

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO
PUC–SP**

**A formação de profissionais em tecnologia da
informação na área de desenvolvimento
de software**

Carlos Olavo de Azevedo Camacho Júnior

**MESTRADO EM TECNOLOGIAS DA INTELIGÊNCIA E
DESIGN DIGITAL**

SÃO PAULO
2011

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO
PUC-SP**

**A formação de profissionais em tecnologia da
informação na área de desenvolvimento
de software**

Carlos Olavo de Azevedo Camacho Júnior

**MESTRADO EM TECNOLOGIAS DA INTELIGÊNCIA E
DESIGN DIGITAL**

Dissertação apresentada à Banca Examinadora da Pontifícia Universidade Católica de São Paulo, como exigência parcial para obtenção do título de MESTRE em Tecnologias da Inteligência e Design Digital com área de concentração em aprendizagem e semiótica cognitiva, sob a orientação da Prof^a. Dr^a Sônia Maria de Macedo Allegretti.

**SÃO PAULO
2011**

Banca Examinadora

Dedicado à memória de Danilo Hessel Reimberg.

A Deus, pela vida.

À minha esposa Catia, pelo amor e companheirismo.

Aos meus pais Carlos e Vanda, pelo amor incondicional.

Aos meus irmãos Henrique e Alline, pelo amor, carinho, amizade e incentivo de sempre.

Aos meus sogros Benedito e Maria, por acolherem-me como um filho desde que me conheceram.

Ao amigo Nelson Paz, pelas valiosas conversas sobre tecnologia e pela revisão do livro publicado durante esta jornada. Ao corpo docente e discente do CEFET-Cubatão por acreditar, apoiar e viver o projeto LOJA .NET em sua plenitude.

Aos amigos da PUC-SP, em especial aos professores Jorge Vieira e Sônia Allegretti, por mostrarem-me novos horizontes. À amiga Edna Conti pelo incentivo sempre perspicaz e em momento oportuno.

Aos amigos Milton Maruyama, Silvana Guardiano, Rosangela Compri, Maria Aparecida Pereira de Souza, Dirce Kazuko Kaneko, Gisele de Oliveira Belome, Marco Antonio Milanez, Héber Calixto e toda a equipe do Banco do Brasil S.A., pelo incentivo e por acreditar nas minhas idéias.

Aos amigos da célula de estudos Microsoft Objective! Club da FATEC-SP pelo compartilhamento de ideais. Em especial aos alunos Danilo Rocha e Felipe Tófoli, que demonstraram atitude em desbravar a linguagem C# .NET.

À Profa. Melania Dalla Torre, Prof. Avelino Pimenta e todos da UNIP – Ribeirão Preto pelo apoio e carinho cativante.

À Laura Generini, Michele Beck Schröer, Nilton e toda a equipe da Editora VisualBooks por patrocinar o projeto de publicação do livro.

Aos leitores do livro por acreditarem que através dele é possível alavancar seus conhecimentos em desenvolvimento de software.

A todos que ajudaram direta ou indiretamente para a realização deste trabalho.

Resumo

O trabalho de pesquisa realizado tem como objetivo geral conhecer em que medida o desenvolvedor de software - estudante de um curso técnico de informática - pode se beneficiar com a utilização de um projeto de aplicação para a assimilação da metodologia de desenvolvimento em camadas.

Atualmente empresas de desenvolvimento de software buscam excelência na prestação de serviços. Com isso acompanhamos o surgimento de diversas ferramentas para o aumento de produtividade. Dentre elas destaca-se a IDE (Integrated Development Environment - Ambiente de Desenvolvimento Integrado) e conhecer essas ferramentas torna-se essencial para este profissional. Mas o conjunto de competências (Perrenoud, Le Boterf) exigidas pelo profissional desenvolvedor de software não se restringe aos aspectos técnicos. Analisamos neste trabalho quais as competências necessárias ao profissional de desenvolvimento de software. Essa análise tem base no pensamento complexo de Edgar Morin.

Visando demonstrar ainda a importância dos padrões quando se trata de tecnologia, buscamos na história da própria Internet (Vint Cert, Demi Getsko) exemplos para analisar a necessidade de padronização. Pesquisamos um padrão denominado "metodologia de desenvolvimento em camadas". Para analisar com maior profundidade a importância dos padrões, criamos neste trabalho um material didático (projeto LOJA .NET) que permitiu ao aluno exercitar conceitos sobre desenvolvimento de software. Tendo analisado as competências exigidas para o profissional de desenvolvimento de software bem como a importância do uso de padrões nesta profissão, analisamos se o projeto LOJA .NET contribui com este profissional no uso de suas habilidades e competências para compreender como utilizar padrões no processo de desenvolvimento de software.

Com o apoio do IFET-SP Instituto Federal de Educação, Ciência e Tecnologia de São Paulo realizamos uma pesquisa qualitativa onde alunos do curso técnico em informática utilizaram o projeto proposto. Utilizar conhecimentos que o aluno já possui; conhecer o que este aluno é capaz de fazer com a ajuda do professor ou de outro aluno e atuar nesse intervalo, ou seja, no caminho que o aluno vai percorrer até que consiga realizar sozinho o que atualmente só consegue realizar com o auxílio de alguém são conceitos advindos da teoria de Vygotsky.

Com a análise desta pesquisa buscamos identificar a relevância do uso de projetos de aplicação na formação do profissional de desenvolvimento de software.

Palavras-chave: desenvolvimento de software, desenvolvimento em camadas, microsoft c# .net, projeto de aplicação, metodologia.

Abstract

This research aims to verify how application projects can help the software developer professional to acquire knowledge about the 3-tier development model.

Nowadays companies want to prove that their services are accomplished with the best quality standards of information technology. To do that many tools are created and a special one is called IDE Integrated Development Environment. An IDE allows the software developer professional to bring efficacy into his project.

We will discuss what skills are important to this professional. This research will analyze the question: Why do companies based on information technology like to use processes with patterns? To demonstrate the importance of patterns in technology field, we search the Internet history (Vint Cert, Demi Getsko) to discuss the patterns used in Internet history. We have written articles about .NET development to demonstrate how to use patterns in a software development project.

This research originated a book titled "Desenvolvimento em Camadas com C#.NET (available in portuguese only)" who was used by IFET-SP students in a study.

We'll analyze this study to get indicators about the efficiency of application project in the educational process of the software developer professionals.

Key-words: software development, 3-tier development model, Microsoft C#.NET, application project, methodology.

Lista de Figuras

Figura 1 – Representação das camadas da Internet	42
Figura 2 – Representação do Modelo de Desenvolvimento em Camadas	47
Figura 3 – Capa do livro Desenvolvimento em Camadas com C# .NET	63
Figura 4 – Palestra proferida no CEFET	64

Lista de Tabelas

Tabela 1 – Relação de disciplinas do Módulo I – CEFET	54
Tabela 2 – Relação de disciplinas do Módulo II – CEFET	54
Tabela 3 – Relação de disciplinas do Módulo III – CEFET	55
Tabela 4 – Relação de disciplinas do Módulo IV – CEFET	55

Lista de Gráficos

Gráfico 1: Conceitos de Programação Orientada a Objetos	69
Gráfico 2: Conceito de Desenvolvimento em Camadas	71
Gráfico 3: Integridade Referencial em Banco de Dados	73
Gráfico 4: Criação de Tabelas em Banco de Dados	75
Gráfico 5: Criação e uso de Stored Procedures	77
Gráfico 6: Funcionamento da construção de Aplicações Windows	79
Gráfico 7: Funcionamento da construção de Aplicações Web	81
Gráfico 8: Como funciona o uso da Segurança da Informação	83
Gráfico 9: Conhecimentos gerais em Desenvolvimento de Sistemas	85
Gráfico 10: Avaliação do projeto LOJA .NET pelos alunos	87

SUMÁRIO

INTRODUÇÃO.....	13
1	CONSTRUÇÃO DE SOFTWARE E AS COMPETÊNCIAS NECESSÁRIAS
1.1	Projeto de Aplicação na Construção de Software..... 18
1.1.1	O início de um Projeto de Aplicação..... 19
1.2	Construção de Software X Competências necessárias..... 20
1.2.1	Aspectos técnicos e perfil comportamental..... 21
1.2.1.1	Aspectos técnicos..... 21
1.2.1.2	O Perfil Comportamental..... 30
1.3	As 5 competências analisadas com o projeto LOJA .NET..... 32
1.3.1	Qual o diferencial do projeto LOJA .NET?..... 33
1.3.1.1	Complexidade..... 33
1.3.1.2	Transdisciplinaridade..... 35
2	DESENVOLVIMENTO EM CAMADAS
2.1	É possível definir Internet?..... 40
2.2	Os padrões na Internet..... 41
2.2.1	Padrões no Desenvolvimento de Software..... 44
2.3	O Desenvolvimento de Aplicações em Camadas..... 46

3	ALUNOS DO CEFET	
3.1	Histórico do IFSP.....	51
3.2	Público-alvo da pesquisa: Alunos do Curso Técnico em Informática.....	53
3.3	O projeto LOJA .NET.....	57
3.3.1	A necessidade de definir ferramentas.....	58
3.3.2	Publicação de artigos.....	59
3.3.3	O livro “Desenvolvimento em Camadas com C# .NET”.....	62
3.4	Análise do Questionário Final.....	64
3.4.1	Contexto da Aplicação de Questionários.....	67
	CONSIDERAÇÕES FINAIS.....	89
	ANEXOS	
	Glossário.....	94
	Abreviaturas.....	99
	Questionário para o(a) Aluno(a).....	100
	Referências Bibliográficas.....	104

Introdução

Sou Bacharel em Ciência da Computação, possuo Licenciatura Plena em Matemática e especialização em Análise e Projeto de Sistemas.

Desde criança tenho fascínio pela informática, meu interesse vai desde tarefas de escritório como imprimir etiquetas auto-adesivas para endereçamento de correspondência até o desenvolvimento de sistemas complexos para grandes empresas como Instituições Financeiras.

Cursei o bacharelado em Ciência da Computação na Universidade Paulista UNIP de 1997 a 2000. Nesse período já me interessava no ensino de disciplinas como linguagem de programação e banco de dados. Não consegui me candidatar para monitoria nestas disciplinas pois como já trabalhava na Instituição em tempo integral não dispunha de tempo para atividades extras.

Um de meus professores que na época também era o coordenador do curso de Ciência da Computação reconheceu o meu interesse em lecionar e sugeriu que eu realizasse uma especialização. Em 2002 realizei uma pós-graduação em Análise e Projeto de Sistemas sob orientação deste mesmo professor.

Ainda em 2002 fui convidado pelo coordenador de cursos da área de Ciências Exatas da Universidade de Santo Amaro UNISA e lecionei disciplinas técnicas para alunos do último semestre do curso superior de Desenvolvimento de Websites.

Foi a partir daí que comecei a pesquisar estratégias que auxiliassem os alunos na assimilação de conteúdos na área de informática.

Em 2005, momento em que ingressei no cargo de Analista de Sistemas para o Banco Nossa Caixa S.A., conheci o amigo Nelson da Silva Paz, professor no Curso Técnico em Informática do CEFET¹ – Cubatão.

Além de trabalharmos para a mesma empresa, temos em comum o interesse no desafio relacionado a desenvolver situações de aprendizagem diferenciadas, estimulando a articulação entre saberes e competências. Temos em comum a preocupação de demonstrar aos alunos possibilidades de uso dos padrões de qualidade de desenvolvimento de *software*² exigidos pelo mercado e como utilizá-los na prática. Desde então passei a refletir constantemente sobre como criar uma situação onde o aluno possa adquirir as competências necessárias à sua atuação profissional como desenvolvedor de *software*.

Segundo o Prof. Nelson, seus alunos do Ensino Médio técnico em informática tinham boas avaliações nas disciplinas isoladamente, mas sentiam dificuldades no final do curso, momento em que era exigido a entrega do Trabalho de Conclusão de Curso. Neste trabalho final era preciso utilizar a soma dos saberes das diferentes disciplinas cursadas para construir um *software*.

No início de 2008 defini como público-alvo da minha pesquisa os alunos do Ensino Médio que estão em formação técnica da área de informática. Norteei meu problema de pesquisa com a questão: A utilização de projetos de aplicação dos conceitos de forma prática fundamentada no método de “Desenvolvimento em Camadas” contribui para que alunos de cursos técnicos em informática desenvolvam competências na construção de *software*?

¹ Centro Federal de Educação Profissional e Tecnológica.

² *Software* é uma sequência de instruções a serem executadas por um computador. *Software* também é o nome dado ao conjunto de produtos desenvolvidos no *processo de software*, onde incluem-se programas, manual de instalação, manual do usuário, dentre outros.

O presente trabalho teve como objetivo principal identificar em que medida a utilização de projetos de aplicação dos conceitos de forma prática fundamentada no método de “Desenvolvimento em Camadas” contribui para que alunos de cursos técnicos em informática desenvolvam competências na construção de *software*. Foi proposto um projeto para alunos a partir do 3º Semestre de um curso técnico em informática, e observamos o aproveitamento destes alunos após a execução deste projeto. Para atingir esse objetivo, alguns objetivos secundários foram pertinentes ao estudo:

Objetivo Geral

Tendo como público-alvo os alunos no Ensino Médio que estão em formação técnica na área de informática, no início de 2008 norteiei meu problema de pesquisa com a questão:

Verificar qual a contribuição do projeto de Desenvolvimento em Camadas para o desenvolvimento de competências na construção de *software*.

Objetivos Específicos

Definir projeto de aplicação na construção de *software*;

Analisar as competências necessárias ao desenvolvedor de *software*;

Analisar o método de desenvolvimento em camadas;

Realizar a análise de um grupo de alunos do curso técnico em informática que utilizou o método de desenvolvimento em camadas.

Metodologia

A análise quantitativa atua sobre os níveis de realidade e tem como objetivo básico trazer à luz dados, indicadores e tendências observáveis. A investigação qualitativa, pelo contrário, trabalha com valores, com crenças, representações, hábitos, atitudes e opiniões.

Realizamos uma pesquisa qualitativa, e utilizamos questionários como instrumento de coleta de dados.

Foi proposto um projeto para alunos a partir do 3º. Semestre de um curso técnico em informática, e observamos o aproveitamento destes alunos após a execução deste projeto.

Instrumentos:

Através de um questionário inicial de sondagem os alunos fizeram uma auto-análise do que conhecem sobre os assuntos relacionados no projeto de aplicação proposto. Após a realização do projeto aplicamos um questionário final a fim de verificar se os alunos perceberam avanços com relação aos tópicos envolvidos no projeto.

Estruturamos a dissertação em três capítulos.

O primeiro capítulo, denominado **Construção de Software e as Competências Necessárias**, traz definições do que vem a ser projeto de aplicação e analisa quais as competências exigidas pelo mercado atual de Tecnologia da Informação quando empresas desejam contratar desenvolvedores de *software*. Analisamos o que difere o método de desenvolvimento em camadas do ensino tradicional. Para essa análise tivemos como alicerce o pensamento complexo (Edgar Morin).

No segundo capítulo – **Desenvolvimento em Camadas** – realizamos um estudo aprofundado sobre os padrões existentes na Internet e sua importância. Tendo esse estudo como alicerce, buscamos justificar a importância do uso de padrões no Desenvolvimento de *Software*.

No terceiro capítulo realizamos uma análise qualitativa sobre as competências em desenvolvimento de *software* demonstradas pelos **Alunos do CEFET** onde colocamos em prática uma sugestão para demonstrar o uso de padrões no Desenvolvimento de *Software*. Primeiramente descrevemos os desafios

encontrados na idealização e criação de um material didático sobre o tema. Analisamos ainda aspectos relevantes observados durante a aplicação do material didático – que consiste no desenvolvimento de um projeto em Tecnologia da Informação – junto aos alunos do Curso Técnico do CEFET – Unidade Cubatão.

Através de um formulário inicial de sondagem verificamos o que os alunos conheciam sobre os assuntos envolvidos no projeto de aplicação proposto.

Após a execução do projeto pelos alunos, que teve duração de seis meses, aplicamos novo questionário para verificar se houve a percepção de melhorias no que se refere aos assuntos envolvidos no projeto realizado.

Não avaliamos o desempenho dos alunos após o término de sua formação, momento de ingresso no mercado de trabalho.

CAPÍTULO 1

Construção de Software e as competências necessárias

Quando se fala sobre a Construção de *Software*, se está necessariamente falando sobre Projeto de Aplicação. É preciso que haja um Projeto de Aplicação para que um *software* seja construído.

Dessa forma, antes de abordarmos a Construção de *Software* e as competências necessárias é importante exemplificar o que é e como surge um Projeto de Aplicação.

1.1 Projeto de Aplicação na Construção de Software

Por que as empresas contratam profissionais para a construção de um *software* se já existem tantos tipos de *software* a venda no mercado?

Os desenvolvedores de *software* são requisitados pelas empresas pois estas têm problemas específicos que precisam ser solucionados. As necessidades específicas de uma empresa não podem ser satisfeitas com um *software* criado para propósitos gerais. É neste cenário que surge o projeto de aplicação.

Imaginemos uma rede de supermercados com lojas espalhadas em diversos Estados. O gerente desta empresa deseja saber a quantidade de TVs de uma determinada marca que foi vendida por loja em cada dia da semana, ou ainda, é seu desejo saber o número exato de TVs vendidas até aquele instante.

A necessidade em analisar dados em tempo real está diretamente ligada à eficácia no gerenciamento de muitas empresas.

Um *software* capaz de prover o administrador de empresas com a informação atualizada a qualquer momento é chamado de SAD – *Software* de Apoio à Decisão.

Um *software* com essa característica não será encontrado à venda em uma loja de informática. Esse tipo de *software* é construído tendo como base as necessidades específicas de uma empresa. Isto exemplifica porque as empresas contratam profissionais da área de desenvolvimento de *software*; para que se mantenham competitivas, objetivando satisfazer seus clientes aumentando constantemente a qualidade de seus serviços.

1.1.1 O início de um Projeto de Aplicação

Quando uma empresa identifica a necessidade da construção de um *software*, geralmente ela contrata uma consultoria especializada para essa missão.

A consultoria, ao ser contratada, agenda uma reunião para que sejam discutidos os detalhes da necessidade de seu cliente.

Dentre os profissionais que a consultoria envia para esta reunião, há um profissional com o perfil técnico. Além de compreender as necessidades do cliente, este profissional possui conhecimentos para a geração de determinados documentos que serão utilizados na concepção de uma solução computacional para solucionar o problema.

Quando o conjunto de documentos necessários ao desenvolvimento de um *software* é criado, dizemos que temos o projeto de aplicação. É uma “tradução” da solução aprovada pelo cliente a ser implementada como uma aplicação computacional.

1.2 Construção de Software X Competências necessárias

Apenas recentemente temos o conceito de competência associado à geração de resultados para as empresas. A primeira referência que encontramos sobre essa associação foi estabelecida com LE BOTERF (1994) onde ele afirma que a competência está associada às realizações das pessoas; está associada ao que elas produzem e entregam. Neste enfoque, a competência é colocar em prática o que se sabe em um determinado contexto.

No trabalho a competência é tida como a reunião de conhecimento, habilidade, atitude e valor.

Conhecimento é a apropriação de um objeto pelo pensamento através da educação formal ou pela experiência prática. Corresponde ainda a uma série de informações que o indivíduo assimila e estrutura, de modo que lhe permite compreender o mundo, ou seja, é a dimensão do saber o que e porquê (MASLOW, 2000).

Habilidade é a capacidade de alguém realizar alguma atividade ou ter determinada aptidão. É a capacidade de desempenhar tarefas e atividades, de aplicar e fazer uso produtivo do conhecimento adquirido e utilizá-lo em uma ação com vista a atingir um propósito específico (CBO/MTE, 2002).

Atitude refere-se aos aspectos sociais e afetivos relacionados ao trabalho. A atitude determina o comportamento, pois está relacionada com percepção, personalidade, aprendizagem e motivação. É a ação cotidiana, a dimensão do querer-saber-fazer. (CBO/MTE, 2002).

Valor representa as crenças, motivação, caráter, costumes, consciência social, regras da vida em sociedade e da conduta entre os homens.

De acordo com a Classificação Brasileira de Ocupações (CBO/MTE) – documento normalizador do reconhecimento, da nomeação e da codificação dos títulos e conteúdos das ocupações do mercado de trabalho brasileiro - competência profissional é tida como um atributo inerente ao ser humano para o desempenho de uma ou várias funções quando se ocupa um cargo dentro da unidade organizacional de uma empresa.

No que se refere ao profissional de Tecnologia da Informação, mais especificamente na área de desenvolvimento de *software*, essas competências podem ser mapeadas como veremos a seguir.

1.2.1 Aspectos técnicos e perfil comportamental

Para ser um bom desenvolvedor de *software* não basta apenas ser um especialista numa determinada linguagem de programação. Um desenvolvedor de *software* deve possuir determinadas habilidades e competências tanto no que se refere a aspectos técnicos como no que envolve seu perfil comportamental.

1.2.1.1 Aspectos técnicos

BOA DOCUMENTAÇÃO: O desenvolvedor de *software* deve ser capaz de criar documentos para descrever corretamente o que faz o programa que ele escreveu. Com quais programas este programa está relacionado? Quais bancos de dados são acessados por este programa? Existem requisitos de sistema operacional, ou seja, algum componente deve estar previamente instalado no computador para que este programa funcione?

