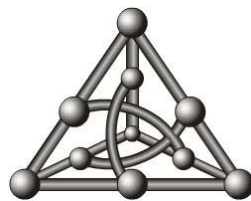


**FUNDAÇÃO UNIVERSIDADE FEDERAL DE  
MATO GROSSO DO SUL**

**UMA ABORDAGEM DE INTEGRAÇÃO E  
EXPLORAÇÃO VISUAL DE DADOS EDUCACIONAIS  
NA PLATAFORMA WEB-PIDE**

JACKSON DIAS SAVITRAZ

Dissertação de Mestrado  
em Ciência da Computação



**Facom**  
**Faculdade de Computação**

**JACKSON DIAS SAVITRAZ**

**UMA ABORDAGEM DE INTEGRAÇÃO E  
EXPLORAÇÃO VISUAL DE DADOS EDUCACIONAIS  
NA PLATAFORMA WEB-PIDE**

Dissertação apresentada como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Curso de Mestrado em Ciência da Computação, Facom – Faculdade de Computação, Fundação Universidade Federal de Mato Grosso do Sul.

**Orientador: Prof. Dr. Marcelo Augusto Santos Turine**

**Co-orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Débora Maria Barroso Paiva**

**Junho 2010**

## **AGRADECIMENTOS**

À minha esposa Dandara e minha filha Désirée, que fazem meus dias mais felizes, bem como os meus pais Ivone e Nereu, responsáveis pela formação do meu caráter e por todo apoio durante minha jornada de aprendizado de vida.

Ao meu orientador Marcelo Turine, que gentilmente apresentou-me ao LEDES (Laboratório de Engenharia de Software), onde aprendi muito e fiz ótimas amizades.

A todos meus colegas de aula no mestrado, em especial, minhas grandes amigas Lucinéia e Neiva e meu ótimo amigo Péricles.

Ao Projeto Observatório de Educação (INEP/CAPES) e FUNDECT pelo apoio financeiro para realização do mestrado.

# UMA ABORDAGEM DE INTEGRAÇÃO E EXPLORAÇÃO VISUAL DE DADOS EDUCACIONAIS NA PLATAFORMA WEB-PIDE

## RESUMO

O Ministério da Educação (MEC) disponibiliza uma série de bases de dados e informações das diversas modalidades de avaliações educacionais, através do Instituto de Estudos e Pesquisas em Educação Anísio Teixeira (INEP). A produção de dados e informações estatístico-educacionais de forma ágil e que retrate a realidade do setor educacional é um dos instrumentos básicos de avaliação, planejamento e auxílio ao processo decisório para o estabelecimento de políticas de melhoria da educação brasileira. Assim, surge uma oportunidade de explorar métodos e técnicas a fim de propor um ambiente integrador e facilitador para o acesso dos indicadores das avaliações educacionais na Internet. O principal problema a ser investigado será definir uma camada de integração entre as diversas aplicações Web (WebApps) e disponibilizar de forma efetiva a visualização dos dados obtidos, onde sua representação visual auxilie na tomada de decisões e seja alcançada intuitivamente através de uma interface com o usuário. A especificação e a implementação de uma arquitetura para plataforma intitulada Web-PIDE (Plataforma aberta para Integração e avaliação de Dados Educacionais na Web) com requisitos de interoperabilidade que propicie um ambiente integrado de visualização e consulta a dados de avaliações educacionais é a principal motivação deste trabalho, pois os dados deste domínio possuem particularidades próprias, exigindo assim um novo enfoque ao problema. O objetivo principal deste trabalho é abordar o problema da integração entre as WebApps da plataforma Web-PIDE, de forma que se mantenha as interfaces de visualização de dados em avaliações educacionais em um único repositório de conhecimento (ambiente integrador). Para validar e testar a abordagem serão selecionadas três ferramentas de diferentes contextos: ferramentas de visualização de dados, datawarehouse e consultas avançadas. Toda a plataforma será baseada em tecnologias de Webservice e software livre.

**Palavras-chave:** Avaliações Educacionais, Ambiente de Integração, WebApps, Webservice, Visualização de Informações e Software Livre.

# **AN APPROACH FOR INTEGRATION AND VISUAL EXPLORATION FOR EDUCATIONAL DATA ON WEB-PIDE PLATFORM**

## **ABSTRACT**

The Ministry of Education (MEC) offers several databases and informations over many forms of educational assessment, through The Research and Studies Institute in Education Anísio Teixeira (INEP). The fast data production and statistical-educational information portraying the reality of the educational sector is one of the basic instruments of assessment, planning and assistance to decision-making process for the establishment of policies to improve Brazilian education. Thus, an opportunity shows up to explore methods and techniques in order to propose an easier integrating environment for accessing indicators of educational assessments on the Internet. The main problem to be investigated will be the definition of an integration layer between different Web applications (WebApps) and the providing of an effective way to visualize the acquired data, in such way that its visual representation is able to help the decision-making and can be intuitively reached through an user interface. Because the data of this domain is so specific, the main motivation of this work is the specification and implementation of an architecture for what is called Web-PIDE (Open Platform for The Integration and Evaluation of Web Educational Data), with interoperability requirements, providing an integrated environment for visualization and query of educational assessment data. The main goal of this work is to approach the integration problem between the WepApps of Web-PIDE platform, in order to keep all educational assessment data interfaces in a single knowledge storehouse (integrating environment). To validate and test this approach, three tools of different contexts will be selected: data visualization tools, data warehouse and advanced queries. The entire platform will be based on web service technologies and Open Source software.

**Key-words:** Educational Assessments, Integrating Environment, WebApps, Webservice, Visualization of Information and Open Source Software.

# Sumário

<b>1. INTRODUÇÃO .....</b>	<b>8</b>
1.1. CONTEXTUALIZAÇÃO .....	8
1.2. MOTIVAÇÕES.....	12
1.3. OBJETIVOS.....	15
1.4. ORGANIZAÇÃO DO TEXTO .....	16
<b>2. VISUALIZAÇÃO DE INFORMAÇÃO .....</b>	<b>17</b>
2.1. CONSIDERAÇÕES INICIAIS.....	17
2.2. TÉCNICAS DE VISUALIZAÇÃO .....	18
2.2.1 Técnicas de Visualização de Informações Tabulares e Gráficas.....	19
2.2.2 Técnicas de Visualização de Informações Geográficas .....	20
2.3. FERRAMENTAS DE VISUALIZAÇÃO DO INEP .....	22
2.4. CONSIDERAÇÕES FINAIS .....	29
<b>3. DESENVOLVIMENTO DE APLICAÇÕES WEB .....</b>	<b>30</b>
3.1. CONSIDERAÇÕES INICIAIS.....	30
3.2. ENGENHARIA DA WEB .....	31
3.3. DESENVOLVIMENTO BASEADO EM COMPONENTES.....	35
3.3.1 Análise de Domínio .....	35
3.3.2 Reuso e <i>Frameworks</i> .....	37
3.4. PROJETO DE INTERFACE DE WEBAPP.....	41
3.4.1 Técnicas de Interface com o Usuário .....	41
3.4.2 Padrões de Interface com o Usuário .....	44
3.5. CONSIDERAÇÕES FINAIS .....	45
<b>4. ABORDAGEM PARA INTEGRAÇÃO E VISUALIZAÇÃO DE DADOS .....</b>	<b>46</b>
4.1. CONSIDERAÇÕES INICIAIS.....	46
4.2. GESTÃO DE PROJETO .....	47
4.3. FUNCIONALIDADES.....	49
4.3.1 Padronização das bases de dados.....	50
4.3.2 Especificação, visualização e exportação dos dados consultados .....	52
4.3.3 Tecnologias adotadas.....	53
4.4. CAMADA DE INTEGRAÇÃO .....	53
4.4.1 Método <i>getDatabases</i> .....	56
4.4.2 Método <i>getColumnns</i> .....	57

4.4.3	Método <i>getData</i> .....	58
4.4.4	Método <i>getQuery</i> .....	59
4.4.5	Configuração do WebService.....	60
4.5.	CONSIDERAÇÕES FINAIS .....	60
<b>5. VALIDAÇÃO DA ABORDAGEM PROPOSTA .....</b>		<b>61</b>
5.1.	CONSIDERAÇÕES INICIAIS.....	61
5.2.	PROPOSTA DE VALIDAÇÃO .....	61
5.2.1	Interface da WebApp.....	62
5.3.	PROCESSO DE INSTANCIACÃO.....	67
5.4.	CONSIDERAÇÕES FINAIS .....	69
<b>6. CONCLUSÃO.....</b>		<b>70</b>
6.1.	CONSIDERAÇÕES INICIAIS.....	70
6.2.	CONTRIBUIÇÕES .....	71
6.3.	TRABALHOS FUTUROS .....	71
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>		<b>73</b>
<b>ANEXOS .....</b>		<b>77</b>

## Lista de Figuras e Tabelas

Figura 1.1: Arquitetura da Plataforma Web-PIDE [TCM+2006] .....	13
Figura 2.1: Modelo de referência para visualização de informações [CMS1999]. .....	18
Figura 2.2: Dados em representação gráfica [ALM2003]. .....	19
Figura 2.3: Representação vetorial de informações geográficas. ....	21
Figura 2.4: Sobreposição de camadas em mapas. ....	22
Figura 2.5: Interface principal do Consulta IDEB. ....	25
Figura 2.6: Interface principal do EDUDATABRASIL. ....	25
Figura 2.7: Interface principal do DataEscolaBrasil. ....	26
Figura 2.8: Exemplo do arquivo de entrada SAS para o SAEB. ....	27
Figura 2.9: Exemplo do arquivo de entrada SPSS para o SAEB. ....	27
Figura 2.10: Exemplo do arquivo de microdados para o SAEB. ....	27
Figura 2.11: Exemplo do arquivo de dicionário de dados do SAEB. ....	28
Figura 3.1: Camadas da Engenharia da Web [PRE2005]. ....	31
Figura 3.2: Modelo genérico de WebApp [RT2001]. ....	34
Figura 3.3: Engenharia de Domínio x Engenharia de Software. ....	36
Figura 3.4: Ciclo de vida de um <i>framework</i> [SIL2000]. ....	40
Figura 3.5: Funcionamento do padrão MVC. ....	45
Figura 4.1: Portal Web-PIDE. ....	47
Figura 4.2: Solução integrada e centralizada no Portal Web-PIDE. ....	48
Figura 4.3: Caso de uso das funcionalidades da abordagem. ....	49
Figura 4.4: Funcionamento da ferramenta DEAR [SCS2008]. ....	50
Figura 4.5: Definição de dicionários na DEAR [SCS2008]. ....	51
Figura 4.6: DW para base de dados do SAEB [MRC+2009]. ....	52
Figura 4.7: Modelo de requisição/resposta de Webservice. ....	54
Figura 4.8: Modelo em camadas da arquitetura de integração. ....	55
Figura 4.9: Fluxograma da lógica do repositório de consultas. ....	59
Figura 5.1: Janela para seleção de base de dados. ....	62
Figura 5.2: Estrutura da interface. ....	62
Figura 5.3: Especificação da consulta. ....	63
Figura 5.4: Janela de visualização da consulta. ....	64
Figura 5.5: Planilha de dados. ....	65
Figura 5.6: Gráfico comparativo em Linhas. ....	65
Figura 5.7: Gráfico comparativo em Barras. ....	66
Figura 5.8: Gráfico comparativo em Área. ....	66
Figura 5.9: Proporção geográfica. ....	67
Tabela 1.1: Bases de Dados do INEP (2010) .....	10
Tabela 2.1: Dados em representação tabular [ALM2003]. ....	19
Tabela 2.2: Ferramentas de Visualização do INEP. ....	23
Tabela 3.1: Taxonomia de reutilização [PAE1999]. ....	37
Tabela 3.2: Custos e benefícios no uso de <i>frameworks</i> [MGB+2002]. ....	40



