



**UNIVERSIDADE ESTADUAL DA PARAÍBA  
CAMPUS CAMPINA GRANDE  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
CURSO DE GRADUAÇÃO DE LICENCIATURA EM COMPUTAÇÃO**

**THIAGO DOS SANTOS MACIEL**

**Aplicação POP na UEPB: Comunicação e Documentação na  
Solução Inteligente de Problemas**

**CAMPINA GRANDE – PB  
2013**

THIAGO DOS SANTOS MACIEL

**Aplicação POP na UEPB: Comunicação e Documentação na  
Solução Inteligente de Problemas.**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação de **Licenciatura em Computação** da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de **Licenciado em Computação**.

Orientador: Prof. Msc. Fábio Luiz Leite Júnior

CAMPINA GRANDE – PB  
2013

## FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL-UEPB

M152a Maciel, Thiago dos Santos.  
Aplicação Pop-UEPB [manuscrito] : Comunicação e documentação na solução inteligente de problemas. / Thiago dos Santos Maciel. – 2013.

**19 f.: Il. color.**

Digitado

**Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2013.**

**“Orientador: Prof. Me. Fábio Luiz Leite Junior, Departamento de Matemática, Estatística e Computação”.**

1. Programação Orientada ao Problema. 2. Desenvolvimento de software. 3. Informática. I. Título.

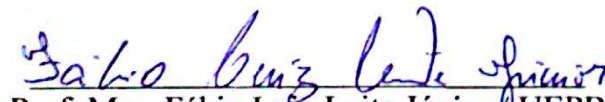
21. ed. CDD 004


**THIAGO DOS SANTOS MACIEL**

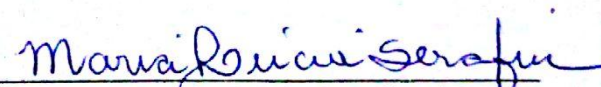
**APLICAÇÃO POP-UEPB: Comunicação e Documentação na  
solução Inteligente de Problemas**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação de **Licenciatura em  
Computação** da Universidade Estadual da  
Paraíba, em cumprimento à exigência para  
obtenção do grau de **Licenciado em  
Computação**.

Aprovada em 27/02/2013.

  
Prof. Msc. Fábio Luiz Leite Júnior / UEPB  
Orientador

  
Prof. Msc. Edson Holanda Cavalcante Júnior / UEPB  
Examinador

  
Profª Msc. Maria Lúcia Serafim / UEPB  
Examinadora

# **Aplicação POP na UEPB: Comunicação e Documentação na Solução Inteligente de Problemas.**

MACIEL, Thiago dos Santos<sup>1</sup>

## **RESUMO**

Os problemas bem definidos são os principais exercícios utilizados por professores nas disciplinas introdutórias de programação em um curso de ciências da computação. Porém, a maioria das situações que os desenvolvedores de empresas se deparam são problemas do tipo mal definidos. Ou seja, existe uma grande diferença entre o que é trabalhado na graduação e as situações reais vivenciadas por desenvolvedores. Então, a Programação Orientada ao Problema - POP - foi desenvolvida no intuito de aproximar ao máximo os alunos iniciantes em programação das situações comuns encontradas por equipes de desenvolvimento de software, colaborando diretamente na formação dos integrantes do processo. Na UEPB, a aplicação da pesquisa ocorreu no período 2012.2 nas turmas introdutórias de programação, evidenciando a dificuldade dos alunos em se trabalhar com situações incomuns, mal elaboradas. A aplicação mostrou que o desenvolvimento de software não se resume em programar, mas a um grande processo que envolve comunicação e documentação como partes permanentes deste. Os autores utilizados como principais referências foram Andrea Mendonça, Sommerville e Pólya.

**PALAVRAS-CHAVE: Programação. Desenvolvimento de Software. Informática. Título.**

## **ABSTRACT**

The well-defined problems are the main exercises used by teachers in introductory programming courses in a computer science course. However, most situations that developers of companies are facing problems such poorly defined. That is, there is a big difference between what's worked in undergraduate and real situations experienced by developers. So, the Problem Oriented Programming - POP - was developed in order to bring the most beginner students in programming of common situations encountered by software development teams, working directly in the training of members of the process. In UEPB, the application of the study was conducted between 2012.2 in introductory programming classes, showing the difficulty of working with students in unusual situations, poorly prepared. The application showed that software development is not limited to programming, but a broader process that involves communication and documentation as a permanent part of this. The authors used as main references were Andrea Mendonca, Sommerville and Pólya.

**KEYWORDS: Programming. Software development. Computing. Title.**

---

<sup>1</sup> Graduando no curso de Licenciatura em Computação pela UEPB e de Tecnologia em Telemática pelo IFPB. Bolsista do PIBICT-IFPB e Integrante do Projeto Tropeiros - UEPB.  
Contato: thi.maciell@gmail.com

## 1 INTRODUÇÃO

Nos cursos da área de tecnologia da informação, o índice de desistência dos alunos iniciantes é relativamente alto. Tomando como exemplo os cursos de ciências da computação, a taxa de evasão chega em média a 32%. Esses altos números são decorrentes, em sua maioria, da dificuldade em matérias como matemática, lógica e programação, disciplinas essenciais para os cursos da área (SIMAS, 2012). As disciplinas introdutórias em programação basicamente são as que mais exigem do aluno, embora vários deles entrem no curso sem nunca terem tido contato algum com programação. Cabe ao professor da disciplina direcioná-los ao desenvolvimento do raciocínio lógico, a sintaxe da linguagem de programação e, claro, a construção de programas.

