



**UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ**  
**CAMPUS LUIZ MENEGHEL - CENTRO DE CIÊNCIAS TECNOLÓGICAS**  
**SISTEMAS DE INFORMAÇÃO**

**JEFFERSON ALMEIDA BARROS VENTURINO**

**ANÁLISE E VERIFICAÇÃO DOS CONCEITOS  
OPERACIONAIS E ESTRUTURAIS EM  
PROGRAMAÇÃO COM O FOCO EM “HERANÇA”**

Bandeirantes

2015

**JEFFERSON ALMEIDA BARROS VENTURINO**

**ANÁLISE E VERIFICAÇÃO DOS CONCEITOS  
OPERACIONAIS E ESTRUTURAIS EM  
PROGRAMAÇÃO COM O FOCO EM “HERANÇA”**

Trabalho de Conclusão de Curso submetido à  
Universidade Estadual do Norte do Paraná,  
como requisito parcial para obtenção do grau  
de Bacharelado e licenciatura em Sistemas de  
Informação.

Orientador: Prof. Me. Christian James de  
Castro Bussmann

Bandeirantes

2015

**JEFFERSON ALMEIDA BARROS VENTURINO**

**ANÁLISE E VERIFICAÇÃO DOS CONCEITOS  
OPERACIONAIS E ESTRUTURAIS EM  
PROGRAMAÇÃO COM O FOCO EM “HERANÇA”**

Trabalho de Conclusão de Curso submetido à  
Universidade Estadual do Norte do Paraná,  
como requisito parcial para obtenção do grau  
de Bacharelado e licenciatura em Sistemas de  
Informação.

**COMISSÃO EXAMINADORA**

---

Prof. Me. Christian James de Castro  
Bussmann  
UENP – *Campus* Luiz Meneghel

---

Prof. Me. Estevan Braz Brandt Costa  
UENP – *Campus* Luiz Meneghel

---

Prof. Me. Lilian Aparecida Teixeira  
UENP – *Campus* Luiz Meneghel

Bandeirantes, \_\_ de \_\_\_\_\_ de 2015

## **AGRADECIMENTOS**

Primeiramente a Deus, que colocou as pessoas certas, nos momentos certos me dando força me motivando para que eu conseguisse alcançar mais esta etapa em minha vida e concluir uma graduação.

Ao meu orientador Christian James de Castro Bussmann, por acreditar na minha capacidade, pela dedicação, paciência, pelos conselhos, em todos os momentos e em cada reunião realizada uma nova aprendizagem não só referente ao trabalho, mas também na vida, uma pessoa que me estendeu a mão e se preocupou comigo, e ao coorientador Estevan Braz Brandt Costa que me aconselhou nas etapas da pesquisa e pelos conselhos de como prosseguir durante minha caminhada.

A minha família, pela paciência, confiança e motivação, pelas palavras amigas quando estava desanimado, por me erguer quando estava caído. As minhas duas irmãs que me deram atenção, palavras, motivação quando estava desanimado, acreditando que conseguiria quando nem eu mesmo acreditava.

A minha namorada Fernanda pela paciência nos finais de semana, me apoiando nas pesquisas, torcendo por mim todo esse tempo.

Aos meus amigos que sempre estiveram presentes em todos os momentos desta minha jornada, aos meus amigos Pedro Rossato e Rafael Pavinatto pela grande amizade e companheirismo durante esses anos e também a todos meus amigos que deram conselhos e me ajudaram durante toda jornada de faculdade.

Aos meus professores do curso, pela aprendizagem adquirida e principalmente paciência e dedicação nesta etapa tão importante de minha vida.

A banca de defesa com a professora Lilian Aparecida Teixeira , pela contribuição para o aprimoramento deste estudo e sempre disposta a conversar e ajudar.

Agradeço a todos que, com boas intenções, colaboraram para a realização e finalização deste trabalho. Confesso que muitos momentos pensei em desistir. Não foi fácil, mas hoje vejo que também não era impossível e que no final dá tudo sempre certo, basta ter força, foco e dedicação.

"Ninguém baterá tão forte quanto a vida. Porém, não se trata de quão forte pode bater, se trata de quão forte pode ser atingido e continuar seguindo em frente. É assim que a vitória é conquistada." (*Rocky Balboa*)

## RESUMO

Este trabalho de conclusão de curso buscou investigar o processo de construção do conhecimento do conceito de Herança, sendo este bastante presente em algumas disciplinas do curso de Sistemas de Informação. Inicialmente foi realizada uma pesquisa documental sobre o assunto tomando por base alguns livros de Programação. Foi adotada uma abordagem qualitativa tendo como objetivo entender como os estudantes do segundo e do terceiro ano do curso de sistema de informação compreendem este assunto. Para tal foi utilizado o conceito de Sfard que argumenta sobre como se dá a construção do conhecimento nos estudantes e assim evidenciando algumas etapas deste processo. Foi realizado um questionário via web em um dos laboratórios do Centro de Ciências Tecnológicas, no qual constavam questões de âmbito operacional e estrutural podendo assim evidenciar os conceitos de interiorização, condensação e reificação. A análise qualitativa dos dados obtidos foram efetivadas por meio de uma tabulação dos dados em que se pode analisar em qual fase se encontra o estudante. Onde se conclui que o conhecimento mobilizado pelos estudantes é em grande maioria processo estrutural.

**Palavras-chave:** Herança, Análise da aprendizagem, Programação, Conceito Operacional e Estrutural.

## **ABSTRACT**

This study had as aim to investigate the construction process of knowledge of the concept of heritage, which is very present in some subjects of the Information Systems course. Initially, a research about the subject was held based on some programming books. A qualitative approach was adopted aiming to understand how the students of the second and third grade comprise the concept of heritage. For this we used the Sfard concepts, who argues about the construction of knowledge of students and thus, he presents some steps of this process. Moreover, a web questionnaire was performed in one of Technological Sciences Center laboratories, in which there were operational and structural questions, can thus demonstrate the concepts of internalization, condensation and reification. The qualitative analysis of data were realized through a tabulation of data that can be analyzed in which stage the student. Thus, it was concluded that the knowledge mobilized by the student is in most structural process.

**Keywords:** Concept of heritage, Learning process, Programming.

## LISTA DE FIGURAS

Figura 1 - Tela Principal .....	35
Figura 2 - Solução do estudante A1 .....	37
Figura 3 - Solução do estudante A11 .....	37
Figura 4 - Solução do estudante A3 .....	37
Figura 5 - Solução do estudante A1 .....	38
Figura 6 - Solução do estudante A4 .....	39
Figura 7 - Solução do estudante A6 .....	39
Figura 8 - Solução do estudante A12 .....	40
Figura 9 - Solução do estudante A5 .....	40
Figura 10 - Solução do estudante A9 .....	41
Figura 11 - Solução do estudante A7 .....	42
Figura 12 - Solução do estudante A12 .....	42
Figura 13 - Solução do estudante A3 .....	43
Figura 14 - Solução do estudante A9 .....	43
Figura 15 - Solução do estudante A1 .....	44
Figura 16 - Solução do estudante A5 .....	45
Figura 17 - Solução do estudante A12 .....	45
Figura 18 - Solução do estudante A13 .....	46
Figura 19 - Solução do estudante A5 .....	47
Figura 20 - Solução do estudante A1 .....	48
Figura 21 - Solução do estudante A2 .....	49
Figura 22 - Solução do estudante A9 .....	50
Figura 23 - Solução do estudante A1 .....	51
Figura 24 - Solução do estudante A3 .....	52
Figura 25 - Solução do estudante A12 .....	53
Figura 26 - Solução do estudante A6 .....	54

## SUMARIO

1. INTRODUÇÃO .....	11
1.1 CONTEXTUALIZAÇÃO.....	11
1.2 FORMULAÇÃO DO PROBLEMA .....	14
1.3 JUSTIFICATIVA .....	15
1.4 OBJETIVOS.....	19
1.4.1 Objetivo Geral .....	19
1.4.2 Objetivos Específicos.....	19
1.5 ORGANIZAÇÃO DO TRABALHO.....	20
2. FUNDAMENTAÇÃO TEÓRICA .....	21
2.1 CONSTRUÇÃO E FORMAÇÃO DO PENSAMENTO .....	21
2.1.2 Interiorização.....	24
2.1.3 Condensação .....	24
2.1.4 Reificação .....	25
2.1.5 Herança .....	26
3. METODOLOGIA.....	27
3.1 Delimitação da natureza e da área de pesquisa .....	27
3.2 Procedimentos da Pesquisa .....	28
3.2.1 Concepção estrutural.....	29
3.2.2 Conceito Operacional.....	29
3.2.3 Interiorização .....	30
3.2.4 Condensação .....	31
3.2.5 Reificação .....	32
3.3 Elaboração do questionário de identificação e coleta de dados baseado em herança.....	33
4. DESENVOLVIMENTO .....	34
4.1 DESENVOLVIMENTO E ELABORAÇÃO DA FERRAMENTA DE COLETA DE DADOS .....	34
4.2 Elaboração e Análise dos dados.....	36
4.2.1 Discussão da Questão 1 .....	36
4.2.2 Discussão da Questão 2 .....	38
4.2.3 Discussão da Questão 3 .....	40
4.2.4 Discussão da Questão 4 .....	41
4.2.5 Discussão da Questão 5 .....	44

4.3.1 Processo Operacional.....	46
4.3.2 Discussão da Questão 6 .....	47
4.3.3 Discussão da Questão 7 .....	50
5. CONCLUSÕES .....	57
6. REFERÊNCIAS .....	59
ANEXO A.....	61

# 1. INTRODUÇÃO

## 1.1 CONTEXTUALIZAÇÃO

A programação, embora seja uma ferramenta extremamente útil e que está presente em praticamente todo o meio social, sendo utilizada em quase todas as nossas atividades diárias, tem intrigado muitos estudiosos pelo seu alto grau de complexidade.

A dificuldade no aprendizado da programação resulta em altas taxas de reprovações ou desistências nos cursos técnicos e superiores, pois se trata de uma disciplina essencial, como no caso dos cursos de análise de sistemas, ciência da computação, sistema de informação, entre muitos outros.

Segundo Jenkins (2002), estudos apontam que a tarefa de aprender a programar é entendida como algo particularmente difícil, e que vem envolvendo diversos conhecimentos e habilidades.

Ainda de acordo com Ambrósio (2011) pesquisas nesta área têm mostrado que a atividade de programar é uma atividade considerada complexa, pois possui uma linguagem que envolve subtarefas que estão ligadas a diferentes domínios de conhecimento, além de uma grande variedade de processos cognitivos que muitas vezes, são desconhecidas por alunos e professores (AMBRÓSIO *et al.*, 2011).

Seguem algumas dessas competências segundo Pea & Kurland *apud* Ambrósio (2011):

[...] A compreensão leitora; o raciocínio crítico e o pensamento sistêmico; as metacomponentes cognitivas de identificação, planejamento e resolução de problemas; a criatividade e curiosidade intelectual; as habilidades matemáticas e o raciocínio condicional; o pensamento procedimental e o raciocínio temporal; o raciocínio analítico e o raciocínio quantitativo; ou ainda, o raciocínio analógico, silogístico e combinatório. (PEA & KURLAND, 1984).

Segundo GOMES (2008), em seu estudo denominado “Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte” apontou que as dificuldades na programação estão presentes não apenas nos estudantes, mas também nos métodos de ensino em sala de aula, que muitas vezes, por terem um período muito curto de tempo não possibilita um *feedback* ou uma supervisão adequada a respeito do conhecimento ter sido absorvido de maneira adequada (GOMES, *et al*, 2008).

Deve-se considerar que além da grande complexidade de programar já mencionadas, os aspectos psicológicos também têm um grande impacto nas altas taxas de dificuldades. Sobre isso Jenkins (2002) afirma que o fato dos alunos saírem de um ambiente já conhecido, como o ensino médio, para um ensino superior, causa muita instabilidade, pois os mesmos se encontram em um período de transição<sup>1</sup> totalmente novo, fazendo com que a disciplina se torne ainda mais complexa.

Na mesma linha de Ambrósio (2011), embora todos esses fatores mostrem que a dificuldade de programar não é culpa exclusivamente do estudante e que seu fracasso se dá por diversos motivos, alguns estudos mostram que com esforço e paciência, conseguem num dado momento, compreender as tarefas de programação, este fenômeno exibe uma espécie de “salto”: onde é vivenciado um *insight* que lhes permitem transitar de uma situação onde encontravam dificuldade para uma posição que os capacitam a realizar as tarefas atividades necessária (AMBRÓSIO *et al*, 2011).

A programação, como qualquer outra linguagem, requer muito persistência e paciência para dominá-la, porém, alguns estudantes têm maior facilidade para aprendê-la, devido à sua formação anterior ou até mesmo, por fatores psicológicos.

Assim a forma de adaptar-se a programar é diferente para cada sujeito.

Todavia, algumas maneiras de aprender, como as fases de elaboração que são (interpretação, processamento, desenvolvimento) podem ser feita por todos.

Contudo, compreende-se que pular qualquer etapa poderá resultar no fracasso da aprendizagem.

---

<sup>1</sup> Transição: entende-se por transição a mudança com que o estudante se encontra saindo do ensino médio e ingressando para o ensino superior onde tal mudança faça com que ocorra uma readaptação na maneira como terá que encarar os desafios encontrados nesta nova fase e a forma com que lidará com as novas grades curriculares.

