ERNESTO HABERKORN

A FORMAÇÃO DO DESENVOLVEDOR DE SOFTWARE APLICATIVO NO BRASIL: PERFIL, DEFICIÊNCIAS, CAUSAS E SOLUÇÕES

Dissertação apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT, para obtenção do título de Mestre em Engenharia de Computação.

Área de Concentração: Engenharia de Software

.

Orientador : Prof. Dr. Robert Liang Koo

São Paulo

2004

Ficha Catalográfica
Elaborada pelo Centro de Informação Tecnológica do IPT

H114f Haberkorn, Ernesto Mário
A formação do desenvolvedor de software aplicativo no Brasil: perfil, deficiências, causas e soluções. / Ernesto Mário Haberkorn. São Paulo, 2004.
115p.

Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Área de concentração: Engenharia de Software.

Orientador: Prof. Dr. Robert Liang Koo

1. Engenharia de software 2. Software aplicativo 3. Aprendizagem 4. Qualificação profissional 5. Brasil 6. Tese I. Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Centro de Aperfeiçoamento Tecnológico II. Título

05-27 CDU 004.41:378(043)

DEDICATÓRIA

Este trabalho é dedicado
à minha mãe Luiza Haberkorn e
meu pai Werner Haberkorn.

AGRADECIMENTOS

Agradeço
à minha esposa Elaine Cristina A. de Faria Haberkorn,
aos meus filhos Alexandre Haberkorn,
Patrícia Sodré Haberkorn,
Daniela Sodré Haberkorn,
Murilo Sodré
por toda a força e compreensão nos momentos de minha ausência.

A todos os Professores,
em especial ao Prof. Dr. Robert Liang Koo
pela dedicada e proveitosa orientação.

RESUMO

Ao longo do tempo, a boa formação do desenvolvedor de software tem se mostrado uma difícil tarefa a ser realizada. De um lado, os avanços tecnológicos que possibilitam o surgimento de ferramentas auxiliares à compreensão e construção de programas e sistemas; do outro, o próprio desenvolvimento tecnológico gerando dificuldades tanto na assimilação quanto no uso de novas técnicas.

O objetivo desta pesquisa é verificar qual é o perfil dos desenvolvedores de software aplicativo do Brasil e quais são as formações complementares necessárias para seu aprimoramento se quisermos ter uma mão de obra de qualidade de primeiro nível para competir no mercado mundial de software. A investigação se baseia nas informações colhidas entre os profissionais atuantes no mercado de trabalho e permite, através da análise dos resultados, compreender a qualificação, o conhecimento e os métodos de trabalho dos desenvolvedores de software do Brasil.

A partir dos resultados pode-se concluir que os desenvolvedores são em geral jovens, com dinamismo requerido para desenvolvimento de software, com disposição e energia necessária para aprender assuntos interdisciplinares e com inovações tecnológicas constantes tão características desta área. A classe social de origem dos desenvolvedores é na sua grande maioria proveniente da classe média baixa, o que mostra, por um lado, ser o setor de software uma oportunidade excelente para ascensão profissional e social, por outro lado, um embasamento deficiente, uma vez que a maioria deles recebeu educação de ensino fundamental e médio nas escolas públicas cujo ensino é reconhecidamente de qualidade duvidosa.

Foi verificado que a maioria dos desenvolvedores adquiriram seus *skills* de programação no trabalho ou em cursos de treinamento patrocinados pelas suas empresas. Isso nos faz repensar sobre o currículo dado em nossas universidades. Provavelmente as universidades deverão dedicar-se mais ***na formação*** de uma base sólida para os profissionais de software, deixando ***a informação*** das ferramentas, linguagens e componentes para os cursos internos de treinamento nas empresas em que irão atuar. O sincronismo entre as duas partes é fundamental para não deixar lacunas na boa formação profissional do desenvolvedor do software..

Por outro lado, notamos que os desenvolvedores estão bem informados sobre as novas tecnologias, principalmente nas empresas que implantaram processos de certificação de qualidade como o CMM, embora estas tecnologias ainda não estão disseminadas como cultura das empresas.

Essa pesquisa trouxe muitas conclusões interessantes para entendermos melhor a qualificação da mão-de-obra em software no Brasil, tanto para orientar sobre as diretrizes educacionais como para os cursos de treinamento e reciclagem dentro das empresas.

Palavras-chave: Perfil dos desenvolvedores de software; Qualificação da mão-de-obra em software no Brasil; Diretrizes educacionais; Formação profissional.

ABSTRACT

The Software Learning Process has for a long time shown to be a very difficult process. Technology brings with it the appearance of a large set of tools that helps the construction of programs but this same technology development makes it difficult to assimilate and apply new techniques.

The objectives of this research is to verify the profile of the Brazilian Software Developer and which are the complementary education programs for their improvement, if we want better computer programmers to compete in the world software market. Investigations were based on information gathered from professionals working in the market and allow, through the analysis result, to understand the qualification, knowledge and working methodology of Brazilian Software Developers. From this result we can plan a strategy for the educational and economical policy, as well as for the recruitment and internal training in companies.

Characteristics of the Brazilian Software Developer, who are relatively young, dynamic and creative are important to establish interdisciplinary aspect joining software Engineering, Education, training and learning . The social class of the developers, in majority of the lower class, permit professional and social growth, but it can limit their development in the work market due to the deficiency found in their elementary and high school education.

To maintain the competitiveness in the software market it is important that the education and qualification of professionals be equivalent to those countries that compete with Brazil. Those deficiencies can be found either in the developers themselves or in the companies where they work.

In some implementations the Software Developers show lack of knowledge and the misuse of synchronization in processes. This can indicate that the applications do not have critical sections or that tools hide details that are not perceived by the developers.

The implementation of quality programs in the companies have encouraged the use of tools and methodologies to assure the software development in the newest technologies.

Synchronize the rolls of the universities and the enterprises in the educations processes is a fundamental requirement to complete contents to avoid gaps such as integrity maintenance lack, error detection, comments and documentation in the most used technologies of the market, giving support to the quality programs.

Key-words: Profile of the Brazilian software developer; Quality programs in the software development; Educations processes; Qualification of professionals.

Lista de tabelas

Lista de abreviaturas, siglas

ACM - Association for Computing Machinery
CPU - Central Processor Unit
CMM - Capability Maturity Model
CMMI - Capability Maturity Model Integration
IA - Inteligência Artificial
IAD - Inteligência Artificial Distribuída
IDE - *Integrated Development Environment*
LDO - Lei de Diretrizes Orçamentárias
MCT - Ministério de Ciências e Tecnologia
MEC - Ministério de Educação
MER - Modêlo de Entidade Relacional
MIT - Massachusetts Institute of Technology
SBC - Sociedade Brasileira de Computação
SEI - Software Engineering Institute
SEPIN - Secretária de Política de Informática do MCT
SGBD - Sistemas de Gerenciamento de Bancos de Dados
SI - Sistemas de Informação
TI - Tecnologia da Informação

Sumário

1 INTRODUÇÃO

1.1 Dificuldades Encontradas na Formação de Desenvolvedores de Software Aplicativo

São constantes as indagações acerca dos conteúdos, componentes, metodologias, didática e material de apoio utilizados na formação de desenvolvedores de software aplicativo, profissionais que codificam programas de computador em diversas áreas de negócio. Essas indagações estendem-se também ao processo de aprendizagem a que são submetidos estes profissionais.

Um dos principais problemas enfrentados na aprendizagem de programação é o primeiro contato que os estudantes têm com o desenvolvimento de programas. Este desenvolvimento envolve um conjunto de atividades que tem por objetivo solucionar problemas por meio de sistemas executáveis em computadores e as dificuldades encontradas são o desconhecimento da linguagem, a preocupação com a sintaxe, a obtenção de uma visão genérica do problema e a dificuldade em idealizar soluções e estabelecer uma representação lógica para resolvê-los [PROU2000].

Através de uma análise das principais conferências mundiais na área de *Computer Science Education* patrocinadas pela *Association for Computer Machinery* (ACM), pode-se notar que essas mesmas dificuldades são encontradas nas diversas instituições de ensino em todo o mundo [MEND2002].

Objetivando a compreensão de como o processo de formação de desenvolvedores de software aplicativo é relevante para o desenvolvimento tecnológico é importante investigar os fatores preponderantes para a aquisição e assimilação de conceitos

necessários na elaboração de programas de computador bem como identificam quais são os aspectos mais relevantes.

Primeiramente é preciso identificar a qualificação dos desenvolvedores de software aplicativo que atuam no mercado de trabalho: seu perfil, conhecimento, habilidades, o uso de ferramentas e sua produtividade.

Por meio de uma tabulação dos resultados obtidos, é possível realizar uma análise mais concreta sobre fatores que contribuiram para a formação de um bom desenvolvedor de software e estabelecer quais são as medidas mais adequadas para elevar o nível de qualidade de mão-de-obra no desenvolvimento de Software Aplicativo no Brasil.

1.2 Elaboração da Pesquisa

Nesta dissertação, a metodologia adotada para a pesquisa é feita através de preenchimento de questionário. A amostragem é constituida pelos desenvolvedores de software, tanto no nível sênior, como no nível pleno e junior, restringindo-se a profissionais de notória qualidade. A tabulação dos dados é efetivada de modo que todos os resultados possam ser revistos, reavaliados e, se necessário, novas questões possam ser feitas para refinar as conclusões finais apresentadas neste documento.

1.3 Critérios na Apresentação da Pesquisa

Para melhor entendimento e acompanhamento da elaboração do modelo lógico de dados e para que as análises sobre o seu desenvolvimento possam ser feitas com qualidade, seguem-se os seguintes critérios:

- Todas as informações obtidas para a pesquisa na proposta desta dissertação foram devidamente tabuladas para poderem ser comparadas a pesquisas semelhantes já existentes.

- As informações obtidas têm associadas a elas as suas respectivas fontes para auxiliar na identificação de possíveis erros.

1.4 Conteúdo do Documento

Objetivando fundamentar o trabalho, inicialmente é fornecida uma visão abrangente de alguns conceitos utilizados na aprendizagem de desenvolvimento de software e uma introdução a respeito do trabalho de pesquisa.

O capítulo dois enfoca o processo de aprendizagem, citando diversas formas na formação de desenvolvedores de software: aprendizagem por memorização, por instrução, por dedução, por indução, por agrupamento e por analogia.

O capitulo três fornece a descrição de algumas técnicas, ferramentas e métodos utilizados no processo de aprendizagem de desenvolvimento de software.

O capítulo quatro descreve os modelos e os métodos empregados na pesquisa.

O capítulo cinco descreve as perguntas do questionário e a filosofia que o norteia.

O capítulo seis descreve a tabulação e a classificação dos dados coletados, bem como a análise e a conclusão resultante do processo de investigação e, conseqüentemente, as considerações finais.

O capítulo sete faz uma análise crítica das respostas obtidas e descreve as constatações observadas.

O capítulo oito descreve as considerações finais sobre as análises das respostas obtidas e sobre as constatações observadas para o entendimento mais amplo sobre a qualificação da mão-de-obra em software no Brasil.

Dois apêndices completam a dissertação:

- A formação nos cursos superiores na área de computação e informática idealizada pelo MEC e

- Os cursos de nível superior na área de ciências da computação e sistemas de informação.

2 APRENDIZAGEM

Como esta dissertação ocupa-se em registrar ponderações, pesquisas e análises sobre a adequada formação e capacitação de desenvolvedores de software aplicativo, é necessária uma rápida abordagem sobre o processo de aprendizagem, mesmo não sendo este o foco principal do trabalho.

2.1 Conceituação de aprendizagem

Compreende-se aprendizagem como o processo que possibilita a obtenção de novos tipos de conhecimento, incluindo as habilidades motora e cognitiva. Um processo de aprendizagem fornece como resultado a descoberta de novos conceitos e novos fatos através de elementos obtidos por observação e/ou experimentação.

"A aprendizagem abrange uma ampla escala de fenômenos. Em uma extremidade do espectro, está o refinamento de habilidades. As pessoas melhoram a execução de suas tarefas através da prática. Quanto mais você anda de bicicleta ou joga tênis, melhor você fica. Na outra extremidade do espectro, está a aquisição de conhecimentos: ... O conhecimento é normalmente adquirido através da experiência ... pessoas também aprendem através de sua própria experiência na solução de problemas ... Depois de resolver um problema complexo, lembramos a estrutura do problema e dos métodos usados, podemos generalizar a nossa experiência e solucionar mais facilmente problemas relacionados ..." [RICH1993].

Neste sentido, o aprendizado passa a ser mais eficiente na medida em que possibilita a compreensão de um maior número de situações, através da aplicação mais abrangente dos conceitos obtidos.

É importante ressaltar a transformação decorrida em função do domínio e o uso da linguagem humana dentro do processo de aprendizado.

"... o momento de maior significado no curso do desenvolvimento intelectual, que dá origem às formas puramente humanas de inteligência prática e abstrata, acontece quando a fala e a atividade prática, então duas linhas completamente independentes de desenvolvimento, convergem... o uso de instrumentos pela criança durante o período pré-verbal seja comparável àquele dos macacos antropóides, assim que a fala e o uso de signos são incorporados a qualquer ação, esta se transforma e se organiza ao longo de linhas inteiramente novas ..." [VIGO1999].

Nesta análise, são identificadas as mudanças comportamentais no momento em que uma criança faz uso da linguagem escrita. Enfim, o aprendizado é comprovado através de expressão externa, seja ela oral, escrita ou prática.

2.2 Processo de aprendizagem

A aprendizagem é um fenômeno extremamente complexo, envolvendo aspectos cognitivos, emocionais, orgânicos, psicossociais e culturais. A aprendizagem é resultante do desenvolvimento de aptidões e de conhecimentos, bem como da transferência destes para novas situações.

O processo de aprendizagem é desencadeado a partir da motivação. Esse processo dá-se no interior do sujeito, estando, entretanto, intimamente ligado às relações de troca que o mesmo estabelece com o meio, principalmente, seus professores e colegas. Nas situações escolares, o interesse é indispensável para que o aluno

tenha motivos de ação no sentido de apropriar-se do conhecimento. Em outras palavras, a aprendizagem está intimamente ligada ao ambiente em que a pessoa se encontra. Nesse sentido, podemos afirmar que as condições sócio-econômicas e culturais têm uma grande influência sobre o processo de aprendizagem.

São várias as formas de aprendizagem. As mais tradicionais e básicas nas áreas de Ciências Exatas consistem na aprendizagem por memorização, instrução, dedução, indução, agrupamento e analogia.

A aprendizagem por memorização é eficiente no processo de solução de problemas, desde que não haja a necessidade de buscar ou deduzir novas informações. Basicamente, a aprendizagem por memorização é obtida através da repetição e o processo de organização e seleção de fatos é identificado como aprendizagem por instrução [MONA2003].

A aprendizagem por memorização geralmente conduz a aprendizagem por dedução. A aprendizagem por dedução baseia-se em explicações e faz uso de conceitos já adquiridos que pode resultar da repetição dos dados e do processo contínuo de organização e seleção de fatos que transforma esses fatos repetitivos em entendimento dos novos conceitos [DEJO1986].

A aprendizagem por indução é a estratégia na qual parte-se de um conceito específico para obter-se um conceito genérico. A aprendizagem por indução ocorre através de inferências indutivas sobre os conceitos específicos de um determinado domínio e a aprendizagem por agrupamento caracteriza-se pela capacidade de reconhecimento de padrões em um conjunto de elementos ou conceitos [RICH1993].

A aprendizagem por analogia pode ser definida como um processo que transfere o conhecimento de como realizar uma tarefa bem compreendida a outra menos compreendida através de uma “propriedade-chave” [MONA2003]. Um exemplo clássico é a analogia feita entre a oscilação dos preços no mercado financeiro e a movimentação física de uma montanha russa. Neste exemplo, a analogia é realizada através da comparação da oscilação dos preços, que consiste em fortes altas e baixas, ao movimento realizado pela montanha russa.

Existe um consenso de que as pessoas que trabalham em programação e desenvolvimento de software estão acostumadas a pensar e a agir de acordo com o paradigma cartesiano, baseado no raciocínio lógico, linear, seqüencial, exato e bem definido, deixando de lado suas emoções, a intuição, a subjetividade, o lado artístico da criatividade, a capacidade de ousar soluções não convencionais. Essas pessoas têm facilidade em lidar com matérias como física, matemática e tem dificuldade com biologia e desenho artístico. O raciocínio disciplinado tem sua relevância na implementação de soluções e a abstração de problemas em representações simbólicas com o respectivo domínio e manipulação destas representações são fatores chave na busca de soluções para os problemas.

Segundo os neurologistas que estudam cérebros, baseados nas imagens da ressonância magnética funcional, do PET/CT e da espectroscopia, os homens saem-se melhor em tarefas que envolvem cálculos, enquanto as mulheres são melhores em habilidades verbais. Isso deve implicar no fato que a maior parte dos desenvolvedores de software são de sexo masculino.

Essas afirmações são mitos ou tem um fundo de verdade? A pesquisa irá reforçar ou refutar essas afirmações.

2.3 Competências e Habilidades no Processo de Aprendizagem

Atualmente, o processo de aprendizagem está passando por uma transformação para se adaptar às novas realidades as quais a sociedade está passando. O principal enfoque desta mudança é o questionamento acerca da aquisição do conhecimento. Muito embora tenham sido introduzidos muitos recursos tecnológicos, o aprendizado concentra-se ainda em conteúdos e em estratégias para disseminar este conteúdo. Dentro deste contexto, o professor é visto como o detentor do conhecimento o qual deve ser repassado em sala de aula ao mesmo tempo que faz uso do livro didático para subsidiar este conhecimento.