O maior recurso utilizado nessas disciplinas pelos professores é a resolução de problemas. Eles, por sua vez, são quase sempre bem definidos, ou seja, os requisitos do sistema estão todos bem explicados no seu próprio enunciado (MENDONÇA et. al., 2009). Porém, no mundo real, é o cliente que define o que será feito. Logicamente, é de responsabilidade dele definir os requisitos do software que deseja obter. Eis o grande problema: salvo os que são da área, os clientes não entendem da linguagem técnica necessária para explicar o software desejado do jeito que se deve. Então, a descrição do problema acaba sendo mal feita, transformando-o em um problema mal definido. De acordo com Mendonça (2010),

Problemas de programação mal definidos são aqueles cujo enunciado não descreve o que deve ser feito de forma clara e completa. Isto ocorre porque o enunciado contém um ou um conjunto de defeitos, tais como, ambiguidades, contradições, falta de informações, informações incorretas e/ou irrelevantes.

O caminho para acabar com essa diferença entre o que é visto pelos alunos durante a graduação e o que ocorre na "vida real" como desenvolvedor, seria utilizar os problemas mal definidos durante todas as disciplinas relacionadas, principalmente nas introdutórias (MENDONÇA, 2010). Com isso, o aluno acaba se acostumando com esse tipo de situação mal elaborada e aprende a solucioná-la com mais facilidade, por ter sido adaptado, treinado a lidar com elas. O futuro desenvolvedor vai utilizar dessas

técnicas de forma automática, instintiva, e vai ter bagagem e experiência para resolver qualquer tipo de situação.

Pensando nisso, a metodologia da *Programação Orientada ao Problema*, ou POP, foi elaborada. Ela adotou uma estrutura de trabalho onde os problemas mal definidos são sua principal ferramenta. Este trabalho visa a aplicação da POP nas disciplinas iniciais de ensino de programação na UEPB, tendo dois objetivos principais: o primeiro está na utilização de problemas mal definidos nas disciplinas introdutórias de programação, avaliando o comportamento e o desempenho desses alunos ao se deparar com esse tipo de problemas; e o segundo objetivo está em auxiliar a transição da licenciatura para o bacharelado em computação, ocorrido no primeiro semestre do ano de 2012. Com a mudança na grade curricular e a aprovação do projeto pedagógico do bacharelado em computação, a programação passou a ser o foco principal do curso da UEPB, e a mudança das metodologias de ensino passou a ser extremamente necessária. É uma pesquisa acadêmica de caráter qualitativo exploratório, com aspectos quantitativos e descritivos, aplicado nas disciplinas de algoritmos dos turnos da tarde e da noite, na segunda unidade temática do período de 2012.2, da turma de Ciências da Computação da Universidade Estadual da Paraíba.

## **2 REFERENCIAL TEÓRICO**

Uma definição clássica de problema defendida por Pozzo, que é comumente utilizada por vários autores, é a de uma situação que um indivíduo ou um grupo quer ou precisa resolver, e não possui um conhecimento e/ou procedimento para solucioná-lo. De acordo com Pozzo (1994), existe uma diferença entre exercício e problema. No exercício, o indivíduo possui e utiliza de recursos que o levam de forma rápida à solução, fato que não ocorre durante o problema. Na prática, deve-se observar a diferente interpretação entre os indivíduos, ou seja, alguns podem enxergar um problema como exercício e outros um exercício como um problema. Isto é mais evidenciado pela diferença entre os veteranos e os novatos, onde em uma situação problemática bem definida, os veteranos apresentam maior rapidez na solução e menor índice de erros, se comparado aos novatos. Justamente, por conta dos conhecimentos prévios disponíveis por eles, que permite reconhecer mais facilmente os atributos do

problema e aplicar os procedimentos apropriados para solucioná-lo. Certamente, os professores utilizam de exercícios e problemas durante a sua prática educativa, pois é uma parte importantíssima do processo que muito favorece ao entendimento dos alunos.

Os processos de resolução de problemas vêm sendo analisados e sofrendo mudanças ao longo do tempo. Algumas abordagens de resoluções de problemas foram analisadas durante a elaboração da metodologia POP e serão relatadas em seguida. John Dewey, em 1910, definiu os passos do seu modelo como sendo: (i) *Definir o problema*, (ii) *Sugerir possíveis soluções*, (iii) *Refletir sobre a possível solução* e (iv) *Testar e provar*. Já outro estudioso chamado Graham Wallas, em 1926, definiu um modelo com três passos: (i) *Preparação*, que é investigar o problema, colher informações, (ii) *Incubação e Iluminação*, que é pensar inconscientemente no problema enquanto realiza outras atividades; e (iii) *Verificação*, relacionada a avaliação da solução. Alexander Simon, 1960, organizou o processo como (i) *Inteligência*, (ii) *Processo* e (iii) *Escolha, Implementação*. Um pouco mais recente, Fadi P. Deek em 1999, definiu o processo como sendo (i) *Formular o problema*, (ii) *Planejar a solução*, (iii) *Projetar e Traduzir a solução* e (iv) *Testar e Entregar a solução*.

Dentre esses e outros modelos existentes, o escolhido como base para a produção da metodologia POP foi o Modelo de Resolução de Pólya, por ser mais completo ao integrar características dos demais processos, e por integrar atividades gerais que permitem a utilização dele em outras áreas de conhecimento. De acordo com Mendonça (2010), o modelo de Pólya é definido em quatro fases: (i) *Entender o Problema*, ou seja, identificar os dados relevantes do problema; (ii) *Estabelecer um Plano*, ou seja, decompor o problema em partes e produzir estratégias para alcançar a solução; (iii) *Executar o Plano*, onde a solução é construída; e (iv) *Examinar a Solução obtida*, que é a avaliação.