Tendo em vista tais dificuldades, foi aplicado para os estudantes do 2º e 3º ano do curso de sistema de informação da UENP - Campus Luiz Meneghel, problemas voltados à programação com o foco em “herança”. Ao final foram avaliados e analisados tais questionários, para entender em que momento os estudantes passam de um processo estrutural para um processo operacional, além de suas fases de elaboração, analisando também quais conhecimentos eles consideraram importantes e se tais conhecimentos se consolidaram.

## 1.2 FORMULAÇÃO DO PROBLEMA

A programação, para muitos, é uma das principais disciplinas dos cursos de Sistemas de Informação, Engenharia de Software, Análise de Sistemas, entre outros cursos presentes na área de TI – Tecnologia da Informação. Porém, é preocupante é o fato desta disciplina ter um alto grau de desistência ou reprovação.

Visto que este problema tem persistido por anos em várias universidades, foi elaborada uma análise no curso de Sistema de Informação, onde foram investigados: quais os conhecimentos sobre o conceito de programação com ênfase em “herança” são mobilizados por estudantes que já cursaram disciplinas com tal conteúdo?

### 1.3 JUSTIFICATIVA

O surgimento da programação se deu a mais de 70 anos atrás e foi se desenvolvendo, devido à necessidade dos países criptografarem mensagens de governos inimigos, e verificar trajetórias de mísseis e análise de balística, por conta das guerras travadas na época. Porém, com o fim de tais guerras, seu desenvolvimento foi voltado à criação de software para auxiliar a sociedade, tornando o trabalho em indústrias privadas e governamentais mais rápidos e lucrativos.

Anos depois, com o surgimento de computadores menores e mais baratos voltados à população, houve a necessidade de criar softwares mais complexos e bem elaborados, devido a grande exigência do mercado.

Assim houve a necessidade da criação de linguagens mais robustas<sup>2</sup> possibilitando facilitar a criação destes novos softwares e fazendo com que chegássemos as linguagens que conhecemos atualmente.

Devido ao aumento desta necessidade e do uso computadores também houve uma carência de profissionais que desenvolvessem tais softwares. Nesta perspectiva a criação de cursos na área de Tecnologia da Informação foi de importância significativa para tentar suprir estas necessidades. Com a criação de tais cursos, disciplinas que envolvem conteúdos de programação ficaram bastante evidentes.

Porém, mesmo com a utilização de linguagem mais robusta e simples como a linguagem “JAVA” para facilitar a aprendizagem e o entendimento dos conteúdos ministrados nos cursos a fim de reduzir a taxa de desistência, ainda assim os cursos possuem um índice de desistências e reprovações elevado comparados a outros cursos, tornando o mercado carente de profissionais.

Devido a tais problemas de reprovações e desistências, nota-se que o ensino da programação de computadores apresenta desafios que persistem após anos de pesquisa na área, em que a tarefa de aprender a programar é entendida como um

---

<sup>2</sup> Robustez- em programação robustez se entende como a habilidade de um software funcionar mesmo em condições anormais, ou a capacidade do software não dar erros não interessando o quanto errado o usuário o utilize.

processo arduamente difícil e que envolve diversos conhecimentos e habilidade (JENKINS, 2002).

Segundo Ambrósio, essas pesquisas mostraram que a programação é uma atividade que envolve subtarefas ligadas a diferentes domínios e uma gama de processos cognitivos (AMBRÓSIO *et al.*, 2011), onde Pea & Kurland *apud* Ambrósio (2011) apontam que os processos são a compreensão leitora, o raciocínio crítico o pensamento sistêmico, os metacomponentes cognitivas de identificação, o planejamento e resolução de problemas, a criatividade e curiosidade intelectual, as habilidades matemáticas e o raciocínio condicional, o pensamento procedimental e o raciocínio temporal, o raciocínio analítico e o raciocínio quantitativo, ou ainda, o raciocínio analógico, silogístico e combinatório (PEA & KURLARD, 1984).

Ainda de acordo com Almeida, essas dificuldades ampliam seu impacto se considerar as fases de transição acadêmica em que os estudantes se encontram e os processos de adaptação (ALMEIDA, 2007).

Os estudantes que veem ingressando em cursos de computação apresentam comportamentos distintos nas disciplinas introdutórias relacionadas com a programação. E alguns demonstram relativa facilidade para programar, enquanto que outros apontam enorme dificuldade (LEITHER & LEWIS, 1978). Destes últimos, alguns conseguem em certo momento, “dar um salto quântico<sup>3</sup>” e avançam na disciplina, porém outros estudantes, mesmo desenvolvendo enorme esforço, não chegam a tal “salto”.

Embora não conhecidas as condições que levam a tais “saltos”, verifica-se que a maioria dos estudantes ao persistir na aprendizagem de programação, acaba por consegui-las (AMBRÓSIO *et al.*, 2011).

Para permitir a compreensão de tais problemas, uma retrospectiva da investigação que delimita a aprendizagem da programação é feita, em que de forma geral programar é vista como uma forma de resolução de problemas que podem ser descompostas em três fases ou em quatro: entender o problema, definir ou planejar

---

<sup>3</sup> Salto Quântico: É a passagem de um processo de aprendizado que se obtem para dar inicio a outro processo de aprendizagem.

uma solução, implementar um plano, e verificar se esta certo ou se é adequado ao resultado final, onde alguns defendem sua última fase (AMBRÓSIO *et al.*, 2011).

Na análise feita por Winslow (1996), concluiu-se que iniciantes conhecem a sintaxe e semântica de comandos individuais, mas não sabem como combiná-los para obter programas válidos, e mesmo quando sabem resolver tais problemas de maneira escrita, têm dificuldade de traduzir sua solução para um programa de computador.

Porém, deve-se levar em conta que as variáveis psicológicas na aprendizagem da programação, que também influenciam no aprendizado, mostram que características com prévio conhecimento matemático e de ciências tem apresentado correlação com o desempenho dos alunos (BYRNE & LYONS, 2001; CHUMRA, 1998; WILSON & SHROCK, 2001). Surpreendentemente, estudos apresentados por Bennedsen e Caspersen (2006) concluem que não existe relação entre a capacidade de abstração e aprendizagem de programar.

Neste sentido, procura-se contrastar as funções cognitivas que poderão explicar a diferenciação entre um bom e um fraco desempenho acadêmico nas disciplinas de programação (AMBRÓSIO *et al.*, 2011). Contudo, a possibilidade de entender como os estudantes conseguem dar o “salto” devem ser aprofundadas através de estudos mais sistemáticos.

Na busca de entender tais “saltos”, Sfard (1991) em seu trabalho denominado *“On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin<sup>4</sup>”*, apresenta características que podem ser significativas.

A autora defende que um conteúdo apresenta tanto uma característica operacional quanto estrutural, que abordaremos melhor na fundamentação teórica). Ainda dentro desta concepção, ela entende que existem processos (inteorização, condensação e reificação) que podem auxiliar no entendimento de tal “salto”.

Tais estudos abordados foram aplicados dentro do conceito de “Herança” que foi escolhido devido a enorme dificuldade de entender as estruturas que são

---

<sup>4</sup> Traduzido para: Dualidade nos Conceitos Matemáticos: Reflexões no processo e objetos como diferentes lados da mesma moeda.

herdadas pelos seus semelhantes e que, por muitas vezes, os estudantes têm uma visão errada ou diferente sobre quais características serão passadas para o outro, e até mesmo o que essa alteração pode impactar na estrutura herdada. Este conceito é visto na disciplina de programação II do curso de Sistemas de Informação na Universidade Estadual do Norte do Paraná, campus Luiz Meneghel.

## **1.4 OBJETIVOS**

A seguir, apresentam-se os objetivos desta pesquisa.

### **1.4.1 Objetivo Geral**

Investigar através de questionários criado por meio de uma ferramenta para a coleta dos dados, se os discentes realizaram a passagem do processo conceitual para o estrutural, diagnosticando assim a integração e assimilação sobre o conteúdo de herança.

### **1.4.2 Objetivos Específicos**

- Identificar as dificuldades em programação;
- Verificar as dificuldades dos discentes em conhecimentos anteriores à disciplina;
- Elaborar questionários baseados nos conteúdos abordados;
- Criar uma ferramenta para a implementação e coleta de informações;
- Analisar as informações e verificar o nível de aprendizagem sobre o assunto;
- Evidenciar se ocorreu a consolidação do conhecimento por parte dos estudantes.

- **1.5 ORGANIZAÇÃO DO TRABALHO**

O trabalho está estruturado da seguinte forma.

A Seção 2 descreve a fundamentação teórica e esta dividida em: Construção e formação do pensamento, Inteorização, Condensação, Reificação e Herança.

Na Seção 3 é apresentada a metodologia para o desenvolvimento e elaboração do experimento de coleta de dados, os quais estão divididos em subseções 3.1 em que são delimitadas a natureza e área da pesquisa; nas subseções 3.2 em que são abordados os procedimentos nos quais a pesquisa foi embasada; na seção 3.3 em que foi explanada a elaboração do questionário de identificação e coleta de dados; nas 3.4 em que é apresentado o questionário de coleta de dados.

O restante do trabalho está organizado da seguinte forma. A Seção 4 apresenta o desenvolvimento do trabalho o qual estão divididos em subseções nas quais abordam a análise dos dados. A seção 5 apresenta a conclusão.

## **2. FUNDAMENTAÇÃO TEÓRICA**

Neste capítulo, apresenta-se a teoria que foi empregada por meio de pesquisas e aspectos teóricos de outros autores para que possa servir de base para a análise e interpretação dos dados coletados para o desenvolvimento e conclusão do presente trabalho.

### **2.1 CONSTRUÇÃO E FORMAÇÃO DO PENSAMENTO**

Nesta sessão, o objetivo é investigar as peculiaridades do pensamento computacional por meio de reflexões epistemológicas e ontológicas, pois dependendo do ponto de vista assumido pode-se ter duas palavras diferentes usadas para construir blocos computacionais. A palavra conceito será mencionada sempre que a ideia é concebida de forma “oficial”.

Assim como as construções matemáticas avançadas, acredita-se que as construções computacionais apresentam as mesmas características, pois são totalmente inacessíveis aos nossos sentidos (estes só podem ser vistos pelos olhos da mente) (SFARD, 1991).

Segundo este autor, ser capaz, de alguma forma, “ver esses objetos invisíveis parece ser um elemento essencial da habilidade matemática e falta desta capacidade pode ser uma das principais razões de que esta disciplina apareça impermeável a tantos”.

Nas mesmas características que Sfard (1991) argumenta sobre habilidade matemáticas, espera-se que o mesmo ocorra ao se referir à habilidades computacionais, pois como na matemática a programação necessita que o estudante, ao iniciar seu processo de aprendizagem, venha a transformar seu pensamento, que a primeiro momento é apenas visível para si próprio, em uma forma estruturada.

Tais formas podem ser vistas de várias maneiras, sejam elas em rascunhos ou até mesmo em softwares que façam com que o estudante transcenda este

pensamento em uma estrutura onde consiga manipular seu pensamento com mais clareza.

A construção desde pensamento pode ser modelada conforme o entendimento do estudante que, a primeiro momento terá que interpretar e transmitir tais interpretações em uma forma visível. Tais formas podem ser modeladas, seja em uma folha de papel ou até mesmo um diagrama utilizando softwares que visam facilitar seu entendimento.

Tornando possível que ao enxergar esta forma, o mesmo consiga adquirir uma concepção e uma visão por trás de sua construção de conhecimento criada de sua mente, tornado possível a reestruturação de sua interpretação e análise do pensamento para uma forma de linguagem de máquina ou uma linguagem de programação e entender como estes programas, métodos, funções, entre outros, venham a funcionar.

Mesmo que esta afirmação seja verdadeira, a análise de definições em livros didáticos mostra que o tratamento de conceitos se refere a objetos abstratos, sendo que esta não é a única possibilidade. Tal concepção, parece prevalecer na computação, em que se aceita definições que apresentem uma abordagem diferente. A partir de agora será chamada de concepção estrutural.

Segundo Sfard (1991), ver uma entidade matemática como um objeto, significa ser capaz de se referir a ele como se fosse uma coisa real – uma estrutura estática, existindo em algum lugar no espaço e tempo. Significa, também, ser capaz de reconhecer a ideia “de manipulá-la como um todo sem entrar em detalhes”.

Nesta perspectiva, pode-se dizer que o pensamento estrutural confere a um conceito um tipo de “fisionomia” permitindo ao estudante pensar como uma coisa única.

A interpretação de um conceito como um processo requer analisá-lo como um potencial, ao contrário de uma entidade real, que passa por existir a pedido de uma sequencia de ações. Assim, pode-se pensar que a concepção estrutural é estática e o conceito operacional é dinâmico.

Desta forma, o conceito estrutural é mais abstrato, enquanto que a concepção operacional é mais detalhada. É importante ressaltar também que as concepções estrutural e operacional não são mutuamente exclusivas. (Alguma coisa ser um processo e um objeto ao mesmo tempo) e então na verdade são complementares. Desta forma, qualquer conceito pode-ser concebido tanto estruturalmente como operacionalmente.

Para esta pesquisa entende-se que a concepção estrutural se refere a uma definição apresentadas nos livros didáticos enquanto a concepção operacional é dinâmico, sequencial e detalhado. Ou seja, apresenta uma característica qualitativa.

Sfard, em seus estudos, diz que a formação de uma concepção estrutural é, em muitas vezes um processo dolorosamente difícil e demorado, porém deixa claro que é necessário analisar a fonte de tais dificuldades (SFARD, 1991).