Além de manter bem documentado os seus programas, é importante que o desenvolvedor mantenha sempre estes documentos atualizados. Infelizmente é comum o desenvolvedor criar apenas a primeira versão da documentação e conforme surgem demandas para a atualização do programa, o desenvolvedor atualiza o programa e não dá a mesma importância para a atualização dos respectivos documentos.

QUALIDADE TÉCNICA DOS TRABALHOS: Muitas vezes um desenvolvedor se vê pressionado por seu gerente para cumprir os prazos estabelecidos por este referentes a entrega de programas.

Ceder à pressão codificando programas sem valorizar a qualidade do que será entregue é uma péssima prática.

Do ponto de vista da empresa contratante, isso tem boa probabilidade de significar aumento de custos, pois um programa construído que não atenda sua especificação¹, certamente precisará ser corrigido.

Se somarmos as horas utilizadas no desenvolvimento sem qualidade às horas utilizadas na correção deste programa, iremos verificar que a empresa teria um melhor custo/benefício se a primeira versão do programa tivesse sido entregue com a qualidade exigida, mesmo que isso demandasse algumas horas a mais para a entrega.

ANÁLISE: Um bom desenvolvedor deve saber analisar todos os documentos de um projeto de aplicação. Ao realizar a análise destes documentos deve compreender o que precisa ser feito; qual é o propósito do programa a ser codificado.

Atualmente a maioria das especificações de programas obedece ao que é conhecido por Programação Orientada a Objetos (POO).

¹ Especificação é o que o cliente espera que o programa execute.

No paradigma da POO, uma das coisas mais importantes que o bom desenvolvedor deve dominar é a UML – *Unified Modeling Language*.

A UML contém em sua definição muitos tipos de diagramas que servem para a especificação de programas e processos relacionados ao projeto de aplicação. Dentre os diagramas mais usados podemos destacar: Casos de Uso, Diagrama de Classe e Diagrama de Sequência.

BOA LÓGICA: Ter boa lógica de programação também é uma exigência para o profissional da área de desenvolvimento de *software*.

Nos bons cursos de programação, geralmente nos primeiros semestres o aluno tem uma grande carga horária relacionada a disciplinas envolvendo tópicos da matemática.

Sabe-se que os computadores trabalham utilizando o sistema binário de numeração. Dessa forma, pode-se dizer que na essência, todo o processamento do computador pode ser traduzido em operações matemáticas. Assim, o grau de conhecimento em lógica matemática determina a eficiência com a qual um programa é desenvolvido.

Paralelamente ao ensino da matemática temos o ensino de algoritmos, que são seqüências de instruções criadas em uma ordem lógica com o objetivo de solucionar um problema apresentado.

RESOLUÇÃO DE PROBLEMA: Há dez anos tínhamos na área de informática papéis muito bem definidos para os cargos e funções existentes. Por exemplo: O Analista de Sistemas tinha por função se reunir com o cliente, entender suas necessidades e gerar os documentos necessários (Casos de Uso, Diagramas de Classe, etc.) para que o Programador pudesse desempenhar sua função que era exclusivamente a de interpretar os documentos que descreviam o problema e então codificar os programas (solucionando o problema).

Nos últimos anos as empresas buscam reduzir cada vez mais os seus custos em todos os setores e o departamento de tecnologia não foge a regra.

Na década de 90 tínhamos as carreiras de: Programador, Analista de Sistemas e Gerente de Informática. O Gerente de Informática era quem se reunia previamente com o cliente para tratar de aspectos burocráticos. Ele firmava o Contrato de Prestação de Serviços, documento que descrevia quais as obrigações em termos de solução tecnológica, o valor cobrado pelo projeto, a forma de pagamento, o prazo de entrega definido para cada etapa do projeto, etc.

A globalização ocasionou uma evolução no mercado de TI de tal magnitude que as empresas brasileiras deste segmento viram-se forçadas a investir em aprimoramentos e sobretudo a reduzir custos para manterem-se competitivas. Isso impactou diretamente a carreira de muitos profissionais de Tecnologia da Informação. As carreiras de Analista de Sistemas e de Programador fundiram-se dando origem ao cargo denominado Analista Programador.

Em resumo, as empresas contratavam um profissional pagando-lhe o mesmo salário de um Programador ou de um Analista de Sistemas, e exigia-lhe que executasse a soma das atribuições destes dois cargos.

Além do aumento das funções, tivemos a feroz evolução dos sistemas operacionais, hardware de computadores, dos sistemas gerenciadores de bancos de dados e das próprias linguagens de programação. Tudo isso fez com que o profissional de desenvolvimento de *software* tivesse que estudar cada vez mais para manter sua empregabilidade.

Nos últimos anos a globalização transformou o profissional de TI em um “recurso mundial”. Se antigamente um dos pré-requisitos de uma vaga era possuir “noções de inglês”, hoje é mandatório ser fluente neste idioma se o desejo é ter uma boa oportunidade no mercado de trabalho.

Empresas multinacionais invadiram nosso País, comprando empresas menores e oferecendo emprego somente aos melhores profissionais do mercado.

Uma competência comumente exigida nas vagas disponibilizadas por estas empresas é que o profissional seja *problem-solving*, ou seja, tenha a capacidade de resolver problemas.

Ter a capacidade de resolver problemas não é algo trivial, principalmente quando estamos falando de grandes organizações compostas por milhares de funcionários e um número expressivo de clientes.

O fato é que se antigamente a capacidade de resolver problemas restringia-se às exigências de um cargo gerencial, hoje isso é exigência para um número muito maior de funcionários, inclusive para o desenvolvedor de *software*.

Tratamos aqui de um assunto complexo, pois para resolver um problema é necessário conhecer muitas variáveis sobre essa ocorrência e também é necessário tomar a decisão correta no menor prazo possível.

Para explicar sobre isso citaremos como exemplo a complexidade do funcionamento de uma Instituição Financeira, um banco.

Imagine que um desenvolvedor construiu uma aplicação de saque. Essa aplicação é executada diariamente por todos os caixas existentes em todas as agências deste banco. É também executada quando um cliente realiza um saque utilizando um Terminal de Auto-atendimento (caixa eletrônico).

Suponha que este desenvolvedor escreveu uma nova versão da aplicação de saque que está preparada para substituir a versão atual na próxima segunda-feira. Tudo foi exaustivamente testado, nada pode dar errado.

O tempo vai passando... sexta-feira, sábado, domingo e finalmente chega o momento de atualizar a aplicação de saque pela nova versão.

O desenvolvedor acompanha tudo de dedos cruzados... é o seu emprego que está em jogo.

No horário agendado o departamento técnico responsável pela atualização de aplicações informa à todos os envolvidos neste processo o resultado da operação:

“- Atualização completa. O procedimento foi realizado com sucesso!”

“- Ufa!” pensa o desenvolvedor. “- Ainda bem que ocorreu tudo conforme o esperado”.

Em menos de um minuto após o procedimento de atualização começa um “corre-corre” para todo lado. Os telefones começam a tocar e não param mais. São ligações desesperadas de funcionários das agências de todo o País afirmando que os clientes não estão conseguindo sacar dinheiro de suas contas.

O telefone toca na mesa do desenvolvedor. Ele atende o telefone e já imagina o que está por vir...

“- Carlos. Bom dia!”. Do outro lado da linha era o gerente que acabara de ligar para o desenvolvedor responsável pela paralisação de todas as agências do banco...

“- Bom dia Carlos. Aqui é o Joseclin! As agências estão paradas. Em quanto tempo você acha que resolve isso?”

O desenvolvedor respira fundo e reflete sobre o problema... Ele se interroga: Por que somente as agências estão reclamando de problema no saque se os caixas eletrônicos também utilizam essa funcionalidade? Normalmente são realizados muitos saques por minuto nos caixas eletrônicos (...). Para certificar-se de que o saque estivesse funcionando nos caixas eletrônicos bastaria ligar para a equipe responsável e fazer este questionamento, assim Carlos saberia se o problema está somente na parte do programa do programa executado pelas agências.

O fato é que Carlos não dispunha de tempo para uma análise detalhada agora. Vejamos o cenário: o funcionamento de todas as agências está parado; o Joseclin está na linha aguardando uma resposta; qual a melhor decisão a ser tomada para solucionar o problema?

Eis que Carlos decide o que fazer e rapidamente responde ao seu gerente:

“ - 5 minutos. Em cinco minutos a situação estará normalizada”. E então o Joseclin responde:

“- Tudo bem Carlos. Te ligarei novamente em cinco minutos...”

Sem soltar o telefone que estava na mão esquerda, com a mão direita Carlos acessa a agenda de contatos de seu celular e faz uma busca pelo nome do seu colega no departamento técnico de atualização de aplicações.

De posse do número de telefone que procurava, Carlos liga para o colega de trabalho:

“- Jackson?! É o Carlos. Tudo bem?”

“- Sim Carlos! Tudo bem e você?”

“- Tudo bem. Preciso de um favor... e é urgente”.

“- Pode falar meu amigo. Em que posso ajudá-lo?”

“- Sabe a nova versão da aplicação de saque que foi atualizada há poucos minutos? Preciso que você desfça o procedimento, de maneira que tenhamos a versão anterior em funcionamento”.

“- Entendi. Fique na linha que farei isso agora mesmo...”.

Alguns segundos mais tarde...

“- Pronto Carlos. Solicitação atendida!”.

“- Muito obrigado Jackson”.

“- Por nada...”.

Após voltar a versão da aplicação para a que estava funcionando, Carlos liga para algumas agências objetivando certificar-se de que tudo voltou ao normal. Sim, felizmente agora os clientes conseguiam sacar dinheiro.

Carlos também recebeu o telefonema do Joseclin parabenizando-o por ter conseguido normalizar a situação no tempo esperado.

A partir desse exemplo sobre a atitude de um desenvolvedor que trabalha em um determinado banco, retomaremos a explanação sobre a complexa competência de resolver problemas.

O bom desenvolvedor, ao analisar um problema que surge, deve saber distinguir o que é a causa e o que é sintoma.

Geralmente quando alguém apresenta um problema ao desenvolvedor, tratam de informações do sintoma. Isso significa que na maioria das vezes não é aquela aplicação ou funcionalidade da qual o cliente está reclamando que contém o erro, mas talvez outra aplicação que se refletiu na aplicação que o cliente utiliza, causando o sintoma.

No exemplo da funcionalidade de saque, temos o seguinte:

A aplicação que o caixa da agência utiliza em seu computador não contém as aplicações de saque, depósito ou pagamento de boletos bancários. Essa aplicação apenas faz uma chamada para outras aplicações que estão fisicamente instaladas no computador central do banco.

Então, quando ligaram para o desenvolvedor Carlos dizendo que havia um problema na agência pois o saque parou de funcionar, na verdade este tratava-se apenas de um sintoma. O problema residia no computador central do banco, local onde o Carlos atualizou a versão da aplicação de saque.

Poderíamos afirmar que a decisão “correta” para solucionar este problema seria Carlos analisar a nova versão da aplicação de saque de modo a identificar o que estava incorreto, corrigir e então realizar a atualização da versão desta aplicação.

Mas e os sintomas? Neste caso os sintomas consistiam na paralisação de um dos negócios mais importantes do banco: o funcionamento das agências.

Verifica-se então que corrigir primeiro o problema para evitar a ocorrência de novos sintomas não pode ser tido como uma regra geral. Em alguns casos é preciso tomar uma decisão imediata que vai apenas solucionar um sintoma para, posteriormente, tomar atitudes para solucionar o problema.

Para resumir a competência de Resolução de Problemas poderíamos ter a seguinte lista:

- O foco em definir um problema é distinguir causa e sintoma;
- Geralmente o que recebemos como descrição de um problema é a descrição de um sintoma;
- A Tomada de Decisão inclui analisar o problema a fim de identificar as soluções viáveis, optar por uma dessas soluções e implementá-la; e
- A decisão “correta” como vimos no exemplo do desenvolvedor Carlos, pode não ser a “melhor” decisão dependendo da circunstância atual.

1.2.1.2 O Perfil Comportamental

COMUNICAÇÃO: Independente do idioma que o desenvolvedor usa no seu dia-a-dia é imprescindível que este consiga se expressar de forma clara e objetiva, tanto oralmente como na escrita.

PRÓ-ATIVIDADE: Se um profissional identifica que algo deve ser feito para o bom andamento do trabalho, não deve hesitar.

No competitivo mercado atual não cabe mais a postura de dizer: “- Isso aqui eu não farei porque não me foi solicitado”.

Equipes vencedoras são formadas por profissionais que ajudam-se mutuamente. Se a equipe fracassa, todos fracassam.

INTELIGÊNCIA EMOCIONAL: Com o crescente aumento das exigências no trabalho, as circunstâncias as quais o profissional é submetido diariamente exigem um consciente controle de suas emoções.

Mesmo com a responsabilidade de administrar e entregar diversos trabalhos, muitas vezes simultâneos, espera-se que o desenvolvedor de *software* saiba lidar de forma harmoniosa com todas as variáveis desta complexa equação. Seja no seu relacionamento com a equipe, com seus superiores ou com clientes, o profissional deve ser atencioso, cordial e profissional.

COMPROMETIMENTO: Uma empresa espera que o desenvolvedor de *software* tenha um vínculo organizacional.

Quando um profissional se identifica com a organização e com os objetivos dela, há o desejo de manter-se como membro. Cria-se uma relação de lealdade, de desejo de contribuir e há o sentimento de orgulho em permanecer na organização. Isso é conhecido como comprometimento afetivo.

Existem empresas que conseguem o comprometimento de suas equipes através de recompensas em termos de salário, status, etc.

Há ainda as que mantêm o comprometimento através de normas internas de comportamento.

Mas como neste trabalho se está abordando o perfil comportamental do desenvolvedor de *software*, não é relevante como a empresa coloca em prática o conceito de comprometimento. O importante é que o desenvolvedor de *software* saiba o que uma empresa espera dele quando diz que ele deve ser comprometido.

Basicamente, o desenvolvedor de *software* precisa saber a resposta da pergunta: - O que é comprometimento?

Pode-se dizer que comprometimento é tudo o que uma pessoa faz a mais, sem que alguém tenha solicitado.

Hoje há muitos funcionários nas empresas que realizam estritamente o necessário, ou seja, fazem o que está nas atribuições de seu cargo, o restante não diz respeito a eles. Devido a essa postura alguns funcionários chegam até a desrespeitar os clientes desta empresa.

Quantas vezes fomos a alguma empresa para resolver algum tipo de problema e a pessoa nos foi passando para outra, dizendo que a pessoa responsável não era ela? Isso quando a ligação não cai misteriosamente.

Você já tentou cancelar uma linha telefônica e passou por um cenário assim? Qualquer semelhança é mera coincidência (...).

Se nos colocarmos na visão do presidente de uma empresa, qual seria a nossa visão com relação aos nossos funcionários? Nosso desejo seria que sempre que um cliente apareça na empresa e fale com um funcionário portando um crachá, este cliente reconheça-o como um representante da corporação, independente de ser a copeira, o atendente ou o diretor. O compromisso em resolver o problema das pessoas, independente da sua responsabilidade, é um grande exemplo de comprometimento.

Por outro lado, um exemplo simples de descomprometimento é um funcionário passar por um pedaço de papel jogado no corredor da empresa e não ter o compromisso de pegar o papel e jogá-lo no cesto de lixo. Ao invés disso o funcionário descomprometido pensa: “eu não sou funcionário da limpeza e isso não é meu problema”.

Tudo o que fazemos a mais, sem o interesse de obter alguma vantagem, é o que faz a diferença hoje no mundo corporativo; e é o que cada vez mais as empresas buscam.

1.3 As 5 competências analisadas com o projeto LOJA .NET

O projeto LOJA .NET proposto aos alunos nos permitiu analisar cinco importantes competências para o desenvolvedor de *software*. São elas:

- 1) Saber utilizar o conceito de Programação Orientada a Objetos. Aqui verificamos se o aluno consegue criar e utilizar corretamente classes, métodos e propriedades.
- 2) Saber utilizar o conceito de Desenvolvimento em Camadas. Verificamos se o aluno, ao construir as classes em uma camada, faz as referências necessárias visualizando a comunicação com a próxima camada (neste caso, com a camada superior). Em outras palavras, o aluno deve prever a comunicação eficaz entre todas as camadas.
- 3) Saber configurar e gerenciar informações em um banco de dados. Nessa competência se verifica se o aluno consegue criar tabelas em banco de dados utilizando o conceito de integridade referencial. Também verificamos a criação e o uso de *stored procedures*.
- 4) Saber construir aplicações utilizando o método de Desenvolvimento em Camadas. Aqui verificamos se o aluno consegue construir aplicações fazendo uso eficaz das camadas construídas previamente. Essa competência é utilizada independente da interface com o usuário a ser utilizada (Aplicação para Windows, Websites, Aplicação para iPhone, etc.).
- 5) Saber usar na prática tópicos sobre Segurança da Informação. Uma das exigências de empresas de qualquer porte é o desejo de garantir o sigilo e a segurança de suas informações. Nessa competência verificamos se o aluno consegue proteger o sistema que construiu contra tentativas de pessoas mal intencionadas que roubam, deletam ou divulgam informações de sistemas invadidos, os chamados *crackers*.

1.3.1 Qual o diferencial do projeto LOJA .NET?

O projeto LOJA .NET é um material didático que utiliza a metodologia de desenvolvimento em camadas e foi criado com o objetivo de analisar junto aos alunos as 5 importantes competências do profissional de desenvolvimento de *software* descritas anteriormente. Mas o que diferencia o uso do método de desenvolvimento em camadas do método de ensino tradicional dos alunos envolvidos nesta pesquisa?

Para nos auxiliar nessa questão, iremos recorrer aos conceitos de complexidade e transdisciplinaridade.

Entendemos que o diálogo inter-disciplinas vem se impondo nos últimos anos como uma necessidade crescente. Dentre os principais motivos que justificam isso temos a transformação do próprio conceito de disciplina científica e do que se entende por cientificidade. Por isso, destacaremos alguns aspectos dessa transformação e o seu impacto sobre o desenvolvimento científico atual.

1.3.1.1 Complexidade

Complexus (ou apenas *plexus*) é o vocábulo que, em latim, significa entrelaçado, tecido em conjunto e constitui a origem etimológica da palavra *complexo*. Hoje esse termo é usado com uma certa frequência e, muitas vezes, a pessoa que utiliza o termo fica isenta de dar maiores explicações sobre o assunto tratado. É comum ouvirmos: esta disciplina é complexa, a situação é complexa, o problema é complexo. Alguns dicionários definem a complexidade como propriedade dos processos caóticos; informação, diversidade, multiplicidade de conexões; alta conectividade.

Daí nos fica a idéia de que complexidade significa caos, desordem, dificuldades, algo complicado. A complexidade e a complicação não se reduzem uma à outra. Para Morin (2001), a complicação é um dos constituintes da complexidade. Esta idéia do que é complexidade está muito mais espalhada no vocabulário corrente do que no vocabulário científico. De uma forma ou de outra a palavra complexidade sempre foi usada, mas só recentemente se fala em teoria da complexidade.

A complexidade do mundo em que vivemos transparece no mundo das artes, no mundo da política, no mundo da ciência, no mundo do comércio, etc. Mas o fato é que só existe um mundo. Todos os mundos citados se entrelaçam num mesmo espaço-tempo em que vivemos.

O mundo de hoje é marcado pela evolução científico-tecnológica, e nesse cenário a complexidade torna-se cada vez mais um conceito fundamental.

No projeto LOJA .NET, pode-se considerar a análise da “complexidade” no que se refere à quantidade de recursos necessários para a construção de cada parte deste *software*. Em um nível mais básico temos o número de passos a serem executados em cada classe e o volume de memória utilizado pelos objetos instanciados a partir destas classes. Assim é possível, por exemplo, analisar antecipadamente se os computadores disponíveis irão atender os requisitos para o pleno funcionamento do projeto.

Segundo o pensamento complexo descrito pelo pensador Morin, será que era possível prever se o projeto LOJA .NET auxiliaria o grupo de alunos na aquisição de competências em desenvolvimento de *software*?

Este pensador afirma que a complexidade necessita de uma estratégia, sabe que não se pode programar a descoberta, o conhecimento, nem a ação.

No entanto, Morin (2001) diz que para compreender o problema da complexidade, é preciso saber primeiro que há um paradigma de simplicidade, que é utilizado com muita frequência. Neste paradigma prevalece a ordem, a lógica, a lei e os princípios, expulsando o caos e a desordem. “O princípio da simplicidade quer separar o que está ligado (disjunção), quer unificar o que está disperso (redução)” (MORIN, 2001).

O paradigma da simplificação não permite pensar a unidade na diversidade ou a diversidade na unidade. Este paradigma deixa perder a riqueza da diversidade, da dinâmica, da desordem. Ao avaliarmos o curso técnico do CEFET dentro de suas edificações, não consideramos suas implicações sociais e econômicas. De lá conhecemos apenas uma parte do real, perdendo assim o conhecimento sobre a imensidão das demais partes desta realidade.

“Vai estudar-se o cérebro como órgão biológico e vai estudar-se o espírito, *the mind*, como função ou realidade psicológica. Esquece-se que um não existe

sem o outro; ou melhor, que um é simultaneamente o outro, embora sejam tratados por termos e conceitos diferentes” (MORIN, 2001).