A mudança na produção e apropriação do conhecimento inicia-se na mudança de postura do professor no desenvolvimento das atividades docentes. Ao invés de ser o detentor do conhecimento e fazer uso de um conjunto de recursos para transmiti-lo, o professor passa a ser um elemento mediador no desenvolvimento das grandes competências e habilidades que o aluno necessita para o seu convívio profissional e social. Essa mudança faz-se necessária para que o aluno abandone a condição de passividade dentro do processo ensino-aprendizagem, deixando de ser um mero “receptor” de informações.

Receber as informações sem conseguir associa-las a uma interpretação que possa ser aplicada a uma outra realidade é um dos paradigmas que pode ser modificado através do desenvolvimento de competências e habilidades no processo educacional.

Olhando sob esse ponto de vista, percebemos que um bom profissional não apenas tem de ter conhecimento de sua área específica, mas uma compreensão correta da realidade e das relações inter-pessoais também fundamentais para o sucesso em sua carreira.

Considerando esse novo enfoque para o sistema educacional, é necessário destacar quais são as competências e quais são as habilidades necessárias para a formação educacional, profissional e social. Inicialmente, é preciso definir-se o significado de competência que pode ser descrita como a capacidade de agir aficazmente em um determinado tipo de situação, apoiada em conhecimento [PERR1999]. Para que seja obtida uma determinada competência é necessário desenvolver em um conjunto de conhecimentos, atitudes, capacidades e aptidões que habilitam alguém para vários desempenhos na vida, identificados como habilidades. As habilidades, por sua vez, estão relacionadas à capacidade de poder realizar determinada tarefa, seja física ou mental. Identificação de variáveis, relacionamento de informações, compreensão de fenômenos, análise de situações, sintetização, inter-relacionamento de conceitos, desenvolvimento crítico e flexível, aquisição e transmissão de informações, compreensão dos princípios das tecnologias, entendimento de fundamentos científicos, desenvolvimento da criatividade são alguns exemplos de habilidades.

As diretrizes do MEC orientam cinco competências: domínio de linguagens, compreensão de fenômenos, construção de argumentações, solução de problemas e elaboração de propostas.

A competência possibilita o envolvimento de conhecimento e a capacidade de identificar os recursos necessários para que se possa obter resultados criativos e eficazes para a solução de problemas.

Muito embora competência seja um termo mais abrangente, englobando uma série de habilidades, não existe exclusividade na relação entre competência e habilidade. Uma habilidade pode contribuir para uma ou mais competências.

Desenvolver competências e habilidades oferece uma orientação diferente para o processo educacional. Ao invés de centrar a atenção nos conteúdos para atender a um determinado objetivo, desenvolver competências e habilidades representa a centralização da aprendizagem no próprio indivíduo. Neste aspecto, as instituições de ensino e educadores devem efetuar a identificação e o planejamento das competências e habilidades inseridas nos cursos e disciplinas.

A área de Computação e Informática é bastante ampla e pode envolver uma série de competências e habilidades. Uma questão preponderante neste aspecto é a definição mais precisa a respeito das atividades de um desenvolvedor de software aplicativo que é o enfoque deste trabalho. Um desenvolvedor tanto pode propor soluções de problemas através de algoritmos e implementá-los em sistemas de computador como pode idealizar soluções de problemas mais abrangentes envolvendo uma análise mais detalhada e minuciosa do processo como um todo. Para ambos os casos, a construção de argumentações, soluções de problemas e elaboração de propostas são competências primordiais que devem ser desenvolvidas. O domínio de linguagens é uma competência que se faz necessária na medida em que a tarefa do desenvolvedor tem um escopo mais abrangente.

A tabela 1 estabelece uma relação entre competências e habilidades necessárias para a formação de um desenvolvedor de Software Aplicativo.

Competência	**Habilidade**
Domínio de Linguagens	
Construção de Argumentações	Relacionamento de Informações
	Inter-relacionamento de Conceitos
	Desenvolvimento Crítico e Flexível
	Aquisição de Informações
	Transmissão de Informações
Solução de Problemas	Identificação de Variáveis
	Relacionamento de Informações
	Análise de Situações
	Sintetização
	Inter-Relacionamento de Conceitos
	Desenvolvimento de Criatividades
Elaboração de Propostas	Relacionamento de Informações
	Sintetização
	Inter-relacionamento de Conceitos
	Desenvolvimento da Criatividade
	Transmissão de Informações
	Compreensão dos Princípios das Tecnologias

Tabela 1 – Classificação de Competências e Habilidades para um Desenvolvedor de Software Aplicativo. (Adaptado de [PERR2000])

2.4 Conclusão

Este capítulo demonstra que o processo de aprendizagem não é único. Como o cérebro transforma informações em conhecimento ainda é um mistério. Provavelmente o aprendizado ocorre através de todos os processos supracitados. É possível que, em uma certa idade e em determinada área, um dos processos é mais eficiente do que outros.

Com as recentes pesquisas sobre o funcionamento do cérebro, a Teoria das Inteligências Múltiplas, a avaliação das aptidões cerebrais dominantes e as técnicas que foram criadas para acelerar a aprendizagem tornou-se muito mais fácil aprender e gravar na memória o que estudamos.

As modernas propostas didáticas consistem em escolher o processo que é mais eficiente para um certo grupo de alunos explorando o desenvolvimento de competências e habilidades mais adequadas para sua formação profissional.

3 TÉCNICAS, MÉTODOS E FERRAMENTAS UTILIZADAS NA APRENDIZAGEM DE DESENVOLVIMENTO DE SOFTWARE

Após algumas reflexões sobre o processo de aprendizagem, faremos uma análise das dificuldades de aprendizagem de software tanto na educação formal nas Universidades como os problemas nos cursos de treinamento de curta duração ministrados pelas empresas de treinamento.

Numa análise mais específica sobre a aprendizagem de desenvolvimento de software, conclui-se que os profissionais do mercado podem ser divididos em duas categorias:

1- pessoas que tiveram uma educação formal em software. Essas pessoas aprenderam primeiro os métodos, a formulação dos problemas, as soluções e a linguagem de programação e, posteriormente, tiveram contato prático com o desenvolvimento de software em seu trabalho.

2- pessoas que trabalham com software, mas não tiveram uma educação formal. São pessoas que conheceram primeiro os desafios de solucionar um problema e foram buscar os conhecimentos necessários através de treinamento *in-house* com seus colegas de trabalho.

3.1 Dificuldades Encontradas no Aprendizado de Desenvolvimento de Software nas Universidades.

Ensinar a desenvolver software tem sido uma constante preocupação e indagação, considerando o surgimento de novas linguagens e a constante evolução tecnológica.

Para que se possa operar uma completa análise sobre o assunto, torna-se necessário, em primeira instância, a compreensão do que vem

a ser um software e como é desenvolvido o processo de sua construção. Como se define a atividade de programar o computador?

"Programar é fornecer um conjunto de instruções que determina ao computador os passos que deve executar em determinado processamento." [HABE1986].

Um programa representa um algoritmo que pressupõe resolver um problema computacional, ou seja,

"... uma seqüência finita de instruções, cuja execução produz um resultado que é a solução de um problema..." [BARB2001].

Antes de começar a solucionar o problema, é necessário conhecer as especificações e os requisitos que possam expressar as características e restrições do produto de software do ponto de vista do usuário. "O ponto de partida do desenvolvedor é a descrição do sistema, onde o analista procura escrever em português claro e de forma pormenorizada tudo aquilo que deve ser feito." [HABE1986].

Neste aspecto, o desenvolvedor obtém do analista uma descrição do que deve ser feito, ou seja, dos requisitos do sistema e, com base nessa descrição, elabora uma solução através de algoritmos que podem ser implementados em sistemas de computador.

A elaboração da solução faz parte da atividade de desenvolvimento e conseqüentemente é um dos problemas e dificuldades existentes em seu aprendizado.

Além deste, foi identificado o problema na aprendizagem de programação nas conferências mundiais na área de *Computer Science Education*, patrocinadas pela *Association for Computer Machinery*

(ACM) que relacionam um conjunto de prováveis causas para esta dificuldade. Entre as causas estão: o alto nível de abstração requerido, a necessidade de alto nível de conhecimento, técnicas para a solução de problemas e o alto grau de complexidade da sintaxe das linguagens sem representações visuais de seus algoritmos [MEND2002].

Foram relacionados também uma série de problemas de aprendizagem de desenvolvimento de software relacionados à maneira como o processo é realizado. Entre os problemas, destacam-se a existência de turmas com conhecimentos e potencialidades heterogêneas, a existência de turmas numerosas comprometendo o acompanhamento individualizado do aluno, a utilização de métodos tradicionais de ensino utilizando recursos e materiais estáticos e a exigência de estudo baseado na prática [MEND2002].

Observou-se que os problemas encontrados na aprendizagem de desenvolvimento de software complementam a própria dificuldade de dimensionar o que se pretende solucionar, encontrar soluções efetivas, transformar a solução encontrada em procedimentos a serem adotados e estabelecer um raciocínio lógico capaz de resolver problemas [PROU2000].

A idéia no ensino de software é partir de premissas simples, segundo a proposta da filosofia do construtivismo, que passa a construir passo a passo a solução. Lisbete M. Barbosa cita no seu livro dois princípios no processo de ensino e aprendizagem de algoritmos [BARB2001]:

- Aprender é desenvolvimento. É um processo de adaptação em que o estudante constrói novas representações e modelos da realidade;

- A força motora da aprendizagem é a reflexão sobre as representações construídas e por meio da descentralização da experiência o estudante pode negociar os significados.

Em se tratando de aprendizagem de algoritmos, o aspecto mais relevante é a vivência de experiências, através das quais o aluno consiga generalizar conceitos e padrões, criando os seus próprios modelos.

"Uma visão construtivista da aprendizagem sugere uma abordagem do ensino que oportunize aos alunos experiências concretas, contextualmente significativas, nas quais eles possam buscar padrões, levantar suas próprias perguntas e construir seus próprios modelos, conceitos e estratégias" [FOSN1998].

A segunda etapa consiste no entendimento e desmembramento da solução. Muitos autores concordam que a abordagem *top-down* é mais motivadora e produz melhor resultado [BARB2001].

A forma como trabalha-se com os alunos no processo de aprendizagem é outro fator bastante relevante. Uma das orientações é o trabalho em grupo.

Os trabalhos em grupo, normalmente associados a resultados positivos na aprendizagem, transferência e desenvolvimento de conhecimento, muitas vezes não apresentam resultados propícios para a criação de um ambiente capaz de auxiliar no exercício do aprendizado colaborativo [SUTH1998]. No modelo em grupo, há a necessidade de um orientador que repasse seus conhecimentos ao grupo.

3.2 Criação de Ambientes Apropriados para o Aprendizado de Desenvolvimento de Software Aplicativo

Uma vez compreendida a resolução do problema através da construção do algoritmo, a próxima etapa consiste na aprendizagem da linguagem de programação.

Outro importante foco de análise para diagnosticar os entraves do processo de ensino-aprendizagem de desenvolvimento de software é a criação de ambientes propícios e facilitadores do processo. Tal preocupação justifica-se pela necessidade de obtenção de melhores resultados. Uma possibilidade, como já fora citado anteriormente, é a aprendizagem em grupo, que ocorre através do uso e aplicação da teoria do conflito sócio-cognitivo, em que são evidenciadas as controvérsias, levantados os diversos pontos de vista do grupo e amplamente discutidas as alternativas e propostas de soluções [CONS2000].

A criação de um ambiente que possa vir a auxiliar o aprendizado envolve, entre outros requisitos, o uso e aplicação não só de métodos e técnicas, mas também o uso de ferramentas desenvolvidas com o objetivo de proporcionar ao aluno melhor visualização e acompanhamento, seja através de testes ou animação de algoritmos.

Aprender a programar significa ser capaz não só de construir mecanismos através de instruções em programas de computador, mas também ser capaz de fornecer explicações [SOLO1986]. Estes mecanismos e explicações fazem parte do processo de soluções de problemas. As instruções de computador indicam apenas como um problema pode ser solucionado dentro de um contexto mais amplo, que origina o desenvolvimento e a elaboração da solução.

O uso e a aplicação de testes em máquinas propicia maior qualidade não só dos programas, como também melhor qualificação dos desenvolvedores de software aplicativo. Os testes propiciam ainda uma revisão da codificação e facilitam o aumento de qualidade de um programa [ZELL2000].

O uso de ferramentas adequadas também é um importante facilitador e colaborador no processo de aprendizagem de desenvolvimento de software, isto é, se utilizadas convenientemente e dentro de um contexto apropriado.

Entre as características básicas das ferramentas de apoio à aprendizagem de desenvolver software, destacam-se a visualização e animação de algoritmos e a possibilidade de verificação, por parte dos alunos, de como determinados grupos ou elementos de dados se comportam e afetam o funcionamento de um algoritmo. Outro aspecto importante é a possibilidade de simulação do algoritmo em tempo real com uma melhor visualização e compreensão de seu comportamento em determinadas situações.

Em última instância, a aprendizagem com o auxílio de ferramentas que envolvam animações proporcionam um maior interesse por parte dos alunos, ao contrário do quando se trabalha através de materiais convencionais [MEND2002]. Programas de depuração (*debuggers*) são exemplos marcantes que podem esclarecer ao desenvolvedor dúvidas impossíveis de serem dirimidas sem o auxílio do instrumento.

A utilização de ferramentas que contemplam ambientes mais simples e com interfaces mais amigáveis tem por objetivo encorajar os alunos a testar, avaliar e corrigir os seus próprios algoritmos, atividades consideradas essenciais para a aprendizagem de

programação. Mendes sugere uma classificação destas ferramentas e destaca a existência de mini-linguagens, ambientes de desenvolvimento controlados, mundos programáveis e ferramentas de animação [MEND2002].

3.3 Considerações sobre cursos de treinamento ministrados internamente pelas empresas ou ministrados pelas empresas de treinamento.

Considerando que a introdução do curso de Ciências de Computação é relativamente recente e que a demanda dos profissionais na área tem sido muito maior que a mão de obra formada pelos cursos de Ciências da Computação de qualidade no país, principalmente na última década, é inegável a contribuição dos cursos de treinamento ministrados tanto internamente como pelas empresas de treinamento e capacitação do setor.

Outro fator importante é a introdução de novas metodologias, ferramentas, linguagens e tecnologias. Mesmo as pessoas com uma ótima formação nas Universidades, após alguns anos de trabalho, têm que de submeter-se a um novo treinamento, sob risco de ficar à margem do mercado de trabalho.

Devido à heterogeneidade da classe e à restrição de tempo, esses cursos, em geral, não se atentam aos conceitos mais abstratos e teóricos, mas tem o objetivo de curto prazo: introduzir a nova tecnologia e capacitar os participantes a resolverem problemas prementes das empresas. Esses cursos, normalmente intensivos, não permitem uma reflexão das matérias lecionadas e o amadurecimento, dessa forma e a absorção real dos conhecimentos só ocorre com as aplicações tecnológicas ao longo do trabalho.

3.4 Novos Enfoques para o Ensino

Uma mudança de postura em relação ao ensino deve ser compreendida de forma abrangente. Um dos fatores mais relevantes é a introdução de novas metodologias, ferramentas e tecnologias no ensino. É importante que o desenvolvedor faça uso intensivo de ferramentas e componentes oferecidos no mercado ou disponíveis na Internet ao invés de criar um novo procedimento para cada problema encontrado [HABE1991].

Considerando que o processo educacional é consolidado através do conhecimento e como existem diferenças conceituais entre conhecimento e informação, é necessário estabelecer a relação entre eles.

A informação é uma representação simbólica e formal de fatos ou idéias, enquanto o conhecimento é a absorção dessas informações, através das experiências vivenciadas, transformadas na forma de conceitos.

As informações oferecidas, através de novos meios e novas tecnologias sugerem uma mudança na postura educacional. Ao contrário de Constantino Gonzáles [CONS2000], Tércio Pacitti sugere um novo ambiente educacional, no qual a sala de aula e o mestre servirão de apoio e orientação para os estudos. Esse ambiente possibilitará uma aprendizagem individualizada, dependendo do potencial de cada um e respeitando o ritmo e as necessidades distintas [PACI2003]. Por outro lado, esse ambiente permite que os questionamentos e as dúvidas dos alunos possam ser esclarecidos através da interação com o professor e com os demais participantes, ainda que virtualmente.

Outra mudança na postura educacional está relacionada ao ensino da Programação Orientada a Objetos. Neste aspecto, Pacitti refere-se à mesma e indica que a maioria das escolas ensina lógica de programação de forma tradicional, apontando a existência de dificuldades na real assimilação destes novos conceitos, especialmente para em quem se formou há 10 ou 15 anos. [PACI2003]

A pesquisa que será realizada junto a gerentes de departamentos técnicos e desenvolvedores esclarecerá quais metodologias e ferramentas deverão ser enfatizadas para a formação de profissionais de alto nível e ainda como aprimorar o ensino nos cursos de Ciências da Computação.

Depois da exposição acima, podemos concluir que não há consenso nos métodos de ensino de desenvolvimento de software. Parece-nos que o método mais comum é ensinar primeiro o projeto de algoritmo e depois a linguagem de programação. Entretanto nem neste aspecto existe consenso, pois, em muitos lugares, ensina-se os alunos a programar diretamente com uma determinada linguagem, supondo que o raciocínio lógico e a formulação do algoritmo é intuitivo ou já tenham sido adquiridos anteriormente, desde o curso primário.

Esse quadro só fica mais complexo com a introdução das novas linguagens e ambientes de programação. As ferramentas que auxiliam a programação nem sempre são encontradas e usadas.