Para que torne esclarecedor à formação de conceitos três etapas podem ser distinguidas no processo, essas três etapas correspondem aos três “graus de estruturação”, que podem ser chamados em relação a análise teórica da relação entre os processos e objetos, onde vamos chamar esses três estágios de desenvolvimento do conceito de interiorização, condensação e reificação (SFARD, 1991, p 18).

Antes de detalhar cada estágio, temos que estar conscientes de uma dificuldade metodológica decorrente do fato de que estarmos lidando com as crenças implícitas dos estudantes sobre a natureza do conhecimento voltado a programação.

Sfard afirma que não é possível analisar os problemas de uma forma direta, porém tais estudos podem vir a ser suficientes para servir como uma ferramenta capaz de diagnosticar, talvez até a medição e a capacidade do aluno de pensar estruturalmente sobre o conceito de programação (SFARD, 1991).

### 2.1.2 Interiorização

Na fase de interiorização um estudante se familiariza com os processos que acabará por dar origem a um novo conceito (como a contagem que leva a números naturais, subtraindo o que produz números negativos).

Este processo são operações realizadas no menor nível dos objetos da programação, que aos poucos o estudante torna-se hábil em realizar tais processos, onde eles venham a ocorrer novamente no menor nível dos objetos da programação, o que torna o aluno hábil em realizar tais processos.

O termo “Interiorização” é usado aqui, em grande parte com o mesmo sentido que lhe foi dado por Piaget *apud* Sfard: Diríamos que um processo foi interiorizado se “pode ser realizada através de mentais representações” e a fim de ser considerado, analisado e comparado já que não precisa ser de fato realizado (SFARD, 1991).

No caso do número negativo, é a fase em que uma pessoa torna-se mais forte em realizar subtrações. No caso da função, é quando a ideia de variável é aprendida e a capacidade de usar uma fórmula para calcular os valores da variável “dependente” é adquirida, ou em programação onde o estudante, neste primeiro momento, começa a entender e vivenciar o conceito aprendido de tal forma que consiga entender como e para quê irá servir determinados comandos, estruturas, funções entre outros métodos que venha a encontrar no aprendizado na programação. Porém tal concepção não necessariamente precisa ser aplicada, apenas vivenciada e analisada.

### 2.1.3 Condensação

A fase de condensação é um período de “espremer” longas sequências de operações em unidades mais gerenciáveis. Nela, a pessoa possui uma maior capacidade de pensar sobre um determinado processo como um todo, sem sentir uma vontade de entrar em detalhes.

É como transformar uma parte recorrente de um programa de computador em um processo autônomo; a partir de então, o estudante remete para o processo em termos de relações de entrada e de saída, em vez de pôr uma indicação de quaisquer operações. Como no caso de procedimentos de computador, um nome pode ser dado a todo este condensado. Neste ponto é onde se pode dizer que um novo conceito foi “oficialmente” originado.

Graças à condensação, combinando-o à outros processos, fazer comparações, e generalizando tornou muito mais fácil de busca pelo conhecimento que antes não havia sido adquirido. Um progresso na condensação se manifestaria também na crescente facilidade para alternar entre diferentes representações do conceito de programação.

A partir disso, o estudante pode investigar funções, desenhar os seus gráficos, combinar pares de funções, até mesmo montar um mapeamento do problema para buscar uma solução possível.

Portanto, a fase de condensação dura tanto quanto uma nova entidade permanece firmemente ligada a um determinado processo. Somente quando uma pessoa se torna capaz de conceber a ideia como um objeto de pleno direito, é dito que o conceito foi reificado.

#### **2.1.4 Reificação**

Reificação é definida como uma mudança ontológica – a capacidade súbita de ver algo familiar em uma “luz” totalmente nova. Assim, considerando que a interiorização e a condensação são graduais, quantitativas ao invés de mudanças qualitativas, a reificação é um salto quântico instantâneo.

Tal “salto” acredita-se ser uma forma de esclarecer o que Ambrósio (2011) explica em seu estudo “Programação de computadores: Compreender as dificuldades de aprendizagem dos alunos”, onde diz que embora fatores mostrem a dificuldade de programar, os estudantes com esforço e paciência, consegue num dado momento compreender esta tarefa, onde este fenômeno exhibe uma espécie de “salto” (AMBRÓSIO *et al.*, 2011).

O estagio de reificação é o ponto em que uma interiorização de conceitos de nível mais alto (aqueles que se originam em processos realizados sobre o objeto em questão) começa. Portanto, tal fase do conhecimento só pode ocorrer se as etapas anteriores forem alcançadas.

Deve-se deixar claro que dentro da concepção estrutural, bem como da concepção operacional os três degraus de estruturação podem ser evidenciados. Tal situação será melhor abordada dentro dos procedimentos metodológicos.

### **2.1.5 Herança**

Herança é o mecanismo pelo qual uma “classe” obtém as características e métodos de outra “classe” para expandi-la ou especificá-la de alguma forma, a fim de evitar uma enorme duplicação do código e de métodos desnecessários. Ainda sob o ponto de vista prático da orientação a objetos, a herança é constituída de ou classificadas por dois tipos: “herança simples” e “herança múltiplas” (SIERRA & BATES, 2005).

A Herança é um mecanismo muito inteligente de aproveitamento de código e que evita a duplicação deste em subclasses. Por consequência disso, a vantagem é a reutilização toda do código já implementado na classe que vamos nos referir como “Pai”, restando apenas à implementação dos métodos e atributos que a classe “Filho” difere-se da classe “Pai” (SIERRA & BATES, 2005).

A Herança também permite que apenas alguns métodos sejam reestruturados, possibilitando que mesmo que uma classe “Filho” herde as características da classe “Pai”, ele tenha suas próprias características distintas. Além disso, também permite que uma classe “Pai” possa transmitir apenas seus métodos sua classe “Filho-1” e transmitir seus métodos para a classe “Filho-2” (SIERRA & BATES, 2005).

Tornando assim, a Herança uma excelente maneira para otimizar a quantidade de linhas de código e possibilitando uma melhor análise e agrupamento de seus métodos e funções, o que se torna um facilitador para a verificação de erros e manutenções em programas, seja eles de fácil, médio ou grande porte.

### 3. METODOLOGIA

Neste capítulo, apresenta-se o encaminhamento metodológico que será assumido nesta pesquisa e assim mostrar o desenvolvimento de nossa investigação. Nela, será discutida a natureza da pesquisa, descrevendo a forma com que pesquisamos, os procedimentos adotados para obtenção de informações e o enfoque utilizado na análise.

#### 3.1 Delimitação da natureza e da área de pesquisa

Na busca de investigar a participação da programação mais especificamente o conceito de Herança na formação inicial de estudantes em Sistemas de Informação na ótica de pesquisadores, procurou-se observar como os estudantes que cursam ou já cursaram a disciplina de programação concebem tais entendimentos sobre estes assuntos.

Considera-se que a abordagem de caráter qualitativo é a mais adequada para essa investigação, pois de acordo com Bogdan e Biklen *apud* Balestri (2008), uma investigação qualitativa é descritiva e o interesse maior é pelo processo de investigação e não simplesmente pelos resultados obtidos. Neste pesquisa apresenta-se uma descrição das informações obtidas para análise posterior.

Segundo estes mesmos autores, nessa modalidade de pesquisa nada pode ser considerado trivial, pois “tudo tem potencial para constituir uma pista que nos permita estabelecer uma compreensão mais esclarecedora do nosso objeto de estudo”.

De acordo com Garnica (2004), algumas das características de uma pesquisa qualitativa são:

- (a) A transitoriedade de seus resultados; (b) a impossibilidade de uma hipótese a priori, cujo objetivo da pesquisa será comprovar ou refutar; (c) a não neutralidade do pesquisador que, no processo interpretativo, vale-se de suas perspectivas e filtros vivenciais prévios dos quais não consegue se desvencilhar; (d) que a constituição de suas compreensões dá-se não como resultado, mas numa trajetória em que essas mesmas compreensões e também os meios de obtê-las podem ser (re) configuradas; e (e) a impossibilidade de estabelecer

regulamentações, em procedimentos sistemáticos, prévios, estáticos e generalistas.

Nesta pesquisa, não se pretendeu alcançar uma verdade absoluta imutável. Portanto, os resultados obtidos são transitórios, e com outros “olhares” sobre o objeto de investigação pode possibilitar uma outra visão que também poderá responder a questão de investigação.

No decorrer do desenvolvimento da pesquisa foi colocado como elemento ativo a obtenção e análise dos dados. No processo de obtenção das respostas foi desenvolvida uma ferramenta para a coleta das informações, para que fosse possível interpretar os dados coletados, formulando e reformulando a compreensão baseadas em nossos estudos.

### **3.2 Procedimentos da Pesquisa**

Para a criação do questionário houve a necessidade de verificar o conceito que seria empregado no mesmo, através da verificação do livro “Use a Cabeça Java” (SIERRA & BATES p. 123 - 131, 2005). Tal conhecimento buscou a estruturação de uma forma que a cada nova questão o estudante enfrente uma nova forma de verificar seu pensamento crítico.

A partir disso, a coleta de dados foi realizada por meio de um “*site*”, possibilitando implementar questionários semiabertos, onde será avaliada tanto a opinião com relação aos problemas apresentados, quanto possíveis estratégias para melhor resolvê-los.

Para a análise mais aprofundada foi utilizado Sfard (1991) que aponta que um conceito abstrato é apresentado como duas concepções, estrutural (objeto) e operacional (processo) bem como suas fases de interiorização, condensação e reificação (SFARD, 1991).

### **3.2.1 Concepção estrutural**

Entender sobre concepção estrutural e analisá-la, leva uma gama de ideias e de possibilidade que surgem em nossas mentes, contudo, o foco é definir tal estrutura na área de programação, especificamente, voltada para “Herança”.

Entende-se que o conceito estrutural e sua concepção é formada quando o estudante consegue transcrever seu pensamento antes abstrato, que apenas existia na forma de ideias e conceitos, e que não correspondiam ao primeiro momento devido a dificuldade de compreender tais conceito devido a significar algo vago para eles.

Porém, conforme este estudante consegue adquirir a capacidade de transcender tais ideias e pensamento em uma forma de escrita, seja em papel ou em algum programa computacional, que não necessariamente necessite de ser em linguagem de máquina, a concepção estrutural será formada.

### **3.2.2 Conceito Operacional**

Sfard (1991), em seus estudos, aponta que, no olhar da perspectiva matemática, uma função no processo estrutural é a “Configuração dos pares ordenados” e no conceito operacional é um “Processo computacional ou método bem definido de obtenção a partir de um sistema para outro” (SFARD, 1991).

Baseado nesta análise, entende-se que na programação o conceito operacional se aplica quando o estudante transcreve seu conhecimento estrutural, em que será olhado por uma perspectiva como sendo algo que tenha a possibilidade ou condições para funcionar ou realizar determinada operação, que por sua vez se dá pela forma com que ele obteve o conhecimento necessário sobre determinado problemas para transcrever seu pensamento à uma maneira que possibilitará uma análise mais aprofundada.

A partir deste ponto, o estudante buscará apenas uma forma de redesenhar seu pensamento para uma linguagem de máquina e com isso, obter uma maneira operacional de ver os problemas analisados para tornar o problema em algo funcional, assim o conceito operacional é formado.

Analisando o olhar operacional pode-se dizer que o processo operacional na programação com base em “Herança” se dá quando o estudante transmite sua estrutura do problema ou conceito, que antes estava apenas em sua mente para uma maneira que seja possível obter uma análise da dificuldade como um todo, assim o conceito estrutural é formado, e torna possível que o estudante consiga ter uma percepção que a resolução de qualquer problema se dá pelo fato de ter que dividir as classes, sendo necessário haver uma hierarquia que possibilita dizer qual classe herdará os métodos e atributos, e qual classe será a dominante.

Pois como Ambrósio explica em seu trabalho, denominado “Programação de computadores: Compreender as Dificuldades de Aprendizagem dos Alunos”, para se obter uma taxa maior de sucesso em seus problemas é necessário que o estudante passe com eles em três etapas que são: interpretação, estruturação e desenvolvimento (AMBRÓSIO, 2011).

Assim, para se obter uma concepção operacional sobre Herança o estudante terá que, além de conseguir que tal problema funcione, ou opere, baseado em sua estrutura montada na fase anterior, conhecer os meios com que a linguagem de máquina entenda essa reestruturação de herança e seus métodos que a compõem.

Tal processo de divisão torna o sucesso, a elaboração, a estruturação do programa mais fácil, possibilitando tratar a estrutura da Herança mais facilmente, tal comportamento leva o estudante a pensar como um computador e assim transformar seu pensamento estrutural em operacional (AMBRÓSIO, 2011).

### **3.2.3 Interiorização**

A interiorização se dá quando o estudante se depara pela primeira vez com um determinado conceito que, a partir disso será possível realizar um pensamento no menor nível do conhecimento. Tal nível, baseado no conceito de Herança se dá quando o estudante passa a entender o conceito básico de que um objeto pode herdar características iguais e que tais objetos tornam rápidos e práticos a execução de problemas, possibilitando facilitar a estruturação em um nível mais aprofundado.

Este conceito embora simples não precisa ser de fato realizado, apenas feito através de representações mentais a fim de ser analisado e comparado, onde o estudante terá que pensar de uma maneira lógica e estruturada.

