesa de Mestrado em Ciência da Computação do Centro Universitário Campo Limpo Paulista.

Campo Limpo Paulista, 25 de setembro de 2018.

Gregory Acacio Seibert Oliveira

Rodrigo Bonacin (Orientador)

FICHA CATALOGRÁFICA

Elaborada pela Biblioteca Central da UNIFACCAMP

O47a

Oliveira, Gregory Acacio Seibert

Uma abordagem semiótica para utilização de modelos tridimensionais na aprendizagem de programação orientada a objetos / Gregory Acacio Seibert Oliveira. Campo Limpo Paulista, SP: UNIFACCAMP, 2018.

Orientador: Prof. Dr. Rodrigo Bonacin

Dissertação (Programa de Mestrado Profissional em Ciência da Computação) – Centro Universitário Campo Limpo Paulista – UNIFACCAMP.

1. Ensino de programação orientada a objetos. 2. Modelos 3D. 3. Semiótica organizacional. 4. Design instrucional. I. Bonacin, Rodrigo. II. Campo Limpo Paulista. III. Título.

CDD-005.721

DEDICATÓRIA

Dedico este trabalho à minha esposa Talita, pois esteve presente em todos os momentos em que mais precisei e sempre foi a minha inspiração para chegar nesta etapa final. Ao meu pai Miguel, que mesmo longe sempre mostrou seu amor por mim em cada momento.

AGRADECIMENTOS

Agradeço a Deus por me dar saúde e paz nos momentos de dificuldades da minha vida acadêmica e profissional.

Agradeço a minha esposa Talita, deu toda a forca que eu precisava, e sempre ajudou direta e indiretamente em toda minha caminhada durante o mestrado.

Agradeço ao meu professor e orientador Rodrigo Bonacin, por ter me aceitado como orientando e também, pela dedicação e paciência que teve comigo em todo estudo dirigido e em todo período de orientação.

Agradeço ao colega de Mestrado, Alessandro Leite, pois tivemos uma parceria em todas as disciplinas, passeios e demais atividades do mestrado.

Agradeço ao UNASP, pois apoiou a minha busca pelo mestrado e forneceu flexibilidade para concluir meus objetivos.

Agradeço a todos colaboradores da UNIFACCAMP, e em especial, aos docentes que tive a oportunidade de conhecer, onde todos deixaram um pouco de inspiração para minha vida acadêmica.

Resumo

O ensino de Programação Orientada a Objetos nos cursos de Tecnologia é um dos principais desafios para os alunos e professores, pois exige muita abstração para entender as maneiras como os métodos de programação se relacionam. A maioria das disciplinas de programação dos cursos técnicos ou de graduação envolvem o desenvolvimento de habilidades de alto nível de complexidade. O aluno ao ser introduzido a esse conteúdo não consegue construir conceitos facilmente, fazendo assim com que o número de evasão no decorrer do curso aumente conforme os conceitos vão se aprofundando. Este estudo visa entender e analisar como objetos em 3D, que representem as relações de conceitos dentro da Programação Orientada a Objetos, podem aprimorar processo de aprendizagem. A literatura exibe vários estudos que confirmam resultados positivos da utilização de objetos 3D em relação a outras áreas de ensino, como ciências biológicas, química, entre outras. No entanto, a área de modelagem em 3D está em evolução e desenvolvimento, isso traz vários desafios de pesquisa, como explorar a objetos 3D virtuais e a manufatura digital para que traga benefícios no processo de aprendizagem de programação. Esta dissertação tem como objetivo analisar e criar métodos e meios inovadores para o ensino de Programação Orientada a Objetos, utilizando formas virtuais modeladas (visando a fabricação digital), para assim, relacionar diversos níveis de abstração no processo de aprendizagem. A Semiótica Organizacional e o Design Instrucional são utilizados como referencial teórico e metodológico. A dissertação avalia a aplicação do método proposto em cursos de programação, tanto no ensino técnico como na graduação. O resultado se da por meio de uma avaliação prática com a utilização dos objetos virtuais em 3D, durante o ensino de Programação Orientada a Objetos.

Palavras Chaves: *Ensino de Programação Orientada a Objetos, Modelos 3D, Semiótica Organizacional, Design Instrucional.*

Abstract

One of the main challenges for students and teachers in technology related courses is how to teach Object Oriented Programming. This activity requires high abstraction skills necessary to understand how the programming methods are relate each other. Most of the programming disciplines of technical or undergraduate courses involve the development of skills with high level of abstraction and complexity. In their first contact with this content, students are not able to construct concepts easily, thus this causes the increase of evasion during the course as they become deeper. This study aims to understand and analyze how 3D objects, which represent concepts relationships within Object Oriented Programming, can inprove the learning process. The literature presents several studies that confirm the positive results of using 3D objects in other educational fields, such as biology and chemistry. However, the 3D modeling is an evolving area; this brings a number of research opportunities and challenges, such as how 3D virtual objects can bring benefits for the programming learning process. This master dissertation aims to analyze and create innovative methods and means for teaching Object Oriented Programming, using virtual forms inspired by digital manufacturing, in order to relate different levels of abstraction in the learning process. To this end, we adopted Organizational Semiotics and Instructional Design as our frame of reference. The dissertation evaluates the application of the proposed method in programming courses in both technical and undergraduate education. The results are experienced through a practical evaluation using the virtual objects created in 3D, during the Object Oriented Programming leaning process.

Keywords: *Object Oriented Programming Learning, 3D Models, Organizational Semiotics, Instructional Design.*

Glossário

ADDIE – Análise, design, desenvolvimento, implementação e avalização (A*nalyssis, design, development, implemention and evaluation*)

CBIE – Congresso Brasileiro de Informática na Educação

CNE – Conselho Nacional de Educação

Fab@School - Escola que utiliza os métodos de Fabricação Digital para o ensino

MEASUR - Methods for Eliciting, Analysing and Specifying Users Requirements

NTLC - *National Technology Leadership Coalition*

POO – Programação Orientada a Objetos

SO – Semiótica Organizacional

STEM - Ciências, tecnologia, engenharia e matemática *(science, technology, engineering, and mathematics)*

Lista de Figuras

Figura 1 - Sequência de passos na apresentação de conteúdo. 7
Figura 2 - Etapas do processo de Design Instrucional 10
Figura 3 – Exemplo de Hierarquia de Classes 13
Figura 4 - Linha do tempo de tecnologias computacionais chaves na educação 20
Figura 5 - Exemplo do signo como uma relação triádica 26
Figura 6 - Framework semiótico 27
Figura 7 -As principais fases do SAM (Bonacin 2004, adaptado de Liu 2000) 32
Figura 8 - Trabalhos associados por tipo e ano 38
Figura 9 - Relação entre as áreas abordadas na pesquisa 49
Figura 10 - Metodologia de Pesquisa 51
Figura 11 - Visão geral MEPOOMD 53
Figura 12 - Cebola semiótica 56
Figura 13 - Percentual de acerto no pré-teste por turmas 63
Figura 14 - Grupos por gênero no Nível 1 65
Figura 15 - Autoavaliação dos alunos do Nível 1 sobre programação 66
Figura 16 - Grupos por gênero no Nível 2 67
Figura 17 - Autoimagem sobre programação, alunos do nível 2 68
Figura 18 - Laboratório de informática e a aplicação do MEPOOMD 69
Figura 19 - Conhecimento prévio sobre ferramentas de desenvolvimento para POO 70
Figura 20 – NetBeans, software de desenvolvimento Java 71
Figura 21 – Blender, software de modelagem 3D 72
Figura 22 - Resultado da pesquisa sobre a utilização do Blender 73
Figura 23 - Lista de vídeo aulas por tema 74
Figura 24 - Lista de vídeo aulas para a utilização do software Blender 75

Figura 25 - Vídeo Aula 6, exemplo de vídeo 76
Figura 26 - Alunos que tem dificuldade de entender determinado assunto em POO..... 77
Figura 27 - Linguagens utilizadas 78
Figura 28 - Método escolhido polos alunos para aprendizagem em programação 79
Figura 29 - Exemplo de Diagrama de classes utilizado na aplicação..... 83
Figura 30 - Aulas de POO disponíveis para download na área virtual do aluno..... 87
Figura 31 - Entrega de atividade aluno do nível 1..... 88
Figura 32 - Entrega de atividade aluno do nível 2..... 88
Figura 33 - Exemplo de espaço para entrega de atividade 89
Figura 34 - Diagrama de classe com objeto carro 90
Figura 35 - Objeto 3D feito por aluno do nível 1, baseado na classe carro..... 91
Figura 36 - Objeto 3D feito por aluno do nível 2, baseado na classe carro..... 92
Figura 37 - Classe carro desenvolvida em Java na IDE 93
Figura 38 - Porcentagem das respostas dos grupos de controle do nível 1 101
Figura 40 - Gráfico com a porcentagem das respostas dos grupos de teste do nível 1 ... 102
Figura 41 - Gráfico com a porcentagem das respostas dos grupos de controle do nível 2103
Figura 42 - Gráfico com a porcentagem das respostas dos grupos de teste do nível 2 ... 103
Figura 43 - Desempenho do grupo teste do nível 1 nas normas comportamentais 106
Figura 44 - Número de ocorrências dos níveis de cumprimentos das normas do nível 1107
Figura 45 - Desempenho do grupo teste do nível 2 nas normas comportamentais 107
Figura 46 - Número de ocorrências dos níveis de cumprimentos das normas do nível 2108
Figura 47 - Curso Técnico de Informática: 1° ano do Ens. Médio..... 123
Figura 48 - Curso Técnico de Informática: 2° ano do Ens. Médio..... 123
Figura 49 - Curso Técnico de Informática: 3° ano do Ens. Médio..... 124

Figura 50 - 1°, 3° e 5° Semestres de Sistemas de Informação do UNASP-HT realizando a atividade de lógica e desenvolvimento. 124

Figura 51 - 2° Ano Curso Técnico em Informática Realizando Questionário avaliativo sobre a aplicação do MEPOOMD 125

Figura 52 - Alunos do 3° Semestre de Sistemas de Informação se preparando para a criação de uma forma que representasse POO em um ambiente 3D. 125

Figura 53 - 5° Semestre de Sistemas de Informação realizando questionário avaliativo sobre o MEPOOMD 126

Figura 54 - Alunos selecionados para o subgrupo de Teste realizando a avaliação para verificar nível de aprendizado 126

Lista de Tabelas

Tabela 1- Projeção de emprego para desenvolvedores 2

Tabela 2 - Strings de busca utilizadas 36

Tabela 3 - Critérios de inclusão e exclusão 37

Tabela 4 - Quantidade de artigos selecionados por base a passo da pesquisa 37

Tabela 5 - Trabalhos selecionados por critério de inclusão 38

Tabela 6 - Trabalhos por método/abordagem de ensino adotada 39

Tabela 7 - Trabalhos por tecnologias e dispositivo utilizado 40

Tabela 8 - Trabalhos por nível de programação e conceitos de POO 41

Tabela 9 - Síntese de Trabalhos relacionados ao ambiente de Aprendizagem utilizando Fabricação Digital 44

Tabela 10 - Modelo Quadro de Normas Comportamentais 59

Tabela 11 - Tabela de Respostas 60

Tabela 12 - Aplicação MEPOOMD, Total de Alunos 63

Tabela 13 - Nível 1, Aplicação MEPOOMD - Curso técnico Tecnologia da Informação e desempenho no pré-teste 64

Tabela 14 - Nível 2, Aplicação MEPOOMD - Graduação em Sistemas de Informação e desempenho no pré-teste 67

Tabela 15 - Descrição dos domínios em cada nível da cebola semiótica 80

Tabela 16 - Quadro de analise baseado no framework semiótico 81

Tabela 17 - Quadro de norma "Criar classe" 84

Tabela 18 - Quadro de norma "Criar atributos" 84

Tabela 19 - Quadro de norma "Criar métodos" 85

Tabela 20 - Quadro de norma "Criar herança" 85

Tabela 22 - Síntese de respostas do questionário 6 no nível 1 98

Tabela 23 - Síntese de respostas do questionário 6 no nível 2 99

Tabela 24 - Resultados estatísticos - Nível 1..... 105

Tabela 25 - Resultados estatísticos - Nível 2..... 105

SUMÁRIO

1. **INTRODUÇÃO** 1

1.1. Contexto e Motivação 1

1.2. Problemática e Justificativa 3

1.3. Objetivos, Contribuições e Métodos 3

1.4. Estrutura da Proposta 4

2. **REFERENCIAL TEÓRICO E METODOLÓGICO** 6

2.1. Aprendizagem de Programação 6

2.1.1 Design Instrucional no ensino de programação 8

2.1.2 Programação Orientada a Objetos 11

2.1.3 Dificuldades de Aprendizagem de conceitos de POO 16

2.2. Modelagem e Fabricação Digital na Educação 17

2.3. Semiótica Organizacional 25

2.3.1 Ferramenta MEASUR 28

2.3.2 Ferramentas da SO aplicadas ao aprendizado 29

3. **TRABALHOS RELACIONADOS** 35

3.1. Métodos de ensino utilizados 39

3.2. Tecnologias e dispositivos utilizados 39

3.3. Níveis de programação e conceitos abordados 41

3.4. Fabricação digital no ambiente de aprendizagem 42

3.5. Discussão sobre Trabalhos Relacionados 46

4.**METODOLOGIA DE ENSINO DE PROGRAMAÇÃO ORIENTADA A OBJETOS COM MODELAGEM DIGITAL – MEPOOMD** 49

4.1 Elaboração do MEPOOMD 49

4.2 Detalhamento MEPOOMD 53

4.2.1 Análise 54
4.2.2 Design 58
4.2.3 Desenvolvimento 58
4.2.4 Implementação 60
4.2.5 Avaliação 61
5. **APLICAÇÃO DO MEPOOMD** 62
5.1 Participantes e contexto de aplicação 62
5.1.1 Descrição dos participantes 62
5.1.2 Contexto de aplicação 68
5.1.3 Ferramentas empregadas 70
5.2 Fases da aplicação do MEPOOMD 76
5.2.1 Análise 76
5.2.2 Design 80
5.2.3 Desenvolvimento 82
5.3.4 Implementação 86
5.3.5 Avaliação 93
6. **ANÁLISE DE RESULTADOS** 95
6.1 Análise Qualitativa 95
6.2 Análise quantitativa 99
6.3 Avaliação das normas comportamentais aplicadas 105
6.4 Discussão 108
7. **CONCLUSÃO** 112
7.1 Contribuições da pesquisa 112
7.2 Trabalhos futuros 113
7.3 Considerações finais 114

8. **REFERÊNCIAS BIBLIOGRÁFICAS**..... 116

ANEXO I – Informações sobre a aplicação do Método MEPOOMD 123

ANEXO II – Questionário aplicado para avaliação de perfil dos alunos..... 130

ANEXO III – Questionário de verificação de nível de programação orientada a objetos: pré-teste e pós-teste a aplicação do MEPOOMD 133

ANEXO IV – Avaliação do ensino de POO 137

ANEXO V – Avaliação do MEPOOMD..... 141

ANEXO VI – Questionario avaliação MEPOOMD – Qualitativo..... 145

1. INTRODUÇÃO

O ensino de Programação Orientada a Objetos (POO) nos cursos de Computação e Tecnologia é um dos principais desafios para os alunos e professores, pois exigem, por parte do aluno, o desenvolvimento de capacidades de abstração para entender estruturas e modelos de programação. Este fato se tornou motivo de evasão e desistência de alunos no decorrer de cursos de programação. O processo de aprendizagem de POO tem se mostrado muito complexo e abstrato para os alunos. Isso gera falta de interesse por parte dos alunos, contribuindo para elevar os índices de desistência dos cursos relacionados ao desenvolvimento de software e programação.

Conforme observado por Rapkiewicz et al. (2006), os alunos que ingressam nestes cursos na maioria das vezes têm baixo nível de conhecimento prévio de programação, ou seja, tentam construir uma base de conhecimento relacionado a lógica e programação sem previamente criar uma estrutura mental para desenvolvimento e fixação de novos conceitos.

Tendo em vista esse contexto, faz-se necessário o desenvolvimento de métodos e artefatos para que o nível de abstração e complexidade no ensino de POO diminua e possibilite trabalhar com os alunos em diversos níveis de abstração. Esses métodos e artefatos podem prover recursos mais dinâmicos, tais como recursos de modelagem 3D, que não estão presentes nas aulas expositivas. Esse contexto é mais próximo a realidade do aluno (Luz, 2008).

1.1. Contexto e Motivação

O cenário, já conhecido pelos professores das disciplinas de programação, exibe muitas dificuldades nas fases introdutórias à programação, incluindo a barreira inicial da compreensão do que é construir um programa. Rapkiewicz et al. (2006, p. 1-2) descreve que a falta de motivação e a dificuldade de desenvolver um raciocínio lógico básico, necessário para a construção dos algoritmos, são os fatores principais da desistência dos ingressantes em programação.

Aprender a desenvolver ou programar utilizando Orientação a Objetos (OO) é complexo e exige muito esforço, força de vontade e métodos específicos de ensino. Uma vez que são exigidas muitas habilidades além de conhecer os detalhes da sintaxe do

código. As experiências em sala de aula demonstram que o problema tem sua base no início da aprendizagem, ao compreender os conceitos e como os mesmos devem ser aplicados, porém estes conceitos são totalmente abstratos e requerem um nível alto de raciocínio. Nesta fase de introdução e aplicação dos conceitos que devemos nos atentar não apenas em ensinar os conceitos e desenvolver as habilidades de programação, mas também na melhoria e aprimoramento das capacidades de resolução de problemas, lógica, raciocínio e tornar clara as abstrações baseadas em objetos para que o aluno entenda as relações de cada conceito.

Segundo *Bureau of Labor Statistics, U.S.* (2014) oportunidades de emprego para desenvolvedores está projetada para crescer 17% de 2014 a 2024, como é mostrado na Tabela 1. A necessidade de novas aplicações em dispositivos móveis e *tablets* irá ajudar a incrementar a demanda por desenvolvedores de software.

Tabela 1- Projeção de emprego para desenvolvedores

Ocupação	**Empregos, 2014**	**Perspectiva de empregos, 2024**	**Mudança, 2014 – 2024**	
			Porcentagem	**Quantidade**
Desenvolvedor de software	1.114.000	1.300.600	17 %	186.600
Desenvolvedor de software, Aplicações	718.400	853.700	19 %	135.300
Desenvolvedor de software, Sistemas	395.600	447.00	13 %	51.300

Fonte: *Bureau of Labor Statistics*, U.S. (2014)

A demanda por profissionais que trabalhem com desenvolvimento de software é enorme, mas infelizmente as vagas não são completadas pois os profissionais que se formam não estão prontos ou qualificados o suficiente para ingressar nesse mercado de trabalho. Nota-se também que há uma queda na procura ou interesse em cursar áreas relacionadas (Bell Witten & Fellows, 2011), isso ocorre, entre outros fatores, pela dificuldade de abstrair conceitos.

1.2. Problemática e Justificativa

Os níveis altíssimos de reprovação nas disciplinas de POO ocorrem em muitos casos devido a necessidade de aprender a lidar com o alto nível de abstração de conceitos. Nessas disciplinas o aluno deve conseguir criar soluções para situações em que estes conceitos devem ser colocados em prática. Segundo Campos (2010), há um nível muito elevado de abstração dentro destas matérias de desenvolvimento.

A programação se tornou uma ferramenta importantíssima (Ferreira, *et al*., 2012), mas há estudantes de programação que só aprendem a programar em uma determinada linguagem específica sem aprender os conceitos relacionados. Ou seja, aprender os conceitos que podem ser aplicados em vários tipos e linguagens de programação. Isso ocorre porque não há uma reflexão sobre a razão do porquê uma técnica ou elemento é aplicado em uma certa circunstância.

Segundo o *Department for education* (2013) do Reino Unido, a impressão 3D (e manufatura digital) deverá ganhar espaço no dia a dia e no ensino e aprendizagem nas instituições e o custo cairá consideravelmente conforme o aumento da procura por uma impressora 3D para projetos relacionados a projetos de engenharia. Entretanto, o professor deverá saber o que fazer com tecnologias 3D virtuais ou físicas, e ele deve incorporar esta ferramenta para o uso em suas metodologias de ensino.

Dentro deste contexto, a utilização de modelos 3D virtuais e impressão 3D podem criar formas que auxilie no entendimento de conceitos abstratos de POO. Essa abordagem pode propiciar meios para formar analogias entre os conceitos abstratos e os objetos físicos e/ou virtuais, melhorando assim o entendimento e aprendizagem dos alunos em tarefas relacionadas ao desenvolvimento de código OO.

1.3. Objetivos, Contribuições e Métodos

Os objetivos desta pesquisa é analisar e criar métodos e meios inovadores para o ensino de POO utilizando modelos virtuais 3D para facilitar a aprendizagem de conceitos abstratos.

A questão principal desta pesquisa é: “Como a utilização de objetos ou formas em 3D podem contribuir e aprimorar o modelo de aprendizagem de POO em cursos

técnicos e na graduação?". Serão avaliados casos práticos e reais com alunos de ensino técnico e graduação na área de tecnologia da informação em processo de formação.

Para tanto, o presente trabalho propõe a construção de um método de aprendizagem baseado em conceitos e técnicas da Semiótica Organizacional e do Design Instrucional. Enquanto o Design Instrucional estabelece fases bem definidas para o design de atividades com alunos, a Semiótica Organizacional permite uma melhor compreensão do relacionamento entre signos desenvolvidos no processo e sua interpretação por alunos e professores. Esse referencial então é utilizado, no método proposto, durante o design de atividades com alunos, de modo que explorem os conceitos de modelagem 3D para possibilitar a compreensão de conceitos complexos para os alunos. Por fim, tal método será avaliado em estudos empíricos com alunos do nível técnico e superior.

Espera-se com isso obter as seguintes contribuições principais:

(1) melhor compreender como a modelagem 3D pode auxiliar no processo de ensino de POO;
(2) proposição de um método para desenvolver atividades de ensino de POO utilizando modelagem 3D, tendo como referencial o Design Instrucional e a Semiótica Organizacional;
(3) obter dados empíricos sobre a aplicação deste método no ensino de POO, incluindo seus pontos fortes e fracos, limitações e identificação de desafios de pesquisa futuros.

1.4. Estrutura da Dissertação

Esta dissertação está organizada textualmente da seguinte forma:

- Capítulo 2: Neste capítulo é apresentado o referencial teórico e metodológico, incluindo assim conceitos, métodos e a dificuldade de aprendizagem de POO, bem como os métodos de modelagem e manufatura de objetos em 3D e a Semiótica Organizacional.
- Capítulo 3: Detalha os trabalhos relacionados a esta dissertação. Foi realizada busca em bases científicas e os resultados foram analisados e organizados de acordo com suas contribuições.

- Capítulo 4: Apresenta a elaboração e detalhamento do método proposto, incluindo suas principais etapas e atividades propostas.
- Capítulo 5: Relata a aplicação do método proposto em um estudo de caso, incluindo a aplicação do método em cursos de POO nos níveis técnico e superior.
- Capítulo 6: Apresenta e discute resultados da análise qualitativa e quantitativa do estudo de caso.
- Capítulo 7: Conclui esta dissertação.
- ANEXO I: Informações sobre a aplicação do Método.
- ANEXO II: Questionário aplicado para a avaliação do perfil dos alunos.
- ANEXO III: Questionário de verificação do nível de programação orientada a objetos.
- ANEXO IV: Questionário para avaliação do ensino de programação orientada a objetos.
- ANEXO V: Questionário para avaliação do MEPOOMD – Quantitativo.
- ANEXO VI: Questionário para avaliação do MEPOOMD – Qualitativo.
- ANEXO VII: Publicações Decorrentes da Dissertação.

2. REFERENCIAL TEÓRICO E METODOLÓGICO

O ensino de POO necessita de muita abstração por parte do aluno, pois os conceitos abordados por professores são comparados com objetos do mundo real, porém para o desenvolvimento e criação por parte do aluno além de muita prática ele necessita entender a teoria que fundamenta estes conceitos. Para melhorar o ensino de POO é necessário que sejam desenvolvidas novas práticas de ensino que estimulem o processo a aprendizagem do aluno por meio de melhor compreensão de conceitos abstratos e pela melhoria dos aspectos motivacionais. Este capítulo apresenta os fundamentos e métodos aplicados neste trabalho, enquanto o capítulo 3 apresenta os trabalhos relacionados.

A seção 2.1 apresenta o referencial teórico e metodológico sobre o ensino de programação para cursos técnicos e ensino superior, incluindo os principais conceitos e métodos de ensino, bem como a aplicação de Design Instrucional nesse contexto; a Seção 2.2 descreve o referencial sobre modelagem 3D e manufatura digital necessário ao desenvolvimento desse trabalho; enquanto, a Seção 2.3 introduz teorias e métodos da Semiótica Organizacional.

2.1. Aprendizagem de Programação

O processo de aprendizagem de programação é complexo, pois é exigido do aluno um nível elevado de abstração e entendimento de linguagem de programação. Pesquisas com mais de 30 anos já apontavam que programar nunca foi um trabalho fácil (Sleeman 1986), ainda mais para aqueles que estão ingressando no curso. Aliás, Gomes et al. (2008) mostram que os alunos ingressantes na maioria dos casos não contemplam as competências básicas para resolver problemas, e esta deficiência é agravada por métodos pedagógicos inadequados para o ensino. Frequentemente, ao iniciar o processo de aprendizagem ocorre um bloqueio por parte do aluno, nesse momento o ensino deveria ocorrer de uma forma dinâmica e simples. Entretanto, o conteúdo normalmente é exibido de forma estática e burocrática, isso não ajuda no aprimoramento das habilidades do aluno, com esta falta de entendimento temos a desmotivação que leva alguns alunos a desistirem no curso ao qual se propuseram a cursar.

A aprendizagem de programação no nível superior normalmente é apresentada de forma estruturada, onde a programação é entendida por meio da apresentação de conceitos, podendo ser entendido como o “modo tradicional” de ensinar programação

(Borges 1998). O modo que é ilustrado na Figura 1 é muito utilizado em várias áreas de conhecimento, mais especialmente nas graduações de tecnologia para o ensino de programação. Na Figura 1 temos as quatro etapas que este modo segue: (1) Apresentação da teoria, (2) Apresentação de exemplos práticos, (3) Proposição de exercícios práticos e (4) Proposição de projetos mais complexos.

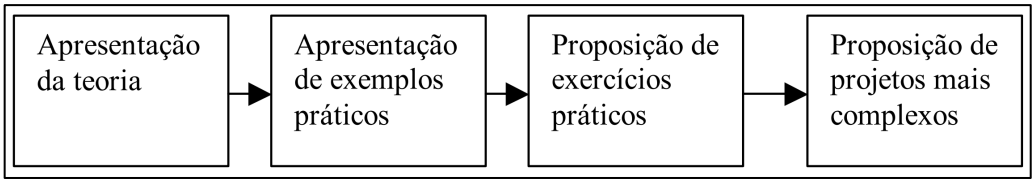


Figura 1 - Sequência de passos na apresentação de conteúdo.

Este método de ensino de programação não é suficiente para despertar o interesse do aluno pelo conteúdo. Os alunos que ingressam em cursos de programação têm mais dificuldade pois estão no início do processo, começando a vislumbrar os conceitos de programação, e a situação consegue ser pior para os que não têm experiência profissional na área. Esta falta de motivação dos alunos se torna crítica quando visualizamos que os conceitos são exibidos sem fazer uma correlação a objetos ou coisas do dia a dia, isto torna muito difícil a compreensão por parte do aluno para perceber o propósito de cada conceito.

Ao decorrer de todo o processo de aprendizagem no ensino de programação é notado que a maioria dos alunos demostra muita dificuldade em entender todos os conceitos abstratos que a programação envolve. Segundo Rodrigues (2002), a disciplina de programação na graduação tem um alto índice de reprovação, fazendo com que isto torne um ponto de reflexão para os professores que ficam aflitos fazendo tentativas de aprimoramento dos métodos de aprendizagem de programação.

Embora fosse uma tentativa válida, o uso de POO como primeira disciplina de programação para mitigar dificuldade que os alunos encontram ao iniciar estudos em lógica de programação não obteve os resultados que eram esperados por professores e instituições. A POO, por exemplo, utiliza encapsulamento e manipula os dados com declarações e armazenamentos, isso impossibilita o acesso ao código, desta forma o aluno não consegue visualizar a solução do problema de uma forma clara (COX, 1999).

Nas disciplinas de POO é apresentado para o aluno um conteúdo teórico, extenso e totalmente baseado em abstrações de objetos com a intenção de localizar suas classes e

objetos. Segundo Santos (2003), atributos e métodos podem ser descritos como propriedades comuns de uma coleção de objetos, formando assim uma classe. Os objetos são responsáveis por fazer ações especificas. Entretanto, para fazer com que o aluno entenda estes conceitos, é necessário fazer uma abstração lembrando de coisas do cotidiano ou do mundo em que vive para localizar e reconhecer esses objetos. O mundo que me refiro é formado por "coisas", que na terminologia de OO, isto é chamado de objeto. Os seres humanos agrupam estes objetos para reconhece-los ou entende-los. Com isso, os detalhes de um grupo de objetos são associados a uma classe. Além disso, todo o reconhecimento das classes é realizado a partir da observação de objetos, exigindo um nível alto de abstração (Santos, 2003).

A abstração é muito complexa para o aluno iniciante, pois há necessidade de identificar todas as "coisas" como objetos e após isso organizá-las de acordo com suas correlações com determinadas classes. Nesta etapa, os alunos apresentam muita dificuldade e os professores não conseguem ou não têm os meios para auxiliar o aluno e conduzi-lo à melhor forma de aprendizagem. Normalmente, o professor faz com que o aluno pratique em alguma linguagem de programação para realizar o reconhecimento dos elementos.

Com a intenção de aprimorar o ensino de POO, existem várias abordagens e ferramentas, como ensino baseado em jogos (Rais *et al.,* 2011) e o desenvolvimento de metodologias de ensino, tais como o iPOO que é um modo de ensino em ciclo de funções (Lopes, 2007). Uma análise feita com as soluções mais recentes exibe a tendência de fazer com que o aluno fique no centro de todo o processo de aprendizagem, fazendo com que o mesmo esteja de forma ativa no processo. Neste sentido, segundo Li *et al.,* (2009) as metodologias de ensino de POO precisam sempre utilizar o aluno como sujeito principal no processo de aprendizagem.

2.1.1 Design Instrucional no ensino de programação

Ao analisar o ensino de programação, Rais *et al.* (2011) aponta que boa parte da dificuldade apresentada por alunos no momento da aprendizagem, ocorre devido ao elevado nível de abstração e detalhamento de conceitos apresentados por professores. Porém é possível compreender que temos ferramentas tecnológicas que podem

contribuir para o aprimoramento do processo de ensino aprendizagem, dando aos discentes a oportunidade de inovar, ser criativo e complementar o processo de desenvolvimento do aluno. Segundo Filatro (2004), a intenção da pesquisa e o campo de aplicação do design instrucional é desenvolver a criatividade e os aspectos humanos e sociais.

O design instrucional pode ser compreendido como a utilização sistemática de métodos de desenvolvimento e planejamento em técnicas e atividades para o ensino e projetos educacionais que sejam apoiados por alguma tecnologia (Filatro, 2004). Este modelo é utilizado para estruturar as fases do método proposto nesta dissertação.

Por meio de uma estrutura planejada de ensino-aprendizagem, o design instrucional é estruturado em fases. Um modelo muito adotado nesta área é o ADDIE (*Analysis, Design, Development, Implemention and Evaluation*). Este modelo, segundo Filatro (2008), é separado em fases para concepção e execução, como apresentado na Figura 2:

1. **Análise:** localização e identificação de requisitos para que ocorra a aprendizagem, a definição de objetos instrucionais e levantamento de regras ou restrições.
2. **Design e desenvolvimento:** Momento no qual ocorre o planejamento das instruções e toda a elaboração de materiais e produtos para realizar a instrução de ensino.
3. **Implementação:** Quando ocorre a aplicação com docentes e ensino a alunos na proposta de design instrucional e a realização de ensino-aprendizagem.
4. **Avaliação:** Todo o acompanhamento, revisão e manutenção do método proposto.

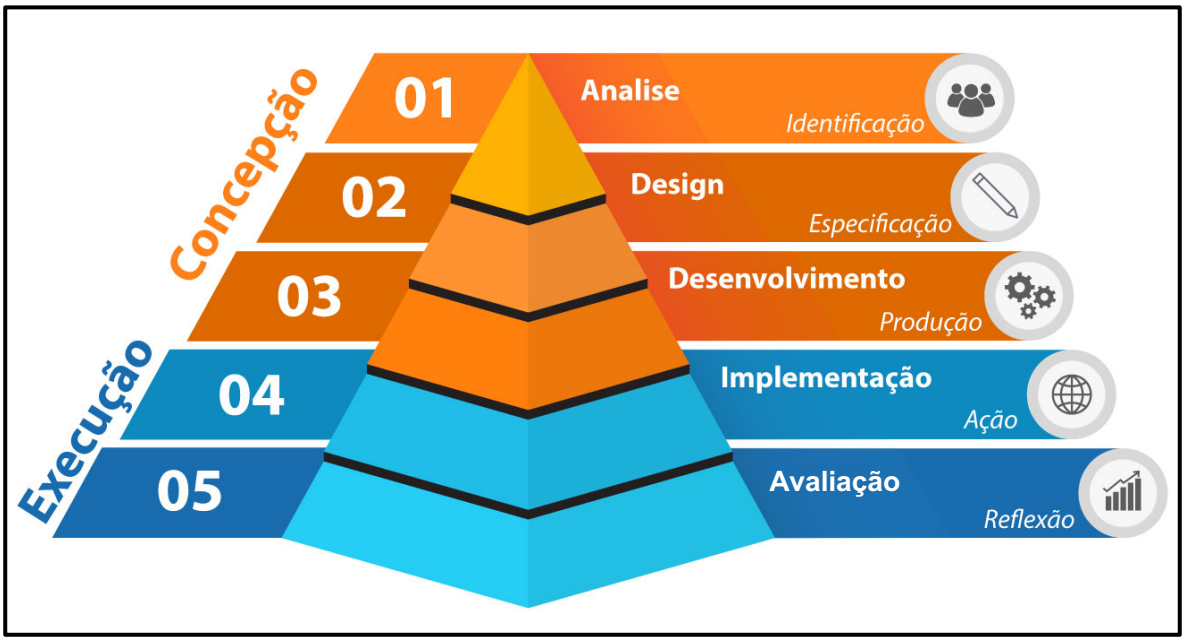


Figura 2 - Etapas do processo de Design Instrucional

Adaptado de Filatro (2008)

Muitos modelos de design de atividades educacionais apresentam estas etapas de análise, design, desenvolvimento e avaliação como apresentado na Figura 2, entretanto no design instrucional estas etapas são contextualizadas e essas operações ocorrem recursivamente durante o processo (Filatro, 2008). Desta forma, no decorrer do processo, o design instrucional consegue trabalhar com profissionais de diferentes áreas como ilustradores, designers, programadores, etc. Uma das principais conceituações é que existe uma ótima comunicação entre as diferentes etapas que se tornam conclusões e finalizações de qualidade. Muitos projetos propostos utilizando design instrucional, com etapas detalhadas com qualidade no processo, obtiveram ótimos resultados ao aplicar o método, a seguir são relatadas duas experiências diferentes onde a aplicação do modelo ocasionou ganho significativos com a utilização do método de design instrucional.

No trabalho proposto por Dantas (2013), por exemplo, foi desenvolvido um jogo chamado Robotmov, o cenário proposto foi o de um bar em que os robôs estivessem se divertindo. Os Robôs devem se esquivar dos obstáculos, o herói precisa escapar dos ataques dos robôs inimigos e coletar as engrenagens perdidas, necessárias para completar o desafio de programação da fase, possui um elaborado design instrucional e combina bem elementos de jogabilidade. Este projeto teve como proposta uma variedade de desafios de programação para atender todas as etapas cognitivas do aprendizado, integrados diretamente com o *Game Design* da aplicação. Este trabalho comprova a

importância que a aplicação motivacional com imagens e formas exemplificam e diminuem a abstração, motivando os alunos a praticarem seus conceitos de forma lúdica e divertida continuamente enquanto isso aprendem praticando utilizando o modelo de design instrucional.

Em Battistella e Von Wangenheim (2016) foi desenvolvido um processo de desenvolvimento de jogos educacionais utilizando a linguagem de programação chamada ENgAGED (*EducatioNAl GamEs Development*). Este é um processo interativo para o desenvolvimento de jogos baseado e fundamentado no design instrucional ADDIE. Com a intenção de auxiliar o desenvolvimento de jogos educacionais, até mesmo por alunos o método desenvolvido obteve resultado de boa qualidade.

Baseado na premissa que projetos nos quais a aplicação de design instrucional ADDIE obtiveram ótimos resultados, pensando que o método desenvolvido na maioria das vezes é útil para várias situações dentro do tema proposto, pois o método criado é considerado consistente, completo, totalmente compreensível e correto.

2.1.2 Programação Orientada a Objetos

Nas linguagens de programação atuais como Java, C++, PHP, Python e .NET o principal tópico é POO na Ciência da Computação. Existem vários aspectos que necessitam de um cuidado redobrado para que o aluno pense orientado a objeto, como classes, objetos, associação, encapsulamento, polimorfismo e herança, que são conceitos primordiais para o entendimento (Börstler, Bruce e Michiels, 2003, p. 84).

Em paralelo a criação da linguagem de programação Smalltalk, foi criada a POO por Alan Kay em 1972. O conceito principal na OO é abstrair ou imaginar que programas simulam o mundo real em que vivemos, um mundo que está cheio de objetos. Desta forma, as linguagens que trabalham os conceitos de simulação do mundo real precisam incluir um modelo de objeto que consiga receber e enviar mensagens e também consiga ter alguma reação em relação a mensagem. No mundo real, necessitamos saber como os objetos funcionam, desta forma a POO cria um ambiente com muitos objetos e eles podem conviver e trocar mensagens entre si (Sebesta, 1988).

Em uma visão OO, nós compreendemos o mundo como um todo que é composto por muitos objetos que interagem entre si, da mesma forma, a POO considera que os

sistemas computacionais não como uma coleção estruturada de processos, mas sim como uma coleção de objetos que interagem (Lewis e Loftus, 2008).

Segundo Mitchell (2003), a Orientação objetos consiste em conceber um sistema computacional como um todo orgânico formado por objetos que se relacionam entre si. Pode ser aplicada tanto na análise de sistemas quanto na programação, e essa é uma das principais vantagens da OO. A mesma metodologia serve tanto para a definição de lógica do sistema quanto para a sua implementação. A POO possui cinco vantagens de sua utilização: (1) Unificação entre dados e processos, (2) Consistência na análise e desenvolvimento, (3) Reutilização e aumento da produtividade, (4) Multidesenvolvimento, e (5) Facilidade em manutenção.

A partir da premissa de que a POO traz benefícios para o desenvolvimento de software e é a tecnologia atual a ser aprendida, a seguir são apresentados os principais conceitos de POO a serem trabalhado na aplicação do método MEPOOMD.

Objetos, Classes, Hierarquia e Instanciação

O termo objeto é utilizado para representar qualquer tipo de elemento do mundo real. Objetos físicos ou abstratos representados em um sistema computacional são instâncias de classe. As classes determinam o tipo de informação que o objeto irá ter e como podemos utilizar o mesmo. É por meio dos objetos que todo o processamento ocorre nos sistemas implementados em POO (Kindler e Krivy, 2011). Um animal como uma "arara azul" por exemplo, pode ser representado por um objeto em um sistema computacional. Esse objeto possui valores associados as suas características como: Nome (ex: Bicudo), Idade (ex: 13 Anos), Tamanho das Asas (ex: 67 cm), Cor das penas (ex: Azul), Cor dos olhos (ex: Preto), Peso (ex: 8 Kg), etc. Estas características são chamadas de atributos, pois descrevem o que o objeto contém (Mitchell, 2003). Neste mesmo objeto existem algumas ações que ele consegue executar, como: voar, comer, dormir, sentar, etc. Estas ações que esse objeto é capaz de realizar são chamadas de métodos ou serviços. É possível fazer com que os objetos tenham interação, e para utilizar esta interação é necessário a utilização dos métodos contidos pelo mesmo por meio de interfaces (Pierce e Benjamin, 2002).

A classes são organizadas em uma hierarquia, onde subclasses herdam atributos e métodos das superclasses. Por exemplo, conforme destaca a Figura 3, *Cães* e *Gatos*

possuem métodos distintos, mas compartilham atributos e métodos comuns de *Mamíferos* que for sua vez compartilham atributos e métodos comuns a todos os *Animais*.

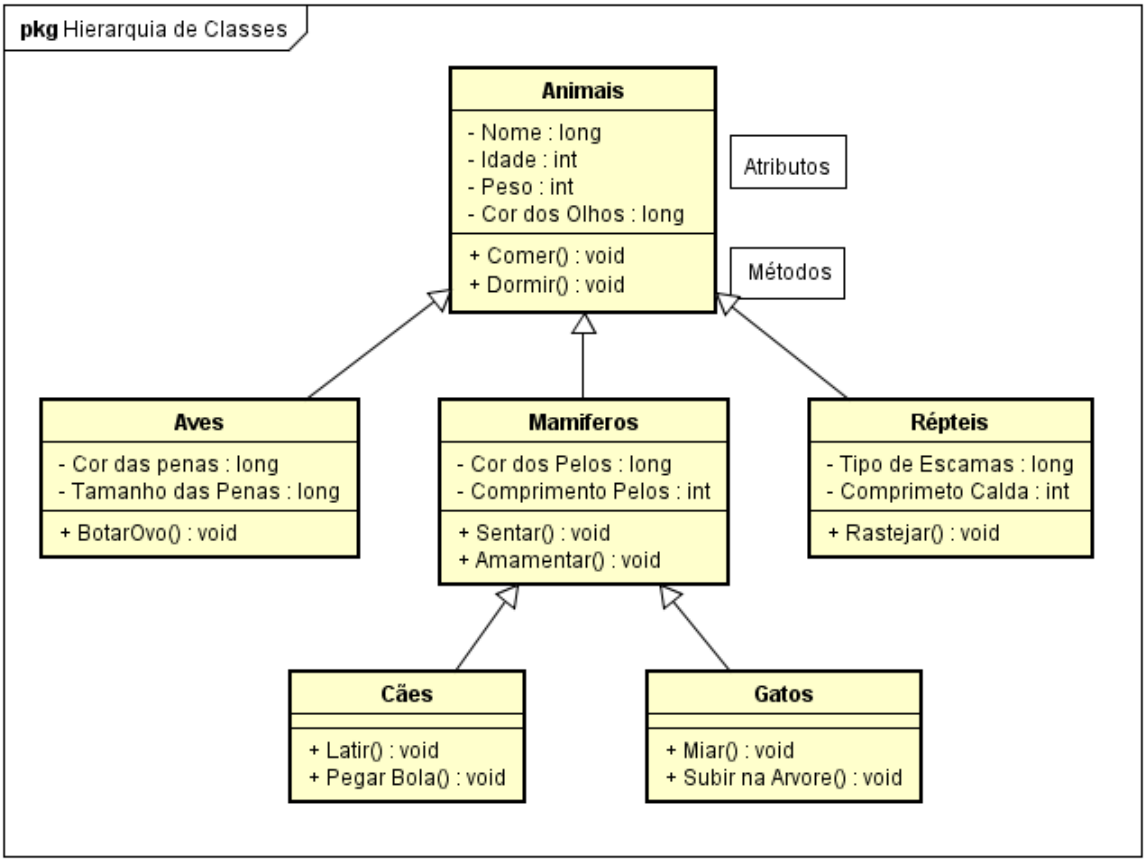


Figura 3 – Exemplo de Hierarquia de Classes

Outro conceito aplicado em POO é o de instanciação, que ocorre quando a classe "produz" um novo objeto. Na OO, um objeto pode ser entendido como uma instância de uma das classes.

Abstração, Encapsulamento, Herança e Polimorfismo

Segundo o dicionário Aurélio (Ferreira, 2010, p. 27), a palavra abstração como definição é o "ato de separar mentalmente um ou mais elementos de uma totalidade complexa (coisa, representação, fato), os quais só mentalmente podem substituir fora dessa totalidade". Também conhecida como ocultamento, abstração é o princípio pelo qual cada componente deve manter oculta sob sua guarda uma decisão. Para a utilização

desse componente, apenas o mínimo necessário para a operação deve ser revelado ou torando público (Mitchell, 2003).

Nós, como seres humanos, possuímos o recurso de abstração como uma forma de entender problemas que achamos complexos. Segundo Pierce e Benjamin (2002), quando encontramos ou estamos diante de um problema complexo tentamos dividir em problemas menores, para desta forma resolver cada um deles até encontrar a solução completa do problema. Este termo “abstração”, quando é aplicado a desenvolvimento de sistemas, significa que só deve ser representado o que realmente será utilizado.

A utilização de abstração também permite que determinadas partes do problema ganhem maior ou menor peso, desta forma, os detalhes que foram desconsiderados em outros momentos para que outros sejam ressaltados (Pierce, Benjamin, 2002). Por exemplo, métodos da classe arara, como voar, comer, dormir e sentar não são importantes para um sistema de inventário de animais de um zoológico, e portando, não devem ser representados. Muito provavelmente só os métodos para manipulação de dados devem ser representados, bem como os atributos pertinentes a um sistema desse tipo. Por outro lado, se o objetivo for criar um avatar, os métodos voar, comer, dormir e sentar devem ser representados.

Em OO atributos são encapsulados e só podem ser alterados por meio dos métodos que respondem a estímulos/eventos internos ou externos (Pierce e Benjamin, 2002). O encapsulamento é o princípio ou motivo pelo qual os componentes do programa devem agregar toda a informação relevante para sua manipulação como uma unidade ou cápsula (Lewis e Loftus, 2008). Com o conceito de ocultamento de informação, o encapsulamento é um poderoso mecanismo em POO. A vantagem do encapsulamento é que ele disponibiliza o objeto com toda a sua funcionalidade sem que seja necessário saber como ele funciona internamente, nem como armazena internamente os dados que talvez sejam necessários (Lewis e Loftus, 2008). Por exemplo, para telefonar para alguém você simplesmente pega o telefone e disca o número de destino, isto ocorre sem que você saiba os procedimentos que ocorreram na empresa telefônica que conectou você ao destino. O encapsulamento também permite realizar alterações internas em um objeto, acrescentando métodos sem afetar os outros componentes do projeto que também utilizam este mesmo objeto (Armstrong, 2012).