A chegada da Tecnologia Orientada a Objeto e a WEB trouxeram mais dificuldades para o dia-a-dia dos alunos, pois os conceitos antes totalmente lineares e sequenciais, agora passam a ser conceitos de objetos e relações entre eles.

3.5 Engenharia de Software e o seu impacto na formação dos desenvolvedores de Software.

A Engenharia de Software exerce, atualmente, um papel preponderante face à sua evolução ao longo dos últimos anos. Essa evolução abrange o uso de um conjunto de conceitos e recursos nas mudanças da arquitetura dos softwares para atender diferentes demandas desde a confecção de um trabalho escolar até a constituição de produtos de grande complexidade e de grande porte, que fazem acesso aos dados através de Sistemas Gerenciadores de Banco de Dados, ambientes gráficos e, mais recentemente, a Internet.

A Engenharia de Software pode ser definida como “o estabelecimento e uso de sólidos princípios de engenharia para se obter economicamente um software confiável e que funcione eficientemente em máquinas reais” [NAUR1969]. A Engenharia de Software engloba três elementos fundamentais: “métodos, ferramentas e procedimentos”.

Esses três elementos possibilitam “ao gerente o controle do processo de desenvolvimento do software e oferecem ao profissional uma base para sua construção com alta qualidade produtivamente” [PRES1995]. Segundo o autor, um bom desenvolvedor que conhece e faz uso da Engenharia de Software está habilitado a produzir sistemas de alta qualidade.

Além disso, com a introdução do conceito de orientação a objeto e de “agentes autônomos”, os produtos de software passam a ter em sua arquitetura forma distribuída de “inteligência” que permite a realização de processamentos de informações semelhantes ao do raciocínio humano. O uso de redes neurais artificiais para o reconhecimento de padrões é um exemplo claro deste tipo de técnica.

Essas novas técnicas também estão modificando radicalmente a forma de construir-se sistemas de computador [PRES1995].

No aspecto de formação e educação dos desenvolvedores de software o impacto é ainda maior. Desde 1996, iniciou-se no Brasil, primeiramente pelo Softex, o esforço de introduzir a Engenharia de Software nas empresas brasileiras. Atualmente, a disciplina de Engenharia de Software é oferecida quase como disciplina obrigatória em todos os cursos de Ciência de Computação, Engenharia de Computação e Sistemas de Informação do país. Essa disciplina pode ser optativa em todos os cursos de Engenharia. Nela são introduzidos conceitos de produtos e processos de software, paradigmas de desenvolvimento, métricas para estimativas e gerenciamento e qualidade de software.

Segundo a Secretaria de Política de Informática do Ministério da Ciência e Tecnologia por intermédio do manual de Qualidade e Produtividade no Setor de Software Brasileiro (2002), as práticas de Engenharia de Software que tiveram destaque no desenvolvimento e manutenção de software foram:

"... modelagem de dados 70%, controle de versão 69%, especificação de projetos 65%, de requisitos e de programas 61%, projeto da interface com o usuário 57%, estimativa de custos 55%, métodos orientados a objetos 54% e prototipação 51% ..."

Esses dados demonstram claramente a preocupação das empresas de software na adoção de práticas que aumentem a produtividade e a qualidade de seus produtos e que as práticas da Engenharia de Software devem fazer parte integrante na formação dos desenvolvedores de software aplicativo.

É importante salientar que os profissionais da área de computação e de informática não recebem treinamento suficiente e adequado em novas técnicas de desenvolvimento de software. As contribuições teóricas importantes da Ciência de Computação ainda não têm influenciado a prática do desenvolvimento de software no país. “Em certas organizações, uma branda forma de anarquia ainda reina. Cada pessoa aborda a tarefa de ´escrever programas´ com a experiência derivada de esforços passados. Algumas pessoas desenvolvem uma abordagem ordeira e eficiente ao desenvolvimento de software por meio de ensaio e erro, mas muitas outras desenvolvem maus hábitos que resultam em qualidade e manutenibilidade deficientes” [PRES1995].

Os grandes entraves para implantação dos procedimentos da Engenharia de Software, além do despreparo dos profissionais, tem sido a falta de tempo devido a urgência dos lançamento de produtos, falta de recursos para implantação e o alto custo para obtenção da certificação.

Um dos objetivos desta pesquisa é verificar se estas técnicas de desenvolvimento de software estão sendo efetivamente utilizadas nas empresas, se os profissionais tem conhecimento e domínio dessas técnicas e principalmente se os conceitos de Engenharia de Software estão afetando na melhoria da disciplina pessoal em desenvolvimento e programação dos desenvolvedores ou todas essas técnicas permanecem apenas no nível gerencial.

3.6 Conclusão

Pela análise feita, vemos que os cursos de Ciências de Computação nas Universidades de primeira linha, pela exigência do MEC de pesquisas e publicações científicas, têm dado prioridade à formação teórica e à conceituação do raciocínio em detrimento da atualização tecnológica e das ferramentas mais usadas no mercado. Sendo a computação a área tecnológica mais dinâmica da ciência, o MEC, através da SBC, tem feito discussão permanente do currículo, não só no sentido tecnológico, mas aprofundando mais na formação conceitual do estudante. O problema é que o aluno formado desta maneira precisa de um período de adaptação e treinamento para ser produtivo nas empresas. Por outro lado, os cursos de treinamento de curta duração são essencialmente orientados para informação, calcados em cima de produtos, metodologias e ferramentas, muitas vezes carecendo de uma visão holística e crítica da área. As pessoas que passam por esses cursos tem a sensação de conhecimento fragmentado, faltando-lhe a capacidade de análise e, principalmente, a capacidade de síntese.

É importante que não haja um distanciamento entre a universidade e o mercado de software, mas sim um forte investimento em pesquisas para a base do conhecimento teórico e conceitual em perfeita sintonia com a tecnologia utilizada pelo mercado.

.

4 OS FUNDAMENTOS DA METODOLOGIA DESSE TRABALHO

Como vimos, o objetivo deste trabalho é entender o perfil do desenvolvedor de software aplicativo no Brasil e estabelecer os fatores que forjaram este perfil profissional.

Queremos saber de onde vieram os desenvolvedores de software, como foi sua formação, o que eles sabem, quais são as suas competências e como é desenvolvido seu trabalho na empresa.

Para responder essas questões, foram usados dois questionários distribuídos entre os desenvolvedores escolhidos aleatoriamente nas empresas. As questões podem ser agrupadas em quatro categorias:

1º.Grupo – Conhecer o pano de fundo e os fatores que contribuem para a formação do profissional desenvolvedor de software.

As perguntas são do tipo: Quais são os background culturais e sócio-eonômicos dos profissionais de software? Como se processou a sua educação primária e secundária? E quais são seus pendores naturais? Como foi o curso universitário? Que tipo de curso e qual a universidade?

2º. Grupo – Conhecer seu perfil profissional, o seu conhecimento sobre todos os aspectos do software, as áreas específicas de aplicação, as técnicas de programação, o domínio das linguagens e das ferramentas.

As perguntas tentam extrair e testar seu conhecimento em sistemas operacionais, banco de dados, redes de computadores, capacidade de análise, algoritmos e linguagens. As perguntas

não são de modo direto, deixando o profissional julgar se ele domina ou não o objeto, mas de modo indireto para que tenhamos uma avaliação mais objetiva de seu conhecimento.

3º. Grupo – Testar o profissional sobre o seu conhecimento no processo de desenvolvimento, principalmente nos aspectos das práticas da Engenharia de Software.

As perguntas desse grupo definem quais etapas ele aplica dentro do ciclo de vida do produto, como definido pela Engenharia de Software. Onde ele adquire o conhecimento para o desenvolvimento de software? Na escola, nos cursos de treinamento, no trabalho ou é auto-didata? Qual é a sua produtividade? Qual a influência dos processos de certificação de qualidade nesse contexto?

4º. Grupo – Conhecer seu ambiente de trabalho.

Quanto tempo ele gasta por ano em cursos de treinamento, em atualização e em estudos auto-didatas? Qual é o tipo de controle de qualidade que a empresa exerce sobre seu trabalho? Qual é o investimento que a empresa faz para melhoria de qualidade de seus profissionais?

As pesquisas foram feitas nas empresas que desenvolvem principalmente Software Aplicativo. Uma vez traçado o perfil dos profissionais, podemos analisar sua formação acadêmica, sua qualificação, sua produtividade e suas deficiências, sugerir as possíveis mudanças não apenas na política educacional do MEC no país, como no conteúdo programático das disciplinas lecionadas e nos métodos de ensino nas áreas de computação das universidades.

Outro possível resultado prático dessa pesquisa é dar subsídio para elaborar uma política governamental estimulando a formação de mão de obra qualificada na área de software, em parceria com as empresas do setor.

5 PESQUISA PARA TRAÇAR O PANORAMA DO PREFIL DOS PROFISSIONAIS NAS EMPRESAS DE DESENVOLVIMENTO DE SOFTWARE

5.1 O escopo da Pesquisa

A população da pesquisa foi selecionada aleatoriamente nas empresas de software, entre profissionais que realmente estão envolvidos no processo de desenvolvimento e que efetivamente implementam software aplicativo.

O número das empresas participantes desta pesquisa é 25. Foram enviados 400 questionários e, entre entrevistas e retornos, obtivemos 162 respostas. O número de profissionais entrevistados de cada empresa varia entre 2 e 15.

Além das informações escolares, profissionais e pessoais do desenvolvedor, foi tomado cuidado especial com a formulação das perguntas que possam traçar o perfil de conhecimento e competência do profissional, pois o grau de conhecimento e competência não pode se basear simplesmente em sua própria declaração.

Para formular essas perguntas partirmos de alguns requisitos que seriam altamente desejáveis para um competente desenvolvedor de software. Esses requisitos são resultado de consenso de alguns dos gerentes das mais conhecidas empresas de software do Brasil.

Para saber se o profissional possui certos requisitos, algumas perguntas foram feitas de modo indireto, na forma de teste de conhecimento.

5.2 Análise dos dados da Pesquisa

As questões respondidas na pesquisa foram consolidadas para a construção de tabelas que possibilitem uma melhor visão dos resultados da pesquisa. Numa segunda etapa, essas respostas foram agrupadas em áreas para que se entenda melhor a correlação entre as respostas.

5.3 Amostra

Os dados de amostra da pesquisa foram definidos de forma dirigida, selecionando-se empresas que mantêm funcionários na área de programação e desenvolvimento de software, cujos produtos e serviços são amplamente reconhecidos pelo mercado. As pessoas que responderam ao questionário são necessariamente funcionários que contribuem na definição e desenvolvimento de seu conteúdo.

A pesquisa foi realizada através da distribuição de questionários impressos e entrevista com parte dos selecionados, tanto para gerentes como para analistas e programadores.

5.4 Da Pesquisa

5.4.1 Formulação do Questionário

O questionário tem como objetivo principal buscar a identificação e o perfil dos bons desenvolvedores de software aplicativo e a investigação das formações básica e complementar destes profissionais.

Para atingir esse objetivo foram utilizados dois questionários envolvendo questões dirigidas. O primeiro questionário visa mais

entender o conhecimento e a competência do profissional e as características da empresa em que ele trabalha. O segundo questionário visa coletar os dados a respeito da formação do desenvolvedor de software aplicativo e seu perfil pessoal.

5.4.2 Validação do Instrumento de Pesquisa

Uma simulação desse instrumento de pesquisa foi realizada antes de enviar o questionário para profissionais de software, visando verificar, validar, avaliar e aperfeiçoar essa pesquisa.

Esta verificação incluiu a eliminação de questões duplicadas ou que gerem dúvidas entre os participantes da pesquisa. A validação do questionário teve como base a simulações nas diversas empresas e principalmente consultando profissionais de competência reconhecida no mercado.

No Brasil estima-se que haja 2.400 empresas (de um total de 5.000 do setor) que efetivamente desenvolvem software aplicativo, segundo o estudo A Industria de Software no Brasil (base SEPIN/MCT). Mais de 82% são micro empresas e apenas 2% são de grande porte o que leva a uma baixa média de desenvolvedores de software por empresa, ou seja, seis técnicos em cada uma (de um total de 32 funcionários), equivalente a 19%, dando um total de 14.400 pessoas que efetivamente desenvolvem. Foram enviados 400 questionários, obtendo-se um retorno de 162. Como pilôto para esta dissertação a amostra é suficiente [MCT2002].

5.5 Procedimentos da Pesquisa

Para a realização da pesquisa foi adotado um procedimento idêntico em todas as empresas. Foram distribuídos questionários em

formulários impressos com um envelope de retorno para gerentes e desenvolvedores de software aplicativo. Tanto os gerentes e os líderes de equipes quanto os desenvolvedores fizeram parte da pesquisa respondendo igualmente ao questionário.

5.6 Elementos Básicos do Questionário

Com relação aos questionários, foram elaboradas perguntas abertas e fechadas (múltipla escolha e exclusivas entre si). As questões são apresentadas a seguir, assim como o comentário sobre o objetivo e a relevância de cada questão aplicada. Esses comentários não foram incluídos na pesquisa.

5.7 Questionários aplicados nas empresas

5.7.1 Formação de Desenvolvedores de Software Aplicativo no Brasil: Competências, Deficiências, Causas e Soluções

Prezado Gerente/Coordenador, Programador

Estamos desenvolvendo uma Dissertação de Mestrado do curso de Engenharia da Computação do Instituto de Pesquisas Tecnológicas com o tema acima, que certamente é do seu interesse.

O objetivo maior deste trabalho é darmos sugestões de como o ensino desta disciplina pode ser melhorado no futuro.

Para tanto sua colaboração é muito importante.

Basta responder o questionário anexo.

Agradecemos antecipadamente e solicitamos o seu e-mail para podermos, em breve, enviar lhe uma cópia da dissertação.

Obs.: As respostas com letras (a, b c, etc) são de múltipla escolha. As respostas com ( ) são exclusivas entre si.

Importante: Todas as informações aqui colhidas serão utilizadas exclusivamente para a elaboração da dissertação, com caráter confidencial e não serão utilizadas para nenhum outro fim.

( ) Gerente ( ) Analista ou Programador

1) Seu nome (facultativo)

---

2) Seu e_mail (facultativo)

Estas duas perguntas tem como objetivo apenas identificar o entrevistado e a resposta é opcional, caso o entrevistado queira ficar oculto.

3) O seu primeiro contato com programação foi
   a) na escola
   b) através de cursos extra curriculares
   c) no trabalho
   d) através de literatura procurando aprender sozinho

É importante saber como ele aprendeu programação, pois quando a aprendizagem é feita de modo informal pode acarretar certas deficiências, tanto em conhecimento como em disciplina.

4) A sua formação é:

a) ensino médio (2º grau)
b) superior incompleto
c) superior completo
d) mestrado
e) doutorado

Caso tenha concluído o nível superior informe o curso e a instituição ?

_____

No exterior, 70% dos pós-graduados trabalham em empresas, enquanto no Brasil, 90% dos que tem títulos de mestres e doutores trabalham em universidades. Queremos constatar se isso é também verdade em informática. Queremos saber também a porcentagem dos profissionais em software que não possuem curso superior.

5) Um bom programador é aquele que:
   a) consegue desenvolver um aplicativo
   b) trabalha com linguagens de baixo nível
   c) desenvolve rotinas para Softwares como Sistemas Operacionais e Compiladores
   d) trabalha com programação orientada a objeto

Aqui queremos saber qual é a visão de um programador. Cremos que todos que responderam à pergunta julgam-se um bom programador ou em vias de ser um. A resposta revela o que ele acha ser mais importante para ser um bom programador.

6) De que forma você acredita que melhor aprendeu sobre programação?
   a) por meio de exposição teórica do professor
   b) com exercícios práticos em laboratório
   c) analisando códigos de programas já elaborados

d) na prática do dia-a-dia de trabalho

Existe ainda uma polêmica sobre o melhor método para ensino de programação e o desenvolvimento lógico e intelectual do programador. As respostas indicarão, sob a ótica do profissional, qual foi o método que mais influenciou no aprimoramento da atividade profissional.

7) As aplicações que você desenvolve tem:
   a) laços de repetição
   b) desvios incondicionais
   c) desvios condicionais
   d) funções reutilizáveis
   e) todas as anteriores
   f) nenhuma das anteriores

Esta questão visa verificar o nível de conhecimento e competência do desenvolvedor. O uso de funções reutilizáveis, por exemplo, indica um nível técnico mais elevado.

8) As aplicações que você desenvolve incorporam rotinas de manutenção de integridade para detecção de erros?

   ( ) Sim ( ) Não ( ) Não Sei

Também aqui procura-se verificar o nível de programação empregado nas aplicações. “Sim” indica que o profissional conhece e usa. “Não” indica que o profissional conhece, mas não usa. “Não sei” indica que o profissional desconhece totalmente a técnica.

9) Você programa aplicações concorrentes com o uso de estruturas do tipo semáforos, monitores ou similares para controle do acesso de regiões críticas?

( ) Sim ( ) Não ( ) Não Sei

Essa pergunta também procura verificar o nível de programação empregado nas aplicações. Mostra, entretanto, uma atuação em questões mais intrincadas de programação. “Sim” indica que o profissional tem uma competência profissional elevada, atua nas áreas que envolvem problemas críticos de programação, acesso a memória e integridade de banco de dados. “Não” indica que o profissional não atua na área, mas conhece o assunto. “Não sei” indica que o profissional desconhece totalmente a técnica.

10) As suas aplicações utilizam ferramentas automáticas de geração de código, como construtores automáticos de analisadores léxicos ou semânticos, estilo lex e yacc?