# Capítulo 1

## Introdução

### 1.1. Contextualização

A produção de dados e informações estatístico-educacionais de forma ágil e de qualidade, que retrate a realidade do setor educacional, é o instrumento básico de avaliação, planejamento e auxílio ao processo decisório para o estabelecimento de políticas de melhoria da educação brasileira. É por meio dos censos educacionais que se busca garantir a utilização da informação estatística neste processo, gerando os indicadores necessários ao acompanhamento do setor educacional [JAN2001].

Neste contexto, o Ministério da Educação (MEC) disponibiliza, por meio do Instituto de Estudos e Pesquisas em Educação Anísio Teixeira (INEP), uma série de bases de dados e informações das diversas modalidades de avaliações educacionais do sistema educacional brasileiro. Após a transformação efetivada pela Lei nº 9.448, de 14 de março de 1997, o INEP teve papel estratégico único e essencial para o fortalecimento da gestão das políticas educacionais e o desenvolvimento da educação brasileira.

O INEP é responsável pelo levantamento e produção das estatísticas básicas da educação nacional, por meio da realização de levantamentos periódicos que abrangem os diferentes níveis e modalidades de ensino. A fim de ampliar o conhecimento sobre a realidade do sistema educacional brasileiro, o INEP desenvolve vários estudos sobre avaliações que possam servir de base para a elaboração de políticas públicas educacionais.

As avaliações aplicadas pelo INEP até o ano de 2010 podem ser mapeadas quanto ao domínio (Educação Básica ou Superior) ou sistema de avaliação. Deste modo, pode-se observar na Tabela 1.1 uma estrutura das avaliações mapeadas juntamente com outros dados importantes: nome de divulgação, ano de aplicação e situação.

O **Censo Escolar** (CE) com atualização anual dispõe de informações sobre as instituições escolares nos diversos níveis de ensino, como o número de matrículas, o volume de alunos, o movimento escolar, características básicas da instituição, equipamentos e edificações existentes, além de dados sobre o pessoal técnico e administrativo e as características dos docentes, entre outros.

O **Exame Nacional para Certificação de Competências de Jovens e Adultos** (ENCCEJA) é uma avaliação anual que mede as competências e habilidades de jovens e adultos, residentes no Brasil e no exterior, que estão concluindo o Ensino Fundamental e Médio. A adesão a esta avaliação é opcional e realizada pelas secretarias de Educação dos Estados, Distrito Federal e Municípios.

O **Sistema Nacional de Avaliação da Educação Básica** (SAEB), com periodicidade bianual, tem como objetivo principal contribuir para a melhoria da qualidade da educação brasileira e para a universalização do acesso à escola, oferecendo informações que permitem a realização de estudos sobre os fatores determinantes do desempenho escolar por meio de exames de proficiência de Português e Matemática com os alunos da primeira, quarta e oitava séries, os quais são solicitados também a responder um questionário acerca de suas condições socioeconômicas. São aplicados ainda questionários específicos para diretores e professores. É composto por dois processos que acontecem simultaneamente a cada dois anos: a **Avaliação Nacional da Educação Básica** (ANEB) e a **Avaliação Nacional do Rendimento Escolar** (ANRESC). A ANEB é realizada por amostragem das Redes de Ensino, em cada unidade da Federação e tem foco na gestão dos sistemas educacionais e por manter as mesmas características do SAEB, recebe o mesmo nome em suas divulgações. A ANRESC é mais extensa e detalhada que a ANEB, tendo foco em cada unidade escolar e por seu caráter universal, é divulgada com o nome de **Prova Brasil**.

O **Exame Nacional do Ensino Médio** (ENEM) tem como objetivos principais oferecer uma referência para que cada cidadão possa proceder a sua auto-avaliação às suas escolhas futuras, estruturar uma avaliação da educação básica que sirva como modalidade alternativa ou complementar aos processos de seleção nos diferentes setores

do mercado de trabalho e aos exames de acesso aos cursos profissionalizantes pós-médios e ao Ensino Superior.

**Tabela 1.1: Bases de Dados do INEP (2010)**

DOMÍNIO	Educação Básica							Educação Superior				
SISTEMA DE AVALIAÇÃO	SAEB Sistema Nacional de Avaliação da Educação Básica							SINAES Sistema Nacional de Avaliação da Educação Superior				
AVALIAÇÃO	CE ↓ Censo Escolar	ENCCEJA ↓ Exame Nacional para Certificação de Competências de Jovens e Adultos	ANEB ↓ Avaliação Nacional da Educação Básica	ANRESC ↓ Avaliação Nacional do Rendimento Escolar	Provinha Brasil ↓ Avaliação da Alfabetização	ENEM ↓ Exame Nacional do Ensino Médio	ENC ↓ Exame Nacional de Cursos	Censo da Educação Superior	Avaliação das Instituições de Educação Superior	Avaliação dos Cursos de Graduação	ENADE ↓ Exame Nacional de Desempenho de Estudantes	
NOME DE DIVULGAÇÃO	Censo Escolar	ENCCEJA	SAEB	Prova Brasil	Provinha Brasil	ENEM	Provão	Censo da Educação Superior	Avaliação das Instituições de Educação Superior	Avaliação dos Cursos de Graduação	ENADE	
ANO DE APLICAÇÃO	1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010	2002 2003 2004 2005 2006 2007 2008 2009 2010	1990 1991 1993 1995 1997 1999 2001 2003 2005 2007 2009 2010	2005 2007 2009 2010	1º Sem. 2008 2º Sem. 2008 1º Sem. 2009 2º Sem. 2009 1º Sem. 2010 2º Sem. 2010	1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010	1996 1997 1998 1999 2000 2001 2002 2003	1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010	2002 2003 2004 2005 2006 2007 2008 2009 2010	2003 2004 2005 2006 2007 2008 2009 2010	2004 2005 2006 2007 2008 2009 2010	
SITUAÇÃO	ATIVO	ATIVO	ATIVO	ATIVO	ATIVO	ATIVO	INATIVO	ATIVO	ATIVO	ATIVO	ATIVO	

O **Exame Nacional de Cursos (ENC)**, conhecido como **Provão**, foi aplicado aos formandos das instituições de ensino superior com objetivo de avaliar os cursos de graduação no que tange aos resultados do processo de ensino-aprendizagem. O Provão foi substituído pelo **Exame Nacional de Desempenho dos Estudantes (ENADE)**.

O **Censo da Educação Superior** é realizado por meio de um questionário eletrônico que as IES respondem e reúne informações sobre as instituições, seus cursos de graduação presencial ou a distância, cursos sequenciais, vagas oferecidas, inscrições, matrículas, ingressantes e concluintes, além de informações sobre docentes, nas diferentes formas de organização acadêmica e categoria administrativa.

O **Sistema Nacional de Avaliação da Educação Superior (SINAES)** é formado por três componentes principais: a Avaliação das Instituições de Ensino Superior, a Avaliação dos Cursos de Graduação e o Exame Nacional de Desempenho

dos Estudantes (ENADE). A Avaliação das Instituições de Ensino Superior é dividida em duas modalidades: a Auto-avaliação - coordenada pela Comissão Própria de Avaliação (CPA) de cada instituição e orientada pelas diretrizes e por um roteiro da auto-avaliação institucional; e a Avaliação externa – realizada por comissões designadas pelo INEP, que tem como referência os padrões de qualidade para a educação superior expressos nos instrumentos de avaliação e os relatórios das auto-avaliações. A Avaliação dos Cursos de Graduação prevê a avaliação periódica dos cursos de educação superior, passando por três tipos de avaliação: para autorização, para reconhecimento e para renovação de reconhecimento.

O **Exame Nacional de Desempenho dos Estudantes** (ENADE) avalia o rendimento dos alunos dos cursos de graduação, ingressantes e concluintes, em relação aos conteúdos programáticos dos cursos em que estão matriculados. O exame é obrigatório para os alunos selecionados e condição indispensável para a emissão do diploma.

Devido à preocupação em abranger todos os níveis e modalidades de ensino do país e atender a uma demanda cada vez mais específica por informações educacionais, o INEP realiza ocasionalmente os Levantamentos Especiais, e que dentre estes, destacam-se o Censo do Professor, o Censo da Educação Profissional, o Censo Escolar Indígena e o Censo da Educação Infantil.

Segundo Cerqueira e Rigotti [CR2001], as bases de dados educacionais, consideradas de maior importância são o Censo Escolar e o SAEB, devido à sua magnitude e abrangência – foco do presente trabalho.

Neste contexto, os indicadores extraídos das bases de dados de avaliações educacionais são instrumentos para controle da gestão e verificação de eficiência e eficácia da educação na administração pública e privada, pois permitem comparar situações em períodos e localidades diferentes no território brasileiro. A necessidade e a importância dos indicadores na administração pública justificam-se, dentre outros motivos, por aumentar a transparência da gestão da educação e facilitar o diálogo entre os mais diversos grupos sociais organizados.

Porém, há um grande volume de dados educacionais produzidos pelo INEP e não existe uma estrutura padrão para armazenar e distribuir tais dados, dificultando o acesso e a integração destes com outros sistemas de informação. Motivado por tais características, surge uma oportunidade de explorar métodos e técnicas a fim de propor

---

um ambiente facilitador para o acesso e visualização dos indicadores das avaliações educacionais na Internet.