Já a herança é um mecanismo da POO pelo qual uma classe obtém as características e métodos de outra para conseguir expandir ou especializa-la de alguma forma, sendo assim uma classe pode herdar características, métodos e até atributos de outras classes (Mitchell, 2003).

Pensando no ponto de vista prático da OO, a herança é um mecanismo muito inteligente de aproveitamento de código. Por meio da herança que os objetos podem compartilhar métodos e atributos, só assim é possível criar uma nova classe (Mitchell, 2003). A vantagem é reutilizar o código que já foi implementado na classe pai, restando apenas implementar os métodos e atributos que tem diferença da classe pai. Assim, a herança então significa que todos os atributos e métodos programados no ancestral já estarão automaticamente em seus descendentes sem necessidade de reescrevê-los (Mitchell, 2003). Segundo Pierce e Benjamin (2002), a herança mais do que apenas economia de código, também significa maior integridade, pois quando se altera um comportamento em uma classe, todas as classes descendentes da mesma também estarão utilizando o método atualizado sem a necessidade de reprogramação.

Outro conceito muito importante na POO é o polimorfismo. Segundo Kindler e Krivy (2011, p. 263) “Polimorfismo é a capacidade de uma variável se referir em tempo de execução a objetos de diversas classes. Ele ocorre quando um método já foi definido no ancestral é redefinido no descendente com um comportamento diferente”. O polimorfismo pode tanto estar ligado ao conceito de sobreposição (como citado) quanto ao conceito de sobrecarga, *i.e.,* onde métodos com mesmo nome, mesmo que em uma mesma classe, assumem comportamentos diferentes de acordo com seus parâmetros. Uma câmera fotográfica ou filmadora e um computador, por exemplo, possuem a mesma função “Gravar”, mas está função é realizada de forma diferente por cada um dos objetos.

2.1.3 Dificuldades de Aprendizagem de conceitos de POO

Embora os conceitos descritos na seção anterior sejam relativamente simples para programadores com experiência em POO, eles são fontes de grandes dificuldades no processo de aprendizagem de alunos iniciantes. Nas universidades o ensino de programação tem se tornado um assunto de discussões em vários trabalhos científicos recentes, como, Amaral (2015), Eagle e Barnes (2008), Ambrosio (2015), Ramos (2015), Galdino (2015) e Hernandez (2010). Estes trabalhos relatam e reconhecem a programação como um assunto complexo que envolve um nível elevado de conhecimentos prévios e relatam a preocupação com os elevados índices de desistência nos cursos de computação que envolvem desenvolvimento e programação.

Ensino e aprendizagem de programação exige um nível altíssimo de concentração e envolvimento por parte do aluno, por este e outros motivos a quantidade de desistência pode chegar a 40% de alunos na disciplina introdutória a programação, ou nos primeiros semestres (Eagle e Barnes, 2008). Segundo Galdino (2015) este nível elevado de evasão é muito preocupante, levando em conta que os alunos relatam dificuldade com a lógica de programação, e dos alunos avaliados a maioria não consegue desenvolver um programa ou código razoável após semestres de estudos em disciplinas. Outro problema encontrado relacionado ao ensino são os livros de programação que abordam apenas processos de programação, problemas típicos de desenvolvimento, porém sem contextualização do problema envolvido (Caspersen and Bennedsen, 2007).

Segundo Galdino (2015), o estudo sobre ensino e aprendizagem de programação apresenta evidência de que:

- Conceitos de programação como metodologia baseada puramente em sintaxe é prejudicial para o processo de aprendizagem dos alunos.
- A falta de empenho por parte dos alunos é devida as dificuldades com as matérias introdutórias de programação, pois não conseguem criar os corretos modelos mentais acerca do conhecimento que está sendo aprendido.
- Existe uma grande necessidade de novas propostas metodológicas, as quais não sejam baseadas puramente em código textual, e sim em contextualização.

2.2. Modelagem e Fabricação Digital na Educação

A cada década ou século, um novo conjunto de habilidades e atividades intelectuais se tornam cruciais para o desenvolvimento de tarefas profissionais e cotidianas. Com o avanço rápido da computação no início dos anos 60's, a programação de computadores se tornou uma dessas novas atividades (Papert, 1991). Mas os computadores nesses anos eram máquinas grandes, caras, estranhas, e a ideia de utilizá-los como um meio de expressão e de aprendizagem pessoal era pouco difundida. O estabelecimento educacional aboliu a ideia de programação como um objetivo pedagógico fundamental: era muito difícil para as crianças, jovens e adultos melhorar o nível de aprendizagem em matemática e ciências ou qualquer que seja a área utilizando a computação.

No entanto, desde os anos setenta, os pesquisadores têm trabalhado duro criando ferramentas para tornar a programação mais fácil de aprender. Ferramentas de programação como Scratch (Resnick et al., 2009) e NetLogo (Wilensky, 1999) conseguiram uma popularidade sem precedentes e tornaram a codificação acessível a milhões de estudantes e professores. O mundo entendeu a ideia de que não só a mídia computacional poderia ser um veículo para ideias poderosas em matemática, engenharia e ciência, mas também um novo tipo de alfabetização e uma atividade acessível nas escolas.

Segundo Anderson (2012) estamos em outro momento de mudança de paradigma. As novas tecnologias de Fabricação Digital serão um dos processos de muita importância do século XXI, uma vez que elas propiciam processos alternativos de produção.

Segundo Miyaska e Fabricio (2015) os primeiros casos de Fabricação Digital estavam voltados à fabricação de tapetes e cortinas na indústria têxtil no começo do século XIX. Com máquinas programáveis era possível tecer com alta qualidade de produção e um custo muito baixo por unidade. Os comandos eram transmitidos para a máquina por pequenos buracos em um papel cartão que sinalizavam comandos positivos ou negativos (binários) para executar a sequência de ações. Com a evolução esta codificação tornou-se totalmente digital, porém muitos princípios permanecem.

Com a evolução das tecnologias de computação e manufatura aditiva é possível criar processos de produção totalmente programáveis e automáticos. Por algumas

questões econômicas nunca foi algo viável no passado, porém a situação é inversa agora, pois a economia para a fabricação tradicional esta invertida (Anderson, 2012).

A fabricação digital poderia ser um novo e importante capítulo neste processo de ideias, alfabetização e ferramentas expressivas para os jovens. Hoje, a gama de conhecimentos disciplinares aceitos expandiu-se para incluir não só programação, mas também engenharia e design (Astrachan, Hambrusch, Peckham, & Settle, 2009; Yasar & Landau, 2003). Além disso, há a necessidade de novas abordagens educacionais que promovam a criatividade e a inventividade.

Segundo Miyaska e Fabricio (2015), a utilização de fabricação digital como ferramenta de ensino tem um nível alto de importância, pois estas ferramentas conseguem ampliar as capacidades cognitivas, motora e de desenvolvimento por parte do aluno.

A analogia com o desenvolvimento do Logo é clara: simultaneamente, as tecnologias de fabricação digital tornaram-se melhores e mais acessíveis, e as atividades intelectuais permitidas pela nova tecnologia tornou-se cada vez mais valorizada e importante. O que o Logo fez para a geometria e programação trazendo matemática ao alcance dos alunos, laboratórios de fabricação podem fazer para design e engenharia, essa é a intenção da fabricação digital (Clements, 1987).

Segundo Blikstein (2013), no final da década de 2000, pesquisadores e educadores começaram a considerar o uso de fabricação digital na educação. Em 2008, a Universidade de Stanford lançou o projeto FabLab @ School e começou a FabLabs em escolas K-12 em todo o mundo. Em 2009, o MC2STEM High School em Ohio (EUA) abriu seu primeiro laboratório de fabricação digital. Em 2011, a Maker Media lançou o projeto MakerSpace com financiamento da DARPA para que vários museus, escolas, centros comunitários e bibliotecas pudessem projetar construções de laboratórios para a fabricação digital. Esse foi um impulso importante para o ressurgimento de laboratórios de fabricação e "*making*" em ambientes formais e informais, que hoje são baseadas em três pilares pedagógicos:

- Educação experiencial;
- Criação;
- Pedagogia crítica.

Papert (1980; 1987; 1991) defende a tecnologia nas escolas não como uma forma de otimizar a educação tradicional, mas como uma ferramenta emancipatória que coloca os materiais de construção mais poderosos nas mãos de crianças e jovens, tornando-os "criadores". Essa adaptabilidade incorporada na tecnologia, permite reconhecer e abraçar diferentes estilos de aprendizagem e epistemologias, gerando um ambiente convivial em que os alunos possam concretizar suas ideias e projetos. Em um ambiente de aprendizagem típico, raramente há um currículo fixo. Os alunos usam a tecnologia para construir projetos e os professores atuam como facilitadores do processo.

> *A linguagem de programação do Logo foi a primeira tentativa de educação para demonstrar que o computador não é apenas um dispositivo de informação e comunicação, mas também uma ferramenta expressiva para construção e auto expressão (Blikstein, 2013).*

No início dos anos noventa, Papert, Mitchel Resnick e Fred Martin estenderam as poderosas ideias de Logo para o mundo físico, tornando robótica acessível às crianças através do Lego Mindstorms Kit e o Cricket (Martin, 1994; Martin & Resnick, 1993) e, em conjunto com colaboradores, extenso trabalho em robótica e *'making'* workshops usando micro controladores e sensores (Resnick, Berg, & Eisenberg, 2000). Sipitakiat e Blikstein alargaram este trabalho aos países em desenvolvimento e as comunidades, trabalhando com hardware de baixo custo, bem como materiais reutilizados (Blikstein, 2008; Sipitakiat, 2000; Sipitakiat, Blikstein, & Cavallo, 2002, 2004). Mais recentemente, colocando novos hardware e software de ponta nas mãos de crianças para conduzir pesquisas científicas avançadas explorações (Blikstein, 2010; Blikstein, Fuhrmann, Greene, & Salehi, 2012), criam vestuarios interativos (Buechley, 2006), Buechley e Eisenberg, 2008, Buechley, Eisenberg, Catchen, & Crockett, 2008), construir eletrônica (Perner-Wilson, Buechley, & Satomi, 2011; Sylvan, 2005), projeto participativo simulações e jogos (Wilensky & Stroup, 1999), videogames de programas (Millner & Resnick, 2005; Kafai, 1995), projetar sistemas robóticos virtuais (Berland, 2008; Berland & Wilensky, 2006), criar mundos 3D sofisticados e jogos através da programação (Cooper, Dann, & Pausch, 2000), construir novos tipos de criaturas cibernéticas (Raffle, Parkes, & Ishii, 2004; Schweikardt & Gross, 2006) e explorar ciência ambiental e sistemas de informação geográfica (Edelson, 2000). Na Figura 4

podemos visualizar a linha do tempo onde conceitos que precedem a Fabricação Digital atual têm sido empregados como estratégia de ensino resultado da pesquisa de trabalhos relacionados com esta dissertação.

Figura 4 - Linha do tempo de tecnologias computacionais chaves na educação

Estes kits ou tecnologias propiciaram os fundamentos para o atual movimento de educação por meio da Fabricação Digital. Estes projetos e pesquisas demostraram que

era possível envolver os alunos em atividades que exploram a tecnologia de forma plena, que esses mesmos alunos poderiam ativamente construir com a tecnologia em vez de apenas consumir produtos tecnológicos. Eles revelaram como as ideias e desenvolvimento intelectual dos alunos podem ser altamente poderosos. Em vez de desenvolvimentos aleatórios, estas tecnologias, materiais e kits de ferramentas foram profundamente influenciados pelas construções teóricas Dewey, Papert e Freire, em torno de usos construtivos da tecnologia, educação culturalmente consciente, aprendizagem experiencial, e currículos de interesse.

Segundo Bell *et al.* (2010), a fabricação digital envolve a transformação de um desenho digital em um objeto físico por meio de um sistema de fabricação controlado por computador. A fabricação digital pessoal torna isso viável em um computador pessoal. É possível a criação de um projeto bidimensional ou tridimensional utilizando estes computadores. Com um software adequado, ou um modelo preexistente, o projeto pode ser transferido na internet. Seu desenho digital é usado para fabricar uma versão física tridimensional do projeto eletrônico, esta etapa é denominada por alguns como um processo de "bits para átomos". Imagens digitais ou projetos tridimensionais de um conjunto de engrenagens, por exemplo, podem ser usados para criar um conjunto físico de engrenagens feitas de plástico, metal ou outro material. Estas novas maneiras de fabricação exigem muitas mudanças na manufatura e engenharia; tanto fabricantes e consumidores estão mudando a maneira como conhecemos e entendem a manufatura.

A pesquisa realizada por Smith, Iversen e Hjoth (2015), por exemplo, revelou que o processo de combinar pensamento abstrato e construir o concreto está longe de ser algo que ocorre automaticamente. Os autores analisaram estudantes com idades entre onze e quinze anos e concluíram que os estudantes não entenderam *a priori* o processo criativo e reflexivo subjacente à fabricação digital. Isto afetou a capacidade de executar determinadas tarefas, bem como suas habilidades para visualizar soluções pessoais ou coletivas para definir desafios de design.

Segundo Tillman (2012), a missão da *Fab@School* (escola que utiliza os métodos de Fabricação Digital para o ensino) é a introdução da fabricação digital na educação como tecnologia instrucional. Alguns dos contextos educacionais perseguidos pela *Fab@School* são ambientes de aprendizagem informais, local de desenvolvimento e criação, desenvolvimento profissional e formação contínua de professores.

Tillman (2012) propôs uma escala de prioridades para fabricação digital no ensino, destacando os quatro tópicos a seguir:

- Desenvolvimento e aperfeiçoamento de currículos;
- Aprimoramento e desenvolvimento de docentes para aplicação e testes de conceitos;
- Implementação de instrumentos para avaliação, bem como análise de dados;
- Continuação e aprimoramento do Hardware e Software.

A fabricação digital na educação, não é apenas orientada para a aprendizagem utilizando algumas atividades de design e construção com tecnologias como impressoras 3D e cortadores a laser. Além disso, Smith, Iversen e Hjoth (2015), enfatizam a concepção para a fabricação digital como um ambiente de aprendizagem híbrido que combina fabricação digital, pensamento de design e inovação colaborativa para resolver desafios sociais complexos. Essa definição enfatiza toda a criação do processo desde ideias precoces, esboços e criação de maquetes até a apresentação inicial de um protótipo, a fabricação se torna um veículo e recurso para abordar questões pessoais ou complexas da sociedade.

> *Quando novas ideias são capturadas por uma criança, elas são primeiramente reconstituídas para as estruturas mentais da criança. Só mais tarde, através da interação de muitos desses elementos, as próprias estruturas mudam numa fase que Piaget chama de acomodação. Fico espantado com o modo como os educadores que tentam seguir as ideias de Piaget ao pensar nas crianças não entendem que a mudança na escola ou em qualquer outro sistema complexo deve ocorrer da mesma maneira. (Papert, 2005, p. 366).*

A aplicação de modelagem em laboratórios tem feito a diferença assim como o software de programação virtual como o Alice e o Scratch. O Alice fornece um ambiente de programação 3D animado que suporta uma abordagem para o ensino da resolução de problemas da maneira específica usada na programação (Cooper, et al, 2006). Alice foi utilizada para desenvolver algoritmos para animar objetos que povoam mundos virtuais.

Foi observado que os alunos ficavam confortáveis usando objetos e invocando métodos sobre esses objetos. Os alunos foram capazes de verificar o que deu errado em seus programas e facilmente depurar e corrigi-los. Já no Scratch, segundo Smith, Iversen e Hjorth (2015), o aluno pode programar as suas próprias histórias, jogos e animações interativos e partilhar as suas criações com outros na comunidade em linha ajudando os jovens a pensar de forma criativa, a raciocinar sistematicamente e a trabalhar colaborativamente competências essenciais à vida. O Scratch é um projecto do Lifelong Kindergarten Group do MIT Media Lab. O uso de software como estes em conjunto com a fabricação digital pode trazer um ambiente mais rico para aprendizagem multidisciplinar que inclua conceitos de programação e fabricação em práticas direcionadas.

Segundo Bell (2011), a revolução da fabricação digital já começou em muitos aspectos. Indivíduos com acesso à Internet e uma habilidade moderada com software de design têm a capacidade de produzir seus próprios gráficos impressos, tais como canecas e relógios de parede por meio de serviços baseados na Web, como CafePress (http://www.cafepress.com) e Zazzle (http://www.zazzle.com). O site Lulu (http://www.lulu.com) oferece aos usuários a oportunidade de publicar seus próprios projetos vinculados, já o Ponoko (http://www.ponoko.com) suporta a fabricação de uma variedade de itens, incluindo brinquedos, joias e móveis.

A fabricação digital tem o potencial de ser um kit poderoso de construção metodologica, com um papel principal e primordial em escolas onde os alunos podem fazer, construir e compartilhar suas criações com segurança (Blikstein, 2013). Com espaços planejados e protegidos para serem convidativos, eles podem atrair tanto os interessados em engenharia e tecnologia, mas também os alunos que só queriam experimentar um projeto com tecnologia, ou melhorar algo que eles já estavam fazendo com a fabricação digital. Abaixo temos a lista de aspectos elencados por Blikstein (2013) ao considerar a Fabricação Digital na educação:

- **A "Síndrome do Chaveiro (*Keychain)*":** uma vez que as máquinas de fabricação digital podem gerar produtos esteticamente agradáveis com pouco esforço, os educadores devem evitar os projetos de demonstração rápida e induzir os alunos para empreendimentos mais complexos;

- **O poder do envolvimento intenso:** os FabLabs proporcionam um ambiente para experiências de design forte e intenso sem precedentes, eles podem levar para múltiplos ciclos de design e novos níveis de frustração e emoção, que os alunos normalmente não experimentam na sua experiência escolar normal;
- **Poderosos projetos interdisciplinares:** os limites artificiais entre disciplinas são completamente reconfigurados no laboratório. A história e a matemática tornam-se intimamente relacionadas, assim como a música e a robótica, e essa riqueza resulta em um ambiente intelectual mais diverso e mais receptivo;
- **Aprendizagem contextualizada em STEM (*Science, Technology, Engineering and Mathematics*):** os alunos têm a oportunidade de se deparar com vários conceitos de engenharia e ciência de uma forma altamente significativa, envolvente e contextualizada. Ideias abstratas tais como a fricção e impulso se tornam significativa e concreta quando elas são necessárias para realizar uma tarefa dentro de um projeto, tal como quando a matemática se torna uma necessidade em um projeto de história;
- **Os alunos trazem suas próprias práticas familiares (artesanato, construção, carpintaria) para o laboratório:** essas práticas são aumentadas usando ferramentas socialmente valorizadas, tais como como conhecimento computacional e matemático. A maleabilidade dos equipamentos e do espaço pedagógico no laboratório possibilita o aumento e o acolhimento de tais práticas, gerando um ambiente que valoriza múltiplas formas de trabalho.

Apesar do potencial dos laboratórios de fabricação digital na educação, educadores e estudiosos devem lembrar que o verdadeiro poder de qualquer tecnologia não está na própria técnica ou no fascínio que ela gera, mas nas novas formas de expressão pessoal que possibilitam, as novas formas de interação humana, facilitando que as ideias tornem acessíveis aos alunos (Bell ,2011).

Com o intuito de explorar, ao menos em partes, as técnicas de fabricação digital para o ensino de POO que este trabalho irá utilizar os métodos de modelagem 3D, e busca a inspiração na fabricação digital para motivar os alunos a produzirem modelos que representem conceitos de POO.

2.3. Semiótica Organizacional

O uso da semiótica como ferramenta tem sido pesquisada em diversas áreas do conhecimento, é muito utilizada nas Ciências Humanas, como Linguística, Educação, Antropologia, se tornando uma filosofia cientifica. A semiótica também tem relevância em Computação, em áreas como Interação Humano-Computador (IHC) (Bonacin, 2004) e Semiótica Computacional.

Segundo Gudwin (2002), a Semiótica é compreendida como uma ciência que aborda o estudo de signos, o seu uso na cognição, na comunicação e na representação. A Semiótica tem suas bases no trabalho de Saussure (2004), conhecida como semiótica "diádica", e também aos trabalhos de Morris (1938) e Pierce (1931 – 1958), conhecida como semiótica "triádica", que é adotada na semiótica organizacional.

A SO estuda a comunicação e a interpretação através das pessoas, em grupos ou mesmo em formato individual. A SO pode ser entendida como um dos ramos da Semiótica e envolve uma organização como um sistema de signos (Bonacin, 2004). A SO utiliza alguns conceitos e técnicas baseadas na Semiótica Pierciana (Pierce, 1931-1958).

Na teoria Peirceana (Peirce 1931-1958) "*um signo, ou representâmen, é aquilo que, sob certo aspecto ou modo, representa algo para alguém. Dirige-se a alguém, isto é, cria, na mente dessa pessoa, um signo equivalente, ou talvez um signo mais desenvolvido. Ao signo assim criado denomino interpretante do primeiro signo. O signo representa alguma coisa, seu objeto. Representa esse objeto não em todos os seus aspectos, mas com referência a um tipo de ideia que eu, por vezes denominei fundamento do representâmen (...)"*.

Com o *representâmen*, o interpretante e o objeto, a tríade de Pierce pode ser graficamente representada por meio do triângulo, como apresentado na Figura 5, que exemplifica o signo como uma relação triádica. Se, na interpretação de alguém (interpretante), o desenho de uma árvore, em uma tela, quadro ou folha de papel representa o objeto árvore (planta), essa relação (a tríade) entre o *representâmen*, o interpretante e o objeto é um exemplo de signo.

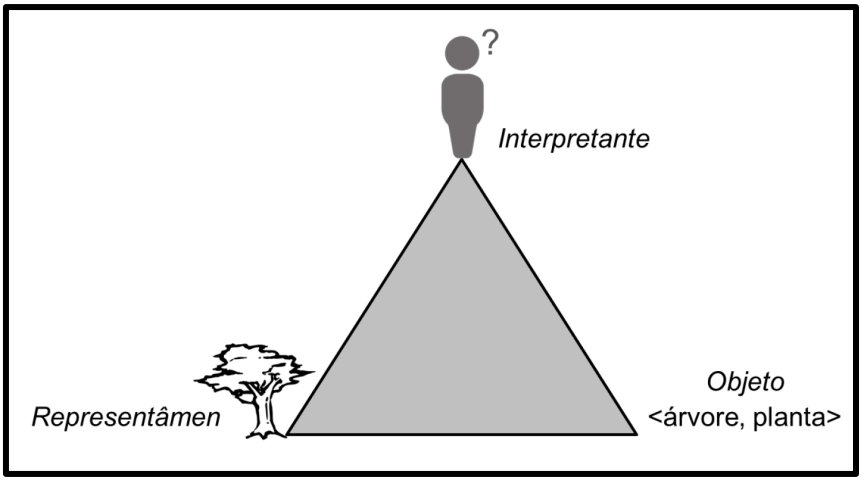


Figura 5 - Exemplo do signo como uma relação triádica

A partir do exposto, a Semiose pode ser compreendida como a produção de sentido, como elemento que compõe a tríade de Pierce. Pierce define que Semiose é "a ação do signo", ou o processo pelo qual o signo tem efeito cognitivo. *"[...]por semiose entendo uma ação, uma influência que seja ou coenvolva uma cooperação de três sujeitos, como, por exemplo um signo, o seu objeto e o interpretante, tal influência tri-relativa não sendo jamais passível de resolução em uma ação entre duplas"* (Peirce, 1931-1958).

Conforme destacado por Eco (1976), na semiótica de Peirce, para estabelecer o significado de um significante é necessário nomear o primeiro significante por meio de outro significante, que a por sua vez conta com outro significante que pode ser interpretado por outro, e assim sucessivamente, constituindo um processo de Semiose Ilimitada. Segundo Eco (1976), esta é a única garantia de um sistema semiótico capaz de explicar-se a si próprio, em seus próprios termos.

A Semiótica Organizacional compreende um campo de pesquisa envolvida com o estudo de organizações, utilizando conceitos e métodos da Semiótica para modelar organizações do ponto de vista do processamento de signos (Liu, 2000; Liu e Li, 2014). Para SO uma "organização" é qualquer sistema ou conjunto organizado ou que demande um comportamento organizado, no qual o processo de geração e interpretação de signos é um componente básico por trás do comportamento dessa organização. Vários estudos de Stamper (*e.g.*, Stamper, 1973) e Liu (*e.g.*, Liu, 2000) abordam principalmente os aspectos da SO aplicados a organizações humanas e a influência decisiva que exercem

sobre o comportamento organizacional, através do intercâmbio de signos de maneira individual ou em grupos.

A classificação e organização serão repetidas para cada nível da escada semiótica podendo haver ou ser criadas questões que ajudem a identificar os elementos. O framework semiótico serve como guia para a análise dos problemas e soluções para o ensino de POO. A intenção é classificar e organizar os problemas e relacionar os domínios dos objetivos didáticos do nível físico ao social de acordo com a escada semiótica (Liu, 2000). Na Figura 6 é possível visualizar como o framework semiótico consegue organizar em níveis aspectos da plataforma tecnológica e do sistema de informação humano.

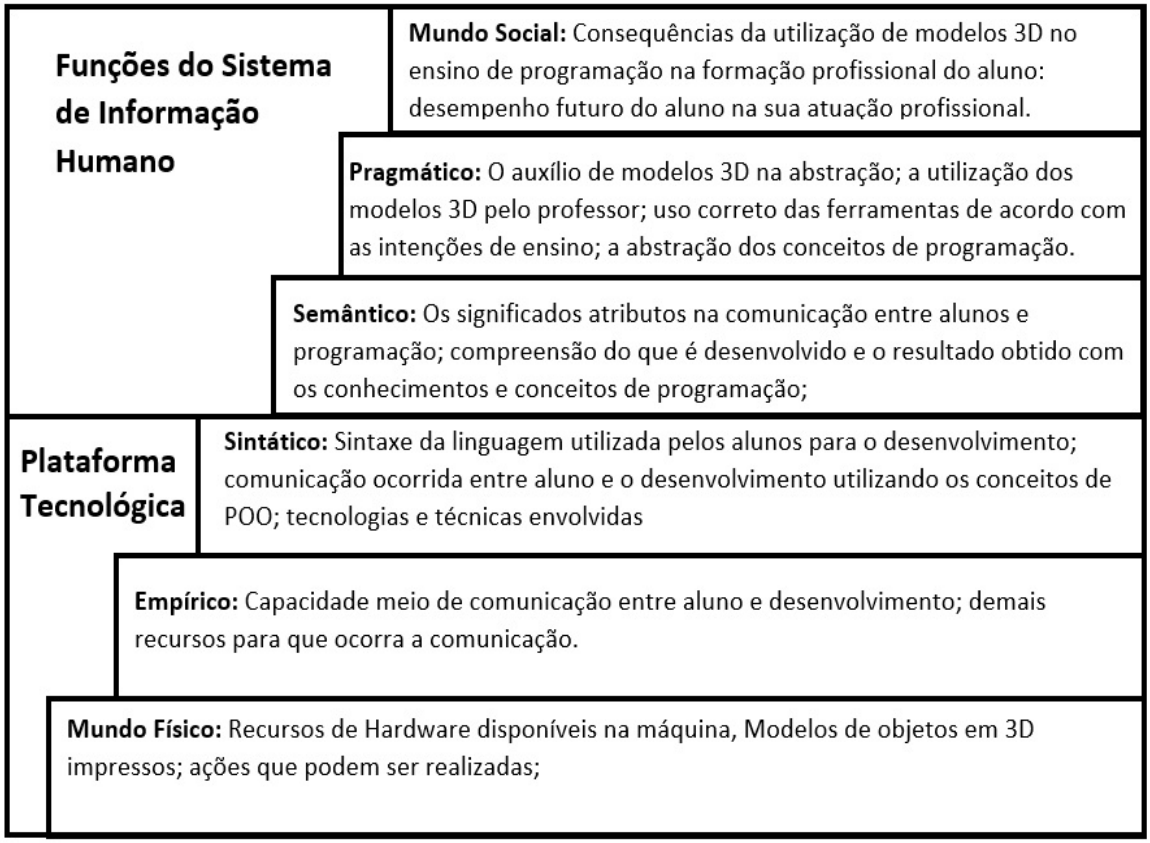


Figura 6 - Framework semiótico

2.3.1 Ferramenta MEASUR

Segundo Stamper (1993), o MEASUR (*Methods for Eliciting, Analysing and Specifying Users Requirements*) é um conjunto de métodos para tratar um problema desde seus estágios iniciais de entendimento, quando suas fronteiras ainda não são claras. Na visão do MEASUR, a solução deve considerar os aspectos sociais, pragmáticos e semânticos que vão além da infraestrutura necessária (aspectos físicos, empíricos e sintáticos).

Em 1993, Stamper definiu o significado de MEASUR como:

- *Methods, Means, Models* for, (Métodos, significados, modelos para);
- *Exploring, Eliciting, Evaluating,* (Exploração, divulgação, avaliação,);
- *Articulating, Analysing, Assenssing, and,* (Articulação, análise, avaliação e);
- *Struturing, Specifying, Simulating*, (Estruturação, especificação, estimulação);
- *User's*, (Do Usuario);
- *Requeriments,* (Requisitos);

Os métodos MEASUR visa apoiar o usuário a transformar um problema, que é compreendido em pedaços desestruturados em novas estruturas enxutas que derivam em soluções técnicas. A ferramenta MEASUR é composta por cinco grandes métodos (Stamper, 1993; Liu, 2000):

- **Métodos para Articulação do Problema** (*Problem Articulation Methods* - PAM): construídos para ser utilizados nas fases iniciais de modelagem, para ajudar na identificação de elementos de maiores importâncias para o modelo. O PAM abrange quatro métodos: Definição da Unidade do Sistema (*Unit System Definition*), Quadro de Valoração (*Valuation Framing*), Análise Colateral (*Collateral Analysis*) e Morfologia do Sistema (*System Morphology*);
- **Método da Análise Semântica** (*Semantic Analysis Method* - SAM): tem como objetivo principal representar os requisitos do problema em um modelo formal. Normalmente, utilizam-se modelos de ontologias que descrevem uma visão de um domínio em foco;
- **Método da Análise de Normas** (*Norm Analysis Method* - NAM): Possibilitam a especificação de padrões gerais de comportamento dos

agentes que participam do sistema organizacional nos níveis pragmático e social. Essa especificação é realizada por meio da análise dos comportamentos regulares e da sua descrição na forma de normas comportamentais;

- **Análise do Controle e Comunicação** (*Communication and Control Analysis*): analisa os diferentes tipos de comunicação entre os agentes participantes da organização, por meio da categorização das mensagens trocadas entre eles;
- **Análise Meta-Sistêmica** (*Meta-Systems Analysis*): observa a organização de um ponto de vista externo, permitindo uma percepção de suas interações com outras organizações, em um nível hierárquico superior.

2.3.2 Ferramentas da SO aplicadas ao aprendizado

Trabalhos anteriores como Zanetti e Bonacin (2014), no contexto de Robótica Pedagógica, e Silva Filho e Bonacin (2016) no contexto de WebQuests, entre outros; propuseram o uso dos métodos de SO na concepção e avaliação de práticas com estudantes. Ao compreender o processo de ensino e aprendizagem como uma organização, a SO pode auxiliar na compreensão e validação de seus métodos. O método PAM pode ser empregado na formulação das práticas e para atender os critérios de avaliação no ensino, ao delinear e especificar o problema em foco. Com o método SAM é possível investigar aspectos semânticos presentes no ambiente de ensino por meio da definição de *affordances*, dependências ontológicas e outros tipos de relacionamentos. Já o método NAM possibilita estudar padrões comportamentais na aprendizagem, assim fornecendo resultados que descrevem efetividade do processo e respectivamente suas falhas.

Métodos PAM

Nessa proposta, o PAM contém um conjunto de métodos a ser aplicados na elaboração de práticas pedagógicas, onde primeiramente será realizado um levantamento dos conceitos a serem abordados e a definição do escopo a ser trabalhado. Este conjunto de métodos são aplicados onde a definição de problema é muito vaga e complexa.

No contexto deste estudo, o aluno é tratado como um *stackholder* (Parte interessada do domínio), o PAM inclui está análise ou validação a partir do *stackholder*.

Método SAM

A utilização do método SAM nesta pesquisa tem como intenção identificar os agentes e os *affordances* do domínio, também como as dependências dentro do contexto das práticas com a utilização de modelos 3D no ensino e aprendizagem POO. Para realizar a análise de domínio é necessário elicitar requisitos para que haja a formação das práticas de forma sistemática e verificar assim os aspectos relacionados a semântica da linguagem utilizada para a comunicação entre os princípios envolvidos e seus padrões comportamentais.

Com foco nos seus agentes e com a intenção de descrever uma organização, o SAM irá modelar e avaliar os *affordances* (*e.g,* modelos 3D) e as dependências, bem como os agentes do ambiente de ensino tais como alunos e professores. O diagrama de ontologia representa a visão dos agentes responsáveis no domínio, comportamentos e *affordances.* O relacionamento entre os signos e decorrência das ações apropriadas é visto através do significado de palavras no ambiente de um modelo semântico.

O modelo semântico e conceitos do SAM adotados nesta pesquisa são baseados no trabalho de Liu (2000) e apresentados a seguir (Figura 7):

- **Mundo:** Socialmente construído através das ações dos agentes, tendo como base o que é oferecido pelo mundo físico para ele próprio;
- ***Affordance***: O conceito foi introduzido por Gibson (1968), que pode ser utilizado para expressar invariantes de repertórios de comportamento de um organismo possíveis pela estrutura do organismo combinada com a de seu ambiente. No SAM este conceito diferencia-se e estende o conceito de Gibson ao considerar invariantes sociais.
- **Agente**: é um tipo especial de *affordance*, que pode ser definido como alguma coisa que tem um comportamento responsável. Um agente pode ser um indivíduo, um grupo cultural, uma comunidade, sociedade, etc. (uma pessoa, um departamento, uma organização, etc. podem ser agentes);
- **Determinante**: é uma invariante de quantidade e qualidade de agentes e *affordances*, pelos quais podemos diferenciar uma instância da outra;

- **Ontologia**: vocabulário com significados comuns para os signos, sendo assim, os agentes devem criar significados comuns para os signos, por exemplo para a comunicação humana é necessário um vocabulário comum;
- **Dependência ontológica**: dependência que ocorre quando um *affordance* só é possível com a existência de outros *affordances*.
- **Especialização**: estruturas genérico-específicas entre agentes e *affordances*, criadas de acordo com as propriedades que eles compartilham ou não;
- **Parte-todo**: quando um agente ou *affordance* pode ser parte de outro agente ou *affordance*;
- **Papel *(Role-Name)***: papel específico que um agente pode desempenhar em função de um *affordance* que ele possui;
- **Responsabilidade**: autoridade dada a um agente responsável pelas ações e trocas de estados em relação aos *affordances*.

O SAM é realizado em quatro fases principais:

- **Definição do problema**. É feita a descrição do problema e ele é estudado de maneira complementar à definição do problema. Podem ser realizadas análises de documentos relevantes e entrevistas com usuários;
- **Geração de *affordances*** candidatos. Os designers analisam o resultado da primeira fase para identificar *affordances*, possíveis agentes e outros tipos de relacionamentos. Uma lista com todas as palavras consideradas relevantes, algumas palavras são deixadas de fora, mas podem ser incluídas no modelo posteriormente;
- **Agrupamento de Candidatos**. Os designers agrupam os candidatos da lista construída durante a última fase em categorias de acordo com os conceitos básicos da análise semântica (como agentes, *affordances*, determinantes, etc). O próximo passo é construir uma tabela com duas colunas: (1) as palavras candidatas a *affordances* e (2) as classificações e alguma informação adicional explicando suas funções. Alguns candidatos novos podem ser incluídos na lista da fase 2. Em seguida, pequenos diagramas ontológicos são construídos como fragmentos de um diagrama maior; estes fragmentos têm o propósito de exibir os efeitos dos agrupamentos nas unidades semânticas;

- **Diagramação Ontológica**. Os fragmentos então são conectados para formar um único diagrama de ontologia, de acordo com as dependências existentes entre eles. Alguns erros cometidos durante a última fase são corrigidos nesta fase. O diagrama de ontologia deve ter um agente raiz único, a "sociedade". Também é recomendado que cada agente deve ter no máximo dois antecedentes nas relações de dependências ontológicas (Stamper, 1973).

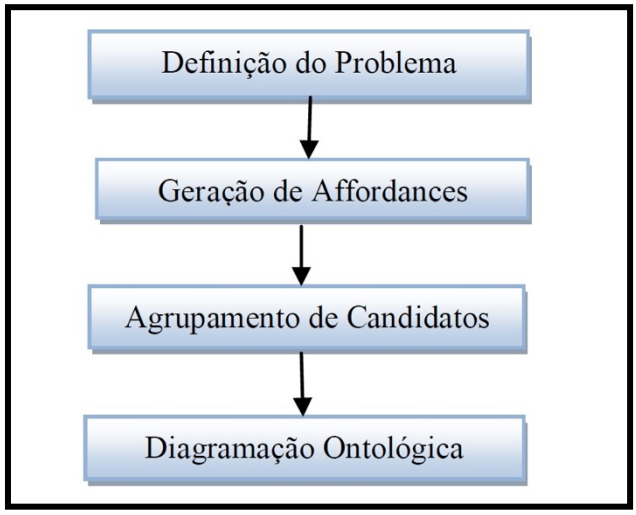


Figura 7 -As principais fases do SAM (Bonacin 2004, adaptado de Liu 2000)

Método NAM

Com a finalidade de modelar aspectos estáticos do comportamento dos agentes, a análise semântica é representada por níveis pragmáticos que descrevem o relacionamento entre a utilização intencional dos signos para a comunicação entre agentes e o comportamento resultante dos agentes no contexto social. Segundo Liu (2000), no nível social, as normas expressam crenças, expectativas, contratos, compromissos, culturas, leis e negócio. Normas correspondem no nível social a ideia de *affordance* no nível individual. As normas também podem ser definidas nos níveis técnico, formal e informal da organização.

A análise de Normas (AN) é representado o comportamento dos agentes nos níveis pragmáticos e social. Em referência à avaliação de resultados, nesta pesquisa será abordada a orientação as normas de aspecto pragmático, social e comportamental. Desta

forma modelando um sistema de avaliação. No processo deste trabalho, cada tipo de norma representa um aspecto específico, conforme listadas abaixo:

- **Normas denotativas:** Serão importantes para a verificação da compreensão dos conceitos e escolha correta do elemento a ser utilizado (funções ou elementos da programação orientada a objetos) e a sua aplicação semântica (a utilização correta da programação orientada a objetos);
- **Normas avaliativas:** Respostas e as expectativas exibidas por meio da execução dos scripts no seu desenvolvimento diante do contexto. Estas normas estão relacionadas à escolha das funções corretas em determinada situação para tentar atingir o objetivo do programa;
- **Normas Perceptuais:** Ajuda na compressão de como os estudantes identificam diferentes métodos e funções executadas pelo software. O software irá proporcionar resultados a cada ação ou método executado pelo usuário.
- **Normas Comportamentais:** Para atingir um objetivo, a programação exige uma sequência de ações ou métodos a seguir. Nesta pesquisa as normas comportamentais definem os comportamentos dos alunos diante das várias situações encontradas no momento de resolver algum problema ou solução para um software.
- **Normas Cognitivas:** permitem com que o aluno seja avaliado em relação aos aspectos de programação e resultados obtidos com as funções e métodos desenvolvidos. A programação exige um alto nível de cognição, envolvendo muita atenção, raciocínio, percepção e linguagem. Com a utilização de modelos em 3D para a representação de conceitos abstratos da programação, as soluções obtidas podem ser mais representativas, auxiliando na compreensão dos conceitos.

A utilização destas normas possibilita planejar atividades com o objetivo de o aluno aprimorar seu comportamento e expertise em relação a programação orientada a objetos.

As normas comportamentais, por exemplo, podem ser expressas de forma explícita e utilizadas na elaboração de atividades com alunos. Segundo Bonacin (2004),

as normas comportamentais descrevem como um agente pode julgar e se comportar perante uma situação, podendo assim expor as decisões foram tomadas para ações, incluindo também alguns aspectos ligados a autoridade de responsabilidades, que podem ser descritas através da lógica deôntica, conforme definido em (Liu 2000, Liu e Li, 2014):

<Norma>::= **sempre** <condição> **se** <estado> **então** <agente> **é** <D> **fazer** <ação>

- < Norma >: Corresponde ao nome da norma;
- < Condição >: especifica uma condição em que a norma é aplicada em determinado estado < estado >;
- < Agente >: Especifica qual agente tem a obrigação, permissão ou proibição, de acordo com operador deôntico < D >;
- < Ação >: Realização de uma determinada ação.

A partir do referencial apresentado podemos ter uma visão macro sobre as áreas apresentadas. Relacionadas podemos separa-las em três bases metodológicas e teóricas, sendo elas: Semiótica Organizacional; Ensino de POO e praticas pedagógicas utilizando ADDIE. Estas três diferentes bases provem uma estrutura sólida para o desenvolvimento do framework com a intenção de aplicar praticas pedagógicas baseados no modelo ADDIE utilizando métodos estruturados e de avaliação baseados na Semiótica. Outra aplicação possível é a junção do ensino de POO utilizando os métodos de avaliação da SO.

3. TRABALHOS RELACIONADOS

Com o avanço rápido da computação no início dos anos 60 a programação de computadores tornou-se parte de um novo conjunto de habilidades e atividades intelectuais desenvolvidas ao longo do século (Papert, 1991). Segundo Papert (1991), a tecnologia pode ser uma ferramenta emancipatória capaz de tonar crianças e jovens "criadores", em um ambiente que os alunos podem concretizar suas ideias e projetos. Com o aumento de acesso à computação, ferramentas de programação como Scratch (Resnick *et al.* 2009) e NetLogo (Wilensky 1999) conseguiram grande popularidade levando conceitos de programação para milhões de estudantes e professores.

Segundo Anderson (2012), estamos em outro momento de mudança de paradigma, uma vez que as novas tecnologias de fabricação digital propiciam processos alternativos de produção que terão grande impacto no século XXI. A tecnologia de fabricação digital pode expandir a visão do aprendiz como "criadores", sendo capaz de ir do digital/abstrato ao físico/concreto, assim trabalhando de forma integrada conceitos de programação, engenharia, design, entre outros.

Essa tecnologia pode ainda ser utilizada para o desenvolvimento do raciocínio formal ao aproximar o pensamento formal do pensamento computacional. Borges, Menezes and Fagundes (2016) apresentam evidências do desenvolvimento do pensamento computacional em atividades de "Maker", tais como a abstração, a decomposição, a generalização, o pensamento algoritmo e a avaliação. Atualmente, existem "vários" laboratórios de fabricação e *making* em ambientes educacionais para o desenvolvimento de múltiplas habilidades, fundamentados em pilares pedagógicos tais como: educação experiencial, criação e pedagogia crítica. O foco desta dissertação é explorar uso de modelos 3D inspirados pela possiblidade de fabricação digital em atividades ligadas ao ensino de POO.

Primeiramente, foram definidas as questões que norteiam o desenvolvimento desta revisão, sendo a seguinte questão principal (QP): *Quais são os estudos sobre o uso de modelagem 3D e fabricação digital no ensino de programação?* A partir da QP definimos três questões específicas:

- (Q1) Quais são as práticas pedagógicas abordadas no ensino de programação usando estas tecnologias?

- (Q2) Quais tecnologias e dispositivos são utilizados para este propósito?
- (Q3) Quais níveis de programação e que tipos de conceitos são ensinados com essas tecnologias?

Com base nestas questões de pesquisa, selecionamos as seguintes bases: Google Acadêmico[1]; *ScienceDirect*[2]; *Scopus*[3]*; IEEE Xplore*[4]; *Springer Link*[5]; *ACM Digital Library*[6] e *Pro Quest*[7]. Além de buscas nas bases do Simpósio Brasileiro de Informática na Educação (SBIE), do Workshop de Informática na Escola (WIE) e da Revista Brasileira de Informática na Educação (RBIE). Foram considerados artigos de 2005 até maio de 2017, utilizando as *strings* de busca destacadas na Tabela 2, que contém termos relacionados à fabricação digital e o ensino de programação em português e inglês.

Tabela 2 - Strings de busca utilizadas

***Strings* de busca**
(“Digital Fabrication” OR “3D Modeling” OR “Digital Models” OR “3D Printing” OR “Computer-aided Design” OR “3D Printer”) AND (“Programming learning” OR “Programming teaching” OR “teaching object-oriented”)
(“Fabricação digital” OR “Modelos 3D” OR “Modelos Digitais” OR “Desenho Assistido por Computador” OR “Impressão 3D”) AND (“Aprendizagem de Programação” OR “Ensino de Programação” OR “Ensino de Orientação a Objetos”)

A revisão foi realizada em cinco passos: (1) Busca em bases cientificas empregando as *strings* apresentadas na Tabela 3; (2) escolha de artigos fundamentados em seus títulos e resumos que estejam relacionados às questões de pesquisa; (3) leitura dos artigos por completo e aplicação de critérios de inclusão e exclusão apresentados na Tabela 4; (4) avaliação crítica, análise e eliminação dos trabalhos com resultados de pouca relevância; e, (5) escalação de trabalhos que serão considerados neste trabalho. Os artigos selecionados têm alto nível de relação e contribuição mais conclusiva no que diz respeito às perguntas analisadas. Os artigos foram categorizados por contribuições ligadas às perguntas Q1, Q2 e Q3.

[1] https://scholar.google.com.br/
[2] http://www.sciencedirect.com/
[3] http://www.scopus.com/
[4] http://ieeexplore.ieee.org/Xplore/home.jsp
[5] http://link.springer.com/
[6] http://dl.acm.org/
[7] http://search.proquest.com/

Tabela 3 - Critérios de inclusão e exclusão

Critérios de Inclusão	
I1	Artigos científicos completos (cinco páginas ou mais, com conteúdo científico).
I2	Artigos que abordem a utilização de fabricação digital/modelagem 3D para o ensino de programação (questão QP).
I3	Trabalho apresenta resultados ou discussão sobre tecnologias ou práticas pedagógicas que envolva o uso de fabricação digital/modelagem 3D no ensino de programação.
Critérios de Exclusão	
E1	Estudos que não se aprofundem na fabricação digital/modelagem 3D para o ensino de programação.
E2	Trabalhos que não contemplem diretamente nenhum conceito relacionado a ensino de programação.
E3	Trabalhos que não estão disponíveis em meio digital.