Verificamos então que Morin aceita a complexidade como princípio de pensamento que considera o mundo e não o princípio revelador da essência do mundo. O pensamento complexo acabou com nossa ilusão de segurança. Verificamos que o pensamento humano é cheio de dúvidas, de incertezas e o erro é o seu principal companheiro, é o que nos ensina a viver. “Estamos condenados ao pensamento inseguro, a um pensamento crivado de buracos, um pensamento que não tem nenhum fundamento absoluto de certeza. Mas somos capazes de pensar nestas condições dramáticas” (MORIN, 2001).

1.3.1.2 Transdisciplinaridade

Em que o método de desenvolvimento em camadas apresentado no projeto LOJA .NET se difere do método tradicional de ensino no Curso Técnico em Informática sob análise?

Para analisar essa questão faremos uma reflexão sobre o que significa transdisciplinaridade para então posteriormente verificarmos se o projeto LOJA .NET caracteriza-se como sendo transdisciplinar.

Necessariamente iremos percorrer os conceitos de disciplinaridade, multidisciplinaridade e interdisciplinaridade para então refletirmos sobre o significado da transdisciplinaridade.

A ciência se constrói a partir de teorias e de verificações experimentais dessas teorias. Podemos encontrar em relatórios científicos uma permanente interação entre teoria e prática. Isto denota a importância da prova empírica para validar o novo saber científico (CARAÇA, 2001).

Partindo dessa idéia geral de ciência como forma de construção do conhecimento, ao olharmos para as diversas dimensões do ser humano, percebemos a importância das diferentes áreas de conhecimento. Em cada área do conhecimento encontramos as especificidades de cada disciplina. Nesse contexto podemos abordar os conceitos de disciplinaridade, interdisciplinaridade, multidisciplinaridade e transdisciplinaridade.

Há algumas décadas era um sonho abordar o conceito de transdisciplinaridade. Atualmente isso é tema de debates e conferências sobre educação, comprovando que se trata de uma realidade.

Disciplinaridade

Segundo NEST (2000), no senso comum a disciplina inclui três entendimentos fundamentais: o primeiro, traduzindo o sentido comum de um campo específico ou particular de conhecimento; o segundo, revelado na postura de obediência e submissão do homem a regras de conduta e, por último, envolvendo regras e códigos comportamentais, específicos e próprios de determinados grupos sociais (militares, religiosos, filosóficos, dentre outros).

Já na visão de HECKHAUSEN (1972), o que distingue uma disciplina de outra são os fatores: 1) domínio material ou objeto de estudo; 2) o conjunto possível de fenômenos observáveis; 3) o nível de integração teórica; 4) os métodos; 5) os instrumentos de análise; 6) as aplicações práticas; e, 7) as contingências históricas.

Multidisciplinaridade

É um conjunto de disciplinas justapostas sem nenhuma cooperação entre elas (JANTSCH, 1972).

No entanto, a multidisciplinaridade ocorre quando “a solução de um problema torna necessário obter informação de duas ou mais ciências ou setores do conhecimento sem que as disciplinas envolvidas no processo sejam elas mesmas modificadas ou enriquecidas”¹

Interdisciplinaridade

O prefixo *inter* originado do latim significa posição intermédia; reciprocidade. Assim, o termo interdisciplinaridade significa o que é relativo ou o que se efetua entre as disciplinas.

¹ Piaget, Jean. The epistemology of interdisciplinary relationships, citado por Chaves (1998)

Para NEST (2000), a interdisciplinaridade revela o caminho entre as disciplinas que é percorrido nas abordagens do conhecimento, em que o observador recebe a contribuição metodológica e informativa das disciplinas individualizadas, sem que estas sejam afetadas ou dirigidas pelos resultados. Outros pensadores afirmam que a interdisciplinaridade é utilizada para designar “o nível em que a interação entre várias disciplinas ou setores heterogêneos de uma mesma ciência conduz a interações reais, a uma certa reciprocidade no intercâmbio, levando a um enriquecimento mútuo” (CHAVES, 1998).

Transdisciplinaridade

Bassarab NICOLESCU (1999), afirma que foi Jean Piaget o primeiro a usar a palavra transdisciplinar como expressão de uma nova abordagem do conhecimento, em 1967: A transdisciplinaridade, como o prefixo "trans" indica, diz respeito àquilo que está ao mesmo tempo entre as disciplinas, através das diferentes disciplinas e além de qualquer disciplina. Seu objetivo é a compreensão do mundo presente, para o qual um dos imperativos é a unidade do conhecimento... (NEST, 2000).

Segundo CHAVES (1998) o conceito de transdisciplinaridade não envolve somente as interações ou reciprocidade entre projetos especializados de pesquisa, mas a colocação dessas relações dentro de um sistema total, sem quaisquer limites rígidos entre as disciplinas.

A partir dessa definição verificamos que a transdisciplinaridade não é uma nova disciplina. No entanto, se alimenta da pesquisa disciplinar que, por sua vez, é clareada de uma nova maneira. As pesquisas disciplinares e transdisciplinares se complementam.

A disciplinaridade, a multidisciplinaridade, a interdisciplinaridade e a transdisciplinaridade são as quatro flechas de um único arco: o do conhecimento (NICOLESCU, 1999). A correta compreensão destes quatro âmbitos do conhecimento permite uma correta articulação nos quatro pilares da educação:

- Aprender a conhecer: ser capaz de estabelecer pontes entre os diferentes saberes. Entre estes saberes e sua significação para o dia-a-dia e ainda entre estes saberes e as capacidades interiores de cada um. Este procedimento

transdisciplinar é indispensável ao procedimento disciplinar, pois irá conduzir à formação de um ser constantemente atento, capaz de se adaptar aos constantes desafios impostos pela vida profissional. Este indivíduo será dotado de uma flexibilidade permanente, orientado para a atualização de suas potencialidades interiores.

- Aprender a fazer: dentro do espírito da transdisciplinaridade significa escolher uma profissão, adquirir os conhecimentos e técnicas e ela associados e exercer essa profissão com criatividade. Fazer significa também inovar, colocar na prática do dia-a-dia as potencialidades criativas.

- Aprender a conviver: acatar as normas da sociedade em que se está inserido. Tais normas devem ser intimamente aceitas pelo indivíduo e não apenas obedecidas como uma lei imposta exteriormente.

- Aprender a ser: cada um deve descobrir a harmonia ou a desarmonia entre a vida interior e social. Sondar os fundamentos de suas crenças, para descobrir o que existe de subjacente.

Por outro lado, a transdisciplinaridade se interessa pela dinâmica gerada pela ação de vários níveis de realidade ao mesmo tempo... (NEST, 2000). Assim, a transdisciplinaridade exige uma mudança do sistema de referência e articula-se em três pontos:

- a) Vários níveis de realidade;
- b) Não mais esperar encontrar a solução de um problema como um simples "verdadeiro" ou "falso" da lógica binária;
- c) A complexidade de um problema reconhece a impossibilidade da decomposição deste problema em partes simples.

Nesse contexto a transdisciplinaridade surge na pós-modernidade como uma nova maneira de enquadrar a ciência. Sem prejudicar a investigação científica disciplinar, mas enriquecendo-a pelo recurso da interdisciplinaridade. Assim é possível partirmos para novos temas que se apresentam com características transversais, ocupando espaços reservados a diversas disciplinas sem que haja a missão de preencher um espaço vazio. Não se trata de criar novas disciplinas, porque não se trata de preencher espaços vazios. Trata-se de individualizar temas que só existem pela junção de várias disciplinas e que só se pode analisá-los de forma integrada.

No projeto LOJA .NET verificamos por exemplo, que a competência de saber configurar e gerenciar informações em banco de dados engloba a análise de três gráficos que representaram o desempenho dos alunos em: 1) compreender e usar o conceito de Integridade Referencial em Banco de Dados; 2) Criar tabelas em banco de dados e 3) Criar e utilizar *stored procedures*.

Já a competência de saber como funciona a construção de aplicações utilizando o modelo de desenvolvimento em camadas engloba a análise de duas disciplinas: 1) Funcionamento da construção de aplicações para MS-Windows e 2) Funcionamento da construção de aplicações para a WEB.

CAPÍTULO 2

Desenvolvimento em Camadas

Desenvolvimento em Camadas é um método muito utilizado para a construção de *software*. Antes da definição deste método se faz necessário um resgate histórico do surgimento da Internet para uma melhor compreensão da arquitetura em camadas.

2.1 É possível definir Internet?

O surgimento da Internet foi possível por meio de padrões de redes para que estas pudessem se comunicar. Se nos restringirmos aos aspectos técnicos, a Internet poderia ser definida como uma rede entre computadores, de abrangência mundial, que adota como padrão determinados protocolos de comunicação.

Atualmente muitas pessoas questionam se é suficiente definir a Internet como sendo apenas a interconexão entre computadores. O próprio Vint Cerf – um dos criadores da Internet – diz que:

“a Internet não se constitui somente de números e protocolos, mas de pessoas, conteúdo e recursos”.

A organização chamada de Cúpula Mundial da Sociedade da Informação também considera a Internet num aspecto mais amplo, conforme o exposto na reunião ocorrida em TUNIS (2005):

“Nós reconhecemos que a Internet, elemento central de infra-estrutura da Sociedade da Informação, evoluiu da facilidade criadas para a pesquisa acadêmica para uma facilidade global disponível ao público” (tradução livre do autor).

Assim temos a indicação de que é correto considerar a Internet tanto em seus aspectos técnicos como nos seus aspectos sociais, jurídicos, econômicos e culturais.

2.2 Os padrões na Internet

Apesar de um consenso sobre a importância de considerar diversos aspectos da Internet como visto no item 2.1, utilizaremos outra definição do que vem a ser Internet para iniciar uma discussão sobre a importância dos padrões desde a sua criação até os dias atuais.

A Internet pode ser definida como uma rede entre computadores que adota protocolos-padrão, principalmente o TCP-IP³, para a transmissão de dados, de maneira a permitir:

- comunicação simultânea entre usuários;
- troca de mensagens eletrônicas;
- acesso a portais eletrônicos;
- criação de comunidades virtuais;
- transmissão de dados por meio de arquivos em diversos formatos (texto, som, imagem, vídeo, etc.);
- comércio eletrônico;
- jogos online;
- difusão de conhecimento, e muitas outras atividades.

Para possibilitar a transmissão desses dados, a Internet está estruturada em basicamente três diferentes camadas.

³ TCP-IP. É um conjunto de protocolos de comunicação entre computadores em rede.

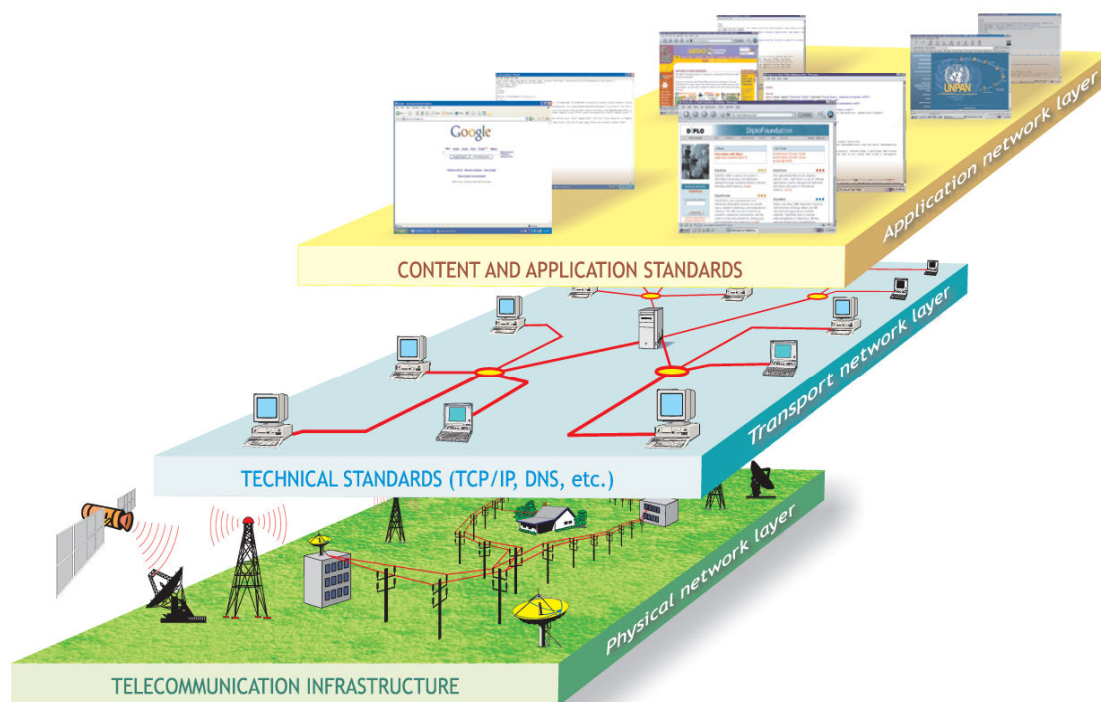


Figura 1 – Representação das camadas da Internet

Fonte: DIPLO Internet Governance and Policy

<<http://www.diplomacy.edu/ig/Resources/display.asp?Topic=DiploImages>>

Analisando a figura de baixo para cima, temos que a primeira camada se chama “Infra-estrutura de Telecomunicações”. Ela é uma representação de que a comunicação entre os computadores da Internet depende de uma estrutura de cabeamento (linhas telefônicas, fibra ótica) ou de ondas (de rádio, satélite). Em resumo, temos que é necessário o estabelecimento de um caminho físico, visível ou não, no “mundo real”.

A segunda camada diz respeito aos padrões tecnológicos. É a linguagem comum que a Internet adotou como padrão para que fosse possível a comunicação entre os computadores. Por serem de fundamental importância para o funcionamento da rede, esses padrões devem ser abertos⁴.

A última camada é onde estão os conteúdos e os padrões de aplicação. Aqui residem muitos conflitos com relação à universalização da rede.

Quando se fala tanto nas camadas de infra-estrutura como na de padrões técnicos, é possível desenhar uma visão universal que possa ser seguida e implementada por todos os países. Entretanto, o assunto torna-se mais complexo quando fala-se de conteúdos e padrões de aplicação. (ver item 2.2.1) É na terceira camada que está a *Web*, que pode ser definida como a parte da Internet acessada através dos navegadores, ou *browsers*. Criada por Tim Berners-Lee, motivado em facilitar a organização e troca de informações entre pesquisadores, a *Web* foi a grande responsável pela popularização da Internet. O resultado disso é que hoje a *Web* é popularmente confundida com a própria Internet.

Com o objetivo de assegurar a compatibilidade entre as diferentes novas tecnologias incorporadas à Web, em 1994 foi fundado o W3C* (*World Wide Web Consortium*). O W3C é administrado por:

1) Laboratório de Ciência da Computação e Inteligência Artificial (CSAIL) do MIT, nos EUA; 2) Consórcio Europeu de Pesquisa para a Informática e Matemática (ERCIM), na França; e 3) Universidade de Keio, no Japão.

⁴ Um padrão aberto ou protocolo não-proprietário é um protocolo de comunicação que não pertence à nenhuma empresa, podendo ser facilmente disseminado pela comunidade técnico-acadêmica.

*Nota: O W3C possui um escritório brasileiro, fundado em 2007. O escritório tem como objetivo disseminar a cultura e adoção de padrões para o desenvolvimento Web; Organizar atividades para promover os padrões e ferramentas W3C; Facilitar a participação da comunidade brasileira na definição dos padrões Web; Promover políticas e procedimentos para a regulamentação e uso da Web no país e recomendar padrões técnicos para o desenvolvimento da Internet no país.

2.2.1 Padrões no Desenvolvimento de Software

Existem diferentes perfis de usuários com relação à utilização da Internet.

Cada perfil de usuário Internet gostaria que fossem preservados e melhorados os benefícios que hoje são percebidos em suas atividades cotidianas. Os perfis são:

1. **Usuário leigo:** É aquele que somente utiliza a Internet para atividades cotidianas, profissionais ou para lazer;
2. **Usuário do governo:** Podem ser os poderes Legislativo e Judiciário enquanto normatizadores e fiscalizadores do cumprimento da Lei em diferentes ambientes, e pode ser também o poder Executivo no que se refere ao desenvolvimento de políticas públicas voltadas à Internet.
3. **Usuário técnico:** O usuário técnico domina a linguagem, conhece a concepção física e os mecanismos pelos quais se estabelece a conexão entre os computadores e demais dispositivos da rede.

É o usuário técnico que busca a aquisição de habilidades e competências necessárias ao Desenvolvimento de *Software*.

O Desenvolvimento de *Software* compreende diversas disciplinas. Vamos analisar, por exemplo, a disciplina de Banco de Dados.

Apesar de haver uma vasta teoria referente aos sistemas de banco de dados, cada empresa fabricante de banco de dados implementa as orientações da teoria a sua maneira.

Vejam alguns exemplos da variedade de empresas e produtos quando se fala em Sistemas de Bancos de Dados: a empresa Microsoft comercializa o banco de dados chamado *Microsoft SQL Server*; a empresa Oracle comercializa o banco de dados que possui o mesmo nome da empresa; já a empresa IBM comercializa um banco de dados chamado DB2. Além disso, todas essas empresas criam diferentes versões de seus produtos, que são atualizadas frequentemente para acompanhar a rápida evolução tecnológica.

Devido ao fato destes fabricantes não adotarem um padrão na criação de seus produtos, o profissional de Desenvolvimento de *Software* encontra especificidades no momento de escrever os programas que irão se conectar e

acessar os dados existentes nestes diferentes sistemas gerenciadores de banco de dados.

Assim como na Internet se faz necessário um mínimo de padrões para permitir a comunicação e interoperabilidade da rede, temos na camada de “conteúdos e padrões de aplicação” da Internet a necessidade da utilização de padrões na criação de *software* que serão acessados por milhares de usuários.

A utilização de padrões na criação de *software* traz inúmeros benefícios. Dentre eles pode-se destacar:

- Os programas desenvolvidos segundo padrões de desenvolvimento podem ser reutilizados em outro *software*, aumentando assim a produtividade no ciclo de desenvolvimento e facilitando futuras manutenções. Vejamos um exemplo: um *website* que comercializa os produtos de uma empresa possui como infra-estrutura de banco de dados uma base Microsoft SQL Server. Num determinado momento, o diretor de Tecnologia resolve migrar a infra-estrutura para uma base Oracle. Se o desenvolvimento seguiu padrões para evitar o acoplamento do *website* ao banco de dados, ou seja, não há a dependência da programação das páginas com relação a um fabricante específico, bastaria substituir o servidor de banco de dados e não seria preciso alterar toda programação já existente para as páginas deste *website*. Por outro lado, se o *website* não foi construído com a idéia de evitar acoplamento, seria necessário um grande esforço para permitir o funcionamento dos serviços disponibilizados pelo *website* quando da migração para outro sistema de banco de dados. O programador precisaria detectar e alterar todas as páginas do *website* que fazem acesso ao banco de dados.
- Se todos os desenvolvedores utilizam o mesmo padrão de desenvolvimento, caso um desenvolvedor assuma um projeto em andamento poderá dar continuidade aos trabalhos imediatamente, pois conhece a metodologia de trabalho utilizada. Se cada desenvolvedor trabalhar como quiser, haverá um gasto enorme

de tempo neste exemplo até que o novo profissional compreenda os documentos relacionados ao projeto em questão. Há casos em que a documentação é insuficiente para mapear os processos do sistema ou simplesmente inexistente. Este é o pior dos casos, pois será preciso analisar todo o código-fonte para, a partir daí, gerar fluxogramas mapeando os processos de maneira que seja possível a manutenção ou até mesmo o desenvolvimento de um sistema completamente novo. Isso é o que chamamos de engenharia reversa.

2.3 O Desenvolvimento de Aplicações em Camadas

Como vimos na seção anterior (Figura 1), a Internet foi estruturada basicamente em três camadas. Existe um consenso na comunidade de desenvolvimento de *software* sobre uma série de conceitos que devem ser seguidos pelos desenvolvedores para que haja um padrão de qualidade que atenda as exigências do mercado.

Um dos conceitos mais difundidos atualmente é conhecido como Modelo de Desenvolvimento em Camadas, também chamado de *N-Tier Development Model* ou N Camadas.

Semelhante ao design de camadas da Internet, o Modelo de Desenvolvimento em Camadas utilizado em desenvolvimento de *software* também é composto por três camadas conforme ilustração a seguir.