## 1.2. Motivações

Como tornar os dados dos Sistemas de Avaliação do Brasil (INEP/MEC) mais acessíveis aos gestores de educação, bem como aos educadores e pesquisadores, a fim de melhorar a Educação Brasileira, facilitando a tomada de decisão na gestão pública a partir da recuperação e reutilização dos dados das avaliações educacionais, é a grande questão tecnológica atual.

O fato das bases de dados mantidas pelo INEP armazenarem um grande volume de informações e não terem uma estrutura padrão, as consultas computacionais tornam-se custosas, dificultando sua análise e impossibilitando uma tomada de decisão eficiente baseada nos dados das avaliações educacionais.

Neste contexto, a Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e o INEP iniciaram em 2007 o Programa Observatório da Educação (<http://observatorio.inep.gov.br>), que é uma iniciativa para fomentar o desenvolvimento de estudos e pesquisas em educação, com a finalidade de estimular a produção acadêmica e a formação de recursos em nível de mestrado e doutorado, em áreas voltadas à pesquisa da educação, por meio de financiamento específico, para consolidar e ampliar o pensamento crítico e estratégico para o desenvolvimento sustentável do País.

Em janeiro de 2007 foi aprovado pelo INEP/CAPES o projeto de pesquisa da Fundação Universidade Federal de Mato Grosso do Sul (UFMS) em parceria com a Universidade Federal de São Carlos (UFScar) intitulado "*Web-PIDE – Uma Plataforma Aberta para Integração e Avaliação de Dados Educacionais na Web*", como parte do Programa Observatório de Educação que objetiva ser um sistema computacional para integrar e disponibilizar os dados<sup>1</sup> educacionais do INEP por meio de uma linguagem comum e padronizada de marcação intitulada LIDE (Linguagem de Marcação de Dados Educacionais). As LIDEs são definidas com a linguagem de marcação XML (*eXtensible Markup Language*), que torna os documentos legíveis para as pessoas e manipuláveis pelos computadores. Para completar a camada responsável pelos dados na arquitetura

---

<sup>1</sup> Os termos "dado" e "informação" são conceitualmente diferentes, porém, para simplificar seu uso, no texto será utilizado somente o termo "dado".

Web-PIDE, o projeto utiliza a tecnologia de *Data Warehouse* (DW) e de *Data Marts* (DM) para sistematizar e armazenar os dados históricos com o objetivo de facilitar a tomada de decisão pelos gestores, como pode ser visto na Figura 1.1.

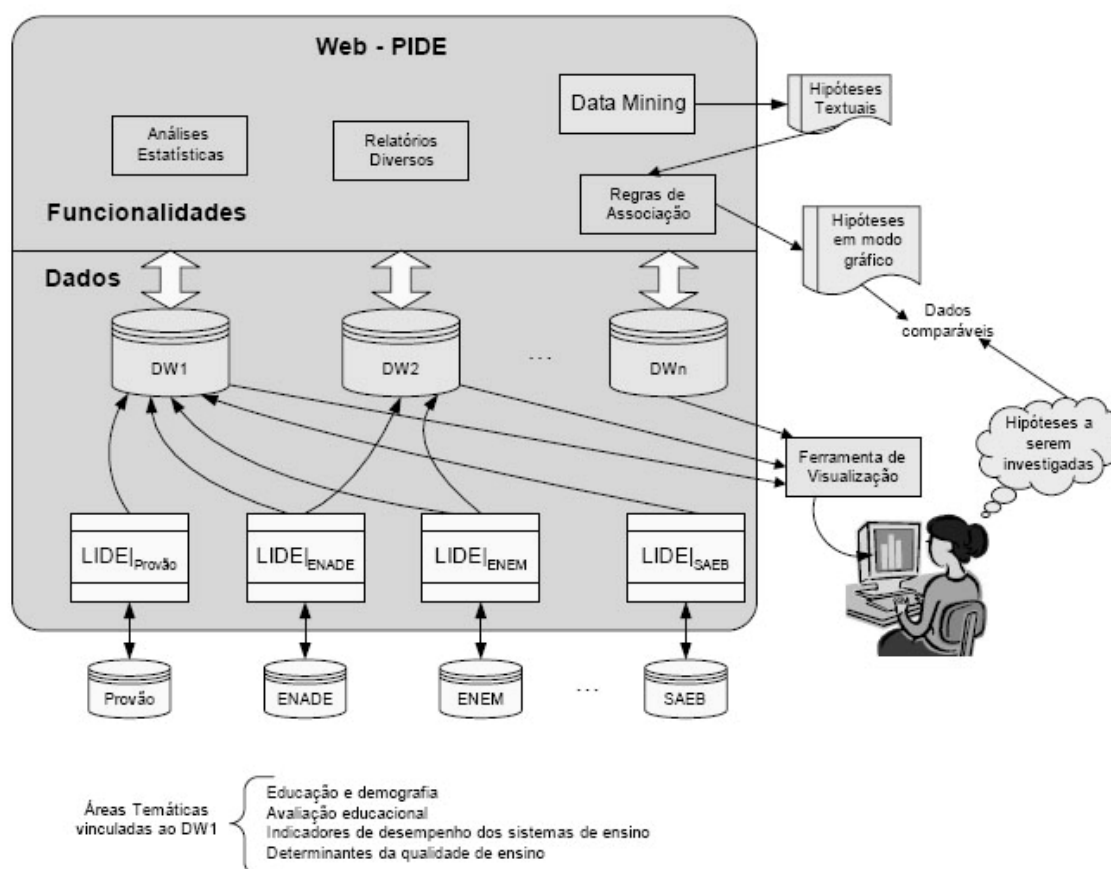


Figura 1.1: Arquitetura da Plataforma Web-PIDE [TCM+2006]

Considerando a necessidade de criação da LIDE, é necessário utilizar mecanismos para analisar os dados referentes às avaliações educacionais do INEP e identificar nesses dados quais questões são usadas, em quais anos e em quais questionários a fim de gerar os possíveis indicadores.

Para a criação da LIDE, considerando todas as atividades necessárias para sua criação, utiliza-se o P-LIDE – Processo para definição da Linguagem para Integração de Dados Educacionais, que tem na SEVTool (*Search and Edition in Visualization*) um apoio computacional para permitir que os dados referentes às avaliações do INEP sejam padronizados sintaticamente, sendo que o uso de visualização facilita essa tarefa. Ambas ferramentas foram desenvolvidas como produtos de uma dissertação de mestrado na UFScar [HER2009].

Outra contribuição para o projeto Web-PIDE, na forma de dissertação de mestrado, é a utilização do SB-INDEX para o projeto de um *Data Warehouse*

geográfico [SIQ2009], que provê mecanismo de indexação capaz de reduzir o tempo de resposta das consultas envolvendo grandes quantidades de dados, utilizando predicados espaciais. Neste mesmo trabalho foi proposto o modelo de representação de DW no esquema estrela.

Segundo Turine *et al.* [TOI+2004], as transformações que vêm ocorrendo no mundo inteiro e que, mesmo tardiamente, chegam à administração pública exigem das organizações maior agilidade e flexibilidade no atendimento das novas e crescentes demandas da sociedade. Este quadro, associado ao elevado grau de competição e um crescente desenvolvimento tecnológico, torna possível e necessária a utilização efetiva das tecnologias de informação e comunicação como ferramenta de produtividade do serviço público, de qualidade dos serviços prestados ao cidadão e de democratização do acesso às informações.

Atualmente o uso da tecnologia vem ganhando grande importância nos processos de tomada de decisão nas organizações que, cada vez mais, procuram agilidade e confiabilidade em seus processos. Diante desse contexto, as WebApps<sup>2</sup> se apresentam como uma excelente alternativa, pois permitem a disponibilização e o processamento eficiente de uma grande quantidade de informações, com um diferencial: a independência geográfica, em que um sistema gerencia informações de diferentes locais a um baixo custo.

A necessidade de extrair informações úteis, não triviais e previamente desconhecidas de bases de dados fez surgir a linha de pesquisa em Descoberta de Conhecimento em Bases de Dados (KDD - *Knowledge Discovery in Databases*) e a Mineração de Dados (*Data Mining*), fase mais importante do KDD, já que é nesta fase que informações úteis são efetivamente extraídas dos dados [FGW2002].

A Mineração de Dados pode utilizar várias abordagens e, dentre elas, utiliza-se neste trabalho a Exploração Visual de Dados que, segundo Almeida [ALM2003], tem se tornado cada vez mais importante, pois faz uso da enorme capacidade humana de interpretar representações visuais. Tal abordagem apresenta como um grande potencial no auxílio à extração de conhecimento de conjuntos de dados.

Assim, a implementação de um ambiente que possibilite a visualização, de forma fácil e concisa, dos diferentes indicadores fornecidos pelas WebApps integradas é

---

<sup>2</sup> No texto convencionou-se utilizar o termo WebApps (*Web Applications*) para caracterizar sistemas e aplicativos baseados na Web, evitando assim utilizar vários termos, como: engenharia de *software* para Web, engenharia de *software* para o desenvolvimento de *websites*, engenharia de desenvolvimento de aplicações para Internet, etc.

a principal motivação deste trabalho, obtendo uma Exploração Visual de Dados educacionais.

### **1.3. Objetivos**

O objetivo principal deste trabalho é propor uma abordagem para integração das diferentes WebApps da plataforma Web-PIDE e visualização de indicadores e dados estatísticos das avaliações educacionais em um único repositório de conhecimento. Para validar e testar a abordagem serão selecionadas três ferramentas de diferentes contextos: ferramentas de visualização de dados, data warehouse e consultas avançadas, com as características do domínio específico, dotado de componentes que atendam as necessidades do projeto de pesquisa descritas no decreto nº 5.803 de 8 de junho de 2006, que dispõe sobre o Observatório da Educação.

A plataforma terá um ambiente visual facilitador para consultas aos indicadores e informações correlacionadas disponibilizadas, público e aberto à comunidade na Web por meio do portal do Programa Observatório da Educação. Os usuários potenciais do portal são gestores públicos, pesquisadores, membros da administração pública e cidadãos que buscam indicadores e sistemas de indicadores sobre a Educação.