A Tabela 4 apresenta a quantidade de resultados em cada um dos passos de seleção distribuídos por base indexada. A revisão, incluindo todos os passos, foi realizada nos meses de novembro de 2016 até maio de 2017.

Tabela 4 - Quantidade de artigos selecionados por base a passo da pesquisa

Nome das Bases	Passo 1	Passo 2	Passo 3	Passo 4	Passo 5
Google acadêmico	232	32	17	15	11
Google acadêmico (português)	24	2	0	0	0
ScienceDirect	2	2	2	0	0
Scopus	30	3	2	1	0
IEEE Xplore	24	13	10	9	7
Springer Link	109	2	0	0	0
ACM Digital Library	21	1	0	0	0
Pro Quest	11	9	5	3	2
SBIE + *WIE* + *RBIE* (português)	2	2	0	0	0
TOTAL	**454**	**66**	**36**	**28**	**20**

Pela escolha por resumos e títulos (passo 2) foram encontrados 66 trabalhos que tem alguma relação com o foco desta pesquisa. Os dados apresentados na Figura 8 demostram a evolução do número de trabalhos ao longo do tempo por tipo de publicação. Isso desmonta uma tendência crescente visto que, por exemplo, apenas durante o período de 2014 a 2016 foram publicados 35 trabalhos na área, ou seja, 53% dos trabalhos encontrados no período de 2005 a 2017.

Conforme exibe Tabela 4, após a execução passos foram excluídos 46 trabalhos. Desses, 27 aplicando o critério E2 (Tabela 3) e 19 pelo critério E1. A Tabela 5 apresenta a relação desses artigos com os critérios de inclusão adotados (Tabela 3). O gráfico 1 apresenta a distribuição dos trabalhos por tipo e ano de publicação (o ano de 2017, não apresentou resultados por a pesquisa ter sido realizada no início do ano).

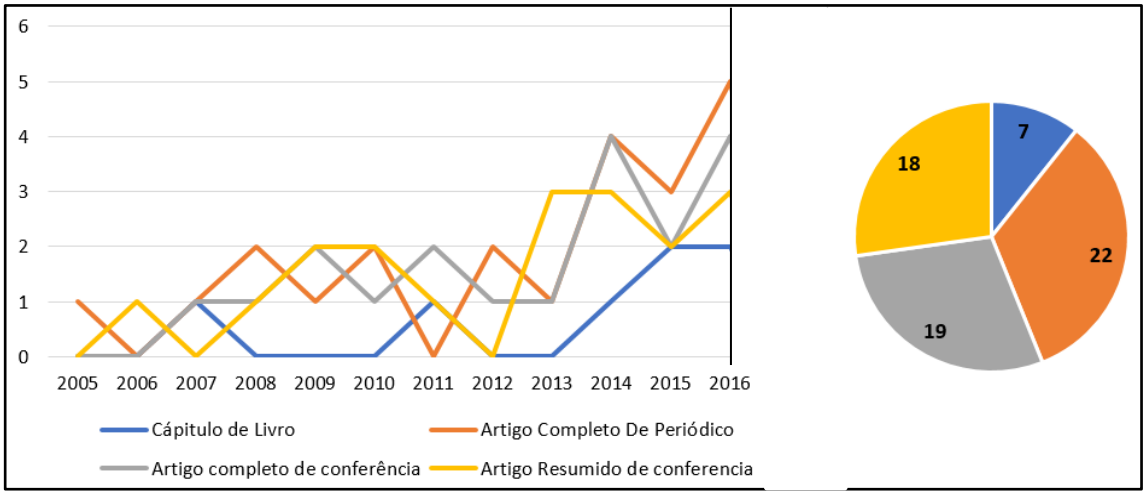


Figura 8 - Trabalhos associados por tipo e ano

Tabela 5 - Trabalhos selecionados por critério de inclusão

	Referências	Qtd
1	[Blikstein 2013; Bell *et al.* 2010; Cooper, Dann and Pausch 2006; Martinez and Stager 2013; Tillman 2012; Smith, Iversen and Hjorth 2015; Iversen *et al.* 2016; Blikstein and Krannich 2013; Alrashidi, Callaghan and Gardner 2014; Bishop-Clark *et al.* 2007; Chandramouli, Zahraee and Winer 2014; Davis 2008; Goulet and Slater 2009; Guenaga *et al.* 2014; Kelleher 2006; Maravilhas and Martins 2016; Martinez and Stager 2013; Pellas 2014; Sharma, Stigall and Rajeev 2015; Stojanova, Stojkovic and Bikov 2012; Sykes 2007]	**20**
2	[Blikstein 2013; Bell *et al.* 2010; Cooper, Dann and Pausch 2006; Martinez and Stager 2013; Tillman 2012; Smith, Iversen and Hjorth 2015; Iversen *et al.* 2016; Blikstein and Krannich 2013; Alrashidi, Callaghan and Gardner 2014; Bishop-Clark *et al.* 2007; Guenaga *et al.* 2014; Maravilhas and Martins 2016; Martinez and Stager 2013; Sharma, Stigall and Rajeev 2015; Stojanova, Stojkovic and Bikov 2012]	**14**
3	[Blikstein 2013; Bell *et al.* 010; Cooper, Dann and Pausch 2006; Martinez and Stager 2013; Tillman 2012; Smith, Iversen and Hjorth 2015; Iversen *et al.* 2016; Blikstein and Krannich 2013; Guenaga *et al.* 2014; Maravilhas and Martins 2016; Martinez and Stager 2013; Sharma, Stigall & Rajeev 2015; Stojanova, Stojkovic and Bikov 2012]	**12**

Durante a análise dos resultados obtidos, os artigos foram divididos em três categorias de acordo com as questões Q1, Q2 e Q3, conforme apresentado nas subseções a seguir.

3.1. Métodos de ensino utilizados

A Tabela 6 apresenta a distribuição dos trabalhos selecionados de acordo com 6 abordagens de ensino principais identificadas a partir da leitura do conteúdo.

Tabela 6 - Trabalhos por método/abordagem de ensino adotada

Método	Referências	Qtd
Expositivo	[Blikstein 2013; Bell *et al.* 2010; Cooper, Dann and Pausch 2006; Martinez and Stager 2013; Tillman *et al.* 2012; Smith, Iversen & Hjoth 2015; Blikstein and Krannich 2013; Sharma, Stigall and Rajeev 2015; Stojanova, Stojkovic and Bikov 2012]	**9**
Narrativo	[Blikstein 2013; Bell *et al.* 010; Cooper, Dann and Pausch 2006; Martinez and Stager 2013; Tillman *et al.* 2012; Blikstein and Krannich 2013; Sharma, Stigall and Rajeev 2015; Stojanova, Stojkovic and Bikov 2012]	**8**
Skimming	[Tillman 2012; Smith, Iversen & Hjoth 2015; Stojanova, Stojkovic and Bikov 2012]	**3**
Programado	[Smith, Iversen & Hjoth 2015; Blikstein and Krannich 2013]	**2**
Estudo de caso	[Blikstein 2013; Tillman *et al.* 2012; Smith, Iversen & Hjoth 2015; Guenaga *et al.* 2014; Maravilhas and Martins 2016; Martinez and Stager 2013]	**6**
Baseado em Problemas	[Blikstein 2013; Cooper, Dann and Pausch 2006; Martinez and Stager 2013; Tillman 2012; Iversen *et al.* 2016; Blikstein and Krannich 2013; Guenaga *et al.* 2014; Maravilhas and Martins 2016; Sharma, Stigall & Rajeev 2015; Stojanova, Stojkovic and Bikov 2012]	**10**

Metade (10) dos trabalhos analisados utilizam métodos de aprendizagem baseados em problema, onde usualmente eram apresentados problemas a serem solucionados utilizando conceitos de programação e tecnologias de fabricação digital. O segundo método mais utilizado é o expositivo (*e.g.,* apresentações de conceitos e soluções) com 9 trabalhos, seguido por narrativas com 8 trabalhos e por estudos de casos com 6 trabalhos. Também foram identificados trabalhos que fazem uso de *skimming* e métodos programados com 3 e 2 trabalhos respectivamente. A análise também revela que a maioria dos trabalhos fazem uso de mais de um método como forma de propiciar uma maneira adequada para explorar fabricação digital no ensino de programação.

3.2. Tecnologias e dispositivos utilizados

A Tabela 7 apresenta a distribuição dos trabalhos selecionados de acordo com as tecnologias e dispositivos utilizados. A tecnologia mais utilizada é software de interação 3D, pois ela foi adotada por com quase metade (9) dos artigos selecionados. O uso de

impressora 3D está presente em 7 artigos, enquanto software para desenho vetorial é utilizado por 4 artigos e cortador a laser, robôs e *BlueJ* em 3 artigos cada. As demais tecnologias foram utilizadas em 1 ou 2 artigos. A análise revela que os trabalhos combinam diferentes tecnologias tridimensionais e de fabricação para construir o ambiente necessário, sendo raros os trabalhos que focam em uma única tecnologia.

Tabela 7 - Trabalhos por tecnologias e dispositivo utilizado

Tecnologia/dispositivo	Referências	Qtd
Software de Desenho Vetorial	[Blikstein 2013; Bell *et al.* 2010; Smith, Iversen and Hjorth 2015; Pellas 2014]	**4**
Cortador a Laser	[Blikstein 2013; Iversen *et al.* 2016; Stojanova, Stojkovic and Bikov 2012]	**3**
Impressora 3D	[Blikstein 2013; Bell *et al.* 2010; Smith, Iversen and Hjorth 2015; Alrashidi, Callaghan and Gardner 2014; Blikstein and Krannich 2013; Sykes 2007; Maravilhas and Martins 2016]	**7**
Robôs	[Blikstein 2013; Blikstein and Krannich 2013; Guenaga *et al.* 2014]	**3**
Software de Interação 3D	[Cooper, Dann and Pausch 2006; Martinez and Stager 2013; Goulet and Slater 2009; Bishop-Clark *et al.* 2007; Sykes 2007; Sharma, Stigall and Rajeev 2015]; Kelleher 2006; Pellas 2014; Maravilhas and Martins 2016]	**9**
Graphtec Silhouette[8]	[Tillman 2012; Alrashidi, Callaghan and Gardner 2014]	**2**
Whiteboard	[Tillman 2012]	**1**
LittleBits[9]	[Smith, Iversen and Hjorth 2015; Iversen *et al.* 2016]	**2**
MakeyMakeys[10]	[Smith, Iversen and Hjorth 2015]	**1**
Spirogator[11]	[Blikstein and Krannich 2013]	**1**
BlueJ[12]	[Davis 2008; Stojanova, Stojkovic and Bikov 2012; Guenaga *et al.* 2014]	**3**
Star logo[13]	[Davis 2008]	**1**
BuzzBoard[14]	[Alrashidi, Callaghan and Gardner 2014]	**1**
Python within Vizard[15]	[Sharma, Stigall and Rajeev 2015; Pellas 2014]	**2**

[8] http://www.graphtecamerica.com/graphtec-america-products-cutting-plotters-silhouette-cameo
[9] https://littlebits.cc/
[10] http://www.makeymakey.com/
[11] https://github.com/nataliefreed/spirogator
[12] https://www.bluej.org/
[13] http://web.mit.edu/mitstep/starlogo-tng/download/index.html
[14] http://www.FortiTo.com
[15] http://docs.worldviz.com/vizard/

3.3. Níveis de programação e conceitos abordados

A Tabela 8 apresenta os trabalhos distribuídos por níveis de programação e conceitos de Programação Orientada a Objetos (POO) abordados. Os níveis estão classificados de 1 a 5, onde 1 é básico e 5 o nível avançado. Todos os trabalhos abordam a maior parte conceitos básicos de programação como sequências, loops, eventos paralelismo, condições, operadores e dados. Entretanto, a maioria dos trabalhos não foca em apresentar conceitos avançados de programação tais como conceitos de POO. Apenas 4 artigos lidam os 6 conceitos elencados e apenas 2 artigos abordam com um grau de profundidade alto. Tais conceitos foram elencados para análise por fazerem parte das atividades cotidianas e serem necessários para muitos profissionais em computação.

Tabela 8 - Trabalhos por nível de programação e conceitos de POO

Referencias	Nível	Classe	Objeto	Associação	Encaps.	Herança	Polimor.
[Blikstein 2013]	**1**						
[Bell *et al.* 2010]	**2**	**X**	**X**				
[Cooper, Dann and Pausch 2006]	**2**	**X**	**X**				
[Martinez and Stager 2013]	**1**	**X**	**X**			**X**	
[Tillman 2012]	**1**	**X**	**X**				
[Smith, Iversen and Hjorth 2015]	**2**	**X**	**X**			**X**	
[Iversen *et al.* 2016]	**4**	**X**	**X**		**X**	**X**	
[Blikstein and Krannich 2013]	**3**	**X**	**X**	**X**	**X**	**X**	**X**
[Alrashidi, Callaghan and Gardner 2014]	**2**	**X**	**X**				
[Bishop-Clark *et al.* 2007]	**3**	**X**	**X**			**X**	
[Chandramouli, Zahraee and Winer 2014]	**5**	**X**	**X**	**X**	**X**	**X**	**X**
[Davis 2008]	**1**						
[Goulet and Slater 2009]	**2**	**X**	**X**				
[Guenaga *et al.* 2014]	**2**	**X**	**X**	**X**			
[Kelleher 2006]	**3**	**X**	**X**				
[Maravilhas and Martins 2016]	**3**	**X**	**X**			**X**	**X**
[Pellas 2014]	**1**						
[Sykes 2007]	**1**						
[Stojanova, Stojkovic and Bikov 2012]	**4**	**X**	**X**	**X**	**X**	**X**	**X**
[Sharma, Stigall and Rajeev 2015]	**5**	**X**	**X**	**X**	**X**	**X**	**X**

Nível: 1 básico - 5 avançado

3.4. Fabricação digital no ambiente de aprendizagem

Foram incluídos trabalhos nesta categoria que descrevem projetos ou experiências que utilizaram a fabricação digital no ambiente de aprendizagem e seus respectivos resultados. Os trabalhos analisados foram selecionados seguindo os critérios apresentados anteriormente e todos têm contribuições para processos de ensino-aprendizagem.

Esta revisão possibilita uma visão ampla da área de trabalhos sobre a utilização de fabricação digital no ensino. Em Blikstein (2013), por exemplo, realiza uma experiência com 4 laboratórios de fabricação digital para o incentivo à criatividade e ensino de conceitos importantes sobre a modelagem 3D. Os autores mencionam que apesar do potencial dos laboratórios de fabricação digital, os educadores devem se lembrar de que, o verdadeiro poder de qualquer tecnologia não está na técnica, mas também nos novos modos de expressão pessoal que permite, as novas formas de interação humana que facilita e as ideias poderosas que torna acessível aos alunos.

Entre os trabalhos analisados, vários fazem uma análise sistemática sobre a fabricação digital. Bell (2010), por exemplo, relata implicações da aplicação da fabricação digital na educação no sistema de educação. O autor defende que a fabricação digital torna possível fabricação de uma peça robótica sob demanda em vez de manter um armazém de peças padronizadas. Estes tipos de peças podem ainda ser fabricadas diversas vezes sem que haja perda de qualidade ou falhas. Por exemplo, é possível fabricar componentes para baterias e fontes de alimentação, bem como outros componentes metálicos e plásticos, como engrenagens e cames. Assim uma bateria pode ser fabricada em tamanhos e formas personalizadas, a gama de opções de design é consideravelmente mais tangível. A aplicação destes conceitos para os alunos tem um ganho imensurável na produção e fabricação de objetos.

Já Cooper (2006) realiza experimento com o software Alice, ele é utilizado para o ensino de solução de problemas por meio de conceitos de programação. O software Alice fornece um ambiente de programação em 3D animado que apoia uma abordagem para o ensino de solução de problemas. Foi utilizado o ‘Alice’ para desenvolver algoritmos para animar objetos que povoam mundos virtuais. Foi observado que os alunos estavam confortáveis usando objetos e invocando métodos sobre aqueles. Os alunos foram capazes de ver o que deu errado em seus programas e facilmente depurar e

corrigi-los. Demonstrando assim que a utilização de objetos 3D no ambiente de ensino é eficaz.

Em Martinez (2013) são demonstradas formas de utilização da fabricação digital para fazer, reparar ou personalizar componentes que utilizamos na engenharia, design e computação. Existe uma revolução tecnológica e criativa quando o aluno usa a tecnologia para fazer, reparar ou personalizar as coisas que precisamos. Esta tarefa traz conceitos de engenharia, design e ciência da computação para as massas. O aluno ativo está no centro do processo de aprendizagem, ampliando a educação. Portanto, os educadores devem levar as oportunidades do movimento criador para cada sala de aula. O autor destaca a associação que ocorre durante o processo de ensino, contribuindo para as atividades práticas, em relação apenas a teoria de forma menos colaborativa.

Já Tillman (2012) focou nas formas pelas quais os alunos participantes realizaram e reagiram quando utilizada instrução semiaberta a baseada em modelagem de conteúdo científico. Para tanto, foi utilizando o '*NASA-themed transmedia'*(Tillman, 2011), que apresenta atividades de fabricação digital. O estudo foi realizado em uma escola primária em Virgínia central, com alunos do 5º ano com conhecimentos em matemática. Esta pesquisa demonstrou que os alunos com pouca idade podem utilizar a fabricação digital como ferramenta para entendimento de conceitos e modelos independentemente do assunto. Este estudo foi realizado em relação da fabricação digital como uma tecnologia instrucional para apoiar a educação de ciências elementares.

O trabalho de Smith, Iversen e Hjorth (2015) demonstrou que a fabricação digital em contextos educacionais formais não é um processo direto. Estes estudos revelaram que a fabricação digital no ensino primário e secundário pode se beneficiar do pensamento de projeto como parte educacional. O pensamento de projeto apoia a capacidade das crianças de trabalhar em um espaço de solução "perversa", em que o fracasso, processos iterativos e reflexões contínuas sobre materiais de fabricação são partes integrantes do processo. A teoria da fabricação digital deve ser adequadamente apropriada para atender crianças e adolescentes em ambientes que motivação e colaboração, direção e visão no processo não podem ser pressupostos, mas devem ser cuidadosamente desenvolvidos.

Em Iversen (2016) foi investigado sobre o uso de fabricação digital na educação pode se beneficiar do pensamento de projeto como meio para promover uma

compreensão dos processos de fabricação digital entre os alunos. Os resultados dos experimentos realizados apontam para a importância do pensamento de design na fabricação digital educacional e para a necessidade de mais investigações.

Finalmente, destacamos o trabalho de Blikstein & Krannich (2013), onde foi apresentado os conceitos relacionados às atividades e utilização da fabricação digital no ensino fundamental e médio. Fabricação digital como movimento social positivo em torno de alunos, segundo os autores será uma oportunidade sem precedentes para educadores para avançar progressivamente neste projeto baseado, centrada no interesse do aluno que está no centro utilizando sua experiência educacional desenvolvida por anos. No entanto, ainda está em fase inicial de implementação das tecnologias em escolas. Educadores e pesquisadores, segundo o autor, devem estar preparados para uma próxima fase de atividades e temas generativos dentro das salas de aulas ou laboratórios, pois já existem pesquisas sólidas de instrumentos, métodos e como os profissionais devem manipular desenvolvimento.

Na Tabela 9 é apresentada uma síntese dos projetos e trabalhos que foram considerados mais significativos que utilizam a fabricação digital em métodos de ensino. Estes trabalhos trouxeram resultados em diversas áreas e níveis de ensino. A tabela apresenta de forma breve os objetivos de cada trabalho, pontos fortes, limitações e referencias.

Tabela 9 - Síntese de Trabalhos relacionados ao ambiente de Aprendizagem utilizando Fabricação Digital

	Objetivos	Pontos Fortes	Limitações	Referência
1.	Realizar experimentações com laboratórios de Fabricação Digital em Instituições. Mostrar a real necessidade da utilização de objetos em 3D.	Foram relatados 4 projetos que utilizam a impressão 3D para ensinar conceitos de outras disciplinas.	Não foram relatados projetos para ensino de desenvolvimento ou Programação Orientada a Objetos.	[Blikstein, 2013]
2.	Realizar uma revisão sistemática sobre a aplicação de Fabricação Digital na educação.	Doze associações educacionais e dez revistas de tecnologia educacional estão relatadas neste trabalho, relatando experiências.	Não foram aplicadas em nenhuma das associações ou revistas quaisquer tipo de ensino de programação com a utilização de fabricação Digital	[Bell, 2010]

3.	Demostrar o software Alice, utilizado para o ensino de programação. Focado na resolução de problemas de forma lógica	O trabalho demostrou que os alunos estavam confortáveis ao utilizar o software Alice, apenas utilizando os objetos para realizar a programação.	O software Alice não tem uma sintaxe automatizada, é necessário a digitação de código, o trabalho também não demostra nenhum conceito de POO.	[Cooper, 2006]
4.	Demonstrar como usar a fabricação digital para fazer, reparar ou personalizar as coisas que precisamos utilizando a engenharia, design e ciência da computação para a realização da mesma.	Em cada ideia ou invenção são aplicados vários conceitos de fabricação digital e modelação 3D.	Não é conceituado nenhum conceito referente a POO, não há experimentações demonstrando se as ideias desenvolvidas terão ganho ou não no momento da aprendizagem da pessoa. Estudo de pequeno porte, com amostra de 43 participantes	[Martinez, 2013]
5.	Estudo da fabricação digital como uma tecnologia instrucional para apoiar a educação das ciências elementares.	O artigo analisou os efeitos das atividades de fabricação digital que foram projetadas para contextualizar a educação das ciências elementares para alunos do ensino fundamental e médio. As atividades que utilizaram a Fabricação obtiveram um ganho significativo.	Este trabalho foi aplicado para ensino Fundamental e Médio, podia ser aplicado a Graduação. Não é ensinado nenhum conceito de POO no trabalho. Trata-se de um estudo de médio porte, com amostra de 94 participantes.	[Tillman, 2012]
6.	O objetivo deste artigo é desenvolver módulos instrucionais baseados em temas de jogos para estudantes de ciência da computação que motivam e envolvem os alunos enquanto contribuem para seus resultados de aprendizagem.	Os resultados do estudo demonstram a eficácia do módulo instrucional e a possibilidade de incluí-lo no currículo existente com alterações mínimas no material do curso existente.	Este artigo discute os parâmetros de projeto e a implementação do módulo e descreve um estudo de caso sobre a adoção do módulo em uma classe existente.	[Sharma, Stigall and Rajeev, 2015]
7.	Realizar uma revisão sistemática sobre os benefícios que a fabricação digital pode promover para uma compreensão mais profunda dos processos aprendizagem entre os alunos independentemente do nível de ensino.	Foram escolhidas duas escolas em torno de 800 alunos. Verificou-se que os alunos não tinham compreensão da complexidade do processo de fabricação digital, impedindo o fluxo de criação e aprendizado.	Este trabalho se aplica a qualquer nível de ensino porem não há nenhuma abordagem sobre a aprendizagem de programação.	[Smith, Iversen e Hjorth , 2015]
8.	Investigar a utilização de fabricação digital na educação para beneficiar as etapas iniciais no	No pré-teste 68% dos alunos nunca utilizaram fabricação digital para entender conceitos	Estudo de pequeno porte, com amostra de 38 participantes, com poucos professores do ensino	[Iversen, Smith, Blikstein, Katterfeldt &

	desenvolvimento do projeto, para promover uma compreensão dos processos de fabricação digital melhor entre os alunos.	novos. Após a utilização da fabricação digital para ensino de conceitos novos, 97,33% acharam útil a utilização da mesma.	fundamenta, médio e graduação.	Read, 2016]
9.	Apresentar os conceitos relacionados às atividades e utilização da Fabricação Digital no ensino fundamental e médio.	Apresenta as vantagens e desvantagens da utilização da fabricação digital e modelos 3D no ensino de novos conceitos.	Este estudo não apresenta resultados experimentais.	Blikstein & Krannich, 2013

3.5. Discussão sobre Trabalhos Relacionados

De uma forma geral, as análises dos trabalhos demonstram que a fabricação digital tem sido empregada recentemente no ensino de conceitos e disciplinas relacionadas a programação. Porém, os estudos também demonstram que são necessárias pesquisas adicionais em relação aos métodos de ensino, eficácia e utilidade nos diversos contextos e níveis de aprendizagem de programação.

Em relação à questão Q1, os trabalhos apresentam vários estudos com a aplicação de diferentes métodos e resultados como apresentado na Tabela 6, de forma geral, são promissores; entretanto, estes mesmos estudos não chegam a um arcabouço conceitual e/ou entendimento comum de como explorar essa nova tecnologia no ensino de programação. Em acordo com Blikstein (2013), o verdadeiro "poder" da tecnologia não está nela própria, mas nas novas formas de expressão e interação humana que facilitam a aprendizagem. Para tanto, é necessário o desenvolvimento de novas aplicações e métodos que explorem fabricação digital em projetos educacionais. Além, disso é importante destacar a necessidade de pesquisar novos meios para apoiar os docentes na elaboração de atividades de modo produtivo e com qualidade.

Em relação à questão Q2, os trabalhos apresentam uma grande variedade de tecnologias utilizadas. É importante salientar que, embora todos os trabalhos selecionados abordassem conceitos relacionados à fabricação digital, apenas 7 fizeram uso de impressoras 3D em estudos de casos. Alguns trabalhos utilizam essa tecnologia em conjunto com robôs [Blikstein 2013; Blikstein and Krannich 2013], software de desenho vetorial [Blikstein 2013; Bell *et al.* 2010; Smith, Iversen and Hjorth 2015], cortador a laser [Blikstein 2013], software de interação 3D [Maravilhas and Martins 2016], Graphtec Silhouette [Alrashidi, Callaghan and Gardner 2014], LittleBits [Smith,

Iversen and Hjorth 2015], MakeyMakeys [Smith, Iversen and Hjorth 2015], Spirogator [Blikstein and Krannich 2013] e BuzzBoard [Alrashidi, Callaghan and Gardner 2014]. Isso pode sugerir a necessidade de integrar a tecnologia de impressão 3D com outras tecnologias avançadas para prover um ambiente propício à aprendizagem de programação.

Em relação à questão Q3, é possível visualizar que a maioria dos trabalhos foca em conceitos básicos de programação. Tradicionalmente, a robótica educacional, programação visual e ambientes virtuais 3D têm sido estudados como alternativas ao desenvolvimento de raciocínio de crianças e jovens. A fabricação digital e tecnologias associadas também surgem como novas alternativas para o desenvolvimento do raciocínio formal e computacional de crianças e jovens [Borges, Menezes and Fagundes 2016]. Assim, vários trabalhos encontrados [Smith and Hjoth 2015; Blikstein 2013; Blikstein and Krannich 2013; Bell *et al.* 2010; Iversen *et al.* 2016; Maravilhas and Martins 2016; Pellas 2014; Sharma, Stigall and Rajeev 2015; Tillman 2012] ressaltam o uso de fabricação digital no ensino fundamental e médio, revelando os benefícios de utiliza-la em prol do ensino de tópicos relacionados ao STEM *(Science, Technology, Engineering and Mathematics*). Entretanto, o uso da fabricação digital na aprendizagem de conceitos avançados de computação ainda é pouco explorado, embora alguns estudos relatem seu potencial, tais como: Chandramouli, Zahraee and Winer (2014) e Sharma, Stigall and Rajeev (2015), que obtiveram o nível mais alto de complexidade (Tabela 8) em função da profundidade dos conceitos abordados. Estes autores ressaltam que abordagem 3D permite aos alunos superar inibições inerentes que possam ter, por exemplo, devido às dificuldades com a lógica e complexidade da programação. Novas pesquisas podem explorar exemplos práticos de conceitos avançados de POO.

O ensino de programação é um grande desafio para alunos e professores em cursos técnicos e de ensino superior. Novas tecnologias, tais como fabricação digital, trazem oportunidades e desafios para o desenvolvimento de métodos e atividades que promovam o ensino de programação. Neste trabalho, foi apresentado um estudo sistemático da literatura sobre como a fabricação digital e modelagem 3D vem sendo exploradas no ensino de programação de computadores. Vinte (20) trabalhos foram selecionados de acordo com sua relevância ao tema, e apresentados e categorizados neste trabalho. Com a análise desses trabalhos foram identificados os principais métodos de ensino empregados nos estudos relatados, as tecnologias e os dispositivos utilizados,

bem como o nível de programação em foco. Com isto, foi possível destacar e discutir o potencial da fabricação digital, as oportunidades relacionadas ao seu uso no ensino de programação e desafios a serem abordados. No próximo capítulo é apresentada a elaboração de um método que apoie os docentes no desenvolvimento de atividades com alunos, de modo a englobar múltiplas tecnologias ligadas à modelagem 3D (e fabricação digital), com foco principal em conceitos avançados de POO.

4. METODOLOGIA DE ENSINO DE PROGRAMAÇÃO ORIENTADA A OBJETOS COM MODELAGEM DIGITAL – MEPOOMD

Este capítulo apresenta a estrutura e construção de um modelo conceitual (*framework*) unindo os conceitos e ferramentas apresentadas nos capítulos anteriores. O modelo que será apresentado a seguir serviu como base para avaliação e mensuração do nível de aprendizagem dos alunos em relação aos conceitos de POO; utilizando modelagem 3D para a aplicação, fundamentado pedagogicamente no modelo de Design Instrucional ADDIE e em métodos da Semiótica Organizacional. Nas subseções a seguir será apresentada a elaboração do MEPOOMD (Seção 4.1) e o detalhamento das suas fases e atividades (Seção 4.2).

4.1 Elaboração do MEPOOMD

O método foi elaborado a partir de conceitos e áreas que foram abordadas nesta pesquisa. Com estas áreas se torna possível a visualização e entendimento do que realmente foi elaborado para que o método tenha aplicabilidade e eficácia. A Figura 9 apresenta o relacionamento existente entre as áreas abordadas durante a utilização e elaboração do método.

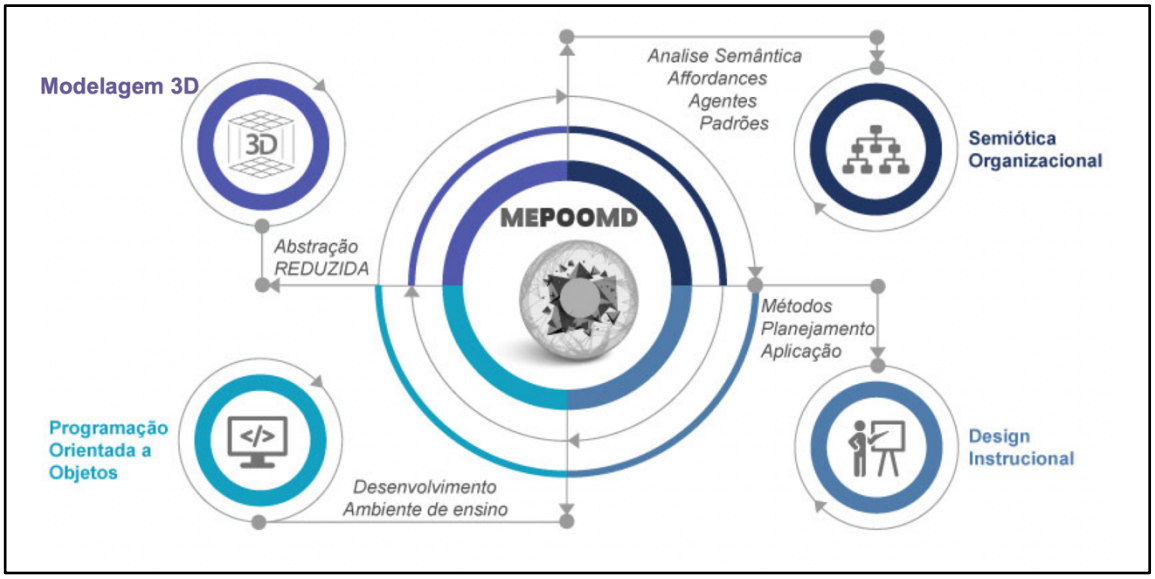


Figura 9 - Relação entre as áreas abordadas na pesquisa

Conforme a Figura 9, a POO por sua vez é o objeto de estudo. Dessa área, também podem ser utilizados ambientes de desenvolvimento de software, que auxiliam no processo de ensino (Figura 9 – canto inferior esquerdo).

A Modelagem 3D e a Fabricação Digital (Figura 9 – canto superior esquerdo) permite trabalhar com múltiplos níveis de abstração, de maneira a possibilitar a redução do nível de abstração (ex: por objetos físicos) quando necessário. Espera-se também, com o uso da Modelagem 3D e Fabricação Digital, obter um maior nível de motivação dos alunos envolvidos no processo de aprendizagem.

Já com a SO é possível identificar aspectos comportamentais, pragmáticos, e semânticos envolvidos em um senário de ensino e aprendizagem, por meio da análise semântica, *affordances*, agentes e padrões (Figura 9 – canto superior direito).

A utilização do Design Instrucional dentro do método permite o melhor planejamento e aplicação das atividades. Ele define cinco fases, são elas: Análise, Design, Desenvolvimento, Implementação e Avaliação (Filatro, 2008). Este método pedagógico é de simples entendimento, e necessita de meios ou padrões para tornar possível a identificação de perfis de alunos, criação de ambientes e padrões de ensino e, por último, uma avaliação concisa e eficiente.

A Figura 10 apresenta a metodologia de pesquisa utilizada para o desenvolvimento do MEPOOMD. Essa metodologia inclui as seguintes etapas:

- **Revisão da literatura:** Revisão detalhada sobre trabalhos relacionados ao método para comprovação da necessidade e importância do mesmo, bem como fundamentação em estudos passados;
- **Proposta do método MEPOOMD:** Nesta etapa foi primeiramente sintetizada a estrutura geral do método, em sequência conforme a análise dos resultados obtidos, o método foi detalhado e refinado;
- **Aplicação do método MEPOOMD no ensino de POO:** Nesta etapa o método foi aplicado com alunos de cursos de POO, conforme detalha capítulo 5.
- **Análise dos resultados obtidos:** Essa fase teve como objetivo analisar os resultados obtidos visando avaliar o método e obter possíveis necessidades de melhorias.

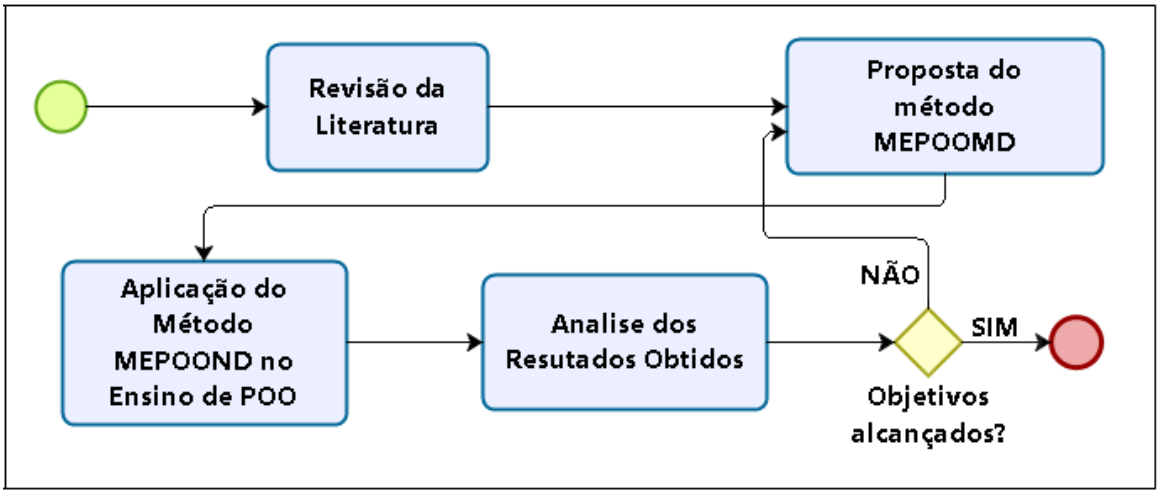


Figura 10 - Metodologia de Pesquisa

Todo o estudo da bibliografia incluindo trabalhos relacionados conforme apresentado no Capitulo 3, indicam a necessidade de investigar e mensurar a aplicação da modelagem 3D e fabricação digital no ensino, em particular no ensino de POO. Destaca-se que não há nenhum trabalho existente com este foco específico. Vários artigos relataram a necessidade de investigar novas formas de estimular o processo de compreensão e criação nos alunos, tais como os aspectos motivacionais.

Mesmo com os números cada vez maiores de publicações científicas e artigos, as avaliações relatadas não são concretas e definitivas em relação ao que o aluno deveria aprender e abstrair utilizando estas novas tecnologias. Existe a carência por métodos que sejam sistemáticos e que consigam planejar e verificar se os comportamentos dos alunos dentre as atividades foram realizados de forma que foram planejadas inicialmente. Por estes motivos, o detalhamento da avaliação aplicada ao método será de suma importância para a comprovação de um método eficiente, prático e de qualidade técnica.

Chamamos de “ciclo de experimentação” a aplicação do método em um primeiro ciclo do seu desenvolvimento, no qual foi realizada uma proposta, o método foi experimentado e os resultados analisados. Com isso foi possível um melhor entendimento em um ambiente real, bem como aprimorar a compreensão do pesquisador de modo a melhor detalhar o método.

Dois grupos de alunos participaram do “ciclo de experimentação”. O primeiro grupo contava com vinte e dois alunos do Curso Técnico em Informática com ensino Médio do UNASP – Hortolândia, este grupo com idades entre quatorze a dezesseis anos.

Pertencentes ao segundo ano do ensino médio e desde o primeiro ano participam de aulas de lógica de programação e POO. O segundo grupo deste ciclo de trabalho, foram vinte e cinco alunos do curso de graduação de sistemas de informação do UNASP-Hortolândia, este grupo com idades entre dezoito e vinte e oito anos. Pertencentes ao 3° Semestre da graduação e desde o segundo semestre já estavam em contato com a linguagem de programação e POO.

A execução deste ciclo inicial foi importante para a formação e detalhamento dos passos para o MEPOOMD, foi possível visualizar que poderia ser aprimorado dentro do método. A aplicação na prática se faz importante pois os perfis de alunos que estão sendo analisados são um tanto quanto diferentes, pensando nos quesitos experiência e idade.

Durante o "ciclo de experimentação", o método foi desenvolvido e elaborado em cinco fases, e cada fase é detalhada e especificada por meio de etapas. Essas etapas foram aplicadas na sequência para permitir agrupar e unir os processos existentes na criação e aplicação das práticas que utilizam a modelagem 3D (e conceitos de Fabricação Digital) para ensino de POO. Com o objetivo de criar um processo com este conjunto de etapas, foram alinhados os métodos de SO para que cada fase fosse sistematicamente explorada. As fases organizadas estão abaixo:

- *Fase 1*: Atributos e atividades que trouxessem uma avaliação do cenário e trouxessem quais as limitações e imposições sobre o problema. Nesta fase é aplicado o método PAM;
- *Fase 2*: Levantamento dos elementos e identificação das unidades semânticas. Etapas desta fase estão ligadas ao método SAM;
- *Fase 3*: Criação do relacionamento entre os valores e comportamentos obtidos nas fases anteriores, utilizando normas para relacionar e avaliar o desempenho dos alunos. Esta fazer é ligada diretamente ao método NAM;
- *Fase 4*: Aplicação do método na prática. Esta fase foram realizadas atividades em um ambiente de ensino, onde o professor realizou a fase prática do método para os alunos, ainda que em caráter inicial;
- *Fase 5*: Avaliação e aquisição de resultados. Esta fase tem o intuito de aplicar avaliações aos alunos participantes do método, e

consequentemente obter os resultados necessários para refinamento do método.

4.2 Detalhamento MEPOOMD

A parir do *feedback* da experimentação inicial e da estrutura do ADDIE, o método foi elaborado em cinco fases, em cada fase temos a elaboração e execução de cada passo. Na Figura 11 é possível visualizar o diagrama definindo as fases e passos a serem realizados de acordo com o método.

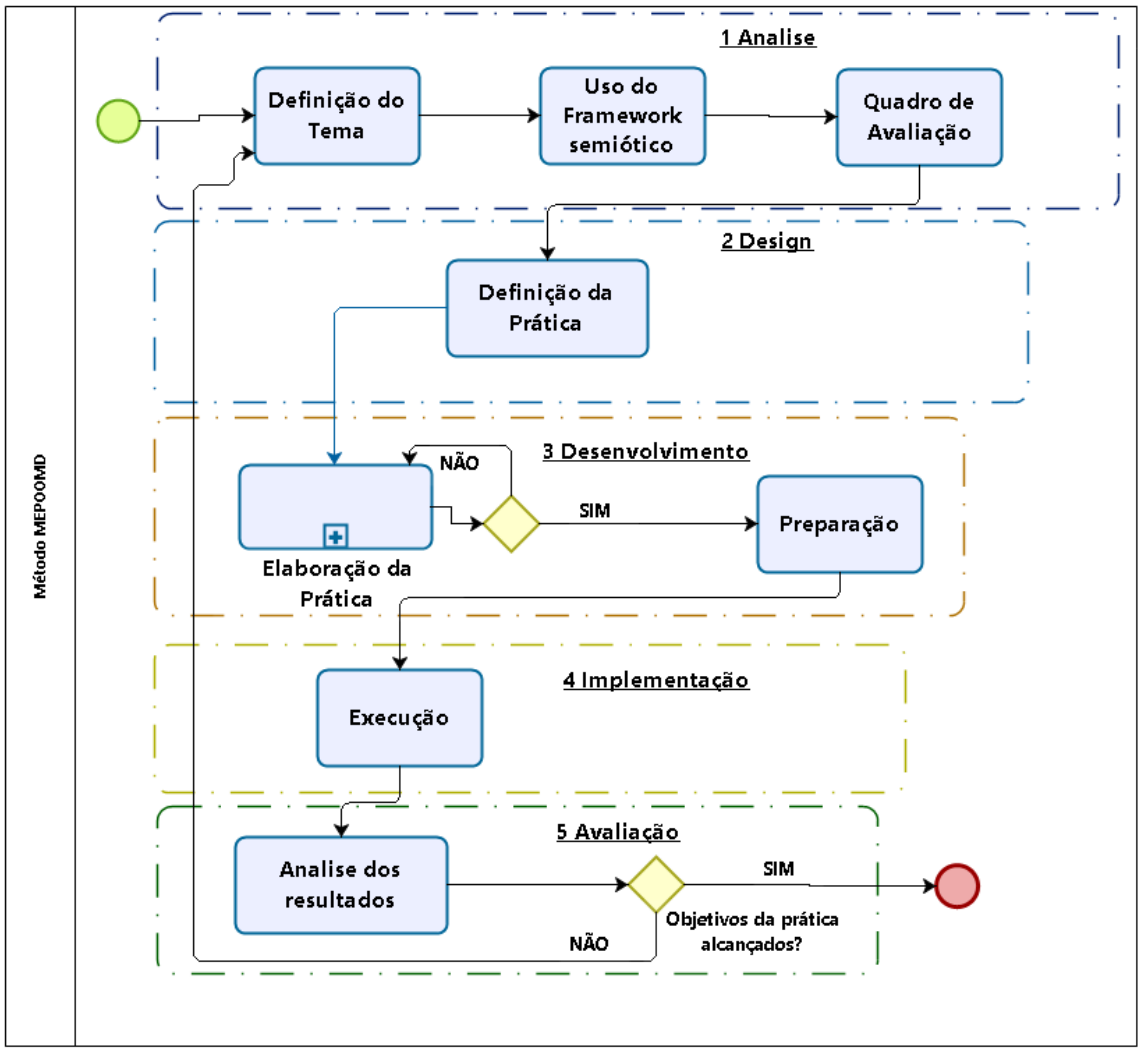


Figura 11 - Visão geral MEPOOMD

4.2.1 Análise

A primeira fase do MEPOOMD é a de análise. Esta é a fase para coletar informações para entender o cenário, as necessidades dos alunos envolvidos e as necessidades do ambiente em que está sendo aplicado o método; para que seja possível o desenvolvimento e aplicação do mesmo, gerando um relatório de diagnóstico e conhecimento. Nesta fase, se faz necessário a aplicação de questionários, entrevistas, solicitar os dados para entender quais são as habilidades e competências que deverão ser desenvolvidas e trabalhadas durante a aplicação do método. Exemplos de informações que podem ser coletadas nesta fase são: relatórios de perfil, relatórios de performance indicando o perfil e a capacidade que poderá ser trabalhada durante a aplicação.

Conforme a Figura 10, esta fase contém três etapas: Definição do Tema (subseção 4.2.1.1) com descrições textuais estruturadas dos principais elementos da fase de análise da ADDIE (*cf.*, Seção 2.1.1); o uso do *Framework* Semiótico (subseção 4.2.1.2) no qual utilizamos o referencial semiótico para classificar e analisar os resultados do Passo 1.a de acordo com a escala semiótica. E o terceiro, Quadro de Avaliação (subseção 4.2.1.3), os interesses, as questões e os problemas dos alunos e instrutores (quanto ao uso de modelagem digital e fabricação para o ensino de POO) são analisados de acordo com o quadro de avaliação. Os resultados desta fase serão o ponto de partida para a segunda fase.

4.2.1.1 Definição do Tema

Nesta etapa da Fase 1 é necessário apresentar o contexto que compreende o diagnóstico da situação em que se encontra o ambiente e pessoas envolvidas, como alunos e professores. Ao realizar este mapeamento e apresentar o contexto basta listar alguns dados importantes relacionados a aplicação do método como: Quem é o público alvo do método; Qual o cenário atual em relação ao ambiente; quais são os conceitos que serão trabalhados; Quais serão as técnicas utilizadas; quais são os possíveis impactos e resultados. Para a definição e estruturação deste mapeamento é sugerido o seguinte esqueleto:

- Público alvo: o perfil, o conhecimento que os principais envolvidos no método possuem.

- Cenário atual em relação ao ambiente: descrição de partes e espaços que pertencem ao cenário em que será realizado a aplicação do método. Uma breve listagem de recursos e necessidade para a ordenação do ambiente.
- Conceitos que serão trabalhados: relação de práticas e conceitos que serão utilizados durante o desenvolvimento e codificação de atividades, desafios e provas.
- Técnicas utilizadas: relação de técnicas empregadas para a visualização, aplicação e desenvolvimento do código durante a prática.
- Impactos e resultados: os resultados e impactos serão avaliados ao findar da execução da tarefa ou avaliação. Estes resultados servem para avaliar e mensurar se é necessário voltar ao primeiro passo ou pode seguir e apenas iniciar uma nova execução durante a aplicação do MEPOOMD.