A partir disso, notamos que programação como a matemática é uma disciplina que o conhecimento só se constrói a partir do conhecimento anterior, por consequência disso será inferido que o estudante já tenha um conhecimento bem definido sobre “classes e objetos”, “métodos e mensagens”, entre outros. Para então chegar ao ponto de nossos estudos que será voltado a Herança. Piaget diz que um processo foi interiorizado se “pode ser realizado através do pensamento e por pequenas representações” (PIAGET, 1970, p.14).

A partir deste pequeno entendimento, a interiorização se tornará mais forte após o estudante compreender que o uso da Herança torna mais rápido e prático a atualização de métodos para transmiti-los a seus herdeiros de forma ágil e eficaz, pois não necessitará atualizar as outras classes. Outro caso é quando o estudante consegue perceber que um “leão”, “gato”, “tigre” embora diferentes possuem certas características semelhantes, em que a percepção de criar uma classe separada para atribuir variáveis para ambos seja algo necessário.

### **3.2.4 Condensação**

A fase de condensação se dá quando o estudante começa a introduzir o conhecimento de forma automática, sem querer buscar detalhes, o que antes era preciso, pois não havia desenvolvido tal capacidade e seu pensamento sobre Herança e como ela seria usada era necessário. Esse processo é como tornar um processo de um computador em um processo autônomo (SFARD, 1991).

No que diz respeito à condensação em um processo estrutural o estudante deve buscar alternar entre diferentes formas de representar o conceito de Herança, como em visualizar que uma os atributos de uma classe “pessoa” possa ser usado tanto em uma classe “funcionário” quanto em “cliente” sendo que ambos possam estar passando suas características para uma outra classe “funcionário-cliente”, onde em Herança chamamos essa atribuição de Herança múltipla, onde também

seria possível o estudante começar a enxergar a Herança estruturada em um papel, já visualizando em uma linguagem e começar a compreender a estrutura por traz da interpretação em sua mente.

Nesta fase, o estudante consegue também investigar a Herança e desenhar claramente em sua mente suas possíveis combinações, em que esta fase dura enquanto a entidade permanece firmemente ligada ao processo de Herança. (SFARD, 1991).

### **3.2.5 Reificação**

Para entrar a fase de reificação Sfard aponta em seus estudos que somente quando uma pessoa se torna capaz de conceber a ideia como um objeto de pleno direito, é dito que o conceito foi retificado. Assim, a reificação é definida como uma mudança ontológica, ou seja a capacidade de enxergar algo familiar em uma “luz” totalmente nova. (SFARD, 1991).

Em Herança, essa “luz” se dá quando o estudante parte do conceito abstrato antes moldado apenas em sua mente e transposto para o papel para outra forma de pensar, não sendo apenas só em desenhos e grafos, em que tal pensamento nada mais é que a linguagem de máquina ou a linguagem “Java”, para um salto quântico instantâneo, como diz Sfard em seus estudos (SFARD, 1991).

Este “salto quântico” pode ser visto no trabalho de Ambrosio (2011) em que seus estudos mostram que a partir desse ponto o estudante tende a compreender totalmente o problema que irá tratar e assim, melhorar seu desempenho na disciplina e esse entendimento já é transposto para o início de outro conceito (AMBRÓSIO, 2011).

Neste sentido, o conceito de reificação é quando o estudante tem total visão para começar a transcrever seus pensamento e grafos para uma linguagem programável, e a partir disso, Sfard (1991) aponta que ao final da reificação do conhecimento é possível começar o processo novamente para adquirir uma outra forma de sabedoria que no caso será o conhecimento em forma operacional (SFARD, 1991).

### **3.3 Elaboração do questionário de identificação e coleta de dados baseado em herança**

O questionário de coleta de dados foi elaborado seguindo alguns procedimentos e levantamentos, visando tirar o máximo de informação dos estudantes, fazendo com que os mesmos conseguissem um pensamento crítico e minucioso para obter um *feedback* satisfatório para a análise dos dados.

Desta forma, a elaboração e formulação das questões deu-se pelo conceito em Herança, em como sua estrutura é formada e de que forma isso poderia ser implementada, assim visando capturar o entendimento dos estudantes e verificar o raciocínio crítico dos mesmos.

Tais procedimentos se deram na estruturação de um questionário semiaberto que possibilita tanto verificar o entendimento dos estudantes num olhar mais subjetivo, como também podendo olhar de uma forma objetiva, assim é possível atingir um campo maior de pensamentos, visões e opiniões que os estudantes têm sobre Herança, onde ao final foram propostas sete questões, as quais se encontram no ANEXO A.

Tais questões foram elaboradas dentro do conceito estrutural e operacional baseado nos conceitos proposto por (SIERRA & BATES, 2005) em seu livro: “Use a Cabeça Java”.

O estudante encontrou desde perguntas abertas e dissertativas até pequenas análises de código que necessitou que ele interpretasse o problema proposto, e até mesmo questões que envolvam a criação de diagramas de UML criados a partir de softwares como “Astah Community” ou qualquer outra ferramenta que o estudante julgasse mais adequado, possibilitando verificar a capacidade com que ele transcrevesse seu pensamento para uma possível forma de estruturação.

Onde cada uma delas o levou a um nível diferente sobre o pensamento em Herança tornando possível verificar até qual nível de pensamento e conhecimento o estudante possui e verificar se o conceito estrutural e operacional foi realmente atingido, e verificar em quais fases o mesmo se encontram.

## 4. DESENVOLVIMENTO

Neste capítulo será apresentado e descrito o desenvolvimento do trabalho.

Foi desenvolvida uma análise para verificar os conhecimentos que são mobilizados por estudantes que cursam a disciplina de programação II do curso de Sistema de Informação na Universidade Estadual do Norte do Paraná, campus Luiz Meneghel.

### 4.1 DESENVOLVIMENTO E ELABORAÇÃO DA FERRAMENTA DE COLETA DE DADOS

Devido à necessidade de coletar um número relativamente grande de dados e que os mesmos pudessem ser maximizados e alocados de uma forma que poderia facilitar os estudos propostos nesse trabalho, foram elaboradas algumas formas de coleta de dados onde ao final à proposta de criar uma página *web*, que se tornou a melhor alternativa, pois além de dispor de mecanismos que tem a função de guardar os dados facilmente ainda possibilitou o salvamento de imagens que para as análises possibilita uma interpretação mais aguçada sobre o assunto.

Para a construção da ferramenta foi utilizado NetBeans versão 8.0.2 que é um ambiente integrado e gratuito para desenvolver softwares nas linguagens Java, C, C++, entre muitas outras. Onde por uso da ferramenta foi necessário criar um projeto JavaWeb que nada mais é que um sistema para internet onde o sistema necessita ser rodado em um servidor do tipo contêiner.

Em seguida foi utilizado o GlassFish que é um servidor de aplicação que suporta todas as especificações da API Java e por ser mais robusto e por possuir muitas bibliotecas já prontas, conteúdo por ser mais robusto ele necessita de mais memória.

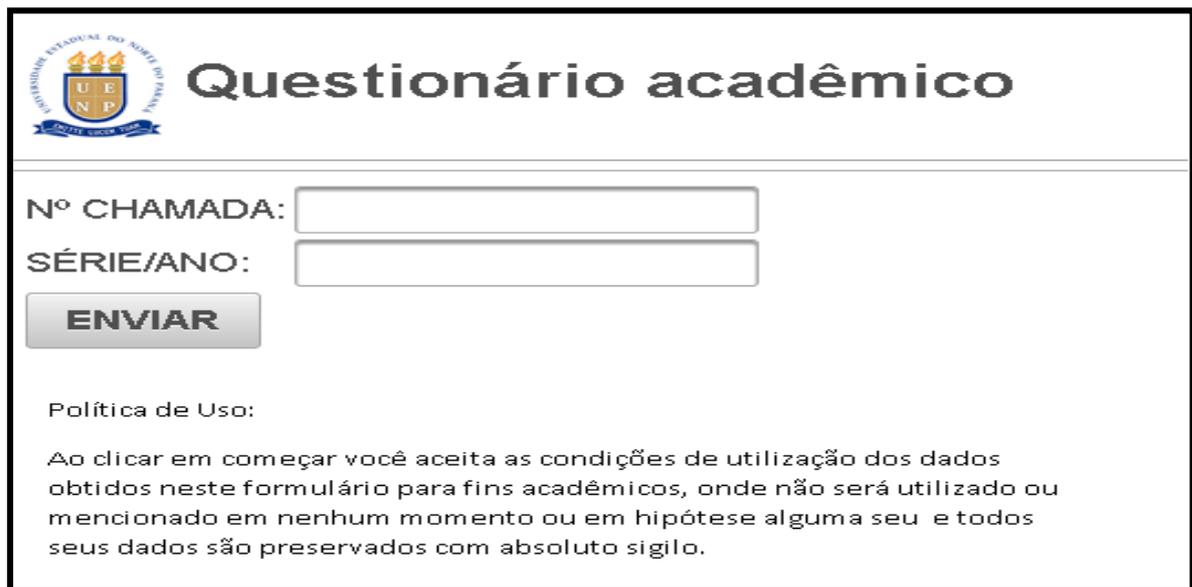
Na parte de infraestrutura o sistema está rodando em um servidor Debian/Linux, e dentro do Debian foi instalado o Java 8 e o GlassFish, e para o armazenamento dos dados foi criado o banco de dados usando o PostgreSQL 9.3, onde para fazer a conexão do banco de dados com o Java foi feito através JDBC (Java Database Connectivity) que é um conjunto de classes e interfaces escritas em

Java que fazem o envio de instruções SQL<sup>5</sup> para qualquer banco de dados relacional.

Porém como é uma ferramenta *web* e foi utilizado JavaWeb é preciso utilizar *Framework* para fazer a parte da interface no *Browser* (Navegador), em que foi usado JSF (JavaServer Faces) para construir a interface, as estrutura das páginas foram usados *XHTML* que é uma linguagem de marcação que por ser mais rígidas eventualmente deu menos problemas.

Logo em seguida pensando levar para os estudantes uma interface mais agradável e que atendessem nossas necessidades foi utilizado a biblioteca PrimeFaces que proporciona interfaces ricas de componentes, estilos para tornas a página algo mais agradável como é visto na Figura 1, onde foi necessário apenas adequar o código já pronto na pagina as necessidades da aplicação.

**Figura 1 - Tela Principal**



**Fonte: Autoria Própria**

Já para a coleta de dados foi criado um meio para que as respostas dos usuários fossem alocadas em um documento em formato PDF<sup>6</sup> para facilitar as análises, e para isso foi necessário utilizar outra biblioteca chamada iText junto com

<sup>5</sup> SQL: Structured Query Language, ou Linguagem de Consulta Estruturada, é a linguagem de pesquisa declarativa padrão para banco de dados relacional.

<sup>6</sup> PDF: *Portable Document Format* (Formato Portátil de Documento) é um formato de arquivo criado pela empresa Adobe Systems para que qualquer documento seja visualizado, independente de qual tenha sido o programa que o originou.

algumas outras bibliotecas que o auxiliam , onde na hora que o estudante enviase a resposta, seria criado um arquivo de texto dentro da pasta do servidor e assim é gerado o arquivo.

Com isso os estudantes poderão acessar a página de formulário de qualquer lugar que disponha de internet seja em dispositivos moveis ou desktop. Feito isso bastaria acessar a página de administrador do formulário e coletar as informações adquiridas ou excluir se assim desejar.

## **4.2 Elaboração e Análise dos dados**

As questões P1 à P5 são de nível estrutural, pois se referem à um âmbito mais voltado a teoria do conceito de Herança, já as questões P6 e P7 são de nível operacional, pois já tem um viés de programação.

Para a análise das questões e preservar a identidade dos estudantes, inicialmente os mesmos receberam codificações A1, A2, A3, A4,... A13 em seguida foram separados de acordo com a categorização já descrita neste trabalho.

### **4.2.1 Discussão da Questão 1**

Os estudantes A1, A4, A12 e A13, se encontram em um nível de interiorização, pois os mesmos não obtiveram o discernimento para buscar em seu pensamento, métodos ou diferenças significativas que os levassem a diferenciar com clareza o que uma programação procedural tem de diferente de uma programação orientada a objetivo como é vista na Figura 2.

**Figura 2 - Solução do estudante A1**

Pergunta: 0.0

R e s p o s t a :

a Programação orientada ao objeto ela é feita por partes, fazendo a comunicação entre as classes. Dessa forma ela fica mais estruturada

**Fonte: Autoria Própria**

Os estudantes A5, A10 e A11 se encontram em um nível de condensação, pois os mesmos conseguem discernir as diferença entre uma programação procedural com uma programação orientada a objeto e ainda entender métodos e funções que remetem a reutilização de funções, Figura 3.

**Figura 3 - Solução do estudante A11**

Pergunta: 0.0

R e s p o s t a :

Ela possui em destaque a função de reuso dos códigos, onde uma classe pode ser utilizada para várias outras finalidades com funções semelhantes.

**Fonte: Autoria Própria**

Já os estudantes A2, A3, A6, A7, A8 e A9 se encontram no nível de reificação, pois conseguiram apontar e distinguir não só a diferenças entre ambas as programações, bem como evidenciar métodos e funcionalidade, o que remete a levá-los a tal nível de pensamento conforme é visto na Figura 4.