( ) Sim ( ) Não ( ) Não Sei

Existe um trabalho intenso dos fabricantes de software em propagar o uso de ferramentas automáticas de geração de código. O uso ou não das ferramentas não indica a competência do profissional, mas a política das empresas e o conhecimento do profissional em adotar ou não o uso dessa ferramenta.

11) Você usa alguma metodologia ou técnica (métricas) para avaliação do desenvolvimento de sistemas ou softwares?
   a) Análise de Pontos por Função
   b) COCOMO
   c) PMBOK (Gerencia de Projetos)
   d) ISO9126
   e) Nenhuma
   f) Outra(s)

Em caso de outra(s), indique qual(is)

Essa pergunta e as perguntas seguintes (12,13, 14 e 15) fazem uma análise da aplicação do uso da Engenharia de Software no processo de desenvolvimento tanto no perfil pessoal do profissional como da empresa na qual atua. Essa pergunta específica mostra a maturidade tanto do profissional como da sua empresa.

12) Em seu processo de desenvolvimento você faz uso de:

a) UML
b) Fusion
c) Análise Estruturada
d) MER (Modelo de Entidade Relacional)

Embora as quatro opções não sejam do mesmo nível, porque quem usa Fusion pode englobar o uso da UML e o uso da MER independe das outras opções, esta pergunta reflete o conhecimento e o uso das técnicas mais recentes da Engenharia de Software. Caso o profissional deixe de assinalar pelo menos uma das opções significa que não usa nenhuma das técnicas acima citadas.

13) Em sua opinião, o entendimento dos requisitos do software a serem desenvolvidos tem como origem:

a) o detalhamento da descrição dos próprios requisitos
b) as conclusões obtidas a partir das reuniões com os clientes
c) as conclusões obtidas a partir das reuniões com a equipe de desenvolvimento
d) a orientação da chefia

Essa pergunta reflete o procedimento das empresas no que se refere à análise de requisitos.

14) Em se tratando do processo de desenvolvimento, você representa os algoritmos:

a) através do uso da UML
b) utilizando as técnicas de DFD
c) utilizando uma meta linguagem
d) através de um fluxograma

Essa pergunta reflete tanto a norma das empresas no desenvolvimento como o procedimento no desenvolvimento do profissional.

15) Em relação à documentação do software aplicativo, você costuma efetuar:

a) comentários por linha depois de cada comando inserido em um método
b) comentários relativos a uma classe de objetos
c) comentários para cada método
d) documentação somente depois de ter o programa funcionando

Uma documentação é o mínimo que toda empresa exige, logo esta pergunta reflete mais o procedimento do profissional no desenvolvimento.

16) Em se tratando de testes, você costuma efetuar:

a) verificação da especificação em relação aos requisitos
b) validação com sistemas de certificação de que o software implementa efetivamente o que foi especificado

c) testes com um conjunto de dados pré-determinados
d) testes com o ambiente e condições de operação do cliente.

Testes e verificação de requisitos são fundamentais para assegurar a qualidade de software. O custo e o tempo normalmente são limitantes para poder efetuar todos os testes programados. O uso dos sistemas de certificação obriga a realização dos testes especificados. A resposta a esta pergunta reflete o nível da qualidade assegurada de software produzida pela empresa.

17) Em se tratando de produtividade, é possível afirmar que:
    a) você consegue cumprir o prazo designado
    b) você conhece o número de linhas de código que consegue produzir por dia/semana/mês
    c) você conhece o tempo necessário para produzir uma determinada rotina ou programa
    d) você conhece a produtividade da sua equipe

O conhecimento da produtividade e a correta estimativa de prazo e custo é um dos problemas básicos que a Engenharia de Software quer enfocar. O não cumprimento do prazo mostra que a empresa falha no planejamento, na estimativa dos custos e esforços de desenvolvimento. Provavelmente também apresenta falha no gerenciamento do projeto de desenvolvimento.
Por outro lado o desconhecimento do profissional da sua própria produtividade mostra a sua inexperiência.

18) As aplicações que você desenvolve tem um ante projeto e um projeto antes da prototipação?

( ) Sim ( ) Não ( ) Não Sei

Tradicionalmente o desenvolvedor inicia o processo de programação antes de ter o projeto bem definido. Esta pergunta verifica se ante-projeto ou prototipagem

tem sido ou não norma nas empresas, assim como se o hábito de não fazer prototipagem ainda persiste entre os desenvolvedores de software. A resposta "não sei" mostra o desconhecimento do profissional sobre como deve ser feito um projeto.

19) Ao tratar grandes volumes de informação, você utiliza gerenciadores de bancos de dados relacionais?

( ) Sim ( ) Não ( ) Não sei

Esta pergunta verifica o uso de banco de dados entre os desenvolvedores.

20) Você usa alguma ferramenta para a análise e modelagem de dados?

( ) Sim ( ) Não ( ) Não Sei

Da mesma forma, esta pergunta visa saber se a modelagem de dados é feita de forma eficiente.

21) Na programação você adota:

a) ferramentas que facilitam a programação: Genexus, GAS
b) linguagens de alto nível: Visual Basic, Delphi, SQL
c) linguagens de baixo nível: C, Assembly
d) ferramentas visuais que facilitam a geração de código: C++, C#, Java
e) Outra(s)

Esta pergunta visa obter a estatística dos desenvolvedores no uso da linguagem de desenvolvimento de programas.

Em caso de outra(s), indique qual(is)

---

22) As aplicações que você desenvolve mantém uma implementação física de banco de dados espelhada em modelagem de dados?

( ) Sim ( ) Não ( ) Não sei

Esta pergunta procura saber se mesmo na implementação é mantida a modelagem dos dados e cruza as informações entre si, no caso, com a pergunta de número 20.

23) Assinale uma ou mais técnicas de orientação a objeto que você utiliza

a) reusabilidade (classes e objetos)
b) encapsulamento
c) herança
d) polimorfismo
e) nenhuma
f) outra(s)

Em caso de outra(s), indique qual(is)

Nesta pergunta é verificado o nível de uso de novas técnicas no trabalho de programação.

24) Nas aplicações em que você desenvolve são representadas as mensagens síncronas ou assíncronas existentes entre os objetos?

( ) Sim ( ) Não ( ) Não sei

Com a Internet e os sistemas operacionais modernos esta questão é muito importante e a pergunta verifica qual a realidade existente.

25) Qual é a linguagem de programação que você mais adota?

a) Java
b) VB
c) Delphi
d) AdvPl
e) Clipper
f) Cobol
g) C++
h) SQL
i) outra(s)

Em caso de outra(s), indique qual(is)

_____

Esta pergunta cruza com a pergunta 21 sobre o uso de linguagens no processo de desenvolvimento de softwares aplicativos no Brasil.

5.7.2 Pesquisa de análise do Perfil do DESENVOLVEDOR DE SOFTWARE APLICATIVO NO BRASIL

O segundo questionário aplicado ao primeiro grupo de 400 desenvolvedores e a um novo grupo de empresas e profissionais, aos quais foi também entregue o primeiro, teve como objetivo traçar de forma mais específica o seu perfil pessoal e mais dados sobre sua formação. Alguns dos profissionais que não responderam à primeira pesquisa estiveram presentes nesta segunda etapa.

Nome (opcional): _____

Idade: _____

Há quantos anos trabalha em Desenvolvimento de Software: ____

Esta pergunta visa conhecer o tempo médio de atuação dos desenvolvedores.

Estado Civil

Solteiro Casado Separado Viuvo

O trabalho do desenvolvedor de software é bastante estressante e descontrolado e chega a provocar consequencias negativas em sua vida familiar. Esta pergunta visa confirmar ou desmentir esta suposição.

Já trabalhou em quantas empresas de TI: ____

A fidelidade pela empresa não é questão primordial nos dias de hoje. A pergunta tem o objetivo de verificar até onde isto é verdade.

Há quantos anos trabalha na empresa atual? _____

Esta pergunta e a seguinte complementam a anterior.

Qual o tempo médio de permanencia em cada empresa: _____

Seu primeiro emprego foi com que idade : ____

Esta pergunta objetiva verificar a qual classe social pertencem os desenvolvedores de software.

Qual foi este emprego:
Office boy
Estagiário na área de TI
Estagiário em outra área (escritório, fábrica, etc)
Ajudante geral
Vendas
Programador

Esta pergunta não só complementa a anterior, como também verifica se desde o ínicio da carreira o desenvolvedor já aderiu à TI.

Profissão do pai:
Operário Escritório Vendas Empresário
Funcionário Público Analista de Sistemas/Programador
Professor Arquiteto/Engenheiro Artista
Advogado/Médico Outros

Profissão da mãe:
Operária Escritório Vendas Empresária
Funcionária Pública Analista de Sistemas/Programadora
Professora Arquiteta/Engenheira Artista
Advogada/Médico Outros Do Lar

Alem de verificar a classe social, procura-se saber se já começa a existir uma tendência de influência paterna ou materna na escolha das profissões na área de TI.

Formação Fundamental e Médio
Tipo de escola
Pública Particular

Completa a busca da classe social do desenvolvedor, pois a escola privada ainda é no Brasil privilégio das classes A e B.

Disciplinas que mais gostava
Portugues Matemática Física/Ciências Química
Biologia História Geografia Inglês
Filosofia Educação Física

Disciplinas que menos gostava
Portugues Matemática Física/Ciências Química
Biologia História Geografia Inglês
Filosofia Educação Física

Com estas perguntas procura-se saber quais disciplinas são preferidas pelos desenvolvedores e verificar se há uma sintonia com aquelas requeridas nos exames vestibulares.

Formação Superior
Tipo de escola
Pública Particular

Mais uma pergunta para definição da classe social e da própria capacidade dos desenvolvedores, visto saber-se que as vagas nas escolas públicas são bem mais restritas.

Tipo de graduação
Nenhuma
Só superior
Superior + MBA ou Pós
Superior + Mestrado
Superior + Mestrado + Doutorado

A formação educacional de nível mais elevado ainda é bastante pequena no Brasil e mais usual entre os profissionais da área acadêmica. Esta pergunta já havia na primeira pesquisa, mas de forma incompleta.

Cursos de
Analise de Sistemas, Processamento de Dados ou Sistemas de Informação, Administração de Empresas
Ciencias da Computação ou Engenharia de Software

Engenharia
Outros

Aquí se procurou confirmar a questão da preferência pelas várias opções oferecidas pelos cursos de nível superior.

Outros Cursos de Formação Superior (MBA, Pós, Mestrado, Doutorado)
Análise de Sistemas, Processamento de Dados ou Sistemas de Informação
Ciencias da Computação ou Engenharia de Software
Administração de Empresas
Engenharia (Elétrica, Metalurgia, Química, etc)
Outros

Esta pergunta completa a anterior.

Passatempos e Hobbies preferidos
Games na Internet ou computador
Leitura de livros técnicos
Leitura de outros livros
Cinema/teatro/shows
Navegar na Internet
Esportes em geral
Atividades com a familia

Há uma controvérsia quanto aos hábitos fora do ambiente de trabalho do desenvolvedor. A análise desta característica foi o foco desta pergunta.

Você tem o hábito de trabalhar em casa, à noite ou nos fins de semana?
Sim Não

Verificar se é verdadeira a mística do profissional de TI ter uma dedicação acima do normal em seu trabalho é o objetivo desta pergunta.

Leitura de livros

Quantos livros você lê por ano: ____

Assuntos:
Política e Economia
Técnicos
Ficção (romances, policiais)
Auto-ajuda

Quantas revistas você lê por semana?: _____

Assuntos:
Política e Economia
Ficção (romances, policiais entre outras)

Técnicas

Você costuma ler jornais com que freqüência:
Diariamente
Semanalmente
Não lê

Estas perguntas visam conhecer os hábitos de leitura e como é a busca de informações dos desenvolvedores de software e a relação com sua profissão.

Você acessa a Internet para quais finalidades:
Notícias
Compras
Assuntos técnicos
Faz downloads
Somente e_mails e conta bancária

Considerando a facilidade de acesso à Internet que dispõe o desenvolvedor de software, cabe verificar se ele de fato faz uso dela e de que maneira.

Como aprendeu a programar:
Na faculdade
Auto-didata (livros, testando)
Cursos extra-curriculares
No trabalho

Esta pergunta também já constava do primeiro questionário, mas aqui ela foi feita de forma mais fechada.

Média de horas de treinamento em TI por ano (cursos, palestras, sem considerar a faculdade)
Até 40 horas
De 40 a 160 horas
Acima de 160 (1 mês)

Quantas vezes foi ao exterior para cursos/eventos de TI: ____

Detalhar mais o treinamento extra-curricular do desenvolvedor de software foi o objetivo desta pergunta.

Conhecimento de Idiomas:

Inglês
Baixo Médio Fluente

Espanhol
Baixo Médio Fluente

Outros
Baixo Médio Fluente

Considerando que os idiomas mencionados são importantes no desempenho profissional do desenvolvedor, torna-se importante saber qual é o seu nível.

Conhece ou usa as seguintes técnicas de Engenharia de Software

	Conhece	**Usa**
MER		
DFD (Diagrama de Blocos, Fluxograma,Programação Estruturada)		
UML		
Analise de Ponto de Função		
Erwin		
IDE (Delphi, WebSphere, .Net)		
OOP Programação Orientada a Objetos		
Fusion		

E finalizando este segundo questionário, mais uma pergunta, sobre o uso da Engenharia de Software no dia a dia do desenvolvedor de software.

6 CLASSIFICAÇÃO E TABULAÇÃO DOS DADOS ENCONTRADOS NA PESQUISA

Após a aplicação da pesquisa, junto aos gerentes de programação e programadores, os dados classificados e tabulados forneceram um amplo material que possibilitou definir qual é o perfil dos desenvolvedores de software aplicativo do Brasil, quais são as formações complementares necessárias para seu aprimoramento se quisermos ter uma mão de obra de qualidade para competir no mercado mundial de software e quais as prováveis deficiências nesta aprendizagem. As tabelas 2, 3 e 4 consolidam as respostas obtidas na aplicação da pesquisa para cada uma das questões.

A Tabela 2 retrata as respostas do primeiro questionário:

	Gerente	Programador	Total
3) O seu primeiro contato com programação foi			
a) Na escola	26	39	65
b) Através de cursos extra curriculares	6	55	61
c) No trabalho	8	26	34
d) Através de literatura procurando aprender sozinho	8	7	15
4) A sua formação é:			
a) ensino médio (2º grau)	4	3	7
b) superior incompleto	20	20	40
c) superior completo	22	101	123
d) mestrado	2	3	5
e) doutorado	0	0	0
5) Um bom programador é um programador que:			
a) Consegue desenvolver um aplicativo	41	114	155
b) Trabalha com linguagens de baixo nível	0	11	11
c) Desenvolve rotinas para Softwares como SO e Compiladores	7	13	20
d) Trabalha com programação orientada a objeto	5	18	23
6) De que forma você acredita que melhor aprendeu sobre programação?			
a) Por meio de exposição teórica do professor	14	27	41
b) Com exercícios práticos em laboratório	14	45	59
c) Analisando códigos de programas já elaborados	10	40	50
d) Na prática do dia-a-dia de trabalho	39	71	110
7) As aplicações que você desenvolve tem:			
a) Laços de repetição	21	34	55
b) Desvios incondicionais	2	6	8
c) Desvios condicionais	14	21	35
d) Funções reutilizáveis	20	38	58
e) Todas as anteriores	24	78	102
f) Nenhuma das anteriores	1	8	9
8) As aplicações que você desenvolve incorporam rotinas de manutenção de integridade para detecção de erros de invariantes ?			
( ) Sim	27	69	96
( ) Não	8	13	21
( ) Não Sei	13	41	54
9) Você programa aplicações concorrentes com o uso de estruturas do tipo semáforos, monitores ou similares para controle do acesso de regiões críticas?			
( ) Sim	25	100	125
( ) Não	15	20	35
( ) Não Sei	8	4	12
10) As suas aplicações utilizam ferramentas automáticas de geração de código, como construtores automáticos de analisadores léxicos, ou semânticos, estilo lex e yacc?			
( ) Sim	16	39	55
( ) Não	23	78	101
( ) Não Sei	9	7	16

11)Você usa alguma metodologia ou técnica (Métricas) para avaliação do desenvolvimento de sistemas ou softwares?			
a) Análise de Pontos por Função	27	46	73
b) COCOMO	0	0	0
c) PMBOK (Gerencia de Projetos)	9	22	31
d) ISO9126	1	1	2
e) Nenhuma	14	68	82
f) Outra(s)	4	10	14
12) Em seu processo de desenvolvimento você faz uso de:			
a) UML	24	44	68
b) Fusion	2	1	3
c) Análise Estruturada	29	93	122
d) MER (Modelo de Entidade Relacional)	34	56	90
13) Em sua opinião, o entendimento dos requisitos do software a ser desenvolvidos tem como origem:			
a) O detalhamento da descrição dos próprios requisitos	46	100	146
b) As conclusões obtidas a partir das reuniões com os clientes	31	74	105
c) As conclusões obtidas a partir das reuniões com a equipe de desenvolvimento	24	37	61
d) A orientação da chefia	0	30	30
14) Em se tratando do processo de desenvolvimento, você representa os algoritmos:			
a) Através do uso da UML	28	46	74
b) Utilizando as técnicas de DFD	21	43	64
c) Utilizando uma Meta Linguagem	20	16	36
d) Através de um fluxograma	9	68	77
15) Em relação à documentação do software aplicativo, você costuma efetuar:			
a) comentários por linha, depois de cada comando inserido em um método	14	42	56
b) comentários relativos à uma classe de objetos	16	41	57
c) comentários para cada método	35	77	112
d) documentação somente depois de ter o programa funcionando	1	16	17
16) Em se tratando de testes, você costuma efetuar:			
a) verificação (verificação da especificação em relação aos requisitos)	14	61	75
b) validação (certificação de que o software implementa efetivamente o que foi especificado)	39	103	142
c) testes com um conjunto de dados pré-determinados	24	37	61
d) testes com com o ambiente e condições de operação do cliente	27	32	59
17) Em se tratando de produtividade, é possível afirmar que:			
a) Você consegue cumprir o prazo designado	10	70	80
b) Você conhece o número de linhas (códigos) que consegue produzir por dia/semana/mês	0	3	3
c) Você conhece o tempo necessário para produzir uma determinada rotina ou programa	26	61	87
d) Você conhece a produtividade da sua equipe	25	50	75