4.2.1.2 Uso do Framework Semiótico

A utilização do framework semiótico na Fase 1 ocorre no reconhecimento dos *stakeholders* envolvidos por meio da "cebola semiótica" (Liu, 2000). Esta ação deverá acontecer até que exista densidade na análise dos *stakeholders*, conforme a entendimento dos papéis do "docente" e "aluno". Nesse âmbito, além de empregar a prática, o docente tem a função de facilitador e formador. Os alunos têm como alvo a resolução do problema sugerido, reconhecimento da solução e execução por meio da programação e desenvolvimento com uma linguagem OO, guiados por modelos tridimensionais que exibem os conceitos. Conforme a Figura 12, a "cebola semiótica" organiza os problemas relacionados à prática e ao envolvimento dos *stakeholders*, sendo categorizados de acordo com os níveis informal, formal e técnico.

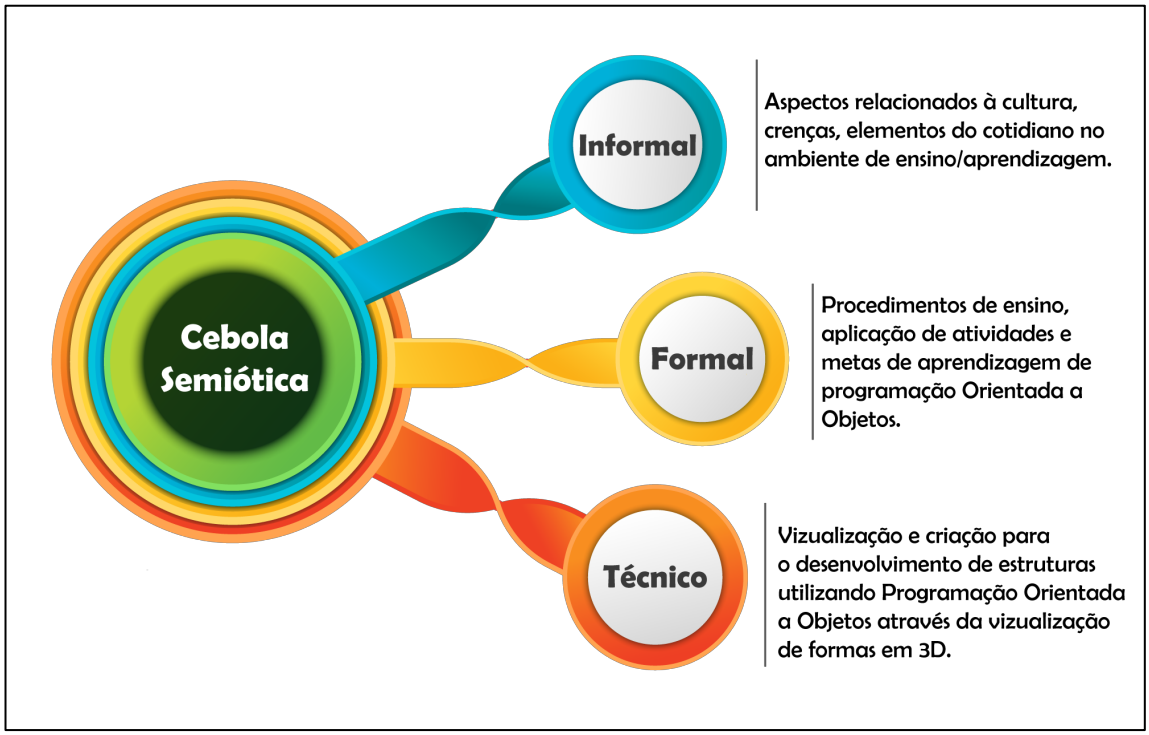


Figura 12 - Cebola semiótica

Na Figura 12 acima podemos visualizar aspectos dos níveis informal e formal, que formam os dados e resultados que o aluno irá utilizar. No nível técnico da cebola semiótica, é composto pela prática de ensino e aprendizagem. Esta estrutura serve para avaliar e entender quem são os *stakeholders*, entendendo que é possível identificar as partes interessadas que influenciam diretamente ou indiretamente, as mesmas são identificas e alocadas nas camadas informal, formal e técnica. Abaixo estão listados níveis da cebola semiótica e como são praticadas:

- Informal: Neste nível são definidos os significados e intenções, são estabelecidos responsabilidade e compromissos de cada parte envolvida. As relações informais entre o discente e docente, onde temos a intenção de desenvolver e demonstrar o conteúdo. Esta relação cria um laço e demonstram os valores envolvidos.
- Formal: Neste nível as regras são desenvolvidas. Definição de como será realizado a aplicação de atividades, também definidos os objetivos a serem conquistados.
- Técnico: A definição do nível formal é necessária para que o nível técnico seja iniciado e explorado. O desenvolvimento e programação só é capaz

se os níveis estiverem cumpridos, é neste nível que se torna possível a criação e desenvolvimento em POO.

A classificação e organização será repetida para cada nível da escada semiótica podendo haver ou ser criadas questões que ajudem a identificar os elementos. O framework semiótico serve como guia para a análise dos problemas e soluções para o ensino de POO. Classificar e organizar os problemas é a intenção e relacionar os domínios dos objetivos didáticos do nível físico ao social de acordo com a escada semiótica (Liu, 2000). Conforme apresentado anteriormente na Figura 6, o framework semiótico consegue organizar em níveis aspectos da plataforma tecnológica e do sistema de informação humano. Em cada um dos níveis na Figura 6, são classificados e organizados os problemas e também possíveis soluções aplicadas neste método.

4.2.1.3 Quadro de Avaliação

Nesta fase é realizada a avaliação referente à real intenção e motivação de discentes e docentes. É necessária uma avaliação para verificar se foi atingido o nível esperado de aprendizagem em cada etapa. Caso, em alguma das fases anteriores este nível esperado não tenha sido alcançado, se faz necessário a volta à primeira etapa do nível de análise.

O uso do quadro de avaliação no MEPOOMD deve conter os tópicos citados abaixo para avaliar a necessidade de retornar à etapa 1 ou seguir e começar a Fase 2 (Design):

- Identificação dos docentes e discentes.
- Mudança desejada.
- Restrições e níveis de aprendizado.
- Métodos pedagógicos.
- Prazo de aplicação e conclusão.
- Impactos no ambiente e no profissional.

Para concretizar e avaliar se a fase de análise está completa, se faz necessário a listagem de dados relevantes para a avaliação da Fase 1.

4.2.2 Design

Esta fase inclui o design do planejamento dos instrutores de acordo com o ADDIE. Esta fase tem somente uma etapa, Definição da prática (subseção 4.2.2.1), que é apoiado pelo SAM. Aplica-se o SAM para construir um diagrama de Ontologia, que inclui *affordances* e dependências ontológicas relacionadas ao problema em foco. O diagrama de Ontologia contribui apoiando projetistas na identificação e esclarecimento das unidades semânticas dos conceitos usados no projeto ADDIE.

4.2.2.1 Definição da prática

Nesta etapa da fase dois, o objetivo é determinar de forma clara os agentes e *affordances,* utilizando uma análise de domínio. Baseado na pesquisa de Liu (2000) e Bonacin (2004), as etapas abaixo seguem as fases da AS. A modelagem deve seguir os passos apresentados na Seção 2.3.2, que inclui:

- Definição do problema.
- Geração de *affordances* candidatos.
- Agrupamento de candidatos.
- Diagramação ontológica.

A partir do diagrama de ontologia, descrições e a definição do design da prática é desenvolvida de acordo com o ADDIE.

4.2.3 Desenvolvimento

Conforme proposto pelo modelo ADDIE, esta fase inclui a elaboração de materiais e produtos utilizados na instrução de ensino. As normas são usadas para representar o comportamento esperado dos alunos e instrutores durante as atividades de aprendizagem. Envolve o momento da definição, dentre outros elementos, das estratégias de ensino, dos recursos didáticos, das ferramentas e tecnologias, das modalidades de avaliação, da preparação dos docentes que deverão atuar no processo das atividades do aprendiz, além da finalização e da análise da coerência do que foi concebido nas fases anteriores.

Esta fase inclui duas etapas: elaboração da prática e preparação. Na etapa de elaboração é realizado o desenvolvimento das práticas de forma iterativamente elaborada (subseção 4.2.3.1), e na etapa preparação, é feito o preparo para a implementação da atividade (subseção 4.2.3.2).

4.2.3.1 Elaboração da prática

Nesta etapa da fase 3 ocorre a identificação, análise e modelagem das normas que são utilizadas no momento da prática, a partir da aplicação é feita uma análise detalhada das normas representando o comportamento de cada agente em seu nível tanto pragmático quanto social (Stamper *et al.*, 2000). A intenção de comunicar e os comportamentos dos agentes pode ser descritos por normas no nível pragmático. Já a demonstração expressa as expectativas, compromissos e crenças é descrita por normas nível social. Nesta etapa existe duas sub etapas:

- Quadro de Normas Comportamentais: baseada na utilização do método NAM, esta aliada aos *affordances* e as normas comportamentais. Serão elaborados quadros demonstrativos de normas, listando o agente, *affordances* e informações sobre a norma utilizando a Lógica Deôntica.
- Lógica Deôntica: está é uma forma de descrição, utilizando os detalhes sobre os *affordances* e agentes no cenário que é determinado pela norma criada será demostrada em um quadro de normas como descrito na Tabela 10.

Tabela 10 - Modelo Quadro de Normas Comportamentais

Norma	***{nome da norma}***
Agentes	*{lista de agentes}*
Affordances	*{lista de affordances}*
Detalhamento da norma	*{expressão em Lógica Deôntica}*

4.2.3.2 Preparação

Após a definição das normas, inicia a organização e elaboração de quadros de resposta de todas as normas criadas. A criação destes quadros de resposta serve para

demostrar que a norma modelada considerada segue o que foi elicitado anteriormente. Assim, é possível visualizar as ações que foram realizadas de forma assertiva, assim como no exemplo da Tabela 11.

Tabela 11 - Tabela de Respostas

{Norma}				
	Questão 1	**Questão 2**	**...**	**Questão N**
Aluno 1				
Aluno 2				
Aluno ...				
Aluno N				

4.2.4 Implementação

Esta fase inclui a entrega de artefatos para os alunos, bem como a aplicação do método. Nesta fase, as atividades de ensino são executadas no ambiente de aprendizagem. As normas comportamentais (construídas na fase anterior) apoiam os instrutores, explicitando o comportamento esperado dos alunos durante a execução das atividades. Esta fase é feita de apenas uma etapa, a Execução (subseção 4.2.4.1).

4.2.4.1 Execução

Nesta etapa é realizada a definição do ambiente e os recursos que o computador deve possuir para que seja possível a execução da tarefa, por exemplo o software NetBeans para o desenvolvimento. Para o desenvolvimento da prática são empregados todos os elementos desenvolvidos nas fases anteriores do MEPOOMD. Na elaboração de uma tarefa é obrigatório a utilização de três elementos: (1) Enunciado; (2) regras e; (3) metas a serem concluídas. Abaixo está detalhado cada um destes elementos:

- **Enunciado:** texto elaborado para detalhar o desafio de desenvolvimento empregado, assim como os objetos a serem criados, classes, hierarquias e instanciação de variáveis.

- **Regras:** texto ao decorrer do enunciado que demostra quais são as características e detalhes que devem ser seguidas no momento de criação e desenvolvimento.
- **Metas a serem concluídas:** texto ao final do enunciado que lista algumas ações que o programa deve cumprir para que seja avaliado com a nota máxima. Estas metas ajudam o aluno a entender qual o principal objetivo da criação, além de motivar o mesmo a produzir e seguir as metas.

4.2.5 Avaliação

Esta fase visa obter avaliações dos resultados (subseção 4.2.5.1). Os formulários de avaliação são aplicados aos alunos que participaram das atividades para obter os resultados necessários para refinar as atividades e o próprio método.

4.2.5.1 Avaliação dos Resultados

A realização da análise e avaliação dos resultados se dá inteiramente por existir o quadro de normas (Tabela 10) para que sejam utilizadas como base para entender o que e como deve ser avaliado em relação ao código desenvolvido. Após a utilização do quadro de normas é necessário a tabela de respostas, onde o docente define o resultado de cada meta do quadro de normas por meio de três níveis: (1) completo; (2) parcialmente completo; e (3) incompleto. Estes valores são somados para gerar a nota final em relação a atividade desenvolvida pelo aluno.

Após a elaboração das tabelas de resposta e utilizada uma regra simples de cálculo que cruza os dados para obter a nota final que o aluno obteve referente a tarefa desenvolvida. Ao avaliar o código desenvolvido pelo aluno, o docente deve entender e relacionar o que foi ensinado com a compreensão que o aluno obteve em relação ao conteúdo.

5. APLICAÇÃO DO MEPOOMD

Neste capítulo é apresentada a aplicação de práticas pedagógicas baseadas no método MEPOOMD. Para tanto, foi realizada uma avaliação com o objetivo de verificar a influência do MEPOOMD no ensino de POO, tanto em termos quantitativos quanto qualitativos, em grupos de alunos de cursos técnico e superior. A Seção 5.1 descreve os materiais empregados e os participantes do estudo, incluindo quais as características dos grupos de análise. Na Seção 5.2 é detalhada como foi realizada a aplicação do MEPOOMD, descrevendo as etapas empregadas.

5.1 Participantes e contexto de aplicação

Nesta seção é apresentada a descrição dos participantes envolvidos na aplicação do método, incluindo uma análise detalhada sobre o perfil destes alunos e um levantamento das ferramentas utilizadas para a prática e execução do método.

5.1.1 Descrição dos participantes

Para a aplicação do método foram selecionados alunos que cursam Técnico em Informática junto ao Ensino Médio e alunos da graduação de Sistemas de Informação do UNASP – Hortolândia, localizado na Rua Pastor Hugo Gegembauer, número 265, no bairro Parque Ortolândia, na cidade de Hortolândia no estado de São Paulo.

Conforme detalha a Tabela 12, o estudo contou com um total de duzentos e sessenta e um (*261*) alunos que foram divididos em dois níveis, sendo eles: curso técnico (Nível 1- com *107* alunos) e graduação (Nível 2 – com *154* alunos).

Para atingir o objetivo da avaliação, supracitado, foi realizado um pré-teste com todos os alunos. O mesmo teste foi aplicado independentemente de níveis e grupos. Esta avaliação chamada de pré-teste é detalhada no ANEXO III desta dissertação. A presente avaliação foi composta por *20* questões, sendo que 8 delas eram referentes a lógica de programação e *12* referentes a POO. São questões de múltipla escolha, sendo que o aluno poderia utilizar o software netbeans para auxiliá-lo em questões referentes a lógica ou a POO. Na figura 13 é possível visualizar o percentual de acertos na aplicação desta prova.

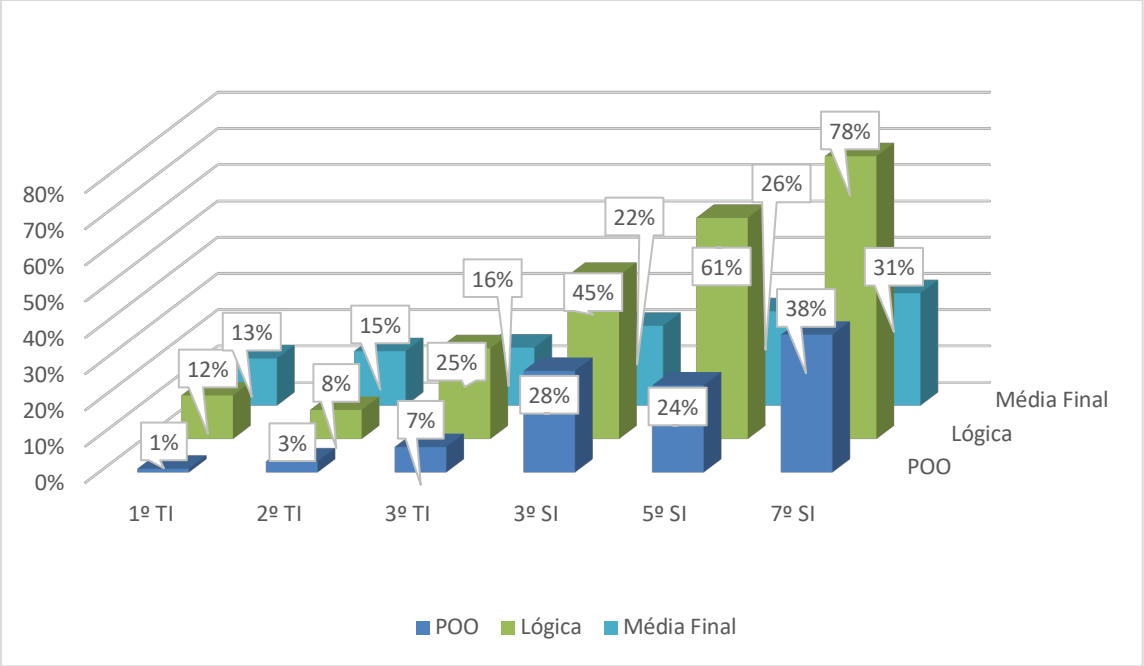


Figura 13 - Percentual de acerto no pré-teste por turmas

Por meio dos resultados contidos na Figura 13, notamos que os alunos pertencentes ao curso técnico não possuem praticamente nenhum conhecimento sobre POO, já os alunos da Graduação têm um nível melhor de aproveitamento quando o assunto é POO. Referente a lógica de programação também é possível notar uma grande diferença, por exemplo entre o 3º TI e o 3º SI, 20% de diferença entre os dois grupos no que se refere a lógica de programação. Para que estes discentes tenham o melhor aproveitamento possível da aplicação do método MEPOOMD foi realizada a separação em dois níveis, nível 1 com um total de *107* alunos e o nível 2 com *154* alunos, como na Tabela 12.

Tabela 12 - Aplicação MEPOOMD, Total de Alunos

Aplicação MEPPOMD Total: 261 alunos	
Nível 1	**Nível 2**
Total: 107 alunos	Total: 154 alunos

Os discentes pertencentes ao Nível 1 são alunos do primeiro, segundo e terceiro anos do curso técnico, como todos do nível 1 já trabalhavam com algoritmos foi possível

apresentar os conceitos de POO. O critério para divisão dos grupos deste nível foram os anos em que os discentes pertenciam, então foram sorteados os mesmos, ou seja, os discentes do primeiro ano do curso técnico formaram o Grupo 1 de *45* alunos, foi subdivido em mais dois subgrupos para formarem o subgrupo de teste com *23* alunos expostos ao método, e o subgrupo de controle com *22* alunos que apenas realizaram a revisão do conteúdo sem a utilização do método. Os estudantes foram escolhidos de forma aleatória.

Grupo 2 com *35* alunos do segundo ano do ensino médio que foram subdivididos em dois grupos de forma aleatória. O subgrupo de teste continha *18* alunos e outro como subgrupo de controle com *17* alunos. E por fim, no nível 1 foi formado o Grupo 3 com *27* alunos do terceiro ano e foram subdivido em dois subgrupos, o teste de *14* alunos e o controle com *13* alunos.

A Tabela 13 apresenta uma síntese da divisão dos estudantes no nível 1, ou seja, dos alunos do curso técnico em Tecnologia da Informação. Esta tabela também apresenta o percentual de acerto dos subgrupos no pré-teste. Os subgrupos de teste e controle de um mesmo grupo apresentaram desempenho similar no pré-teste.

Tabela 13 - Nível 1, Aplicação MEPOOMD - Curso técnico Tecnologia da Informação e desempenho no pré-teste

Nível 1 **Curso Técnico Tecnologia da Informação** **Total: 107 alunos**					
Grupo 1		**Grupo 2**		**Grupo 3**	
45 alunos		35 alunos		27 alunos	
Teste	**Controle**	**Teste**	**Controle**	**Teste**	**Controle**
23 alunos	22 alunos	18 alunos	17 alunos	14 alunos	13 alunos
29%	30%	33%	30%	34%	28%

Neste nível é possível notar por meio da Figura 14 que a maioria dos alunos são do sexo masculino, pois *73* dos 107 alunos pertencem ao sexo masculino. Neste mesmo nível temos *34* do gênero feminino. Na mesma imagem é possível visualizar que

conforme temos novas turmas o sexo masculino vem se impondo de forma mais significativa no curso técnico em Informática do UNASP-HT.

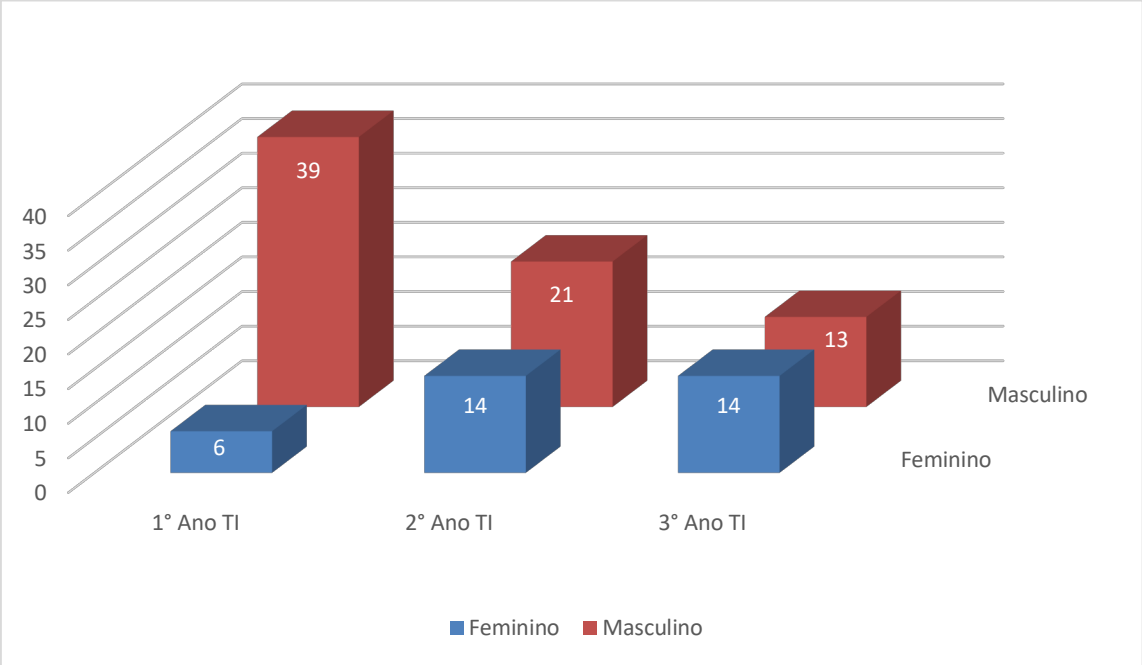


Figura 14 - Grupos por gênero no Nível 1

Na Figura 15 é possível visualizar que a auto avaliação dos alunos do nível 1 é representada por um baixo nível de conhecimento e rendimento relacionados a programação. Na pergunta "*Como você descreve seu rendimento com programação*?" *47%* dos alunos mencionaram ter um rendimento ruim, ou seja, a maioria; *28%* se descrevem com um nível regular; *14%* com nível bom; e, apenas, *8%* se auto descrevem com nível muito bom. Na segunda pergunta "*Você tem afinidade com programação orientada a objetos?*" *86%* dos alunos responderam que não tem afinidade com POO. Apenas 3% de *107* aluno mencionaram ter afinidade.

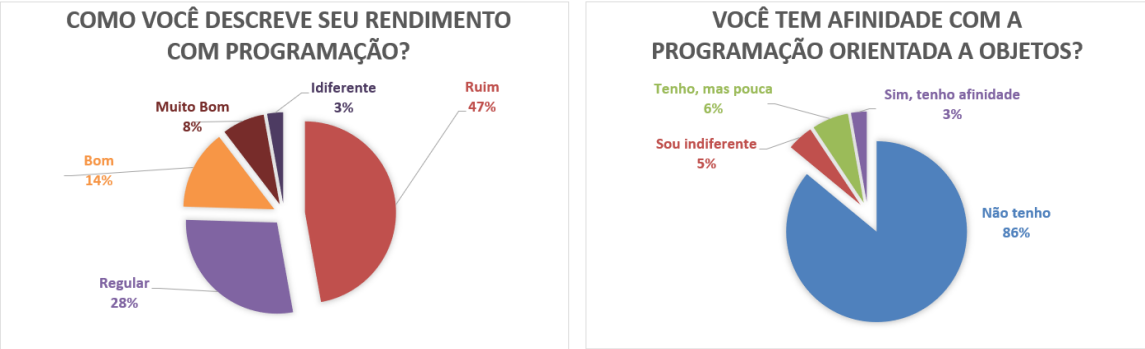


Figura 15 - Autoavaliação dos alunos do Nível 1 sobre programação

O Nível 2 com total de *154* alunos, foram formados 3 grupos, o critério para a divisão destes grupos foram os três semestres em que pertenciam na graduação 3°, 5° e 7° Semestre. Não foi realizado o estudo no 1° semestre pois não possuíam conhecimento prévio mínimo para a aplicação do método. De forma aleatória, o Grupo 1 com *55* alunos pertencentes ao 3° semestre da graduação de Sistemas de Informação, este grupo foi subdividido em dois. O subgrupo de teste foi tinha *29* alunos e um subgrupo de controle com *26* alunos. As diferenças numéricas entre os subgrupos se deve a desistência de alunos.

O Grupo 2 com *51* alunos pertencentes ao 5° semestre da graduação e foram subdivididos em dois subgrupos, um de teste de *23* alunos e o outro de controle de *26* alunos. Já o grupo 3 com *48* alunos pertencentes ao 7° semestre, subdivididos em dois subgrupos, um de teste com *24* alunos e outro de controle com *24* alunos.

Na tabela 14 podemos visualizar como foi realizada a divisão no nível 2, ou seja, dos alunos do curso de Graduação em Sistemas de Informação. Esta tabela também apresenta o percentual de acerto dos grupos e subgrupos no pré-teste. Os subgrupos de teste e controle de um mesmo grupo apresentaram desempenho similar no pré-teste.

Tabela 14 - Nível 2, Aplicação MEPOOMD - Graduação em Sistemas de Informação e desempenho no pré-teste

Nível 2 Graduação Sistemas de Informação Total: 154 alunos					
Grupo 1		**Grupo 2**		**Grupo 3**	
55 alunos		51 alunos		48 alunos	
Teste	**Controle**	**Teste**	**Controle**	**Teste**	**Controle**
29 alunos	26 alunos	26 alunos	23 alunos	24 alunos	24 alunos
50%	48%	57%	54%	62%	62%

Neste nível é possível notar por meio da Figura 16 que a maioria dos alunos são do sexo masculino, pois *130* dos *154* alunos pertencem ao sexo masculino, neste temos apenas *24* do gênero feminino.

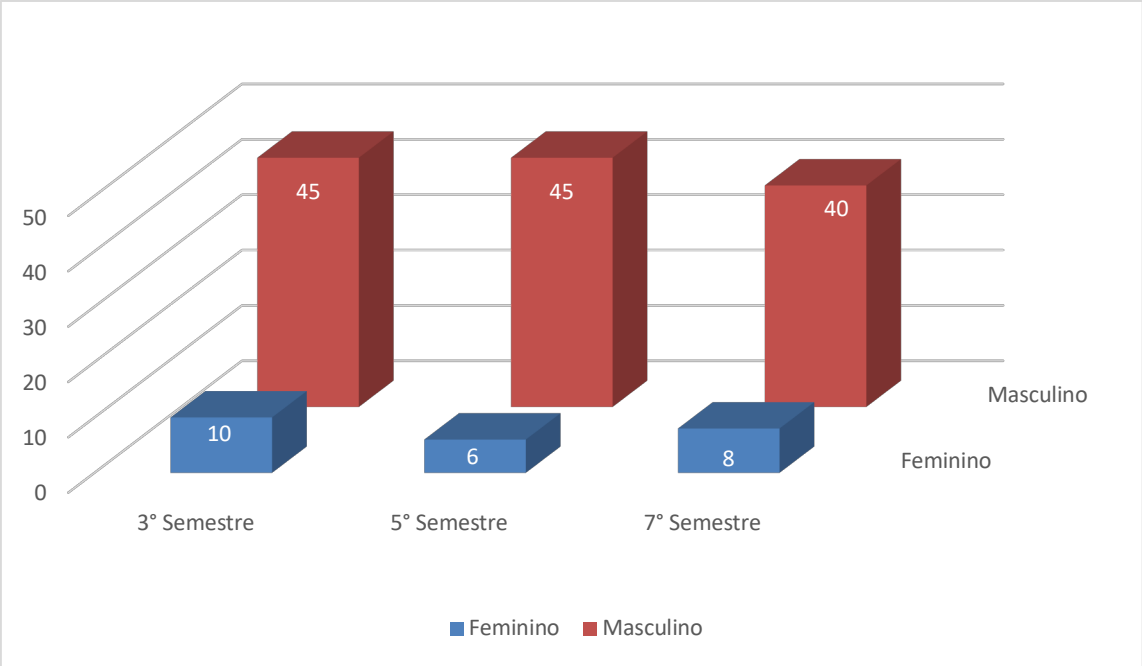


Figura 16 - Grupos por gênero no Nível 2

Na Figura 17 é possível visualizar que a autoavaliação dos alunos do nível 2 é relativamente boa ao que se refere ao conhecimento e rendimento relacionados a

programação. Ao comparar com a Figura 15 podemos visualizar uma grande diferença entre a autoavaliação destes dois níveis, pois aqui temos como resposta da pergunta "*Como você descreve seu rendimento com programação*?" *19*% mencionam que são muito bons em programação; *30*% se descrevem como bons em programação; *38*% regular e apenas *11*% se auto definem como ruins em programação. Quanto a pergunta "*Você tem afinidade com programação orientada a objetos?*" também é possível notar uma melhor autoavaliação do nível 2 quando comparada ao nível 1, pois *17%* responderam ter afinidade com a POO; *26%* tem afinidade, mas pouca; *52%* não tem e *5%* são indiferentes.

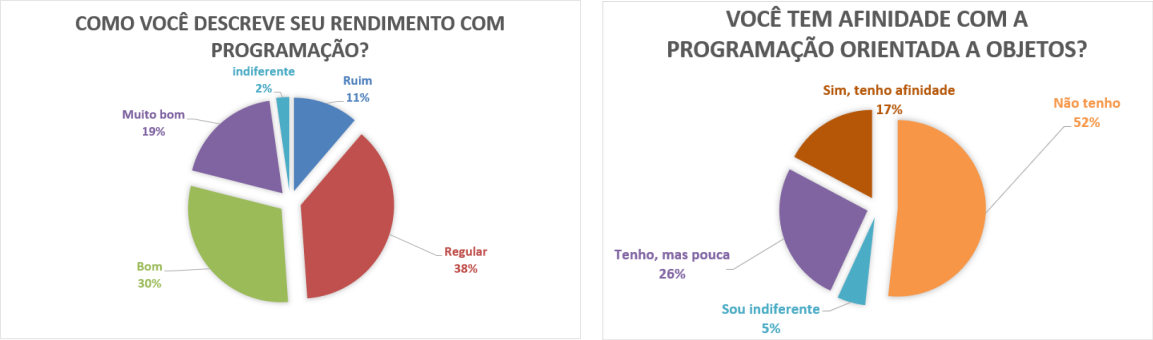


Figura 17 - Autoimagem sobre programação, alunos do nível 2

5.1.2 Contexto de aplicação

O método foi apresentado aos alunos na forma de oficinas, que ocorreram no período de 6 semanas, três vezes por semana, durante o período entre o mês de abril e maio de 2017. Como conteúdo das oficinas foram apresentados os principais conceitos de POO tais como: objetos, classes, hierarquia, entre outros citados no Capítulo 2.1.2. Também foram apresentados os conceitos e formas de modelagem 3d (e fabricação digital) que iriamos trabalhar (Capitulo 2.2).

Essas oficinas tiveram como objetivo trabalhar conceitos de POO em diversos níveis de abstração, possibilitando criar analogias entre objetos 3D (virtuais) e conceitos de fabricação digital com os conceitos de POO. Com isso, espera-se uma maior motivação e um contato direto com a programação de forma mais íntima possível. Durante a semana era realizado três encontros, no primeiro era apresentado os conceitos de POO, no segundo encontro eles eram colocados em prática, e então passado um

exercício que deveria ser entregue para o terceiro encontro, onde era realizada a correção do exercício dado no encontro anterior. Fotos sobre o ambiente em que os alunos realizaram os encontros semanais e exercícios são apresentadas no Anexo I.

Para iniciar a aplicação do método, foi realizada a aplicação de um questionário para verificação do perfil do aluno (Anexo II). Após a aplicação do questionário de perfil do aluno foi aplicado o Pré-Teste (Anexo III). Durante as 5 primeiras semanas de aplicação do MEPOOMD foram utilizados os softwares Netbeans e Blender para realizar as aulas práticas nos laboratórios do UNASP – Hortolândia. Esses encontros ocorreram durante a aula, pois todos tinham disciplinas voltadas para programação, onde o método foi aplicado aos subgrupos de teste. A partir do mês de maio, na sexta semana do método, foi aplicado o pós teste para todos os grupos de alunos, possibilitando a verificação do nível de POO (Anexo III). Na sequência, foi aplicado o questionário para verificar a eficiência do método para os grupos de teste e controle, conforme apresenta Anexo IV. Este último questionário teve como objetivo avaliar as opiniões dos alunos após a aplicação e estudos de POO.

Para a execução do método com seus respectivos exercícios práticos, aulas expositivas e projetos exploratórios foram utilizados os laboratórios de informática do UNASP-HT. Foram utilizadas em torno de *238* maquinas, todas com Windows 10 e com os softwares mencionados na subseção 5.1.3 previamente instalados. Este ambiente ficou disponível para o aluno tanto para uso durante as aulas como também após as aulas para realização de práticas com vídeo aulas. A Figura 18 apresenta um laboratório onde os discentes realizaram as atividades.

Figura 18 - Laboratório de informática e a aplicação do MEPOOMD

5.1.3 Ferramentas empregadas

Nas práticas foram utilizados os softwares NetBeans V. 8.1 para exercícios práticos de POO (Figura 19) e Blender V. 2.6 para exibição e criação de modelos 3D e explorar conceitos de Fabricação Digital (Figura14).

A Figura 19 apresenta o conhecimento prévio dos alunos sobre as ferramentas de desenvolvimento de POO (Netbeans, Eclipse e Visual Studio). Entre as três, o Netbeans foi mencionado mais vezes. Este software de desenvolvimento em Java foi escolhido como principal pois *71,7%* dos alunos já haviam utilizado a mesma. Todos os alunos do curso de superior em Sistemas de Informação já conheciam o Netbeans, bem como *98%* dos alunos do curso técnico em tecnologia de informação. Este fato levou a escolha do Netbeans como ferramenta de desenvolvimento de POO para o curso.

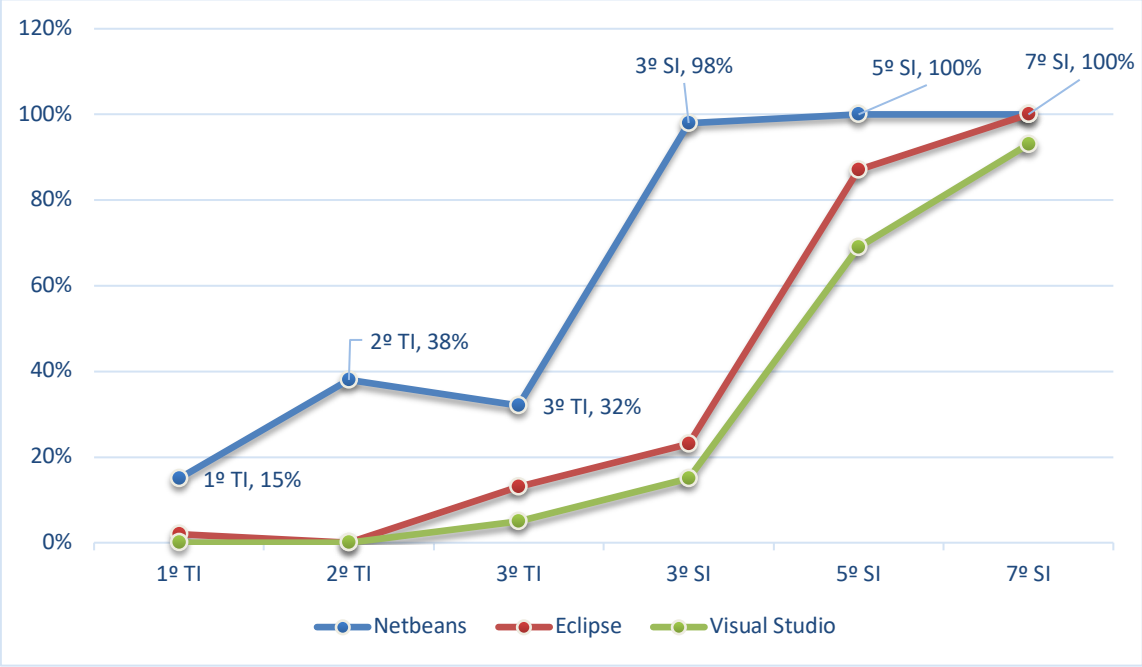


Figura 19 - Conhecimento prévio sobre ferramentas de desenvolvimento para POO

Além disso, o software NetBeans foi escolhido ser ambiente de desenvolvimento integrado gratuito e de código aberto para desenvolvedores de software. O NetBeans é uma IDE que pode ser executado em muitas plataformas, como Windows, Linux, Solaris e MacOS. Ele oferece aos desenvolvedores todas as ferramentas necessárias para criar aplicativos profissionais de desktop, empresariais Web e móveis multiplataformas. Na Figura 20 é possível visualizar um exemplo de exercício realizado com os alunos utilizando o NetBeans.

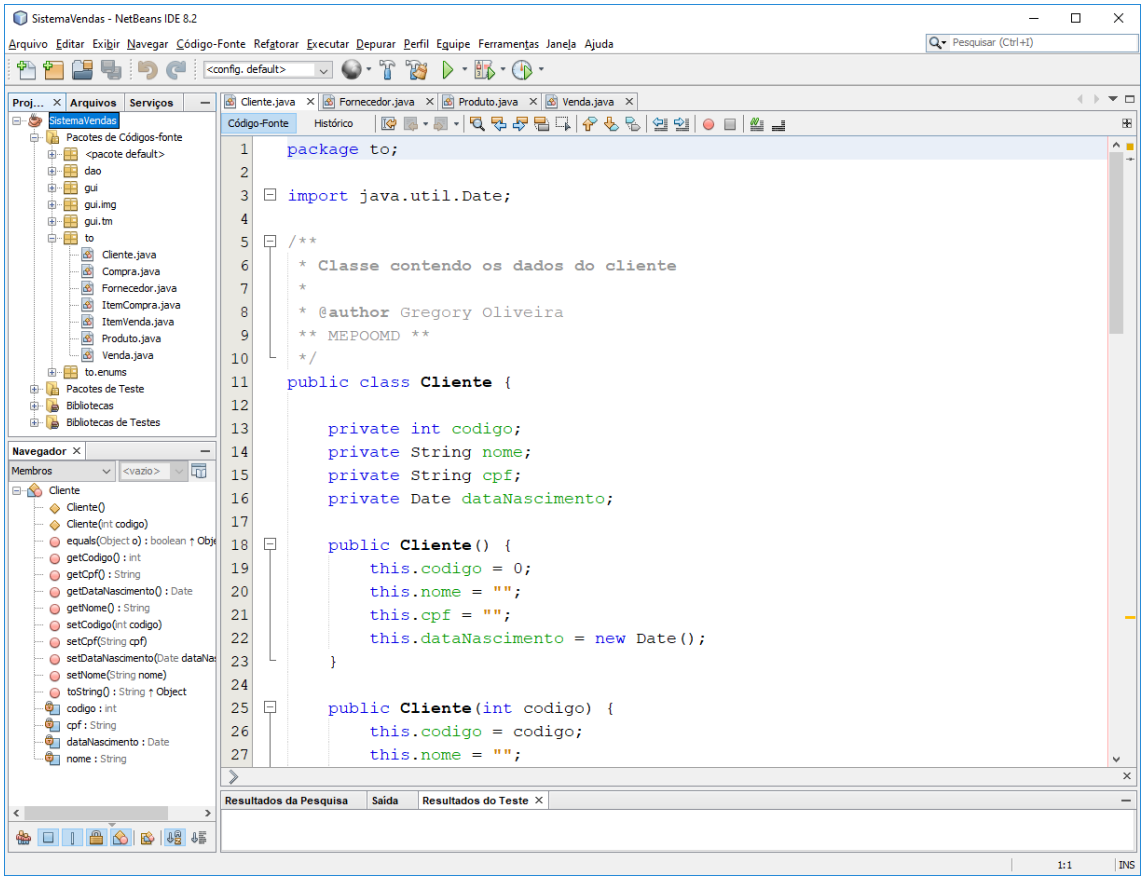


Figura 20 – NetBeans, software de desenvolvimento Java

O software Blender (Figura 21) foi escolhido por ser um programa de computador de código aberto, para modelagem, animação, texturização, composição, renderização, edição de vídeo e criação de aplicações interativas em 3D, tais como jogos, apresentações e outros.

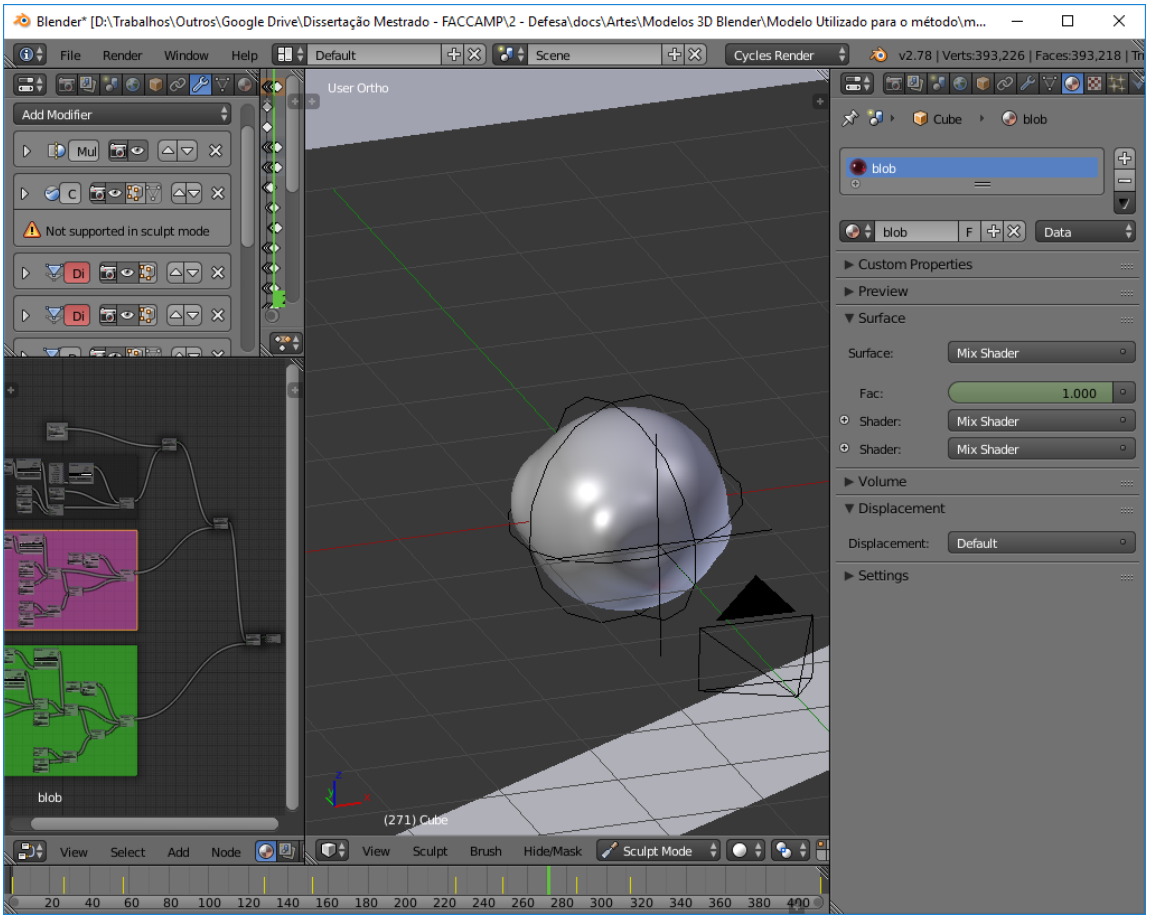


Figura 21 – Blender, software de modelagem 3D

O Blender foi utilizado no método para criação das formas 3D. Apenas *12%* dos alunos na média já haviam utilizado o Blender (Figura 22). Devido a este número reduzido de alunos que já conheciam a ferramenta foram necessárias aulas de desenvolvimento em ambiente virtual 3D.