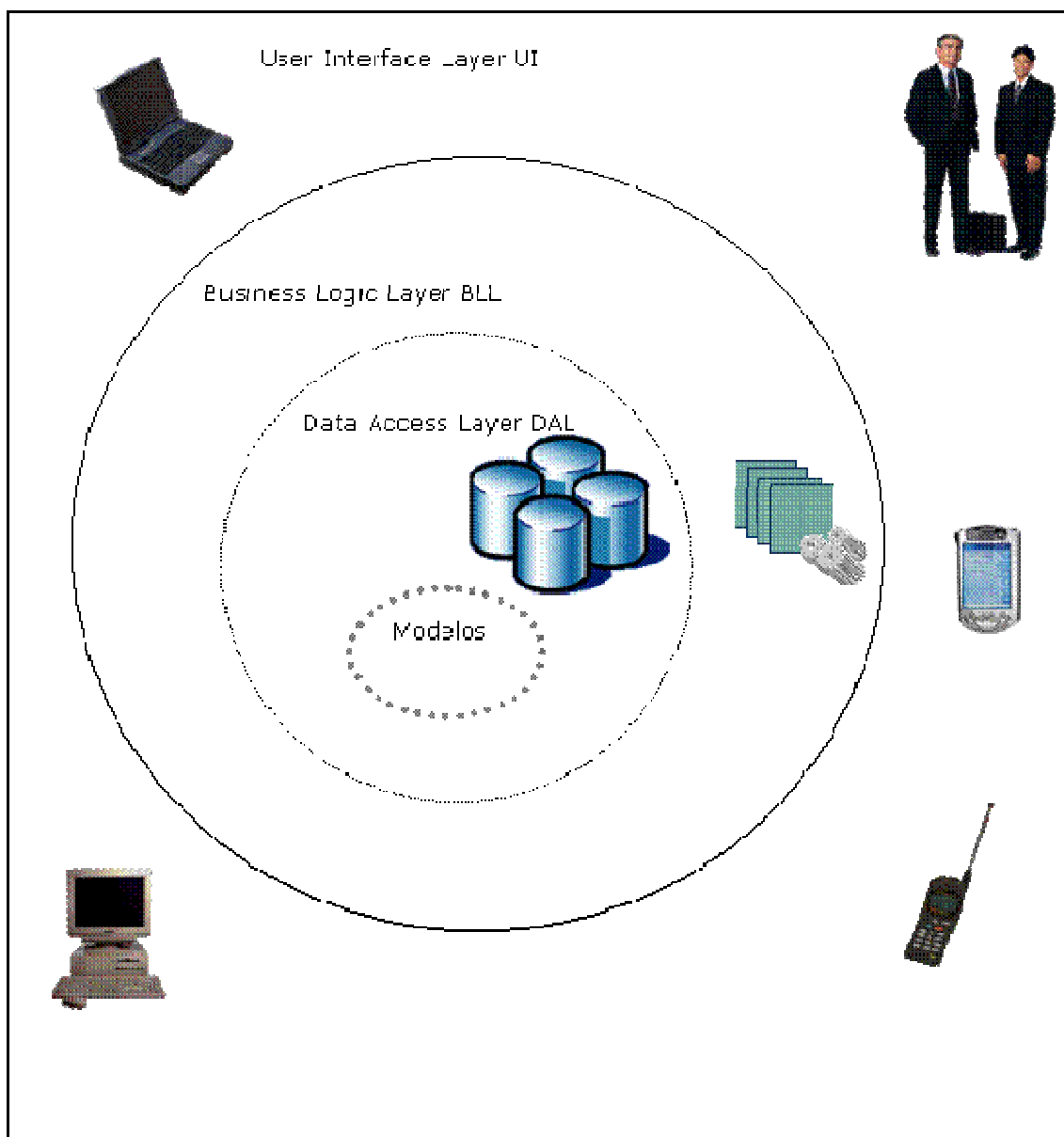


Figura 2 – Representação do Modelo de Desenvolvimento em Camadas.

Autor: Carlos Olavo de Az. Camacho Jr.

Interpretando a figura da camada interna para a mais externa, temos o seguinte:

A primeira camada é a Camada de Acesso a Dados que também é chamada de *Data Access Layer* (DAL);

A segunda camada é a Camada de Regras de Negócio, também chamada de *Business Logic Layer* (BLL); e

A terceira camada é a Camada de Apresentação ou Interface com o Usuário, também chamada de *User Interface* (UI).

A Figura 2 representa um **modelo**. A semântica da palavra modelo no nosso contexto tem origem na matemática. Quando utilizamos a palavra modelo, desejamos significar um fragmento de um sistema segundo uma estrutura de conceitos experimentais.

Neste nível de abstração não se especifica qual o banco de dados utilizado no sistema computacional, como por exemplo, Microsoft SQL Server ou Oracle. O modelo está antes da implementação física do projeto; estamos refletindo sobre a fase de planejamento.

O planejamento de projetos é essencial na área de informática, por isso existem ciências como a Engenharia de *Software* e Engenharia de Sistemas.

Iniciamos nossa análise da figura da camada interna para a mais externa, pois no material didático que produzimos foi essa a ordem utilizada quando saímos do modelo abstrato para a efetiva implementação do sistema.

Na **Camada de Acesso a Dados**, realizam-se todas as tarefas onde é necessário estabelecer a conexão com o sistema gerenciador de banco de dados. Tarefas como cadastrar um novo cliente, alterar dados de um cliente, consultar ou excluir informações sobre um cliente são implementadas nessa camada.

A **Camada de Regras de Negócio** é usada para armazenar as regras de como os negócios de uma empresa funcionam. Em outras palavras, aqui definimos quais regras são obrigatórias sobre o gerenciamento dos produtos e serviços de uma empresa.

Para exemplificar um uso dessa camada, imaginemos um *software* que controla os produtos oferecidos aos clientes em um hipermercado. Cada vez que um cliente adquire um pacote de café, ao realizar o pagamento no caixa é realizada a atualização automática desse produto no estoque. O gerente da loja pode definir uma regra para que, sempre que um produto estiver com a quantidade de estoque inferior a 1000 unidades, este deve aparecer em um relatório chamado “produtos para reposição”. Esse relatório então seria acompanhado pelo gerente para que fossem realizados os pedidos necessários aos fornecedores de maneira a prevenir a falta de produtos no hipermercado.

A **Camada de Apresentação** é a interface do *software* com o usuário, onde ocorre a interação homem-máquina. No início da Internet era comum se ouvir dizer que a interface com o usuário eram “as telas”. Isso acontecia porque toda a interação possível com os computadores era estabelecida através da exibição de telas nos monitores, de maneira que o usuário pudesse, por exemplo, informar a opção desejada em um menu ou preencher determinado campo para a realização de uma tarefa.

Com a evolução das Tecnologias da Informação e Comunicação, hoje a interface homem-computador pode ser realizada através de um número cada vez maior de dispositivos. Temos como exemplos:

- Formulários Web;
- Sistemas e aplicativos em computadores portáteis como notebooks e netbooks;
- Aparelhos celulares;

- Videogames já possuem alta capacidade de processamento e acesso à Internet;
- Canais de TV fornecidos em padrão digital;
- Carros com sistemas digitais embutidos como localizadores GPS; dentre outros.

As inovações para facilitar a interação do ser humano com os dispositivos eletrônicos estão indo além do que muitos imaginavam ser possível. Atualmente a integração dentre estes dispositivos também está cada vez maior e isso é o que denominamos convergência digital.

Até equipamentos do cotidiano como geladeira, fogão e máquina de lavar tendem a fazer parte dessa integração tecnológica que visa trazer-nos cada vez mais benefícios.

CAPÍTULO 3

Alunos do CEFET

3.1 Histórico do IFSP

Antes da inauguração do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP), esta instituição de ensino era conhecida como Centro Federal de Educação Tecnológica – CEFET.

O Centro Federal de Educação Tecnológica de São Paulo – CEFET-SP é uma instituição vinculada diretamente à Secretaria de Educação Profissional e Tecnológica – SETEC, do Ministério da Educação, e integra o Sistema Federal de Ensino.

Os CEFETs têm por finalidade formar e qualificar profissionais em diferentes níveis e modalidades de ensino para diversos setores da economia, realizar pesquisa e desenvolvimento (P&D) de novos processos, produtos e serviços em estreita articulação com os setores produtivos e a sociedade, de maneira a oferecer mecanismos para a educação continuada.

A história do CEFET-SP registra, em suas diversas fases, além da manutenção de educação pública de qualidade, oferta de formação profissional orientado ao desenvolvimento econômico brasileiro, especialmente no Estado de São Paulo. Criado em 1909 como Escola de Aprendizes e Artífices, seu projeto era voltado para a formação de operários e contramestres. Entre os anos de 60 e 90, como Escola Técnica Federal de São Paulo, formou técnicos de nível médio para as áreas de Mecânica e Edificações, depois para Eletrotécnica, Eletrônica, Telecomunicações, Processamento de Dados e Informática Industrial.

A segunda unidade foi inaugurada em 1987, no município de Cubatão, litoral paulista, e em 1996 a terceira unidade, em Sertãozinho, localizada no interior do Estado.

Nesse último período houve o reconhecimento público da Escola Técnica Federal de São Paulo pelo excelente preparo profissional dos alunos egressos. A partir do ano 2000 houve a remodelação e expansão da oferta de ensino da instituição. Atualmente o CEFET-SP possui aproximadamente 7000 alunos matriculados e atua nos seguintes níveis e modalidades: Cursos Tecnológicos (nível superior), Formação de Professores (nível superior), Cursos Técnicos, Qualificação Profissional Básica e Ensino Médio.

A proposta pedagógica que norteia o seu currículo se baseia nos princípios explícitos da contextualização e da interdisciplinaridade e, implicitamente, na estética da sensibilidade, na política da igualdade e na ética da identidade.

O desenvolvimento dos currículos contempla tanto a transmissão dos conhecimentos acumulados historicamente pela humanidade como a construção de novos conhecimentos. Tem como objetivo prover através do ensino as condições básicas para a inserção no mercado de trabalho, a plena atuação na vida cidadã e as habilidades metacognitivas para continuar aprendendo.

Desde o início de 2009 o CEFET-SP passou a designar-se Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP. O IFSP - Campus Cubatão, instituição onde obtivemos autorização para a realização deste trabalho, é autorizado pela portaria nº 158 do Ministro da Educação, publicada no DOU de 12/03/1987.

Localizado no Jardim Casqueiro no município de Cubatão, o campus conta hoje com aproximadamente 1000 alunos distribuídos nos seguintes cursos:

- Ensino Médio;
- Educação de Jovens e Adultos Profissionalizante;

- Curso Técnico de Informática;
- Curso Técnico de Eletrônica;
- Curso Técnico de Automação;
- Curso Superior de Tecnologia em Automação Industrial;
- Curso Superior de Tecnologia em Gestão de Turismo.

3.2 Público-alvo da pesquisa: Alunos do Curso Técnico em Informática

O Curso Técnico em Informática está estruturado em 4 semestres/módulos.

Apesar de não ser foco do nosso trabalho explorar detalhes quanto ao conteúdo programático deste curso, faz-se necessária uma breve análise das habilidades e competências previstas em algumas disciplinas. Com isso buscamos justificar a escolha do grupo de alunos para a realização do projeto de pesquisa, detalhado posteriormente neste capítulo.

A seguir tem-se a relação de disciplinas do curso sob análise, organizada em módulos.

Disciplinas Módulo I
1. Fundamentos de Processamento de Dados
2. Sistemas Operacionais
3. Técnicas de Programação
4. Prática de Laboratório
5. Linguagem de Programação
6. Língua Portuguesa (Português Instrumental)
7. Inglês Instrumental
8. Ciências Aplicadas à Estatística
9. Ciências Aplicadas à Matemática

Tabela 1 – Relação de disciplinas do Módulo I – Curso Técnico em Informática CEFET-SP

Disciplinas Módulo II
1. Fundamentos de Processamento de Dados
2. Sistemas Operacionais
3. Técnicas de Programação
4. Prática de Laboratório
5. Linguagem de Programação
Estrutura / Modelagem de Dados
Inglês Instrumental
Matemática Financeira

Tabela 2 – Relação de disciplinas do Módulo II – Curso Técnico em Informática CEFET-SP

Disciplinas Módulo III
Linguagem de Programação Orientada a Objetos
Sistemas Operacionais
Linguagem de Programação
Banco de Dados
Estudo dos Sistemas de Informação
Gestão de Negócios
Redação Técnica

Tabela 3 – Relação de disciplinas do Módulo III – Curso Técnico em Informática CEFET-SP

Disciplinas Módulo IV
Linguagem de Programação
Linguagem de Banco de Dados
Projeto de Sistemas
Segurança do Trabalho
Gestão de Negócios
Trabalho de Conclusão de Curso

Tabela 4 – Relação de disciplinas do Módulo IV – Curso Técnico em Informática CEFET-SP

A disciplina Técnicas de Programação, existente nos módulos I e II, tem dentre seus objetivos desenvolver habilidades de:

- Selecionar e utilizar estruturas de dados na resolução de problemas computacionais;
- Elaborar e executar casos e procedimentos de testes de programas;
- Redigir instruções de uso dos programas implementados;
- Aplicar técnicas de modularização, especificação e verificação de *software*;
- Aplicar Técnicas de programação orientada a objetos.

Já a disciplina Linguagem de Programação, existente nos módulos I e II, tem dentre seus objetivos desenvolver as habilidades:

- Utilizar os conceitos de desenvolvimento de aplicações em linguagem própria para a internet;
- Identificar e elaborar aplicações para internet com planejamento e estrutura de organização;

A disciplina Estrutura e Modelagem de Dados, lecionada no módulo II, tem por objetivo desenvolver as seguintes habilidades:

- Interpretar as estruturas modeladas usando um banco de dados e aplicando as regras de negócio definidas;
- Utilizar ambientes e linguagens para manipulação de dados nos diversos modelos de SGBD (Sistemas de Banco de Dados: Relacional, Orientado a objetos, dentre outros);
- Utilizar os conceitos de desenvolvimento de aplicações em camadas, como cliente, servidores de aplicações, regras de negócio e servidores de banco de dados.

A experiência da apresentação do desenvolvimento de um protótipo que contemple as três camadas previstas no modelo de desenvolvimento em camadas foi realizado junto aos alunos do terceiro módulo pois, para um melhor aproveitamento, é importante que os alunos tenham concluído as disciplinas: Técnicas de Programação, Linguagem de Programação e Estrutura e Modelagem de Dados.

No material didático produzido para auxiliar os alunos, é apresentado o protótipo de uma aplicação construída em três camadas que recebeu o nome de LOJA .NET.

3.3 O projeto LOJA .NET

O projeto LOJA .NET foi criado com o objetivo de auxiliar o aluno na aquisição da competência de utilizar o método de desenvolvimento em camadas no desenvolvimento de *software*.

Mas o que vem a ser competência?

Segundo Philippe Perrenoud, competência é a “faculdade de mobilizar um conjunto de recursos cognitivos (saberes, capacidades, informações, etc) para solucionar com pertinência e eficácia uma série de situações. Estão ligadas a contextos culturais, profissionais e condições sociais”.

Ao invés da memorização de conteúdos, para desenvolver competências é necessário permitir que o aluno exercite suas habilidades. Isso o levará à aquisição de novas competências.

Enquanto aprendizes, interiorizamos aquilo que de alguma forma está ligado ao conteúdo por um desafio, necessidade ou motivação.

Vejamos um exemplo: um indivíduo aprende a andar de bicicleta porque decidiu aceitar o desafio de aprender essa atividade. Estava motivado a aprender e, essa motivação, permitiu levantar diversas vezes até que a competência fosse alcançada.

Vê-se então que a competência é uma construção mental e não apenas a execução de tarefas. Quem está executando a ação deve saber por que a está realizando dessa forma e não de outra maneira.

Na área de desenvolvimento de *software* é impossível memorizar todos os comandos existentes em qualquer linguagem de programação. Por isso é tão importante a abstração da implementação física dos programas para modelos matemáticos.

Se o aluno possui competência na utilização de orientação a objetos, por exemplo, poderá utilizá-la na construção de um *software* para a Internet independente da linguagem de programação utilizada.

3.3.1 A necessidade de definir ferramentas

Para desenvolver competências é necessário trabalhar com a utilização de projetos. Propor tarefas complexas e desafios aos alunos permite mobilizar seus conhecimentos e habilidades.

Visando permitir a implementação de um projeto real capaz de demonstrar o uso e as vantagens do modelo de desenvolvimento de *software* em camadas, houve a necessidade da adoção de determinado ambiente integrado de desenvolvimento (ou *IDE Integrated Development Environment*¹), uma linguagem de programação específica e um Sistema Gerenciador de Banco de Dados.

¹ IDE, do inglês Integrated Development Environment ou Ambiente Integrado de Desenvolvimento, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de *software* com o objetivo de agilizar este processo.

Geralmente os IDEs facilitam a técnica de RAD (de Rapid Application Development, ou "Desenvolvimento Rápido de Aplicativos"), que visa a maior produtividade dos desenvolvedores.

Considerando a experiência do pesquisador na implementação de soluções de informática utilizando a tecnologia Microsoft, para a criação do projeto LOJA .NET utilizou-se:

- Sistema Operacional: Microsoft Windows XP Professional;
- Linguagem de Programação: C# .NET;
- Sistema Gerenciador de Banco de Dados: Microsoft SQL Server 2005;
- Ambiente de Desenvolvimento: Microsoft Visual Studio 2005 Professional Edition.

3.3.2 Publicação de artigos

Tendo idealizado o projeto LOJA .NET, era necessário compartilhá-lo junto aos alunos do curso técnico em informática do CEFET para que fosse possível obter um feedback da eficácia desta proposta.

Com isso em mente, pesquisamos os principais portais na Internet de Tecnologia da Informação para a submissão de uma série de artigos para aprovação.

Após algumas semanas de pesquisa identificamos um forte candidato para a submissão de artigos, tratava-se de um portal de tecnologia denominado Linha de Código.

O primeiro artigo foi aceito e publicado pelo portal Linha de Código em 08/04/2008¹ e o último, de uma série de 9 artigos, foi publicado no mesmo site em 01/08/2008².

No período de abril a agosto de 2008 verificou-se um crescente interesse por parte de programadores iniciantes e a área de comentário dos artigos virou uma espécie de comunidade virtual, local onde os programadores trocavam experiências relacionadas ao conteúdo da série de artigos.

A série de artigos foi construída como sendo um relatório científico. Era necessário permitir a qualquer leitor uma visão global do estudo realizado, proporcionando a fácil consulta e fornecendo de modo objetivo as informações mais relevantes. Além dos desafios pedagógicos inerentes ao Ensino à Distância, houve desafios técnicos como a tentativa de prever a série de problemas que poderiam ocorrer caso os leitores não estivessem com uma gama de opções pré-definidas em seus computadores.

O pesquisador adquiriu equipamentos para planejar e montar em sua residência um verdadeiro Laboratório para uma centena de testes demandados pelo projeto LOJA .NET. Diferentes versões de sistemas operacionais (Windows XP Home, Windows XP Professional, Windows Vista), diferentes versões de Sistemas Gerenciadores de Bancos de Dados (Microsoft SQL Server 2000, 2005 e 2008) bem como diferentes versões da própria IDE Microsoft Visual Studio (MSVS 2003, 2005, 2008) foram utilizadas para criar variados cenários. O projeto LOJA .NET foi exaustivamente testado pois o desejo do pesquisador foi produzir um material com a melhor qualidade possível.

¹ <http://www.linhadecodigo.com.br/Artigo.aspx?id=1898>

² <http://www.linhadecodigo.com.br/Artigo.aspx?id=1938>

Outro fator a ser ressaltado foi a missão de realizar uma redação simples e objetiva. O objetivo era permitir a prática do desenvolvimento de *software* sem a necessidade da transmissão explícita de teorias computacionais ou matemáticas. O pesquisador desenvolveu esse material didático como um guia prático de maneira a permitir que os leitores pudessem desenvolver o projeto inteiro sem que tivessem, necessariamente, domínio das teorias envolvidas.

Assim, o leitor pode se aprofundar na teoria relacionada ao tópico estudado conforme sua necessidade e interesse. Conforme vimos no pensamento complexo de Morin explicitado no primeiro capítulo, o Aprender a Conhecer e o Aprender a Fazer, dependem sobretudo das atitudes do aprendiz. A transdisciplinaridade exige do aprendiz mais do que obedecer a ordens impostas pelas regras da sociedade em que vive. Não basta interiorizar esse conjunto de regras, o aprendiz deve se conhecer para então poder escolher corretamente sua profissão. Quanto maior o autoconhecimento, maior a chance de se escolher a profissão em que o aprendiz se sentirá realizado, motivado, feliz. Amar o seu ofício é o que permite não só se especializar constantemente, mas sobretudo inovar.

Alguns leitores da série de artigos demonstraram satisfação com esse trabalho e sugeriram ao pesquisador uma expansão dos artigos e a publicação de um livro no mesmo formato didático.

O pesquisador imaginava os desafios envolvidos no projeto de publicação de um livro mas não hesitou. Seguiu em frente, persistiu, e para a felicidade dos leitores que enviaram mensagens de apoio, obteve sucesso nessa empreitada.

3.3.3 O livro "Desenvolvimento em Camadas com C# .NET"

O pesquisador recebeu muitos e-mails dos leitores dos artigos disponibilizados no portal Linha de Código. Algumas dessas mensagens eram de incentivo para a expansão do conteúdo e a publicação de um livro sobre a metodologia de desenvolvimento em camadas utilizando a linguagem de programação C# .NET.

Vislumbramos a real possibilidade da publicação de uma obra que abordasse de forma totalmente prática o conceito de desenvolvimento em camadas. Os desafios relacionados ao sucesso na realização deste projeto eram conhecidos:

- O conteúdo da obra é inédito no mercado?
- Qual é o público-alvo?
- O que difere a obra das já existentes sobre o tema?

Essas questões nortearam o trabalho de expansão do projeto LOJA .NET para atender a demanda tanto dos leitores como do mercado.

Após a conclusão da expansão do projeto LOJA .NET, o primeiro capítulo foi submetido à diversas editoras especializadas em livros técnicos até que o projeto de publicação do livro foi acordado com a Visual Books Editora Ltda.