**Figura 4 - Solução do estudante A3**

Pergunta: 0.0

R e s p o s t a :

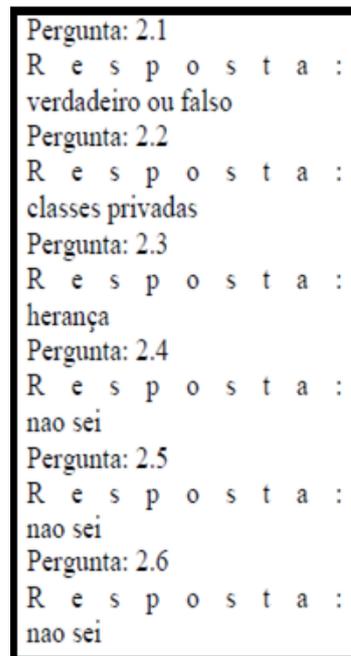
A programação orientada a objeto nos possibilita utilizar os dados de forma inteligente, podendo criar objetos e reutilizá-los em outras classes ou funções. Assim podemos economizar tempo e código utilizando os conhecimentos de herança e herdando dados de outras classes, sendo desnecessário escrevê-los novamente. Já a programação procedural não podemos criar objetos e reutilizá-los.

**Fonte: Autoria Própria**

### 4.2.2 Discussão da Questão 2

Como mostra a Figura 5 a seguir, os estudantes A1 e A2 se encontram em um nível de interiorização, pois os mesmos demonstraram apenas responder de uma forma simplificada sem buscar detalhamento nas questões, o que acreditamos que estes não tem uma clareza no que diz respeito a possuir o entendimento das funcionalidades de programação mais voltado para a área de herança.

**Figura 5 - Solução do estudante A1**



Pergunta: 2.1  
R e s p o s t a :  
verdadeiro ou falso  
Pergunta: 2.2  
R e s p o s t a :  
classes privadas  
Pergunta: 2.3  
R e s p o s t a :  
herança  
Pergunta: 2.4  
R e s p o s t a :  
nao sei  
Pergunta: 2.5  
R e s p o s t a :  
nao sei  
Pergunta: 2.6  
R e s p o s t a :  
nao sei

**Fonte: Autoria Própria**

O estudante A4 se encontra no nível de condensação, pois conforme é demonstrado na Figura 6 o mesmo demonstra um entendimento e uma compreensão sobre a maioria das subquestões, porem ainda não demonstra uma clareza em sua descrição para avançar para o processo seguinte.

**Figura 6 - Solução do estudante A4**

Pergunta: 2.1  
 R e s p o s t a :  
 Verdadeiro ou falso  
 Pergunta: 2.2  
 R e s p o s t a :  
 define variavel ou classes privadas  
 Pergunta: 2.3  
 R e s p o s t a :  
 recebe uma herança de outra classe  
 Pergunta: 2.4  
 R e s p o s t a :  
 não sei  
 Pergunta: 2.5  
 R e s p o s t a :  
 classe pai  
 Pergunta: 2.6  
 R e s p o s t a :  
 conta uma determinada quantidade de caracteres

**Fonte: Autoria Própria**

Já os estudantes A3, A5, A6, A7, A8, A9, A10, A11, A12 e A13 é possível verificar observando a Figura 7 que estão no nível de reificação, pois nota-se em todas as subquestões que além de entenderem o significado dos comandos que estão representados, buscam dar maiores detalhes tendo assim uma explicação mais aguçada e minuciosa.

**Figura 7 - Solução do estudante A6**

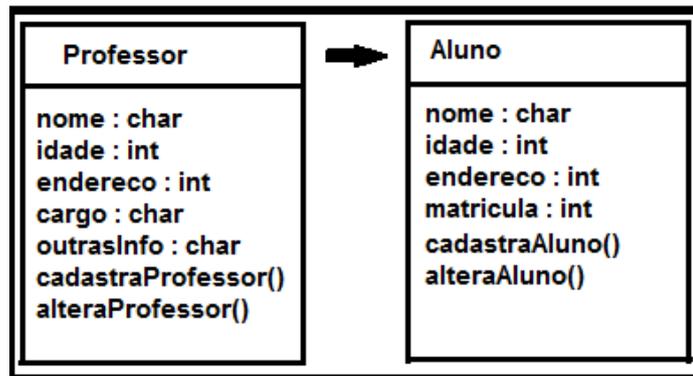
Pergunta: 2.1  
 R e s p o s t a :  
 No caso de BOOLEAN ser um retorno de um método, significa que esse método irá retornar sempre TRUE ou FALSE  
 Pergunta: 2.2  
 R e s p o s t a :  
 os métodos PRIVATE são de acesso exclusivos do Objeto, não podendo ser chamados de fora.  
 Pergunta: 2.3  
 R e s p o s t a :  
 Essa técnica permite através de Herança, estender os Atributos e métodos de um objeto para outro  
 Pergunta: 2.4  
 R e s p o s t a :  
 Static é pode ser um Objeto ou variável que pode ser chamada de qualquer local do programa, pois quando ele é estanciado se torna fixo(estático) para todo o sistema.  
 Pergunta: 2.5  
 R e s p o s t a :  
 Super, é utilizado em Herança, quando pretende-se utilizar um método/variável na classes pai, que possui o mesmo nome nas classes pai e filho.  
 Pergunta: 2.6  
 R e s p o s t a :  
 é um método, da variável complexa STRING que permite contar a quantidade de caracteres de uma variável desse tipo.

**Fonte: Autoria Própria**

### 4.2.3 Discussão da Questão 3

Observando a Figura 8 é possível verificar que o estudante A12 demonstrou que sua forma de pensar remete ao menor nível de interiorização devido ao fato de não obter o discernimento para distinguir que tanto “aluno” e “professor” poderiam partilhar de características em comuns.

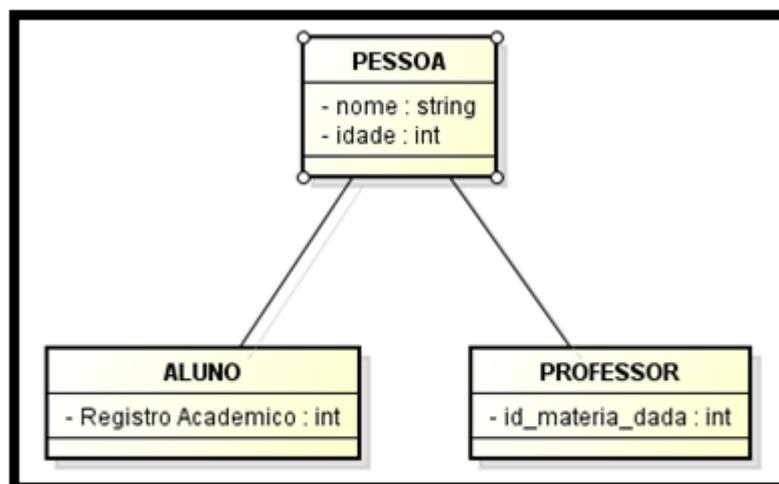
Figura 8 - Solução do estudante A12



Fonte: Autoria Própria

Já os estudantes A5, A6, A7, A8 e A10 estão em um nível de condensação, pois é nítida a diferença e o modo de pensar do estudante A12, onde os mesmos demonstram total entendimento e distinção de características distintas e semelhantes que o problema os levou a pensarem, Figura 9.

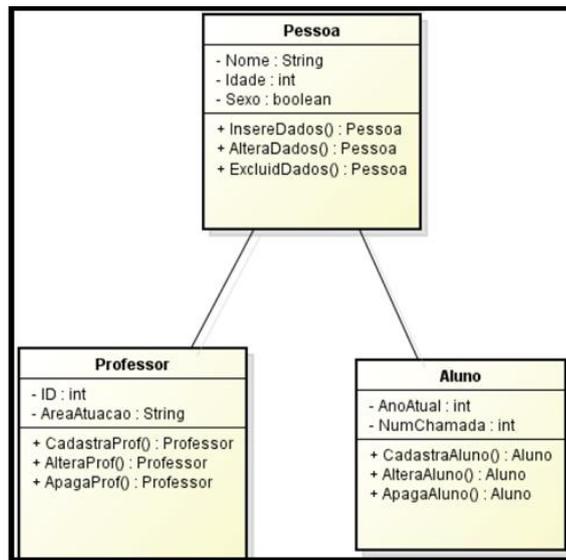
Figura 9 - Solução do estudante A5



Fonte: Autoria Própria

Porém os estudantes A1, A2, A3, A4, A9, A 11 e A13 se encontram em um nível de reificação, pois a diferença de pensamento que é possível notar nestas análises é a forma de pensar com que eles minuciosamente introduzem detalhes e métodos em seus problemas. Tais informações os levam a entrar mais rapidamente em um processo operacional por seu pensamento estar rico em detalhes, o que facilita a transição de um processo para o outro, Observe a Figura 10.

**Figura 10 - Solução do estudante A9**

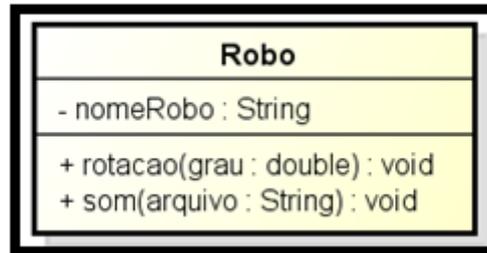


**Fonte: Autoria Própria**

#### 4.2.4 Discussão da Questão 4

A questão número 4 embora muito semelhante à questão anterior, mostrou grandes desafios no que diz respeito à interpretação como é visto na Figura 11 logo abaixo, pois alguns estudantes mostraram não entender o que o problema estava querendo tratar, logo pode-se inferir que nessa questão em especial os estudantes A6, A7 e A8 não conseguiram alcançar o nível de interiorização, pois não conseguiram discernir as características chaves que constituem a herança.

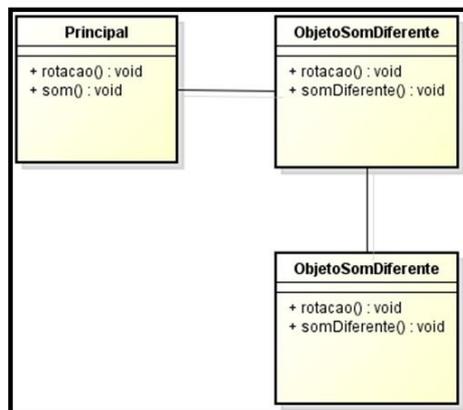
**Figura 11 - Solução do estudante A7**



**Fonte: Autoria Própria**

Os estudantes A1, A11 e A12 embora alguns passos mais adiante em sua forma de pensar, encontraram-se em um nível de interiorização conforme a Figura 12, pois embora compreendessem o que o problema estava buscando tratar, não conseguiram assimilar as ramificações que eram propostas a eles, contudo como pode ser verificado em nossos estudos, a maneira de buscar uma contrapartida para reutilizar comando que favoreçam o problema, os leva a interiorizar o conteúdo e possivelmente uma possível transição para o processo de condensação.

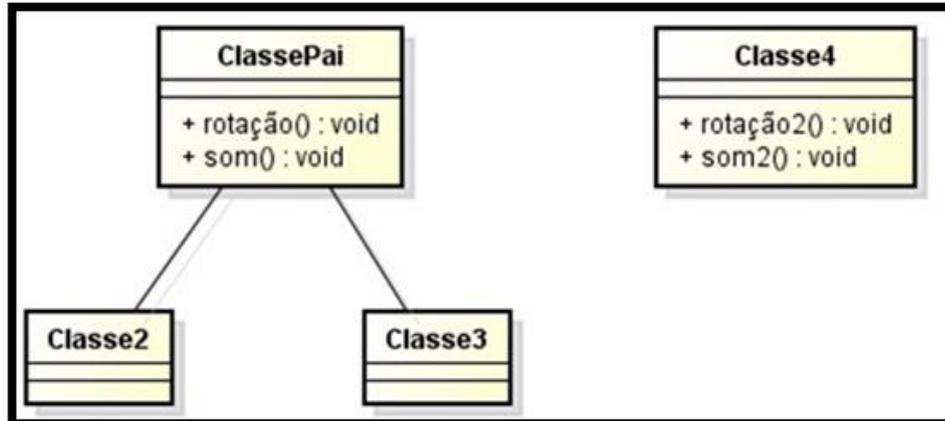
**Figura 12 - Solução do estudante A12**



**Fonte: Autoria Própria**

Dando ênfase no que foi dito anteriormente, nota-se que os estudantes A3 e A4, diferentemente dos anteriores, apresentaram uma resposta relativamente significativa do que esperamos conseguir, logo, seu pensamento embora meio confuso os levasse a um nível de condensação, o que pode ocorrer sem o seu próprio consentimento, conforme é visto na Figura 13.

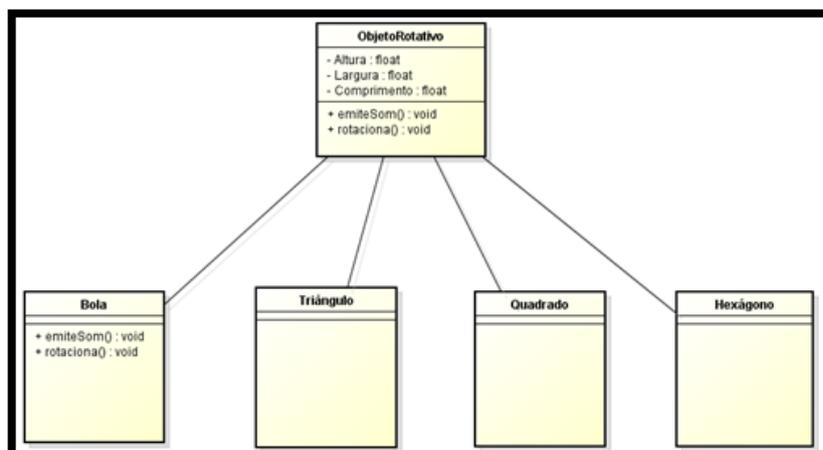
Figura 13 - Solução do estudante A3



Fonte: Autoria Própria

Ainda voltado com o olhar para a questão 4, é possível verificar que os estudantes A2, A5, A9, A10 e A13 estão no nível de reificação, pois diferente dos demais, estes demonstram uma interpretação muito ampla, onde conseguem compreender claramente o que é proposto e a melhor forma de demonstrar conforme é visto na Figura 14, onde com o resultado obtido é possível que os demais estudantes consigam trazer essa maneira de pensar que está constituída nos gráficos para um novo processo ou uma nova aprendizagem, levando-os para uma transição do pensamento estrutural para um processo operacional de sua estrutura.