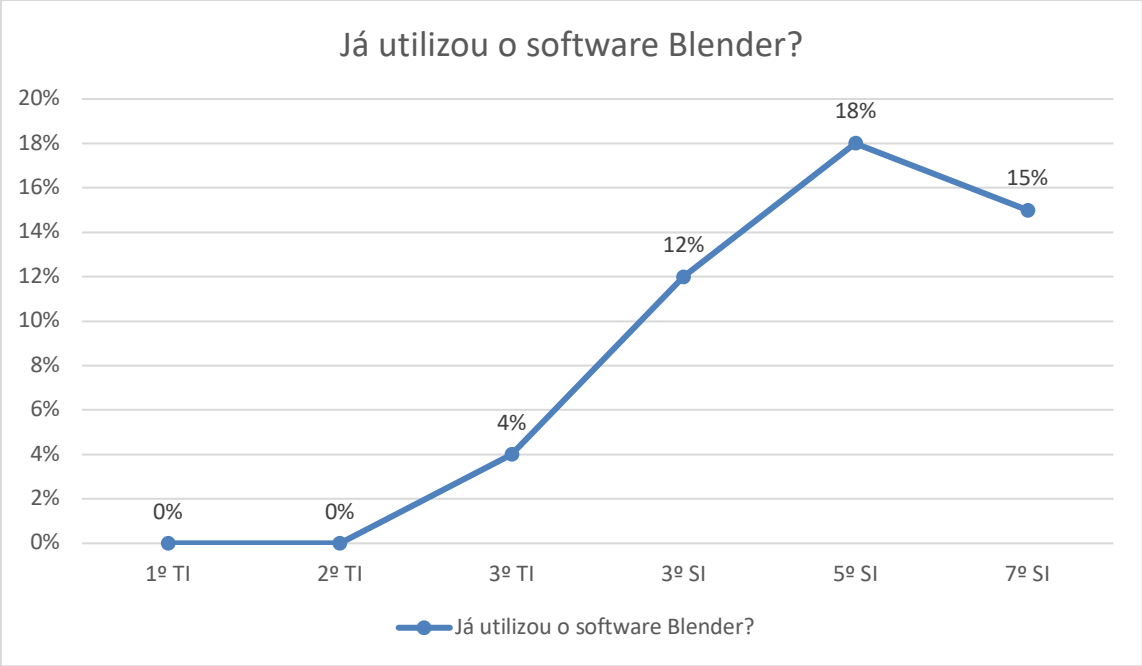


Figura 22 - Resultado da pesquisa sobre a utilização do Blender

Estas aulas foram feitas de modo virtual, por meio da ferramenta Moodle customizada pela instituição, chamada UNASP Virtual. Como é possível visualizar na Figura 16, foi elaboradas *16* vídeos aulas, contendo práticas e exercícios a serem realizadas pelos alunos como pré-requisito para o início das aulas com a utilização do método. Na Figura 23 é possível visualizar as *16* vídeo aulas que foram feitas, com os seguintes temas:

- Vídeo Aula 1: Interface – Parte 1
- Vídeo Aula 2: Interface – Parte 2
- Vídeo Aula 3: Transformações Geométricas
- Vídeo Aula 4: Modo Edição
- Vídeo Aula 5: Iniciando Modelagem
- Vídeo Aula 6: Aplicando Suavidade e divisão de Meshes
- Vídeo Aula 7: Modo Esculpir
- Vídeo Aula 8: Extrusions
- Vídeo Aula 9: Modificadores
- Vídeo Aula 10: Personagem
- Vídeo Aula 11: Aplicando Material
- Vídeo Aula 12: Câmera
- Vídeo Aula 13: Animação

- Vídeo Aula 14: Armature
- Vídeo Aula 15: Armature no Personagem
- Vídeo Aula 16: Movimentando o Personagem

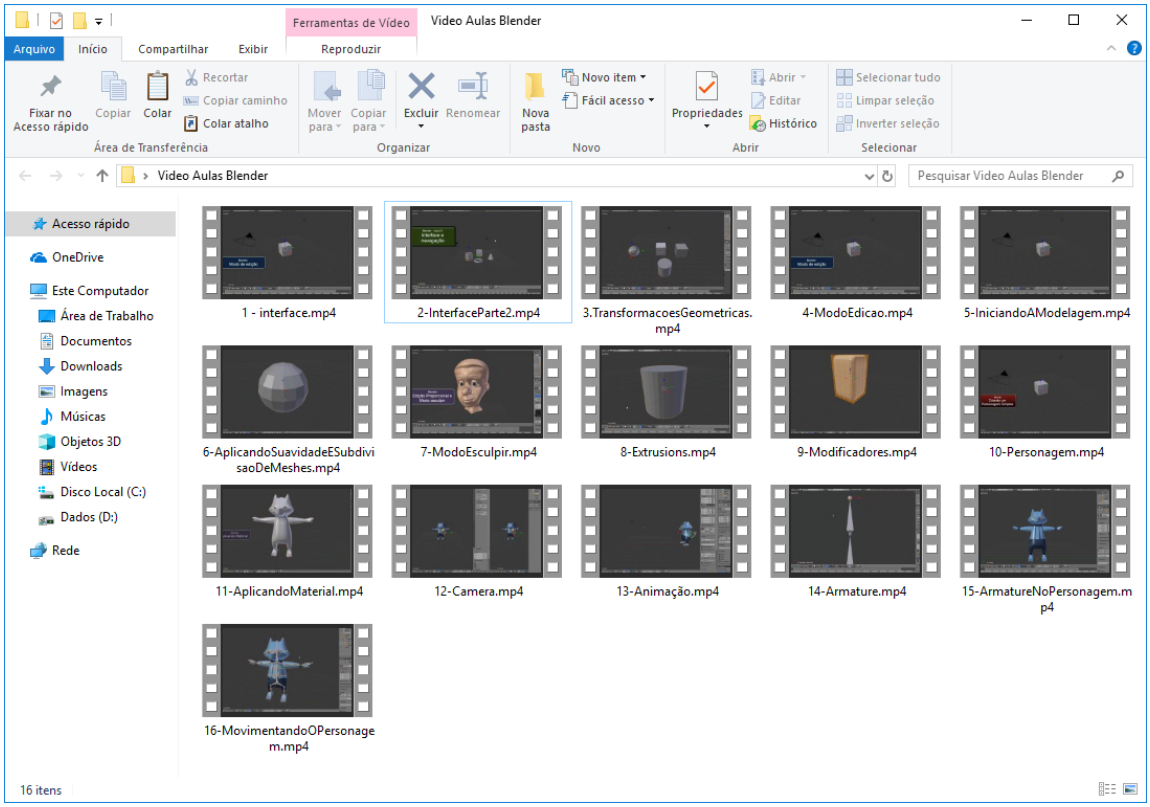


Figura 23 - Lista de vídeo aulas por tema

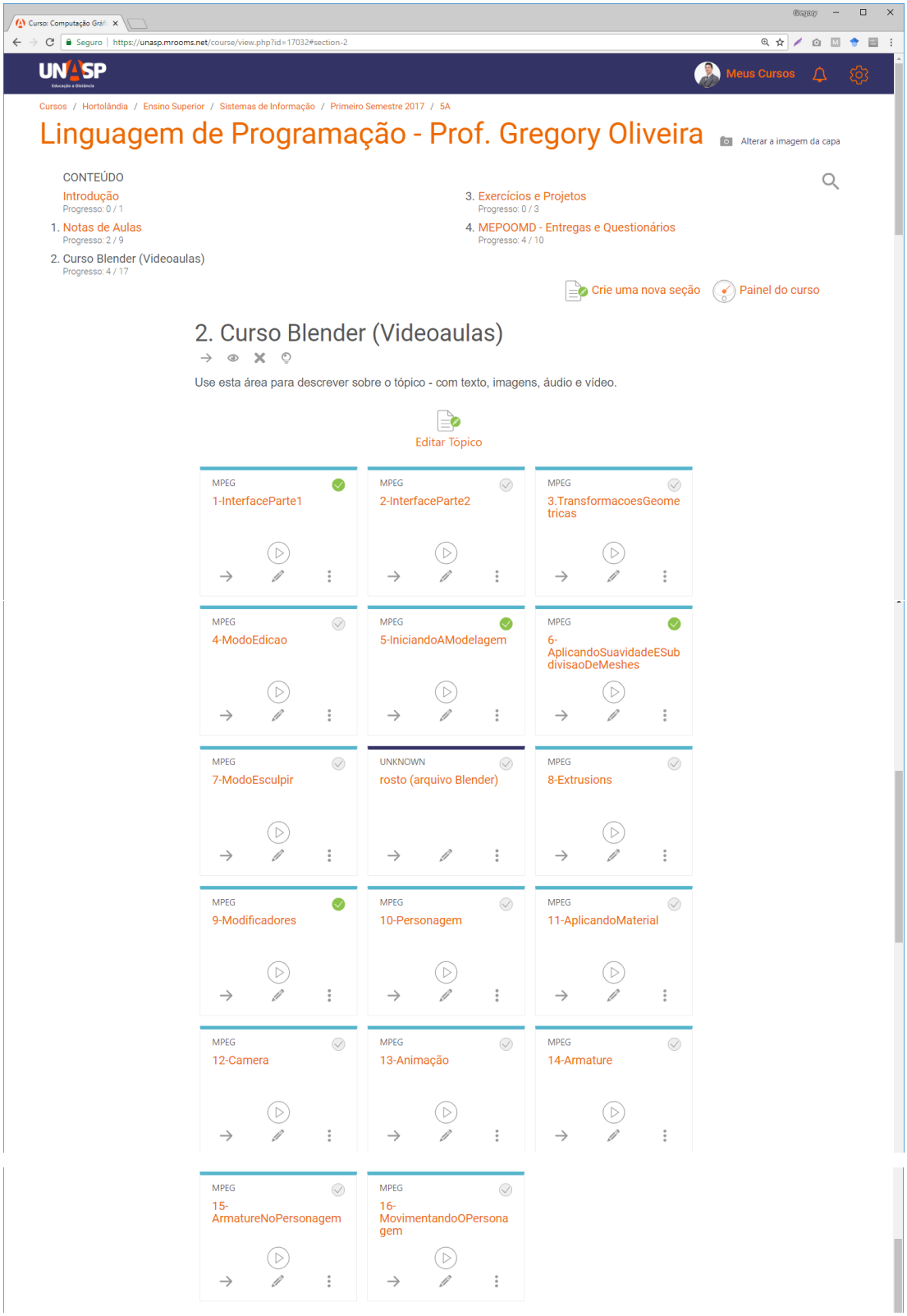


Figura 24 - Lista de vídeo aulas para a utilização do software Blender

Foi realizada uma instrução passo a passo sobre cada um dos temas mencionados. A Figura 25 apresenta exemplo de instrução, onde o aluno tinha que diminuir a separação entre os campos e deixar essa esfera em um formato circular, e não mais com aspecto quadrado exibido.

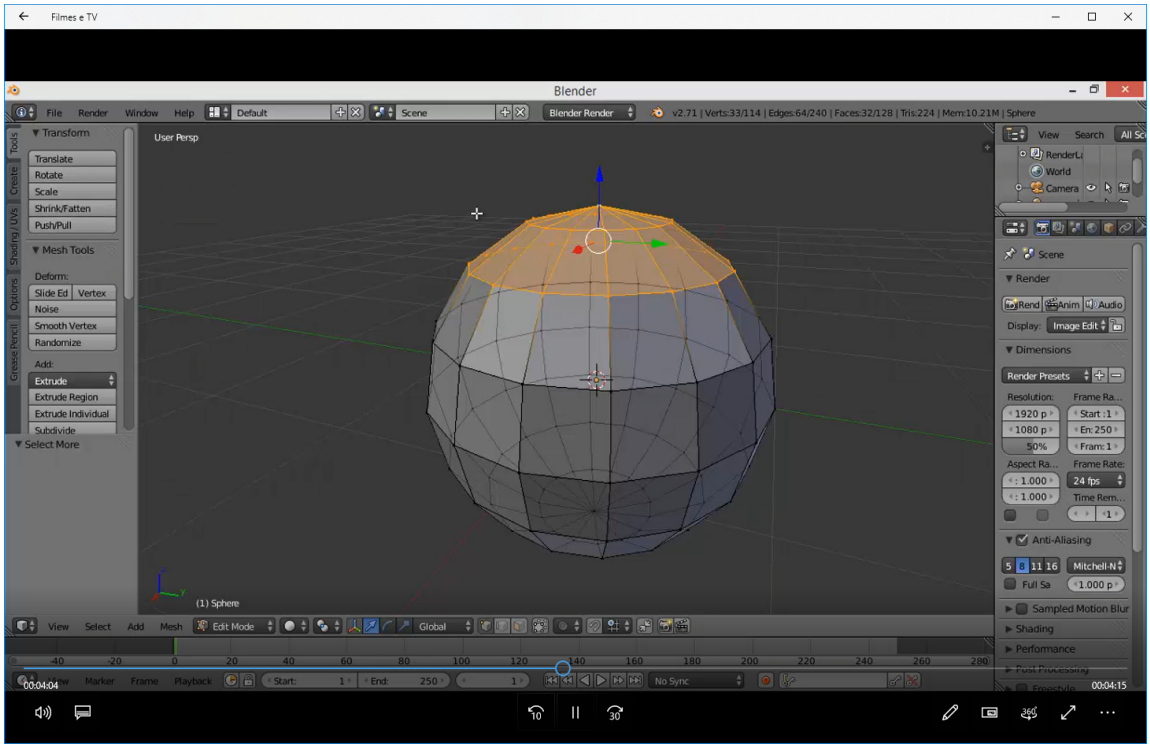


Figura 25 - Vídeo Aula 6, exemplo de vídeo

5.2 Fases da aplicação do MEPOOMD

A seguir serão apresentadas as fases de aplicação do método MEPOOMD, incluindo as fases iniciais executadas pelo professor e a aplicação com os grupos de alunos selecionados nos subgrupos de teste.

5.2.1 Análise

Esta fase inicia-se com a coleta de informações para entender o cenário, as necessidades dos alunos envolvidos e as necessidades do ambiente em que está sendo aplicado o método; para que seja possível o desenvolvimento e aplicação do mesmo, gerando um relatório de diagnóstico e conhecimento. Para tanto, se faz necessário a aplicação de questionários, entrevistas, coletar dados para entender quais são as habilidade e competências que deverão ser desenvolvidas e trabalhadas durante a

aplicação do método. Foram levantadas informações como perfil do aluno, incluindo o desempenho que indica o conteúdo que poderá ser trabalhado durante a aplicação.

Para análise do perfil, foi realizado um questionário online que está no ANEXO II, como detalhado na seção 5.1.2. Os resultados apresentados demostram um baixo nível de programação, uma vez que a minoria se considera com conhecimento suficiente para desenvolvimento utilizando os conceitos de POO.

Devido ao pequeno número de alunos que já programam ou tem algum tipo de afinidade com POO, se faz necessário o entendimento sobre os conceitos que se tem mais dificuldade. Na Figura 26 é possível visualizar quais os conceitos relacionados a POO os alunos têm maior dificuldade em assimilar.

Para tanto, foi realizada a seguinte questão para os alunos: "*Quais os conceitos abaixo relacionados a POO você tem dificuldade de assimilar?*". A Figura 26 sintetiza que a maioria dos alunos tem mais dificuldade em entender construtores, interfaces e *structs* independente do nível que se encontram. Dentre os alunos que responderam que já programam ou desenvolvem, a maioria relatou que possui o conhecimento sobre POO, porém não sabe como aplicar os projetos na sua empresa ou no decorrer do curso nas atividades desenvolvidas. Esta resposta demonstra a falta de conhecimento aprofundado nos conceitos de POO.

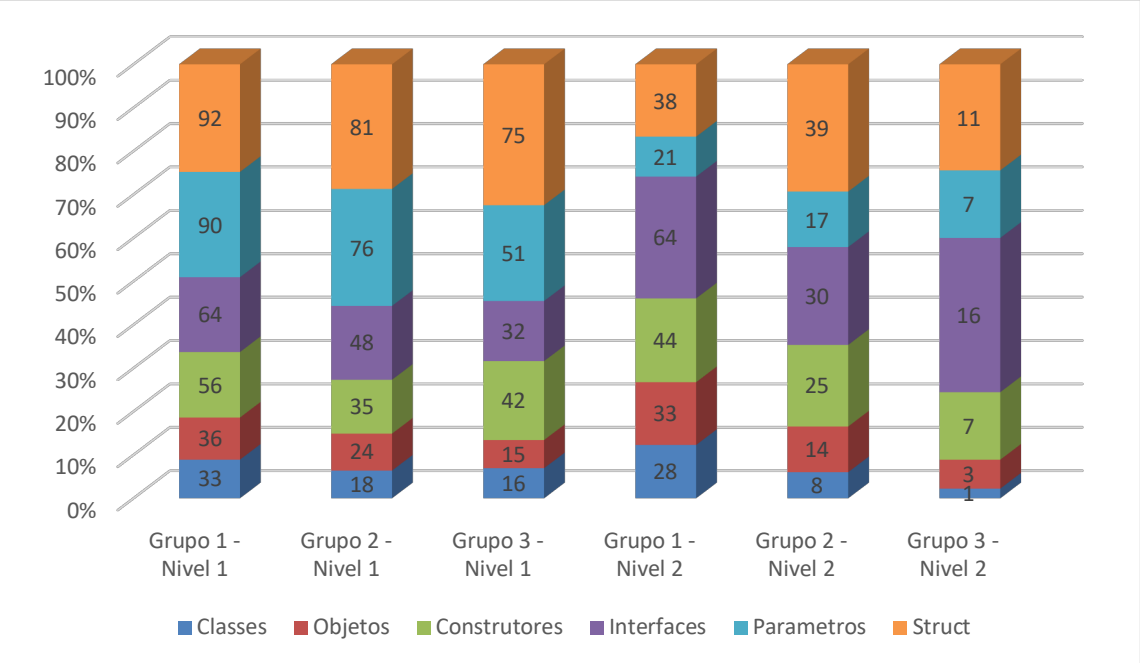


Figura 26 - Alunos que tem dificuldade de entender determinado assunto em POO

Sobre os conceitos de POO foi realizada a seguinte pergunta: "*Você trabalha atua na área de programação e utiliza POO? Se sim quais linguagens utiliza*". Na Figura 27 temos a principal linguagem utilizada pelos alunos selecionados para a pesquisa. A linguagem é Java é a mais utilizada, seguida pela linguagem C#. Devido a este resultado foi utilizado no método a linguagem *java*, pois a maioria dos alunos já tinham afinidade com a mesma.

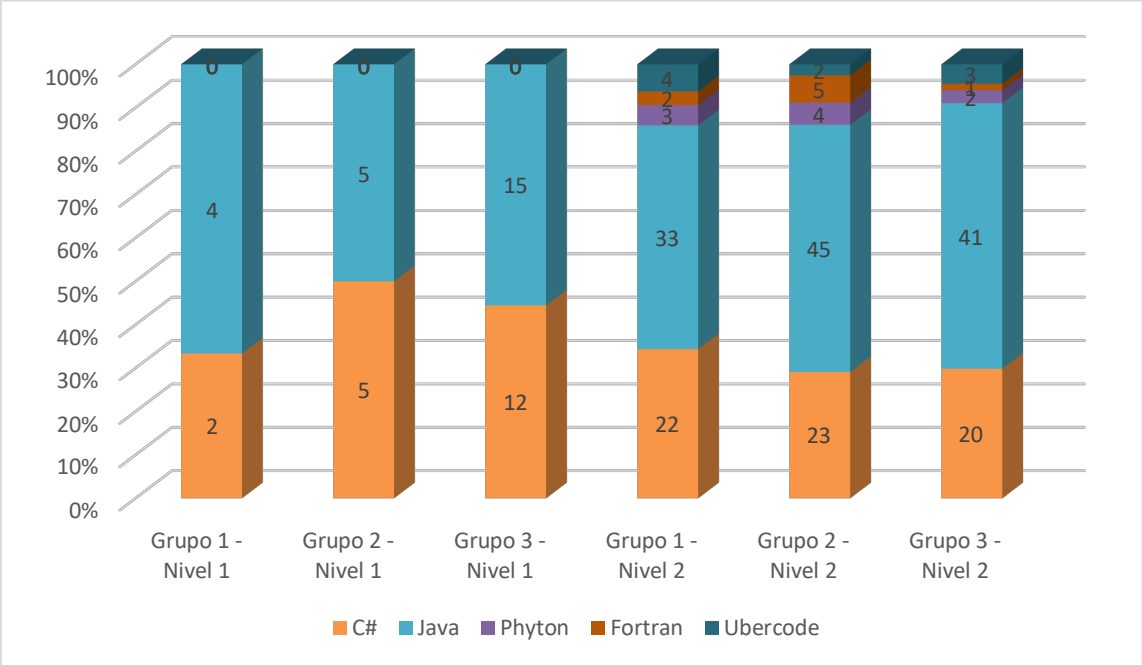


Figura 27 - Linguagens utilizadas

Em seguida, foi realizada a seguinte pergunta: "*Em sua opinião, qual forma para você é mais eficiente para aprender a programar?"* Os participantes da pesquisa definiram que o principal método para aprender e se aprimorar em programação é fazer exercícios práticos, com desafios relacionados ao seu dia a dia. Nesta pergunta (Figura 28), foram utilizadas as seguintes alternativas:

a) Lendo livros de programação.
b) Utilizar vídeo tutoriais da internet
c) Se desafiar e desenvolver algo que te ajude no dia a dia.
d) Contratar um curso específico de programação presencial.
e) Contratar por um curso específico de programação a distância.

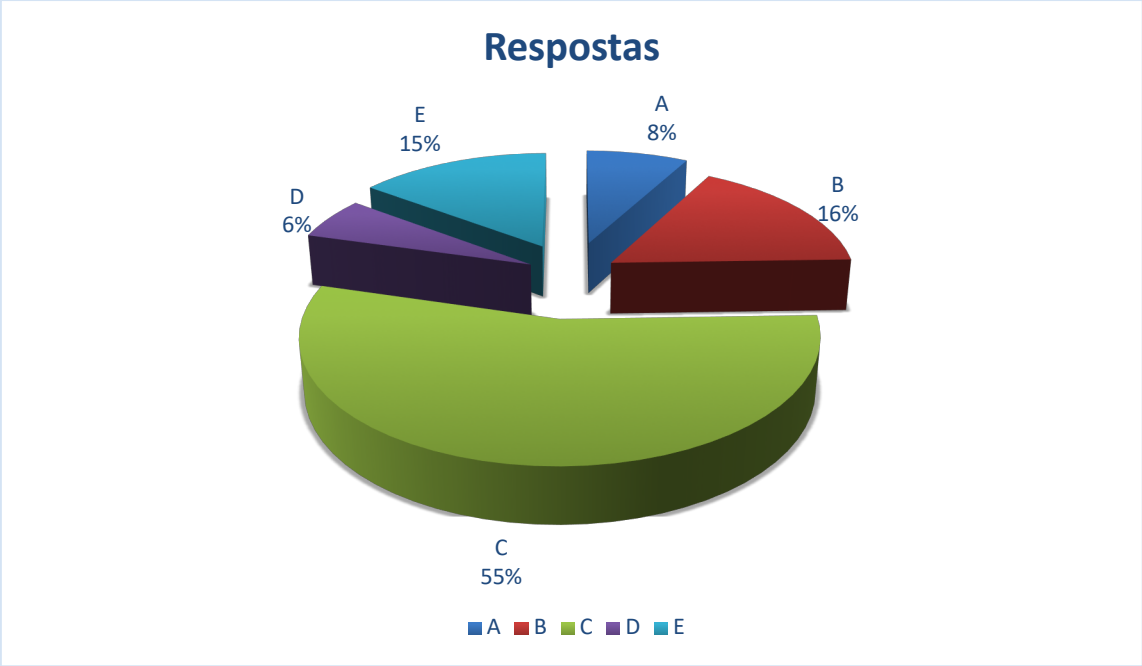


Figura 28 - Método escolhido polos alunos para aprendizagem em programação

Para viabilizar a alternativa "c" (Se desafiar e desenvolver algo que te ajude no dia a dia) é necessário pensar em projetos que consigam de alguma forma viabilizar uma ideia de prática, que seja conhecido por todos. Por exemplo, um dos estudos de caso foi o desenvolvimento de um sistema para controle de vendas (o que já é de domínio comum), onde os alunos precisam utilizar conceitos de POO.

Na etapa de Análise também é necessário realizar o diagnóstico da situação em que se encontra o ambiente e pessoas envolvidas, tais como alunos e professores. Abaixo está listada a definição e estruturação deste mapeamento:

- **Público alvo:**
  - Professor e pesquisador: responsável pela organização e aplicação dos métodos, assim como a análise dos resultados ao final.
  - Professor: Atua como agente responsável pela aplicação da prática. Acompanha os alunos durante os processos e aplica as avaliações.
  - Alunos: São os responsáveis pela apresentação de soluções com estruturas de POO.
- **Cenário atual em relação ao ambiente:**
  - Ambiente de desenvolvimento em linguagem Java.
  - Ambiente para criação de objetos 3D.

- Computador do laboratório com os softwares devidamente instalados para uso do publico alvo.

- **Conceitos que serão trabalhados:**
  - Programação orientada a objetos, com seus principais conceitos: classes, objetos, herança, polimorfismo e interface.
  - Ambiente 3D: interface, transformações geométricas, modificadores e animação.
- **Técnicas utilizadas:**
  - Serão utilizadas as formas de ensino expositivas e prática.
- **Impactos e resultados:**
  - Os resultados e impactos serão avaliados no fim da execução da tarefa ou em avaliações durante a aplicação do método. Estes resultados são utilizados para reconhecer se é necessário recomeçar ou fazer ajustes.

5.2.2 Design

Nesta fase, foi utilizada a cebola semiótica; para, assim, avaliar e entender quem são os *stakeholders* e suas necessidades nas camadas informal, formal e técnica. A Tabela 15 lista os níveis da cebola semiótica, os domínios e os responsáveis.

Tabela 15 - Descrição dos domínios em cada nível da cebola semiótica

Nível	Domínio	Responsável
Informal	• Ensinar conceitos de POO. • Utilizar o ambiente 3D para apresentação de conceitos. • Avaliar o desempenho dos alunos.	Professor
	• Aplicar conceitos de POO no desenvolvimento de soluções. • Ser capaz de criar uma forma em ambiente 3D.	Aluno
Formal	• Especificação e Desenvolvimento da prática. • Apresentação formal e aplicação da pratica • Definição dos conceitos de desenvolvimento de POO utilizados na prática. • Delimitação dos objetivos.	Professor
	• Compreensão estruturada do problema que será	Aluno

	aplicado. • Apresentação de soluções lógicas para as práticas aplicadas. • Aplicação das soluções.	
Técnico	• Ensinar a criação de objetos 3D na ferramenta. • Estruturar uma rotina de avaliação automática. • Analise dos resultados da implementação.	Professor
	• POO na linguagem. • Operação do ambiente 3D. • Compreensão da técnica dos recursos disponíveis em linguagem de POO.	Aluno

Por meio do framework semiótico (apresentado na seção 2.3) foi criado um quadro, como ilustra a Tabela 16, ele apresenta um resumo das ações para satisfazer as questões do framework nos diversos níveis. Estas ações ajudam na identificação dos componentes que auxiliam na solução da prática.

Tabela 16 - Quadro de analise baseado no framework semiótico

Nível	Ações em cada nível
Social	• Proporcionar ao aluno contato com novas tecnologias. • Desenvolver capacidade de resolução de problemas. • Ampliar a forma de análise crítica. • Capacitar para discernir soluções e aplicações tecnológicas. • Possibilitar uma visão tecnológica diversificada.
Pragmático	• Utilizar o ambiente 3D como forma de aprendizado. • Correlacionar conceitos de POO com formas 3D. • Apresentar e utilizar novas ferramentas. • Criar tarefas com a utilização de POO para solução de problemas.
Semântico	• Compreender como estruturar uma solução OO que utilize a forma 3D • Compreender as respostas, podendo definir a solução por meio da mesma. • Proporcionar um nível de abstração, na qual possa compreender os significados dos conceitos de POO. • Promover meios para o aluno ser capaz de interpretar os problemas relativos às respostas provenientes dos recursos do ambiente 3D.

Sintático	• Definir as estruturas dos conceitos de POO utilizadas na práticas. • Selecionar quais são os recursos do ambiente 3D são condizentes com as estrutura sintáticas da POO. • Relacionar a estrutura sintática da POO com o ambiente 3D.
Empírico	• Analisar se os recursos presentes no ambiente 3D condizem com o que foi programado.
Físico	• Compreender a relação da POO com efeitos e representação físicas. • Identificar em cada solução encontrada qual foi o recurso de POO alocado para fornecer respostas ao sistema.

A descrição do domínio também foi realizada utilizando AS para um refinamento e formalismo. Para a primeira fase da AS foi elaborado um enunciado para descrição da prática utilizando ambiente 3D, o enunciado utilizado foi o seguinte: “Criar um ambiente participativo, motivador e colaborativo para o ensino dos conceitos de POO, utilizando ambiente 3D através de meio sistemático, com o objetivo de auxiliar no desenvolvimento da aplicação, avaliação e nas práticas executadas pelos alunos”.

Este enunciado não exibe muitos detalhes do cenário em si, mas requisita a presença de entidades que possam solucionar o problema que foi apresentado. É possível notar que a presença dos agentes pode assumir papeis de professor e aluno. O anunciado reforça a presença de um agente que tenha papeis, onde no contexto ontológico, torna-se um tipo de agente pessoa.

5.2.3 Desenvolvimento

Esta fase inclui a elaboração de materiais e produtos utilizados nas atividades de ensino. As normas são usadas para representar o comportamento esperado dos alunos e professores durante as atividades de aprendizagem. Esta fase, envolve a definição das estratégias de ensino, dos recursos didáticos, das ferramentas e tecnologias, das modalidades de avaliação, da preparação dos docentes que deverão atuar no processo das atividades do aprendiz, além da finalização e da análise da coerência do que foi concebido nas fases anteriores.

Esta fase inicia-se com a definição dos quadros de normas comportamentais, ou seja, a elaboração dos quadros comportamentais. Estes quadros detalham no nível pragmático quais são os comportamentos esperados dos alunos perante o problema apresentado. Por meio desse recurso é possível definir as regras para obter uma solução esperada, com recursos do ambiente 3D no ensino de POO. A utilização das normas comportamentais durante a aplicação do método se faz significativa ao demonstrar indicativos de avaliação do desempenho de cada aluno durante a aplicação do MEPOOMD.

A Figura 29 apresenta um diagrama de classe que foi apresentado para os alunos durante uma das práticas executadas durante o método. Nesta imagem podemos identificar algumas tarefas que o aluno deve desenvolver para que este modelo se torne aplicável e utilizável dentro dos conceitos de POO. A seguir são apresentados os quadros referentes as normas criadas para aplicação do método.

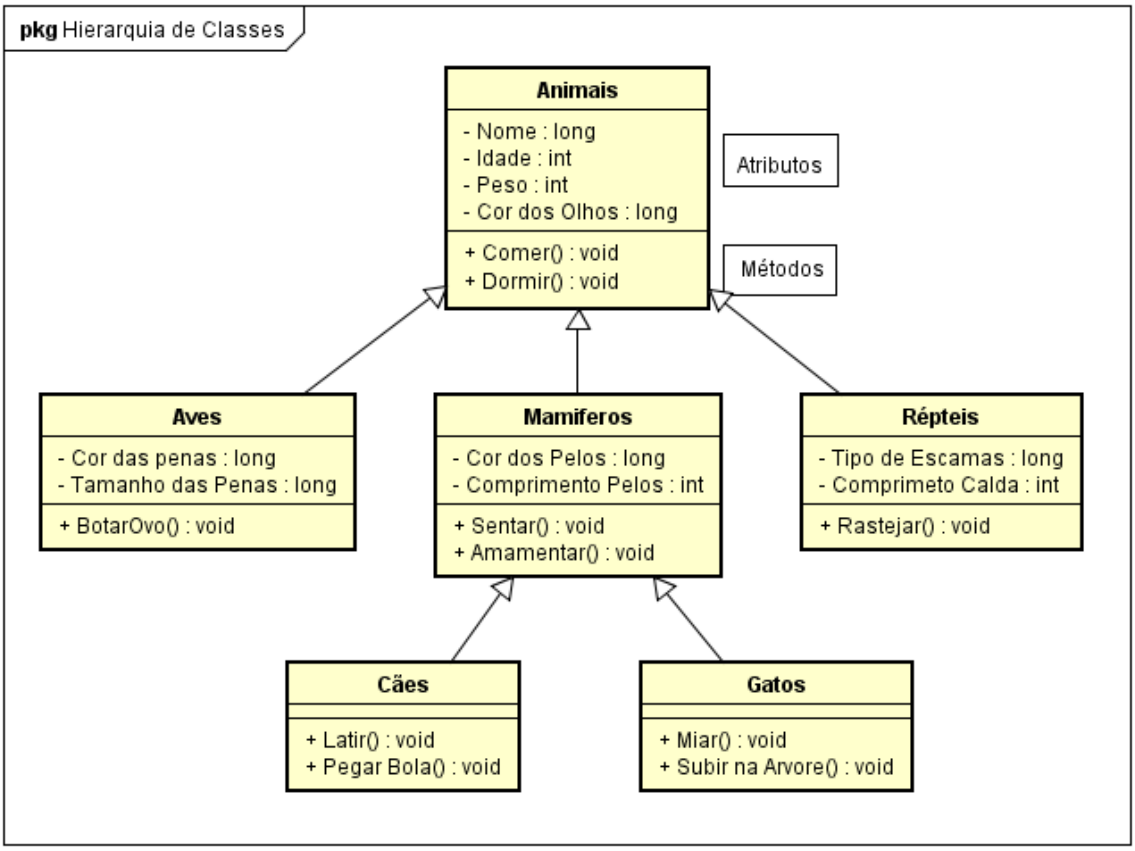


Figura 29 - Exemplo de Diagrama de classes utilizado na aplicação

A norma "Criar classe" (Tabela 17), apresenta a solução necessária para que o aluno crie uma classe no ambiente da IDE Java. Nele o aluno precisa aplicar os conceitos relacionados ao entendimento sobre os relacionamentos em POO e utilizar os

recursos expostos pelo agente professor para utilização do ambiente de desenvolvimento.

Tabela 17 - Quadro de norma "Criar classe"

Norma	***Criar classe***
Agentes	Aluno
Affordances	Ambiente 3D (*Affordance* da sociedade); Programação em JAVA (*Affordance* aluno com ambiente 3D); desenvolver classe com suas características (*Affordance* aluno por meio da programação visualizando objetos 3D)
Detalhamento da norma	**Sempre** que o aluno identificar no diagrama de classes um retângulo com o nome da classe; **se** é parte do diagrama criar no ambiente da IDE**; então** o aluno é; obrigado a; **utilizar** a estrutura de criação de classes disponível na IDE.

A norma “Criar atributos” (Tabela 18) detalha a solução para que o aluno consiga criar os atributos referentes a classe dentro do diagrama de classes. Na execução da prática o aluno deve identificar dentro do retângulo onde é encontrada a classe os atributos da mesma e criar dentro da classe os atributos listados dentro do diagrama. Era esperado que o aluno utilizasse os conceitos de objetos e atributos para a criação do mesmo dentro da classe.

Tabela 18 - Quadro de norma "Criar atributos"

Norma	***Criar atributos***
Agentes	Aluno
Affordances	Ambiente 3D (*Affordance* da sociedade); Programação em JAVA (*Affordance* aluno com ambiente 3D); criação dos atributos referente a classe (*Affordance* aluno por meio da programação visualizando objetos 3D)
Detalhamento da norma	**Sempre** que o aluno identificar no diagrama de classes um retângulo com o nome da classe e dentro uma lista de atributos referente a mesma; **se** é parte do diagrama criar no ambiente da IDE**; então** o aluno é; obrigado a; **utilizar** a estrutura de criação de classes disponível na IDE.

A norma “Criar métodos” (Tabela 19) detalha a solução para que o aluno consiga criar os métodos referentes a classe dentro do diagrama de classes. Na execução da prática o aluno deve identificar dentro do retângulo onde é encontrada a classe os métodos da mesma e criar dentro da classe os métodos listados dentro do diagrama. Era esperado que o aluno utilizasse os conceitos de objetos e métodos para a criação do mesmo dentro da classe.

Tabela 19 - Quadro de norma "Criar métodos"

Norma	***Criar métodos***
Agentes	Aluno
Affordances	Ambiente 3D (*Affordance* da sociedade); Programação em JAVA (*Affordance* aluno com ambiente 3D); criação dos métodos referente a classe (*Affordance* aluno por meio da programação visualizando objetos 3D)
Detalhamento da norma	**Sempre** que o aluno identificar no diagrama de classes um retângulo com o nome da classe e dentro uma lista de métodos referente a mesma; **se** é parte do diagrama criar no ambiente da IDE**; então** o aluno é; obrigado a; **utilizar** a estrutura de criação de métodos disponível na IDE.

A norma "Criar herança" (Tabela 20), o aluno deve ser capaz de identificar no diagrama de classes, retângulos representado classes, e entender quando uma referência demostra herança entre as mesmas. Sempre que encontrar é necessário que vá até a classe que está referenciando a outra e adicionar um trecho de código que representa que a mesma está herdando os dados da classe de nível superior. Para solução é esperado que o aluno utilize os conceitos de herança apresentados pelo agente professor.

Tabela 20 - Quadro de norma "Criar herança"

Norma	***Criar herança***
Agentes	Aluno
Affordances	Ambiente 3D (*Affordance* da sociedade); Programação em JAVA (*Affordance* aluno com ambiente 3D); criação da herança entre as classes (*Affordance* aluno por meio da programação visualizando objetos 3D)
Detalhamento da norma	**Sempre** que o aluno identificar no diagrama de classes um retângulo referenciando uma classe a um nível superior**; então** o aluno é; obrigado a; **utilizar** a estrutura de herança disponível na IDE.

Nesta etapa, também foram definidas as questões para avaliar a aplicação das práticas e tarefas relacionando os aspectos cognitivos, perceptuais, denotativos e avaliativos. Com a elaboração do quadro de respostas, para cada aspecto foi criado uma questão que pudesse gerar respostas dissertativas que expressassem as impressões dos alunos em relação aos aspectos da aplicação. As questões estão listadas abaixo:

1. Houve dificuldades na compreensão da tarefa?
2. O objetivo da tarefa foi facilmente percebido?
3. Teve alguma dificuldade em perceber os efeitos dos conceitos de POO?

4. Houve dificuldades em entender as respostas da IDE para as execuções realizadas?
5. Os resultados obtidos foram satisfatórios para seu aprendizado?
6. A prática ajudou na aquisição de novos conceitos de POO ou ajudou a concretizar ainda mais seu conhecimento?
7. Os conceitos de POO foram escolhidos adequadamente de acordo com o tipo de tarefa?

Para que não houvesse dupla interpretação ou alguma falha ao responder o questionário, foram utilizados termos simples e comuns aos alunos. O questionário é apresentado no Anexo VI e a síntese de respostas obtidas no capitulo 6.

5.3.4 Implementação

Nesta fase foi realizada a definição do ambiente e os recursos que o computador deve possuir para que seja possível a execução das tarefas, por exemplo o software NetBeans para o desenvolvimento e Blender para o ambiente 3D. Para o desenvolvimento da prática são empregados todos os elementos desenvolvidos nas fases anteriores do MEPOOMD. Na elaboração de uma tarefa é obrigatório a utilização de três elementos: (1) enunciado, (2) regras e (3) metas a serem concluídas

Para que uma prática fosse executada com sucesso, foi necessário que o nível 1 tivesse uma preparação desde a parte introdutória de POO até o mais avançado, já o nível 2 iniciou com aulas de nível intermediário de POO. As aulas de POO ocorreram de forma expositiva, por meio de apresentações, onde estas apresentações eram utilizadas passo a passo. Enquanto eram apresentados conceitos novos de POO, o professor demostrava na prática utilizando o Netbeans. A Figura 30 ilustra alguns exemplos de aulas e temas abordados.

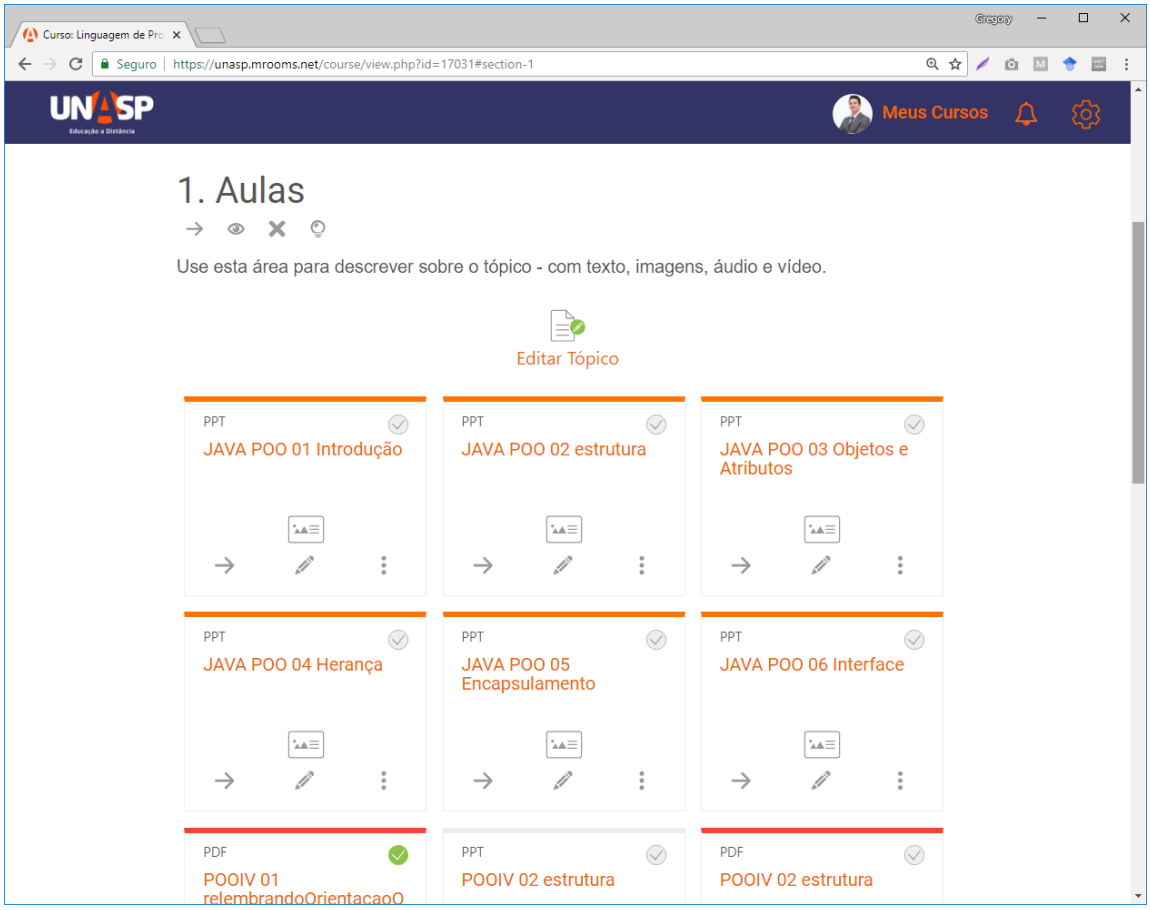


Figura 30 - Aulas de POO disponíveis para download na área virtual do aluno

No que diz respeito ao ambiente 3D, foi criado um espaço para que o aluno postasse criações em ambientes virtuais. Também foi realizada uma preparação para utilização do Blender, como detalhado na seção 5.1.3. Para iniciar as aulas com contexto e aprendizagem em POO foram necessárias 2 aulas prévias sobre o ambiente 3D. Após as duas primeiras aulas, os alunos foram requisitados a realizar a criação de uma forma que, realizasse algum movimento. Na Figura 31 é possível visualizar um exemplo de entrega realizada por um aluno do nível 1. Neste nível os alunos ainda não possuem um controle efetivo sobre as ferramentas, muitos ainda estão conhecendo o computador, como visto na análise do grupo na seção 5.1.2.

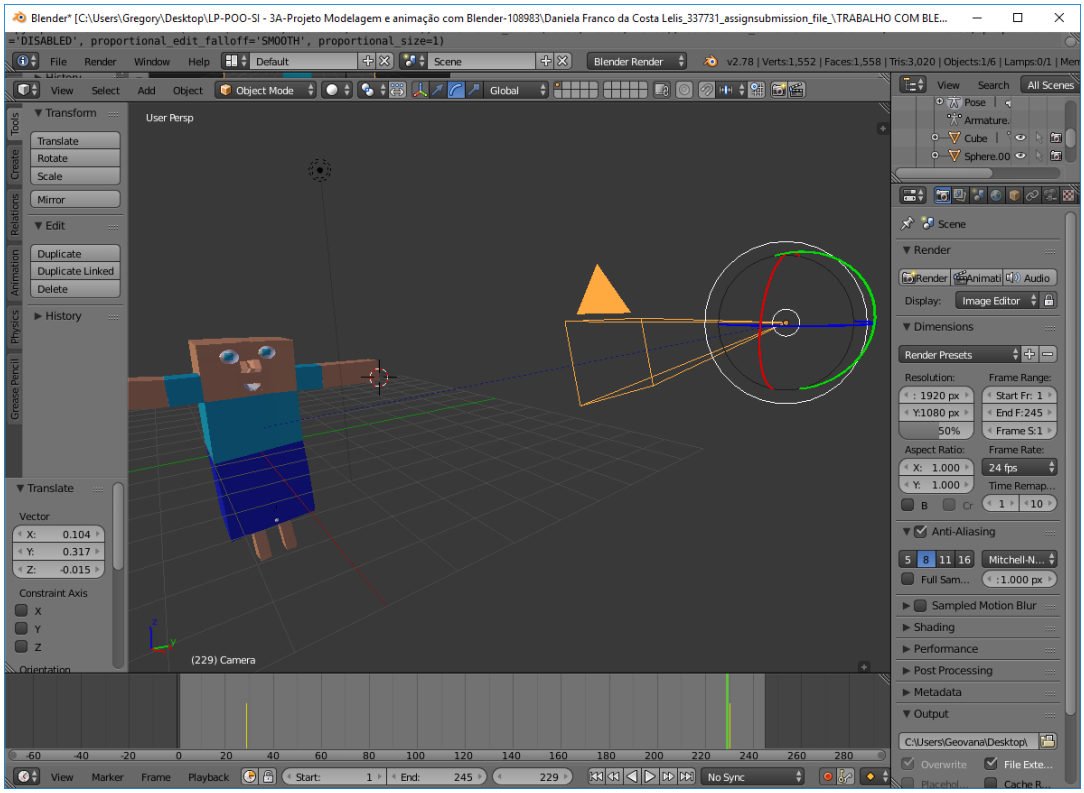


Figura 31 - Entrega de atividade aluno do nível 1

A Figura 32 mostra a entrega de um aluno do nível 2, onde a idade e experiência é mais elevada, com objetos mais concisos e reais.