Tendo realizado as formalidades contratuais, iniciamos em equipe um trabalho de revisão dos capítulos, ilustração da capa e outros detalhes. Em novembro de 2008 foi publicado o livro “Desenvolvimento em Camadas com C# .Net”, obra que teve como ilustração de capa a imagem a seguir:

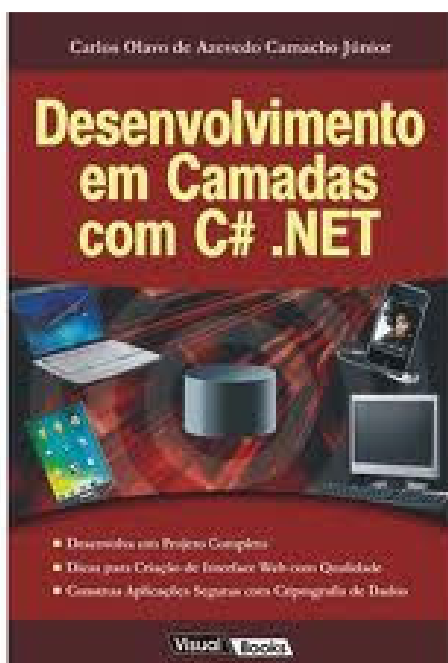


Figura 3 - Capa do livro Desenvolvimento em Camadas com C# .Net
Designer: Júlio Cesar Winck

A partir do 2º semestre de 2008, o Prof. Nelson Paz começou a utilizar a série de artigos publicados no Portal Linha de Código como apoio na disciplina Linguagem de Programação III – (LPG-III). Após a publicação do livro em novembro de 2008, este foi adotado como material de apoio para esta disciplina, ministrada aos alunos do 3º Semestre do Curso Técnico em Informática do IFSP.

O pesquisador foi convidado pela administração do CEFET – Cubatão para proferir uma palestra na “Semana Nacional de Ciência e Tecnologia: Ciência para o Desenvolvimento Sustentável”. Esse evento ocorreu no dia 22 de Outubro de 2010 e a imagem a seguir tem grande valor para o pesquisador. É o registro do momento em que o professor Nelson Paz realiza a cerimônia de entrega do Certificado de Palestrante após a realização do evento.



Figura 4 – Palestra proferida para os alunos do Professor Nelson Paz - CEFET – Cubatão. 22/10/2010. Carlos Camacho (à esquerda) e Prof. Nelson Paz.

3.4 Análise do Questionário Final

O grupo de trabalho no 3º Semestre do Curso é composto em média por um número de 15 alunos.

Conforme o citado no item 3.2 (público-alvo), antes de iniciar este projeto os alunos já haviam concluído disciplinas como orientação a objetos, momento em que aprenderam a linguagem de programação chamada Java.

A pesquisa realizada com questionários junto aos alunos após a realização do projeto LOJA .NET nos permitiu fazer uma tabulação das habilidades adquiridas e percebidas pelos alunos ao término do processo.

O professor Nelson colaborou com as atividades apresentando o projeto LOJA .NET para os alunos do CEFET e juntos definimos um plano de trabalho.

No segundo semestre de 2008, o professor realizava a construção dos programas juntamente com os alunos, utilizando-se do *datashow* como um dos recursos didáticos.

Já no primeiro semestre de 2009, o professor percebeu um melhor aproveitamento da turma quando utilizou o quadro negro (branco) e giz (pincel) para as aulas de programação em laboratório de informática. No início da aula o professor fazia exposição do objetivo da aula no quadro para que todos compreendessem. Feita a apresentação, o professor solicitava que cada grupo de alunos codificasse a solução dos exercícios propostos em seus computadores, permitindo a consulta entre grupos.

Ao fazer junto com os alunos a programação dos exercícios, mais dúvidas foram esclarecidas e percebeu-se um melhor aproveitamento da aula. O professor podia utilizar-se da lousa para realizar explicações mais detalhadas enquanto os alunos tinham liberdade para implementar à sua maneira os conceitos de programação demonstrados em laboratório. Verificamos que anteriormente quando se usava intensamente somente o recurso Datashow, alguns alunos ficavam dispersos.

Ocorrendo durante todo o processo de ensino-aprendizagem, essa avaliação favoreceu a negociação e a autonomia do aluno no gerenciamento de seu aprendizado, possibilitando o desenvolvimento de competências sócio-profissionais.

É importante que o educador consiga determinar “aonde seu aluno já chegou” quando o desejo é que ele tenha um grupo de habilidades necessárias à competência em desenvolver *software*. O aluno já sabe o que é uma “classe” e como implementá-la?

Ao realizar esse questionamento, se quer a certificação de que o aluno tem o conceito e sabe construir uma “classe” por conta própria, sem a necessidade da ajuda de outras pessoas.

Uma das maneiras de identificar isso é quando o professor está com esse aluno em laboratório de informática. O educador propõe um problema e analisa o que o aluno faz para tentar solucioná-lo.

Se o aluno constrói sozinho uma classe, significa que ele tem a capacidade de realizar esta tarefa de forma independente. Isso é o que Vygotsky chama de **nível de desenvolvimento real**.

As funções psicológicas que fazem parte do nível de desenvolvimento real do aluno em determinado momento de sua vida são aquelas já bem estabelecidas naquele momento.

Conhecendo o que seu aluno pode fazer por conta própria, é importante agora saber o que ele consegue fazer com a ajuda de outras pessoas mais capazes. A capacidade de desempenhar tarefas com a ajuda de outras pessoas, é o que Vygotsky denomina **nível de desenvolvimento potencial**.

Existem tarefas que o aluno não consegue desempenhar sozinho, mas após uma demonstração do professor ou de pistas dadas, ele é capaz de realizá-las. Daí a justificativa para o pesquisador propor ao professor Nelson que os alunos se organizassem em grupos de trabalho. A interação social tem grande importância no processo de construção das funções psicológicas humanas.

Agora podemos definir o que, segundo Vygotsky, é conhecida por **zona de desenvolvimento proximal**. É a distância entre o nível de desenvolvimento real e o nível de desenvolvimento potencial. É, assim, o caminho que o aluno vai percorrer até que consiga realizar por conta própria as tarefas que hoje só consegue realizar com o auxílio de alguém.

A zona de desenvolvimento proximal está em constante mutação na vida de um ser humano: aquilo que uma pessoa só é capaz de realizar com a ajuda de alguém hoje, será possível ser realizado de maneira independente no futuro. (Vygotsky, 1997, p. 58-60)

3.4.1 Aplicação de Questionários

No início do 1º. Semestre de 2009 aplicamos um questionário junto aos alunos como sondagem para averiguar como eles estavam com relação ao nível de desenvolvimento real.

Após a execução do projeto LOJA.NET, que finalizou em julho de 2009, aplicamos um novo questionário para verificar como os alunos se auto-avaliavam com relação ao nível de desenvolvimento real.

Estes questionários serviram de base para as análises de habilidades e competências adquiridas que realizaremos posteriormente neste capítulo.

Paralelamente ao andamento da disciplina LPG-III, os alunos desenvolviam o Trabalho de Conclusão de Curso –TCC. Neste contexto o TCC consiste na construção de um *software* onde cada aluno tem autonomia para definir a linguagem de programação a ser utilizada bem como todos os recursos envolvidos no projeto.

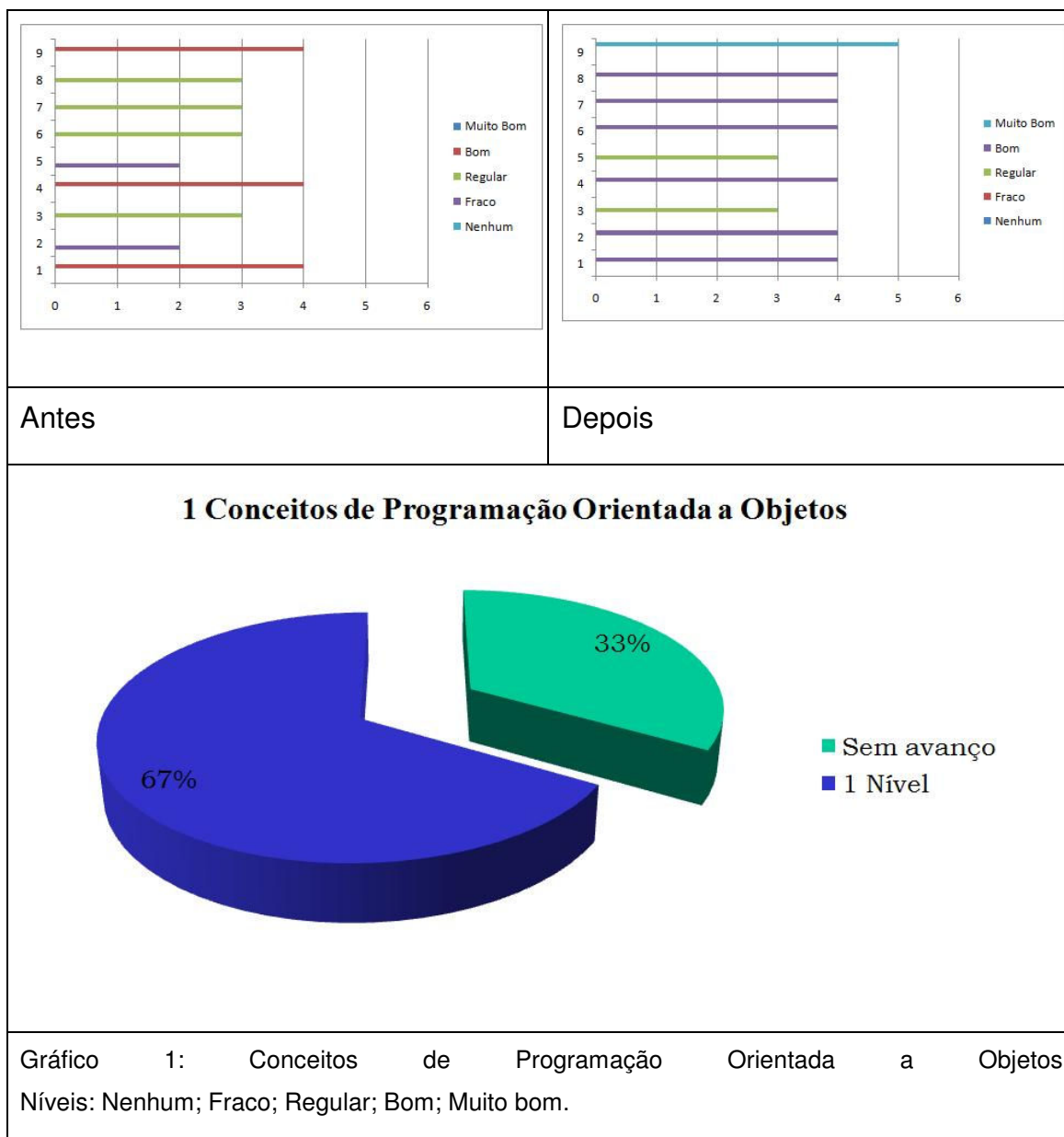
O questionário denominado “Anexo A - Questionário para o(a) Aluno(a)” foi aplicado antes e após a realização do projeto LOJA .NET pelos alunos que cursaram a disciplina LPG-III no 1º Semestre de 2009. O objetivo da aplicação do questionário foi permitir aos alunos uma auto-avaliação sobre determinadas habilidades consideradas relevantes ao processo de desenvolvimento de *software* utilizando a metodologia de Desenvolvimento em Camadas.

Para cada habilidade acompanhada junto aos alunos durante a aplicação do projeto LOJA .NET, temos a seguir:

- 1) Um gráfico que representa o questionário aplicado **antes** do projeto LOJA .NET;
- 2) Um gráfico que representa o questionário aplicado **depois** do projeto LOJA .NET;
- 3) Um gráfico que resume o desempenho dos alunos na habilidade em questão. Os níveis disponíveis nos questionários são: 1) Nenhum conhecimento; 2) Fraco; 3) Regular; 4) Bom; 5) Muito bom. Se um aluno assinalou Bom no primeiro questionário e, no questionário final,

assinalou Muito Bom na sua auto-avaliação, no gráfico resumo este aluno entrará na estatística de alunos que tiveram 1 nível de avanço. Se, por exemplo, um aluno estava inicialmente com o nível Fraco e no questionário final assinalou Bom, entrará na estatística de alunos que tiveram 2 níveis de avanço, pois o nível imediatamente posterior ao Fraco é o Regular, e depois temos o nível Bom.

1) Programação Orientada a Objetos (Uso de classes, métodos e propriedades)



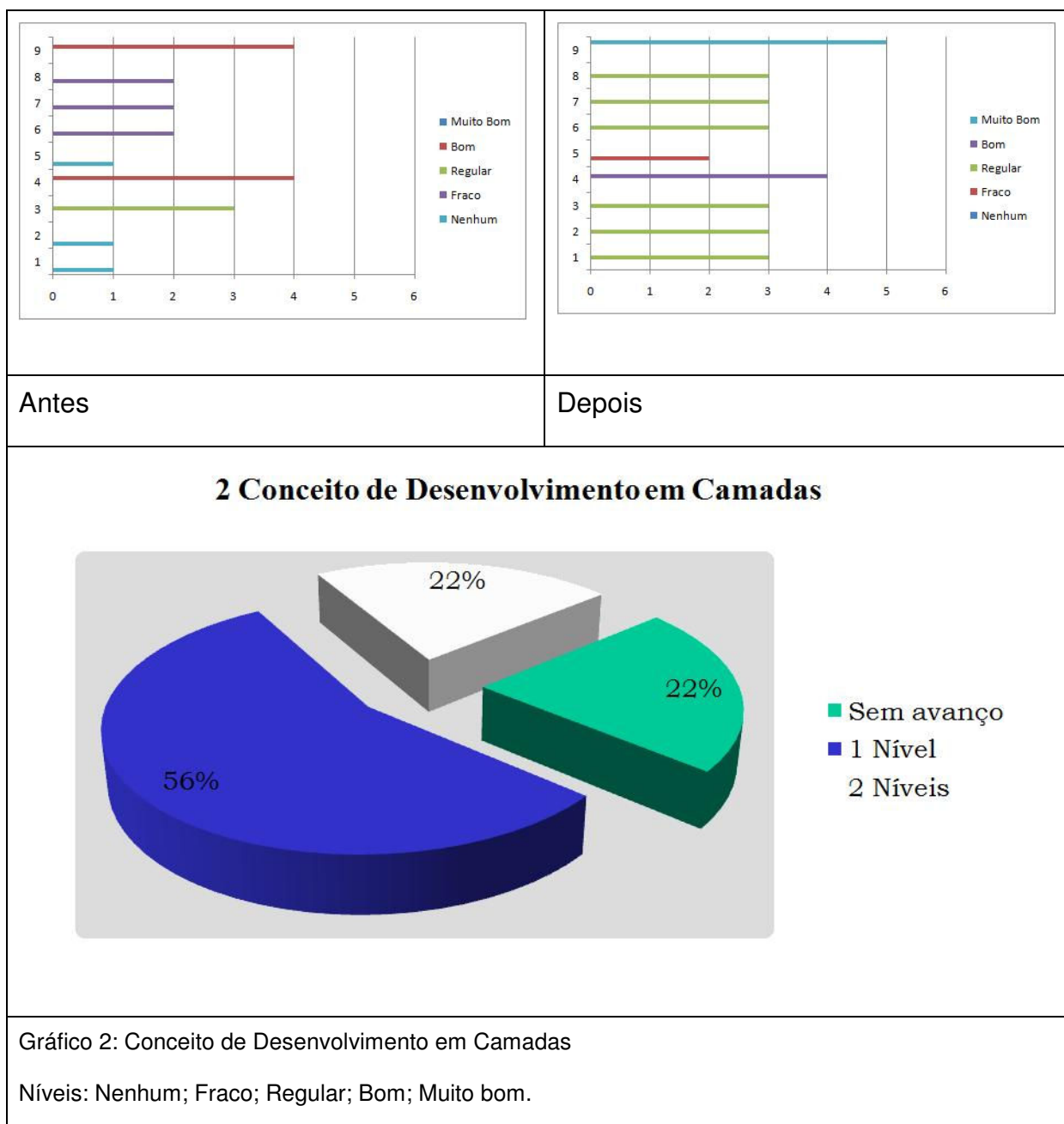
67% dos alunos afirmaram terem notado melhoria sobre o uso do conceito de Orientação a Objetos quando do desenvolvimento de *software*.

No capítulo inicial verificamos que dentre as competências técnicas que o desenvolvedor de software deve ter está a necessidade de saber analisar os documentos que especificam o que o *software* deve fazer.

A prática desse conceito é importante pois é através da experiência que o desenvolvedor pode adquirir a competência de interpretar o que precisa ser feito quando tem em mãos um documento que especifica o que um *software* deve fazer.

Verificamos que com o projeto LOJA .NET os alunos do CEFET tiveram a oportunidade de melhorar seu entendimento sobre como aplicar o conceito de Orientação a Objetos na prática.

2) Conceito de Desenvolvimento em Camadas



78% dos alunos disseram que assimilaram como desenvolver *software* utilizando o método de desenvolvimento em camadas sendo que destes, 22% afirmaram que com o projeto LOJA .NET foi possível compreender muito bem como se faz isso.

No primeiro capítulo mapeamos as cinco competências essenciais ao desenvolvedor de software. Vimos que a segunda competência mapeada consiste em saber utilizar o conceito de Desenvolvimento em Camadas.

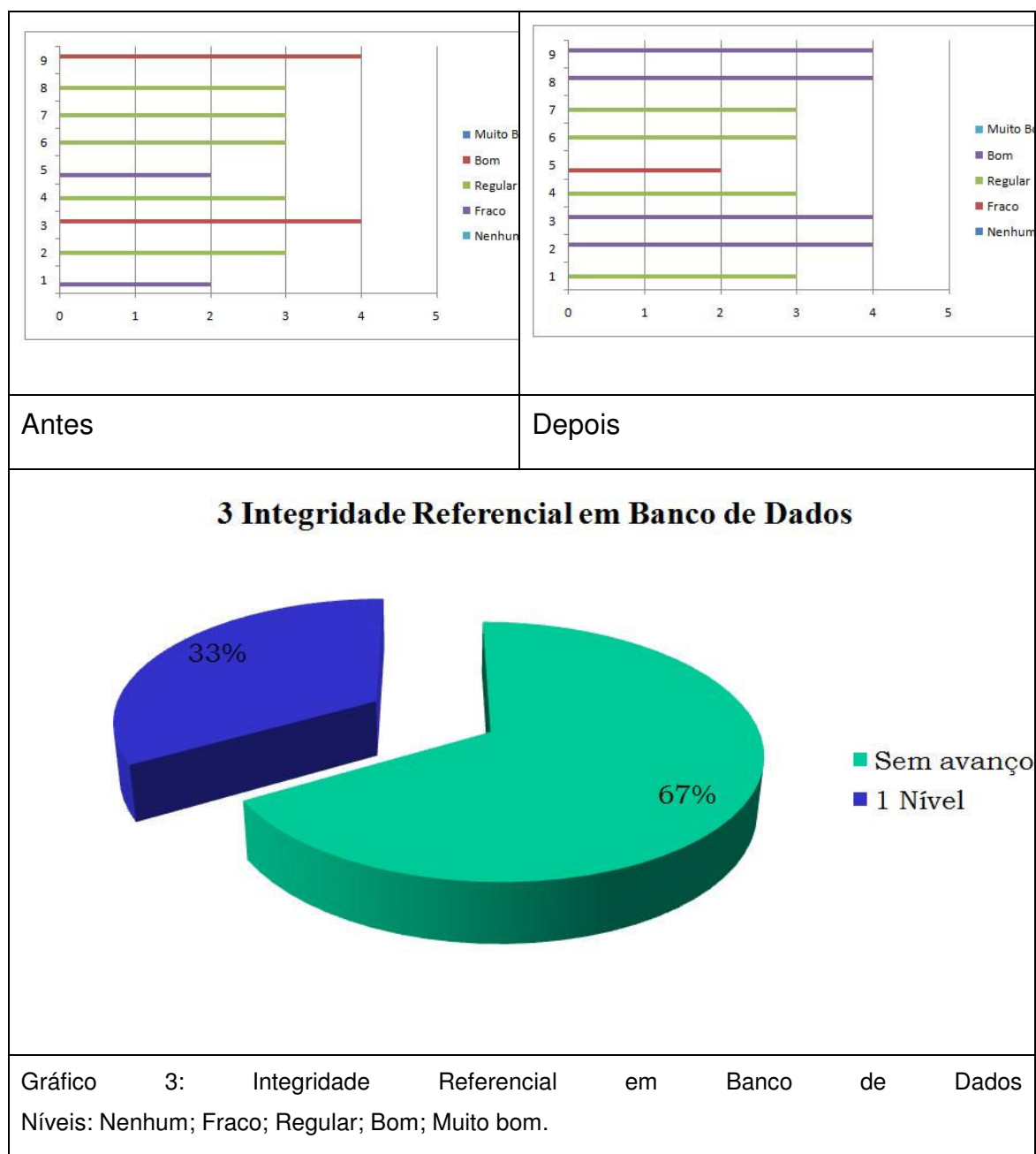
Quando o aluno tem domínio sobre a metodologia de Desenvolvimento em Camadas, significa que ele sabe construir as classes de uma camada e também é capaz de estabelecer a comunicação entre as camadas existentes em um *software*.

Verificamos através do gráfico 2 que a maioria dos alunos compreendeu o processo de comunicação entre as classes.

Em termos técnicos quando uma camada está se conectando à outra dizemos que a primeira está fazendo uma “referência” à segunda. Quando o programador consegue fazer estas referências, ele é capaz de reutilizar tudo o que já está funcionando em uma camada existente. Não é necessário utilizar a programação para reescrever algo que já foi escrito anteriormente por ele mesmo ou por outro desenvolvedor.