18) As aplicações que você desenvolve tem um ante projeto e um projeto antes da prototipação?			
( ) Sim	32	78	110
( ) Não	16	44	60
( ) Não Sei	0	2	2
19) Ao tratar grandes volumes de informação, você utiliza gerenciadores de bancos de dados relacionais?			
( ) Sim	48	88	136
( ) Não	0	28	28
( ) Não Sei	0	8	8
20) Você usa alguma ferramenta para a análise e modelagem de dados?			
( ) Sim	28	43	71
( ) Não	18	66	84
( ) Não Sei	2	15	17
21) Na programação que você desenvolve, você adota:			
a) Ferramentas que facilitam a programação: Genexus, GAS	1	14	15
b) Linguagens de alto nível: Visual Basic, Delphi, SQL	36	81	117
c) Linguagens de baixo nível: C, Assembly	7	17	24
d) Ferramentas visuais que facilitam a geração de código: C++, C#, Java	25	58	83
e) Outra(s)	3	35	38
22)As aplicações que você desenvolve matem uma implementação física de banco de dados espelhada em modelagem de dados?			
( ) Sim	45	84	129
( ) Não	2	36	38
( ) Não Sei	1	4	5
23)Assinale uma ou mais técnicas de orientação a objeto que você utiliza			
a) Reusabilidade (Classes e Objetos)	44	97	141
b) Encapsulamento	45	65	110
c) Herança	39	78	117
d) Polimorfismo	27	52	79
e) Nenhuma	0	24	24
f) Outra(s)	0	0	0
24) Nas aplicações em que você desenvolve são representadas as mensagens síncronas ou assíncronas existentes entre os objetos?			
( ) Sim	30	21	51
( ) Não	6	55	61
( ) Não Sei	12	46	58
25) Qual é a linguagem de programação que você mais adota?			
a) Java	2	19	21
b) VB	27	40	67
c) Delphi	2	7	9
d) AdvPl	4	44	48
e) Clipper	0	2	2
f) Cobol	4	3	7
g) C++	9	1	10
h) SQL	22	22	44
i) Outra(s)	5	27	32

Tabela 2 – Consolidação das Respostas Obtidas na Pesquisa.

A tabela 3 demonstra a média das respostas obtidas para cada uma das questões na segunda pesquisa, lembrando que a maior parte das questões são de múltipla escolha.

Questão	**Média**
Idade	31 anos
Número de anos de trabalho	9 anos
Desenvolvimento de software	
Estado Civil	
Solteiro	47,00%
Casado	50,00%
Separado	0,00%
Viúvo	0,00%
Número de empresas que já trabalhou	3 anos
Número de anos na atual empresa	5 anos
Idade do primeiro emprego	16 anos
Especificação do primeiro emprego	
Office boy	17,92%
Estagiário em TI	31,13%
Estagiário em outra área	13,21%
Ajudante geral	15,09%
Vendas	3,77%
Programador em TI	13,21%
Profissão do pai	
Operário	23,58%
Funcionário público	8,49%
Advogado/ médico	2,83%
Vendas	8,49%
Artista	0,94%
Professor	0,94%
Analista de sistemas/ programador	0,94%
Arquiteto , engenheiro	0,94%
Empresário	4,72%
Outros	25,47%
Profissão da mãe	
Operária	3,77%
Funcionaria pública	5,66%
Artista	0,94%
Advogada/médica	3,77%
Vendas	0,94%
Professora	14,15%
Outros	10,38%
Analista de sistemas/ programadora	0,00%
Empresária	0,94%
Arquiteta/ engenharia	2,83%
Do lar	50,94%

Formação fundamental e ensino médio	
Escola pública	66,98%
Escola particular	23,58%
Disciplinas de maior interesse	
Português	26,42%
Matemática	76,42%
Física/ ciências	60,38%
Química	27,36%
Biologia	18,87%
História	36,79%
Geografia	31,13%
Inglês	39,62%
Filosofia	8,49%
Educação física	33,02%
Disciplina de menor interesse	
Português	38,68%
Matemática	6,60%
Física/ ciências	9,43%
Química	33,02%
Biologia	29,25%
História	26,42%
Geografia	24,53%
Inglês	11,32%
Filosofia	17,92%
Educação física	11,32%
Formação Superior	
Escola Pública	31,13%
Escola Particular	64,15%
Tipo de graduação	
Nenhuma	3,77%
Superior	53,77%
Superior + MBI ou pós	34,91%
Superior + Mestrado	2,83%
Superior + Doutorado	0,00%
Cursos concluídos	
Analise de sistemas, processamento de dados ou sistemas de informação	47,17%
Ciências da computação ou engenharia de software	27,36%
Admistração de empresas	10,38%
Engenharia elétrica, metarlúgica, química etc	5,66%
Outros	9,43%
Tipo de passatempos ou hobby	
Games na Internet ou computador	25,47%
Leitura de livros técnicos	29,25%
Leitura de outros livros	49,06%
Cinema/teatro/shows	67,92%
Navegar na internet	24,53%
Esportes em geral	52,83%
Atividade com a família	54,72%
Hábito de trabalhar em casa, à noite ou nos finais de semana	
Sim	60,38%
Não	37,74%

Quantidade de livros lidos ao ano	6
Assunto de maior interesse na literatura	
Política e economia	24,53%
Assuntos técnicos de computação e informática	71,70%
Ficção, romances e aventuras policias	55,66%
Auto ajuda	18,87%
Quantidade de revistas lidas por semana	0,015
Assuntos de maior interesse nos peródicos	
Política e economia	62,26%
Assuntos técnicos de computação e informática	56,60%
Ficção, romances e aventuras policias	6,60%
Auto ajuda	2,83%
Periodicidade de leitura de jornais	
Diariamente	33,96%
Semanalmente	42,45%
Nenhuma	17,92%
Finalidade dos acessos na Internet	
Noticias	88,68%
Compras	54,72%
Assuntos técnicos	90,57%
Dowloads	69,81%
Somente e-mail e transações bancárias	28,30%
Pricipal fonte de aprendizagem de programação	
Na faculdade	38,68%
Auto didata (livros , testando , etc)	52,83%
Cursos extra curriculares	29,25%
No trabalho	36,79%
Média de horas de treinamento em TI por ano (cursos , palestras, sem considerar a faculde)	
Até 40 horas	33,96%
De 40 a 160 horas	46,23%
Acima de 160	8,49%
Qtas vezes foi ao exteriors para cursos/ eventos TI	0
Conhecimento de idiomas	
Inglês	7,55%
Baixo	61,32%
Médio	27,36%
Fluente	
Espanhol	
Baixo	54,72%
Médio	30,19%
Fluente	4,72%
Outros	
Baixo	52,83%
Médio	13,21%
Fluente	

Tabela 3 – Consolidação das respostas obtidas na pesquisa das características básicas dos programadores de software de aplicação e a fonte de seu aprendizado na área de programação.

A tabela 4 consolida a pesquisa relativa às questões quanto ao conhecimento e o uso das técnicas de engenharia de software.

Conhecimento e uso de técnicas de engenharia de software	Conhece	Usa
MER	66,98%	55,66%
DFD	70,75%	55,66%
UML	50,00%	21,70%
Analise por fonto de Função	42,45%	7,55%
Erwin	37,74%	21,70%
IDE(delphi, Websphere, .Net)	48,11%	37,74%
OOP Programação Orientada a Objeto	59,43%	41,51%
Fusion	2,83%	0,00%

Tabela 4 – Conhecimento e uso de técnicas de engenharia de software.

7 ANÁLISE DOS DADOS OBTIDOS NA PESQUISA

A escolha das empresas envolvidas na pesquisa baseou-se em critérios qualitativos que representem um escopo com amplitude nacional. Dentro destes critérios, as empresas a serem escolhidas, em uma primeira instância, devem ser empresas que desenvolvem, implementam e comercializam sistemas de tecnologia da informação, software e aplicações voltadas para a Internet. Isto gerou a necessidade, para a pesquisa, da busca de profissionais que atuam em empresas que representam diferentes classes de tecnologia na área de computação e informática. A abrangência em todo o território nacional teve por objetivo garantir que a pesquisa realizada refletisse efetivamente o perfil do desenvolvedor no Brasil. Por fim, para consolidar a escolha das empresas que atendessem estes critérios, a ASSESPRO (Associação das Empresas Brasileiras de Tecnologia da Informação, Software e Internet) possibilitou, por meio de seus cadastros o acesso aos profissionais pretendidos para a pesquisa.

Por se tratar do perfil e da formação do desenvolvedor de software no Brasil e considerando que a maioria das empresas atua em todo o Brasil e ainda que os cursos na área de computação e informática estão orientados pelas diretrizes curriculares do MEC, não foram evidenciadas e nem requeridas diferenças regionais.

A pesquisa, distribuída a mais de 400 desenvolvedores de software aplicativo e respondida por 162 profissionais, proveniente de mais de 60 empresas diferentes de pequeno e médio porte incluindo três multinacionais de desenvolvimento de softwares na área comercial, representa o perfil típico do desenvolvedor de software aplicativo no Brasil. As respostas obtidas correspondem a 40% dos questionários enviados e representam 1,12% do total estimado de desenvolvedores no Brasil, que é de 14.400 pessoas, segundo os dados

do MCT. Há de se considerar ainda que a pesquisa foi feita entre os profissionais de incontestável êxito profissional e muitos deles candidatos aprovados na primeira fase do exame de seleção para ingressar no Mestrado Profissional de Engenharia de Computação do IPT.

Mas será que essa amostragem é suficiente para determinar o perfil dos desenvolvedores de software aplicativo no Brasil? Qual é nível de confiança e o erro máximo cometido?

Segundo o livro Estatística Aplicada de Larson e Faber [LAR2004], a fórmula para determinar-se o tamanho mínimo da amostra com um determinado nível de confiança e um erro máximo de estimativa, independentemente do tamanho do universo, é:

$$N = pq \left(z/E \right)^2$$

Onde, N = tamanho da amostra

p e q : estimativas preliminares

z: nível de confiança, significando que o universo da nossa amostra cai com grau de confiabilidade z dentro da curva normal de distribuição Gaussina. Neste caso,

90% z = 1,645

95% z = 1,96

99% z = 2,575

E: corresponde ao erro máximo

Considerando a pior estimativa preliminar (nenhuma resposta/opção é considerada preliminarmente como a mais provável), ou seja, 0,5 para p e q, temos:

Para N = 162

$0,5 * 0,5 (1,645/0,065)^2$

ou seja, para um nível de confiança de 90% e erro máximo de 6,5%, a amostra mínima é de 162.

Para esta dissertação cremos que um nível de confiança de 90% e um erro máximo de 6,5% pode ser considerado aceitável. Portanto os 162 questionários respondidos são considerados suficientes para obtenção de um resultado preliminar.

Para obter resultados mais acurados no futuro, pode-se considerar os resultados dessa pesquisa como “piloto”, para cálculo do tamanho de amostragem para outros níveis de confiança e outros valores de erro máximo.

O resultado da análise da pesquisa pode ser dividido em quatro grupos: os fatores pessoais que determinaram a formação do desenvolvedor; o seu perfil profissional, ou seja, o seu conhecimento técnico; a sua assimilação dos conceitos de Engenharia de Software, fator indispensável para um desenvolvimento com qualidade e, finalmente, uma análise de seu ambiente de trabalho, em especial no que diz respeito à existencia de processos de avaliações sobre o seu trabalho.

7.1 Grupo: Fatores pessoais

Os resultados das perguntas desse grupo são apresentados abaixo

Profissão do pai	
Operário	23,58%
Funcionário público	8,49%
Advogado/ médico	2,83%
Vendas	8,49%
Artista	0,94%
Professor	0,94%
Analista de sistemas/ programador	0,94%
Arquiteto , engenheiro	0,94%
Empresário	4,72%
Outros	25,47%

Tabela 5 – Profissão exercida pelo pai do desenvolvedor de software.

Formação fundamental e ensino médio	
Escola pública	66,98%
Escola particular	23,58%

Tabela 6 – Características das escolas de ensino fundamental e ensino médio.

O desenvolvedor de software tem sua origem nas classes C e D. Isto se comprova não só pela profissão dos pais, como tambem pelo fato de que a maioria fez o curso fundamental e médio em escola pública, conforme a tabela 5 e a tabela 6. E decorrência deste fato não conseguem realizar o curso superior nessas escolas, optando pela faculdade particular, conforme mostra a tabela 7.

Formação Superior	
Escola pública	31,13%
Escola particular	64,15%

Tabela 7 – Características das escolas de ensino superior.

A causa dessa carreira não sensibilizar jovens de classes mais altas pode estar no fato de que no Brasil técnicos não são considerados pessoas de destaque, tanto socialmente como financeiramente. Aliás são poucas as empresas que oferecem uma carreira em Y, ou seja, remunerar adequadamente o especialista. Normalmente, para auferir

rendimentos mais elevados o técnico sai dessa carreira para tornar-se coordenador, gerente, supervisor ou outro cargo similar atingindo assim a sonhada posição de conforto financeiro e reconhecimento social. Foi constatado também que 84% dos desenvolvedores de software são do sexo masculino, fato que comprova essa tese se compararmos com outros setores de atividade (suporte e vendas, por exemplo), em que a presença feminina é bem maior.

Conforme os dados da pesquisa consolidados na tabela 8, a idade média de 31 anos mostra que a transformação dos conceitos técnicos de desenvolvimento ocorrida nos últimos anos fez com que os programadores mais antigos não conseguissem acompanhar esta evolução, restando a eles concentrarem-se nas empresas que ainda trabalham naquelas plataformas, desatualizadas mas, nem por isso, ineficientes.

Questão	**Média**
Idade	31 anos

Tabela 8 – Idade média dos desenvolvedores de software.

A tabela 9, que demonstra um baixo número de profissionais com títulos de mestrado e doutorado, nos leva a supor que estas graduações não tem atraido os técnicos da maneira como deveria e que alguma atitude deve ser tomada nesse sentido.

Tipo de graduação	
Nenhuma	3,77%
Superior	53,77%
Superior + MBI ou pós	34,91%
Superior + Mestrado	2,83%
Superior + Doutorado	0,00%

Tabela 9 – Titulação dos desenvolvedores de software.

Também percentagem maior (47% contra 27% consolidados na tabela 10) de desenvolvedores oriundos das faculdades de Sistemas de Informação, Análise de Sistemas e Processamento de Dados em confronto com a de Ciências da Computação é um fato que preocupa se considerarmos ser ela a mais adequada para o cumprimento dos desafios que a carreira apresenta.

Cursos concluídos	
Analise de sistemas, processamento de dados ou sistemas de informação	47,17%
Ciências da computação ou engenharia de software	27,36%
Admistração de empresas	10,38%
Engenharia elétrica, metarlúgica, química etc	5,66%
Outros	9,43%

Tabela 10 – Área de formação dos desenvolvedores de software.

Na análise da questão sobre passatempos e hobby dos desenvolvedores de software conclui-se que estes indivíduos não justificam mais a imagem de serem profissionais isolados socialmente por apresentarem apenas conhecimentos técnicos que durante tanto tempo os perseguiram pois, além de priorizarem atividades comuns, como cinema, teatro e shows (67%), preferem atividades em família (55%), conforme tabela 11.

A maioria é casado e não se constatou nenhum caso de estado civil separado.

Outra constatação interessante é que poucos aderem a atividade de games, seja na internet ou no computador.

Tipo de passatempos ou hobby	
Games na Internet ou computador	25,47%
Leitura de livros técnicos	29,25%
Leitura de outros livros	49,06%
Cinema/teatro/shows	67,92%
Navegar na internet	24,53%
Esportes em geral	52,83%
Atividade com a família	54,72%

Tabela 11 – Preferências de lazer.

Por outro lado, analisando-se as questões sobre hábitos de leitura, constata-se que os desenvolvedores não são pessoas altamente politizadas e tem baixa aderência ao conhecimento cultural. Preferem assuntos técnicos e de ficção. Isto é confirmado pela preferência de uma leitura semanal dos jornais, ao invés da diária e pela forma como acessam a Internet, conforme dados da tabela 12, tabela 13, tabela 14 e tabela 15.

Quantidade de livros lidos ao ano	
Assunto de maior interesse na literatura	
Política e economia	24,53%
Assuntos técnicos de computação e informática	71,70%
Ficção, romances e aventuras policiais	55,66%
Auto ajuda	18,87%

Tabela 12 – Aspecto quantitativo do interesse por literatura.

Assuntos de maior interesse nos periódicos	
Política e economia	62,26%
Assuntos técnicos de computação e informática	56,60%
Ficção, romances e aventuras policiais	6,60%
Auto ajuda	2,83%

Tabela 13 – Assuntos de interesse.

Periodicidade de leitura de jornais	
Diariamente	33,96%
Semanalmente	42,45%
Nenhuma	17,92%

Tabela 14 – Aspectos quantitativos do interesse por periódicos.

Finalidade dos acessos na Internet	
Notícias	88,68%
Compras	54,72%
Assuntos técnicos	90,57%
Downloads	69,81%
Somente e-mail e transações bancárias	28,30%

Tabela 15 – Finalidade dos acessos na Internet.