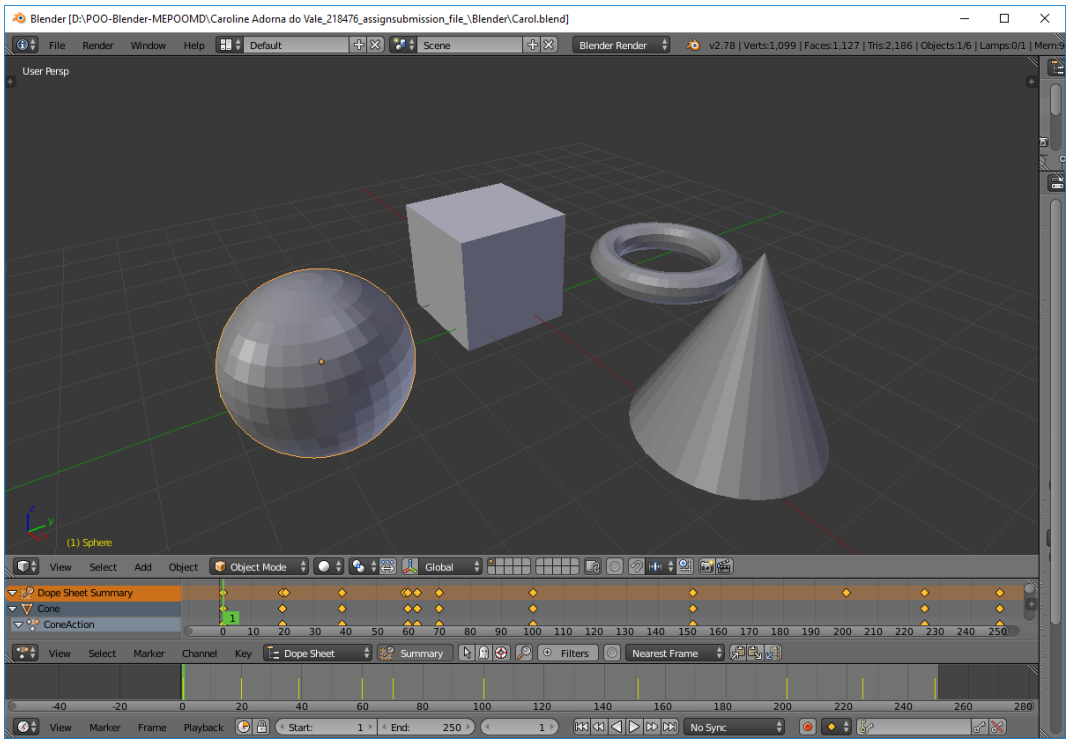


Figura 32 - Entrega de atividade aluno do nível 2

Esta tarefa foi entregue por meio do ambiente virtual disponível pela instituição. Essa forma de entrega se manteve durante as outras atividades desenvolvidas na aplicação deste método. A Figura 21 exibe um exemplo de tarefa entregue pelos alunos até o dia 22 de maio de 2017 as 12:00.

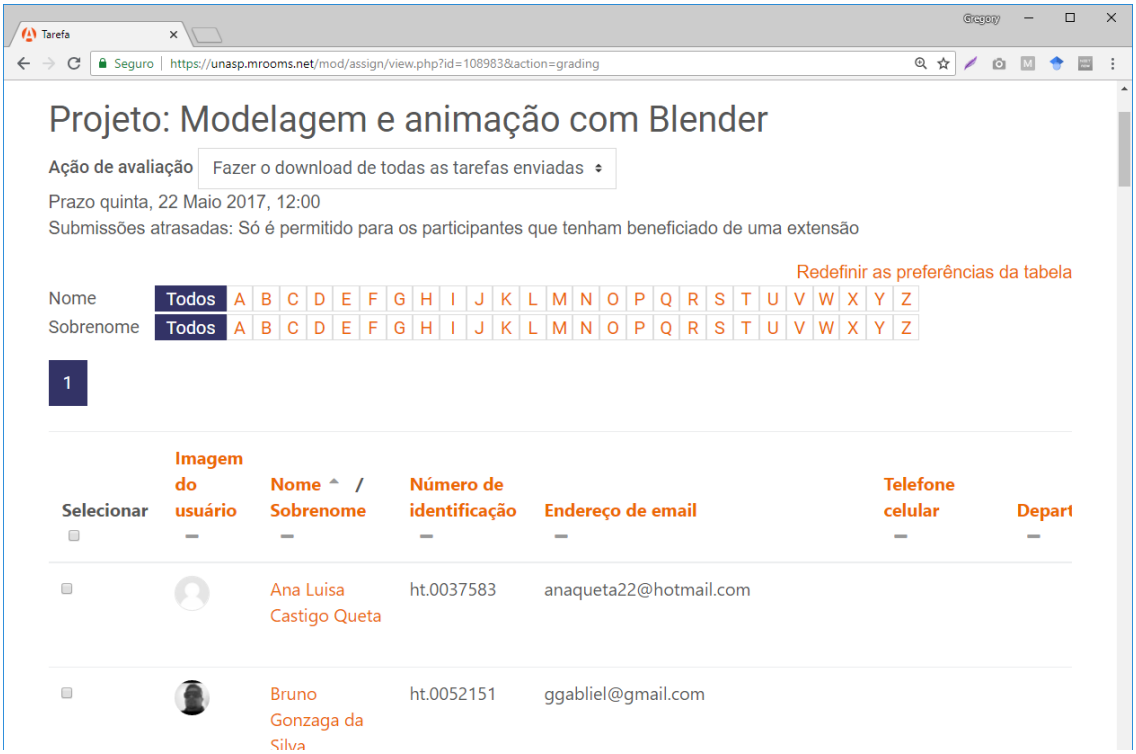


Figura 33 - Exemplo de espaço para entrega de atividade

Na atividade 1(Anexo I), os alunos deviam utilizar coisas do mundo real para a compreensão e aplicação da OO. Desta forma, foi utilizado o objeto carro como parte da primeira atividade. A atividade 1 foi aplicada para o aluno utilizando os métodos descritos na seção 4.2.4., com os três elementos: enunciado, regras e metas a serem concluídas. A Figura 34 contém um diagrama de classe, com a classe carro. O carro contém os seguintes atributos: marca, cor, dono, comprimento, largura e Ano; bem como também possui métodos: acelerar; frear; virar à direita; virar a esquerda e acender o farol.

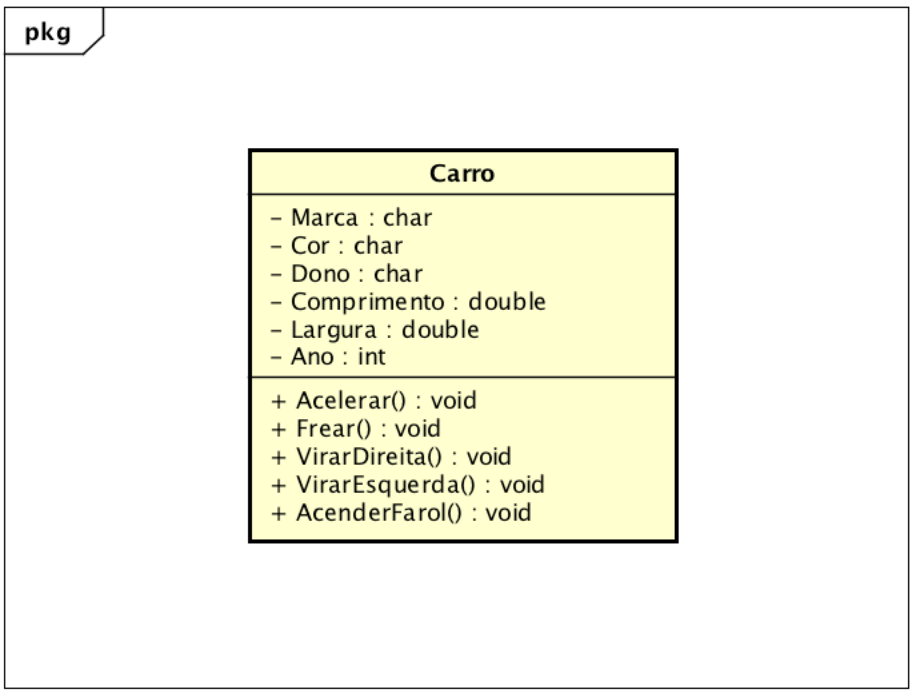


Figura 34 - Diagrama de classe com objeto carro

Devido aos elevados níveis de abstração para a utilização de POO na criação do projeto como discutido nos capítulos anteriores. Foi requisitado para o aluno antes da criação da classe em Java dentro do ambiente do Netbeans, que fosse criado no ambiente 3D o objeto que estava sendo representado, incluindo atributos e métodos que representam elementos no mundo real para ele. Após a definição de como as ferramentas seriam utilizadas, foram descritos os enunciados para a realização das práticas. As descrições são apresentadas a seguir:

- **Enunciado da prática:** Para o diagrama de classe apresentado, o enunciado proposto foi "O objeto carro exibido no diagrama de classe deve ser representado de forma mais realista possível ao mundo real, onde deve ser possível ter os atributos e métodos relatados dentro do diagrama, após a criação deste modelo de forma tridimensional utilizando o Blender, criar a classe utilizando conceitos aprendidos nas aulas de POO, criar a classe carro com seus atributos e métodos como demostrado na figura".
- **Regras para solução:** As descrições destas regras refletem diretamente no comportamento esperado por parte do aluno, portanto foram apresentadas além da descrição detalhada de regras e limitações das

ferramentas, as normas comportamentais a serem seguidas durante a realização da prática.

- **Objetivos a serem alcançados:** Os objetivos foram apresentados aos alunos de forma geral, definindo o que cada elemento dentro do diagrama descreve e o que deveria ser realizado. Como objetivo principal a criação do objeto tridimensional e a classe em Java representando o objeto carro.

Na Figura 36 é possível visualizar a visão que um aluno do nível 1 tem sobre a classe apresentada na Figura 34. É possível identificar que o aluno foi direto ao que a classe descreve como método para este objeto, e deixou de lado a parte visual, uma vez que o carro não exibe janelas e outras formas parecidas com um carro que temos hoje em dia.

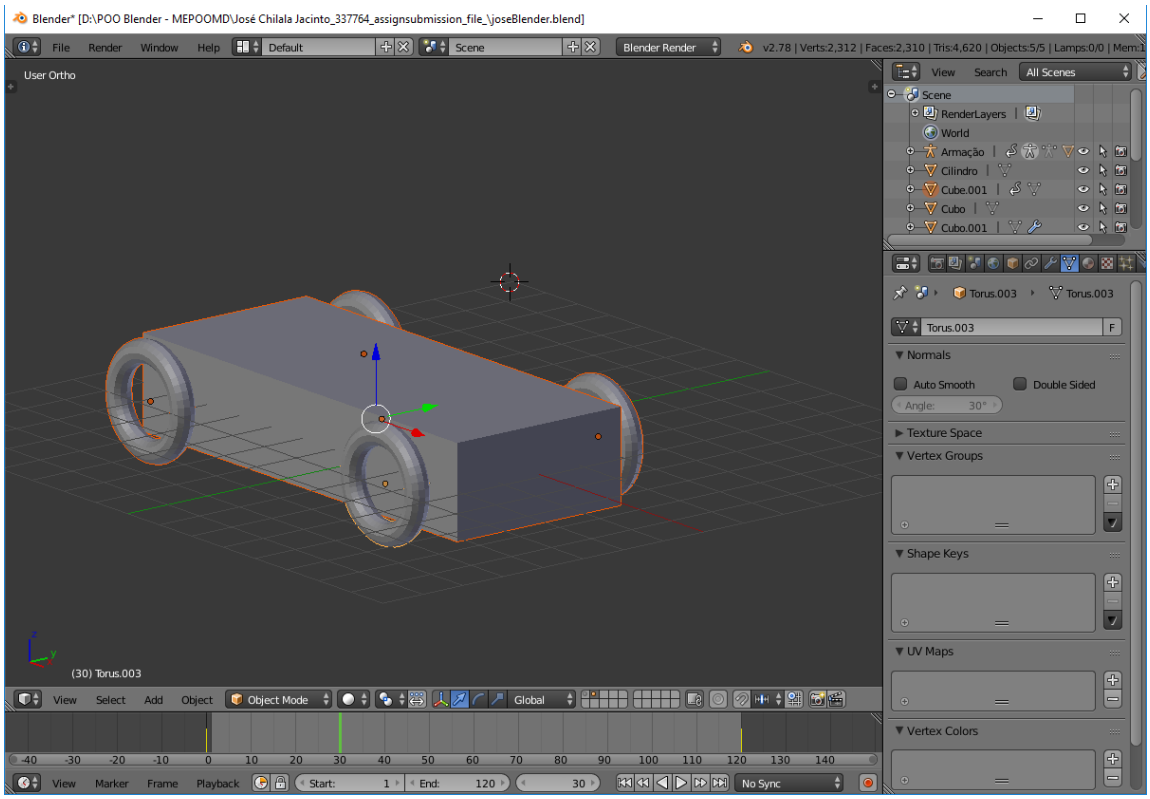


Figura 35 - Objeto 3D feito por aluno do nível 1, baseado na classe carro.

Na Figura 36 é exibido um carro criado por um aluno do nível 2. O aluno o representa um carro com janelas, cor, e alguns detalhes que não são exibidos por um aluno de nível 1.

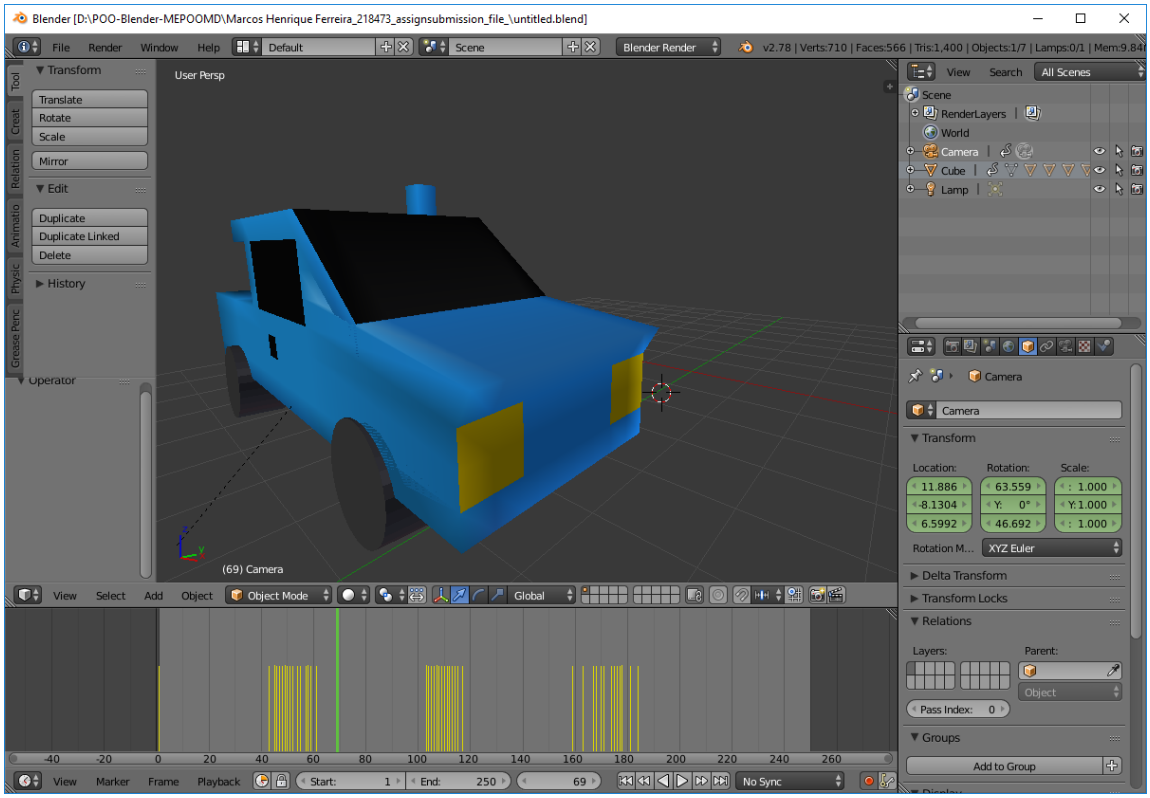


Figura 36 - Objeto 3D feito por aluno do nível 2, baseado na classe carro.

Independentemente do nível apresentado referente ao desenho, os dois níveis cumpriram com a exibição de um objeto que representasse o que a classe foi passada como referência no diagrama de classe.

Após o desenvolvimento e criação de uma forma que representasse o que foi passado no diagrama de classe, o aluno deve desenvolver uma nova classe dentro de um projeto em JAVA, na IDE Netbeans. Esse projeto deve conter os atributos e os métodos da classe carro. A Figura 37 exibe um exemplo de uma entrega realizada por aluno.

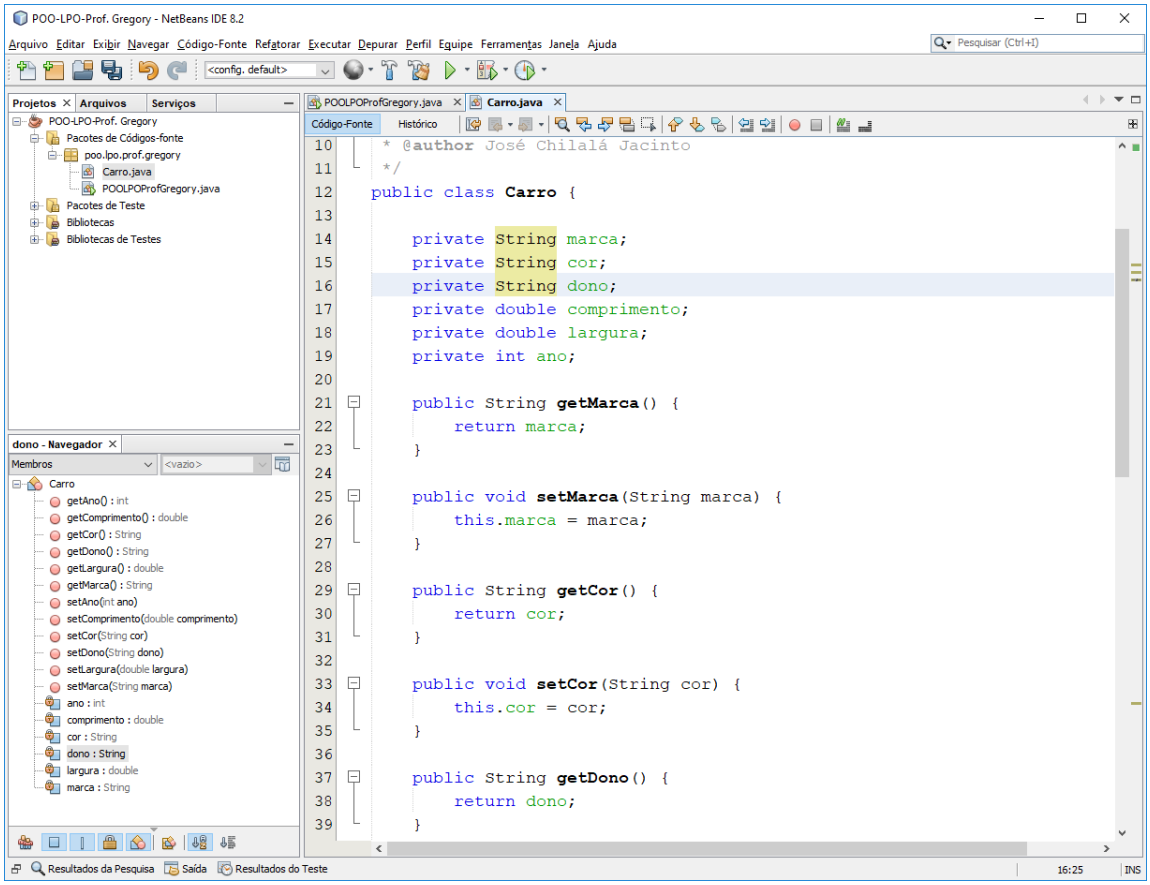


Figura 37 - Classe carro desenvolvida em Java na IDE

5.3.5 Avaliação

Como descrito na etapa “Avaliação de Resultados” do MEPOOMD, foi criado um quadro de avaliação para acompanhamento da execução e desempenho perante as normas comportamentais. O quadro apresenta uma avaliação com os termos “não cumpriu”, “cumpriu parcialmente” e “cumpriu plenamente”. Os quadros de respostas, referente à etapa “Elaboração”, foram aplicados por meio de um questionário online, juntamente com um questionário para avaliação quantitativa, utilizando o padrão de respostas apresentado no Anexo V. O resultado e a síntese dos questionários são apresentados no Capítulo 6.

O quadro de normas (Tabela 10) foi desenvolvido para a realização da avaliação dos resultados. Elas foram utilizadas como base para entender o que e como deve ser avaliado o código desenvolvido. Após a utilização do quadro de normas é necessário a tabela de respostas, onde o docente define o resultado de cada meta do quadro de normas

por meio de três níveis: (1) completo, (2) parcialmente completo, e (3) incompleto. Estes valores são somados para gerar a nota final em relação a atividade desenvolvida pelo aluno.

Os resultados de avaliação deste método estão descritos no capitulo 6, tanto de e forma qualitativa quanto quantitativa.

6. ANÁLISE DE RESULTADOS

Esse capítulo apresenta a análise dos resultados obtidos por meio dos questionários (Anexos II, III, IV e V) e avaliações realizadas pelo professor. As discussões sobre os resultados dos questionários serão divididas pelo caráter qualitativo e quantitativo. Na Seção 6.1 temos a análise das respostas referentes à etapa "Elaboração dos Quadros de Respostas", por meio de discussões sobre as respostas dissertativas dos grupos participantes do experimento. A análise dos resultados quantitativos será apresentada na Seção 6.2. A Seção 6.3 discute os resultados da avaliação e do acompanhamento do professor sobre as normas comportamentais.

Conforme apresentado no capítulo 5, os subgrupos de controle e teste possuem o mesmo perfil. Os discentes pertencentes ao Nível 1 são alunos do primeiro, segundo e terceiro anos do curso técnico. O Nível 2 são alunos do 3°, 5° e 7° Semestres de graduação em Sistemas de Informação.

É importante relembrar que todos os subgrupos de testes, tanto dos grupos 1, 2 e 3 do nível 1 e os grupos 1, 2 e 3 do nível 2, efetuaram as práticas com o MEPOOMD. Os subgrupos de controle apenas participaram das oficinas e resolveram os problemas práticos normalmente, isso é, sem o MEPOOMD . Com os resultados da experimentação pode ser feita uma comparação direta ente os subgrupos controle e subgrupos de teste, por serem membros pertencentes ao mesmo nível de POO e também por terem sido divididos randomicamente. As comparações entre nível 1 e 2 se dão por diferença entre os perfis, iniciantes e intermediários.

6.1 Análise Qualitativa

A primeira análise baseia-se nas respostas obtidas no Questionário 6 (Anexo VI), que organiza as questões relacionadas às normas perceptuais, cognitivas, avaliativas e denotativas. Esse questionário foi aplicado aos três grupos de cada nível, incluindo os subgrupos de Controle e de Teste, para assim termos uma base comparativa relacionada às normas, e assim analisar a influência do método MEPOOMD.

Primeiramente, os pesquisadores classificaram as respostas entre negativas e positivas. Esta análise focou principalmente nas respostas negativas, pois apontam dificuldades e problemas encontrados pelos alunos. Ao analisarmos os subgrupos de

controle por meio dessas respostas é possível perceber de maneira mais clara em quais aspectos (perceptuais, cognitivos, avaliativos ou denotativos) houve maior dificuldades. Para, assim, apontarmos o que pode ser melhorado. As respostas negativas dos grupos 1, 2 e 3 de cada nível apresentam problemas presentes na aplicação da prática com o método MEPOOMD, assim como pontos a serem melhorados e possíveis correções para o modelo.

Após a classificação entre respostas positivas e negativas, os pesquisadores selecionaram de comum acordo as respostas mais significativas dos alunos, uma vez que muitas delas são repetitivas. As respostas selecionadas foram categorizadas por questão e por tipo de norma envolvida, bem como foram considerados cada um dos grupos e subgrupos participantes do experimento.

Com relação às questões 1 (*Houve dificuldades na compreensão da tarefa?*) e 2 (*O objetivo da tarefa foi facilmente percebido?*), os pesquisadores perceberam que os subgrupos de controle apresentaram alguns problemas em relação a identificação dos objetivos da prática e os efeitos de desenvolvimento com POO utilizando o ambiente 3D. Algumas das respostas da questão 2, sobre a dificuldade na compreensão dos objetivos, e na questão 3 (*Teve alguma dificuldade em perceber os efeitos dos conceitos de POO?*) demonstraram isso. Destaca-se aqui uma resposta negativa típica da Questão 2, que demonstra falta de detalhamento sobre os objetivos, evidenciando a necessidade de melhorar a apresentação.

"Não. Foi difícil de entender, pois tinha varias ações a serem feitas e não tinha sido explicado detalhadamente como deveria ser feito".

Na questão 3 foi possível identificar dificuldades para perceber a relação entre os conceitos de POO e o ambiente 3D. Exibindo assim uma deficiência na apresentação de elementos que pudessem demonstrar de forma mais eficiente a relação entre elementos 3D e POO, tal como é destacado na resposta abaixo:

"Tive dificuldade, pois os conceitos em POO são mais complexos que os de criação em ambientes 3D".

Já nas respostas para questão 4 (*Houve dificuldades em entender as respostas da IDE para as execuções realizadas?*), que é relacionada a norma cognitiva, foi possível identificar dificuldades quando os alunos foram questionados sobre a aplicação dos conceitos de POO, tais como:

“Ao utilizar os conceitos de POO surgia sempre uma dificuldade com os conceitos mais avançados”.

“Quanto aos conceitos aprendidos durante o curso de POO, tive algumas dificuldades em entender o conceito. Mas ficou claro que é possível aprimorar outros conceitos que apenas foi visto de forma introdutória”.

Na questão 5 (*Os resultados obtidos foram satisfatórios para seu aprendizado?*), que está relacionada diretamente as normas avaliativas, os subgrupos de controle demonstraram alguns problemas. Alguns ficaram incertos referentes aos conceitos aplicados e ensinados. Entre as questões que apontam o problema estão:

“As vezes a criação feita em ambiente 3D não demostrava o que eu realmente queria em relação ao POO”.

“Em alguns casos percebi uma certa dificuldade...”.

Em relação a questão 6 (*A prática ajudou na aquisição de novos conceitos de POO ou ajudou a concretizar ainda mais seu conhecimento?*) e a questão 7 (*Os conceitos de POO foram escolhidos adequadamente de acordo com o tipo de tarefa?*) Alguns mencionaram que a prática não ajudou necessariamente o aprendizado e sim a parte visual utilizando o ambiente tridimensional.

“Atividades e momentos de aprendizado utilizando o ambiente 3D me proporcionou uma motivação e aprendizado melhor.”

A Tabela 22 apresenta uma síntese das respostas dos grupos em seus respectivos níveis que foram analisadas, é possível notar que em todos os grupos do nível 1, há uma incidência maior de respostas positivas em relação aos subgrupos de controle.

Tabela 21 - Síntese de respostas do questionário 6 no nível 1

Questão	Nível 1											
	G1				G2				G3			
	Teste		Controle		Teste		Controle		Teste		Controle	
	P	N	P	N	P	N	P	N	P	N	P	N
1	22	0	22	1	15	0	17	3	12	1	12	2
2	21	0	22	2	16	1	16	2	11	3	10	3
3	23	1	21	0	15	0	17	3	10	2	11	4
4	21	0	22	2	17	0	17	1	12	1	12	2
5	22	2	20	1	14	1	16	4	11	2	11	3
6	22	1	21	1	15	1	16	3	13	0	13	1
7	23	0	22	0	13	0	17	5	14	0	13	0
% Neg.	2,66%		4,54%		2,5%		20%		10,97%		18,97%	

G1 – Grupo
P – Positivo
N – Negativo

Ao analisar a Tabela 23 é possível notar que tanto maior o nível de conhecimento em POO, menor o percentual de avaliações negativas para os subgrupos de teste (com o método). É notável a diferença entre os subgrupos de teste e controle para os grupos G2 e G3 no nível 2.

Tabela 22 - Síntese de respostas do questionário 6 no nível 2

Questão	Nível 2											
	G1				G2				G3			
	Teste		Controle		Teste		Controle		Teste		Controle	
	P	N	P	N	P	N	P	N	P	N	P	N
1	25	4	23	3	25	1	22	1	24	0	21	3
2	26	3	23	3	25	1	20	3	24	0	22	2
3	25	4	22	4	24	2	19	4	24	0	22	2
4	24	5	23	3	24	2	20	3	23	1	21	3
5	26	3	22	4	23	3	20	3	23	1	20	4
6	27	2	23	3	24	2	20	3	22	2	21	3
7	27	2	25	1	25	1	21	2	22	2	21	3
% Neg.	12,77%		13,04%		7,05%		%13,38		3,07%		13,51%	

G1 – Grupo 1

P – Positivo

N - Negativo

Entretanto, vale ressaltar que as respostas fornecidas pelos alunos muitas vezes não possuem um detalhamento suficiente para a identificação do problema envolvido e as dificuldades de uma maneira clara e precisa. Porém, mesmo com a baixa granularidade nas respostas, é possível notar que o MEPOOMD obteve menos respostas negativas entre alunos mais avançados.

6.2 Análise quantitativa

A análise quantitativa foi feita baseado nos dados extraídos do questionário 4 (Anexo IV), neste questionário foi utilizado a escala *Likert* para compor as opções de resposta e múltipla escolha pelos alunos. Esse questionário permite relacionar o uso de um ambiente 3D e a aceitação deste modelo para o ensino de POO, nos diferentes níveis e grupos de alunos que foram analisados.

As questões do questionário 4, estão listadas abaixo, na ordem em que foram expostas para os alunos:

1. Comparando com a metodologia tradicional de ensino, você considera a metodologia que foi utilizada para o ensino de POO mais eficaz?
2. Comparando com a metodologia de ensino de programação com a qual você já teve contato, você considera a metodologia que foi utilizada para o ensino de POO mais eficaz?
3. Considerando os níveis abaixo, qual nível seria correspondente ao conteúdo que foi exposto a você durante a aplicação e ensino de POO?
4. Você considera estas aulas e o material exposto essências para a sua formação como profissional?
5. Você acha que as aulas apresentadas deveriam ser mais práticas?
6. Você acha que as aulas deveriam ser mais rápidas?
7. Com os conhecimentos expostos em laboratório estou pronto para encarar o mercado de trabalho desenvolvendo utilizando a técnica de Orientação a Objetos
8. Durante as aulas fiquei muito confortável em desenvolver utilizando a técnica de programar orientada a objetos, que tenho interesse de me aprofundar no assunto
9. Aprender POO neste formato de ensino foi útil para que eu possa aprender efetivamente a programar?
10. Durante as aulas expositivas e práticas, ficou claro para mim que podemos diminuir o nível de abstração da POO ao colocar em prática o conceito aprendido
11. Ao realizar as atividades e as provas você conseguiu imaginar de forma rápida soluções computacionais para os problemas envolvidos
12. Ao decorrer das aulas, tarefas e avaliações o desenvolvimento orientado a objeto se mostrou de fácil desenvolvimento e aplicação
13. Você acredita que o desenvolvimento Orientado conseguiu lhe dar uma base estrutural para desenvolver em alto nível
14. Após a experiência vivida durante projeto, é de seu interesse ingressar em alguma certificação profissional em POO?
15. Aprender POO neste formato foi muito interessante e acredito que será possível recordar os conceitos abordados ao programar futuramente

16. Aprender POO é muito abstrato e não consegui entender os conceitos expostos durante o projeto
17. A quantidade de exercícios e aplicações feitas para entender POO utilizando fabricação digital foi tão grande que me deixou entediado
18. Estou ansioso para procurar cursos adicionais que exigem programação.
19. Não gosto nem de imaginar que precisarei ter que aprender (mais) programação.
20. Estou confiante de que vou ter sucesso em cursos adicionais que requerem programação.

A Figura 38 exibe a porcentagem de respostas dos subgrupos de controle do nível 1, ou seja, são alunos que possuem nível baixo de POO, e também não foram expostos a aplicação do MEPOOMD. Ao analisar o gráfico podemos perceber que mesmo os alunos do grupo controle de um nível baixo de POO deram respostas positivas, 35% concordo totalmente e 49% concordo parcialmente.

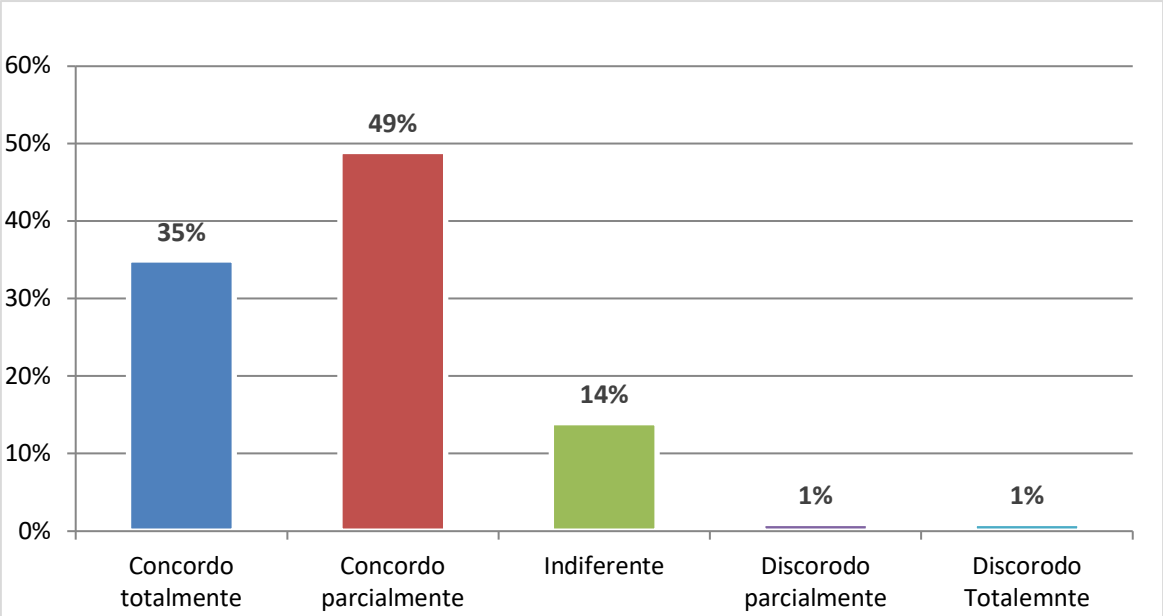


Figura 38 - Porcentagem das respostas dos grupos de controle do nível 1

Na Figura 38 é possível visualizar as respostas do subgrupo de teste do nível 1, ou seja, alunos tem um nível baixo de POO, porem utilizaram o MEPOOMD para aprendizagem de POO. É possível perceber que a aceitação do uso de ambiente 3D para a criação de objetos relacionados a POO foi positiva, 47% concorda totalmente e 38% concorda parcialmente. Neste gráfico também é possível notar o baixo nível de

reprovação em relação a aplicação e utilização do MEPOOMD.

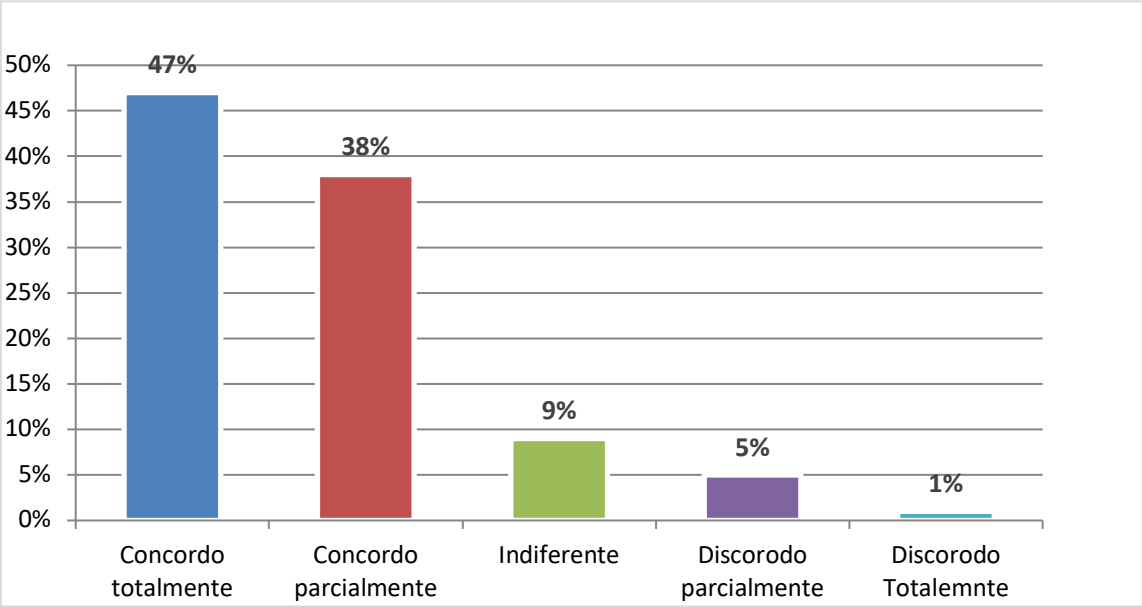


Figura 39 - Gráfico com a porcentagem das respostas dos grupos de teste do nível 1

Se comparado as repostas do grupo de controle que não foi exposto ao método, e o grupo teste que utilizou o método como ferramenta para seu aprendizado relacionado a POO, é possível notar uma melhoria quando aplicado o MEPOOMD.

A Figura 40 exibe a porcentagem de respostas dos grupos de controle do nível 2, são alunos que possuem nível intermediário de POO, e não foram expostos a aplicação do MEPOOMD. Ao analisar o gráfico é possível perceber que mesmo os alunos do subgrupo controle deram respostas positivas, 25% concordo totalmente e 57% concordo parcialmente.

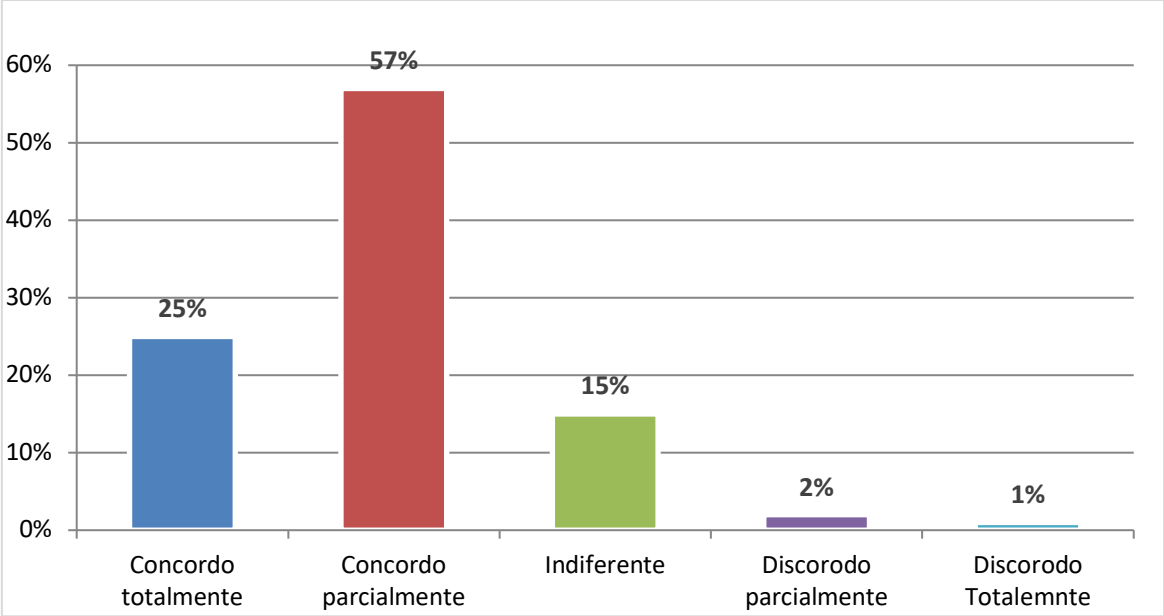


Figura 40 - Gráfico com a porcentagem das respostas dos grupos de controle do nível 2

Na Figura 41 é possível visualizar as respostas do subgrupo de teste do nível 2, são alunos que possuem nível intermediário de POO, que utilizaram o MEPOOMD para aprendizagem de POO. É possível perceber que a aceitação do uso de ambiente 3D para a criação de objetos relacionados a POO foi positiva, 37% concorda totalmente e 49% concorda parcialmente. Neste gráfico também é possível notar o baixo nível de reprovação em relação a aplicação e utilização do MEPOOMD.

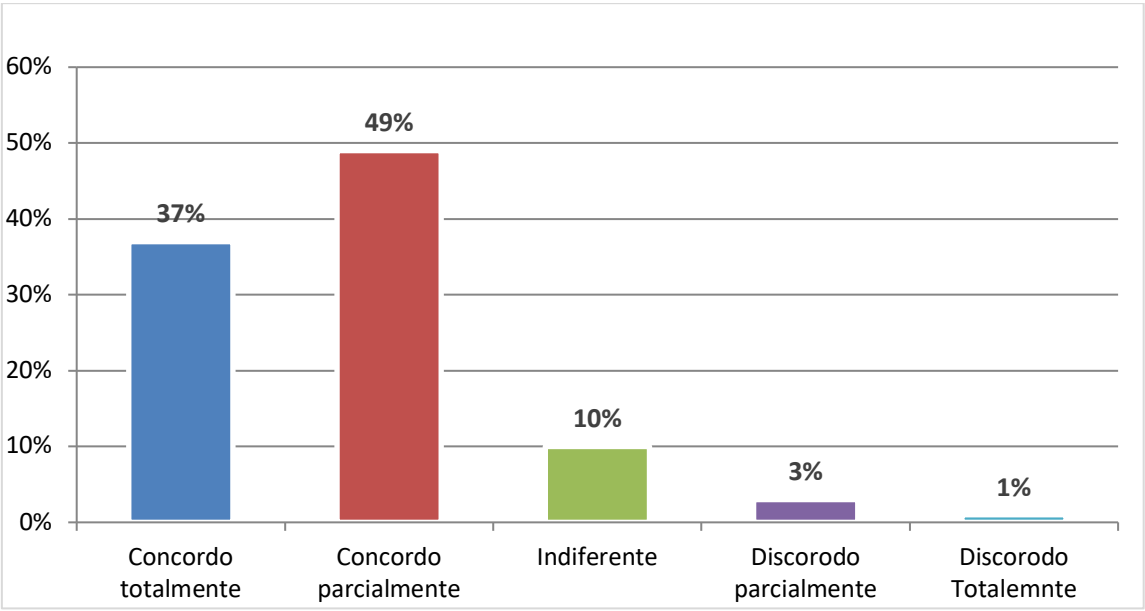


Figura 41 - Gráfico com a porcentagem das respostas dos grupos de teste do nível 2

Ao se comparar as repostas do subgrupo de controle que não foi exposto ao método, e o subgrupo teste que utilizou o método como ferramenta para seu aprendizado relacionado a POO, é possível notar uma ligeira melhoria quando aplicado o MEPOOMD. Mostrando uma boa aceitação nos dois subgrupos.

Já em relação à performance dos alunos. Os resultados do pré-teste mostram que as diferenças não estatisticamente significativas entre os subgrupos controle e teste. Os resultados variaram de *26%* a *29%* de acertos, considerando todos os grupos e subgrupos.

Os resultados dos formulários de pós-teste apresentaram *49,86%* de acertos nos subgrupos de controle e *69,96%* de acertos nos subgrupos de teste (considerando a média de todos os grupos). Ou seja, os alunos que utilizaram o MEPOOMD foram bem melhores que os que não utilizaram. Satisfeitas as condições amostrais dos testes, foram aplicados testes estatísticos para verificar se houve ganho significativo na performance para a população estudada.

Tabelas 24 e 25 apresentam medidas estatísticas de variância, graus de liberdade, *t* e *p* bilaterais, calculadas a partir dos resultados de 3 exercícios práticos (Anexo I). Estes foram desenvolvidos com o objetivo de medir o desempenho dos grupos e subgrupos na resolução de problemas usando conceitos OO.

Todos os alunos receberam 3 pontuações (uma para cada exercício), o desempenho geral é apresentado nas Tabelas 24 e 25. Quando adotamos o nível de significância de $p <= 0,01$ (*1*%), concluímos (para essa população) que a introdução da modelagem digital e fabricação resultam em melhora significativa nas pontuações dos exercícios. Como mostram as duas tabelas, a pontuação dos subgrupos de teste supera significativamente os subgrupos de controle em todos os grupos e níveis avaliados.

Tabela 23 - Resultados estatísticos - Nível 1

	Curso Técnico - Tecnologia da Informação					
	Grupo T1		*Grupo T2*		*Grupo T3*	
	Controle	*Teste*	*Controle*	*Teste*	*Controle*	*Teste*
N° de Estudantes	**22**	**22**	**18**	**18**	**13**	**13**
Média	**5.07**	**8.07**	**4.94**	**8.18**	**5.15**	**8.27**
Variância	**0.96**	**1.00**	**1.50**	**0.72**	**1.78**	**0.92**
Graus de Liberdade	**42**		**34**		**24**	
(t)	**10.0368**		**9.2142**		**6.8276**	
(p) bilateral	**< 0.0001**		**< 0.0001**		**< 0.0001**	

Tabela 24 - Resultados estatísticos - Nível 2

	Graduação – Sistemas de Informação					
	Grupo T1		*Grupo T2*		*Grupo T3*	
	Controle	*Teste*	*Controle*	*Teste*	*Controle*	*Teste*
N° de Estudantes	**26**	**29**	**26**	**25**	**24**	**24**
Média	**5.22**	**8.15**	**5.07**	**8.04**	**5.22**	**8.25**
Variância	**1.12**	**1.46**	**1.89**	**1.26**	**1.01**	**1.32**
Graus de Liberdade	**53**		**49**		**46**	
(t)	**9.49**		**8.41**		**9.74**	
(p) bilateral	**< 0.0001**		**< 0.0001**		**< 0.0001**	

6.3 Avaliação das normas comportamentais aplicadas

O método MEPOOMD utilizou como método de avaliação de suas atividades o cumprimento de normas. Esta seção foca em apresentar a avaliação, por parte do professor, se os alunos cumpriram ou não as normas modeladas para os grupos de controle. A Figura 42 apresenta o resultado das atividades aplicadas no anexo I, onde

foram desenvolvidas três atividades e feito a média em relação ao aproveitamento dos grupos de teste do nível 1.