Com isso os alunos demonstraram um ganho de produtividade pois conseguiram desenvolver o projeto LOJA .NET no menor tempo possível.

3) Conceito de Integridade Referencial em Banco de Dados



A terceira competência essencial ao desenvolvedor mapeada no primeiro capítulo denomina-se "Configurar e Gerenciar informações em Banco de Dados".

Essa competência envolve as habilidades representadas nos gráficos de número 3, 4 e 5.

A primeira habilidade - representada no gráfico 3 - é o conceito de Integridade Referencial. Analisando o gráfico verificamos que apenas 33% dos alunos perceberam alguma melhoria na assimilação desse conceito.

Apesar da integridade referencial não ser um assunto trivial para programadores iniciantes, no projeto LOJA .NET houve um esforço no sentido de demonstrar a importância e o funcionamento desse conceito na prática.

Quando realizamos uma compra no supermercado por exemplo, compramos diversificados produtos. Um *software* construído para gerenciar as vendas deste supermercado precisa armazenar dados sobre as vendas em um sistema de banco de dados.

Os alunos viram que quando se utiliza um banco de dados, criamos tabelas distintas para armazenar dados sobre assuntos diferentes.

As informações do cliente comprador e da venda não ficam armazenadas em uma única tabela deste sistema de banco de dados.

Os dados sobre o cliente como nome e cpf ficam armazenados em uma tabela e os dados sobre a venda como nome do produto e valor unitário ficam armazenados em outra tabela.

Dessa forma, é preciso criar algo que relacione um cliente da tabela de clientes com todos os produtos que ele comprou - armazenados na tabela de vendas.

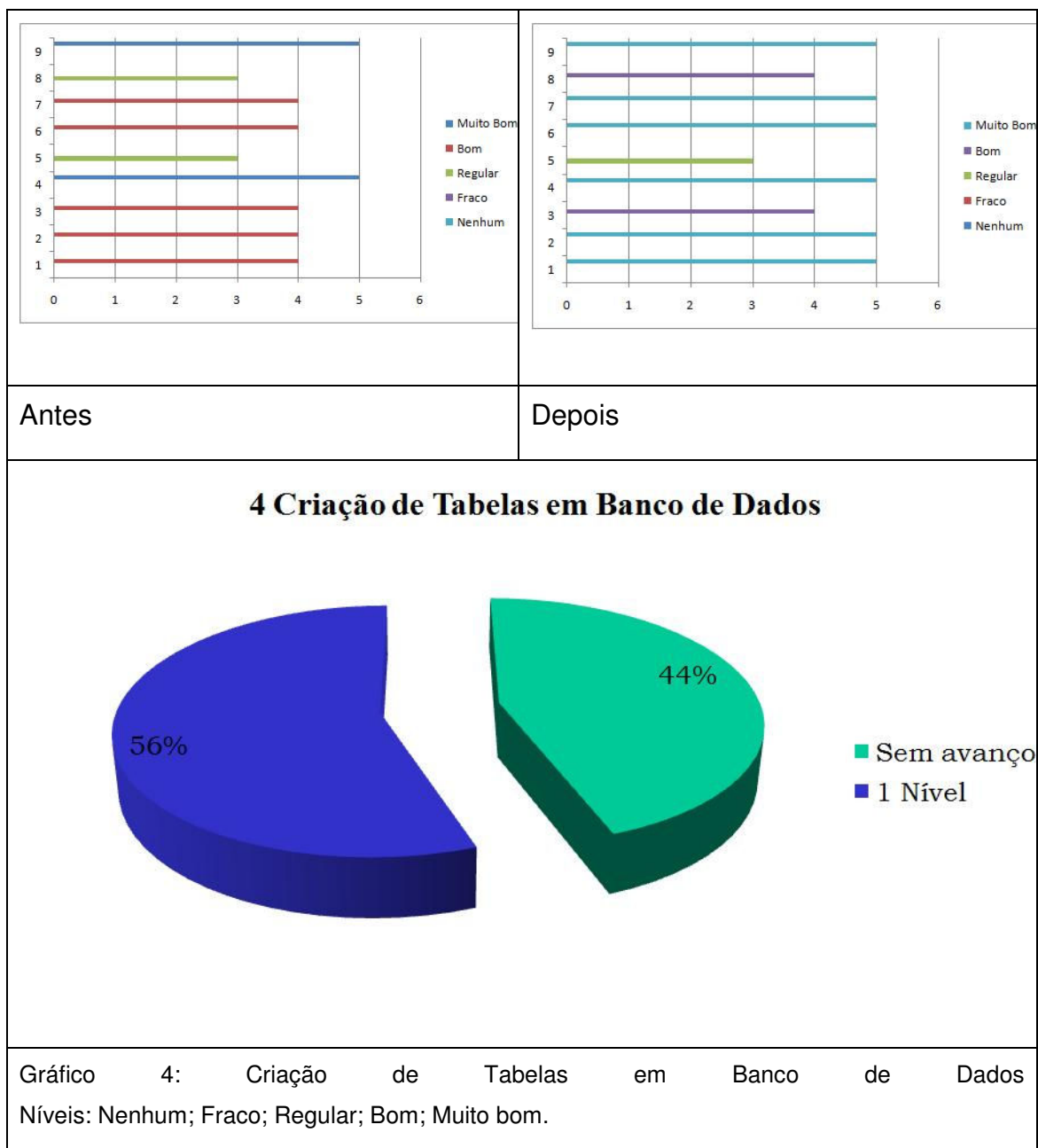
Saber criar e manter esse relacionamento consistente entre as tabelas é o que chamamos de integridade referencial.

O desenvolvedor precisa dominar essa habilidade pois do contrário podemos encontrar um produto registrado na tabela de vendas sem que seja possível identificar qual cliente o comprou.

Os alunos também verificaram através do projeto LOJA .NET que a integridade referencial é importante por questões de performance da utilização do *software*. Quando o funcionário operador do caixa do supermercado está registrando os produtos, o *software* deve responder de forma rápida à essas solicitações para evitar filas e, conseqüentemente, aborrecimentos por parte dos clientes.

Talvez seja necessário reavaliar o modelo de dados utilizado no projeto LOJA .NET para simplificá-lo. É possível também que as explicações durante o processo de desenvolvimento e uso do banco de dados não tenham sido suficientes para a assimilação desse conceito pelos alunos.

4) Criação de Tabelas em Banco de Dados



Com relação a criação de tabelas em banco de dados, 56% dos alunos notaram melhorias após a execução do projeto.

Um sistema de banco de dados é uma estrutura complexa composta por diversas tabelas que se relacionam.

Como vimos na teoria de Morin comentada no primeiro capítulo, para compreender o problema da complexidade é preciso reconhecer o paradigma da simplicidade. No paradigma da simplicidade prevalece a lógica, a ordem.

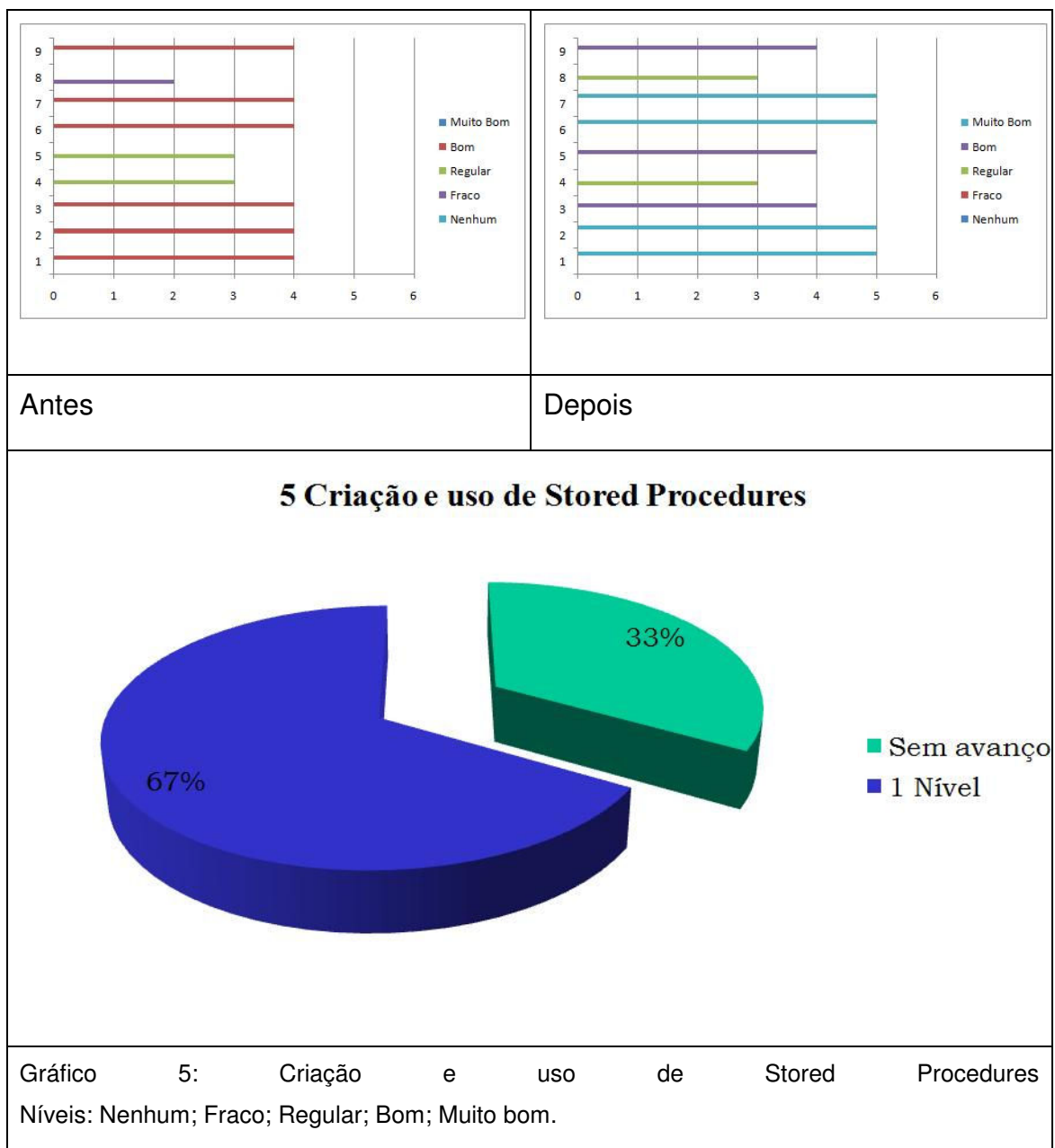
Através do projeto LOJA .NET os alunos puderam verificar que a construção de um sistema de banco de dados começa com a construção de uma tabela de cada vez. Se for preciso criar uma tabela para armazenar os dados de clientes e uma tabela para armazenar dados das vendas, qual delas o aluno deve criar primeiro? Existirá algum problema se o aluno criá-las na ordem que desejar?

No exercício dessa habilidade os alunos compreenderam que é importante obedecer a uma ordem. Primeiro criamos a tabela de clientes e posteriormente criamos a tabela de vendas. Mas por quê?

Para que no momento do registro da venda, já existam clientes cadastrados de forma que o sistema possa relacionar qual cliente está adquirindo determinados produtos.

Os alunos puderam verificar que é aqui o momento de usar, também, o conceito de integridade referencial. Compreenderam que se as tabelas não forem criadas na ordem correta o *software* não vai funcionar de maneira adequada.

5) Criação e uso de *Stored Procedures*



Já sobre a criação e uso de *stored procedures*, 67% dos alunos disseram terem percebido melhorias.

Essa habilidade foi analisada no gráfico 5 e aqui os alunos perceberam que uma *stored procedure* não pode existir sem que haja um sistema de banco de dados.

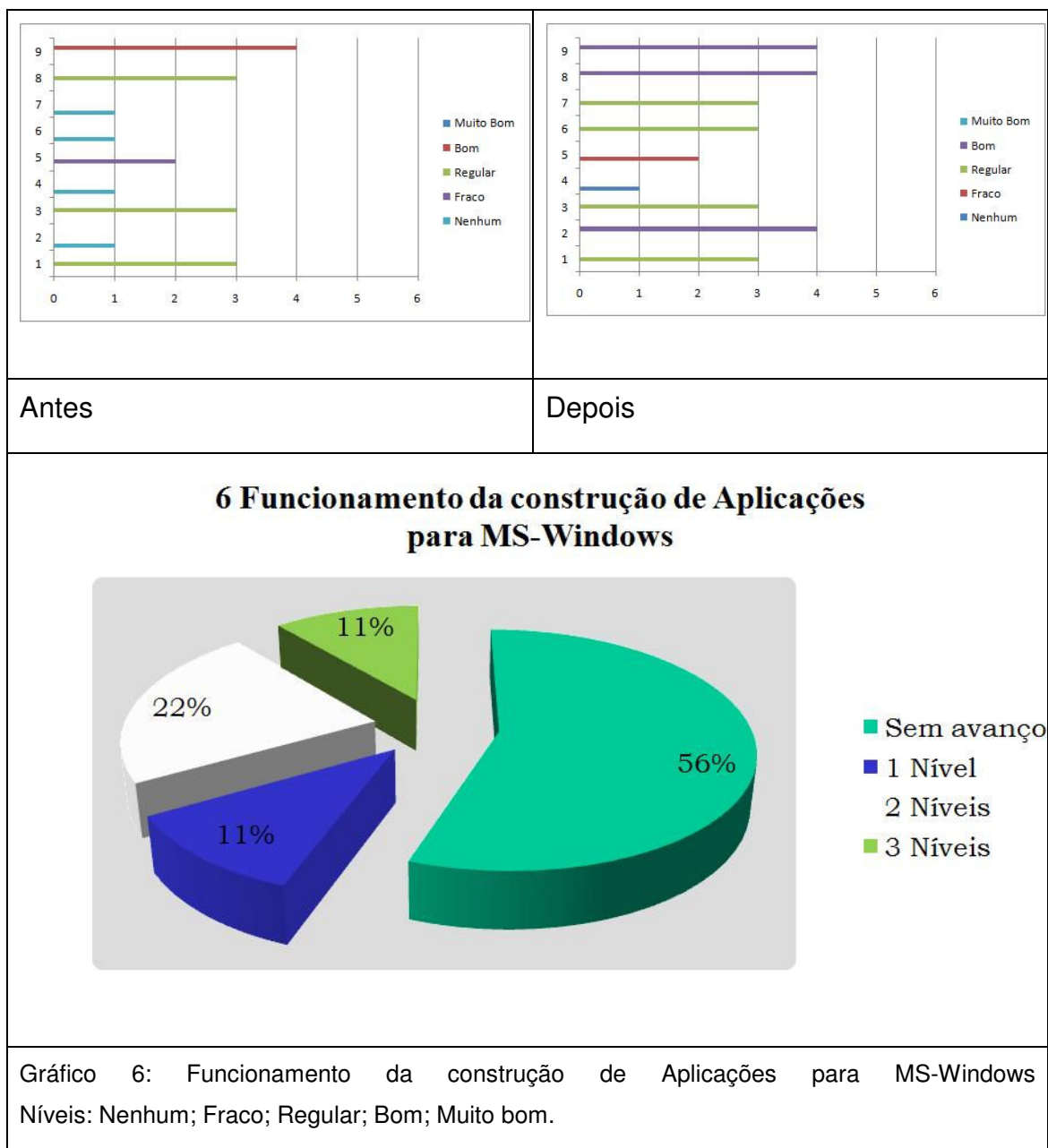
No capítulo inicial Morin afirma: “Vai estudar-se o cérebro como órgão biológico e vai estudar-se o espírito, *the mind*, como função ou realidade psicológica. Esquece-se que um não existe sem o outro; ou melhor, que um é simultaneamente o outro, embora sejam tratados por termos e conceitos diferentes”.

Assim também se relacionam as *stored procedures* e os sistemas de bancos de dados. Para que o aluno compreenda melhor a criação e uso de uma *stored procedure*, é necessário compreender o funcionamento do sistema de banco de dados. Uma *stored procedure* é um conjunto de comandos que são enviados para que o sistema de banco de dados execute.

Se esse conjunto de comandos estiver dentro de um programa, este conjunto não constitui uma *stored procedure*. *Stored procedure* ou Procedimento Armazenado é, como o próprio nome diz, um conjunto de comandos que são armazenados dentro do próprio sistema de banco de dados.

No projeto LOJA .NET os alunos aprenderam a armazenar estes procedimentos no banco de dados e como um *software* pode invocar a execução destes procedimentos.

6) Funcionamento da construção de aplicações para MS-Windows



Construir Aplicações utilizando o modelo de desenvolvimento em camadas:

Essa competência foi analisada a partir dos gráficos de número 6 e 7. Sobre o desenvolvimento de aplicações para Windows, 44% dos alunos perceberam melhorias sendo que destes, 33% perceberam o projeto como um diferencial para a assimilação desta competência.

No segundo capítulo verificamos que o surgimento da Internet foi possível devido a um acordo entre diversas empresas de tecnologia com relação à adoção de padrões. Padrões estes que foram adotados para a construção de cada camada da Internet.

A construção de aplicações para MS-Windows se encontra na Camada de Aplicação da Internet. Ela aproveita as camadas anteriores, como por exemplo, o protocolo de comunicação TCP/IP que permite o envio e recebimento de informações dentre as redes formadas por computadores com diferentes sistemas operacionais.

Como o desenvolvedor de *software* tem a vantagem de não precisar se preocupar com protocolos de comunicação entre redes, pode ter foco na tarefa de construir sua aplicação.

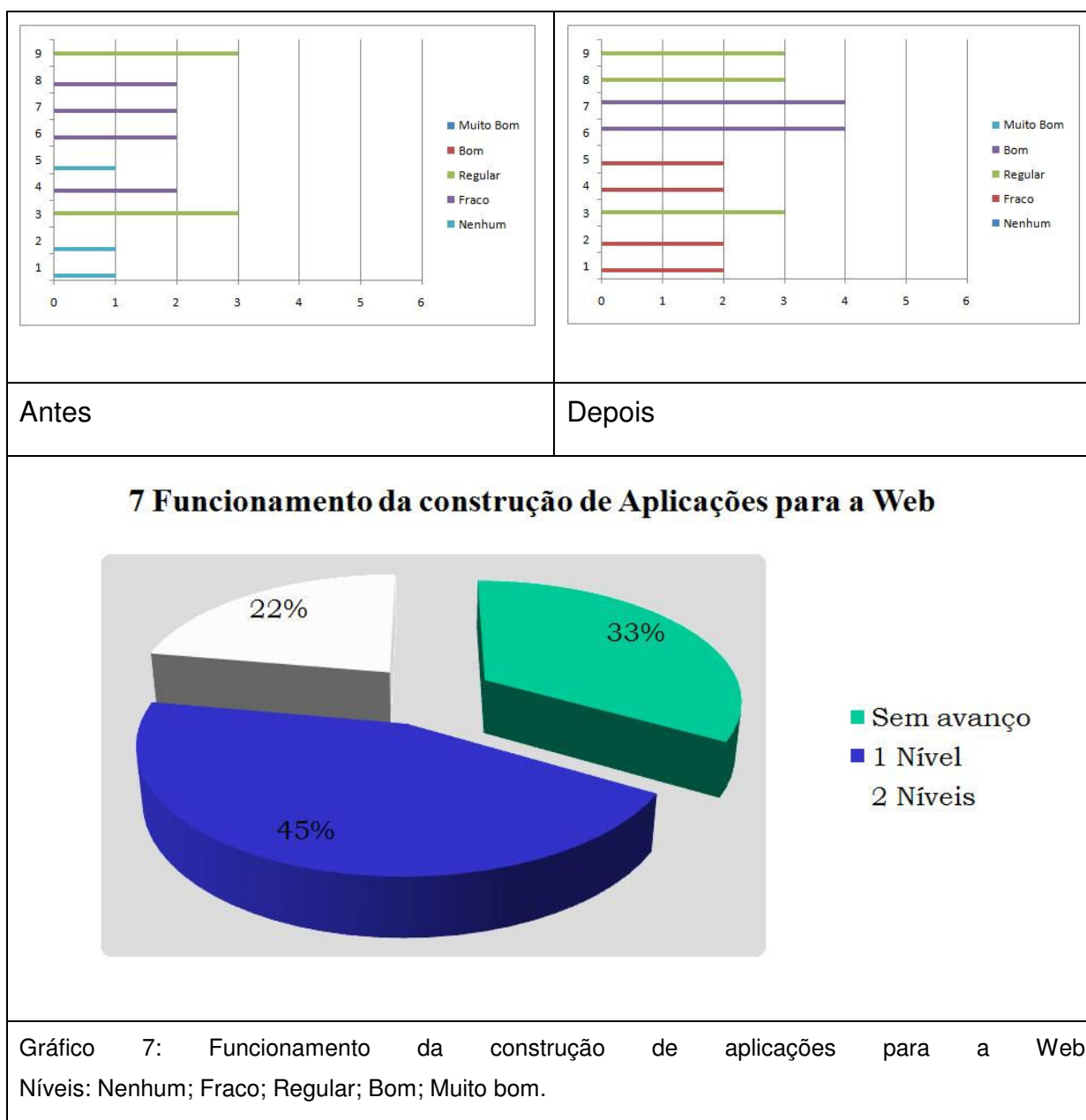
Os alunos construíram no projeto LOJA .NET uma aplicação que foi executada no sistema operacional Microsoft Windows.