7.2 Grupo: Conhecimento técnico

Um conjunto de perguntas das pesquisas visou averiguar o nível do conhecimento técnico do desenvolvedor de software no Brasil

Uma das questões dirigidas (pergunta 9) solicitava a caracterização dos programas desenvolvidos buscando a identificação do tipo de rotinas utilizadas nos processos e atualizações de dados, conforme a consolidação em percentuais.

Programação de aplicações concorrentes com o uso de estruturas do tipo semáforos, monitores ou similares para controle do acesso de regiões críticas	Gerente	Programador	Total
Sim	52%	81%	73%
Não	31%	16%	20%
Não sei	17%	3%	7%
Total	100%	100%	100%

Tabela 16 – Uso de controles em situações críticas.

A tabela 16 sugere que tanto gerentes como programadores pesquisados incorporam sofisticadas rotinas nas aplicações desenvolvidas. Associada aos resultados obtidos na tabela 17 que aponta para o fato de não ser prática comum o uso de ferramentas que auxiliam o desenvolvimento e tampouco o uso de linguagens de baixo nível, nos leva a conclusão que estes processos são feitos apenas no nível superficial das aplicações. Talvez com a possibilidade que o

Software Livre ofereçe, ao permitir um acesso ao código fonte do sistema operacional, das ferramentas e de aplicativos mais complexos, como editores de texto e de apresentações, planilhas eletrônicas e banco de dados haja uma mudança neste cenário, surgindo no Brasil empresas que desenvolvam software básico. O Fredows, desenvolvido pela Cobra, é um exemplo desta tendência

Ferramentas adotadas na programação	Gerente	Programador	Total
Ferramentas que facilitam a programação: Genexus, GAS	1%	7%	5%
Linguagens de alto nível: Visual Basic, Delphi, SQL	50%	40%	42%
Linguagens de baixo nível: C, Assembly	10%	8%	9%
Ferramentas que facilitam a geração de código: C++, C#, Java	35%	28%	30%
Outra(s)	4%	17%	14%
Total	100%	100%	100%

Tabela 17 – Ferramentas utilizadas.

Uma das questões relevantes na pesquisa é a origem do conhecimento acerca da programação. Nesse sentido é importante identificar qual foi o primeiro contato com a programação e qual foi o fator preponderante para que houvesse efetivamente a compreensão, assimilação e aplicação dos conceitos de programação em aplicações reais. Considerando as respostas obtidas na pergunta 3, muito embora o primeiro contato com a programação para a grande maioria, conforme a tabela 18, tenha sido através de cursos extra curriculares, a tabela 19 (pergunta 6) demonstra que a prática do dia-a-dia de trabalho consolida de maneira mais adequada o aprendizado nessa área. Este fato pode ser considerado um indicativo de que os profissionais buscam a adequação dos conhecimentos para os padrões e métodos utilizados nas empresas e que falta na escola um aprofundamento do ensino da programação propriamente dito.

Primeiro contato com programação	Gerente	Programador	Total
Na escola	54%	31%	37%
Através de cursos extra curriculares	13%	43%	35%
No trabalho	17%	20%	19%
Através de literatura procurando aprender sozinho	17%	6%	9%
Total	100%	100%	100%

Tabela 18 – Primeiro contato com programação.

Melhor forma de aprendizado de programação	Gerente	Programador	Total
Por meio de exposição teórica do professor	18%	15%	16%
Com exercícios práticos em laboratório	18%	25%	23%
Analisando códigos de programas já elaborados	13%	22%	19%
Na prática do dia-a-dia de trabalho	51%	39%	42%
Total	100%	100%	100%

Tabela 19 – Formas de aprendizados mais freqüentes na área de programação.

Considerando que a área de computação e informática é bastante ampla e que um programador pode atuar no desenvolvimento de diferentes maneiras e com utilização de variadas linguagens e ferramentas é importante saber, sob o ponto de vista dos profissionais entrevistados, qual é o conceito que se tem a respeito de um bom desenvolvedor. Quais os requisitos necessários, qual o perfil e quais os instrumentos mais importantes que devem estar na posse de um programador para que seja qualificado com excelentes níveis de satisfação profissional. As questões sobre rotinas de manutenção de integridade, o uso de estruturas do tipo semáforos, monitores ou similares para o controle do acesso de regiões críticas visa estabelecer o nível de complexidade dos programas desenvolvidos pelos profissionais pesquisados, conforme a tabela 20, pergunta 8.

Incorporação de rotinas de manutenção de integridade para detecção de erros de invariantes	Gerente	Programador	Total
Sim	56%	56%	56%
Não	17%	11%	12%
Não Sei	27%	33%	32%
Total	100%	100%	100%

Tabela 20– Identificação de uso de rotinas para a manutenção de integridade.

Associada à questão na qual a grande maioria não faz uso de ferramentas automáticas de geração de código, consolidada na tabela 17, sugere como um indicativo de que as aplicações têm tornado-se cada vez mais complexas e conseguir desenvolver um aplicativo é tarefa que, além de exigir um vasto conhecimento de técnicas, metodologias, linguagens e ferramentas, acima de tudo está a habilidade de aplicar-se todos os conceitos na prática para o desenvolvimento efetivo de um aplicativo. Neste ponto, de acordo com a consolidação dos dados da pesquisa constante na tabela 21, pergunta 5, grande parte dos gerentes e programadores que responderam à pesquisa acredita que um bom desenvolvedor pode ser definido como aquele que consegue desenvolver um aplicativo independentemente da técnica ou metodologia utilizada. Esta observação leva-nos a concluir que o desenvolvedor brasileiro não vê necessidade do uso de técnicas sofisticadas no seu trabalho, o que não deixa de ser um ponto questionável se considerarmos o atual estágio da tecnologia mundial.

Descrição de um bom programador	Gerente	Programador	Total
Consegue desenvolver um aplicativo	77%	73%	74%
Trabalha com linguagens de baixo nível	0%	7%	5%
Desenvolve rotinas para SO e Compiladores	13%	8%	10%
Trabalha com programação orientada a objeto	9%	12%	11%
Total	100%	100%	100%

Tabela 21 – Características de um bom programador sob o ponto de vista dos próprios programadores.

A questão da utilização de gerenciadores de bancos de dados relacionais tem por objetivo evidenciar e identificar as ferramentas, metodologias e procedimentos adotados no dia-a-dia dos profissionais da área de programação. Neste ponto, na análise dos dados da pesquisa, na tabela 22, verifica-se que no tratamento de grandes volumes de informações já consolidou-se o uso de bases adequadas ao momento tecnológico.

Utilização de gerenciadores de bancos de dados relacionais para o tratamento de grandes volumes de informação	Gerente	Programador	Total
Sim	100%	71%	79%
Não	0%	23%	16%
Não Sei	0%	6%	5%
Total	100%	100%	100%

Tabela 22 – Uso de gerenciadores de banco de dados relacional.

7.3 Grupo: uso de Engenharia de Software.

A tabela 25 mostra que o desenvolvedor de software no Brasil tem conhecimento das técnicas de Engenharia de Software, pois fazem parte hoje de qualquer curriculo escolar seja nos cursos de Ciências da Computação como naqueles que pela pesquisa predominam, pelo menos quantitativamente, as escolas de Sistemas de Informação e Análise de Sistemas.

Foi constatado que são utilizados gerenciadores de banco de dados e, no entanto, não são utilizadas ferramentas para análise e modelagem de dados na mesma intensidade, conforme demonstra a consolidação dos dados da pesquisa constante na tabela 23.

Utilização de ferramentas para a análise e modelagem de dados	Gerente	Programador	Total
Sim	58%	35%	41%
Não	38%	53%	49%
Não Sei	4%	12%	10%
Total	100%	100%	100%

Tabela 23 – Uso de ferramentas para modelagem de dados.

Neste sentido, a não utilização de ferramentas para análise e modelagem associada ao fato de que as aplicações desenvolvidas tem um ante projeto e um projeto antes da prototipação, conforme a tabela 24, pergunta 18, sugere que a maioria dos desenvolvedores cria técnicas específicas para essa tarefa deixando de utilizar algo que hipoteticamente é mais eficiente.

Projeto antes da prototipação das aplicações desenvolvidas	Gerente	Programador	Total
Sim	67%	63%	64%
Não	33%	35%	35%
Não Sei	0%	2%	1%
Total	100%	100%	100%

Tabela 24 – Elaboração de projetos antes da prototipação.

As questões que efetuam o levantamento das ferramentas, linguagens e metodologias utilizadas priorizam a identificação de quais sejam as mais utilizadas. As respostas destas questões, de acordo com a tabela 25, pergunta 12 e a tabela 26, pergunta 14, evidenciam a utilização, por parte dos profissionais, de metodologias tradicionais, embora já haja uma tendência forte para o uso de tecnologias mais avançadas, principalmente ligadas à Engenharia de Software.

Uso ferramentas, metodologias e linguagens no processo de desenvolvimento de aplicativos	Gerente	Programador	Total
UML	27%	23%	24%
Fusion	2%	1%	1%
Análise Estruturada	33%	48%	43%
MER (Modelo de Entidade Relacional)	38%	29%	32%
Total	100%	100%	100%

Tabela 25 – Uso de ferramentas e metodologias no desenvolvimento.

Representação dos algoritmos em se tratando do processo de desenvolvimento	Gerente	Programador	Total
Através do uso da UML	36%	27%	29%
Utilizando as técnicas de DFD	27%	25%	25%
Utilizando uma Meta Linguagem	26%	9%	14%
Através de um fluxograma	12%	39%	31%
Total	100%	100%	100%

Tabela 26 – Formas de representação de algoritmos.

Esse fato pode ser enfatizado pelas respostas obtidas na questão que aborda a representação das mensagens síncronas ou assíncronas entre os objetos, consolidada na tabela 27, pergunta 24.

Representação de mensagens síncronas ou assíncronas existentes entre os objetos nas aplicações desenvolvidas.	Gerente	Programador	Total
Sim	63%	17%	30%
Não	13%	45%	36%
Não Sei	25%	38%	34%
Total	100%	100%	100%

Tabela 27 – Representação da troca de mensagens.

Houve um grande número de respostas negativas ou de desconhecimento da questão, principalmente por parte dos programadores. Isto vem de encontro à questão que aponta a preferência da análise estruturada no processo de desenvolvimento, conforme já demonstrado na tabela 25.

Essas considerações sustentam que é ainda baixo o nível de compreensão das mudanças tecnológicas e conseqüentes modificações das aplicações tanto em sua concepção e desenvolvimento quanto em sua implementação em plataformas que inovam os conceitos tradicionais de processamento.

No entanto, essas considerações são mais evidentes nos programadores pesquisados. Em se tratando da consolidação dos dados obtidos a partir das pesquisas realizadas para os gerentes, observa-se que existe um percentual maior que afirma fazer uso da UML no processo de desenvolvimento e representação dos algoritmos, conforme a tabela 26. Outra evidência que aponta para uma investigação um pouco mais detalhada consiste no fato de que existe um percentual muito maior de gerentes que representam mensagens síncronas ou assíncronas existentes entre os objetos nas aplicações, como pode ser observada na tabela 27.

Uma interpretação para os resultados obtidos na tabela 25, 26 e 27 poderia sugerir que os programadores tem mais interesse em trabalhar com análise estruturada e utilização de fluxogramas do que o uso efetivo da linguagem de modelagem UML para o desenvolvimento de aplicações. Situação semelhante pode ser observada na tabela 23, na qual os gerentes, de forma contrária aos programadores, utilizam ferramentas para a análise e modelagem de dados. Ambos os casos podem sugerir que o nível de conhecimento maior por parte do gerente gera um interesse maior por representações através de metodologias e linguagens que possam retratar as novas tecnologias no processo de desenvolvimento de aplicativos.

Entretanto, a introdução de um fator teste pode confirmar ou não essa interpretação e provar de forma mais clara a relação existente entre as variáveis independentes (cargo) e as variáveis dependentes

(uso de ferramentas de modelagem de dados, metodologias no processo de desenvolvimento e a representação de mensagens síncronas e assíncronas) [YIN2001].

Elegendo o nível de escolaridade como uma variável de teste, é possível postular “se as pessoas que mantém um cargo gerencial fossem menos instruídas, elas mostrariam menos interesse por ferramentas, metodologias e linguagens que sejam capazes de auxiliar o processo de desenvolvimento e representação de aplicativos”. Para eliminar a influência que o grau de instrução possa vir a apresentar na amostra é necessário comparar o uso de ferramentas para análise e modelagem de dados por parte de gerentes e programadores de igual nível de instrução.

Utilização de ferramentas para a análise e modelagem de dados por grau de instrução	Até 2º Grau			Nível Superior		
	Gerente	Programador	Total	Gerente	Programador	Total
Sim	50%	70%	60%	65%	27%	34%
Não	50%	26%	38%	26%	59%	53%
Não Sei	0%	4%	2%	9%	14%	13%
Total	100%	100%	100%	100%	100%	100%

Tabela 28 – Utilização de ferramentas de análise por grau de instrução.

A tabela 28, pergunta 20, demonstra que classificando a amostra em grupos de igual nível de instrução, o interesse por ferramentas para análise e modelagem de dados aumenta para os profissionais com nível superior, especialmente entre os gerentes. Porém, a relação original observada na tabela 23 entre gerentes e programadores quanto ao interesse de ferramentas acentua-se sensivelmente. Isto demonstra que para os programadores, ao contrário dos gerentes, o grau de instrução

não é um fator preponderante para mudança de postura, comportamento ou atitude profissional.

Uma nova variável de teste pode ser introduzida na tabulação dos dados para a verificação desta interpretação. Elegendo programas de qualidade implementados nas empresas como uma variável de teste é possível postular "se as pessoas que tem um cargo gerencial trabalhassem em empresas que não mantém programas de qualidade, elas mostrariam menos interesse por ferramentas, metodologias e linguagens que sejam capazes de auxiliar o processo de desenvolvimento e representação de aplicativos".

A tabela 29, que foi baseada em constatação posterior em que verificou-se quais entrevistados trabalham em empresas com avaliação CMM nível 2 ou superior, demonstra que igualando os grupos no que se refere à implementação de programas de qualidade, a relação original é consideravelmente atenuada de tal forma que se não fosse pela implantação de programas de qualidade, não haveria relação entre o cargo e o uso de ferramentas. Além de constatar o fato de que tanto gerentes como programadores fazem efetivamente o uso de ferramentas de análise em empresas que implementam programas de qualidade, o resultado pode ser interpretado: gerentes têm preferência por uso de ferramentas para análise e modelagem porque têm uma maior preocupação com o nível de qualidade.

Cargo ⟶ Qualidade ⟶ Interesse por ferramentas

Quanto maior o cargo, maior a preocupação com a qualidade e, conseqüentemente, maior o interesse por ferramentas. Desta maneira, a interpretação inicial de que os programadores têm menos interesse por ferramentas revela-se inadequada, pois o interesse é o mesmo em

qualquer cargo se o nível de manutenção de programas de qualidade é o mesmo.

Utilização de ferramentas para a análise e modelagem de dados	Em empresas que não implementam CMM			Em empresas que implementam CMM		
	Gerente	Programador	Total	Gerente	Programador	Total
Sim	53%	48%	51%	85%	83%	84%
Não	45%	52%	48%	15%	17%	16%
Não Sei	2%	0%	1%	0%	0%	0%
Total	100%	100%	100%	100%	100%	100%

Tabela 29 – Utilização de ferramentas para análise e modelagem em empresas com CMM.

Situação semelhante observa-se na questão da pesquisa que levanta as ferramentas, metodologias e linguagens no processo de desenvolvimento e representação de aplicativos, conforme as tabelas a seguir.

Uso ferramentas, metodologias e linguagens no processo de desenvolvimento de aplicativos	Em empresas que não implementam CMM			Em empresas que implementam CMM		
	Gerente	Programador	Total	Gerente	Programador	Total
UML	26%	25%	26%	77%	83%	80%
Fusion	2%	2%	2%	0%	0%	0%
Análise Estruturada	33%	43%	37%	8%	8%	8%
MER	39%	30%	35%	15%	8%	12%
Total	100%	100%	100%	100%	100%	100%

Tabela 30 – Utilização de ferramentas no processo de desenvolvimento em empresas com CMM.

Representação dos algoritmos em se tratando do processo de desenvolvimento	Em empresas que não implementam CMM			Em empresas que implementam CMM		
	Gerente	Programador	Total	Gerente	Programador	Total
Através do uso da UML	35%	29%	33%	69%	63%	67%
Utilizando as técnicas de DFD	27%	24%	26%	15%	13%	14%
Utilizando uma Meta Linguagem	26%	9%	19%	8%	0%	5%
Através de um fluxograma	12%	38%	23%	8%	25%	14%
Total	100%	100%	100%	100%	100%	100%

Tabela 31 – Representação de algoritmos em empresas com CMM.

Os resultados obtidos nas tabelas 30 e 31 demonstram claramente que nas empresas que implantam programas de qualidade tanto gerentes como programadores tem maior preferência da UML não apenas na representação, mas fundamentalmente no processo de desenvolvimento de aplicativos.

Deve-se comentar que está em curso no Brasil um processo encabeçado pelo Softex e entidades de classe que visa oferecer uma alternativa às pequenas empresas na obtenção de avaliações na maturidade de modelos de capacitação. Essa alternativa, batizada de MPS – Maturação no Processo de Software - segue os princípios do CMM-I da SEI, mas não tem uma avaliação internacional e seu reconhecimento por ora visa atender à demanda de qualidade do governo e mercado brasileiros.