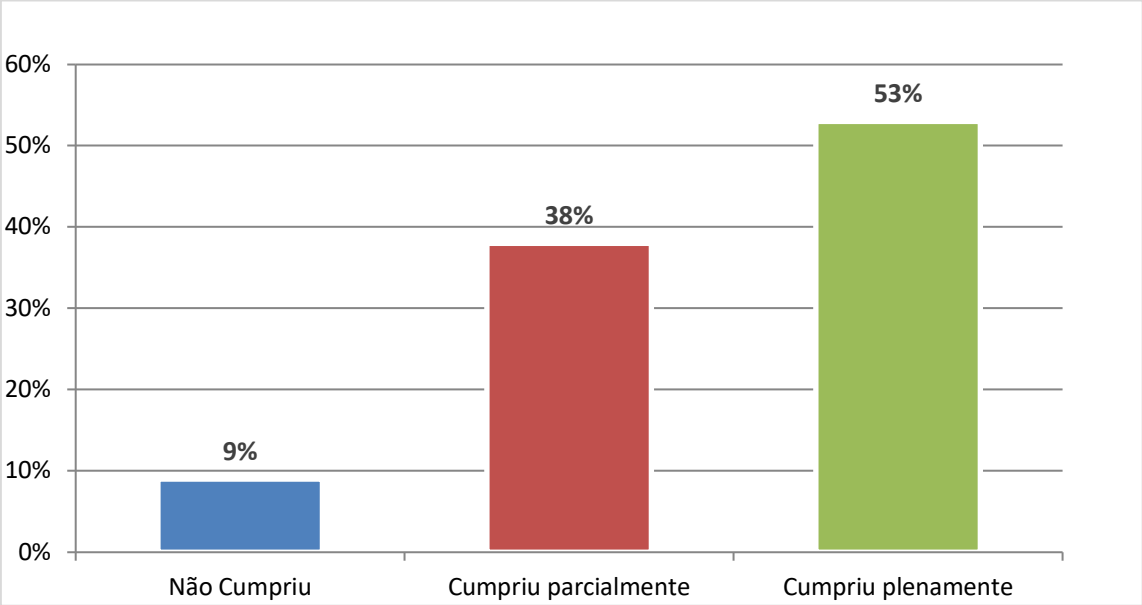


Figura 42 - Desempenho do grupo teste do nível 1 nas normas comportamentais

A Figura 42 exibe que o desempenho dos alunos dos subgrupos teste do nível 1 referente a tarefa foi satisfatório, pois *53%* dos alunos cumpriram plenamente as três atividades, *38*% cumpriu parcialmente as normas. Estes números demostram que mesmo com um nível baixo de POO do nível 1, os alunos encontravam a solução em relação a atividade desenvolvida. O não cumprimento da norma ocorreu apenas com 9% dos alunos.

Conforme exibe a Figura 43, a avaliação baseada nas normas torna possível uma avaliação por passos, podendo assim verificar onde o aluno está errando ou não cumprindo a norma de alguma forma. A tarefa de criar a herança foi a que gerou maior número de não cumprimento ou cumprimento parcial em relação as normas.

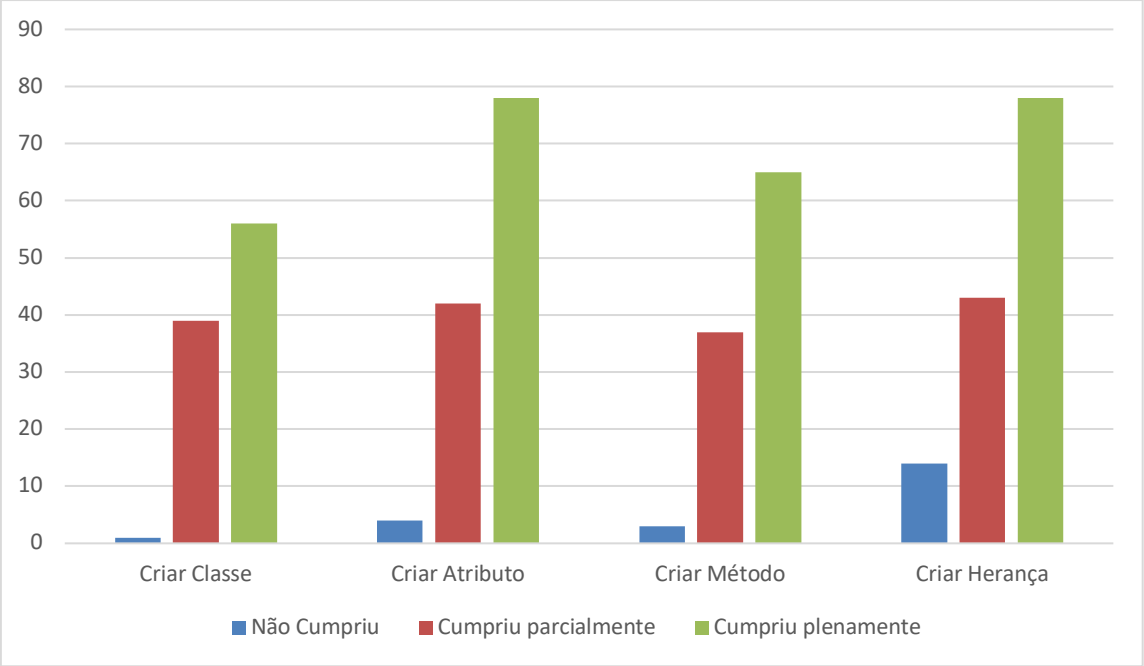


Figura 43 - Número de ocorrências dos níveis de cumprimentos das normas do nível 1

A Figura 44 exibe que o desempenho dos alunos do subgrupo teste do nível 2 em relação as normas modeladas. O desempenho também foi satisfatório, uma vez que *58%* dos alunos cumpriram plenamente as três atividades, *37%* cumpriram parcialmente as normas. O não cumprimento da norma ocorreu apenas com 5% dos alunos.

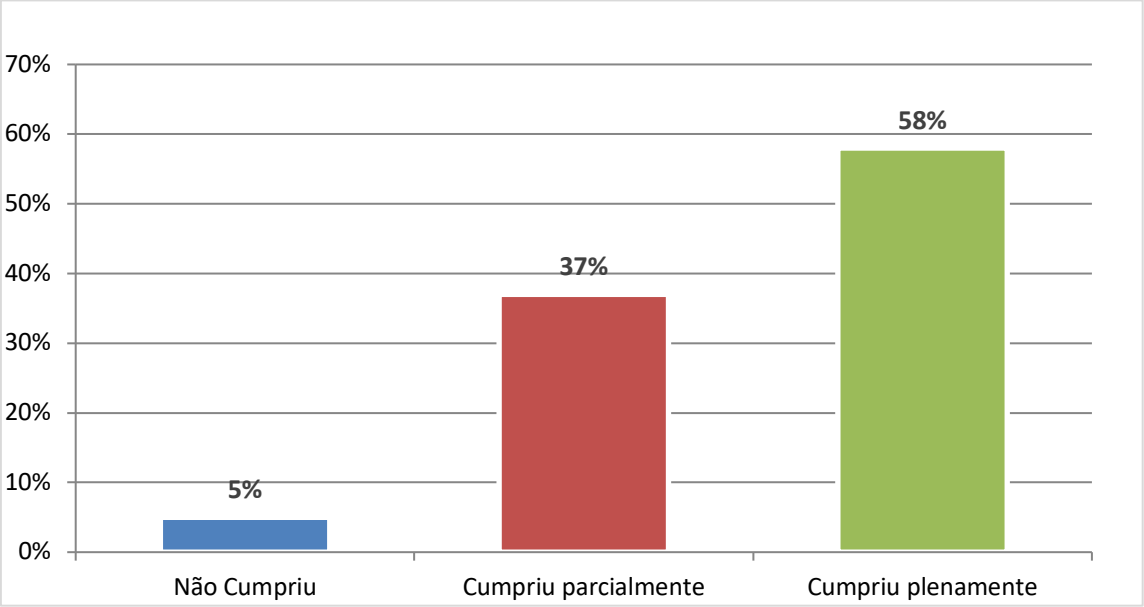


Figura 44 - Desempenho do grupo teste do nível 2 nas normas comportamentais

Conforme exibe a Figura 45, a avaliação baseada nas normas torna possível uma avaliação por passos, podendo assim verificar onde o aluno está errando ou não cumprindo a norma de alguma forma. A tarefa de criar a herança foi a que gerou maior número de não cumprimento, enquanto a tarefa de criar método gerou o maior número de cumprimento parcial no nível 2.

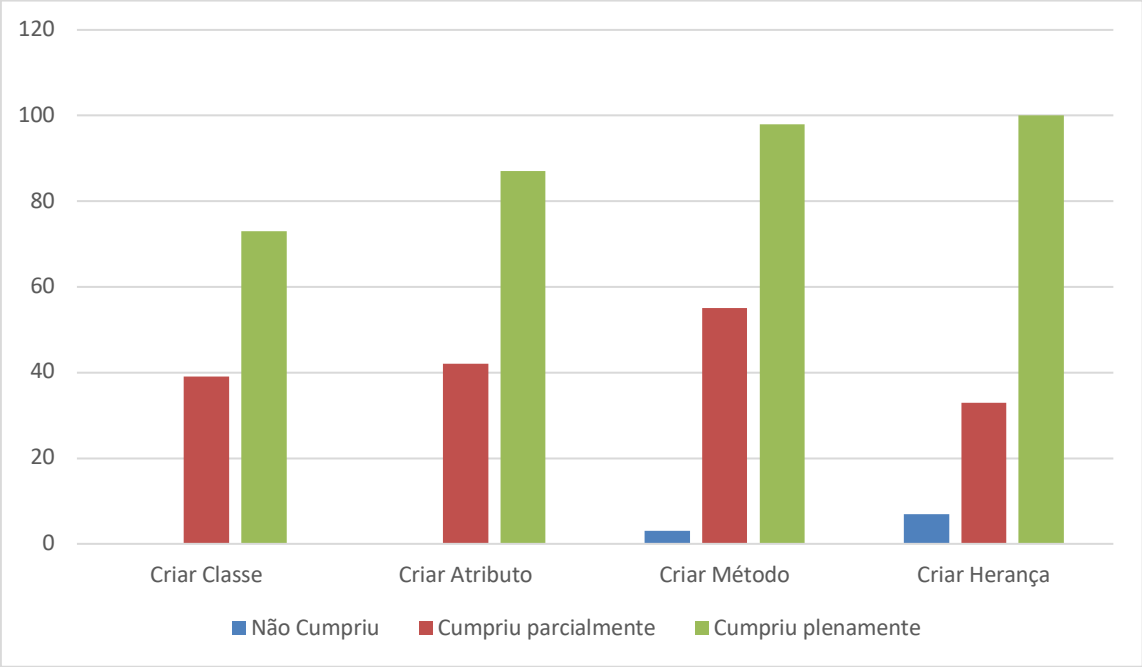


Figura 45 - Número de ocorrências dos níveis de cumprimentos das normas do nível 2

6.4 Discussão

O estudo da literatura apresentado neste trabalho confirma que o ensino de programação tem sido um grande desafio para alunos e professores em cursos técnicos e de ensino superior. Novas tecnologias, tais como modelagem 3D e fabricação digital, trazem oportunidades e desafios para o desenvolvimento de métodos e atividades que promovam o ensino de programação. Por meio da análise foi possível realizar um estudo comparativo sobre o uso dessas tecnologias. Na análise desses trabalhos foram identificados os principais métodos de ensino empregados nos estudos relatados, as tecnologias e os dispositivos utilizados, bem como o nível de programação em foco. Com isto, foi possível destacar e discutir o potencial da modelagem 3D e da fabricação

digital, as oportunidades relacionadas ao seu uso no ensino de programação e desafios a serem abordados. Esse estudo serviu de base para o MEPOOMD, que também foi fundamentado na semiótica organizacional e no design instrucional.

No trabalho de Blikstein (2013), por exemplo, foi realizado uma experiência com 4 laboratórios de fabricação digital para o incentivo à criatividade e ensino de conceitos importantes sobre a modelagem 3D. Os autores mencionam que apesar do potencial dos laboratórios de fabricação digital, os educadores devem se lembrar de que, o verdadeiro poder de qualquer tecnologia não está na técnica, mas também nos novos modos de expressão pessoal que permite, as novas formas de interação humana que facilita e as ideias poderosas que torna acessível aos alunos. Neste trabalho, foi possível identificar por meio da aplicação do MEPOOMD que a utilização das tecnologias não é o fim nela mesmas, uma vez que é possível abordar o mesmo tema com várias tecnologias e ferramentas diferentes para que o aluno se sinta motivado e consiga entender o assunto no qual o docente tenta lhe ensinar. Assim como em outros trabalhos, o aspecto motivacional pode ter influenciado positivamente na aprendizagem e no desenvolvimento das atividades como um todo.

Este trabalho está alinhado a resultados de outros trabalhos que exploram ambientes 3D (*e.g.*, Cooper (2006)) para o ensino de programação. Na aplicação do método MEPOOMD, os alunos fizeram a criação do objeto no ambiente 3D e também da programação em si. Tornando o nível de aprendizado mais apurado e preciso em relação as duas áreas, ou seja, tanto no uso da ferramenta 3D quanto na POO.

Com base nos resultados apresentados, os estudos com o MEPOOMD mostram vantagens do uso de modelagem 3D para apoiar o aprendizado de POO. Os resultados estatísticos do desempenho nos exercícios tanto no nível 1 quanto no nível 2 mostram melhoras significativas no desempenho dos alunos nos testes aplicados. Os conceitos do design instrucional foram utilizados para planejar as atividades de forma sistemática. A semiótica organizacional fornece uma visão sistêmica, na qual o ambiente de aprendizado é composto por um sistema de signos. Este sistema foi analisado em seus níveis semântico, pragmático e social.

Uma limitação em relação a avaliação qualitativa, é o fato de que as respostas fornecidas pelos alunos muitas vezes não possuem um detalhamento suficiente para a identificação do problema envolvido e as dificuldades de uma maneira clara e precisa.

Tais aspectos, podem ser analisados em pesquisas futuras, por exemplo, com técnicas de entrevistas e atividades participativas que permitam os alunos expressarem de forma mais detalhada as dificuldades encontradas. Porém, mesmo com a baixa granularidade nas respostas, é possível notar que o MEPOOMD obteve menos respostas negativas entre alunos mais avançados. O método foi avaliado de forma muito positiva em relação as questões apresentadas na avaliação realizada com alunos avançados, e de forma igual ou inferior ao grupo de controle nos alunos iniciantes. Uma das possíveis hipóteses a ser investigada é que a familiaridade com as ferramentas adotadas tenha influenciado o resultado. Algo a ser testado, é a possiblidade de uso de ferramentas de desenvolvimento e modelagem 3D mais simples com alunos iniciantes.

A avaliação quantitativa comparou as repostas do subgrupo de controle que não foi exposto ao método, e o subgrupo teste que utilizou o método como ferramenta para seu aprendizado relacionado a POO, é foi possível notar uma ligeira melhoria quando aplicado o MEPOOMD, mostrando uma boa aceitação nos dois grupos.

Como parte da avalição do método foi utilizado as normas da semiótica para mensurar o desempenho dos alunos dos subgrupos teste do nível 1 e do nível 2. De maneira geral os desempenhos nas tarefas foram satisfatórios, pois em média *45%* dos alunos cumpriram plenamente as três atividades, *32*% cumpriu parcialmente as normas. Estes números demostraram que mesmo com um nível baixo de POO, os alunos encontravam soluções em relação as atividades desenvolvidas. O não cumprimento da norma ocorreu apenas com 7% dos alunos relacionados as atividades propostas.

Algumas características do método não foram possíveis de serem exploradas durante a avaliação. Em particular o método propõe a modelagem de diagrama de ontologia durante a faze de design. Tal diagrama poderia delinear de forma mais precisa as relações de dependência entre os *affordances* existentes no ambiente de ensino. Entretanto, vale ressaltar que a construção de tal diagrama requer tempo para modelagem. Tal aspecto leva a pesquisa futuras por alternativas mais “leves” que permitam o design de forma sistemática e bem definida.

Devido a limitações de recursos e tecnológicas, embora o MEPOOMD preveja o uso de Fabricação Digital, não foi possível explorar plenamente este recurso no estudo de caso conduzido. A Fabricação Digital serviu como fator motivacional para que os alunos fizerem a modelagem em ambiente virtual. A fabricação dos objetos modelados

demandaria passos intermediários que requerem tempo e podem se distanciar do objetivo principal de uma disciplina de POO. Por outro lado, existem o potencial de propiciar uma educação integrada e contextualizada do STEM.

Trabalhos como o de Bell (2010) e Smith, Iversen e Hjorth (2015) demonstram os potenciais da fabricação digital, bem como destacam que o uso da fabricação digital em contextos educacionais formais não é um processo direto. Estes estudos revelaram que a fabricação digital no ensino primário e secundário pode se beneficiar do pensamento de projeto como parte educacional. Assim, o potencial do MEPOOMD com o uso intensivo de fabricação digital na educação deve ser explorado em novos estudos, em particular em disciplinas que envolvam não só POO mas o desenvolvimento de soluções integradas. É importante ressaltar, entretanto, que a modelagem de objetos virtuais, inspiradas na fabricação digital, já demostrou ganhos significativos no processo de aprendizagem de POO, para o grupo estudado.

7. CONCLUSÃO

Este capítulo apresenta a conclusão desta dissertação, contendo as contribuições da pesquisa, trabalhos futuros e considerações finais. A Seção 7.1 destaca as principais contribuições da pesquisa sobre as questões que delinearam todo o trabalho e os avanços resultantes deste estudo. A Seção 7.2 apresenta uma discussão sobre os trabalhos futuros, destacando limitações da pesquisa e método proposto, assim como demandas por novas pesquisas. Na Seção 7.3 são apresentadas as considerações finais com caráter crítico-analítico.

7.1 Contribuições da pesquisa

Esta pesquisa tem como objetivo responder a questão: “Como a utilização de objetos ou formas em 3D podem contribuir e aprimorar o modelo de aprendizagem de POO em cursos técnicos e na graduação?”. Foram avaliados casos práticos em ambiente real com alunos de ensino técnico e graduação na área de tecnologia da informação. A revisão de trabalhos relacionados indicou a falta de uma metodologia capaz de organizar e delimitar as práticas e avaliações para o ensino de POO.

Para tanto, o presente trabalho propôs a construção de um método de aprendizagem baseado em conceitos e técnicas do Design Instrucional e da Semiótica Organizacional. O Design Instrucional permitiu estabelecer fases bem definidas para o design de atividades com alunos. Já a Semiótica Organizacional forneceu ferramentas para analisar os *stakeholders*, estabelecer ações nos diversos níveis da escada semiótica, modelar o comportamento esperado dos alunos e estruturar avaliação quanto os aspectos perceptuais, cognitivos, avaliativos e denotativos.

Esse referencial, foi utilizado no método proposto, principalmente durante o design de atividades com alunos, de modo que explorem os conceitos de modelagem 3D (inspirados pela manufatura digital) para possibilitar a compreensão de conceitos abstratos pelos alunos. Assim, pode-se melhor compreender como a modelagem 3D pode auxiliar no processo de ensino de POO, propor de um método para desenvolver atividades de ensino de POO utilizando modelagem 3D e obter dados empíricos sobre a aplicação deste método no ensino de POO.

Com base na metodologia adotada, é necessário para a criação e desenvolvimento das práticas, o levantamento de quais conceitos devem ser abordados, como eles serão trabalhados e quem será envolvido nesse processo. A partir desta etapa, uma análise apresenta a forma sistemática da criação de práticas pedagógicas apoiada pelo método pedagógico ADDIE, organizando e elaborando em ordem de processo. A definição de normas apresenta-se de uma maneira a localizar e delimitar erros cometidos pelos alunos.

Dentre as principais contribuições deste trabalho estão:

- Desenvolvimento de um método organizado e sistemático para orientar a o design de atividades e avaliações utilizando ambiente 3D, por meio de uma proposta inovadora, utilizando técnicas de Semiótica Organizacional tendo como base o processo ADDIE.
- Aplicação de uma metodologia por meio do pesquisador/professor. O MEPOOMD possibilitou a um pesquisador-professor, com conhecimento sobre as áreas 3D, SO e ADDIE, uma forma de ensino de POO que podem ser utilizadas e adaptadas para temas específicos.
- Resultados de análise empírica de estudo sobre impacto da utilização de um ambiente 3D no ensino de POO no nível técnico em informática e também no nível de graduação. Por meio dessa análise, foi possível verificar a eficácia e deficiências do método, bem como apontar desafios e direcionamento para pesquisas futuras.

Os resultados obtidos na aplicação do MEPOOMD demostram que o método tem potencial para o desenvolvimento e o design de atividades e de instrumentos de avaliação para o ensino de POO com recursos 3D. O diferencial deste processo é a utilização do ADDIE em conjunto com métodos da Semiótica Organizacional na formatação das práticas com recursos 3D.

7.2 Trabalhos futuros

Entre as limitações desta metodologia, é possível destacar a dificuldade de tratar situações em que os alunos podem alterar a forma de como é imaginado o objeto que represente algum conceito em POO. As normas preveem uma ordem pré-definida, onde

é aguardada a criação do ambiente 3D e após a utilização da IDE como ferramenta de desenvolvimento Java.

O MEPOOMD deve ainda ser expandido para trabalhar de forma mais explícita com os elementos do ensino do STEM, uma vez que a fabricação digital foi pouco explorada (o método focou mais no modelo virtual). Por exemplo, é possível integrar a fabricação digital, a programação de dispositivos e o ensino de orientação objetos. Ainda, se faz necessário o entendimento detalhado do impacto da modelagem 3D no ensino de POO.

Em face das limitações do método, das limitações da pesquisa realizada e do direcionamento obtido a partir dos resultados desta dissertação, é possível citar os seguintes trabalhos futuros principais:

- Avançar no uso de tecnologias de fabricação digital no ensino de POO em diferentes momentos, tais como: para o planejamento e elaboração de objetos de ensino (físicos e virtuais), para criar novas maneiras para formar analogias e para propiciar a prática integrada do ensino do STEM.
- Desenvolver ambiente integrado de programação, modelagem OO e modelagem 3D para o ensino de POO.
- Desenvolvimento de um "guia prático" para a aplicação de modelagem 3D como apoio ao ensino de POO por professores (não pesquisadores da área).
- Explorar o método MEPOOMD em públicos diferentes de alunos e etapas diferentes da aprendizagem de OO como um todo.
- Aprofundar estudos qualitativos e uma análise quantitativa mais detalhada para entender como a modelagem digital e a manufatura afetam a motivação dos alunos.
- Estudo sobre o impacto do MEPOOMD na aprendizagem dos alunos em longo prazo.

7.3 Considerações finais

A utilização de um ambiente 3D para o ensino de POO é um recurso de muito potencial, tanto em questões ligadas a motivação dos alunos, quanto a diversidade de

ferramentas que podem ser utilizadas para oferecer novos modos para o ensino de programação. Porém a aplicação em curso técnico ou na graduação não é algo trivial e prático. O método proposto nesta dissertação é baseado na Semiótica Organizacional e no ADDIE, onde o professor aplicador deve estar ciente e pronto para utilizar estas ferramentas ao seu favor. A criação de um método com tais características possibilita ao professor conduzir a criação de novas atividades de maneira mais eficiente e coesa, com os objetivos esperados.

Nas pesquisas na literatura foram encontrados diversos trabalhos que demonstram os benefícios de utilizar a tecnologia 3D como ferramenta de ensino, por exemplo, em Fab Labs. Entretanto, ainda faltam trabalhos que mostrem a sua eficiência de maneira sistemática e envolvendo o ensino de POO como objeto principal, bem como faltam trabalhos com aplicações empíricas extensas com alunos em ambiente real de aprendizagem.

Os métodos e conceitos discutidos nesta dissertação tiveram como objetivo a elaboração de atividades e a avaliação de resultados considerando os aspectos comportamentais e pragmáticos envolvidos. Os conceitos de design instrucional foram utilizados para planejar as atividades de forma sistemática. A semiótica organizacional fornece uma visão sistêmica, na qual o ambiente de aprendizado contém um sistema de signos.

Para evidenciar a eficiência do MEPOOMD foi conduzido um experimento com duzentos e sessenta e um alunos com diferentes níveis de conhecimento e proficiência em POO. Os resultados apontaram uma melhoria significativa no desempenho e uma leve melhora na aceitação dos grupos de alunos que utilizaram o MEPOOMD. Com base nos resultados, os estudos com o MEPOOMD mostram vantagens do uso de modelagem 3D para apoiar o aprendizado de POO.

8. REFERÊNCIAS BIBLIOGRÁFICAS

Ambrosio, A.P. Costa, F.M.; Almeida, L.; Franco, A.; Macedo, J. Identifying cognitive abilities to improve CS1 outcome. *In: Frontiers in education conference (FIE); 2011,* out 12-15; Rapid City, SD, USA. IEEE; 2011. p.F3G-1 - F3G-7.

Amaral, L.R.; Silva, G.B.; Pantaleão E. Plataforma Robocode como Ferramenta Lúdica de Ensino de Programação de Computadores- Extensão Universitária em Escolas Públicas de Minas Gerais. *In: Simpósio brasileiro de informática na educação SBIE;* 2015, out 26-30; Maceió, AL, Brasil. 2015. p.200-208.

Anderson, Chris. Maker: A nova revolução industrial. São Paulo, SP: Elsevier Editora, 2012.

Astrachan, O., Hambrusch, S., Peckham, J., & Settle, A. (2009). *The present and future of computational thinking*. *Paper presented at the Proceedings of the 40th ACM technical symposium on Computer science education*, Chattanooga, TN, USA.

Battistella, Paulo Eduardo; VON WANGENHEIM, Christiane Gresse. ENgAGED: Um Processo de Desenvolvimento de Jogos para Ensinar Computação. *In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 2016. p. 380.

Bell, L. et al. A Special Editorial: Educational Implications of the Digital Fabrication Revolution. *TechTrends, Washington*, v. 54, n. 5, p. 2-5, 09 2010.

Bell, T., Witten, I, H. e Fellows M. (2011). Computer Science Unplugged: Ensinando Ciência da Computação sem o uso do computador. Tradução coordenada por Luciano Porto Barreto Blikstein, Paulo. (2008)

Blikstein, P. (2013). Digital Fabrication and 'Making' in Education: The Democratization of Invention. *In J. Walter-Herrmann & C. Büching (Eds.), FabLabs: Of Machines, Makers and Inventors.*

Blikstein, P & Krannich, D. The makers' movement and FabLabs in education: *experiences, technologies, and research, Proceedings of the 12th International Conference on Interaction Design and Children,* June 24-27, 2013, New York, New York, USA [doi>10.1145/2485760.2485884]

Bonacin, R. (2004). *Um modelo de desenvolvimento de sistemas para suporte a cooperação fundamentado em design participativo e semiótica organizacional.* Tese de Doutorado em Ciência da Computação, Instituto de Computação, Universidade Estadual de Campinas, Campinas-SP.

Börstler, Jürgen; BRUCE, Kim; MICHIELS, Isabel. Sixth workshop on pedagogies and tools for learning object-oriented concepts. *In: ECOOP, 17., 2003, Darmstadt. Anais* [Darnastadt]: [s.n.], [2003]. p. 84-87.

Borges, K., de Menezes, C., & Fagundes, L. (2016, November). Projetos Maker Como Forma de Promover o Desenvolvimento do Raciocínio Formal. In *Anais do Workshop de Informática na Escola* (Vol. 22, No. 1, p. 515).

Borges, M.A.F. Uma nova abordagem para o ensino de banco de dados. *Anais do Workshop de Ensino de Informática WEI98/CSBC98, Volume 1, pp.445-453, 1998.*

Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, (2014) Edition, Software Developers, Disponivel em: http://www.bls.gov/ooh/computer-and-information-technology/software developers.htm Acesso em: 23 de Jan 2017

Caspersen, M.E.; Bennedsen, J. Instructional design of a programming course. In: *International computing education research ICER '07; 2007, set 15-16; Atlanta, GA,* USA. ACM; 2007. p. 111-122.

Campos, R. L. B (2010). *ERM2C: Uma metodologia para melhoria do ensino-aprendizado de lógica de programação*. Disponível em: http://www.fejal.br/erbase2010/papers/weibase/65502.pdf. Acesso em: 03 de jun 2016.

Clements, Douglas H. Longitudinal study of the effects of Logo programming on *cognitive abilities and achievement. Journal of Educational Computing Research, v.* 3, n. 1, p. 73-94, 1987.

Cooper, S., Dann, W., & Pausch, R. (2006). Alice: a 3-D tool for introductory programming concepts. *Paperpresented at the Journal of Computing Sciences in Colleges.*

Cox, B. J. Programação orientada a objetos. São Paulo: McGraw Hill, 1999.

Dantas, Vanessa F. et al. Combinando desafios e aventura em um jogo para apoiar a aprendizagem de programação em vários níveis cognitivos. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE).* 2013. p. 920.

Deborah J. Armstrong (2012). The Quarks of Object-Oriented Development. A survey of nearly 40 years of computing literature which identified a number of fundamental concepts found in the large majority of definitions of OOP, in *descending order of popularity: Inheritance, Object, Class, Encapsulation, Method, Message Passing, Polymorphism, and Abstraction.*

Department for education. 3D printers in schools: uses in the curriculum. Enriching the teaching of STEM and design subjects. United Kingdon. 2013. Disponível em: http://dera.ioe.ac.uk/19468/ . Acesso em: 11 jun 2017

Eagle, M.; Barnes, T. Wu's Castle: Teaching arrays and loops in a game. In: 13th annual conference on Innovation and technology in computer science education ITiCSE '08; 2008, jun 30 – jul 02; Madrid, Spain. ACM; 2008. p. 245-249, 2008.

Eco, U. (1976) *Tratado geral de Semiótica*. Editora Perspectiva.

Eychenne, Fabien; NEVES, Heloisa. FAB LAB: A vanguarda da nova revolução *industrial. São Paulo: Fab Lab Brasil*, 2013.

Filatro, Andréa. Design Instrucional Contextualizado. Senac São Paulo, São Paulo-SP, 2004.

Filatro, Andréa. Design Instrucional na prática*,* Pearson, São Paulo-SP, 2008.

Ferreira, Aurélio Buarque de Holanda. Dicionário da língua portuguesa. 5. ed. Curitiba: Positivo, 2010. 2222 p. ISBN 978-85-385-4198-1

Ferreira, J. J., de Souza, C. S., de Castro Salgado, L. C., Slaviero, C., Leitão, C. F., e de Moreira, F. (2012). Combining cognitive, semiotic and discourse analysis to explore the power of notations in visual programming. In *Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium,* 101-108. IEEE.

Galdino, C.B.T.; Neto, S.R.S.; Costa, E.B. KidCoder: Uma Proposta de Ensino de Programação de forma Lúdica. In: *Simpósio brasileiro de informática na educação SBIE*; 2015, out 26-30; Maceió, AL, Brasil. 2015. p.687-691.

Goebel, Kym “How Adults Learn: Theory and Research”. 2009

Gomes, A., Areias, C., Henriques, J., e Mendes, A. J. (2008). Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte. *Revista Portuguesa de Pedagogia, v. 42, n. 2.*

Gudwin, R. R. (2002). Semiônica: uma proposta de contribuição à semiótica computacional. *UNICAMP. Campinas*.

Hernandez, C.C.; Silva, L.; Segura, R.A.; Schimiguel, J.; Ledon, M.F.; Bezerra, L.N.; et al. *Teaching Programming Principles through a Game Engine. Sciences-New York,* p. 1-8, 2010.

Iversen, O. S., Smith, R. C., Blikstein, P., Katterfeldt, E.-S., & Read, J. Eds.. 2016. *Digital Fabrication in Education, Special Issue of. International Journal for Child Computer Interaction.*

John C. Mitchell, Concepts in programming languages, Cambridge University Press, 2003, ISBN 0-521-78098-5, p.278. Lists: *Dynamic dispatch, abstraction, subtype polymorphism, and inheritance.*

Kitchenham, B. (2004) “Procedures for performing systematic reviews”, Keele, UK, Keele University, 33(2004):1–26.

Kindler, E.; Krivy, I. (2011). Object-Oriented Simulation of systems with sophisticated control International Journal of General Systems

Lewis, John; Loftus, William (2008). *Java Software Solutions Foundations of Programming Design 6th ed* Pearson Education Inc. [S.l.] ISBN 0-321-53205-8.

LI, Longshu, XU, Yi. 2010. The Teaching Research on a Case of Object-Oriented Programming. *The 5th International Conference on Computer Science & Education. IEEE.*

Liu, K. (2000). *Semiotics in Information System Engineering*. *Cambridge: Cambridge University Press,* XII, 218 p.

Lopes, Mauricio Capobianco. iPOO: uma metodologia para o ensino introdutório de orientação a objetos. Seminário de Informática - RS (SEMINFO RS'2007). *I WEI Tchê - Workshop sobre Educação em Informática*.

Luz, R A. (2009). Análise de aplicações de realidade aumentada na educação professional*: um estudo de caso no SENAI DR/GO*. Disponível em: http://www2.fc.unesp.br/wrva/artigos/50460.pdf

Martinez, S.L. and Stager, G.S. 2013. Invent to Learn: Making, Tinkering, and Engineering in the Classroom. Constructing Modern Knowledge Press.

Mitchell, R. B. (2003). Knowledge systems for sustainable development. *Proceedings of the national academy of sciences*, *100*(14), 8086-8091.

Miyasaka, Elza Luli; FABRÍCIO, Márcio Minto. Digital fabrication in Brazil: *Academic production in the last decade*. *In: 16th international conference,* caad futures 2015, 16., 2015, São Paulo. The next city New technologies and the future of the built environment. São Paulo: Universidade Estadual de Campinas Biblioteca César Lattes, 2015. v. 16, p. 422 – 433

Morris, C. W. (1938). Foundations of the Theory of Signs. In *International encyclopedia of unified science* (pp. 1-59). Chicago University Press.

Papert, S. (1980). Mindstorms: children, computers, and powerful ideas. New York: Basic Books.

Papert, S. (1987). Computer Criticism vs. Technocentric Thinking. Educational Researcher, 16(1).

Papert, S. (1991). Situating Constructionism. In S. Papert & I. Harel (Eds.), Constructionism. Cambridge, MA: MIT Press.

Papert, S. (2005). You can't think about thinking without thinking about thinking about something. Contemporary Issues in *Technology and Teacher Education*, 5(3/4), 366-367.

Peirce, C. S. (1931-1958). *Collected Papers*, Cambridge, Mass: Harvard University Press.

Pierce, Benjamin (2002). Types and Programming Languages. MIT Press. ISBN 0-262-16209-1., section 18.1 "What is Object-Oriented Programming?" Lists: Dynamic dispatch, encapsulation or multi-methods (multiple dispatch), subtype polymorphism, inheritance or delegation, open recursion ("this"/"self")

Rais, Aimi Elliyana, Sulaiman, Shahida, Syed-Mohamad, Sharifah Mashita. 2011. Game-based Approach and its Feasibility to Support the Learning of Object-Oriented Concepts and Programming. Malaysian Conference in Software Engineering (MySEC). IEEE.

Rapkiewicz, Clevi, E. . Estratégicas pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. *Rio Grande do Sul, 2006.* Disponível em: http://www.cinted.ufrgs.br/renote/dez2006/artigosrenote/25157.pdf Acesso em: 22 Jul 2017.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K.,.S lverman, B. (2009). Scratch: programming for all. *Communications of the ACM, 52*(11), 60-67.

Rodrigues, M. C. (2002) "Como Ensinar Programação?". Informática – Boletim *Informativo Ano I n° 01*, ULBRA. Canoas, RS, Brasil.

Santos, Rafael. (2003). Introdução a programação orientada a objetos usando Java. *2. ed. Rio de Janeiro: Elsevier.*

Sassure, C. (Ed.). (2004). *The Cambridge companion to saussure*. Cambridge University Press.

Sebesta, R. W. (1989) Concepts of Programing Languages. California: The Benjamin Cummings Publishing Company, Inc.

Silva filho, SS; BONACIN R, Best Practices in WebQuest Design: Stimulating the higher levels of Bloom's Taxonomy. *2016 IEEE 16th International Conference on Advanced Learning Technologie*s, pp. 391-395.

Sleeman, D. (1986). The challenges of teaching computer programming. *Communications of the ACM*, vol. 29(9), 840-841.

Stamper, R. K. (1973*). Information in Business and Administrative Systems*, Hoboken: John Wiley & Sons (ed.).

Stamper, R. K."Social Norms in requirements analysis an outline of MEASUR". In *Requirements Engineering, Technical and Social Aspects, Academic Press*, 1993.

Stamper, R. (1994). Social norms in requirements analysis: an outline of MEASUR. In *Requirements engineering* (pp. 107-139). San Diego: Academic Press Professional, Inc.

Stamper, R., Liu, K., Hafkamp, M., e Ades, Y. (2000). Understanding the roles of signs and norms in organizations-a semiotic approach to information systems design. *Journal of Behaviour & Information Technology,* vol 19(1), 15-27.

Smith, R., Iversen, O. S., & Hjorth, M. 2015. Design thinking for digital fabrication in education. International Journal of Child-Computer Interaction.

Tillman, D. (2011). Building model NASA satellites: 5lh-grade advanced mathematics students studying science using a NASA-themed transmedia book featuring digital fabrication activities.

Tillman, D. Digital fabrication as an instructional technology for supporting upper elementary and middle school science and mathematics education. *2012. 133. (Order No. 3528579) - University of Virginia, Ann Arbor*, 2012.

Zanetti, H. A. P.; BONACIN, R. Uma Metodologia Baseada em Semiótica para Elaboração e Análise de Práticas de Ensino de Programação com Robótica Pedagógica. *In: XXV Simpósio Brasileiro de Informática na Educação (SBIE),* 2014, Dourados. *Anais do XXV Simpósio Brasileiro de Informática na Educação. Porto Alegre: S*BC, 2014. p. 1233-1242.

Wilensky, U. (1999). NetLogo. Evanston, IL: Center for Connected Learning and Computer-Based Modeling. Disponivel em: http://ccl.northwestern.edu/netlogo. Acesso em: 06 fev 2017

Yasar, O., & Landau, R. H. (2003). Elements of computational science and engineering education. *SIAM review*, 787-805.

Xinogalos, S., Sartatzemi, M., Dagdilelis, V., Evangelidis, G. (2006). Teaching OOP with BlueJ: A case study. *Proceedings of the Sixth International Conference on Advanced Learning Technologies* (ICALT'06), IEEE.

ANEXO I – Informações sobre a aplicação do Método MEPOOMD

Item A – Imagens do ensino de POO utilizando MEPOOMD com alunos do UNASP Hortolândia

Figura 46 - Curso Técnico de Informática: 1° ano do Ens. Médio.

Figura 47 - Curso Técnico de Informática: 2° ano do Ens. Médio.

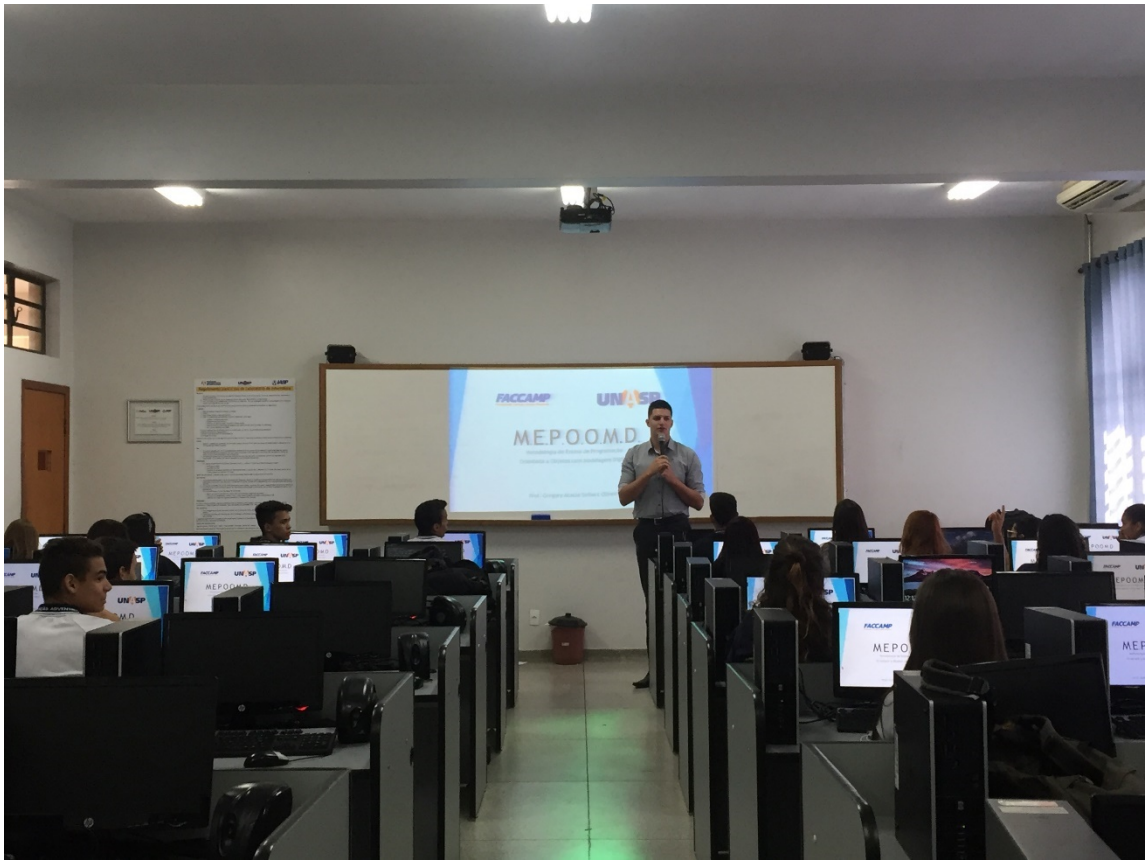


Figura 48 - Curso Técnico de Informática: 3° ano do Ens. Médio.

Figura 49 - 1°, 3° e 5° Semestres de Sistemas de Informação do UNASP-HT realizando a atividade de lógica e desenvolvimento.

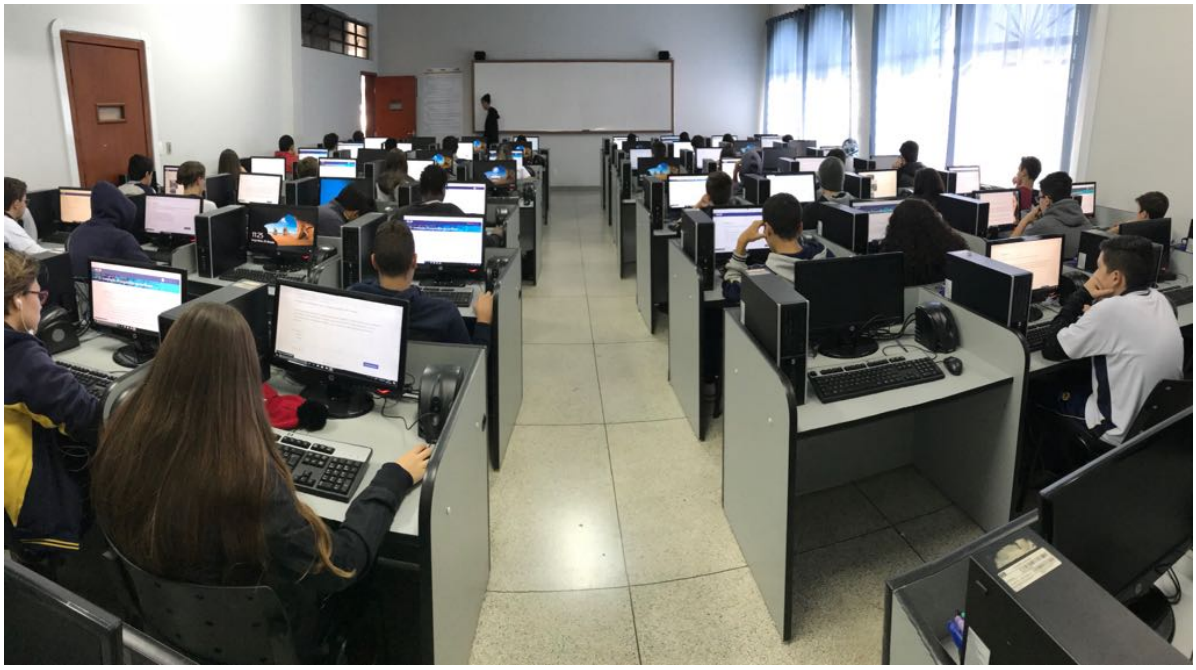


Figura 50 - 2° Ano Curso Técnico em Informática Realizando Questionário avaliativo sobre a aplicação do MEPOOMD

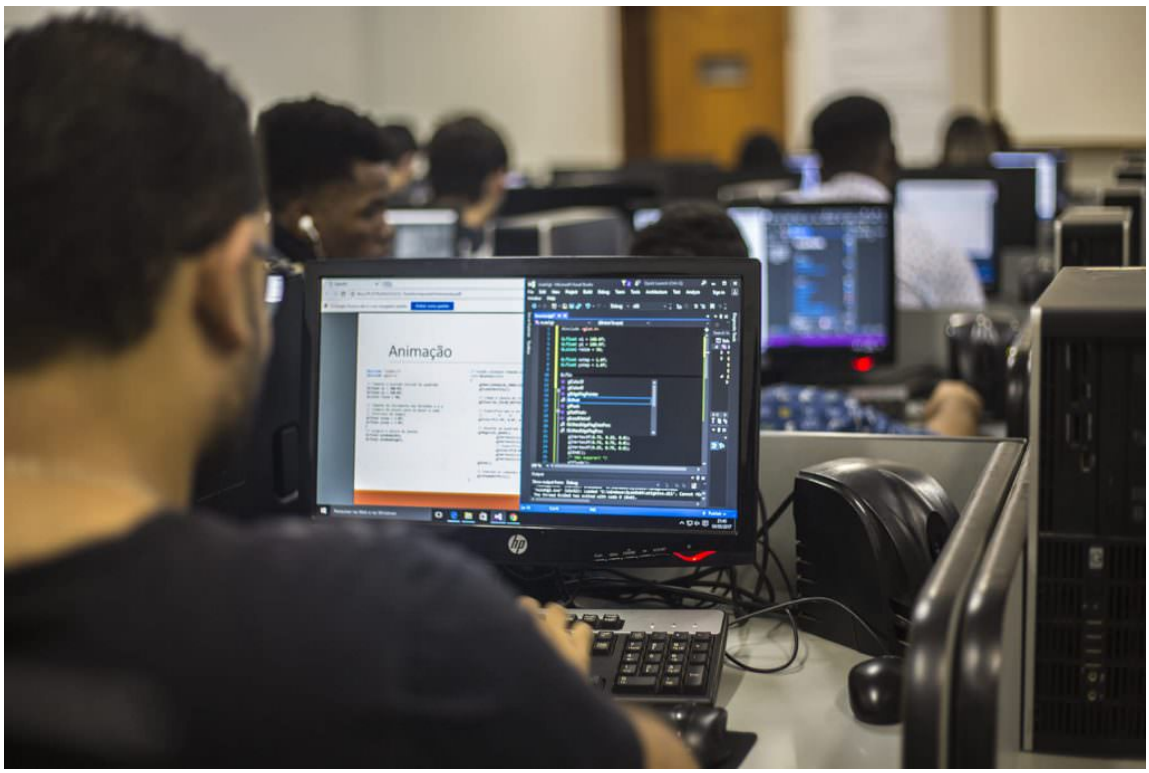


Figura 51 - Alunos do 3° Semestre de Sistemas de Informação se preparando para a criação de uma forma que representasse POO em um ambiente 3D.