Figura 14 - Solução do estudante A9

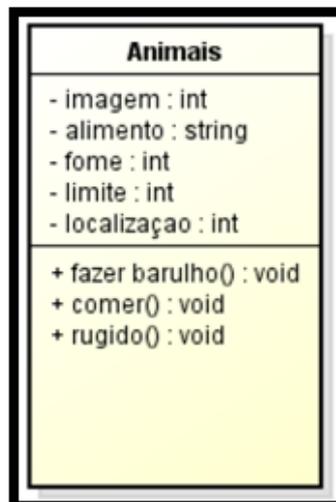


Fonte: Autoria Própria

#### 4.2.5 Discussão da Questão 5

A questão de número 5 foi elaborada de uma forma que levasse os estudantes a estar em um nível de pensamento bem mais estruturado e crítico para tratar o que era pretendido, contudo observando a Figura 15 nota-se que os estudantes A1, A4, A6, A7 e A10 não se encontram nem no menor nível, pois não há sequer traços que indiquem o contexto básico da pergunta, o que se pode evidenciar que os mesmos não obtiveram uma interpretação concreta do que era previsto, tornando seu resultado em um processo falho, no que diz respeito ao nível de processo estrutural no contexto de herança.

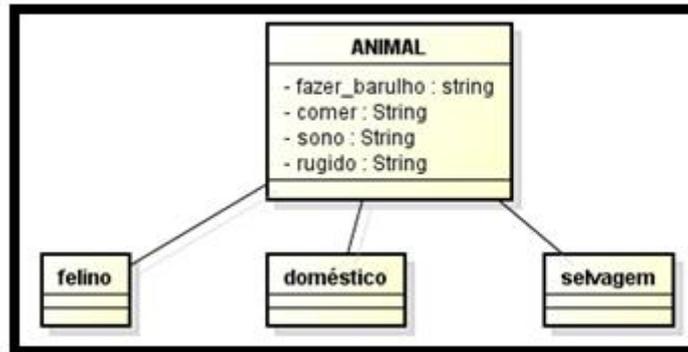
Figura 15 - Solução do estudante A1



Fonte: Autoria Própria

Contudo diferente dos estudantes anteriores, o estudante A5 demonstra estar em um nível de interiorização, pois embora não demonstre uma capacidade relativamente alta para interpretar o problema, demonstrou noções sobre herança e um entendimento que demonstra características distintas que alguns animais tinham em comum com os outros conforme é visto na Figura 16, podendo assim classificar e construir um modelo que facilite seu desenvolvimento para uma nova estrutura.

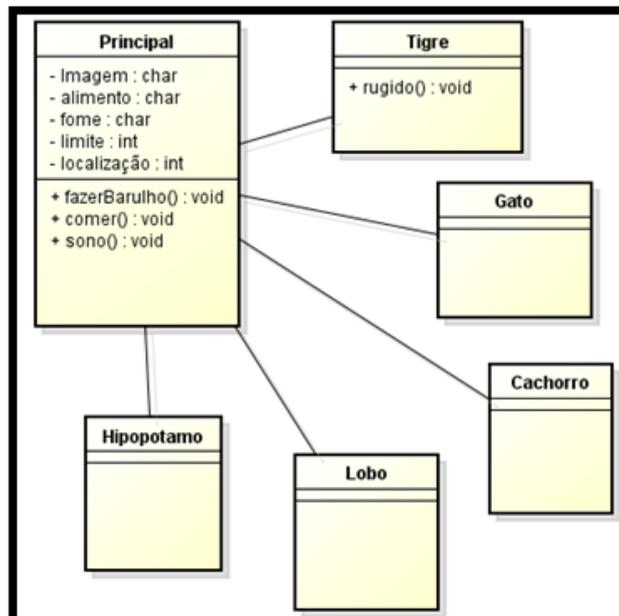
Figura 16 - Solução do estudante A5



Fonte: Autoria Própria

Já os estudantes A2, A3, A8 e A12 se encontram em um nível de condensação, pois suas interpretações levaram a tratar o problema de forma a estrutura-los e a reutilizar métodos e atributos fazendo com que ao ser implementados, torne possível tratar e verificar o contexto de seu pensamento de uma maneira mais evidente e de fácil interpretação como é visto na Figura 17.

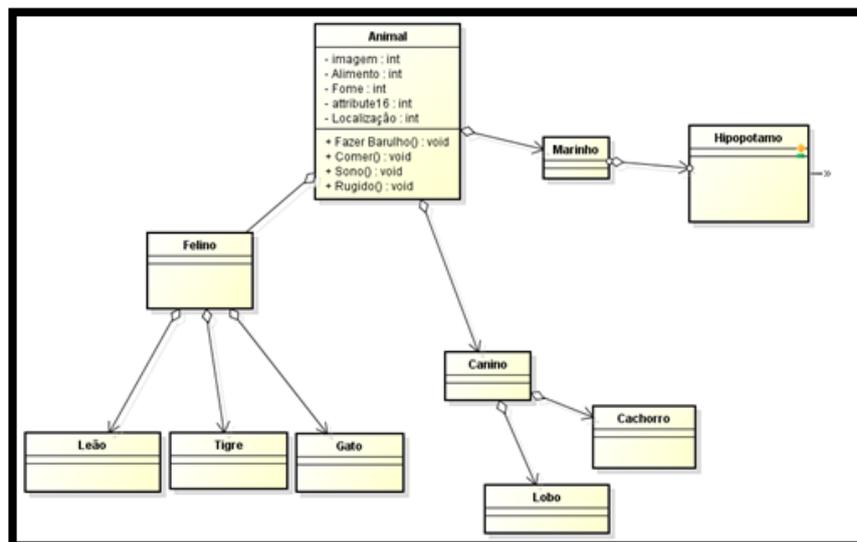
Figura 17 - Solução do estudante A12



Fonte: Autoria Própria

Os estudantes A9, A11 e A13 demonstram que seu pensamento voltado a esta questão em especial se constitui em um nível de reificação conforme se observa na Figura 18, pois diferente dos demais conseguiram discernir as características distintas que os animais possuem e classificá-los em grupos, o que demonstra uma mudança ontológica que como cita Sfard (1991) é a capacidade de enxergar algo familiar em uma luz totalmente nova, mostrando assim uma interpretação aguçada e um salto na passagem de um processo estrutural para um operacional.

**Figura 18 - Solução do estudante A13**



**Fonte: Autoria Própria**

### 4.3.1 Processo Operacional

As questões a seguir, que são denominadas de P6 e P7, são questões voltadas para um processo operacional, onde objetivou-se levar os estudantes que já tinham seu pensamento transposto de uma forma visível que não necessariamente necessite ser uma linguagem de máquina para uma forma mais prática, onde tais pensamentos serão voltados para uma forma de se pensar como um computador e como seus comandos e estruturas tende a trabalhar em conjunto.

### 4.3.2 Discussão da Questão 6

Observando a Figura 19, verificou-se que os estudantes A5, A6 e A11 não conseguiram concluir nenhum resultado positivo ou mesmo algo que chegasse perto do que poderíamos inferir como satisfatório para o menor nível de processo, pois metades das alternativas não foram feitas por motivos que não cabe a este trabalho analisar, mas que se pode inferir a partir disso é que tais estudantes não possuem uma interpretação que os levassem a entender o problema, quanto ao entendimento e prática no processo operacional.

**Figura 19 - Solução do estudante A5**

<b>Pergunta: 6.0</b>	<b>Pergunta: 7.3</b>
Resposta:	Resposta:
<b>Pergunta: 6.1</b>	<b>Pergunta: 7.4</b>
Resposta:	Resposta:
<b>Pergunta: 6.2</b>	<b>Pergunta: 7.5</b>
Resposta:	Resposta:
<b>Pergunta: 6.3</b>	<b>Pergunta: 7.6</b>
Resposta:	Resposta:
<b>Pergunta: 6.4</b>	<b>Pergunta: 7.7</b>
Resposta:	Resposta:
<b>Pergunta: 6.5</b>	<b>Pergunta: 7.8</b>
Resposta:	Resposta:
<b>Pergunta: 6.6</b>	<b>Pergunta: 7.9</b>
Resposta:	Resposta:
<b>Pergunta: 6.7</b>	<b>Pergunta: 8.0</b>
Resposta:	Resposta:
<b>Pergunta: 6.8</b>	<b>Pergunta: 8.1</b>
Resposta:	Resposta:
<b>Pergunta: 6.9</b>	<b>Pergunta: 8.2</b>
Resposta:	Resposta:
<b>Pergunta: 7.0</b>	<b>Pergunta: 8.3</b>
Resposta:	Resposta:
<b>Pergunta: 7.1</b>	<b>Pergunta: 8.4</b>
Resposta:	Resposta:
<b>Pergunta: 7.2</b>	<b>Pergunta: 8.5</b>
Resposta:	Resposta:

**Fonte: Autoria Própria**

Porém tentando alcançar o pensamento operacional dos estudantes, os estudantes A1 e A4 embora não muito avançados no contexto operacional, obtiveram o mínimo de acertos para entrar no nível de interiorização como

demonstra a Figura 20, o que mostra pelo menos uma interpretação e análise dos comandos que foram dados como problema especificamente nesta questão.

**Figura 20 - Solução do estudante A1**

<b>Pergunta: 6.0</b>	<b>Pergunta: 7.3</b>
Resposta:	Resposta:
extends	<b>Pergunta: 7.4</b>
<b>Pergunta: 6.1</b>	Resposta:
Resposta:	<b>Pergunta: 7.5</b>
<b>Pergunta: 6.2</b>	Resposta:
Resposta:	<b>Pergunta: 7.6</b>
<b>Pergunta: 6.3</b>	Resposta:
Resposta:	<b>Pergunta: 7.7</b>
<b>Pergunta: 6.4</b>	Resposta:
Resposta:	<b>Pergunta: 7.8</b>
<b>Pergunta: 6.5</b>	Resposta:
Resposta:	<b>Pergunta: 7.9</b>
<b>Pergunta: 6.6</b>	<b>Pergunta: 8.0</b>
Resposta:	Resposta:
<b>Pergunta: 6.7</b>	<b>Pergunta: 8.1</b>
Resposta:	Resposta:
<b>Pergunta: 6.8</b>	<b>Pergunta: 8.2</b>
Resposta:	Resposta:
<b>Pergunta: 6.9</b>	<b>Pergunta: 8.3</b>
Resposta:	Resposta:
<b>Pergunta: 7.0</b>	<b>Pergunta: 8.4</b>
Resposta:	Resposta:
<b>Pergunta: 7.1</b>	<b>Pergunta: 8.5</b>
Resposta:	Resposta:
<b>Pergunta: 7.2</b>	
Resposta:	

**Fonte: Autoria Própria**

Já com uma interpretação mais abrangente vemos que os estudantes A2, A3, A8, A10 e A12 estão em um processo de condensação, pois diferente dos demais estudantes, suas margens de respostas corretas são maiores e mais precisas do que se pode levar em consideração. Desta forma observando a Figura 21 nota-se que estes estudantes sabem o significado e a aplicação dos elementos que constituem a herança e onde devem ser devidamente colocados.

Figura 21 - Solução do estudante A2

Pergunta: 6.0	Pergunta: 7.3
Resposta: extends	Resposta: Boat
Pergunta: 6.1	Pergunta: 7.4
Resposta: Boat	Resposta: Boat
Pergunta: 6.2	Pergunta: 7.5
Resposta: void	Resposta: setLength
Pergunta: 6.3	Pergunta: 7.6
Resposta: b1	Resposta: setLength
Pergunta: 6.4	Pergunta: 7.7
Resposta: public	Resposta: testBoats
Pergunta: 6.5	Pergunta: 7.8
Resposta: rowTheBoat	Resposta: Saiboat
Pergunta: 6.6	Pergunta: 7.9
Resposta: intb1	Resposta: extends
Pergunta: 6.7	Pergunta: 8.0
Resposta: static	Resposta: static
Pergunta: 6.8	Pergunta: 8.1
Resposta: stroke	Resposta: void
Pergunta: 6.9	Pergunta: 8.2
Resposta: intb1	Resposta: Natasha
Pergunta: 7.0	Pergunta: 8.3
Resposta: intb2	Resposta: Pergunta: 8.4
Pergunta: 7.1	Resposta:
Resposta: intb3	Pergunta: 8.5
Pergunta: 7.2	Resposta:
Resposta: Int	

Fonte: Autoria Própria

Os estudantes A7 e A9 se encontram em um nível de reificação, pois praticamente em quase todas as respostas que foram apresentadas a eles, os mesmos demonstraram uma capacidade de interpretação e a visão dos pontos de aplicação e dos termos de herança, contudo visto que seu pensamento se encontra no maior nível, podemos inferir que os estudantes possuem um raciocínio mais crítico e detalhista, que os leva a ter antes de concluir determinada ação, passarem pelas fases de desenvolvimento como AMBRÓSIO (2011) cita em seu trabalho denominado “Programação de computadores: Compreender as dificuldades de aprendizagem dos alunos”, como é observado na Figura 22.