Neste contexto, os objetivos específicos são:

- 1) propor e discutir como as tecnologias de Engenharia de Software podem auxiliar o desenvolvimento de ferramentas reutilizáveis no contexto de visualização de dados educacionais;
- 2) pesquisar as ferramentas de livre acesso, disponibilizadas atualmente pelo INEP em seu Portal;
- 3) propor uma arquitetura baseada em WebServices na plataforma Web-PIDE que permitirá a integração das diversas WebApps, definindo mecanismos e estratégias de busca, comparação e visualização dos indicadores;
- 4) avaliar e testar a abordagem proposta por meio da implementação do ambiente intitulado Amb-PIDE; e
- 5) disponibilizar recursos e/ou ferramentas para monitoramento e acompanhamento que subsidiem a tomada de decisões governamentais.



## **1.4. Organização do Texto**

O presente texto está organizado em cinco capítulos. Neste Capítulo foram apresentados o contexto, as motivações e os objetivos principais do presente trabalho. No Capítulo 2 são correlacionados os diferentes métodos de visualização da informação, descrevendo as técnicas principais e seus domínios de utilização. O desenvolvimento de Aplicações Web é apresentado no Capítulo 3, bem como toda a estrutura necessária para a Engenharia de Desenvolvimento, métodos para reutilização de código, projeto e teste de interfaces com o usuário. A abordagem para integração e visualização de dados a ser utilizada neste trabalho é descrita no Capítulo 4. Para validar e testar a abordagem proposta, é implementado o ambiente Amb-PIDE, apresentado no Capítulo 5. Como conclusões deste trabalho são descritas as contribuições e trabalhos futuros no Capítulo 6. Por fim, são apresentadas as referências bibliográficas utilizadas e os anexos.

## Capítulo 2

# Visualização de Informação

### 2.1. Considerações Iniciais

A visualização de informações, definida por Card *et al.* [CMS1999], é a representação visual de dados abstratos de forma interativa, baseada no computador para amplificar a cognição. Atualmente pode-se gerar rapidamente uma representação visual com o uso do processamento do computador e, assim, oferecer ao usuário a habilidade de não somente obter uma apresentação clara e compreensível, mas também de modificá-la, a fim de permitir o destaque dos pontos de seu interesse.

Sendo assim, esta representação deve expressar as suas propriedades mais relevantes, mostrando como os diferentes itens estão relacionados entre si. A projeção de uma representação que funcione para grandes conjuntos de dados é um problema freqüente [CAV2002], e geralmente induz a necessidade da limitação das informações a serem exibidas por meio de filtros delimitados pelo próprio usuário.

Neste contexto, o desenvolvimento de sistemas de visualização de informações deve considerar a melhor forma de mapear os dados para uma representação que facilite a interpretação por parte dos usuários.

Enfim, o objetivo final da visualização de informações é auxiliar no entendimento de determinados assuntos, o qual, sem uma representação visual, exigiria um maior esforço de compreensão [GMS2008]. Desta forma, abordam-se neste Capítulo as técnicas de visualização de informações tabulares, gráficas e geográficas,

como descritas na Seção 2.2. Na Seção 2.3 são enumeradas as ferramentas de visualização existente atualmente no portal do INEP. As considerações finais deste Capítulo serão apresentadas na Seção 2.4.

## 2.2. Técnicas de visualização

Um modelo de referência para visualização de informações é sugerido por Card *et al.* [CMS1999], e transcreve por meio de passos desde as transformações que os dados sofrem até serem mapeados em uma representação visual. Como visto na Figura 2.1, a primeira transformação trabalha o "Dado Bruto", armazenando-o em "Tabelas de Dados". O segundo passo prevê a escolha de uma melhor "Estrutura Visual" (linear, mapa, tri-dimensional, temporal ou árvore) para o dado e finalmente são aplicadas "Transformações Visuais" a partir de parâmetros gráficos (posição, escala).

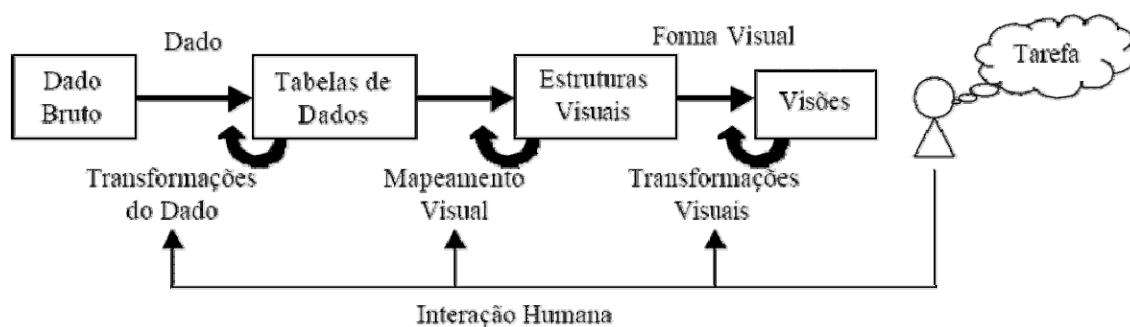


Figura 2.1: Modelo de referência para visualização de informações [CMS1999].

Resumidamente, as técnicas de visualização de informações procuram representar graficamente dados de um determinado domínio de aplicação de modo que a representação visual gerada explore a capacidade de percepção humana e este, a partir das relações espaciais exibidas, interprete e compreenda as informações apresentadas e, finalmente, deduza novos conhecimentos.

Uma das principais considerações do processo de visualização é a determinação de qual técnica deve ser empregada em uma determinada aplicação ou situação. Esta escolha é certamente dependente do tipo de informação que está sendo tratada e das tarefas que precisam ser realizadas pelo usuário. A seguir são apresentadas as principais técnicas de visualização de dados para a utilização neste trabalho.

## 2.2.1 Técnicas de Visualização de Informações Tabulares e Gráficas

Uma das técnicas mais simples de visualização de informações é a tabulação, que consiste na organização dos conjuntos de dados em tabelas, ou seja, dispor os registros em linhas e a sua dimensionalidade de atributos correspondendo às colunas.

É fato que os seres humanos têm dificuldades de processar representações de dados tabulares, devido ao fato de sua memória de curto prazo ser pequena, e isto implica que, após a leitura de alguns números as pessoas começam a esquecer os primeiros.

Os gráficos podem apresentar centenas, milhares ou até milhões de registros de dados em uma única representação visual. Assim, a utilização de técnicas de visualização de informações na forma de gráficos facilita sensivelmente o processo de busca e compreensão de padrões e anomalias nos mesmos.

Exemplificando estes conceitos, a Tabela 2.1 apresenta a visualização de um conjunto de dados em formato tabular. Para a maioria das pessoas, é difícil a detecção de um padrão utilizando este formato. Já na visualização do mesmo conjunto de dados apresentados na Tabela 2.1 utilizando representação gráfica mostrada na Figura 2.2, é mais fácil interpretar o padrão mapeado para um modelo senoidal.

Tabela 2.1: Dados em representação tabular [ALM2003].

X	0	0,259	0,5	0,707	0,866	0,966	1	0,966	0,866	0,707
Y	12	13	14	15	16	17	18	19	20	21
X	0,5	0,259	0	-0,259	-0,5	-0,707	-0,866	-0,966	-1	-0,966
Y	22	23	24	25	26	27	28	29	30	31
X	-0,866	-0,707	-0,5	-0,259	0	0,259	0,5	0,707	0,866	0,966
Y	32	33	34	35	36	37	38	39	40	41

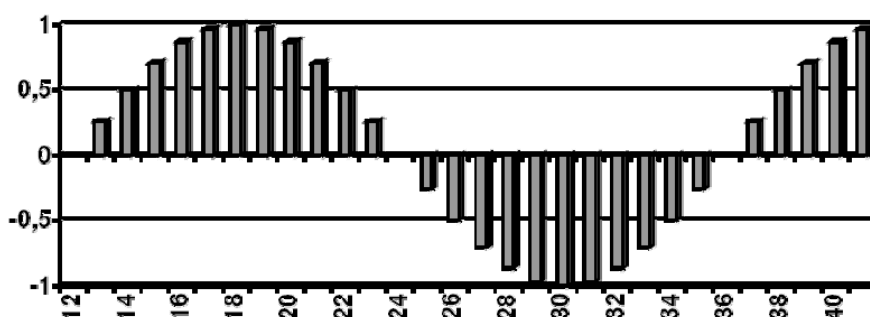


Figura 2.2: Dados em representação gráfica [ALM2003].

A tabulação dos dados é a base não só para a representação gráfica, mas também para qualquer outra técnica de representação visual, e assim, a representação

em tabelas de listagem é a mais comum para visualização de indicadores em qualquer domínio de informações.

### 2.2.2 Técnicas de Visualização de Informações Geográficas

As técnicas de visualização de informações geográficas representadas em um Sistema de Informação Geográfico (SIG ou GIS<sup>3</sup>), são utilizados especificamente para apoiar a aquisição, gerenciamento, manipulação, análise, consulta e representação de dados espacialmente referenciados [FID2005].

Os dados espacialmente referenciados (dados geográficos) representam objetos ou fenômenos cuja localização geográfica é uma característica essencial à informação e indispensável à sua análise. Assim, os mesmos não podem ser considerados sozinhos no espaço porque a sua análise e interpretação dependem basicamente de sua localização neste espaço de acordo com outros dados, ou seja, a sua referência espacial é extremamente importante. Informações geográficas podem conter tanto uma referência geográfica explícita (latitude e longitude), como também uma referência implícita (endereço, código postal, *etc.*).

Baseado no enfoque de banco de dados, Soares [SOA2002] define SIG como um banco de dados no qual a maioria de seus dados está indexada espacialmente, e para o qual se especifica um conjunto de procedimentos de modo a responder consultas a respeito das entidades espaciais envolvidas.

Estes bancos de dados têm sido coletados em forma digital há mais de 30 anos e o alcance total desta coleção aumenta rapidamente com avanços na tecnologia, tais como sistemas de imagens de satélite de alta resolução e sistemas de posicionamento global (GPS<sup>4</sup>), como também o crescente número de pessoas e organizações que coletam ou utilizam estes dados.