Pólya (1975) afirma que na primeira fase, *Entender o Problema*, é essencial para o aluno o interesse e o desejo de buscar a solução do problema, pois é difícil trabalhar na resolução de uma problemática que não nos interessa. Então, cabe ao professor evitar que esse tipo de situação ocorra, justamente na escolha dos problemas



a serem apresentados aos alunos. "O problema deve ser bem escolhido, nem muito difícil nem muito fácil, natural e interessante, e um certo tempo deve ser dedicado à sua apresentação natural e interessante. " (Pólya, 1975). Existe um caminho a ser descoberto entre a primeira fase e a segunda fase, *Estabelecimento de um Plano*. De acordo com Pólya, tal descoberta pode ser através da compreensão da idéia de um plano, de forma gradativa ou então após várias tentativas sem sucesso, aparecer repentinamente uma "idéia brilhante". O professor, então, possui a função de propiciar, discretamente, o surgimento dessa boa idéia, com as discussões ou questionamentos sobre o problema, e até mesmo passar sua experiência sobre outras situações vivenciadas. Tendo em mãos um plano de ação consistente, a terceira fase, *Execução do Plano*, se torna bem simples, exigindo do aluno apenas a paciência e atenção. Mas, é preciso ter convicção de que esse plano de ação contém os detalhes necessários a resolução do problema (Pólya, 1975). Para tal, cabe ao professor realizar uma verificação dos passos do plano de ação, evitando com isso, maiores transtornos na execução. Existe a possibilidade do aluno esquecer ou desviar do plano de execução. As maiores chances de que isso ocorra são nos casos de planos traçados por terceiros ou então ter sido aceito por influência do professor. Mas se o próprio aluno tiver preparado o plano, mesmo que com algum auxílio, dificilmente ele se perderá na execução, concebendo um certo período de tranquilidade ao professor (Pólya, 1975). A última fase, *Examinar a Solução Obtida* ou *Retrospecto*, como Pólya (1975) se refere em seu livro, é uma das mais difíceis de ser cumprida, não por exigir mais conhecimento técnico do aluno, mas sim pelo excesso de confiança adquirido com a solução do problema em mãos. Isto faz com o que a maioria dos alunos encerrem o problema e passem a uma próxima situação. A revisão dos passos do processo de resolução do problema, analisando o resultado final e o caminho percorrido para chegar a tal, faz com o que o conhecimento adquirido seja consolidado, e a capacidade de solucionar problemas do aluno se torna mais aguçada (Pólya, 1975).

Trazendo mais para o nosso foco, surge um conceito de Somerville (2007) que define o processo de desenvolvimento de software como um conjunto de atividades que leva a um produto de software. Não existe um processo ideal de produção de software, ou seja, é dependente da criação e organização de cada equipe de desenvolvimento, que vai adequar a sua melhor forma de trabalho. Porém, algumas atividades mais

fundamentais são comuns a todos esses processos, como a *Especificação*, o *Projeto*, a *Implementação*, a *Validação* e *Evolução de Software*. Comparando essas atividades fundamentais do processo ao Modelo de Pólya, verifica-se que eles são diretamente relacionados. Tomando por exemplo a primeira etapa, *Especificação de Requisitos*, é uma parte fundamental para o entendimento do problema, pois é onde são definidos os seus escopos e o seu objetivo principal. Fazendo uma comparação, ela recai diretamente na primeira fase do Modelo de Pólya, de *Entender o problema*.

Uma outra definição é defendida por Paula Filho (2005), afirmando que em engenharia de software, processos podem ser definidos para atividades como desenvolvimento, manutenção, aquisição e contratação de software, podendo ainda determinar subprocessos para cada um deles. Um bom exemplo é no caso de desenvolvimento, onde os seus subprocessos envolvidos são na determinação dos requisitos, na análise, no design, na implementação e nos testes.

Devemos deixar claro a importância de se ter um planejamento que perdure por todo o processo de desenvolvimento de qualquer produto de software. Para se ter um software de alta qualidade, as práticas utilizadas para obtê-lo também devem ser de alta qualidade. Segundo McConnel (2005),

Um bom planejador elimina os principais riscos do caminho o quanto antes, para que a maior parte do projeto possa prosseguir do modo mais suave possível. Os riscos mais comuns no desenvolvimento de software são requisitos mal delineados e um planejamento de projeto mal feito.

McConnel (2005) elaborou uma relação de qualidade e estágio de projeto: se o desenvolvedor busca a qualidade no final do projeto, então enfatiza os testes de sistema. Um grande risco, pois os testes não identificam falhas como a construção de um produto errado. Se ele busca a qualidade no meio do projeto, então enfatiza as práticas de construção. Um risco menor, porém nessa fase, já se estabelece a base do sistema, para um sucesso ou falha. Se ele busca a qualidade no início do projeto, então ele enfatiza o planejamento, logo exige mais e projeta um software de alta qualidade. São relações importantes que definem principalmente o custo do desenvolvimento do sistema.

A Programação Orientada ao Problema - POP, é uma metodologia que contempla um conjunto de atividades a serem inseridas nas disciplinas introdutórias de programação, no intuito de auxiliar os estudantes na especificação dos requisitos do sistema e na produção dos documentos de especificação (Mendonça, 2010). Tal metodologia consiste em fazer com o que o aluno tenha um contato mais de acordo com a realidade de um programador. Na UEPB, com a migração da licenciatura para o bacharelado, tornou-se mais que necessário realizar uma avaliação na metodologia do ensino de programação. Quando licenciatura em computação, o principal foco era a formação de profissionais capazes de utilizar das ferramentas computacionais em prol da educação. Já como bacharelado, o lado educacional deixa de ser a principal formação, dando espaço para a programação, focando o desenvolvimento.