Existem alguns cientistas da computação e alunos dedicados em projetos com o objetivo de fazer com que um software desenvolvido possa ser executado da mesma forma nos diferentes sistemas operacionais existentes (como Microsoft Windows, Linux, MacOS, dentre outros).

No laboratório de informática utilizado pelos alunos do CEFET tínhamos licenças de MS-Windows devido a um programa de parceria com a Microsoft. Devido a esse fato e também pelo sistema operacional Microsoft Windows ser o mais utilizado atualmente, foi este o ambiente escolhido para o desenvolvimento do projeto LOJA .NET.

Os alunos puderam experimentar como reunir os recursos de bancos de dados, sistema operacional e linguagem de programação para possibilitar a construção de um software segundo os padrões de qualidade exigidos pelo mercado.

7) Funcionamento da construção de aplicações para a WEB



Construir Aplicações utilizando o modelo de desenvolvimento em camadas:

Com relação ao desenvolvimento de aplicações para a Web, 67% dos alunos avaliaram positivamente o projeto e destes, 22% afirmaram que o projeto foi de grande relevância para seu aprendizado.

Verificamos no capítulo sobre Desenvolvimento em Camadas a importância dos padrões no Desenvolvimento de Aplicações.

Os alunos puderam construir uma página Web que pode ser acessada por qualquer computador conectado à Internet. Isso é possível devido a padronização existente e exigida das empresas que desenvolvem os *browsers* ou navegadores Internet. Alguns exemplos de navegadores são: Internet Explorer e Mozilla Firefox.

Devido ao fato de todos os navegadores só exibirem uma linguagem chamada HTML, é necessário que o desenvolvedor utilize essa linguagem para que seja possível o acesso e visualização do conteúdo existente no site criado.

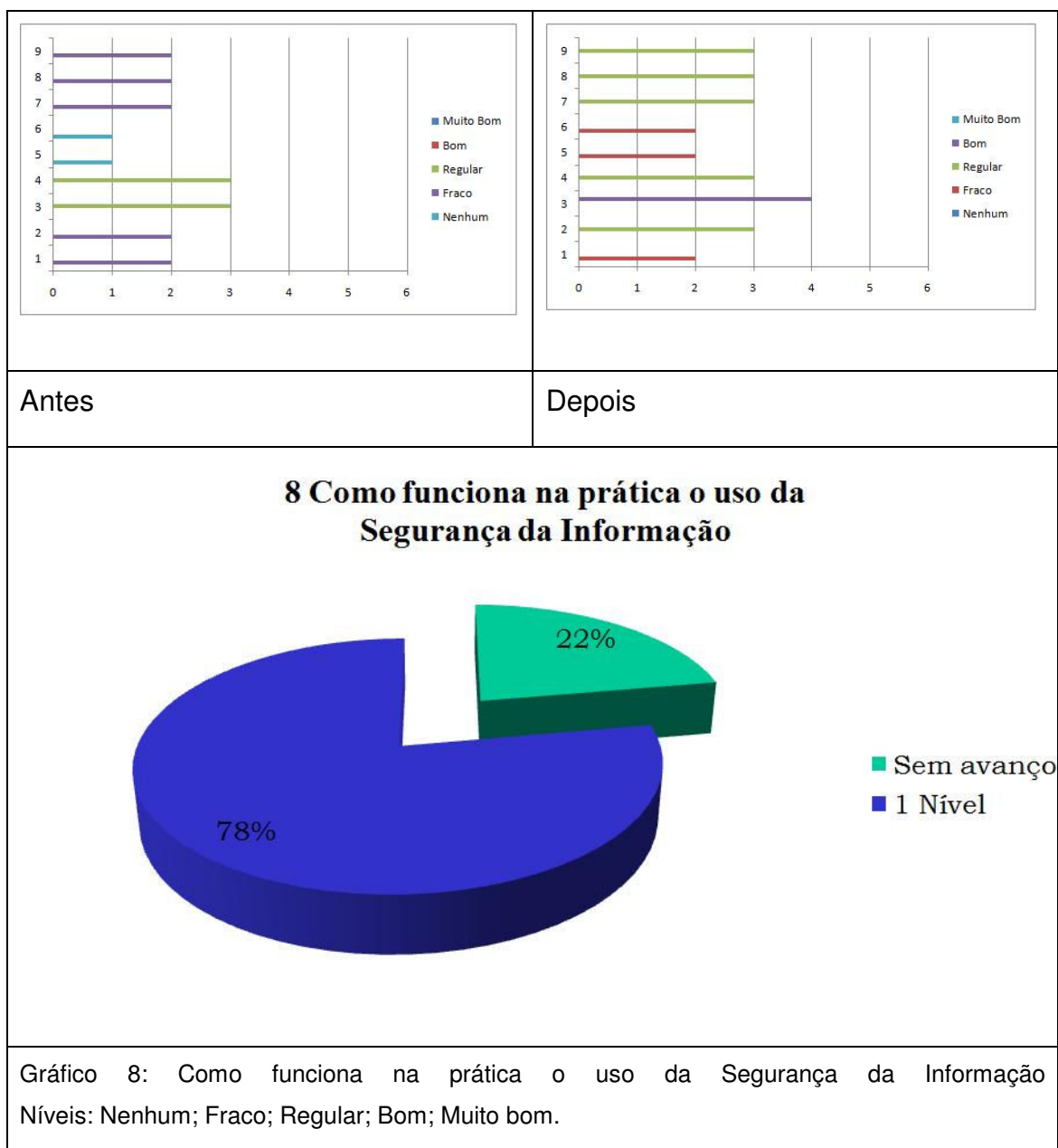
Os alunos experimentaram como a ferramenta Microsoft Visual Studio permite a criação de páginas para a Internet reaproveitando as camadas que eles mesmos criaram quando da construção da aplicação para MS-Windows.

Na aplicação MS-Windows os alunos precisaram construir uma camada para realizar o acesso ao banco de dados e uma camada para contemplar determinadas regras de negócio (Exemplo: na inclusão de cliente o campo nome não pode ser vazio).

Quando desenvolveram a página Web, os alunos verificaram que foi possível reutilizar estas duas camadas. Dessa forma não foi necessário reescrever as partes do *software* responsáveis por estas tarefas pois isso já tinha sido feito anteriormente.

No segundo capítulo vimos que em 2007 foi criado um escritório brasileiro do W3C (*World Wide Web Consortium*). Isso demonstra a importância de padrões pois a missão do W3C é assegurar a compatibilidade entre as diferentes novas tecnologias incorporadas à Web. Existe um grande grupo de pessoas dedicadas nessa missão em todo o mundo, e isso é o que possibilita a alta produtividade e reaproveitamento de código que os alunos do CEFET puderam vivenciar no projeto LOJA .NET no momento em que construíram uma aplicação para a Web.

8) Como funciona na prática o uso da Segurança da Informação



Verificamos que 78% dos alunos afirmaram ter sido positivo o uso do projeto LOJA .NET para a assimilação de como usar tópicos de Segurança da Informação no processo de construção de *software*.

No capítulo 2, vimos na seção "Padrões no Desenvolvimento de Software" que existem três perfis de usuários quando o assunto é o uso da Internet: Usuário leigo, Usuário do governo e Usuário técnico.

O usuário técnico conhece como a Internet foi construída fisicamente e quais mecanismos estabelecem a conexão entre os computadores e demais dispositivos da rede.

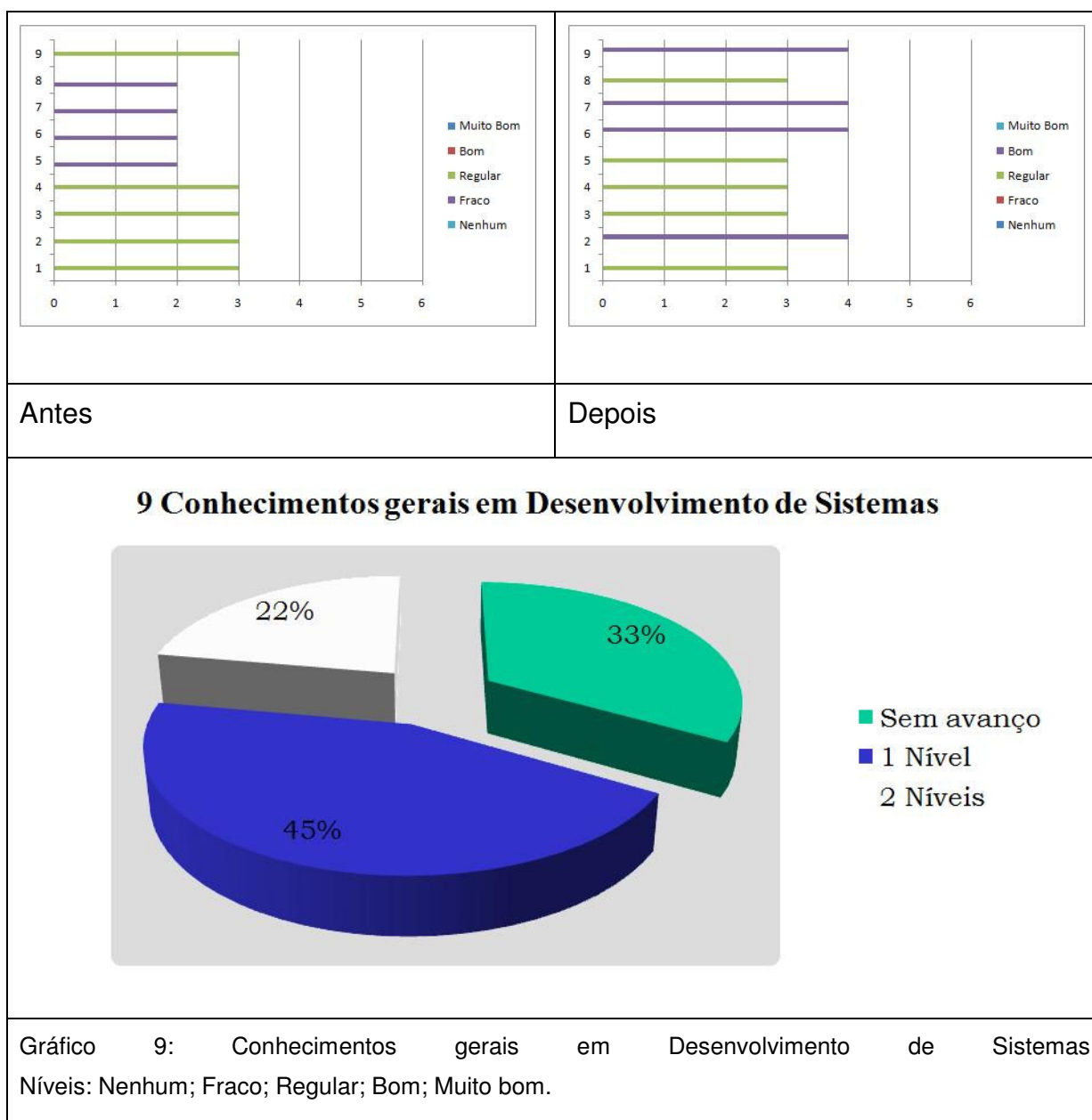
Infelizmente existem no mundo pessoas boas e pessoas más. Quando uma pessoa má é um usuário técnico isso se torna um risco às empresas e à sociedade, pois na Internet realizamos compras, acessamos nossa conta bancária para realizar pagamentos e usufruímos de diversos serviços disponíveis.

No livro "Desenvolvimento em Camadas com C# .NET" os alunos do CEFET compreenderam que o desenvolvedor precisa conhecer mecanismos para tornar o seu *software* seguro contra pessoas mal intencionadas.

No projeto LOJA .NET utilizado no livro, os alunos utilizaram técnicas para esconder informações referentes a localização do computador que armazena os dados acessados pelo *software* que desenvolveram. Isso inibe a ação de pessoas que tentarem invadir esse computador para roubar ou destruir informações.

Os alunos compreenderam que criar um *software* seguro é uma exigência de empresas de pequeno, médio e grande porte.

9) De uma forma geral, como você avalia o seu conhecimento em Desenvolvimento de Aplicações comerciais que atendam as exigências de qualidade das empresas?



67% dos alunos afirmaram que o projeto LOJA .NET contribuiu positivamente para a aquisição de conhecimento sobre desenvolvimento de aplicações utilizando a metodologia de desenvolvimento em camadas.

Verificamos neste terceiro capítulo que os alunos do CEFET vivenciaram através de projeto LOJA .NET todas as fases de desenvolvimento de um *software*.

O *software* desenvolvido pelos alunos atende as exigências atuais de mercado pois usa uma metodologia de desenvolvimento - a metodologia de Desenvolvimento em Camadas - assim como possui mecanismos de segurança de maneira a se prevenir contra tentativas de invasões.

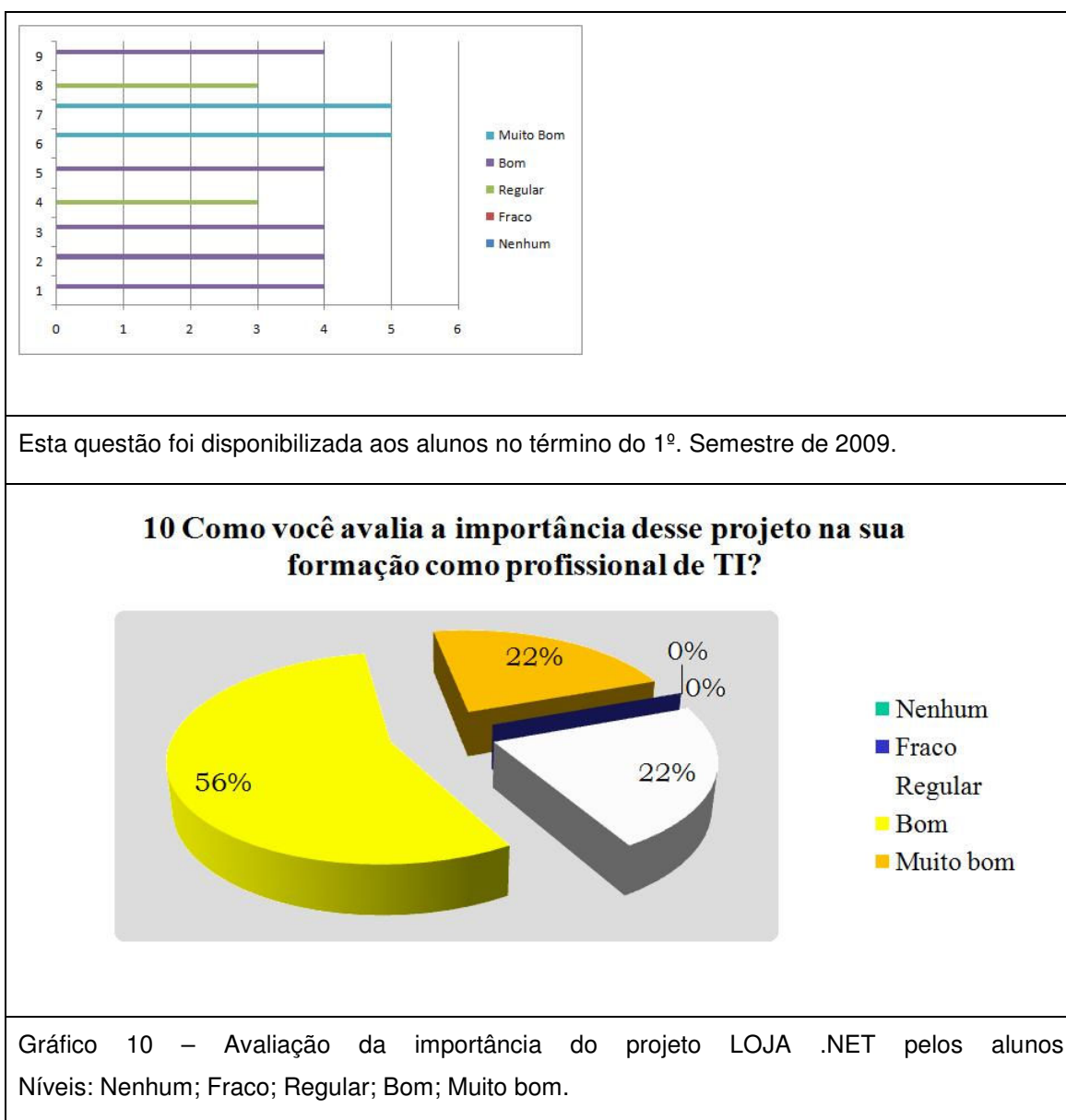
Analizamos neste gráfico como os alunos se sentiam preparados para desenvolver um *software* com todas as características exigidas pelo mercado antes e após executarem o projeto LOJA .NET.

Neste capítulo vimos ainda que durante a execução do projeto LOJA .NET houve uma constante mutação da zona de desenvolvimento proximal dos alunos.

Inicialmente havia muitos tópicos que eles desconheciam. Alguns estavam próximos de se tornarem conhecimento real, pois os alunos conseguiam executar determinadas construções com a ajuda de alunos mais avançados. Depois de algumas semanas, estes alunos já dominavam e conseguiam executar sozinhos o que anteriormente só conseguiam fazer com a ajuda de alunos mais avançados. Neste assunto que estes alunos que acabaram de conhecer atuavam como multiplicadores, ajudando os que ainda não haviam conquistado esse patamar.

Tivemos a oportunidade de explorar esse ciclo do “aprender a fazer” à luz da teoria de Vygotsky, que nos deu suporte na afirmação de que o ser humano está sempre aprendendo coisas novas, e o que uma pessoa hoje só faz com a ajuda de alguém será possível ser realizado de forma independente no futuro.

10) Como você avalia a importância desse projeto na sua formação como profissional de tecnologia da informação?



Essa questão foi disponibilizada aos alunos somente no questionário final, momento em que eles já tinham terminado a execução do projeto LOJA .NET em Laboratório de Informática.

Verificamos que 22% dos alunos consideraram o projeto regular, 56% afirmaram que a importância do projeto na sua formação foi boa e 22% dos alunos participantes do projeto disseram que o projeto foi muito bom para a sua formação.

Foi importante verificar como os alunos avaliavam a relevância do projeto LOJA .NET em sua formação, pois o objetivo desse projeto foi propiciar à eles experiência em desenvolvimento de *software* de forma a agregar habilidades e competências úteis na sua atuação profissional.

CONSIDERAÇÕES FINAIS

Nossa investigação consistiu em analisar a questão: A utilização de projetos de aplicação dos conceitos de forma prática fundamentada no método de “Desenvolvimento em Camadas” contribui para que alunos de cursos técnicos em informática desenvolvam competências na construção de *software*?

No capítulo inicial denominado “Construção de Software e as Competências Necessárias” foi possível analisar primeiramente o que vem a ser um projeto de aplicação. Foi criado um projeto de aplicação chamado LOJA .NET utilizado como material didático para que os alunos do CEFET pudessem ter um contato aprofundado com um Ambiente de Desenvolvimento Integrado – o Microsoft Visual Studio 2008. Analisamos a diferença entre habilidade e competência e verificamos que o profissional de desenvolvimento de *software* deve possuir não só habilidades técnicas, mas também é exigido com relação a aspectos comportamentais. Mapeamos cinco importantes competências e estas foram analisadas junto ao grupo de alunos no CEFET.

Para possibilitar a experimentação de um projeto prático transdisciplinar que permitisse aos alunos compreender como diversas disciplinas de seu currículo estão relacionadas no processo de construção de *software*, selecionamos um dos padrões de desenvolvimento de sistemas bastante utilizado atualmente: o método de desenvolvimento em camadas.

Era preciso explicitar a importância do uso de padrões na área de desenvolvimento de *software* e em que consistia o Desenvolvimento em Camadas. Para essa análise o segundo capítulo propiciou conhecer como a Internet surgiu. Essa história retrata a importância do uso de padrões e traz de forma objetiva o conceito de desenvolvimento em camadas. Verificamos que a Internet surgiu como uma interconexão de muitas redes de computadores. Para permitir essa comunicação entre diferentes computadores com diferentes sistemas operacionais foi imprescindível a adoção de padrões. Dividida em camadas, verificamos que na arquitetura de construção da Internet uma camada é dependente da outra e utiliza todos os seus serviços, reaproveitando-os de forma que estes serviços não precisam ser criados em

duplicidade. Esse modo de construção da Internet permite que ela evolua, pois a criação de novos serviços pode ocorrer sem a necessidade de grandes modificações no que já está sendo utilizado pelos usuários da Internet atualmente.

Através de uma comparação, vimos que o conceito de reaproveitar o que já está em funcionamento também é aplicado no método de desenvolvimento em camadas. Técnica que foi utilizada pelos alunos durante a execução do projeto LOJA .NET.

O capítulo final contribuiu para uma análise qualitativa pois com o projeto LOJA .NET os alunos: Utilizaram um projeto de aplicação para construir um *software*; Conheceram e utilizaram o método de desenvolvimento em camadas e também exercitaram cinco importantes competências mapeadas no capítulo inicial.

Através de questionários aplicados antes e após a execução do projeto LOJA .NET pelos alunos, no terceiro capítulo foi possível demonstrar graficamente a evolução dos alunos nos conhecimentos envolvidos.

Foi possível verificar que o projeto não contribuiu satisfatoriamente com relação à assimilação do conceito de Integridade Referencial, habilidade importante para a competência de Configuração e Gerenciamento de informações em Banco de Dados.