Figura 22 - Solução do estudante A9

Pergunta: 6.0	Pergunta: 7.2
Resposta: extends	Resposta: Saiboat
Pergunta: 6.1	Pergunta: 7.3
Resposta: Boat	Resposta: Saiboat
Pergunta: 6.2	Pergunta: 7.4
Resposta: void	Resposta: b2, b3
Pergunta: 6.3	Pergunta: 7.5
Resposta: length	Resposta: move
Pergunta: 6.4	Pergunta: 7.6
Resposta: public	Resposta: move
Pergunta: 6.5	Pergunta: 7.7
Resposta: setLength	Resposta: b2
Pergunta: 6.6	Pergunta: 7.8
Resposta: intlen	Resposta: Saiboat
Pergunta: 6.7	Pergunta: 7.9
Resposta: void	Resposta: extends
Pergunta: 6.8	Pergunta: 8.0
Resposta: moveu	Resposta: void
Pergunta: 6.9	Pergunta: 8.1
Resposta: public	Resposta: move
Pergunta: 7.0	Pergunta: 8.2
Resposta: static	Resposta:
Pergunta: 7.1	Pergunta: 8.3
Resposta: void	Resposta:
	Pergunta: 8.4
	Resposta:
	Pergunta: 8.5
	Resposta:

Fonte: Autoria Própria

### 4.3.3 Discussão da Questão 7

A questão P7 foi criada no intuito de levar para os estudantes não somente os comandos de programação, mas tentar fazer com que os mesmos tivessem o entendimento do contexto e do que tais comandos e suas características fariam atribuídos àquelas cadeias de códigos. Contudo conforme as análises feitas notou-se que os estudantes A1 e A2 não se encontram em um nível de processo aceitável para classificá-los como interiorização como é visto na Figura 23, pois dentro das análises coletadas ambos não mostraram compreender as cadeias de comandos impostas a eles e o que a pergunta realmente buscava.

**Figura 23 - Solução do estudante A1**

<b>Pergunta: 7.0</b>	<b>Pergunta: 8.0</b>
Resposta:	Resposta:
Os eletrodomésticos serão cadastrados aqui.	<b>Pergunta: 8.1</b>
<b>Pergunta: 7.1</b>	Resposta:
Resposta:	<b>Pergunta: 8.2</b>
<b>Pergunta: 7.2</b>	Resposta:
Resposta:	classe publica onde sera instanciada os objetos e itens dos eletrodomésticos
esta criando uma classe eletrodoméstico onde vai conter as informações dos itens	<b>Pergunta: 8.3</b>
<b>Pergunta: 7.3</b>	Resposta:
Resposta:	<b>Pergunta: 8.4</b>
vai verificar se o eletrodoméstico esta ligado	Resposta:
<b>Pergunta: 7.4</b>	<b>Pergunta: 8.5</b>
Resposta:	Resposta:
variável que informa um numero, se é 110 ou 220	<b>Pergunta: 8.6</b>
<b>Pergunta: 7.5</b>	Resposta:
Resposta:	<b>Pergunta: 8.7</b>
variável float que vai indicar o consumo do eletrodoméstico	Resposta:
<b>Pergunta: 7.6</b>	<b>Pergunta: 8.8</b>
Resposta:	Resposta:
faz a verificação do eletrodoméstico instancia o objeto e recebe os dados citados	recebe o dado e altera
<b>Pergunta: 7.7</b>	<b>Pergunta: 8.9</b>
Resposta:	Resposta:
recebe o valor e altera	recebe o dado e altera
<b>Pergunta: 7.8</b>	<b>Pergunta: 9.0</b>
Resposta:	Resposta:
recebe o valor e altera	recebe o dado e altera
<b>Pergunta: 7.9</b>	
Resposta:	
recebe o valor e altera	

**Fonte: Autoria Própria**

Já os estudantes A3, A5, A8 e A13 demonstram em alguns casos uma análise embora simples conforme é possível observar na Figura 24, mas que se pode consolidar o pequeno entendimento das características em herança que foram colocadas para ser analisadas, sendo assim os mesmos encontram-se em um nível de interiorização, pois embora consigam discernir e entender conceitos básicos, os conceitos voltados para herança não são explicado de forma clara e objetiva.

Figura 24 - Solução do estudante A3

<b>Pergunta: 7.0</b>	<b>Pergunta: 8.0</b>
Resposta:	Resposta:
É uma classe eletrodoméstico	É uma classe TV
<b>Pergunta: 7.1</b>	<b>Pergunta: 8.1</b>
Resposta:	Resposta:
Pacote tiexpert	Pacote tiexpert
<b>Pergunta: 7.2</b>	<b>Pergunta: 8.2</b>
Resposta:	Resposta:
Classe pública Eletrodoméstico	Classe pública TV que herda dados de Eletrodoméstico
<b>Pergunta: 7.3</b>	<b>Pergunta: 8.3</b>
Resposta:	Resposta:
Atributo boleano e pública com o nome ligado	Atributo privado inteiro canal
<b>Pergunta: 7.4</b>	<b>Pergunta: 8.4</b>
Resposta:	Resposta:
Atributo do tipo inteiro com o nome voltagem	Atributo privado inteiro volume
<b>Pergunta: 7.5</b>	<b>Pergunta: 8.5</b>
Resposta:	Resposta:
Atributo do tipo inteiro com o nome consumo	Atributo privado inteiro tamanho
<b>Pergunta: 7.6</b>	<b>Pergunta: 8.6</b>
Resposta:	Resposta:
Objeto Eletrodoméstico que passa por parâmetro os atributos boleano e dois inteiros	Objeto TV que recebe como parâmetro atributos inteiros
<b>Pergunta: 7.7</b>	<b>Pergunta: 8.7</b>
Resposta:	Resposta:
Ligado atualiza conteúdo recebendo ligado	Chamando dados da classe pai
<b>Pergunta: 7.8</b>	<b>Pergunta: 8.8</b>
Resposta:	Resposta:
Voltagem atualiza conteúdo recebendo voltagem	canal atualiza conteúdo recebendo canal
<b>Pergunta: 7.9</b>	<b>Pergunta: 8.9</b>
Resposta:	Resposta:
Consumo atualiza conteúdo recebendo consumo	volume atualiza conteúdo recebendo volume
	<b>Pergunta: 9.0</b>
	Resposta:
	tamanho atualiza conteúdo recebendo tamanho

Fonte: Autoria Própria

Entretanto, os estudantes A4, A10 e A12 conseguiram atingir níveis de pensamento satisfatórios, seja na forma de pensar ou de captar determinados comandos que foram impostos a eles e transcrever em uma forma de escrita, possibilitando classificar tais estudantes em um nível de condensação, observando a Figura 25 é possível verificar que os estudantes demonstram não apenas conhecer a prática da linguagem, mas toda a teoria incorporada a ela.

Figura 25 - Solução do estudante A12

<b>Pergunta: 7.0</b>	<b>Pergunta: 8.0</b>
Resposta:	Resposta:
Possui as características	Criando uma classe TV
<b>Pergunta: 7.1</b>	<b>Pergunta: 8.1</b>
Resposta:	Resposta:
Pacote onde estão localizadas as classes	Pacote
<b>Pergunta: 7.2</b>	<b>Pergunta: 8.2</b>
Resposta:	Resposta:
Declarando a classe Eletrodomestico	A classe tv esta herdando da classe eletrodomestico
<b>Pergunta: 7.3</b>	<b>Pergunta: 8.3</b>
Resposta:	Resposta:
Retorna se o objeto esta ligado ou não	Declarando uma variável privada que aceita um valor interio canal
<b>Pergunta: 7.4</b>	<b>Pergunta: 8.4</b>
Resposta:	Resposta:
Retorna a o voltagem do eletrodomestico	Declarando uma variável privada que aceita um valor interio volume
<b>Pergunta: 7.5</b>	<b>Pergunta: 8.5</b>
Resposta:	Resposta:
Método privado que poderá ser acessado por getter e setter retornando o valor do consumo	Declarando uma variável privada que aceita um valor interio tamanho
<b>Pergunta: 7.6</b>	<b>Pergunta: 8.6</b>
Resposta:	Resposta:
Recebe como entrada se esta ligado ou não, a voltagem e o consumo	Recebe como entrada a voltagem, o sonsumo, o canal, o volume e o tamanho
<b>Pergunta: 7.7</b>	<b>Pergunta: 8.7</b>
Resposta:	Resposta:
Inicializando o atributo ligado.	Chamando o construtor da classe superior
<b>Pergunta: 7.8</b>	<b>Pergunta: 8.8</b>
Resposta:	Resposta:
Inicializando o atributo voltagem.	Inicializando o atributo canal
<b>Pergunta: 7.9</b>	<b>Pergunta: 8.9</b>
Resposta:	Resposta:
Inicializando o atributo consumo	Inicializando o atributo volume
	<b>Pergunta: 9.0</b>
	Resposta:
	Inicializando o atributo tamanho

Fonte: Autoria Própria

Para finalizar nossas análises, a Figura 26 demonstra que os estudantes A6, A7, A9 e A11 demonstraram clareza, interpretação na questão abordada, levando-os a se encontrar no nível de reificação, pois embora não tão diferentes do nível anterior, o ato de buscar trazer os conceitos práticos para uma forma teórica que demonstre não apenas um entendimento, mas um domínio do que estão trabalhando os levaram seu pensar a um nível de processo maior, pois como AMBRÓSIO (2011) aponta em seu estudo denominado “Programação de computadores: Compreender as Dificuldades de Aprendizagem dos Alunos”, um maior taxa de sucesso requer três etapas: interpretação, estruturação e desenvolvimento, os quais estes estudantes mostraram obter e trazer isso de uma forma visível.

Figura 26 - Solução do estudante A6

<b>Pergunta: 7.0</b> Resposta: nome da classe e o tipo dela	<b>Pergunta: 8.0</b> Resposta: nome da classe e o tipo dela
<b>Pergunta: 7.1</b> Resposta: criação de um pacote	<b>Pergunta: 8.1</b> Resposta: criação de um pacote
<b>Pergunta: 7.2</b> Resposta: criação de uma classe com o nome Eletrodoméstico	<b>Pergunta: 8.2</b> Resposta: criação de uma classe TV que herda a classe ELETRODOMESTICO.
<b>Pergunta: 7.3</b> Resposta: criação de um atributo do tipo booleano, que armazena o conteúdo TRUE ou FALSE. Esse atributo/variável é pública ou seja, pode ser utilizada por qualquer parte do método.	<b>Pergunta: 8.3</b> Resposta: criação de uma variável com nome CANAL do tipo INTEIRO.
<b>Pergunta: 7.4</b> Resposta: criação de uma variável com nome VOLTAGEM do tipo INTEIRO, que só pode ser utilizada pelo método, por ser privada.	<b>Pergunta: 8.4</b> Resposta: criação de uma variável com nome VOLUME do tipo INTEIRO.
<b>Pergunta: 7.5</b> Resposta: criação de uma variável com nome CONSUMO do tipo INTEIRO, que só pode ser utilizada pelo método, por ser privada.	<b>Pergunta: 8.5</b> Resposta: criação de uma variável com nome TAMANHO do tipo INTEIRO.
<b>Pergunta: 7.6</b> Resposta: criação de um método com nome ELETRODOMESTICO, que recebe como entrada alguns parâmetros.	<b>Pergunta: 8.6</b> Resposta: criação de um método com nome ELETRODOMESTICO, que recebe como entrada alguns parâmetros.
<b>Pergunta: 7.7</b> Resposta: esse código está dizendo que a variável da classe FILHO está recebendo um conteúdo.	<b>Pergunta: 8.7</b> Resposta: chamada de um método, passando parâmetros
<b>Pergunta: 7.8</b> Resposta: esse código está dizendo que a variável da classe FILHO está recebendo um conteúdo.	<b>Pergunta: 8.8</b> Resposta: esse código está dizendo que a variável da classe FILHO está recebendo um conteúdo.
<b>Pergunta: 7.9</b> Resposta: esse código está dizendo que a variável da classe FILHO está recebendo um conteúdo.	<b>Pergunta: 8.9</b> Resposta: esse código está dizendo que a variável da classe FILHO está recebendo um conteúdo.
	<b>Pergunta: 9.0</b> Resposta: esse código está dizendo que a variável da classe FILHO está recebendo um conteúdo

Fonte: Autoria Própria

Feito as análises, construímos dois quadros possibilitando mostrar a relação entre os estudantes e as fases conforme as questões respondidas, que ficaram evidentes durante estas análises, tanto nas questões com concepção estrutural quanto na concepção operacional. Os dados do Quadro 1 referem-se às questões, como por exemplo, a questão 2 possui três itens, Interiorização, Condensação e Reificação, onde é possível verificar quantos estudantes estão nos níveis mencionados anteriormente e em qual processo foi abordado.