Nas disciplinas introdutórias de programação, os problemas passados aos alunos são do tipo bem elaborado, ou seja, o aluno precisa apenas da interpretação do enunciado para resolver o que se pede. Na metodologia, o foco já está na utilização dos problemas mal definidos, onde se exige além da interpretação do enunciado, o desenvolvimento de técnicas para a identificação dos requisitos do problema.

Os participantes da metodologia são: O professor, assistentes e o desenvolvedor. O professor e os assistentes exercem o papel de clientes-tutores, onde são responsáveis por tirar as dúvidas dos estudantes sobre o problema, enquanto clientes, e administrar as interações dos grupos, marcar as reuniões e orientar durante a realização da atividade, enquanto tutores. Já os desenvolvedores são responsáveis por cumprir o ciclo de desenvolvimento do POP e produzir os artefatos esperados, no caso, o documento de especificação, os casos de testes de entrada e saída, programa e testes automáticos. O modo de trabalho dos desenvolvedores possui 2 etapas de trabalho: individualmente, trabalham na solução do problema e elaboração de protótipos do programa e dos casos de testes automáticos; em grupo, identificam os requisitos, elaboram o documento de especificação e os casos de testes de entrada/saída.

## 2.1 Ciclo de Resolução POP

O POP possui um ciclo de atividades, que consiste em: *Elaborar a especificação inicial, Iniciar a implementação, Concluir a especificação, e Concluir a implementação.*

#### 2.1.1 Elaborar a Especificação Inicial

Fase inicial do processo, onde o cliente-tutor apresenta à equipe de desenvolvimento, o problema a ser resolvido. Um dos membros da equipe é escolhido para ser responsável pela elaboração do documento inicial, registrando o que ocorre na primeira reunião, devendo compartilhar com os demais incluindo o tutor, através de um editor de texto colaborativo. Ao final dessa etapa, a equipe deve ter produzido o documento de especificação inicial, descrevendo textualmente os requisitos que o programa deve ter, e os casos de testes de entrada/saída que mostram o resultado esperado através de exemplos.

#### 2.1.2 Iniciar a Implementação

Segunda etapa, onde se inicia a construção do protótipo da solução, de forma individual por cada membro da equipe. A linguagem de programação adotada deve ser a mesma aplicada na disciplina. Os testes automáticos devem adotar os mesmos exemplos de entrada/saída descritos no documento da fase anterior. Comumente, surgem mais dúvidas sobre o problema, devendo estas serem relatadas e discutidas no grupo de trabalho online, sob supervisão do tutor. Em cada nova dúvida, novos requisitos podem surgir, fazendo com o que uma alteração seja feita no documento de especificação. Esta passa a ser de responsabilidade de todo o grupo, e não apenas de um membro. Ao final dessa fase, teremos a base do código-fonte, com os testes automáticos e o documento de especificação devidamente atualizado.

#### 2.1.3 Concluir a Especificação

Terceira etapa, onde mais uma reunião presencial é marcada entre o grupo e o cliente-tutor. O grupo deve estar com os protótipos dos programas, e deve fornecê-los ao cliente, que irá realizar os testes. Esse processo é fundamental na identificação de novos requisitos e na possível correção ou refinamento dos requisitos que já foram documentados. Novamente, um membro do grupo é destinado a atualizar o documento de especificação e os casos de testes de entrada/saída, e compartilhar as alterações

com os demais. Nessa fase, o documento de especificação deve ser evoluído do inicial para o final, contendo todos os requisitos desejados pelo cliente-tutor.

#### 2.1.4 Concluir a Implementação

Nessa fase, o documento de especificação de requisitos deve estar completo, contendo todos os requisitos descritos e desejados pelo cliente-tutor, como também os testes de entrada/saída. Com isto em mãos, cada desenvolvedor do grupo deve atualizar os protótipos do programa e os casos de teste automáticos, podendo haver interação entre eles através da lista de discussão. O objetivo final é, claramente, a conclusão do programa fonte, que atenda aos requisitos propostos pelo cliente-tutor. Lembrando que a atividade é realizada, tendo uma data limite pré-determinada para a entrega do documento de especificação, dos programas e dos testes automáticos, finalizando assim, o ciclo de atividade do POP.

### **3 REFERENCIAL METODOLÓGICO**

Este artigo é de caráter exploratório, com características quantitativas e qualitativas. Para aplicação do POP no âmbito da Universidade Estadual da Paraíba, algumas adaptações foram realizadas na versão original da metodologia, por falta de tempo hábil para a aplicação da pesquisa e por falta de voluntários para o acompanhamento. Então, o papel de Cliente-Tutor foi realizado por mim, aplicador da pesquisa, onde orientei todos os grupos, em ambos os turnos. O papel dos desenvolvedores, feito pelos alunos, também teve alterações, onde no nosso caso, todas as atividades, incluindo o processo de desenvolvimento do código-fonte, eram realizadas apenas em grupo. Não foram adotados testes automáticos, devido a inexperiência dos grupos e ao pouco tempo disponível, impossibilitando a pesquisa e o autoaprendizado.