7.4 Grupo: ambiente profissional e avaliação do desenvolvedor de software.

Inicialmente cabe destacar que o ambiente profissional é aquele que ainda serve como maior indutor ao processo de aprendizado técnico do desenvolvedor de software aplicativo, conforme mostra a tabela 19.

Notou-se ainda que empresas localizadas em pólos de tecnologia ou em centros de incubadoras conseguem proporcionar aos seus técnicos um ambiente de sinergia favorável, com uma intensa troca de conhecimentos entre os desenvolvedores das inúmeras empresas ali presentes. Essas constatações foram verificadas nas visitas feitas ao Pólo Tecnológico TecnoPuc, de Porto Alegre e ao Pólo de São Leopoldo, onde foram entrevistados seis desenvolvedores.

As empresas de Software têm mostrado uma preocupação em dar um treinamento complementar aos seus colaboradores, conforme mostra a tabela 32 permitindo, assim, uma permanente atualização técnica de seus conhecimentos.

Média de horas de treinamento em TI por ano (cursos , palestras, sem considerar a faculdade)	
Até 40 horas	33,96%
De 40 a 160 horas	46,23%
Acima de 160 horas	8,49%

Tabela 32 – Aspectos quantitativos de treinamentos extra-curriculares.

Os processos de avaliação existentes nas empresas de desenvolvimento, nas quais o planejamento detalhado dos prazos de entrega, qualidade e custos são monitorados através de sistemas de Gestão de Projetos mundialmente reconhecidos, completam essa preocupação (PMI-PMBOK).

Outro resultado está no fato de que o desenvolvedor descarta padrões e componentes prontos e desenvolve tecnologia no próprio ambiente de trabalho, conforme mostra a tabela 33.

As suas aplicações utilizam ferramentas automáticas de geração de código, como construtores automáticos de analisadores léxicos, ou semânticos, estilo lex e yacc?			
( ) Sim	33%	31%	32%
( ) Não	48%	63%	59%
( ) Não sei	19%	6%	9%

Tabela 33 – Uso de ferramentas automáticas para geração de código.

Quanto à questão das metodologias de desenvolvimento de um novo projeto, aspectos estes que dependem muito mais da própria definição da empresa do que da simples vontade do desenvolvedor, constatou-se que a utilização de avançadas metodologias de desenvolvimento ainda não são predominantes, mas a análise estruturada e o uso do MER substituem esse processo.

Em seu processo de desenvolvimento você faz uso de:			
UML	27%	23%	24%
Fusion	2%	1%	1%
Análise Estruturada	33%	48%	43%
MER (Modelo de Entidade Relacional)	38%	29%	32%

Tabela 34 – O uso de linguagens de modelagem e metodologias de desenvolvimento.

Uma conduta bastante exigida por parte das empresas é a boa documentação dos sistemas nos mais diversos níveis e etapas de seu desenvolvimento.

O resultado da pesquisa enfatiza os procedimentos adotados pelos profissionais da área de desenvolvimento em relação à documentação. A pesquisa revela que são poucos os profissionais que elaboram a documentação somente após os testes e respectiva simulação do software, conforme tabelas 35 e 36. Isto demonstra claramente que o processo de documentação está presente em fases

anteriores ao processo de desenvolvimento, no qual a maioria dos pesquisados apontam uma documentação para cada método adotado no projeto.

Em sua opinião, o entendimento dos requisitos do software a ser desenvolvido tem como origem:			
O detalhamento da descrição dos próprios requisitos	46%	41%	43%
As conclusões obtidas a partir das reuniões com o cliente	31%	31%	31%
As conclusões obtidas a partir das reuniões com a equipe	24%	15%	18%
A orientação da chefia	0%	12%	9%

Tabela 35 – O entendimento dos requisitos de software nas empresas.

Em relação à documentação do software aplicativo, você costuma efetuar:			
Comentários por linha, depois de cada comando inserido	21%	24%	23%
Comentários relativos a uma classe de objetos	24%	23%	24%
Comentários para cada método	53%	44%	46%
Documentação somente depois de ter o programa funcionando	2%	9%	7%

Tabela 36 – Procedimentos adotados em relação à documentação.

Com a introdução de programas de certificação de qualidade nas empresas, os testes são fatores fundamentais para garantir o nível de qualidade desejado. As respostas obtidas na pesquisa realizada, conforme ilustração da tabela 37, refletem que a grande preocupação está na realização de testes que possam validar o produto final em relação à sua especificação. No entanto, existe pouca preocupação na verificação desta especificação ou mesmo em submeter o programa no ambiente e nas condições em que o cliente se encontra.

Em se tratando de testes, você costuma efetuar:			
Verificação	13%	26%	22%
Validação	38%	44%	42%
Testes com um conjunto de dados pré-determinados	23%	16%	18%
Testes com com o ambiente e condições de operação do cliente	26%	14%	18%

Tabela 37 – Procedimentos adotados para os testes.

Um dos aspectos bastante desenvolvido nos meios empresariais é o conhecimento da produtividade e a correta estimativa de prazo e de custo. Os resultados da pesquisa revelam que o planejamento e o gerenciamento do desenvolvimento de software contam com a colaboração de profissionais que conhecem o tempo necessário para produzir uma determinada rotina e, conseqüentemente, conseguem cumprir os prazos designados. Este resultado pode ser enfatizado com a elaboração de um projeto antes da prototipação por parte da maioria dos profissionais de desenvolvimento de software, conforme os dados consolidados nas tabelas 38 e 39.

Em se tratando de produtividade, é possível afirmar que:			
Você consegue cumprir o prazo designado	16%	38%	33%
Você conhece o número de linhas (códigos) que consegue produzir por dia/semana/mês	0%	2%	1%
Você conhece o tempo necessário para produzir uma determinada rotina ou programa	43%	33%	36%
Você conhece a produtividade da sua equipe	41%	27%	31%

Tabela 38– Nível de conhecimento da produtividade.

As aplicações que você desenvolve tem um ante projeto e um projeto antes da prototipação?			
( ) Sim	67%	63%	64%
( ) Não	33%	35%	35%
( ) Não sei	0%	2%	1%

Tabela 39 – A elaboração de projetos antes da prototipação.

8 CONSIDERAÇÕES FINAIS

A pesquisa conseguiu atingir satisfatoriamente seus objetivos, ou seja, foi possível traçar o perfil do desenvolvedor de software aplicativo no Brasil, entendendo o seu *background* social e cultural, assim como sua competência profissional e com isso ter uma idéia melhor sobre a competitividade do país dentro do panorama mundial.

Os nossos desenvolvedores são jovens, o que não difere muito dos desenvolvedores de software do resto do mundo, cheios de energia, dinâmicos e criativos. Oriundos, em sua maioria das camadas de classe média baixa, mostrando não apenas o grande crescimento do setor, como também que trabalhar com software é uma boa oportunidade de ascensão social e profissional. Por outro lado, uma vez que esses profissionais, em sua maioria, tiveram um ensino fundamental e médio deficientes, pode ser um fator limitante em sua atuação profissional, porque um bom desenvolvedor, além de conhecer as técnicas de software, precisa de maturidade científica, uma boa bagagem humanística e cultural para que tenha uma atuação forte dentro do cenário mundial. Esse fator pode ser significativo, uma vez que nos países que competem com o Brasil no mercado de software, os desenvolvedores geralmente possuem mestrado ou especialização equivalente.

O fato dos desenvolvedores ter aprendido os seus *skills* de programação no trabalho ou em cursos de treinamento patrocinados pelas suas empresas, faz-nos repensar sobre o currículo dado em nossas universidades. Caso esses treinamentos em técnicas e conhecimentos de trabalho somente sejam possíveis dentro da empresa, devido à particularidade da atuação de cada empresa, as universidades deverão dedicar-se em estabelecer uma base sólida para os profissionais de software, deixando a apresentação de ferramentas,

linguagens e componentes para os cursos internos de treinamento nas empresas em que irão atuar.

Se esse for mesmo o papel da universidade, ou seja, o de fornecer uma base de conhecimento sólida e consistente ao aluno, a participação das empresas de ponta no processo educacional será fundamental fornecendo disciplinas extra-curriculares, agora inclusive obrigatórias por determinação do MEC, apresentando suas ferramentas mais atuais e mostrando as perspectivas do mercado, motivando o aluno e reduzindo os atuais índices de desistência apresentado nas faculdades.

Por outro lado, notamos que os desenvolvedores estão bem informados sobre as novas tecnologias, seja nas técnicas de projeto de software como nas ferramentas e metodologias de programação. Seu uso no dia-a-dia, no entanto, ainda não está disseminado. Algumas técnicas imprescindíveis como incorporação de rotina de manutenção de integridade, detecção de erro, comentários e documentação ainda não foram totalmente incorporados ao hábito dos programadores. O problema pode estar tanto nos desenvolvedores, como nas empresas em que eles trabalham.

Em termos do uso das técnicas de Engenharia de Software, o retrato espelha tanto a atuação do desenvolvedor individualmente como da sua empresa. Na especificação e projeto de software quase todos utilizam alguma metodologia. A grande parte ainda usa técnicas tradicionais como fluxograma e DFD. Uma parte significativa dos profissionais ainda não faz prototipagem rápida para validar antes o desenvolvimento das aplicações. Na implementação, o desconhecimento ou o não uso de sincronismo de processo como semáforo pode significar que as aplicações não contêm seções críticas

ou as ferramentas de banco de dados escondem esses pormenores de implementação que não são percebidos pelos programadores.

A pesquisa constatou também que as empresas que implantaram processos de certificação de qualidade como o CMM têm, principalmente, a nível gerencial, grande preocupação no uso de ferramentas e metodologias que assegurem um desenvolvimento de software mais adequado às exigências tecnológicas atuais, notadamente ligados à Engenharia de Software.

As empresas em geral mostram uma grande preocupação com a qualidade e a produtividade, incentivando seus funcionários a participarem de treinamento principalmente em Engenharia de Software. Entretanto, a implementação dessas técnicas e o uso das ferramentas de produtividade ainda estão longe do ideal.

Essa pesquisa trouxe muitas conclusões interessantes para entendermos melhor a qualificação da mão-de-obra em software no Brasil. Em primeiro lugar, ter mão-de-obra qualificada em software implica em elevar a qualidade do nosso ensino fundamental e médio, tanto nas áreas exatas, como matemática e física, como nas áreas humanas como Português e Inglês. Em segundo lugar, os cursos de reciclagem e atualização tecnológica devem ser uma das tarefas prioritárias de todas as empresas de software, não só para elevar e suprir as deficiências do conhecimento técnico científico dos seus funcionários, como servem para melhor os hábitos e as atitudes na execução dos projetos. Para trabalhos futuros, sugerimos que a mesma pesquisa seja feita a cada dois anos, ampliando e refinando suas perguntas. Baseado nos resultados obtidos nessa pesquisa, um método estatístico para validar o tamanho da amostra deve ser estudado com mais acuidade.

REFERÊNCIAS BIBLIOGRÁFICAS

[BARB2001] BARBOSA, L. M. **Ensino de algoritmos em cursos de computação**. São Paulo: EDUC, 2001.

[CARB1983] CARBONELL, J. G. **Langley P. Learning by Analogy. Machine Learning: An Artificial Intelligence Approach.** Tioga Publishing Company, CA, 1983, p. 137-162.

[CHAV2003] CHAVES, E. O. C. **People LOGO: Uma Introdução.** Disponível em: http://www.chaves.com.br/TEXTSELF/EDTECH/peoplelogo.htm; [24.01.2003]

[CONS2000] CONSTANTINO-GONZÁLEZ, M.A.; SUTHERS, D.D. A Coached Collaborative Learning Environment for Entity-Relationship Modeling. In: INTELLIGENT TUTORING SYSTEMS. **Proceedings of the 5th International Conference (ITS 2000), G. Gauthier, C. Frasson and K. Van Lehn (Eds.)**. Springer-Verlag: 2000. p. 325-333.

[DEJO1986] DEJONG, G.; MOONEY, R. J. **Explanation-Based Learning: An Alternative View.** Machine Learning 2, Vol. 1, 1986, p. 145-176.

[FOSN1998] FOSNOT, C. T. **Construtivismo – teoria, perspectivas e prática pedagógica.** Tradução de Sandra Costa. Porto Alegre: ArtMed, 1998.

[GOME2002] GOMES, A.; Mendes, A. Suporte à aprendizagem da programação com o ambiente SICAS. **Atas do V Congresso Ibero-Americano de Informática Educativa**. Viña del Mar, Chile, 2002.

[HABE1975] HABERKORN, E. **Computador e Processamento de Dados**. São Paulo: Editora Atlas, 1975.

[HABE1986] HABERKORN, E. **O Computador na Administração de Empresas**. São Paulo: Editora Atlas, 1986, p. 60.

[HABE1991] HABERKORN, E. **Clipper: modelos de programas**. São Paulo: Editora Atlas, 1991.

[HABE1996] HABERKORN, E. **Delphi 3.0 – Desenvolvendo uma Aplicação Comercial**. São Paulo: Treinart Editora, 1996.

[HABE1999] HABERKORN, E. **Teoria do ERP**. São Paulo: Makron Books, 1999.

[KAHN2002] KAHN, K. **Animated Programs**. Disponível em: http://www.utad.pt/~leonelm/JVLC-Portugues.html; [09.12.2002]

[LAKA2000] LAKATOS, E. M. **Metodologia cienfífica.** São Paulo: Atlas, 2000.

[LAR2004] LARSON, R. & FARBER B. **Estatística Aplicada.** São Paulo: Pearson, 2004

[NAUR1969] NAUR, P. B. **Software Engineering: A Report on a Conference Sponsored by the NATO Science** Committee. NATO, 1969.

[MEC2002] **Censo Escolar**. Disponível em: http://www.inep.gov.br/download/estatisticas/grandes_numeros/2001/folder_brasil.xls; [12.03.2003]

[MCT2002] A indústria de software no Brasil 2002: fortalecendo a economia do conhecimento /MIT. Capítulo Brasil do Projeto *Slicing the Knowledge-Based Economy in India, China and Brazil*

[MEND2002] MENDES, A. J. N. **Software educativo para apoio à aprendizagem de programação**. Disponível em: http://www.c5.cl/ieinvestiga/actas/tise01/pags/charlas/charla_mendes.htm; [03.12.2002]

[MONA2003] MONARD, M. C.; BATISTA, G. E. A. P. A.; KAWAMOTO, S; PUGLIESI, J.B. **Uma Introdução ao Aprendizado Simbólico de Máquina por Exemplos**. Disponível em: http://labic.icmsc.sc.usp.br/didatico/PostScript/ML.html; [24.01.2003]

[PACI2003] PACITTI, T. **Do Fortran à Internet : Construindo o Futuro Através da Educação**, 3.ed. São Paulo: Pinoneira Thompson Learning, 2003.

[PAPE1993] PAPERT, S. **Children's Machines**. Basic Books, 1993.

[PITA2002] PITASSI, C.; LEITÃO S. P. Tecnologia de Informação e Mudança: Uma abordagem crítica. **Revista de Administração de Empresas – RAE,** v. 42, n.1, p. 77-87, abril/junho 2002.

[PERR2000] PERRENOUD, P. **Dez novas competências para ensinar.** Porto Alegre: Artes Médicas Sul, 2000.

[PRES1995] PRESSMAN, R. S. **Engenharia de Software**. São Paulo: Makron Books, 1995.

[PROU2000] PROULX, V.K. Programming Patterns and Design Patterns in the Introductory Computer Science Course. **Proceedings of the 31st SIGCSE**. Auxtin, TX, USA, march, 2000, p. 7-12.

[RAGG1992] RAGGET, J.; Bains, W. **Artificial Intelligence from A to Z**. Chapman & Hall, 1992.

[ROBE2001] ROBERTS, E. . An overview of MiniJava. **Proceedings of the 32nd SIGCSE Techical Symposium on Computer Science Education.** 2001, p. 1–5.

[RICH1993] RICH, E.; KNIGHT, K. **Inteligência Artificial**, 2. ed. São Paulo: Makron Books, 1993.

[SEES1998] SECRETARIA DE EDUCAÇÃO SUPERIOR. **Diretrizes Curriculares de Cursos na Área de Computação e Informática.** Comissão de Especialistas de Ensino de Computação e Informática – CEEInf, 1998, 24 p.

[SEPI2002] SECRETARIA DE POLÍTICA DE INFORMÁTICA. **Qualidade e Produtividade no Setor de Software Brasileiro.** Brasília, 2002, 258 p.

[SOLO1986] SOLOWAY, E. Learning to Program = Learning to Construct Mechanisms and Explanations. **Com. Of the ACM, 29(9)**. September 1986.

[SUTH1998] SUTHERS, D.D., **Computer Aided Education and Training Initiative.** Technical Report, Learning Research and Development Center, University of Pittsburgh, 1998.

[YIN2001] YIN, R. K. **Estudo de Caso: Planejamento e Métodos.** Trad.: Daniel Grassi. 2. ed. Porto Alegre: Bookman, 2001.

[VIGO1999] VIGOTSKI, K. S. **A Formação Social da Mente**. São Paulo: Martins Fontes, 1999.

[YODE1994] YODER, S. Discouraged?.. Don't Dispair! [sic], Logo Exchange. **Revista do Special Interest Group do ISTE para educadores que usam Logo**, v. 12, n. 4, Verão de 1994.

[ZELL2000] ZELLER, A. Making Students Read and Review Code. ACM ITICSE, 2000.

APÊNDICE I

A FORMAÇÃO NOS CURSOS SUPERIORES NA ÁREA DE COMPUTAÇÃO E INFORMÁTICA IDEALIZADA PELO MEC

Construir cursos de formação na área de computação e informática que efetivamente incluam metas, objetivos e conteúdos eficazes, eficientes e capazes de superar as expectativas e necessidades do mercado é a constante preocupação de especialistas e profissionais da educação.