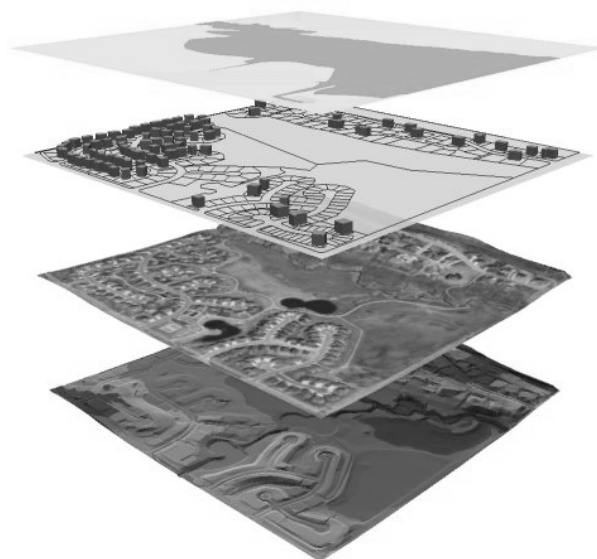
A representação vetorial é a forma visual mais simples e mais freqüentemente usada. É constituída por entidades espaciais básicas especificadas por atributos e localização geográfica, ou seja, informações sobre pontos, linhas e polígonos são codificadas e armazenadas como uma coleção de coordenadas x, y. A representação vetorial é útil para descrever características discretas, mas menos utilizadas para descrever características continuamente variantes (vegetações, por exemplo).

---

<sup>3</sup> Os Sistemas de Informações Geográficas são comumente referenciados pela sigla GIS devido seu termo em inglês *Geographical Information System*.

<sup>4</sup> Os sistemas de posicionamento global são vulgarmente conhecidos por GPS, devido o acrônimo do inglês *Global Positioning System*.





**Figura 2.4: Sobreposição de camadas em mapas.**

Portanto, os sistemas de informação geográfica integram informações espaciais provenientes de dados cartográficos, dados de censo e cadastro urbano e rural, imagens de satélite, redes e modelos numéricos de terreno, a fim de oferecer mecanismos para análise geográfica, através de facilidades para consultar, recuperar, manipular, visualizar e plotar o conteúdo da base de dados geo-codificados [CFP+2002].

Esta forma de representação permite uma análise mais intuitiva e precisa para grandes aglomerações de informações e extração de conhecimento por exploração visual das informações geo-codificadas, permitindo assim a utilização dos indicadores educacionais desta proposta.

### **2.3. Ferramentas de Visualização do INEP**


Como ferramenta de visualização, o INEP disponibiliza os Sistemas de Consulta em seu portal (<http://www.inep.gov.br>), que utilizam basicamente técnicas de representação tabular e gráfica de indicadores no domínio educacional.

Com recursos similares e sem integração entre suas bases de dados, estes sistemas de consulta não oferecem ferramentas que permitam uma melhor visualização dos dados consultados.

A Tabela 2.2 enumera as ferramentas de consulta e visualização existentes no portal do INEP sob o título de “Sistema de Consulta” e fornece classificação quanto à forma de visualização apresentada:

- Tabular – quando há apresentação de tabelas construídas na própria página, em formato HTML (Linguagem de Marcação de Hipertexto);
- Planilha – conteúdo armazenado em planilha no formato XLS (Microsoft Office Excel), geralmente várias tabelas;
- Documento – arquivo de documento digital no formato PDF (*Portable Document Format*) contendo relatórios textuais com tabelas e gráficos ilustrativos; e
- Gráfico – apresentação visual na forma de gráficos apresentados na página como imagens, normalmente no formato JPEG (*Joint Photographic Expert Groups*), sem a necessidade de utilizar outros programas.

Tabela 2.2: Ferramentas de Visualização do INEP.

FERRAMENTAS	FORMAS DE VISUALIZAÇÃO			
	 Tabular (HTML)	 Planilha (XLS)	 Documento (PDF)	 Gráfico (JPEG)
Consulta IDEB	✓			
DataEscolaBrasil	✓			
EDUDATABRASIL	✓	✓		
Investimentos Públicos em Educação	✓	✓	✓	
Sinopses Estatísticas da Educação Básica e Superior	✓	✓	✓	

A **Consulta IDEB** (<http://sistemasideb.inep.gov.br> – Figura 2.5) é a ferramenta que disponibiliza consultas ao Índice de Desenvolvimento da Educação Básica, um indicador de qualidade educacional que combina informações de desempenho em exames padronizados (Prova Brasil ou SAEB) obtido pelos estudantes ao final das



etapas de ensino (4<sup>a</sup> e 8<sup>a</sup> séries do ensino fundamental e 3<sup>a</sup> série do ensino médio) com informações sobre rendimento escolar (aprovação).

O **EDUDATABRASIL** (<http://www.edudatabrasil.inep.gov.br> – Figura 2.6), Sistema de Estatísticas Educacionais, tem o objetivo de permitir à sociedade acompanhar a evolução dos indicadores educacionais do país, pretende facilitar a consulta de informações tanto por parte dos pesquisadores, como por parte dos gestores de políticas públicas e do público em geral, ao oferecer um amplo leque de alternativas para a leitura do banco de dados do INEP. O sistema conta com informações atuais, assim como dados de anos anteriores, o que possibilita a construção de séries históricas. Reúne dados de matrículas, docentes, infraestrutura e indicadores de eficiência e rendimento educacional de todos os níveis de ensino. As variáveis estão disponíveis com detalhamento até a esfera municipal e podem ser analisadas em diferentes dimensões, de acordo com o interesse do usuário.

O **DataEscolaBrasil** (<http://www.dataescolabrasil.inep.gov.br> – Figura 2.7), Sistema de Estatísticas Educacionais da Educação Básica, oferece as sinopses estatísticas da Educação Básica e apresenta dados referentes aos estabelecimentos de ensino, às matrículas, à função docente e rendimento escolar para as diferentes modalidades de ensino: Ensino Regular (Educação Infantil e Ensino Fundamental e Médio), Educação Especial e Educação de Jovens e Adultos. Os dados estão distribuídos de acordo com as regiões brasileiras e suas respectivas unidades da Federação. Não há representação visual dos dados estatísticos, apenas tabular.

Os indicadores brasileiros de Investimentos Públicos em Educação fornecem informações de cunho orçamentário e financeiro sobre a aplicação de recursos públicos em todos os níveis de ensino, desagregados por natureza de despesas, subfunção e por dependência administrativa. As Sinopses Estatísticas da Educação Básica e Superior apresentam dados referentes a estabelecimento, matrícula, função docente, movimento e rendimento escolar, para as diferentes modalidades de ensino brasileiras. Ambas as ferramentas são apresentadas como sistemas de consulta no Portal do INEP, mas apenas disponibilizam *download* de arquivos (no formato de planilha ou documento digital) divididos em *links* de categorias das informações.

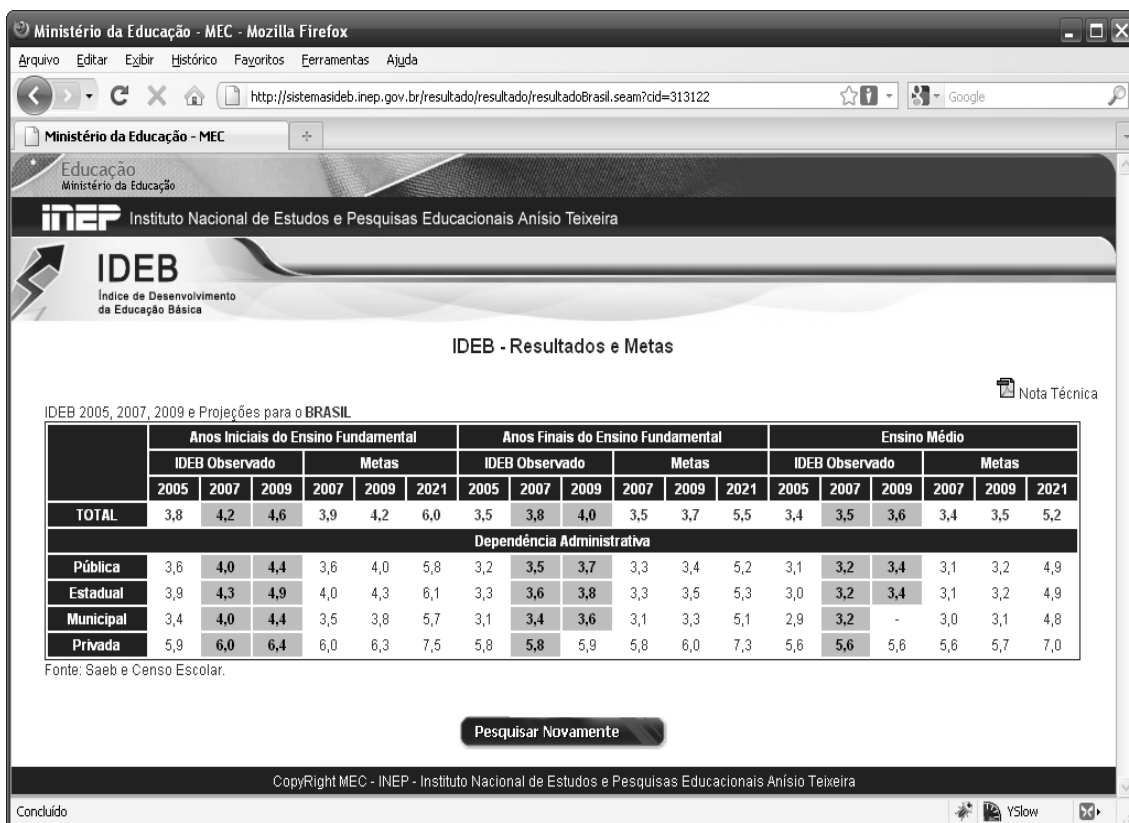


Figura 2.5: Interface principal do Consulta IDEB.