#### 3.1 Divisão dos grupos

A pesquisa foi iniciada com 52 alunos participantes, de duas turmas de algoritmos, uma a tarde e uma a noite, do período de 2012.2 do curso de bacharelado em Ciências da Computação, da Universidade Estadual da Paraíba. A turma da tarde esteve com 24 alunos, divididos em 4 grupos de 3 alunos e 3 grupos de 4 alunos,

totalizando 7 grupos de trabalho. Já na turma da noite, foram 28 alunos divididos em 4 grupos de 3 alunos e 4 grupos de 4 alunos, totalizando 8 grupos. Em ambas as turmas, a divisão foi feita pelos próprios alunos, levando em consideração a afinidade, o método de trabalho que eles desenvolveram entre si, etc. Dessa maneira, as atividades foram proporcionalmente divididas entre os alunos.

### 3.2 Acompanhamento

O acompanhamento foi realizado por um tutor, que marcava reuniões presenciais antecipadamente, e à distância utilizando a ferramenta *google groups*, que disponibiliza vários recursos favoráveis a aplicação da pesquisa, incluindo o google docs, muito utilizado para a elaboração de todos os documentos textuais do ciclo POP. Foi destinado um grupo de trabalho do google para cada equipe, com os mesmos tópicos e exercícios sendo passados a todos os grupos, já que o acompanhamento estava sendo feito pelo mesmo tutor.

### 3.3 As atividades

A atividade foi dividida em 2 partes. Em cada parte, foram passados 2 problemas mal definidos e para cada problema, foi pedido um documento de especificação e o código-fonte da solução. Cada grupo de trabalho teve a oportunidade de realizar uma entrevista com o cliente que solicitou o problema. Nessa entrevista, os desenvolvedores tentavam através de perguntas, esclarecer as dúvidas sobre os programas que o cliente pediu e identificar quais eram os requisitos corretos para a construção do código-fonte. Após cada etapa de entrevista, um prazo de uma semana era definido, para que houvesse um processo de validação dos dados pelos próprios alunos. Ou seja, o grupo analisava mais detalhadamente o enunciado mal elaborado, juntamente com os requisitos colhidos na entrevista, e com isso tentavam preencher essas lacunas com o que conseguiram encontrar.

Dando continuidade, entra em ação um **layout de requisitos** (vide anexo 2), que é um template com os possíveis tipos de requisitos. A função deste foi de orientar o grupo para a identificação dos requisitos corretos em cada problema. A estrutura do layout foi definida pensando em facilitar a identificação, já que os nomes dos tipos de requisitos estavam em espaços determinados. Tal layout foi entregue para cada grupo,

que o preencheu nos espaços determinados, com requisitos encontrados durante a entrevista. Após o preenchimento, ele foi entregue ao tutor para avaliação da qualidade e da quantidade dos requisitos encontrados. Resumindo, o ciclo de resolução POP foi realizado uma vez para cada problema, com seu devido layout, sua documentação e seu programa, associados.

### 3.4 Avaliação

A avaliação foi realizada levando em consideração dois fatores: a quantidade dos REQUISITOS encontrados e a DOCUMENTAÇÃO elaborada do programa. Os requisitos são as características desejáveis para a construção de um software, e a documentação nada mais é do que um texto que acompanha o software, explicando suas características principais e o modo como ele funciona. A qualidade desses dois artefatos foi definida, comparando o que era produzido e identificado por cada grupo, com a resposta do tutor. Ou seja, a quantidade e a qualidade dos requisitos encontrados por cada grupo era comparado com os requisitos do tutor, previamente definidos antes do início da atividade. Da mesma forma, ocorreu com o documento de especificação; era elaborado um por grupo que era comparado a o documento final do tutor.

## 4 DADOS E ANÁLISE DA PESQUISA

A aplicação da pesquisa se estendeu entres os meses de setembro a dezembro, do ano de 2012. Por ter sido aplicada apenas na segunda unidade letiva deste semestre, alguns alunos não quiseram dar o devido empenho a aplicação da metodologia POP. O motivo principal foi o de não ter conseguido obter um bom rendimento durante o início da disciplina, que ocasionou um grande desânimo para a realização das atividades e a desistência destes antes do final da aplicação da metodologia. Para os que não desistiram, os problemas foram com a falta de horário, pois muitos dos alunos exercem outras atividades, como por exemplo, emprego. Não podemos deixar de mencionar as outras disciplinas do semestre, que também exigem dos alunos um tempo considerável para estudo. O resultado desses contratempos foi que nem a turma da tarde e nem a da noite, entregou o documento de especificação e os programas relacionados aos problemas 3 e 4 da atividade POP. As turmas chegaram

a ter contato com os problemas e realizar as entrevistas com os clientes, porém o produto final não foi entregue.

Os passos 1 e 2 do ciclo de desenvolvimento POP, que envolvem o início da documentação e da implementação, foram os mais complicados para os alunos. Durante o desenvolvimento da atividade, um problema em comum vivenciado por eles foi com relação à identificação dos requisitos dos sistemas. No primeiro contato com os enunciados dos problemas, todos os grupos já fizeram pequenos esboços do programa, antes mesmo da entrevista com o cliente, e usaram isto como método de identificação de requisitos. Na primeira entrevista, todos os grupos tiveram dificuldades nesse diálogo com o cliente. Fizeram perguntas muito técnicas, envolvendo linguagem de programação desejada pelo cliente, estruturas lógicas a serem utilizadas, métodos de saídas. São assuntos que um cliente "normal" não tem condições de responder. O resultado dessa primeira entrevista não poderia ser outro: muitas dúvidas ficaram pendentes e muitos requisitos não foram identificados. Um efeito cascata acabou acontecendo, pois houve falha na entrevista, conseqüentemente falhas também na documentação e no primeiro protótipo. A primeira reunião com o tutor se tornou crucial para a sequência da metodologia. Foi nela que o tutor esclareceu todas as dúvidas relacionadas a abordagem, e os grupos encontraram o rumo certo da metodologia, pois aprenderam a técnica chave para identificação de requisitos: a comunicação simples com o cliente.