Torna-se imperativo que essa preocupação seja realmente constante, uma vez que o surgimento de novas linguagens e a evolução tecnológica também o são.

As diretrizes da Secretaria da Educação Superior sintetizam essa preocupação e estabelecem a orientação de uma estrutura básica a ser detalhada a fim de possibilitar o desenvolvimento, criação e implementação curricular de disciplinas que irão compor os cursos nas áreas de Computação e Informática.

Essas Diretrizes foram consolidadas através da colaboração de diversos professores e de resultados obtidos no Workshop de Educação em Computação (WEI/98), no Seminário dos Consultores do SESu/roberta.naufal@citigroup.com <roberta.naufal@citigroup.com> (Belo Horizonte, agosto/1998) e através de contribuições a eles enviadas em decorrência do Edital Nº 4.

As Diretrizes Curriculares compõem um documento que justifica e esclarece a nomenclatura utilizada na área, estabelece os objetivos da formação e a estrutura curricular básica para os cursos de formação

e descreve como efetuar o detalhamento das disciplinas para a formação dos cursos na área de Computação e Informática.

Embora o termo informática seja amplamente utilizado para a denominação de tudo o que se relaciona a computador, o termo Computação representa a área sob o aspecto da formação de recursos humanos e também sob o aspecto do desenvolvimento científico e tecnológico. Para atender a esses diferentes aspectos, o documento que descreve as Diretrizes Curriculares identifica a área através dos termos Computação e Informática.

No que diz respeito à formação dos recursos humanos na área de Computação e Informática, o documento das Diretrizes Curriculares identifica os objetivos a serem atendidos, esclarecendo que:

"... Os cursos da área de computação e informática têm como objetivo a formação de recursos humanos para o desenvolvimento tecnológico da computação (hardware e software) com vistas a atender necessidades da sociedade, para a aplicação das tecnologias da computação no interesse da sociedade e para a formação de professores para o ensino médio e profissional ..."

Além de estabelecer os objetivos de forma genérica, o documento de Diretrizes Curriculares descreve os cursos de forma mais detalhada e destaca as necessidades da sociedade:

- armazenamento de grandes volumes de informações;
- criação de mecanismos eficientes para a recuperação de dados;
- processamento de problemas e cálculos matemáticos de forma eficiente;
- eficiência e segurança na comunicação de dados;

- eficiência no processamento de cálculos que envolvem grande volume de dados;
- processamento de imagens;
- desenvolvimento de ferramentas e jogos que possam auxiliar o processo de aprendizado.

O documento inclui ainda as necessidades da sociedade nas áreas de Computação e Informática classificando os sistemas por área de aplicação. Em destaque estão: os sistemas de informações econômicas, financeiras e administrativas; os sistemas de processamento de imagens; sistemas na área de saúde; sistemas para controle de tráfego aéreo; sistemas bancários e de comunicação através da Internet.

Mais do que a informação básica e o conhecimento necessário para a formação de futuros profissionais, a Secretaria de Educação Superior, através do documento de Diretrizes Curriculares, orienta uma formação de futuros cidadãos críticos e conscientes de seu papel perante a sociedade.

Nesse aspecto, a formação contempla conteúdos que garantam o desenvolvimento de profissionais que possam manter perfeita harmonia entre as capacidades humanas e sociais e o próprio desenvolvimento tecnológico. Para o desenvolvimento de um profissional com esse perfil, são orientadas as **áreas de formação básica, tecnológica, complementar e humanística**. Cada uma dessas áreas concentra um conjunto de conhecimentos e abordagens que devem ser enfatizados e detalhados dependendo do perfil que venha a ser caracterizado.

A área de **formação básica** contempla as seguintes disciplinas: Ciência da Computação englobando Programação, Computação,

Algoritmos e Arquitetura de Computadores. As disciplinas de Matemática, Física, Eletricidade e Pedagogia são classificadas ainda dentro dessa área.

Na área de **formação tecnológica**, estão classificadas as disciplinas que oferecem abordagens a respeito de Sistemas Operacionais, Redes de Computadores, Sistemas Distribuídos, Compiladores, Banco de Dados, Engenharia de Software, Sistemas Multimídia, Interface Homem-Máquina, Realidade Virtual, Inteligência Artificial, Computação Gráfica, Processamento de Imagens e Prática do Ensino de Computação.

A área de **formação complementar** tem por objetivo introduzir disciplinas de outras áreas, uma vez que o profissional da área de Computação e Informática desenvolve e implementa soluções para problemas pertinentes as mais diversas áreas, seja administrativa, financeira, econômica etc. Assim sendo, a área de formação complementar tem por finalidade fornecer o conhecimento de outras áreas ao futuro profissional.

A História da Ciência da Computação, o Empreendedorismo, a Ética, a Sociologia, a Filosofia incluindo uma análise dos impactos do uso do computador na sociedade são abordagens categorizadas na área de **formação humanística.**

APÊNDICE II

CURSOS DE NÍVEL SUPERIOR NA ÁREA DE CIÊNCIAS DA COMPUTAÇÃO E SISTEMAS DE INFORMAÇÃO

1 Cursos em Ciência da Computação ou Engenharia da Computação

O documento de Diretrizes Curriculares de cursos da área de Computação e Informática, consolidado pela Secretaria de Ensino Superior do MEC [SEES1998], estabelece a orientação de quatro categorias de cursos. São elas:

1.1 Bacharelado em Ciência da Computação ou Engenharia da Computação

Os cursos de bacharelado em Ciência da Computação ou Engenharia da Computação são cursos para a formação de profissionais que tenham como atividade fim o desenvolvimento e criação de programas.

Esses cursos devem possibilitar que o futuro profissional seja capaz de manter as atividades de pesquisa em suas atividades rotineiras de trabalho, objetivando promover o desenvolvimento científico e tecnológico através do desenvolvimento de projetos de software e hardware.

1.2 Bacharelado em Sistemas de Informação

Os cursos de bacharelado em Sistemas de Informação são cursos voltados para à formação de profissionais que mantenham a computação como atividade meio. Esses cursos têm por objetivo

formar profissionais que possam idealizar, desenvolver, projetar e implantar soluções em sistemas de informação, a fim de resolver problemas encontrados nas organizações.

Os cursos de bacharelado em Sistemas de Informação devem oferecer ao futuro profissional a capacidade de usar de forma eficiente as tecnologias disponíveis dentro das organizações.

1.3 Licenciatura em Computação

Os cursos de Licenciatura em Computação têm por objetivo a formação de futuros educadores da área de Computação e Informática em instituições de Ensino Médio.

1.4 Tecnologia

Os cursos de tecnologia são os chamados cursos de curta duração, justamente em função da demanda a que se propõem atender. Seu principal objetivo é suprir o mercado de trabalho de profissionais que detenham recursos tecnológicos para atender as necessidades imediatas e urgentes.

O curso de tecnologia é uma forma eficiente para atender à demanda nas áreas de Computação e Informática, em função de seu alto grau de dinamismo e rapidez com novos recursos tecnológicos que surgem.

2 Disciplinas das áreas de Computação e Informática

Em conformidade com as diretrizes curriculares estabelecidas pela Secretaria de Educação Superior, os cursos nas áreas de Computação e Informática devem enfocar conceitos de hardware e de

software. Esses conceitos devem ser ministrados através de um conjunto de disciplinas que abordem diversos assuntos, a fim de integrar o conhecimento necessário e capacitar o aluno a desenvolver e propor soluções. Esses assuntos são classificados como sub-áreas:

2.1 Programação

- Conceitos e princípios fundamentais: Abstração, Associação, Avaliação, Atribuição, Chamada de procedimento, Envio de mensagens, Passagem de parâmetros, Herança, Polimorfismo e Encapsulamento;
- Solução de problemas através de algoritmos: Especificação, Projeto, Validação, Modelagem e Estruturação de programas e dados;
- Linguagens de programação;
- Técnicas de programação: Modularização e Estruturação;
- Tipos de programação: Imperativa, Funcional, Lógica e Orientada a objetos;
- Desenvolvimento de Algoritmos: Estruturas de Dados, Processos de busca e Técnicas de ordenação.

2.2 Computação e Algoritmos

- Conceitos e princípios fundamentais: algoritmos e modelos de computação;
- Linguagens formais: uma abordagem do que pode ser resolvido através da computação;
- Linguagens formais: sintaxe e semântica;
- Eficiência de algoritmos e estruturas de dados.

2.3 Arquitetura de Computadores

- Unidade Central de Processamento: características internas;
- Organização funcional: acesso à memória e funcionamento dos diversos periféricos;
- Projeto de computadores: sistema de memória, barramento e comunicação com periféricos;
- Projeto Lógico: lógica digital, interrupções e microprogramação.

2.4 Matemática

- Matemática discreta: linguagens, autômatos e métodos, gráficos e aritmética intervalar;
- Matemática do contínuo: cálculo diferencial, integral, álgebra linear, geometria analítica e cálculo numérico.

2.5 Física e Eletricidade

- Leis básicas da eletricidade;
- Representação matemática e unidades de medidas das grandezas elétricas;
- Princípio de operação dos dispositivos semicondutores;
- Teoria eletromagnética e ondas;
- Fenômenos ópticos.

2.6 Pedagogia

- A pedagogia e seus impactos sociais;
- Dispositivos pedagógicos: tecnologias, métodos e estratégias;
- Configuração escolar: organização curricular;

2.7 Sistemas Operacionais

- Gerenciamento de processos: comunicação, sincronização, escalonamento e resolução de conflitos;
- Gerenciamento de memória: endereçamento, hierarquias de memória e memória virtual;
- Gerenciamento de arquivos: diretórios, endereçamento, acesso, segurança, compartilhamento e proteção;
- Gerenciamento de entrada e saída: interrupções, dispositivos, interfaces e controladores de acesso.

2.8 Redes de Computadores

- Conceitos e princípios básicos: topologia, modos de transmissão, multiplexação, modulação, comutação e técnicas de detecção de erros;
- Protocolo de comunicação e serviços de rede: tipos de protocolos, serviços e funções das diversas camadas de rede;
- Tipos de rede: redes locais, metropolitanas e geograficamente distribuídas;
- Gerenciamento de redes: projeto, gerenciamento e avaliação de desempenho;
- Segurança de redes;
- Prática de serviços de redes (laboratório): instalação, gerência e segurança.

2.9 Sistemas Distribuídos

- Conceitos e princípios fundamentais: algoritmos distribuídos, sistemas operacionais e kernels, ambientes de programação, linguagens, base de dados, sistemas multimídias, sistemas em tempo real;
- Sistemas abertos: padrões utilizados;
- Utilização da orientação a objetos;

- Suportes disponíveis para os sistemas distribuídos: sistemas abertos, padrões, orientação a objetos, técnicas de escalonamento;
- Inteligência artificial distribuída (IAD): sistemas multi-agentes e resolução distribuída de problemas.

2.10 Compiladores

- Conceitos e princípios fundamentais: modularidade, portabilidade e custos de software;
- Classificação dos compiladores: ambientes para linguagens de programação, ambientes para o processamento de linguagens naturais e ferramentas para compatibilização entre dispositivos de hardware;
- Estrutura dos compiladores;
- Análise de programas-fonte: sintaxe, semântica, tabela de símbolos e gerenciamento de erros;
- Ferramentas para a geração automática dos componentes de um compilador;
- Máquinas abstratas e otimização de código intermediário;
- Ambientes de tempo de execução;
- Síntese de programas-objeto.

2.11 Banco de Dados

- Problemas relativos aos dados: organização, modelagem, integridade, armazenamento, integração, distribuição e empacotamento;
- Sistemas de gerenciamento de banco de dados SGBD: arquitetura, interfaces, linguagens de interpretação, processamento de consultas, controle de concorrência,

recuperação, segurança, indexação, gerenciamento de *buffers* e arquivos.

2.12 Engenharia de Software

- Conceitos e princípios fundamentais: características, histórico e evolução de software;
- Desenvolvimento e manutenção de software: requisitos, análise, arquitetura, projeto, programação, testes, manutenção, qualidade e gestão;
- Gerência de projeto de software: estimativa de recursos, análises de custo-benefício, planejamento do desenvolvimento e montagem das equipes, gestão do processo e produto de software;
- O processo de desenvolvimento de software: métodos de engenharia de requisitos, análise e projeto de software, técnicas de programação, técnicas de geração de documentação e técnicas de teste.

2.13 Sistemas Multimídia

- Conceitos e princípios fundamentais: programação visual, editoração, composição, retórica, comunicação e cognição;
- Técnicas e conceitos: computação gráfica, computação sônica e construção de peças multimídia;
- Tecnologia multimídia: Java, Open GL, Midi, Java Sound;
- Aplicação multimídia: publicação científica, tutoriais em qualquer área de conhecimento, medicina cirúrgica, marketing, arte e entretenimento.

2.14 Interface homem-máquina

- Interfaces de interação homem-máquina: técnicas de projetos de sistemas interativos;
- Técnicas que dão suporte aos elementos de usabilidade: técnicas analíticas, empíricas e modelos cognitivos.

2.15 Realidade Virtual

- Conceitos e princípios fundamentais: computação gráfica tridimensional;
- Plataformas computacionais de alto desempenho;
- Interfaces tridimensionais;
- Dispositivos multisensoriais de entrada e saída;
- Softwares e linguagens para desenvolvimento da aplicação da realidade virtual;
- Modelagem e animação tridimensional.

2.16 Inteligência Artificial

- Conceitos e princípios fundamentais: representação de conhecimento, aprendizagem automática, métodos heurísticos, reconhecimento de padrões e processamento de linguagem natural;
- Representação do conhecimento simbólica: lógica, redes semânticas e *frames*;
- Representação do conhecimento não simbólica: redes neurais, algoritmos genéticos, redes bayesianas;
- Aplicações na área de IA: sistemas especialistas, robótica, sistemas de reconhecimento de voz e imagens, jogos, sistemas tutoriais inteligentes, tradutores automáticos, mineração de dados, recuperação de informação e interfaces adaptativas;

- O uso de técnicas de IA em outras áreas: banco de dados, engenharia de software, sistemas distribuídos, redes de computadores, computação gráfica e informática na educação.

2.17 Computação Gráfica e Processamento de Imagens

- Representação e comunicação de imagens: digitalização, visualização e adequação em modelos matemáticos;
- Técnicas de geração de imagens: tratamento da informação pictorial;
- Processamento de imagens: realce, filtragem, restauração, análise e reconstrução.

3 Cursos de Formação Complementar

As áreas de Computação e Informática são conhecidas pelo dinamismo e velocidade com que são introduzidas novas ferramentas e novos produtos. É importante que seja caracterizada a distinção entre a formação básica e fundamental da informação obtida a respeito de ferramentas e produtos.

Uma formação básica e fundamental envolve o conhecimento de conceitos genéricos e conteúdos teóricos, que oferecem ao aluno instrumentos eficientes para capacitá-lo na análise, desenvolvimento e aplicação de forma multidisciplinar.

4 Cursos extra curriculares

Este conhecimento básico e conceitual é complementado, nas áreas de Computação e Informática, através do conhecimento quanto ao uso de ferramentas e produtos que sejam capazes de viabilizar sua aplicação. Em última análise, distinguir um conhecimento mais

eclético de um conhecimento menos eclético, um conhecimento mais teórico de um conhecimento mais prático.

Esse conhecimento mais prático, quanto ao uso de ferramentas e produtos existentes no mercado, é complementado através de cursos chamados extra curriculares.

Nas áreas de Computação e Informática, diversas entidades e instituições surgiram para prestar esse serviço. Algumas dessas entidades, bastante sintonizadas e em perfeita sincronia com o mercado, oferecem informações quanto ao uso e manuseio de ferramentas e produtos que ainda serão lançados em novas versões. São as entidades que visam oferecer treinamento de primeira linha, como por exemplo Dot Corp, IBPI, Impacta, Bráz & Figueiredo, etc.

Outras entidades tais como SOS, Bit Company, Microcamp, dentre outras, visam disseminar o conhecimento do uso e manuseio de ferramentas e produtos de Computação e Informática para uma grande parcela da população menos favorecida, através de cargas horárias menores e com conseqüentes custos mais acessíveis.

O mercado de trabalho reconhece o papel prestado por essas entidades de tal modo que muitas delas, visando manter uma posição de destaque, oferecem certificações de seus principais patrocinadores tais como Microsoft, Oracle, Sun, Cisco, Novell, Borland, IBM, Macromedia, etc.

5 Conclusão do estudo sobre os cursos de nível superior

A Secretaria de Educação Superior do MEC traçou uma grade curricular contendo os aspectos básicos e relevantes para os cursos de Ciências da Computação e Sistemas de Informação sem tolher, porém,

a liberdade de escolha de cada uma das instituições de ensino em particular. Essa grade curricular está organizada de tal forma que possibilita o enriquecimento de seu conteúdo através do exercício da criatividade das instituições de ensino superior na concepção de currículos diversificados enfocando o perfil desejado para os alunos egressos de seus cursos.

Por outro lado há de se considerar a proliferação de diversos cursos extra curriculares que mantêm estrutura totalmente a parte e, perante o mercado, grande evidência a tal ponto de existir no Brasil um guia de certificação para as diversas áreas da computação e informática.

Atualmente, os softwares passam a ter pouca credibilidade caso não tenham em si um plano de certificação. Além da credibilidade, essas certificações impedem que o mercado seja contaminado por falsos profissionais da área. Nesse aspecto, essas certificações exercem um papel relevante na complementação da formação dos desenvolvedores de software aplicativo.