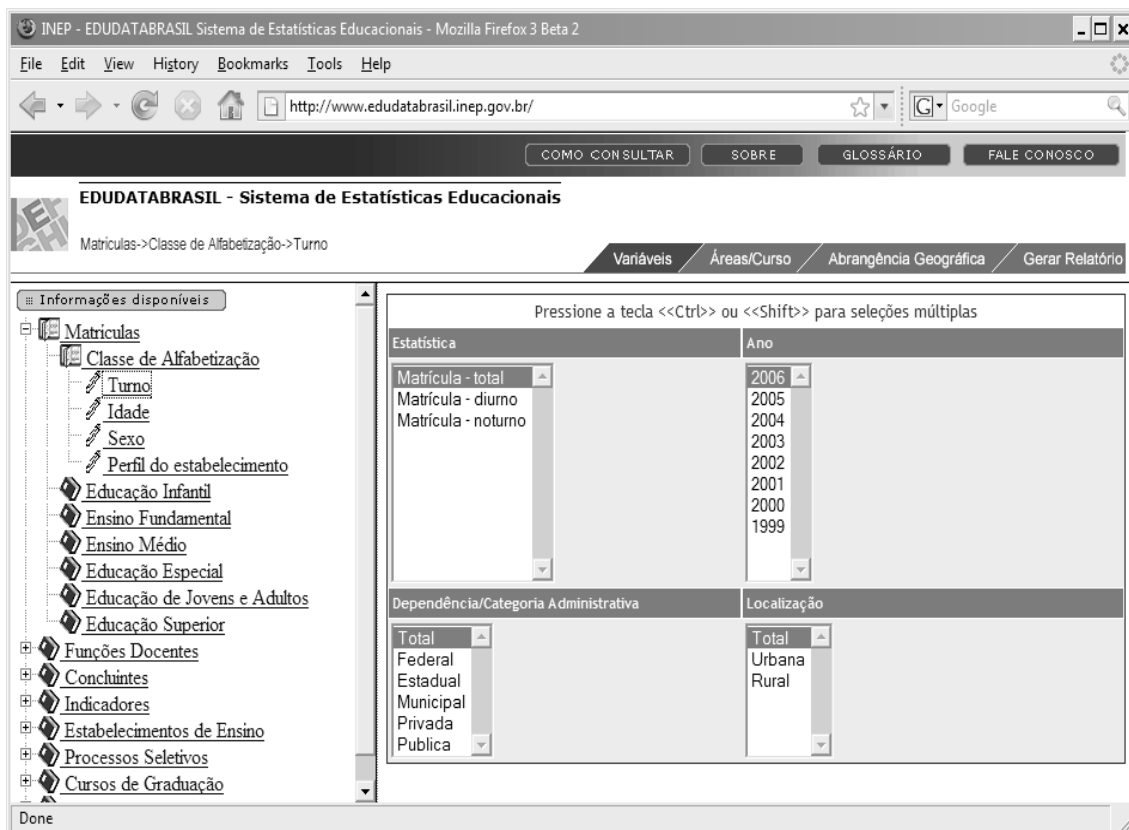
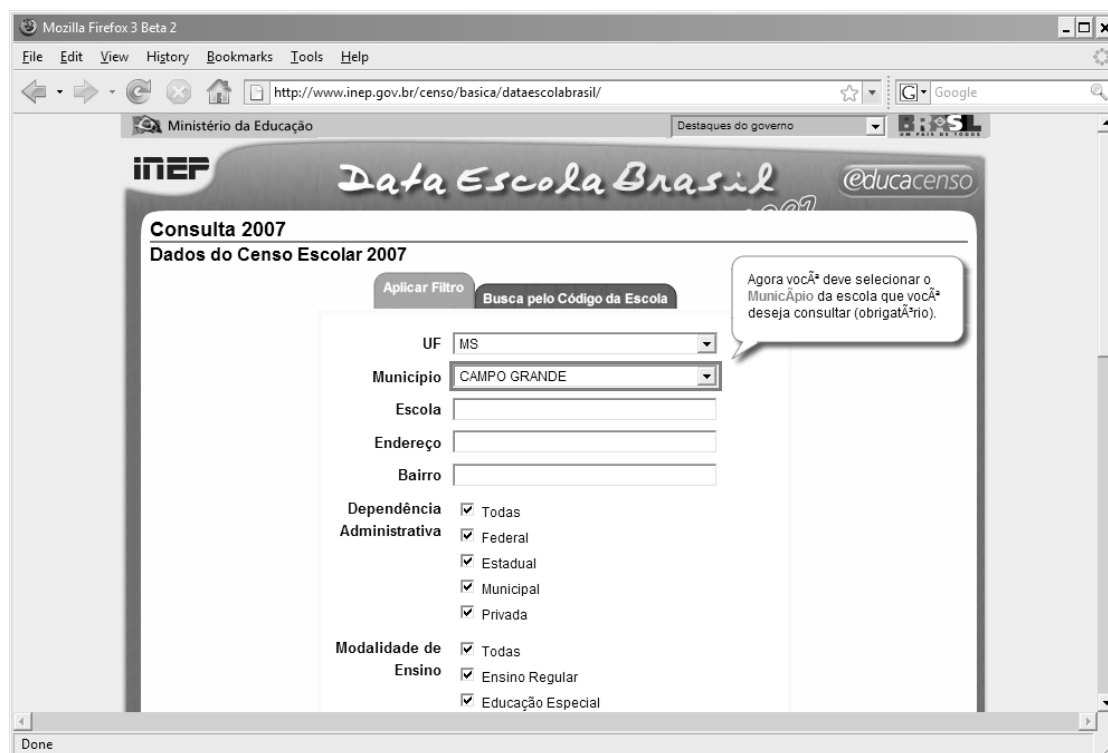


Figura 2.6: Interface principal do EDUDATABRASIL.



**Figura 2.7: Interface principal do DataEscolaBrasil.**

Além das ferramentas de consultas e visualização citadas, o INEP mantém em seu Portal o acesso a páginas específicas de cada avaliação realizada com disponibilização dos resultados de cada ano de aplicação no formato de planilhas para download ou em sinopses dos indicadores em formato PDF (*Portable Document Format*).

Também são disponibilizados microdados das principais pesquisas e avaliações realizadas, em formato textual fixo (ASCII - *American Standard Code for Information Interchange*), por meio de *download* no portal (<http://www.inep.gov.br/basica/levantamentos/acessar.htm>) ou fornecidos em CD mediante a solicitação.

Os microdados podem ser utilizados como entrada de *softwares* estatísticos, como por exemplo, SAS (Figura 2.8) e SPSS (Figura 2.9). Outros programas que não interpretem as extensões disponibilizadas (.sas e .sps) necessitarão da re-especificação da estrutura do arquivo de dados (.txt - Figura 2.10) por meio da leitura do dicionário de dados encontrado em formato digital portátil (.pdf - Figura 2.11).

```

DATA MATEMATICA_03ANO;
    INFILE 'D:\DADOS\ALUNOS\MATEMATICA_03ANO.TXT' LRECL=238 MISSOVER;
    IMPU:

01 MASCARA      8. /* MASCARA */
09 ANO_MASCARA  8. /* ANO_MASCARA */
017 ANO         4. /* Ano de aplicação */
021 SERIE       1. /* Série */
022 DISC        $char1. /* Disciplina */
023 TURMA       2. /* Número da turma */
025 TURNO       $char9. /* Turno */
034 ALUNO       3. /* Código do aluno */
037 ESTRATO     9. /* Estrato da amostra */
046 UPA         8. /* Unidade Primária de Amostragem */
054 DEP_ADN     1. /* Dependência Administrativa (Estadual/Municipal/Particular) */
055 LOCAL       1. /* Localização (Urbano/Rural) */
056 REDE        1. /* Rede de Ensino (Pública/Particular) */
057 UF          2. /* Unidade da Federação */
059 REGIAO     1. /* Região */
060 PROVA       $char1. /* Prova */
    
```

Figura 2.8: Exemplo do arquivo de entrada SAS para o SAEB.

```

* SELECIONE AS INFORMAÇÕES ABAIXO ANTES DE EXECUTAR O PROGRAMA*.

GET DATA /TYPE = TXT
/FILE = 'D:\DADOS\ALUNOS\MATEMATICA_03ANO.TXT'
/FIXCASE = 1
/ARRANGEMENT = FIXED
/FIRSTCASE = 1
/IMPORTCASE = ALL
/VARIABLES =
/1
MASCARA      0-7  F8
ANO_MASCARA  8-15  F8
ANO          16-19 F4
SERIE        20-20 F1
DISC         21-21 A1
TURMA        22-23 F2
    
```

Figura 2.9: Exemplo do arquivo de entrada SPSS para o SAEB.

20072500	199919993M	2Integral	23	98000010200725001115351292.7565	72.7647
20698646	199919993M	1Integral	2	80000012206986463124341388.8028	85.4537
20072500	199919993M	2Integral	30	98000010200725001115351303.6229	72.7647
20166353	199919993M	2Integral	2	97000001201663531115351315.3963	23.6154
20698646	199919993M	1Integral	9	80000012206986463124341361.6169	85.4537
20166353	199919993M	2Integral	9	97000001201663531115351299.0588	23.6154
20698646	199919993M	1Integral	16	80000012206986463124341235.3251	85.4537
20166353	199919993M	2Integral	16	97000001201663531115351191.3711	23.6154
20698646	199919993M	1Integral	23	80000012206986463124341356.4479	85.4537
21731661	199919993M	1Integral	1	55000003217316613123131330.9999	51.6684
20166353	199919993M	2Integral	23	97000001201663531115351240.5457	23.6154
20698646	199919993M	1Integral	30	80000012206986463124341337.2420	85.4537
21731661	199919993M	1Integral	8	55000003217316613123131272.4624	51.6684
20166353	199919993M	2Integral	30	97000001201663531115351258.0098	23.6154
21731661	199919993M	1Integral	15	55000003217316613123131322.3218	51.6684
20375908	199919993M	1Integral	1	83000002203759083125051261.0734	18.1474
21533726	199919993M	1Integral	4	79000011215337263124341387.9306103.9100	

Figura 2.10: Exemplo do arquivo de microdados para o SAEB.

DADOS ESTADÍSTICOS DO SISTEMA NACIONAL DE AVALIAÇÃO DA EDUCAÇÃO BÁSICA - Dicionário de Dados do arquivo CIÊNCIAS_04SERIE, GEOGRAFIA_04SERIE, HISTORIA_04SERIE, PORTUGUES_04SERIE E MATEMATICA_04SERIE.TXT									
Nome da Variável	Descrição	Categorias		Variáveis Categróricas		Início	Tamanho	Tipo	
		Categoria		Descrição					
A041_019	aspirador de pó	4	Quatro.			97	1	Numérica	
		5	Cinco.						
		6	Seis.						
		0	Nenhum.						
		1	Um.						
		2	Dois.						
		3	Três.						
		4	Quatro.						
		5	Cinco.						
		6	Seis.						
A041_020	telefone residencial	0	Nenhum.			98	1	Numérica	
		1	Um.						
		2	Dois.						
		3	Três.						
		4	Quatro.						
		5	Cinco.						
		6	Seis.						
A041_021	telefone celular	0	Nenhum.			99	1	Numérica	
		1	Um.						
		2	Dois.						
		3	Três.						
		4	Quatro.						
		5	Cinco.						
		6	Seis.						
A041_022	computador	0	Nenhum.			100	1	Numérica	
		1	Um.						
		2	Dois.						
		3	Três.						
		4	Quatro.						
		5	Cinco.						
		6	Seis.						
A041_023	automóvel	0	Nenhum.			101	1	Numérica	
		1	Um.						
		2	Dois.						
		3	Três.						
		4	Quatro.						
		5	Cinco.						
		6	Seis.						
A041_024	Em sua casa trabalha alguma empregada doméstica, todos os dias úteis? Quantas?	0	Nenhuma.			102	1	Numérica	
		1	1.						
		2	2.						
		3	3.						
		4	4.						
		5	5.						
		6	6 ou mais.						
A041_025	Quantas pessoas moram COM VOCÊ?	1	Nenhuma, moro sozinho(a).			103	1	Numérica	
		2	Uma pessoa.						
		3	Dois pessoas.						
		4	Três pessoas.						
		5	Quatro pessoas.						
		6	Cinco pessoas.						
		7	Seis pessoas.						
		8	Sete pessoas.						
		9	Mais do que sete pessoas.						
A041_026	Você mora com seu pai?	1	Sim.			104	1	Numérica	
		2	Não.						
A041_027	Você mora com sua mãe?	1	Sim.			105	1	Numérica	
		2	Não.						