Com uma nova visão metodológica, os passos 3 e 4 do ciclo, que envolveram a conclusão da documentação e do programa, foram mais simples. Na segunda entrevista, percebeu-se a notória evolução. Os alunos conseguiram ir direto ao ponto nas perguntas, usando uma linguagem simples e de melhor entendimento para o cliente. Os resultados obtidos nessa segunda entrevista foram totalmente diferentes, onde cada equipe, respeitando as devidas diferenças, preencheram as lacunas deixadas na primeira reunião, identificando os requisitos que eles necessitavam para a resolução do problema.

Os dados coletados com a aplicação do layout de requisitos comprovam a evolução. A figura I mostra os resultados do desempenho dos grupos da turma da noite,



relacionados ao problema da "Poupança Programada". O eixo vertical do gráfico representa o número de requisitos e o eixo horizontal, o nome dos grupos participantes. A parte em verde do gráfico representa o número ideal de requisitos, determinados pelo tutor da metodologia, antes mesmo da apresentação do exercício aos alunos. A parte em azul, determina os excessos dos requisitos encontrados, e a parte em vermelho, os requisitos faltantes, comparando com a resolução ideal do problema, pré-determinada. Os grupos Alfa, Gama e Zeta foram os que excederam o número ideal, onde encontraram 15, 16 e 17 requisitos, respectivamente. Os grupos Capa e Lambda ficaram abaixo da meta determinada, com 11 e 13 requisitos cada. Já os grupos Beta e Delta conseguiram identificar o número ideal dos requisitos para essa solução.

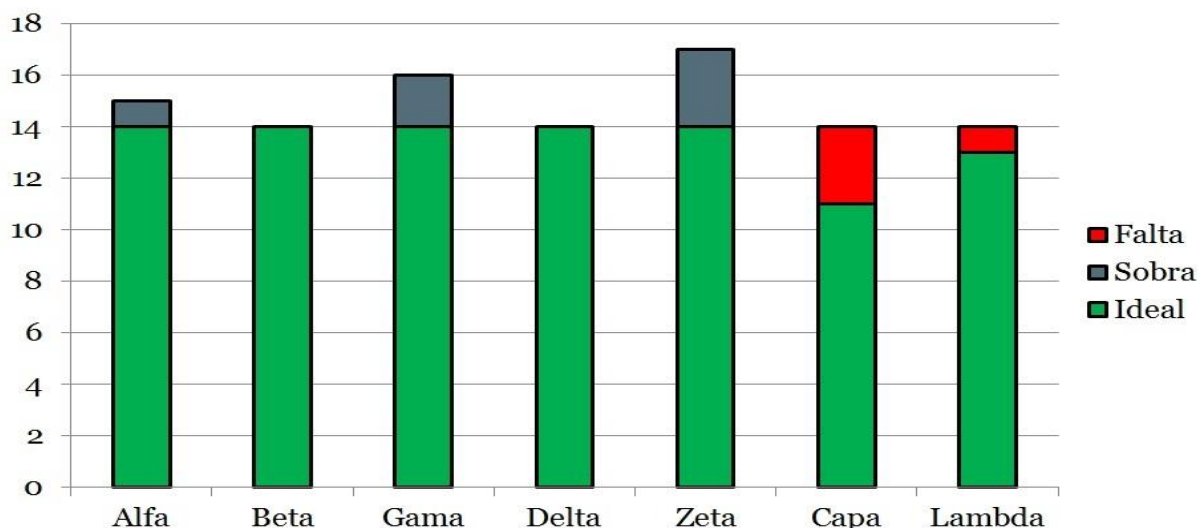


Figura I: Requisitos encontrados no Problema da Poupança Programada - Turma da Noite

A figura II apresenta os resultados obtidos no mesmo problema, na turma da tarde. Pelo exposto, podemos perceber que o rendimento foi um pouco menor, se comparado a turma da noite. O grupo Alfa e o grupo Sigma foram os que apresentaram maior variação no número de requisitos, onde o primeiro encontrou 19, enquanto o outro apenas 10. Apenas o grupo Zeta encontrou o número ideal de requisitos, para a solução desse problema. O grupo Beta também apresentou uma variação considerada alta, tomando como base o número ideal de requisitos, com apenas 11 encontrados. Os demais grupos apresentaram variações dentro da média esperada, entre 1 e 2 requisitos de diferença, para mais ou para menos.