	Questões com concepção Estrutural				Questões com concepção Operacional			
	Interiorização	Condensação	Reificação	Nenhuma	Interiorização	Condensação	Reificação	Nenhuma
<b>P1</b>	4	3	6					
<b>P2</b>	2	1	10					
<b>P3</b>	1	5	7					
<b>P4</b>	3	2	5	3				
<b>P5</b>	1	4	3	5				
<b>P6</b>					2	5	2	3
<b>P7</b>					4	3	4	2

Tabela 1 - Quadro Comparativo das análises dos estudantes

Pelo quadro, podemos inferir que, com relação às questões de concepção estrutural, os estudantes obtiveram melhores resultados devido à maioria estar em um processo de reificação, diferente das questões operacionais que poucos conseguiram atingir tal feito.

O quadro da tabela 2 demonstra detalhadamente as etapas que cada estudante se encontra ao passar por cada questão, possibilitando desta forma verificar seu pensamento em cada processo seja ele operacional ou estrutural.

Questões com concepção Estrutural						Questões com concepção Operacional	
Estudantes	P1	P2	P3	P4	P5	P6	P7
<b>A1</b>	Interiorização	Interiorização	Reificação	Interiorização	Não se Aplica	Interiorização	Não se Aplica
<b>A2</b>	Reificação	Interiorização	Reificação	Reificação	Condensação	Condensação	Não se Aplica
<b>A3</b>	Reificação	Reificação	Reificação	Condensação	Condensação	Condensação	Interiorização
<b>A4</b>	Interiorização	Condensação	Reificação	Condensação	Não se Aplica	Interiorização	Condensação
<b>A5</b>	Condensação	Reificação	Condensação	Reificação	Interiorização	Não se Aplica	Interiorização
<b>A6</b>	Reificação	Reificação	Condensação	Não se Aplica	Não se Aplica	Não se Aplica	Reificação
<b>A7</b>	Reificação	Reificação	Condensação	Não se Aplica	Não se Aplica	Reificação	Reificação
<b>A8</b>	Reificação	Reificação	Condensação	Não se Aplica	Condensação	Condensação	Interiorização
<b>A9</b>	Reificação	Reificação	Reificação	Reificação	Reificação	Reificação	Reificação
<b>A10</b>	Condensação	Reificação	Condensação	Reificação	Não se Aplica	Condensação	Condensação
<b>A11</b>	Condensação	Reificação	Reificação	Interiorização	Reificação	Não se Aplica	Reificação
<b>A12</b>	Interiorização	Reificação	Interiorização	Interiorização	Condensação	Condensação	Condensação
<b>A13</b>	Interiorização	Reificação	Reificação	Reificação	Reificação	Reificação	Interiorização

**Tabela 2 - Quadro Comparativo dos estudantes em cada fase**

Após essa análise, foi possível verificar que em relação às questões de concepção estrutural, as fases mais evidenciadas foram as de condensação e reificação, pois os estudantes conseguiram transcrever seu pensamento para uma forma visível, onde através deste pensamento e teorias aplicadas aos outros conceitos, obtiveram o discernimento para definir em muitos casos o conceito aplicado.

Já nas questões com concepção operacional, os estudantes mostraram dificuldades para entender e interpretar a teoria aplicada a uma linguagem de máquina, em que os mesmos demonstraram pular as fases citadas por AMBRÓSIO

(2011), onde nos remete a concluir que o conceito operacional pode ser algo ainda mal assimilado pelos estudantes, por se tratar de algo que requer estruturas mentais mais concretas.

## 5. CONCLUSÕES

Com base nas análises dos dados produzidos pela aplicação do questionário, pode-se observar que ao responderem questões de caráter estrutural, a maior partes dos estudantes mostrou não ter grandes dificuldades pois a maioria conseguiu explicar de forma significativa o que estava ocorrendo nestas questões e assim conseguimos observar os processos de condensação e reificação.

No entanto, ao responderem questões de caráter operacional, pode-se verificar que a maioria atingiu apenas os níveis de interiorização e condensação.

Tornando claro que a busca pelo processo de reificação requer uma análise mais minuciosa devido ao fato da programação ser uma linguagem que requer prática e outros fatores, sejam eles mentais ou pelo ambiente no qual os estudantes estão inseridos os quais não cabe este trabalho verificar.

Para que se consiga a reificação em um processo operacional é necessário não apenas obter uma lógica e interpretação avançada, mas também o discernimento para usar um comando, desta forma seu processo estrutural é ativado automaticamente.

Evidencia-se que com base nesses estudos, o avanço em programação é uma união entre processos operacional e estrutural, pois assim espera-se que torne-se possível haver a passagem para uma nova aprendizagem.

Portanto afirma-se que essa junção de ideias e processos nos leva a entender que as três fases citadas por AMBRÓSIO (2011) (interpretação, estruturação e desenvolvimento), são junções dos processos estruturais e operacionais, onde a ausência de processos que são reificados pelos estudantes se daria pela falta das etapas citadas acima por AMBRÓSIO (2011), onde fica evidente que tais etapas não foram seguidas corretamente.

Porém devemos salientar que a necessidade de se obter sucesso em resolver tais problemas levam os estudantes à atingir uma espécie de “salto quântico” onde em determinada situação encontravam dificuldade, agora passam

para uma posição que lhes permitam ter total domínio e capacidade para resolver as tarefas necessárias.

Desta forma consegue se evidenciar que não só as fases e níveis de processos que são mobilizados pelos estudantes, mas também podemos inferir que o processo no qual AMBRÓSIO (2011) cita como “salto quântico” possa ser os processos os quais foram utilizados para validar as fases no qual cada estudante se encontrava.

Portanto, podemos concluir que os conhecimentos mobilizados pelos estudantes foram em grande parte de caráter estrutural, onde a concepção operacional também se mostrou bastante evidente. Contudo não obtivemos o mesmo grau de resultado quando colocamos concepções estruturais em foco, embora as fases ocorressem em todos os registros, havendo destaque para condensação e reificação.

## 6. REFERÊNCIAS

ALMEIDA, L. S. (2007). Transição, adaptação acadêmica êxito escolar no ensino superior. *Revista Galego-Portuguesa de Psicoloxía e Educación*, 15, 203-215.

AMBRÓSIO. L, ALMEIDA. S, MACEDO, SANTOS & FRANCO. H. (2011). Programação de computadores: Compreender as dificuldades de aprendizagem dos alunos. *Revista galego-portuguesa de psicoloxía e educación*, v. 19, n, 1, 16º-2011.

BALESTRI, D, R. **A participação da História da Matemática na Formação Inicial de Professores de Matemática na Ótica de Professores e Pesquisadores.** Dissertação (Mestrado em Ensino de Ciências e Educação Matemática) – Universidade Estadual de Londrina, 2008 . p. 29-30.

BENNESEN, J., & CASPERSEN, M. E. (2006). Abstraction ability as an indicator of success for learning object-oriented programming? *SIGCSE Bulletin*, 38 (2), 39-43.

BUSSMANN, C. J. C. **Conhecimentos mobilizados por estudantes do curso de matemática sobre o conceito de grupo 2009.** 91 páginas. Dissertação (Mestrado em Ensino de Ciências e Educação Matemática) – Universidade Estadual de Londrina, 2009

BYRNE, P., & LYONS, G. (2001). The effect of student attributes on success in programming. In Fincher, S., Klein, B., Culwin, F., & McCracken, M. (Eds.), **Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education.** NY: ACM, 49-52.

FARINELLI, F.(2007) **Conceitos básicos de programação orientada a objetos,** 2007.

GARNICA, A. V. M. **História Oral e educação Matemática.** In: BORBA, M. C.; ARAÚJO, J.L (org.) **Pesquisa Qualitativa em Educação Matemática.** Belo Horizonte: Autêntica, 2004.

GOMES. A, Areias. CRISTIANA, HENRIQUES. J, & MENDES. J. A (2008). Aprendizagem de Programação de Computadores: Dificuldades e ferramentas de suporte – **Revista Portuguesa de Pedagogia**, v. 42, p. 161-179, 2º-2008

GOMES. G, MARTINHO. J, BERNARDO. M, MATOS. FERNANDO, & ABRANTES. P. (2012). Dificuldades na aprendizagem da programação no ensino profissional – Perspetiva dos alunos. **Congresso Internacional TIC e Educação, instituto da Educação da Universidade de Lisboa.**

LEITHER, H. E., & LEWIS, H. R. (1978). Why **Johnny can't program**: A progress report. *SIGCSE Bulletin*, 10 (1), 266-276.

JENKINS, T. (2002). On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*. Loughborough: University United Kingdom, 54-58.

PEA, R. D., & KURLAND, D. M. (1984). **On the cognitive prerequisites of learning computer programming** (Technical Report No.18). New York: Bank Street College of Education Center for Children and Technology.

SFARD, A. (1991). On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin. **Educational Studies in Mathematics**, 22, 1-36.

SIERRA, K. & BATES, B. (2005). **Use a cabeça! Java**. Editora Alta Books, p. 123-143.

TiExpert.net.Java:**Herança de classes**.

<<http://www.tiexpert.net/programacao/java/heranca.php>>. Acessado em out. de 2014.

WINSLOW, L. E. (1996). Programming pedagogy: A psychological overview. **ACM SIGCSE Bulletin**, 28 (3), 17-25.

## ANEXO A

## Questionário para aplicação e coleta de dados

1. A programação orientada a objeto possui muitas funções que difere da programação procedural. Sabendo disso, explique o que a programação orientada a objetivo possui de diferente da procedural?
2. Na programação orientada a objeto alguns métodos se destacam mais que outros. Sabendo disso explique o que cada método desses faz ou significa?

Boolean:  
Private:  
Extends:  
Static:  
Super:  
Length:

3. A programação procedural e a programação orientada a objeto possuem suas vantagens e desvantagens para cada caso específico. Pensando por esta perspectiva, elabore um diagrama de classe para uma escola, lembrando que é necessário cadastrar professores e alunos dentro do sistema de registro (adaptado de Farinelli, 2007).
4. Ainda com o olhar voltado para a programação orientada a objetivo uma empresa requer que seja criado um método de rotação e que tal método faça um som toda vez que seu método é chamado. Neste primeiro momento a empresa disponibilizou os seguintes dados:
  - O programa deve conter quatro objetos;
  - Cada classe deve possuir dois métodos, rotação e som;
  - Em um dos objetos seus métodos de rotação e som são diferentes das demais.

Antes de começar a programar a empresa requer que seja criado um escopo do projeto para que seja mostrado a seus clientes e verificar se suas especificações estão condizentes com o que foi pedido no início do projeto.

Sua tarefa é criar um escopo do projeto usando diagramas e explicar o que as classes estão fazendo (adaptado de Sierra & Bates, 2005).

5. Em um zoológico uma empresa foi solicitada para projetar um programa de simulação que permitisse ao usuário reunir vários animais diferentes em um ambiente para ver o que acontece. Recebemos uma lista com alguns dos animais que estarão no programa:

- Leão;
- Tigre;
- Gato;
- Hipopótamo;
- Lobo;
- Cachorro.

O zoológico definiu alguns atributos e métodos que precisam ser colocado na representação.

**Atributos:**

- Imagem
- Alimento
- Fome
- Limite
- Localização

**Métodos:**

- Fazer Barulho
- Comer
- Sono
- Rugido

Sua tarefa é representar o estado de comportamento que eles têm em comum e de que forma ira fazer isso. Lembrando que não é necessário nenhuma forma de programação (adaptado de Sierra & Bates, 2005).

6. Um programador foi contratado para desenvolver um programa porem seu código ocorreu um erro e alguns trechos de código se perderam no processo. Sua tarefa é achar os trechos corretos e implementar no código (adaptado de Sierra & Bates, 2005).

```

Public class Rowboat _____ {
    Public _____ rowtheboat ( ){
        System.out.print (“Stroke Natasha”);
    }
}
Public class _____ {
    Private int _____;
    _____ void _____ ( _____){
        Length = len;
    }
    Public int getlength ( ){
        _____ _____;
    }
    Public _____ move ( ) {
        System.out.print (“_____”);
    }
}
Public class TestBoats {
    _____ _____ _____ main ( `String[ ] args ){
        _____ b1 = new _____ ( );
        Rowboat _____ = new Rowboat ( );
        b2.setLength(32);
        b1._____( );
        b3._____( );
        _____.move( );
    }
}
Public class _____ Boat{
    Public _____ ( ){
        System.out.print (“_____”);
    }
}

```

Use os códigos abaixo para completar o código e ajudar o programar a resolver seu problema.

Boat, Rowboat, Saiboat, Testboats, return, continue, break, b1, b2, b3, stroke, Natasha, int, int b1, int b2, int b3, length, len, setLenght, getLenght, host, sail, String, void, static, public, private, rowTheBoat, subclasses, extends, drift, move

7. O código abaixo foi desenvolvido para ajudar uma empresa de eletrodoméstico e mesmo não estando pronto os clientes gostariam de entender como o código funciona antes de prosseguir, por isso você foi contratado. Sua tarefa é explicar para os clientes o que cada linha de código faz (adaptado do site TiExpert.net, 2014).

### **Classe 1: Eletrodoméstico**

```
package tiexpert;

public class Eletrodomestico {
    private boolean ligado;
    private int voltagem;
    private int consumo;

    public Eletrodomestico(boolean ligado, int voltagem, int consumo) {
        this.ligado = ligado;
        this.voltagem = voltagem;
        this.consumo = consumo;
    }
    // (...)
}
```

### **Classe 2: TV**

```
package tiexpert;
public class TV extends Eletrodomestico { private int canal;
private int volume;
private int tamanho;

public TV(int voltagem, int consumo, int canal, int volume, int tamanho) {
    super(false, voltagem, consumo);
    this.canal = canal;
    this.volume = volume;
    this.tamanho = tamanho;
}
//(...)
}
```