Figura 2.11: Exemplo do arquivo de dicionário de dados do SAEB.

## **2.4. Considerações Finais**

Neste Capítulo foi apresentada uma breve taxonomia das principais abordagens de visualização de informação existentes a fim de enumerar recursos de visualização que serão empregados neste trabalho, e as ferramentas de visualização utilizadas no portal do INEP. No próximo Capítulo serão revisados os tópicos essenciais para o desenvolvimento de aplicações Web para o embasamento deste trabalho.

# Desenvolvimento de Aplicações Web

### 3.1. Considerações Iniciais

Inicialmente, a World Wide Web (WWW) era baseada em uma infraestrutura que dava suporte somente a páginas estáticas, ou seja, apenas documentos hipertextos entregues diretamente ao navegador do cliente como resposta a requisições HTTP (HyperText Transfer Protocol, protocolo utilizado na WWW) [W3C2003].

Com o surgimento das tecnologias e de linguagens de programação para a Web, tais como PHP (em 1994), Microsoft ASP (em 1995) e Java Servlets (em 1996) / JSP (em 1999), os servidores se tornaram mais robustos, permitindo o desenvolvimento de aplicações de negócio na Web. Surge o conceito de aplicação Web, ou WebApp, que consiste em um conjunto de páginas Web que interagem com o usuário provendo, armazenando e processando informações [GM2001], que são como sistemas de informação tradicionais, porém disponíveis na Internet, ou em uma *Intranet*, como é o caso de lojas virtuais, ambientes cooperativos, sistemas de gerência empresarial, dentre muitos outros.

Neste Capítulo apresenta-se uma visão geral sobre os conceitos envolvidos e utilizados neste trabalho, principalmente Engenharia Web. Na Seção 3.2 analisam-se as principais características que diferem a Engenharia da Web da Engenharia de Software convencional; o reuso de uma WebApp é descrito na Seção 3.3, e por fim, apresenta-se o projeto da interface com o usuário, bem como seus padrões e análises, na Seção 3.4. As considerações finais deste Capítulo serão apresentadas na Seção 3.5.

## 3.2. Engenharia da Web

Devido à crescente preocupação com o desenvolvimento e manutenção de aplicações Web, surgiu uma área específica da Engenharia de Software: a Engenharia Web, ou WebE (*Web Engineering*). A Engenharia Web tem como objetivo estudar abordagens sistemáticas e quantificáveis (conceitos, métodos, técnicas e ferramentas) para a análise de requisitos, projeto, implementação, teste, operação e manutenção de aplicações Web de alta qualidade [KAP2004].

Por representar uma evolução da engenharia de software convencional, algumas dificuldades adicionais motivaram as pesquisas relacionadas à engenharia de aplicações Web. Seu foco está em como desenvolver uma aplicação correta e completa, de acordo com os requisitos do usuário, com o diferencial de que esta deve ser desenvolvida no contexto de um projeto que considera a infraestrutura Web para sua execução e disponibilização.

Seguindo a abordagem em camadas da Engenharia de Software, a Engenharia Web também deve ter Foco na Qualidade - evitando assim as falhas iniciais que se propagam até o término do projeto.

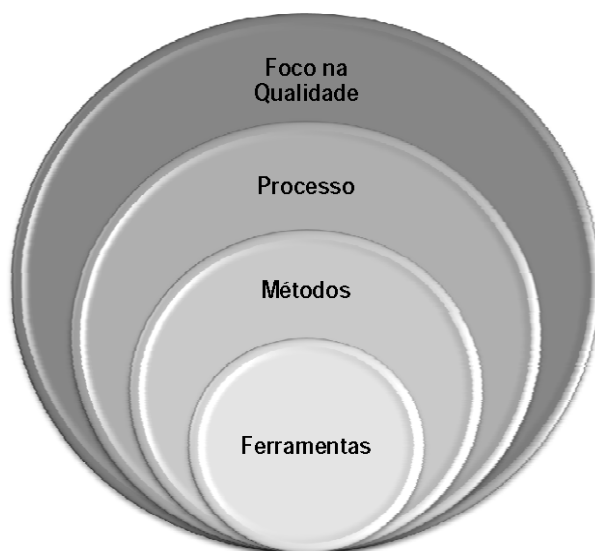


Figura 3.1: Camadas da Engenharia da Web [PRE2005].

Definindo as demais camadas, tem-se a camada de Processo, Métodos e Ferramentas. O Processo determina quais atividades e tarefas devem ser realizadas no desenvolvimento do projeto da aplicação Web, e estes são os alicerces para os Métodos. Os Métodos são conjuntos de tarefas técnicas que habilitam o engenheiro Web a entender, caracterizar e construir uma WebApp de alta qualidade. E por fim, as



ferramentas englobam todas as ferramentas de autoria utilizadas para desenvolvimento das aplicações, linguagem de programação, navegadores para visualização do conteúdo, sistemas de gerenciamento de banco de dados (SGBDs), ferramentas para gerenciamento de configuração, teste, análise e gestão, etc.

De acordo com a literatura, as seguintes características são encontradas na grande maioria dos aplicativos para a Web (ou, WebApps):

- **Rede intensiva:** aplicativos para a Web utilizam recursos de rede por natureza. Por estarem em uma rede, devem atender as necessidades de diversas comunidades de clientes. Pode estar disponível na Internet (permitindo comunicação com o mundo todo), em uma *Intranet* (implementando comunicação em uma organização) ou ainda em uma *Extranet* (comunicação inter-redes);
- **Dirigido ao conteúdo:** em muitos casos, a função primária do aplicativo para a Web é usar hipermídia para apresentar textos, gráficos, e vídeo para os usuários; e
- **Evolução contínua:** ao contrário dos aplicativos convencionais que evoluem através de uma série de versões planejadas e lançadas em determinados intervalos de tempo, os aplicativos para a Web evoluem continuamente.

Pressman [PRE2005] também cita as principais diferenças entre desenvolver um aplicativo para a Web e desenvolver um software tradicional:

- **Imediatismo:** o tempo que um site completo precisa ficar pronto pode ser apenas alguns poucos dias ou semanas. Desenvolvedores devem, portanto, utilizar métodos de planejamento, análise, projeto, implementação e teste que estejam adaptados para estes cronogramas comprimidos necessários no desenvolvimento para a Web;
- **Segurança:** para aplicativos para a Web que estão disponíveis via rede, é difícil ou até mesmo impossível limitar a população de usuários que virão acessá-lo. Para poder proteger o conteúdo e fornecer métodos seguros de transmissão de dados é preciso implementar medidas rígidas de segurança no aplicativo e na infraestrutura do mesmo; e

- **Estética:** é inegável que boa parte do apelo dos aplicativos para a Web é o seu visual. Quando um aplicativo é projetado para vender produtos ou idéias, a estética pode ser tão importante para o sucesso quanto o projeto técnico.

Estas características gerais se aplicam a todas as WebApps, mas com diferentes graus de influência. Logo, pode-se categorizar as WebApps como:

- **Informacional.** Conteúdo apenas para leitura é fornecido com navegação simples e *links*;
- **Download.** Um usuário faz o *download* de informações dos servidores apropriados;
- **Personalizável.** O usuário personaliza o conteúdo para suas necessidades específicas;
- **Interação.** Comunicação entre uma comunidade de usuários ocorre em salas de bate-papo, fóruns ou mensagens instantâneas;
- **Entrada de Usuário.** Entradas baseadas em formulários são os mecanismos primários para a comunicação necessária;
- **Orientado a transações.** O usuário faz um pedido que é atendido pelo aplicativo;
- **Orientado a serviços.** O aplicativo fornece um serviço para o usuário.
- **Portal.** O aplicativo direciona o usuário para outros conteúdos ou serviços fora do domínio do portal do aplicativo;
- **Acesso a Banco de Dados.** O usuário faz uma consulta em um banco de dados e extrai informações; e
- **Data warehousing.** O usuário consulta uma coleção de grandes bancos de dados e extrai informações.

Um metamodelo pode ser representado em UML (*Unified Modeling Language*), retratando a estrutura genérica de uma WebApp (Figura 3.2).

Neste modelo, no centro há a classe “*WebPage*” (página Web), que contém a informação que é exibida ao usuário e os “*links*” (vínculos) de navegação que levam as outras “*WebPages*”. Também inclui facilidades de organização e de interação como: “*frames*” (quadros) e “*forms*” (formulários). A navegação de uma página a outra é

representada pela auto-associação da classe “*WebPage*” nomeada de “*link*”. Pelo modelo observa-se que uma página pode ser de dois tipos: estática ou dinâmica.