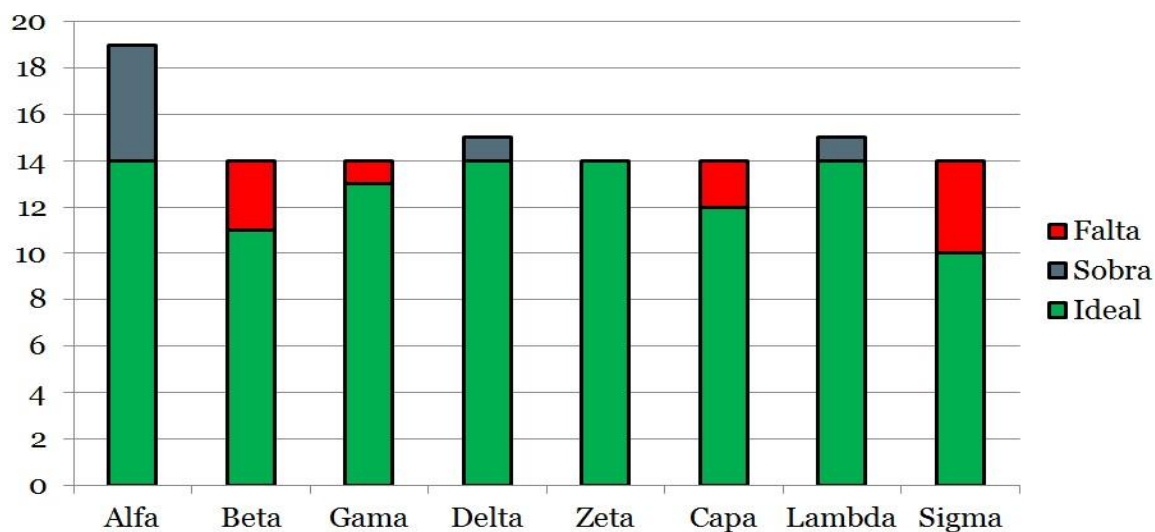


Figura II : Requisitos encontrados no Problema da Poupança Programada - Turma da Tarde

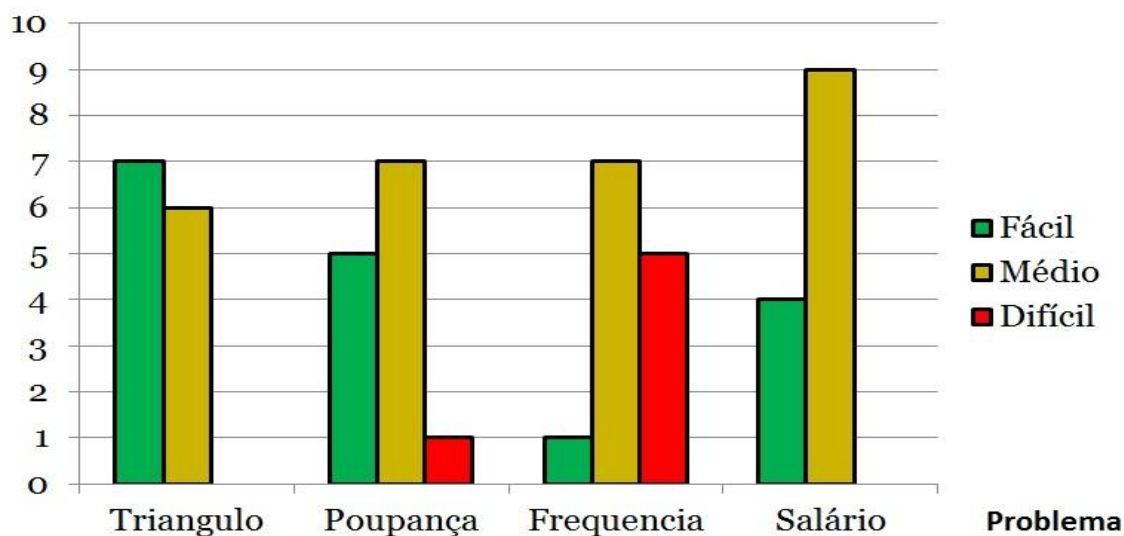
Ao ser questionado sobre as dificuldades encontradas para entender e especificar os problemas, um dos alunos participantes respondeu da seguinte forma:

O entendimento da comunicação entre o cliente e os desenvolvedores quanto a extração dos requisitos foi a maior dificuldade. Muitas vezes o cliente não entende a linguagem dos desenvolvedores, e com isso, podem esquecer de explicitar algumas funções do sistema que ele deseja.

A resposta desse mesmo aluno, quando questionado sobre o que foi aprendido no roteiro da metodologia, respondeu:

Aprendi que uma boa comunicação com o cliente é essencial, pois torna mais fácil a extração dos requisitos, o desenvolvimento do documento de especificação e do programa.

Os alunos avaliaram o nível de dificuldade dos problemas elaborados para a metodologia POP. O resultado é mostrado na figura III, que está organizada da seguinte maneira: No eixo horizontal está disposto os tipos dos problemas aplicados durante a metodologia POP. Já no eixo vertical do gráfico, está a quantidade dos alunos que responderam ao questionário. A maioria dos alunos achou a dificuldade dos problemas mediana. O problema mais fácil foi o triângulo, com 7 votos, e o mais difícil, o da frequência das palavras, com 5 votos.



**Figura III: Avaliação dos Alunos da Tarde sobre o Nível de Dificuldade dos Problemas POP**

Uma discussão que surgiu após a aplicação da metodologia POP está relacionada ao nível de experiência das turmas em que ela deve ser aplicada. A metodologia original sugere a aplicação em turmas de algoritmos, para alunos que estão ingressando na universidade, sem experiência alguma com programação. Nesse caso, existe um grande benefício didático, pois vão aprender juntamente com a programação, outras técnicas importantes relacionadas ao processo de desenvolvimento, como a documentação do processo e também os métodos de abordagem de clientes. O problema surge na cobrança dos testes automáticos, originalmente pedido na metodologia, por conta que os alunos não conseguem adquirir, em apenas um semestre, a experiência e nem a maturidade necessárias para lidar com esse tipo de exigência. Por essa falta de maturidade, surgiu a necessidade de se aplicar a metodologia POP em turmas mais avançadas de programação, para analisar o desempenho desses alunos mais experientes, e verificar se os resultados da pesquisa se tornem mais consistentes.