Com a análise realizada no capítulo final verificamos que de um modo geral, o projeto LOJA .NET mostrou-se relevante para que os alunos desenvolvessem competências sobre desenvolvimento de *software*. Vale ressaltar que apesar do livro "Desenvolvimento em Camadas com C# .NET" ser um guia passo a passo sem prévias explanações teóricas em cada capítulo, o projeto LOJA .NET existente nessa obra foi concebido com foco no uso de padrões e metodologias de desenvolvimento, e não no uso de tecnologias. Em outras palavras, pode-se afirmar que devido ao nível de abstração existente no projeto LOJA .NET, ao invés de utilizarmos como sistema operacional, linguagem de programação e sistema gerenciador de banco de dados a tríade Microsoft Windows, Microsoft C# .NET e Microsoft SQL Server, poderíamos ter utilizado, respectivamente, Linux, Java e MySQL.

Longe de esgotar as possibilidades, esperamos num futuro próximo continuar esta pesquisa, em busca de novas estratégias para auxiliar a formação do profissional de tecnologia da informação na área de desenvolvimento de *software*.

Por fim, esperamos que este trabalho possa constituir uma ferramenta que auxilie o futuro pesquisador em sua própria jornada.

ESTUDOS FUTUROS

Se por um lado a interação dos alunos com o projeto LOJA .NET nos permitiu navegar nesse mar de tecnologia, por outro agora é possível vislumbrar novas possibilidades.

Os alunos do CEFET participantes deste trabalho compreenderam, dentre outros tópicos, como as informações são passadas de uma camada para outra utilizando a programação orientada a objetos. Esse conhecimento constitui uma boa base e com isso talvez seja possível avançar em um tópico de padrões de desenvolvimento conhecido por *Test-Driven Development* ou TDD. O TDD é uma metodologia aplicada no desenvolvimento de *software* onde primeiro se desenvolve os testes e depois as regras de negócio são implementadas, ou seja, trata-se de um desenvolvimento dirigido por testes.

O uso do TDD permite eliminar os defeitos do código no início do processo, evitando longos períodos de revisão técnica do projeto. Essa técnica oferece um grande nível de confiança no código uma vez que um grande número de testes ajuda a limitar a quantidade de defeitos.

O pesquisador que se identificar com essa pesquisa científica talvez possa encontrar apoio em ferramentas como o NUnit da IDE Microsoft Visual Studio e também o JUNIT existente na IDE Eclipse. Atualmente estas IDEs já oferecem suporte ao conceito de TDD. Se empresas, profissionais de desenvolvimento e a comunidade científica utilizarem com mais frequência tais ferramentas a tendência é que elas evoluam.

Outra possibilidade de estudo para dar continuidade ao projeto LOJA .NET é explorar o uso do *Entity Framework*. No projeto LOJA .NET utilizamos na camada de acesso a dados comandos específicos do sistema de banco de dados Microsoft SQL Server. E se o projeto LOJA .NET fosse um sistema comercial e surgisse a necessidade de substituir o sistema de banco de dados Microsoft SQL Server para outro sistema de banco de dados qualquer, por exemplo o Oracle? Isso não seria uma tarefa simples, pois o sistema LOJA .NET possui acoplamento ao SQL Server. Significa que ele é dependente, contém comandos específicos para o gerenciamento do SQL Server. O uso do *Entity Framework* permite o desacoplamento do sistema de banco de dados, ou

seja, o desenvolvedor de *software* não precisará reescrever a camada de acesso a dados quando houver a necessidade de substituir o sistema de banco de dados utilizado pelo software que construiu.

GLOSSÁRIO

Browser. É um programa de computador que habilita seus usuários a interagirem com documentos virtuais da Internet, também conhecidos como páginas HTML, que estão hospedadas em um servidor Web.

Código fonte (código-fonte, ou *source code* em inglês) é o conjunto de palavras ou símbolos escritos de forma ordenada, contendo instruções em uma das linguagens de programação existentes, de maneira lógica. Existem linguagens que são compiladas e as que são interpretadas. As linguagens compiladas, após ser compilado o código fonte, transformam-se em *software*, ou seja, programas executáveis. Este conjunto de palavras que formam linhas de comandos severa estar dentro da padronização da linguagem escolhida, obedecendo critérios de execução. Atualmente, com diversificação de linguagens, o código pode ser escrito de forma totalmente modular, podendo um mesmo conjunto de códigos ser compartilhado por diversos programas e, até mesmo, linguagens.

Cracker. Pessoa que possui conhecimentos avançados de computação e usa esse conhecimento para invadir e destruir sistemas alheios.

Desenvolvimento em Camadas. Conceito da área de Engenharia de Sistemas. É uma metodologia que permite a construção de um *software* em partes separadas; estas partes são chamadas de camadas. Nesta metodologia de desenvolvimento, os programas ficam logicamente separados de maneira a facilitar a manutenção e a reutilização dos programas já desenvolvidos.

E-mail. (*Electronic mail*: mensagem eletrônica). É um método que permite compor, enviar e receber mensagens através de sistemas eletrônicos de comunicação.

Engenharia de *software*. É uma área do conhecimento da computação voltada para a especificação, desenvolvimento e manutenção de sistemas de *software* aplicando tecnologias e práticas de gerência de projetos e outras disciplinas, objetivando organização, produtividade e qualidade. Atualmente, essas tecnologias e práticas englobam linguagens de programação, banco de dados, ferramentas, plataformas, bibliotecas, padrões, processos e a questão da Qualidade de *Software*.

Os fundamentos científicos para a engenharia de *software* envolvem o uso de modelos abstratos e precisos que permitem ao engenheiro especificar, projetar, implementar e manter sistemas de *software*, avaliando e garantindo suas qualidades.

Engenharia reversa. Consiste em usar a criatividade para, a partir de uma solução pronta, retirar todos os possíveis conceitos novos ali empregados.

É um processo de análise de um artefato (um aparelho, um componente elétrico, um programa de computador, etc.) e dos detalhes de seu funcionamento, geralmente com a intenção de construir um novo aparelho ou programa que faça a mesma coisa, sem realmente copiar alguma coisa do original.

Entity Framework (EF). É um conjunto de tecnologias desenvolvido pela Microsoft que suporta o desenvolvimento a aplicações orientadas a dados. Arquitetos e desenvolvedores de aplicações orientadas a dados tem se debatido com a necessidade de alcançar dois objetivos distintos: Modelar entidades, relacionamentos e problemas da lógica de negócio; Trabalhar com sistemas de armazenamento de dados para armazenar e consultar informações. Os dados podem estar distribuídos em múltiplos sistemas de armazenamentos, cada um com suas particularidades e protocolos. Mesmo em situações onde um único sistema de armazenamento é usado ainda existe a necessidade de balancear os requisitos de armazenamento com os requisitos de escrever um código eficiente. O Entity Framework permite aos desenvolvedores trabalhar com dados na forma de propriedades e objetos específicos do domínio (como clientes, produtos) sem a obrigação de relacioná-los com as tabelas do banco de dados e as colunas onde os dados estão armazenados. Isto é possível pela elevação do nível de abstração no qual os desenvolvedores podem trabalhar quando estão tratando com os dados e pela redução do código que é necessário para manter as aplicações orientadas a dados.

Fluxograma. É um tipo de diagrama, e pode ser entendido como uma representação esquemática de um processo, muitas vezes feito através de gráficos que ilustram de forma descomplicada a transição de informações entre os elementos que o compõem. Podemos entendê-lo, na prática, como a documentação dos passos necessários para a execução de um processo

qualquer. É uma das Sete Ferramentas da Qualidade. Muito utilizada em fábricas e indústrias para a organização de produtos e processos.

O Diagrama de Fluxo de Dados (DFD) utiliza do Fluxograma para modelagem e documentação de sistemas computacionais.

GPS. O sistema de posicionamento global, popularmente conhecido por GPS (acrônimo do original inglês *Global Positioning System*, ou do português "geo-posicionamento por satélite") é um sistema de navegação por satélite que fornece a um aparelho receptor móvel a posição do mesmo, assim como informação horária, sob quaisquer condições atmosféricas, a qualquer momento e em qualquer lugar na Terra, desde que o receptor se encontre no campo de visão de quatro satélites GPS.

Hacker. Originário do inglês, o termo hacker é utilizado no português. Os Hackers utilizam todo o seu conhecimento para melhorar *software* de forma legal. A maioria dos hackers são usuários avançados de *Software* Livre como o sistema operacional GNU/Linux.

HTML. (acrônimo para a expressão inglesa *HyperText Markup Language*, que significa Linguagem de Marcação de Hipertexto) é uma linguagem de marcação utilizada para produzir páginas na Web. Documentos HTML podem ser interpretados por navegadores (ou *browsers*).

Interoperabilidade. É a capacidade de um sistema (informatizado ou não) de se comunicar de forma transparente (ou o mais próximo disso) com outro sistema (semelhante ou não). Para um sistema ser considerado interoperável, é muito importante que ele trabalhe com padrões abertos. Seja um sistema de portal, seja um sistema educacional ou ainda um sistema de comércio eletrônico, ou *e-commerce*, hoje em dia se caminha cada vez mais para a criação de padrões para sistemas.

Internet. Também chamada de WWW ou *World Wide Web*. É a definição para a rede mundial de computadores.

MILNET. Rede de computadores interligados. Foi criada pelos militares do Departamento de Defesa dos EUA.

N Camadas. Um programa de n camadas é um aplicativo desenvolvido de forma a ter várias camadas lógicas. Cada camada é auto-contida o suficiente de forma que a aplicação pode ser dividida em vários computadores em uma rede distribuída.

A forma mais comum da arquitetura é a aplicação em três camadas (3-Tier), comumente usada em aplicações Web, na qual as camadas são: interface com o usuário, lógica do negócio, e banco de dados.

Cada camada dessa arquitetura é normalmente mantida por um servidor específico para tornar-se mais escalonável e independente das demais. Essa arquitetura tem como características: baixos custos de disponibilização; baixos custos na mudança da base de dados; baixos custos na mudança da lógica de negócios; eficiente armazenamento e reutilização de recursos.

Servidor Web. A expressão servidor web pode significar duas coisas:

1. Um programa de computador responsável por aceitar pedidos http de clientes, geralmente os navegadores, e servi-los com respostas http, incluindo opcionalmente dados, que geralmente são páginas web, tais como documentos HTML com objetos embutidos (imagens, etc.);
2. Um computador que executa um programa e provê a funcionalidade descrita acima.

Software. É uma seqüência de instruções a serem executadas por um computador. *Software* também é o nome dado ao conjunto de produtos desenvolvidos no processo de *software*, onde incluem-se programas, manual de instalação, manual do usuário, dentre outros.

Stored procedure ou Procedimento armazenado é uma coleção de comandos em SQL para serem executados no Banco de dados. Normalmente é usado para tarefas que se repetem em um sistema. O procedimento armazenado aceita parâmetros de entrada e retorna um valor de status que indica se a coleção de comandos foi ou não executada com sucesso. O procedimento armazenado também permite reduzir o tráfego na rede, melhorar a performance e criar mecanismos de segurança.

TCP-IP. É um conjunto de protocolos de comunicação entre computadores em rede (também chamado de pilha de protocolos TCP/IP). Seu nome vem de dois protocolos: o TCP (*Transmission Control Protocol* – Protocolo de Controle de Transmissão) e o IP (*Internet Protocol* – Protocolo de Interconexão). O conjunto de protocolos pode ser visto como um modelo de camadas, onde cada camada é responsável por um grupo de tarefas, fornecendo um conjunto de serviços bem definidos para o protocolo da camada superior. As camadas mais altas estão logicamente mais perto do usuário (chamada camada de aplicação) e

lidam com dados mais abstratos, confiando em protocolos de camadas mais baixas para tarefas de menor nível de abstração.

W3C. O *World Wide Web Consortium* é um consórcio de empresas de tecnologia, atualmente com cerca de 500 membros. Fundado por Tim Berners-Lee em 1994 para levar a Web ao seu potencial máximo, por meio do desenvolvimento de protocolos comuns e fóruns abertos que promovem sua evolução e asseguram a sua interoperabilidade. O W3C desenvolve padrões para a criação e a interpretação dos conteúdos para a Web. Sites desenvolvidos segundo esses padrões podem ser acessados e visualizados por qualquer pessoa ou tecnologia, independente de hardware ou *software* utilizados, como celulares (em Portugal, telemóvel), PDAs, de maneira rápida e compatível com os novos padrões e tecnologias que possam surgir com a evolução da internet.

ABREVIATURAS

CEFET. Centro Federal de Educação Profissional e Tecnológica.

IDE. (*Integrated Development Environment*). Ambiente integrado de desenvolvimento.

IFSP. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.

NFS. Fundação Nacional de Ciência dos EUA.

SETEC. Secretaria de Educação Profissional e Tecnológica.

TCP-IP. *Transmission Control Protocol –Internet Protocol*.

W3C. *World Wide Web Consortium*.

ANEXO:
Questionário para o(a) Aluno(a)

IFET Instituto Federal de Educação, Ciência e Tecnologia

Carta de Apresentação

Prezado(a) Aluno(a),

Meu nome é Carlos Olavo de Azevedo Camacho Júnior. Sou um professor pesquisador e atualmente realizo Mestrado em Tecnologias da Inteligência e Design Digital na Pontifícia Universidade Católica de São Paulo PUC-SP.

O objetivo da minha pesquisa é verificar se o uso de projetos em laboratórios é capaz de transmitir conhecimentos necessários para a atuação do profissional de Tecnologia da Informação nas empresas.

No questionário, cada item de conhecimento está descrito em uma linha. Cada linha deve ter somente uma resposta (marque um “X” na coluna desejada). Escolha dentre as opções que vão de “nenhum” até “muito bom”, indicando o seu nível de conhecimento em cada item. A primeira página do questionário refere-se a avaliação dos seus conhecimentos antes da utilização do projeto proposto e a última página deve refletir como estão os seus conhecimentos após a execução do projeto em laboratório de informática.

Após o preenchimento, por favor devolva o questionário para o professor Nelson Paz.

Muito obrigado por participar dessa pesquisa! Você está contribuindo para a melhoria das técnicas de ensino e a formação de muitos profissionais da área de Tecnologia da Informação.

Carlos Camacho.

webmaster@camachojunior.com.br

IFET Instituto Federal de Educação, Ciência e Tecnologia

Questionário para o(a) Aluno(a)

Tema: Uso de Projeto no Desenvolvimento em Camadas com C# .NET

1. Antes da utilização do projeto prático em laboratório de informática:

Como você avalia os seus conhecimentos em:

<i>Conhecimento</i>	<i>Nenhum</i>	<i>Fraco</i>	<i>Regular</i>	<i>Bom</i>	<i>Muito Bom</i>
Programação Orientada a Objetos (Uso de classes, métodos e propriedades)					
Conceito de Desenvolvimento em Camadas					
Conceito de Integridade Referencial em Banco de Dados					
Criação de Tabelas em Banco de Dados					
Criação e uso de Stored Procedures					
Funcionamento da construção de aplicações para MS-Windows					
Funcionamento da construção de aplicações para a WEB					
Como funciona na prática o uso da Segurança da Informação					
De uma forma geral, como você avalia o seu conhecimento em Desenvolvimento de Aplicações comerciais que atendam as exigências de qualidade das empresas?					

2. Depois da utilização do projeto prático em laboratório de informática:

Como você avalia os seus conhecimentos em:

<i>Conhecimento</i>	<i>Nenhum</i>	<i>Fraco</i>	<i>Regular</i>	<i>Bom</i>	<i>Muito Bom</i>
Programação Orientada a Objetos (Uso de classes, métodos e propriedades)					
Conceito de Desenvolvimento em Camadas					
Conceito de Integridade Referencial em Banco de Dados					
Criação de Tabelas em Banco de Dados					
Criação e uso de Stored Procedures					
Funcionamento da construção de aplicações para MS-Windows					
Funcionamento da construção de aplicações para a WEB					
Como funciona na prática o uso da Segurança da Informação					
De uma forma geral, como você avalia o seu conhecimento em Desenvolvimento de Aplicações comerciais que atendam as exigências de qualidade das empresas?					
Como você avalia a importância desse projeto na sua formação como profissional de tecnologia da informação?					
Observações (opcional):					

REFERÊNCIAS BIBLIOGRÁFICAS

CAMACHO JÚNIOR, Carlos Olavo de Azevedo. **Desenvolvimento em camadas com C# .Net**. Florianópolis: Visual Books, 2008.

CARAÇA, João. **O que é CIÊNCIA**. 2ª edição, Coimbra: Quimera Editores, 2001.

CBO/MTE, 2002 BRASIL. Portaria Nº 397, 09 de outubro de 2002. Brasília: Ministério do Trabalho e Emprego, CBO – **Classificação Brasileira de Ocupações**, 2002.

CGI.br, **Revista .br** Edição 01 Ano 1, 2009.

CHAVES, Mário. **Complexidade e Transdisciplinaridade, Uma abordagem multidimensional do Setor da Saúde**, Brasil, 1998.

COAD, Peter; YOURDON, Edward. **Análise baseada em objetos**. Rio de Janeiro, Campus, 1992.

_____. **Projeto baseado em objetos**. Rio de Janeiro, Campus, 1993.

COLEMAN, Derek; ARNOLD, Patrick; BODOF, Stephanie; DOLLIN, Chris; GILCHRIST, Helena; HAYES, Fiona; JEREMAES, Paul. **Desenvolvimento orientado a objetos – O método Fusion**. Rio de Janeiro, Campus, 1996.

EMBLEY, David W.; JACKSON, Robert B.; WOODFIELD, Scott N. **OO Systems Analysis: Is it or isn't it?** IEEE *Software*. 1995.

GUNDERLOY, Mike. **Developing and Implementing Windows-Based Application with Microsoft Visual Basic .NET and Microsoft Visual Studio .NET**. MCAD / MCSD Training Guide Exam (70-306). Indianapolis, USA: Publishing, 2003, 1152p.

HECKHAUSEN, Heinz. **Discipline et interdisciplinarité**. In Ceri (eds.) L'interdisciplinarité. Problèmes d'enseignement et de recherche dans les Universités. Paris: UNESCO/OCDE, 1972.

JANSTCH, Erich. **Vers l'interdisciplinarité et la transdisciplinarité dans l'enseignement et l'innovation**. In Ceri (eds.) L'interdisciplinarité. Problèmes d'enseignement et de recherche dans les Universités, Paris: UNESCO/OCDE, 1972.

LE BOTERF, Guy. **De la compétence: essai sur un attacteur étrange**. Paris: s.n., 1994.

MARTIN, James; ODELL, James J. **Análise e Projeto Orientado a Objetos**. São Paulo, Makron Books, 1995.

MASLOW, Abraham H. **Maslow no Gerenciamento**. RJ: Qualitymark, 2000. 392p.

MICROSOFT Corporation. **Developing Microsoft ASP .NET Web Applications Using Visual Studio .NET**. MSDN Training. Redmond, USA: Microsoft Press, 2002. 794 p.

MICROSOFT Corporation. **Programming with C#**. MSDN Training. Redmond, USA: Microsoft Press, 2002. 739p.

MORIN, Edgar. **Introdução ao pensamento complexo** (Título original: Introduction à la pensée complexe. Paris, ESF éd., 1990), 2001.

NEST. **Introdução à Metodologia Transdisciplinar**, Núcleo de Estudos Superiores Transdisciplinares (Gustavo Korte e Dalva Alves), São Paulo, 2000.

NICOLESCU, Bassarab. **Um novo tipo de conhecimento - transdisciplinaridade**. In Educação e Transdisciplinaridade. Brasília: Ed. UNESCO, 1999.

IBGE, **O Setor da Tecnologia da Informação e Comunicação no Brasil 2003-2006**. Estudos e Pesquisas – Informação Econômica n. 11: Rio de Janeiro, 2009.

Notas de Aula do Prof. Demi Getschko. Disciplina: **Redes e Tecnologia em Gestão do Conhecimento**. Programa de Pós-Graduação em Tecnologias da Inteligência e Design Digital, PUC-SP, 1º Semestre/2009.

OLIVEIRA, Martha Kohl de. **Vygotsky: aprendizado e desenvolvimento: um processo sócio-histórico**. São Paulo, Scipione, 1997. (Pensamento e ação no magistério)

PARRA, Cecilia. **Didática da Matemática: reflexões psicopedagógicas** / Cecilia Parra, Irma Saiz et.al.; tradução Juan Acuña Llorens. Porto Alegre: Artes Médicas, 1996.

PARSONS, Jeffrey; WAND, Yair. **Using objects for System Analysis**. Communication of ACM vol. 40, 1997.

PERRENOUD, Philippe. **Construir as competências desde a escola**. Porto Alegre: Artmed, 1999.

PRADO, Fernando Leme do. **Complexidade e metodologia de projetos: melhorando a prática docente em cursos de graduação tecnológica**. PUC-SP, 2007. Doutorado em Educação e Currículo.

PRESSMAN, Roger. **Engenharia de Software**. São Paulo, Makron, 1995.

Internet

CEFET – SP. Documentação e relatórios. <http://www.cefetsp.br> acessado em 05/06/2009.

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP. <http://www.ifsp.edu.br> acessado em 14/02/2011.

Martin Fowler. <http://martinfowler.com/articles/newMethodology.html> The New Methodology - acessado em 14/12/2010.

Project Management Institute. <http://www.pmi.org.br> acessado em 02/05/2010.

World Wide Web Consortium – **W3C**. Documentação e relatórios. <http://www.w3c.br> acessado em 04/03/2011.