Figura 52 - 5° Semestre de Sistemas de Informação realizando questionário avaliativo sobre o MEPOOMD

Figura 53 - Alunos selecionados para o subgrupo de Teste realizando a avaliação para verificar nível de aprendizado

ITEM B – Exercício 01 - realizado durante a aplicação do MEPOOMD

IASP – INSTITUTO ADVENTISTA SÃO PAULO
UNASP – Campus Hortolândia

Exercício 01
Linguagem de programação Orientado a Objetos

1. Crie uma classe chamada Complex para representar números complexos e escreva um programa para testá-la.

2. Escolha uma representação para os números complexos, usando a forma retangular ou a forma polar.

3. Forneça três construtores que permitam que objetos dessa classe sejam inicializados ao serem alocados na memória:
   - Um construtor sem parâmetros que inicializa o objeto como zero 1
   - Um construtor com um parâmetro representando a parte real; a parte imaginária será zero
   - Um construtor com dois parâmetros representando as partes real e imaginária

4. Defina operações para obter a parte real, a parte imaginária, o módulo (valor absoluto) e o ângulo de um número complexo.

5. Forneça a operação para determinar o inverso aditivo de um número complexo.

6. Forneça as operações aritméticas básicas com números complexos: adição, subtração, multiplicação e divisão.

7. Forneça as operações relacionais que permitem comparar dois números complexos.

8. Defina a operação toString para converter um número complexo em string Utilize o formato (a,b), onde a é a parte real e b é a parte imaginária. 8. Escreva um aplicativo de teste que demonstra as capacidades da classe Complex.

Este exercício prático deve ser entregue através da feramente de EAD UNASP Virtual, dentro da seção Exercícios.

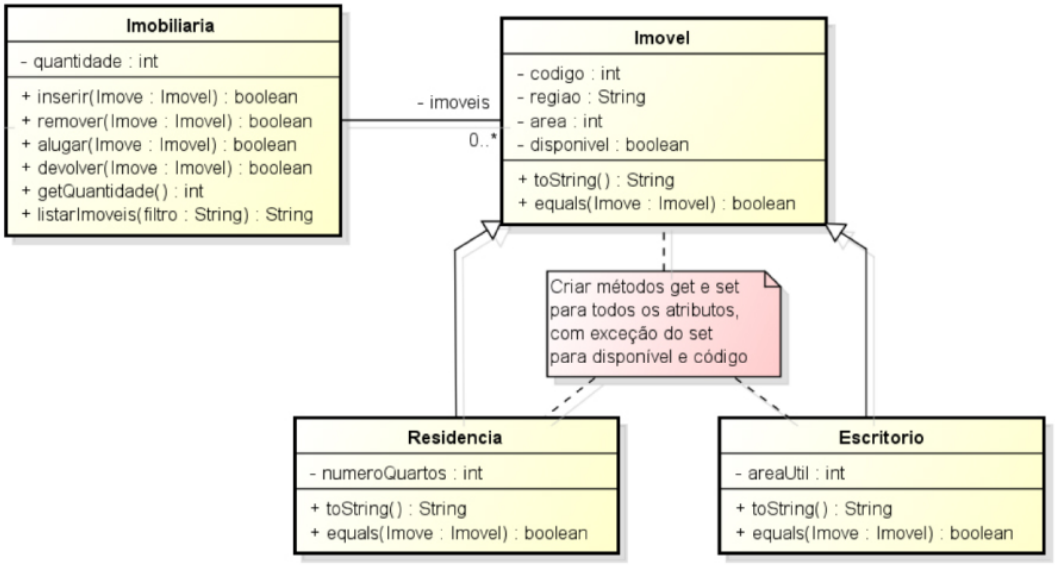
ITEM C – Exercício 02 - realizado durante a aplicação do MEPOOMD

IASP – INSTITUTO ADVENTISTA SÃO PAULO
UNASP – Campus Hortolândia

Exercício 02
Linguagem de programação Orientado a Objetos

Campus Hortolândia

Utilize os conceitos sobre herança e crie métodos get e set para todos os atributos com exceção do set para disponível e código:



Este exercício prático deve ser entregue através da feramente de EAD UNASP Virtual, dentro da seção Exercícios. (Exercício 02)

ITEM D – Exercício 03 - realizado durante a aplicação do MEPOOMD

IASP – INSTITUTO ADVENTISTA SÃO PAULO
UNASP – Campus Hortolândia

Exercício 03

Linguagem de programação Orientado a Objetos

UNASP Campus Hortolândia

1. Crie uma Classe Pessoa, contendo os atributos encapsulados, com seus respectivos seletores (getters) e modificadores (setters). Atributos: String nome; String endereço; String telefone; b. Considere, como subclasse da classe Pessoa (desenvolvida no item anterior) a classe Fornecedor.
2. Considere que cada instância da classe Fornecedor tem, para além dos atributos que caracterizam a classe Pessoa, os atributos valorCredito (correspondente ao crédito máximo atribuído ao fornecedor) e valorDivida (montante da dívida para com o fornecedor).
3. Implemente na classe Fornecedor, para além dos usuais métodos seletores e modificadores, um método obterSaldo() que devolve a diferença entre os valores dos atributos valorCredito e valorDivida.
4. Depois de implementada a classe Fornecedor, crie um programa de teste adequado que lhe permita verificar o funcionamento dos métodos implementados na classe Fornecedor e os herdados da classe Pessoa.
5. Considere, como outra subclasse da classe Pessoa, a classe Empregado. Considere que cada instância da classe Empregado tem, para além dos atributos que caracterizam a classe Pessoa, os atributos codigoSetor (inteiro), salarioBase (vencimento base) e imposto (porcentagem retida dos impostos).
6. Implemente a classe Empregado com métodos seletores e modificadores e um método calcularSalario.

   **** Escreva um programa de teste adequado para a classe Empregado.***

7. Implemente a classe Administrador como subclasse da classe Empregado. Um determinado administrador tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo ajudaDeCusto (ajudas referentes a viagens, estadias, ...). Note que deverá redefinir na classe Administrador o método herdado calcularSalario (o salário de um administrador é equivalente ao salário de um empregado usual acrescido da ajuda de custo).

   ****Escreva um programa de teste adequado para esta classe.***

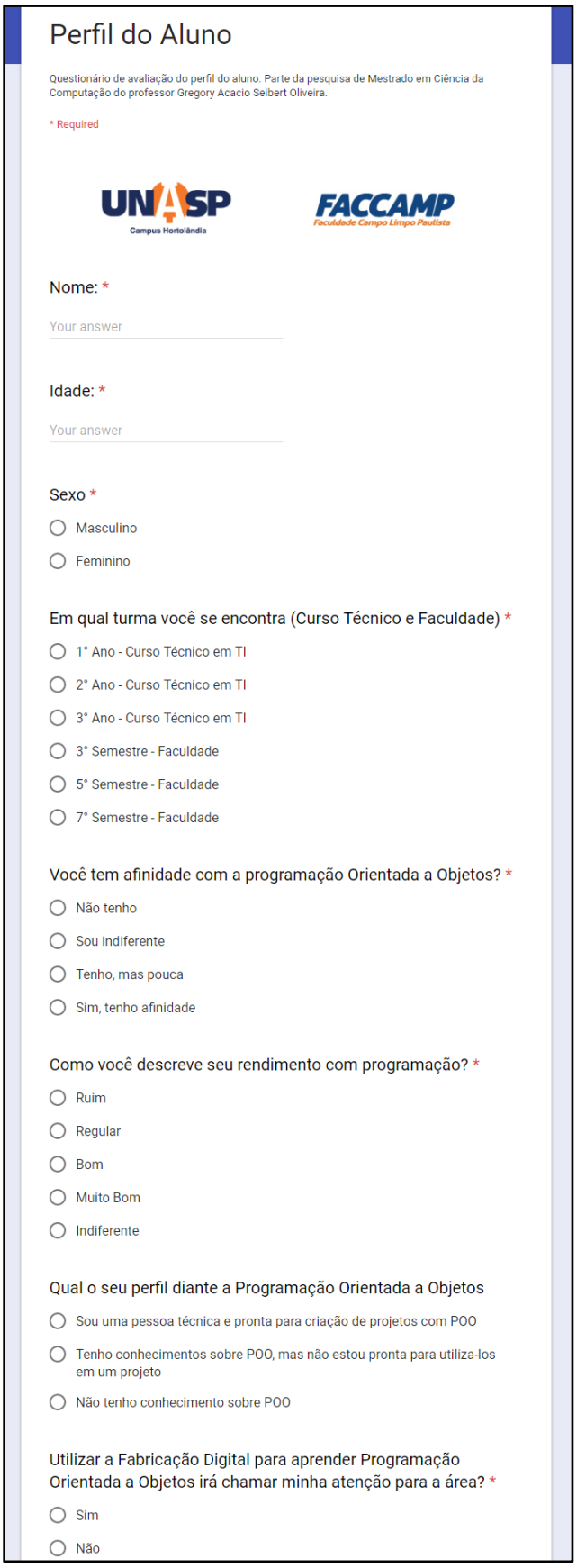
8. Implemente a classe Operario como subclasse da classe Empregado. Um determinado operário tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo valorProducao (que corresponde ao valor monetário dos artigos efetivamente produzidos pelo operário) e comissao (que corresponde à porcentagem do valorProducao que será adicionado ao vencimento base do operário). Note que deverá redefinir nesta subclasse o método herdado calcularSalario (o salário de um operário é equivalente ao salário de um empregado usual acrescido da referida comissão).

   ****Escreva um programa de teste adequado para esta classe.***

9. Implemente a classe Vendedor como subclasse da classe Empregado. Um determinado vendedor tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo valorVendas (correspondente ao valor monetário dos artigos vendidos) e o atributo comissao (porcentagem do valorVendas que será adicionado ao vencimento base do Vendedor). Note que deverá redefinir nesta subclasse o método herdado calcularSalario (o salário de um vendedor é equivalente ao salário de um empregado usual acrescido da referida comissão).

   ***Este exercício prático deve ser entregue através da feramente de EAD UNASP Virtual.**

ANEXO II – Questionário aplicado para avaliação de perfil dos alunos



Perfil do Aluno

Questionário de avaliação do perfil do aluno. Parte da pesquisa de Mestrado em Ciência da Computação do professor Gregory Acacio Seibert Oliveira.

* Required

UNASP Campus Hortolândia — FACCAMP Faculdade Campo Limpo Paulista

Nome: *

Your answer

Idade: *

Your answer

Sexo *

- Masculino
- Feminino

Em qual turma você se encontra (Curso Técnico e Faculdade) *

- 1° Ano - Curso Técnico em TI
- 2° Ano - Curso Técnico em TI
- 3° Ano - Curso Técnico em TI
- 3° Semestre - Faculdade
- 5° Semestre - Faculdade
- 7° Semestre - Faculdade

Você tem afinidade com a programação Orientada a Objetos? *

- Não tenho
- Sou indiferente
- Tenho, mas pouca
- Sim, tenho afinidade

Como você descreve seu rendimento com programação? *

- Ruim
- Regular
- Bom
- Muito Bom
- Indiferente

Qual o seu perfil diante a Programação Orientada a Objetos

- Sou uma pessoa técnica e pronta para criação de projetos com POO
- Tenho conhecimentos sobre POO, mas não estou pronta para utiliza-los em um projeto
- Não tenho conhecimento sobre POO

Utilizar a Fabricação Digital para aprender Programação Orientada a Objetos irá chamar minha atenção para a área? *

- Sim
- Não

De quanto tempo você dispõe para praticar programação no seu dia a dia? *

- ◯ 0h
- ◯ 1h
- ◯ 3h
- ◯ 4h
- ◯ 5h ou mais

Seu objetivo fazendo este curso é aprender Programação? *

- ◯ Sim
- ◯ Não

Você possui algum curso de programação que utilizou POO? Se sim quantos *

- ◯ Sim
- ◯ Não
- ◯ Other: _____

Quais os conceitos abaixo relacionados a POO você tem dificuldade de assimilar? *

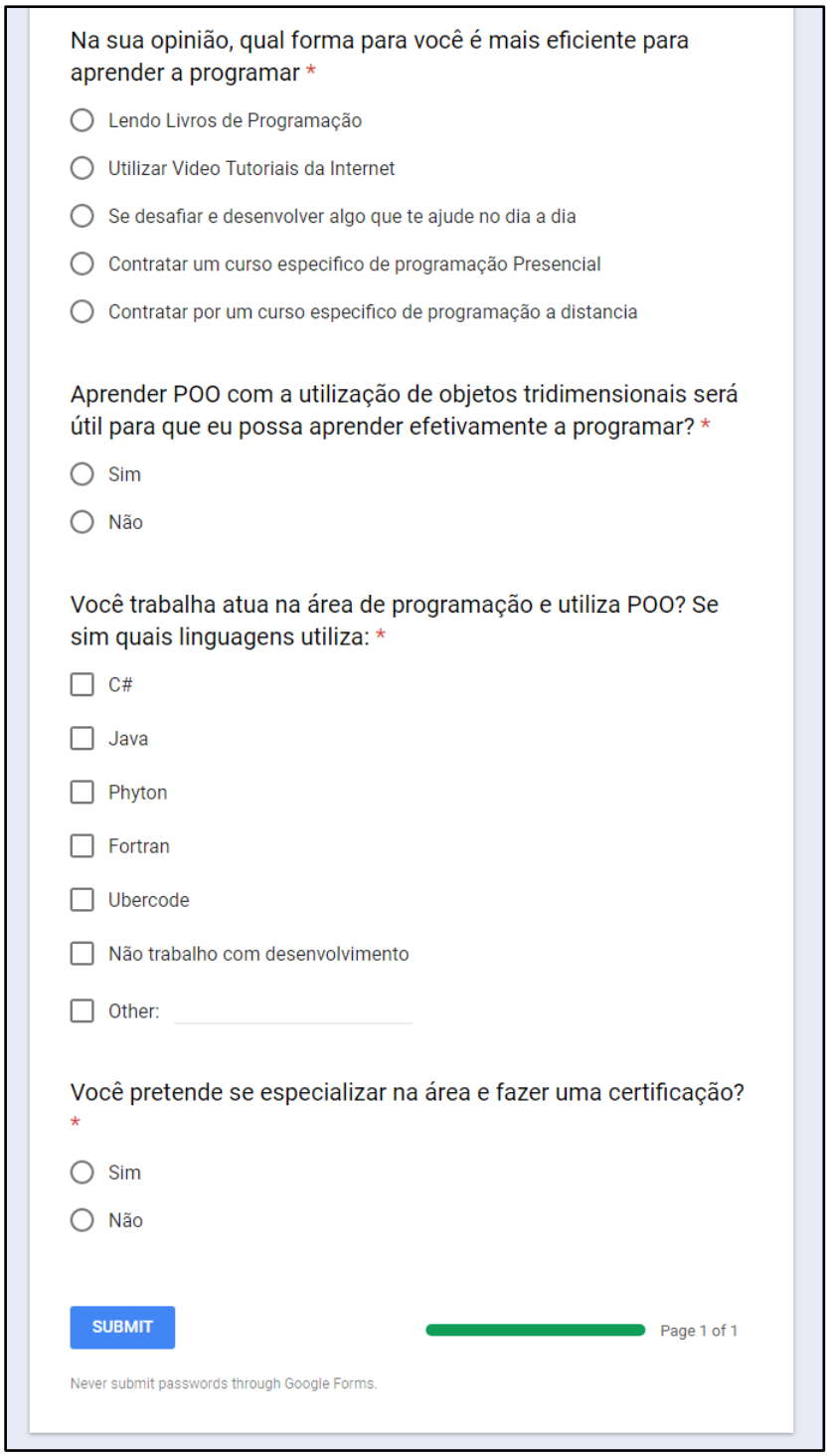
- ☐ Classes
- ☐ Objetos
- ☐ Construtores
- ☐ Interfaces
- ☐ Métodos
- ☐ Parametros
- ☐ Struct

Você acha que programação orientada a objetos está relacionado a tarefas e execuções que você realiza no seu dia a dia? *

- ◯ Sim
- ◯ Não

Na sua opinião, qual tipo de aula é melhor para aprender os conceitos de POO

- ◯ Expositiva
- ◯ Com uso de dinâmicas
- ◯ Com exercícios práticos
- ◯ Apresentação de Trabalhos
- ◯ Debates



Na sua opinião, qual forma para você é mais eficiente para aprender a programar *

- ○ Lendo Livros de Programação
- ○ Utilizar Video Tutoriais da Internet
- ○ Se desafiar e desenvolver algo que te ajude no dia a dia
- ○ Contratar um curso especifico de programação Presencial
- ○ Contratar por um curso especifico de programação a distancia

Aprender POO com a utilização de objetos tridimensionais será útil para que eu possa aprender efetivamente a programar? *

- ○ Sim
- ○ Não

Você trabalha atua na área de programação e utiliza POO? Se sim quais linguagens utiliza: *

- ☐ C#
- ☐ Java
- ☐ Phyton
- ☐ Fortran
- ☐ Ubercode
- ☐ Não trabalho com desenvolvimento
- ☐ Other:

Você pretende se especializar na área e fazer uma certificação? *

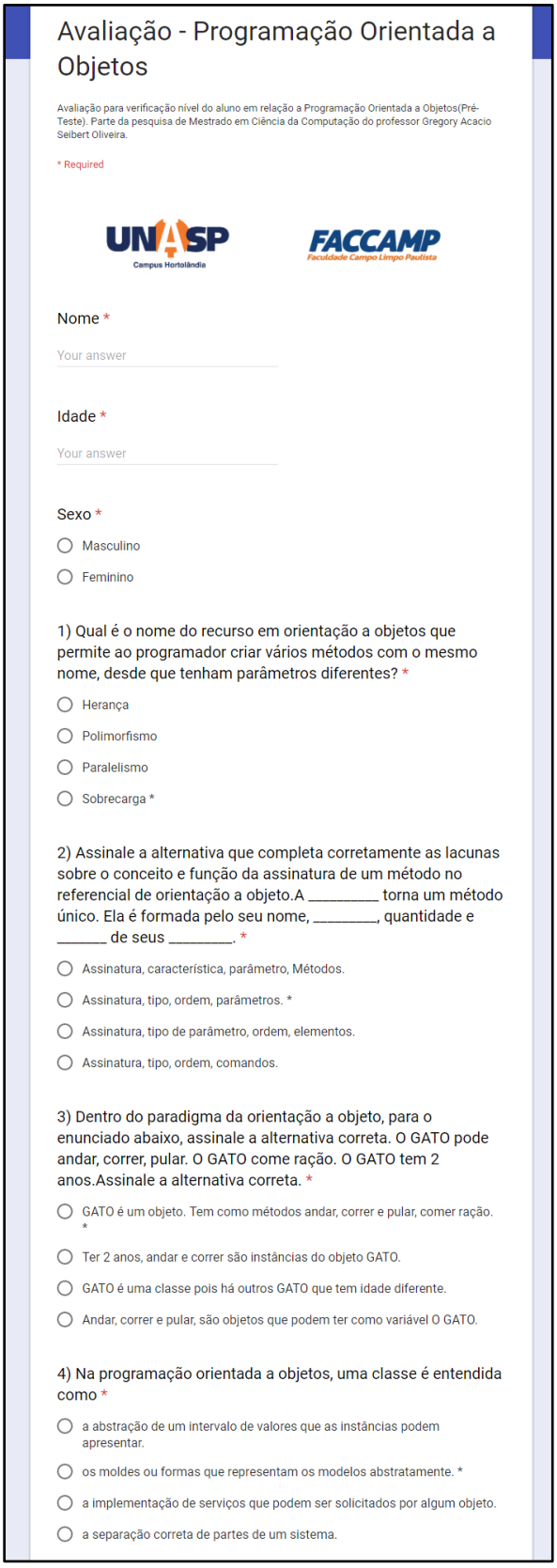
- ○ Sim
- ○ Não

SUBMIT

Page 1 of 1

Never submit passwords through Google Forms.

ANEXO III – Questionário de verificação de nível de programação orientada a objetos: pré-teste e pós-teste a aplicação do MEPOOMD



Avaliação - Programação Orientada a Objetos

Avaliação para verificação nível do aluno em relação a Programação Orientada a Objetos(Pré-Teste). Parte da pesquisa de Mestrado em Ciência da Computação do professor Gregory Acacio Seibert Oliveira.

* Required

Nome *

Your answer

Idade *

Your answer

Sexo *

- Masculino
- Feminino

1) Qual é o nome do recurso em orientação a objetos que permite ao programador criar vários métodos com o mesmo nome, desde que tenham parâmetros diferentes? *

- Herança
- Polimorfismo
- Paralelismo
- Sobrecarga *

2) Assinale a alternativa que completa corretamente as lacunas sobre o conceito e função da assinatura de um método no referencial de orientação a objeto.A _____ torna um método único. Ela é formada pelo seu nome, _____, quantidade e _____ de seus _____. *

- Assinatura, característica, parâmetro, Métodos.
- Assinatura, tipo, ordem, parâmetros. *
- Assinatura, tipo de parâmetro, ordem, elementos.
- Assinatura, tipo, ordem, comandos.

3) Dentro do paradigma da orientação a objeto, para o enunciado abaixo, assinale a alternativa correta. O GATO pode andar, correr, pular. O GATO come ração. O GATO tem 2 anos.Assinale a alternativa correta. *

- GATO é um objeto. Tem como métodos andar, correr e pular, comer ração. *
- Ter 2 anos, andar e correr são instâncias do objeto GATO.
- GATO é uma classe pois há outros GATO que tem idade diferente.
- Andar, correr e pular, são objetos que podem ter como variável O GATO.

4) Na programação orientada a objetos, uma classe é entendida como *

- a abstração de um intervalo de valores que as instâncias podem apresentar.
- os moldes ou formas que representam os modelos abstratamente. *
- a implementação de serviços que podem ser solicitados por algum objeto.
- a separação correta de partes de um sistema.

5) Em se tratando de orientação a objetos, o polimorfismo refere-se *

- ao reconhecimento do comportamento variado de um método, o que melhora o aproveitamento do código. *
- à transmissão dos métodos e atributos de uma classe para suas subclasses, quando ad hoc.
- à variação das saídas de um método com relação às entradas recebidas, caso em que é considerado universal.
- ao uso que um objeto faz dos recursos de outro objeto.
- à utilização de métodos específicos para manipular dados com proteção por meio de encapsulamento.

6) Embora use alguns conceitos de modelos e técnicas que lhe precederam, a orientação a objetos utiliza um conceito inovador, que corresponde a uma de suas características básicas. Esse conceito é o de *

- encapsulamento de dados e operações. *
- generalização / especialização.
- modularização.
- reaproveitamento de código.
- abstração

7) A aplicação do conceito de polimorfismo na orientação a objetos é realizada de forma objetiva no caso de utilização de *

- pacotes e interfaces.
- pacotes e sobrescrita de métodos.
- sobrescrita de método e atributos estáticos.
- atributos estáticos e classes abstratas.
- classes abstratas e sobrescrita de métodos. *

8) Considere três classes: Prova, Questão e PerguntaSimples. A classe Prova tem os atributos Data e Assunto, além do método criar(..). A classe Questão tem os atributos NúmeroDaQuestão e Pergunta, além do método asssociarÀprova(..). A classe PerguntaSimples tem os atributos RespostaCerta e Referência, além do método corrigir(..). De acordo com essa descrição, é possível identificar entre as classes Prova e Questão e entre as classes Questão e PerguntaSimples, respectivamente, os seguintes conceitos da orientação a objetos. *

- especialização e agregação
- herança e especialização
- agregação e decomposição
- agregação e especialização *
- classificação e decomposição

9) carro (objeto); tamanho; capacidade; porta-malas; quantidade de portas; trava porta; destrava porta; Nas informações listadas acima, referentes ao objeto carro, a informação trava porta, quando analisada em relação à orientação a objetos, representa *

- uma classe.
- um método. *
- uma herança.
- uma estrutura.
- um atributo.

10) No contexto de análise e programação orientada a objetos, é incorreto afirmar que: *

- ◯ a maneira de se obter ou alterar o estado de um objeto é pelo envio de mensagens.
- ◯ as mensagens resultam na ativação de métodos.
- ◯ o emissor da mensagem precisa saber como o objeto receptor organiza o seu estado interno. *
- ◯ os objetos trocam mensagens entre si.
- ◯ os objetos, quando compartilham uma única interface, são agrupados em classes.

11) Em POO (Programação Orientada a Objetos), dizer que a classe A estende a classe B é o mesmo que dizer que: *

- ◯ a classe B é subclasse de A;
- ◯ a classe A é superclasse de B;
- ◯ a classe A é derivada de B; *
- ◯ a classe B é derivada de A;
- ◯ as classes A e B são irmãs.

12) Em programação orientada a objetos, a possibilidade de haver funções de mesmo nome, com funcionalidades similares em classes sem nenhuma relação entre elas, denomina-se *

- ◯ encapsulamento.
- ◯ objeto.
- ◯ classe.
- ◯ polimorfismo. *
- ◯ relacionamento hierárquico.

13) Dentro do paradigma de programação orientada a objetos (POO), há um mecanismo utilizado para impedir o acesso direto ao estado de um objeto, restando apenas os métodos externos que podem alterar esses estados. Assinale a alternativa que apresenta o nome deste mecanismo. *

- ◯ Mensagem
- ◯ Herança
- ◯ Polimorfismo
- ◯ Encapsulamento *
- ◯ Subclasse

14) Método especial destinado ao preparo de novos objetos durante sua instanciação. Pode ser acionado por meio do operador new, recebendo parâmetros como métodos comuns, o que permite caracterizar os objetos já na instanciação. Trata-se de *

- ◯ operação polimórfica.
- ◯ construtor. *
- ◯ atributo.
- ◯ herança polimórfica.
- ◯ herança múltipla.

15) Dentro do paradigma de programação orientada a objetos (POO), há um mecanismo pelo qual uma classe pode estender outra classe, aproveitando seus métodos e atributos. Assinale a alternativa que apresenta o nome desse mecanismo. *

- ◯ herança. *
- ◯ mensagem.
- ◯ encapsulamento.
- ◯ polimorfismo.
- ◯ subclasse.

16) A descrição de um conjunto de entidades (reais ou abstratas) de um mesmo tipo e com as mesmas características e comportamentos. Trata-se da definição de *

- ◯ String.
- ◯ Método.
- ◯ Conjunto.
- ◯ Classe. *
- ◯ Objeto.

17) Que característica NÃO é fundamental em uma linguagem de programação orientada a objeto? *

- ◯ Criação de classes.
- ◯ Encapsulamento.
- ◯ Herança múltipla. *
- ◯ Herança simples.
- ◯ Instanciação de objetos.

18) A passagem de parâmetros por referência é, de fato, uma passagem de parâmetro por valor de um endereço de memória. *

- ◯ Certo *
- ◯ Errado

19) A especificação de uma comunicação entre objetos, que contém informações relacionadas ao que se espera resultar dessa atividade, é *

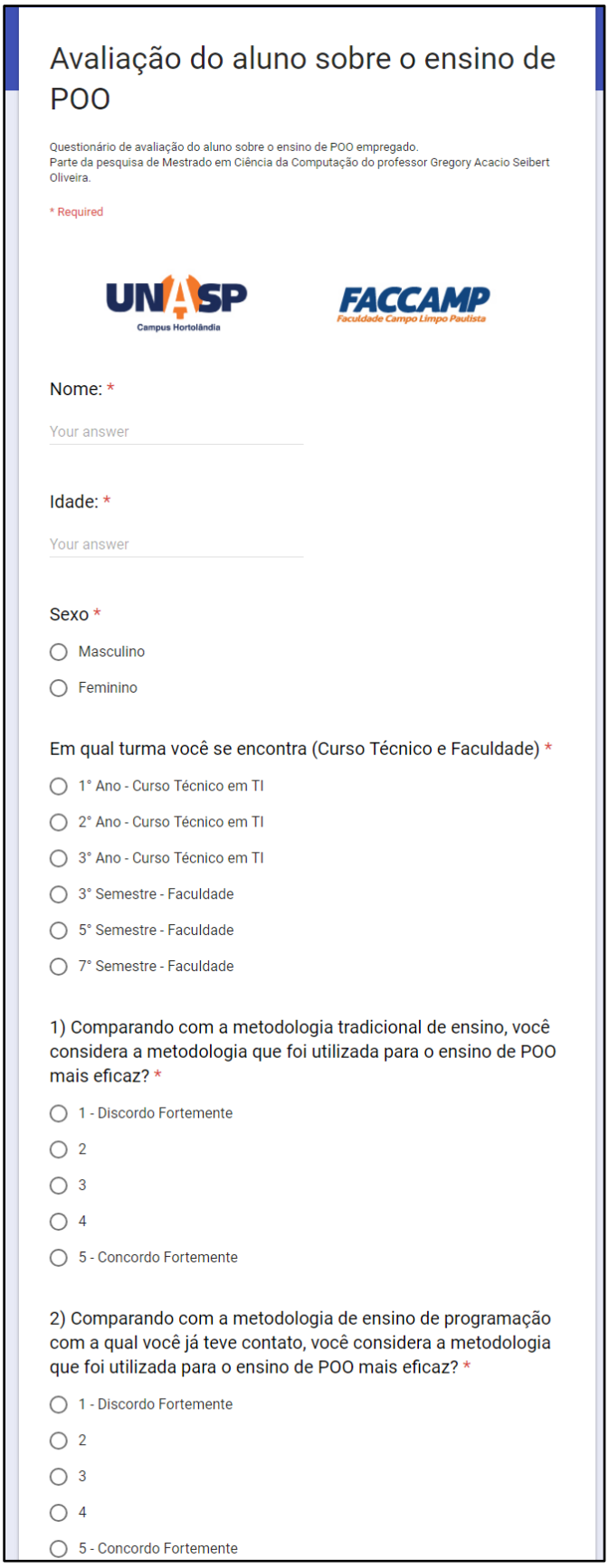
- ◯ uma restrição.
- ◯ uma mensagem. *
- ◯ uma operação.
- ◯ um processo oculto.
- ◯ um diálogo

20) Numa linguagem de programação orientada a objetos é importante restringir a visibilidade de alguns atributos para garantir o conceito de: *

- ◯ classe.
- ◯ encapsulamento. *
- ◯ herança.
- ◯ instanciação.
- ◯ polimorfismo.

SUBMIT

ANEXO IV – Avaliação do ensino de POO



Avaliação do aluno sobre o ensino de POO

Questionário de avaliação do aluno sobre o ensino de POO empregado.
Parte da pesquisa de Mestrado em Ciência da Computação do professor Gregory Acacio Seibert Oliveira.

* Required

UNASP Campus Hortolândia

FACCAMP Faculdade Campo Limpo Paulista

Nome: *

Your answer

Idade: *

Your answer

Sexo *

- Masculino
- Feminino

Em qual turma você se encontra (Curso Técnico e Faculdade) *

- 1° Ano - Curso Técnico em TI
- 2° Ano - Curso Técnico em TI
- 3° Ano - Curso Técnico em TI
- 3° Semestre - Faculdade
- 5° Semestre - Faculdade
- 7° Semestre - Faculdade

1) Comparando com a metodologia tradicional de ensino, você considera a metodologia que foi utilizada para o ensino de POO mais eficaz? *

- 1 - Discordo Fortemente
- 2
- 3
- 4
- 5 - Concordo Fortemente

2) Comparando com a metodologia de ensino de programação com a qual você já teve contato, você considera a metodologia que foi utilizada para o ensino de POO mais eficaz? *

- 1 - Discordo Fortemente
- 2
- 3
- 4
- 5 - Concordo Fortemente

3) Considerando os níveis abaixo, qual nível seria correspondente ao conteúdo que foi exposto a você durante a aplicação e ensino de POO? *

- ◯ 1 - Muito Facil
- ◯ 2 - Facil
- ◯ 3 - Itermediario
- ◯ 4 - Difícil
- ◯ 5 - Muito Difícil

4) Você considera estas aulas e o material exposto essências para a sua formação como profissional? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

5) Você acha que as aulas apresentadas deveriam ser mais praticas? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

6) Você acha que as aulas deveriam ser mais rápidas? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

7) Com os conhecimentos expostos em laboratório estou pronto para encarar o mercado de trabalho desenvolvendo utilizando a técnica de Orientação a Objetos *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

8) Durante as aulas fiquei muito confortável em desenvolver utilizando a técnica de programar orientada a objetos, que tenho interesse de me aprofundar no assunto *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

9) Aprender POO neste formato de ensino foi útil para que eu possa aprender efetivamente a programar? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

10) Durante as aulas expositivas e praticas, ficou claro para mim que podemos diminuir o nível de abstração da POO ao colocar em pratica o conceito aprendido *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

11) Ao realizar as atividades e as provas você conseguiu imaginar de forma rápida soluções computacionais para os problemas envolvidos *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

12) Ao decorrer das aulas, tarefas e avaliações o desenvolvimento Orientado a Objeto se mostrou de fácil desenvolvimento e aplicação *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

13) Você acredita que o desenvolvimento Orientado conseguiu lhe dar uma base estrutural para desenvolver em alto nível *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

14) Após a experiência vivida durante projeto, é de seu interesse ingressar em alguma certificação profissional em POO? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

15) Aprender POO neste formato foi muito interessante e acredito que será possível recordar os conceitos abordados ao programar futuramente *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

16) Aprender POO é muito abstrato e não consegui entender os conceitos expostos durante o projeto *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

17) A quantidade de exercícios e aplicações feitas para entender POO utilizando Fabricação digital foi tão grande que me deixou entediado *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

18) Estou ansioso para procurar cursos adicionais que exigem programação. *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

19) Não gosto nem de imaginar que precisarei ter que aprender (mais) programação. *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

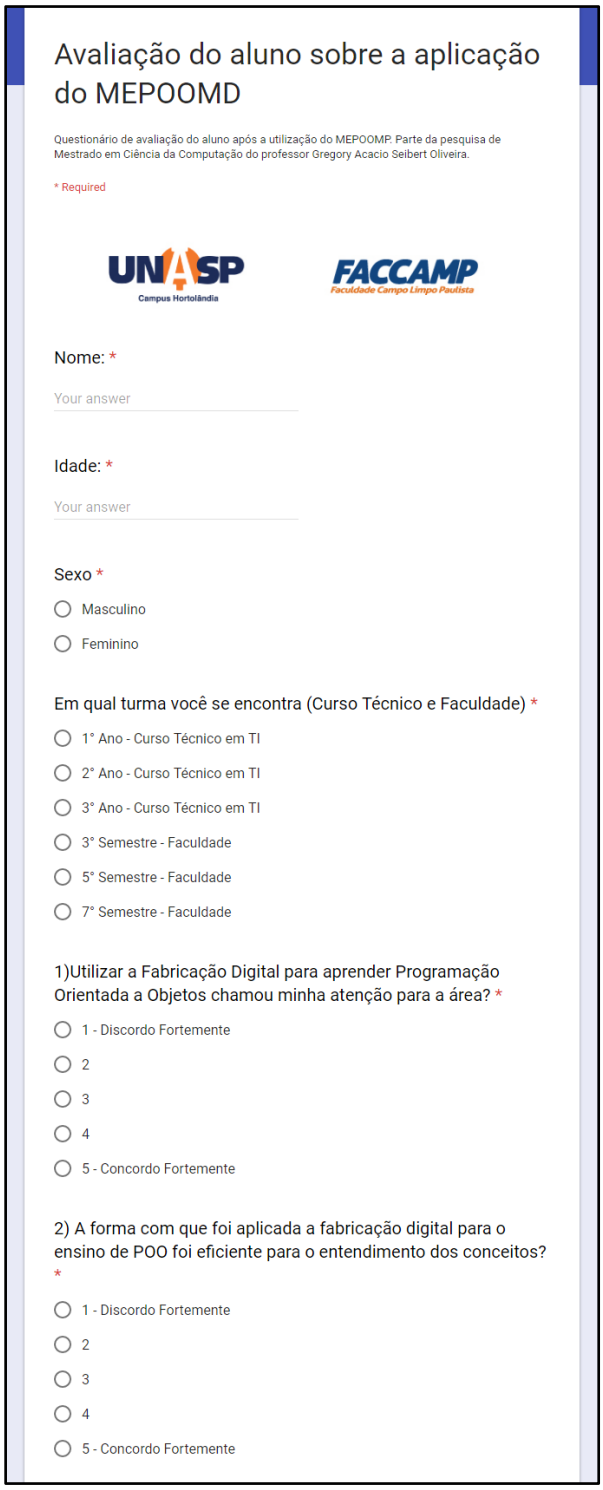
20) Estou confiante de que vou ter sucesso em cursos adicionais que requerem programação. *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

SUBMIT

Page 1 of 1

ANEXO V – Avaliação do MEPOOMD



Avaliação do aluno sobre a aplicação do MEPOOMD

Questionário de avaliação do aluno após a utilização do MEPOOMP. Parte da pesquisa de Mestrado em Ciência da Computação do professor Gregory Acacio Seibert Oliveira.

* Required

UNASP Campus Hortolândia

FACCAMP Faculdade Campo Limpo Paulista

Nome: *

Your answer

Idade: *

Your answer

Sexo *

- Masculino
- Feminino

Em qual turma você se encontra (Curso Técnico e Faculdade) *

- 1° Ano - Curso Técnico em TI
- 2° Ano - Curso Técnico em TI
- 3° Ano - Curso Técnico em TI
- 3° Semestre - Faculdade
- 5° Semestre - Faculdade
- 7° Semestre - Faculdade

1)Utilizar a Fabricação Digital para aprender Programação Orientada a Objetos chamou minha atenção para a área? *

- 1 - Discordo Fortemente
- 2
- 3
- 4
- 5 - Concordo Fortemente

2) A forma com que foi aplicada a fabricação digital para o ensino de POO foi eficiente para o entendimento dos conceitos? *

- 1 - Discordo Fortemente
- 2
- 3
- 4
- 5 - Concordo Fortemente

4) As atividades realizadas em sala para colocar em pratica os conceitos de POO utilizando objetos tridimensionais atrairão minha atenção? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

5) As varias formas de visualizar os objetos tridimensionais prenderam minha atenção ao assunto? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

6) Valeu a pena o esforço de utilizar a fabricação digital para aprender a programar com orientação a objetos? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

7) Foi prazeroso trabalhar com fabricação digital para aprender POO? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

8) Eu gostei tanto de utilizar a fabricação digital para aprender a programar orientado a objetos que tenho o interesse de aprender mais sobre isto *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

9) Aprender POO com a utilização de objetos tridimensionais será útil para que eu possa aprender efetivamente a programar? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

10) Esta muito claro para mim o quanto é viável e fácil a utilização de fabricação digital para aprender POO *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

11) Os materiais, aulas expositivas, exercícios, exemplos realmente me mostraram como a fabricação digital pode ser importante para o aprendizado de POO? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

12) Eu consegui visualizar como a programação orientada a objetos utilizando fabricação digital está relacionada aos meus pensamentos e ações do meu dia a dia *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

13) Eu consegui entender muito pouco do material que foi exibido para o aprendizado de POO *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

14) Os exercícios de POO com a utilização de objetos tridimensionais eram muito difíceis? *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

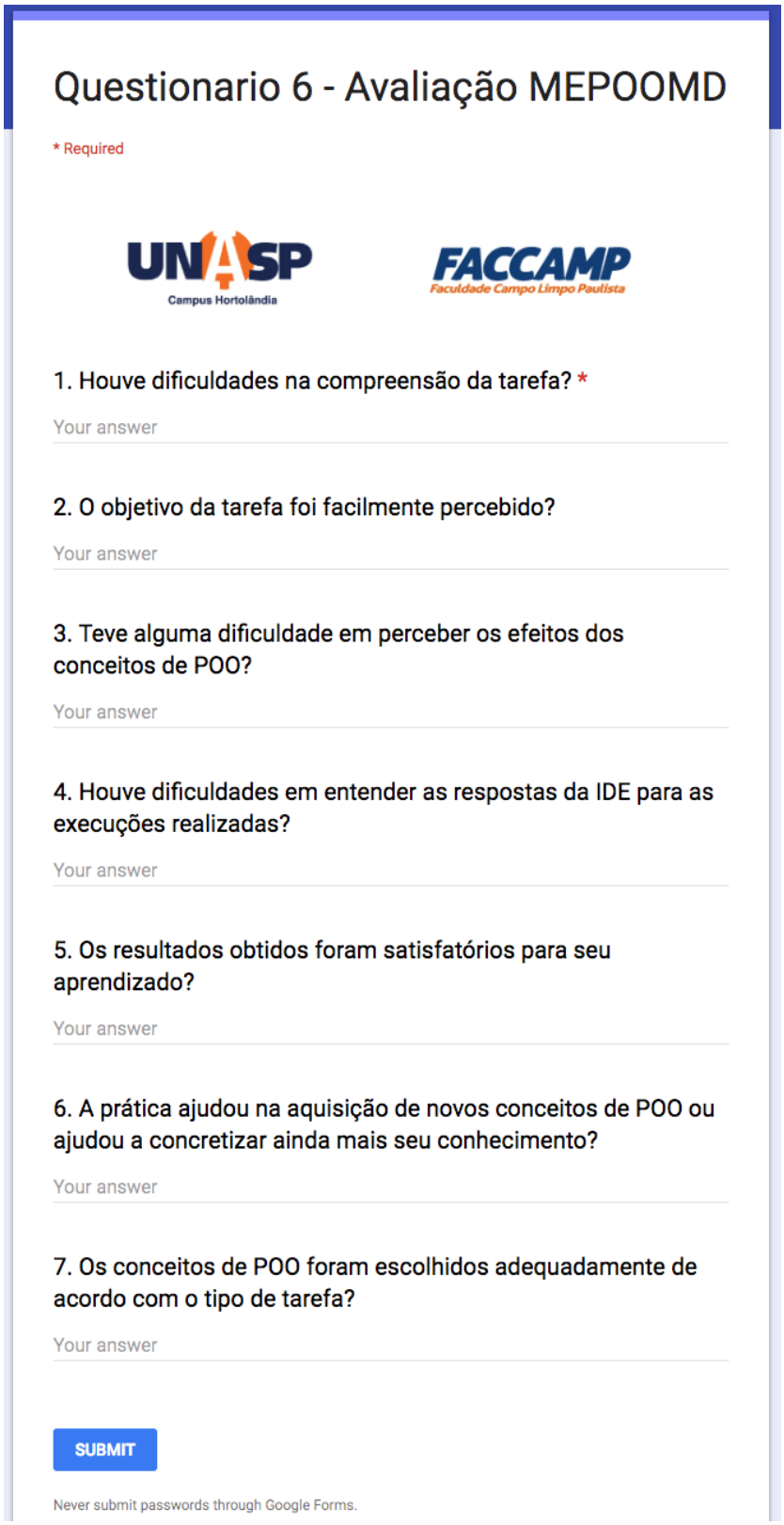
15) Aprender POO utilizando objetos tridimensionais foi fácil e também será fácil de recordar os conceitos abordados ao programar futuramente *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

16) Aprender POO com o auxilio de Objetos tridimensionais foi tão abstrato que não consegui entender os conceitos. *

- ◯ 1 - Discordo Fortemente
- ◯ 2
- ◯ 3
- ◯ 4
- ◯ 5 - Concordo Fortemente

ANEXO VI – Questionario avaliação MEPOOMD – Qualitativo



Questionario 6 - Avaliação MEPOOMD

* Required

UNASP Campus Hortolândia

FACCAMP Faculdade Campo Limpo Paulista

1. Houve dificuldades na compreensão da tarefa? *

Your answer

2. O objetivo da tarefa foi facilmente percebido?

Your answer

3. Teve alguma dificuldade em perceber os efeitos dos conceitos de POO?

Your answer

4. Houve dificuldades em entender as respostas da IDE para as execuções realizadas?

Your answer

5. Os resultados obtidos foram satisfatórios para seu aprendizado?

Your answer

6. A prática ajudou na aquisição de novos conceitos de POO ou ajudou a concretizar ainda mais seu conhecimento?

Your answer

7. Os conceitos de POO foram escolhidos adequadamente de acordo com o tipo de tarefa?

Your answer

SUBMIT

Never submit passwords through Google Forms.

ANEXO VII – Publicações Decorrentes da Dissertação

Título: A Method for Teaching Object-Oriented Programming with Digital Modeling

Resumo/Abstract:

Object-Oriented Programming (OOP) learning is one of the main challenges for students and educators in technical and undergraduate courses. Most of the OOP disciplines involve high level of complexity, which frequently generates lack of interest and increases the dropout rates. OOP requires the comprehension of high abstract concepts, as well as the use of these concepts with advanced programming methods. Several studies point out the potential of digital modeling and fabrication as alternatives for supporting integrated learning of concepts from various disciplines, such as programming, engineering, and design, among other disciplines related to STEM (Science, Technology, Engineering and Mathematics). However, the use of these technologies in formal educational contexts is not a straightforward process. Researches on methods to support their application during OOP learning activities are still needed. In this paper, we propose an innovative method for planning and executing OOP learning activities with the support of digital modeling and fabrication. This method is based on concepts of instructional design and organizational semiotics. Studies with 260 students of technical and undergraduate courses point out the potentialities and the limitations of the approach.

Referência: G. A. S. Oliveira and R. Bonacin, "A Method for Teaching Object-Oriented Programming with Digital Modeling," *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, Mumbai, 2018, pp. 233-237.

Disponível em: https://ieeexplore.ieee.org/document/8433503/

Título: Digital Fabrication and 3D Modeling Applied to the Learning of Object Oriented Skills

Resumo/Abstract:

Digital fabrication and 3D modeling bring new opportunities and challenges to teaching activities. These technologies may represent a step towards developing integrated learning of concepts from various disciplines, such as programming, engineering, and design. However, there is no single methodology for using these technologies in education, particularly in programming learning. In this paper, we present a review of existing literature on programming learning with the use of digital fabrication and 3D modeling, focusing on high-level Object Oriented concepts. This article analyzes and categorizes existing works according to the adopted teaching methods, technologies explored in the activities, and programming concepts covered. We expect to contribute to the field by mapping the adopted strategies used in these studies, as well as by presenting challenges to be faced in future research.

Referência: G. A. S. Oliveira and R. Bonacin, "Digital Fabrication and 3D Modeling Applied to the Learning of Object Oriented Skills," *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, Mumbai, 2018, pp. 290-292.

Disponível em: https://ieeexplore.ieee.org/document/8433518/