Analisando qualitativamente o desempenho e a evolução dos alunos como resultados, mesmo com alguns contratemplos, foram bem satisfatórios. Eles conseguiram entender que a resolução de um problema computacional não se resume em programar, mas sim em todo um processo envolvendo diálogos com os clientes,

documentação de todos os passos e vários tipos de testes envolvidos na construção da solução.

## **5 CONCLUSÃO**

A metodologia POP na UEPB, apesar de alguns contratempos durante a aplicação e acompanhamento, rendeu resultados satisfatórios. Os alunos que participaram da metodologia conseguiram compreender a real importância da comunicação entre cliente e desenvolvedor. Eles entenderam o quanto a documentação desde o início do processo facilita a vida de toda a equipe de desenvolvimento, pois o raciocínio é organizado e registrado, diminuindo consideravelmente os riscos de desvios do que se realmente pediu. Utilizando métodos de engenharia de software, principalmente a entrevista, conseguiram extrair os requisitos do sistema que não estavam no enunciado do problema proposto pelo cliente, obtendo grande êxito nas suas soluções.

Como continuidade, proponho a aplicação dessa metodologia para os alunos mais avançados em programação, ou seja, os que já tiveram cursado as disciplinas introdutórias, como algoritmos, para conhecer o rendimento deles ao se depararem com situações envolvendo problemas mal elaborados. Sugiro que a aplicação dessa metodologia seja um processo prolongado entre as disciplinas da área, que tenha duração de, no mínimo, 2 semestres consecutivos, para que se obtenham um número maior e mais consistentes de dados para avaliação. Deve-se descobrir se os alunos de períodos mais avançados, conseguem ter a comunicação e a organização necessária para um trabalho em equipe satisfatório. É de extrema importância obter esses dados, avaliá-los, para que haja uma mudança de postura na formação de novos desenvolvedores, mais comunicativos, mais entrosados com a equipe, e utilizando da documentação de software não apenas no final, mas sim em todo o processo de desenvolvimento.

## REFERENCIAS

McCONNELL, Steve. Code Complete: Um guia prático para a construção de software; tradução João Tortello - 2. ed. Bookman - Porto Alegre. 2005

MENDONÇA, Andréa. **Programação Orientada ao Problema: Uma Metodologia para Entendimento de Problemas e Especificação no Contexto de Ensino de Programação para Iniciantes**. 2010. 187 f. Tese (Doutorado em Ciências da Computação), Universidade Federal de Campina Grande - Campina Grande, 2010.

MENDONÇA, Andréa.; OLIVEIRA, Clara de.; GUERRERO, Dalton.; COSTA, Evandro. **Difficulties in Solving Ill - Defined Problems: A Case Study with Introductory Computer Programming Students**. In: IEEE International Conference of Frontiers in education conference, 39th. San Antonio, Texas., pages 1171–1176. 2009.

PAULA FILHO, Wilson de Pádua. **Engenharia de Software: fundamentos, métodos e padrões**. LTC Editora. Rio de Janeiro, 2005.

PÓLYA, George. **A Arte de Resolver Problemas**. Editora Interciência. Universidade de Stanford. 1975.

POZZO, Juan Ignacio. **La Solución de Problemas**. Editora Santillana. Madrid, 1994.

SIMAS, Anna. **As Graduações Campeãs de Desistência**. Disponível em: <<http://www.gazetadopovo.com.br/vida-universidade/nocampus> >. Acesso em: 18 dez. 2012.

SOMMERVILLE, Ian. **Engenharia de Software - 8ª edição**. Editora Pearson, 2007.

## **ANEXOS**

### **1 Problemas mal definidos aplicados no POP- UEPB**

#### Problemas Mal Definidos – Parte 1

##### Problema 1: TRIÂNGULO

Um professor de matemática de uma escola municipal de Campina Grande necessita de um programa que dadas às medidas de um triângulo, determine o tipo dele e apresente-o impresso na tela. Faça um programa para atender a necessidade do professor.

##### Problema 2: POUPANÇA PROGRAMADA

Em função da crise financeira mundial, tem crescido os investimentos na poupança programada, pois é um investimento rentável e com baixíssimo risco. Um professor de administração financeira da UFCG deseja simular investimentos na poupança programada que atenda a solicitação desse professor, considerando que a poupança rende 5% ao mês, sendo tempo e capital programado pelo investidor com isenção total do imposto de renda.

#### Problemas Mal Definidos – Parte 2

##### Problema 3: FREQUÊNCIA DAS PALAVRAS

Um professor de português precisa de um programa que, dado um parágrafo, apresente com que frequência as palavras se repetem.

##### Problema 4: SALÁRIO DOS FUNCIONÁRIOS

“Tijolos & Cia.” é uma loja de material de construção. O dono da loja precisa de um programa para calcular o salário líquido de seus vendedores. Os vendedores recebem salário fixo e comissão sobre as vendas e sobre esse valor deve incidir alguns descontos como IR. A comissão paga pela loja aos vendedores é de 3% sobre o valor total das vendas até R\$ 10.000,00 e 4% sobre o que ultrapassar esse valor. Faça um programa que atenda as necessidades do dono da “Tijolos & Cia.”.

## 2 Layout de Requisitos usado durante a aplicação POP na UEPB:

Universidade Estadual da Paraíba - UEPB  
 Centro de Ciências e Tecnologia – CCT  
 Departamento de Matemática, Estatística e Computação  
 Aplicação do POP – Programação Orientada ao Problema

### Layout de Requisitos

<b>Entradas</b>	
<b>Formatação das Entradas</b>	
<b>Restrição das Entradas</b>	
<b>Fórmulas</b>	
<b>Saídas</b>	
<b>Formatação das Saídas</b>	
<b>Mensagens de erro para entradas inválidas</b>	