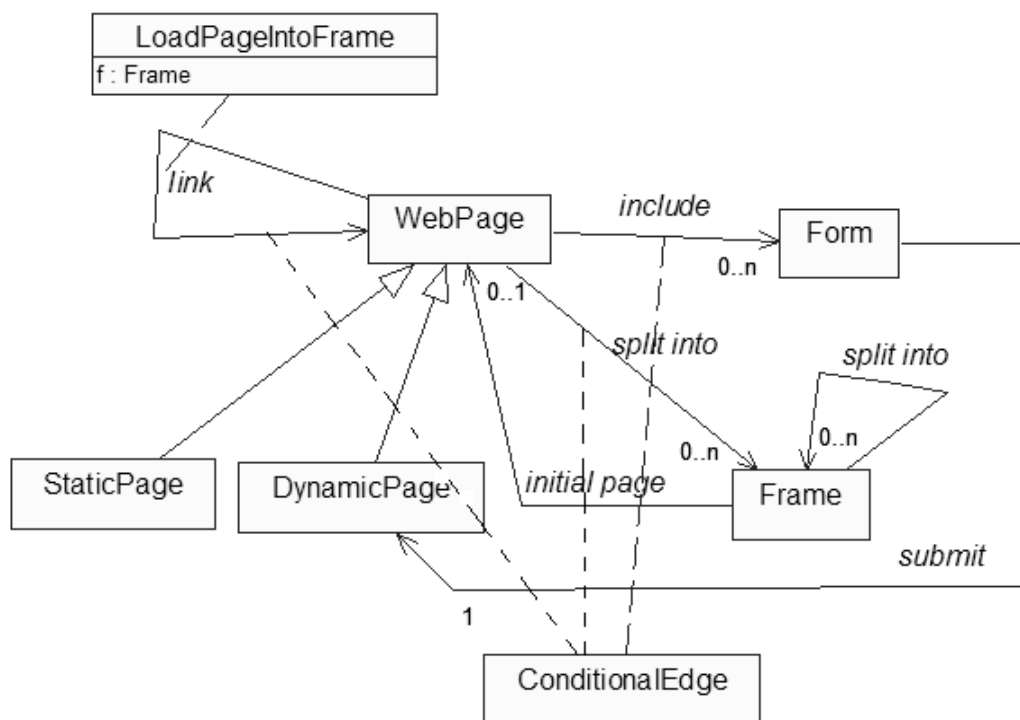


Figura 3.2: Modelo genérico de WebApp [RT2001].

Enquanto o conteúdo de uma página estática é fixo, o conteúdo de uma página dinâmica é gerado em tempo de execução pelo servidor. Um “*frame*” é uma área retangular disposta dentro da página, onde a navegação pode acontecer independentemente das outras áreas da “*WebPage*”. A divisão da página em “*frames*” pode ser recursiva (auto-associação “*split-into*” na classe “*frame*”). Quando a navegação em uma página força a “*carga*” de outra página em um “*frame*” diferente do atual, o “*frame*” alvo torna-se um parâmetro da classe de associação “*LoadPageIntoFrame*” (carregar página dentro do quadro).

Em HTML, entradas/requisições de usuários podem ser tratadas pelo uso de “*forms*” (formulários). Uma página Web pode incluir um número qualquer de “*forms*” através da associação “*include*” (incluir). Cada “*form*” é caracterizado pelas variáveis de entrada que são fornecidas pelos usuários. Valores coletados pelos “*forms*” são submetidos ao servidor Web através de um vínculo especial “*submit*” (submeter). Como “*links*”, “*frames*” e “*forms*” são partes do conteúdo da página Web, e, para páginas dinâmicas o conteúdo pode depender das variáveis de entrada, até a organização da página, em geral, é não-fixa e depende dos valores de entrada. Daí, a razão para a classe

de associação “*ConditionalEdge*” (borda condicional), a qual opcionalmente adiciona uma condição booleana (verdadeiro/falso), função da entrada das variáveis, representando a existência de uma condição da associação que pode ser um “*link*”, “*include*” ou “*split-into*” (dividir). O alvo página, “*form*” ou “*frame*” é referenciado pelo código da página dinâmica apenas quando os valores de entrada satisfazem a condição em “*ConditionalEdge*” [RT2001].

### 3.3. Desenvolvimento Baseado em Componentes

Segundo Bosch *et al.* [BMM+1999], a reutilização de software é um propósito enunciado quase que simultaneamente com o surgimento da própria Engenharia de Software, e atualmente usa-se conceitos de Análise de Domínio e Frameworks para facilitar o processo de reuso.

#### 3.3.1 Análise de Domínio

A análise de domínios evoluiu na década de 80, a fim de maximizar a reutilização no desenvolvimento de software, por meio da identificação e organização do conhecimento a respeito de uma classe de problemas - um domínio de aplicações [BMM+1999]. É um passo fundamental na criação de artefatos de softwares reutilizáveis, já que o modelo produzido capta a funcionalidade essencial requerida por um domínio e pode ser reutilizado durante a análise de sistemas dos mesmos domínios.

Arango *et al.* [AP1991] define a Análise de Domínio como sendo o processo de identificação e organização de conhecimento a respeito de uma classe de problemas (domínio de aplicações) para suportar a descrição e solução destes problemas. Logo, a Análise de Domínio é um passo fundamental na criação de artefatos de softwares reusáveis, já que capturam a funcionalidade essencial requerida por um domínio.

O objetivo de um procedimento de Análise de Domínio é produzir subsídios para a reutilização de software. Estes subsídios correspondem a uma descrição do domínio contida em um modelo do domínio. Esta descrição de domínio pode conter apenas uma descrição de características e funcionalidades do domínio, ou pode conter artefatos reutilizáveis. No primeiro caso, a descrição atua como uma fonte de informação do domínio a ser consultada em um desenvolvimento de aplicação, e no segundo caso, a descrição do domínio contém partes reutilizáveis no desenvolvimento de aplicações.

Uma abordagem prática para Análise de Domínio, baseando-se no paradigma da Engenharia do Conhecimento, é proposta por Arango *et al.* [AP1991] e consiste nas seguintes etapas:

1. Identificar as fontes de conhecimento;
2. Adquirir o conhecimento existente;
3. Representar o conhecimento, através de um modelo de domínio.

Assim, pode-se afirmar que a Análise de Domínio envolve duas grandes etapas:

- Elicitação de conhecimento do domínio;
- Produção do modelo do domínio.

O processo de Análise de Domínio tem como resultado o modelo do domínio, e as informações para a construção deste modelo são obtidas a partir de especialistas do domínio, de literatura técnica, de aplicações existentes, de pesquisas de mercado, etc. A construção do modelo do domínio das metodologias de Análise de Domínio pode ser assistida por ferramentas de apoio às atividades, como editores e analisadores de especificação, ferramentas de Engenharia Reversa, *etc.*

A especificação do sistema na Engenharia de Software está relacionada com a especificação da infraestrutura na Engenharia do Domínio, enquanto o projeto e a implementação abordada na Engenharia de Software estão relacionados com a implementação da infraestrutura da Engenharia de Domínio. A ligação entre as duas disciplinas não se dá apenas por meio de uma analogia, esta ligação é intrínseca (Figura 3.3).

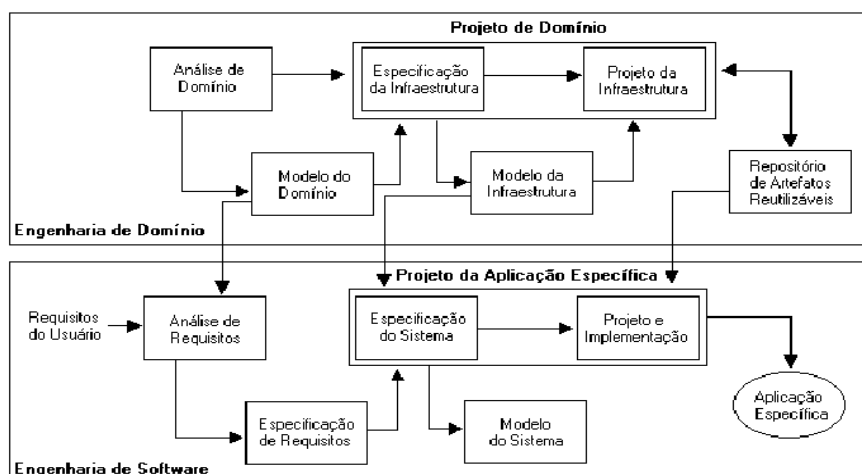


Figura 3.3: Engenharia de Domínio x Engenharia de Software.

### 3.3.2 Reuso e Frameworks

Segundo Carvalho [CAR2005], o modelo de Programação Orientada a Objetos (POO) se caracteriza principalmente pelo seu suporte a três propriedades: encapsulamento, reusabilidade e extensibilidade.

Dentre as três propriedades citadas, a reusabilidade é uma técnica discutida há tempo, já que a dificuldade em se conseguir a reutilização de *Software* vem de décadas. A reutilização, segundo Pressman [PRE2005], é uma estratégia de resolução de problemas utilizada na maioria das atividades do ser humano.

A reutilização de software surgiu no final da década dos anos 60 como uma alternativa para superar a crise do *Software*, e tem como objetivo principal desenvolver *Software* utilizando artefatos já especificados, implementados e testados.

Na maioria dos casos, faz-se reutilização apenas do código fonte da aplicação por meio de rotinas armazenadas em bibliotecas (conjunto de procedimentos com funcionalidades de uma mesma categoria).

Sendo assim, não se pode considerar como suficiente esta forma de reutilização, pois há a necessidade de reutilizar-se, além do código fonte, também o projeto de um *Software*. Portanto, a reutilização precisa abranger tanto as fases de implementação quanto de análise e projeto.

Paez [PAE1999] aborda uma categorização das técnicas de reutilização propondo uma taxonomia geral da reutilização (conforme Tabela 3.1). Para tal, são enumerados cinco aspectos de comparação: aproximação tecnológica, enfoque metodológico, modificação, alcance do desenvolvimento e alcance do domínio. Com isso, pode-se determinar o contexto de uso de uma técnica de reutilização a partir dos aspectos propostos.

**Tabela 3.1: Taxonomia de reutilização [PAE1999].**

Aspecto	Tipo	Descrição
<b>Aproximação Tecnológica</b>	Por geração	Parte de uma especificação do problema que será refinada em iterações sucessivas.
	Por composição	Utiliza pequenas partes de software como invariantes para prover nova funcionalidade por meio de sua integração.
<b>Enfoque Metodológico</b>	Desenvolvimento para reutilização	Centrado no desenvolvimento de componentes adequados e específicos para um problema específico – visão provedor.

	Desenvolvimento com reutilização	Centrado no processo de construção de soluções de software a partir de componentes armazenados numa biblioteca – visão demanda.
<b>Modificação</b>	Caixa-branca	Componentes são utilizados por modificação e adaptação.
	Caixa-preta	Componentes que são reutilizados sem nenhuma modificação.
	Adaptativo	Utiliza grandes estruturas de software como invariantes e restringe a variabilidade a um conjunto de argumentos ou parâmetros.
<b>Alcance do Desenvolvimento</b>	Interno	Mede o nível de reutilização de componentes de um repositório do mesmo projeto.
	Externo	Mede o nível de reutilização de componentes que provêm de repositórios externos ou a proporção de produtos que foram adquiridos.
<b>Alcance do Domínio</b>	Vertical	Reutilização dentro do mesmo domínio de aplicação.
	Horizontal	Reutilização dentro de diferentes domínios de aplicação.

Neste contexto, o projetista poderia, portanto, construir uma espécie de aplicação semi-acabada (arcabouço), constituído de classes associadas e atuando em conjunto. Quando novas funcionalidades forem necessárias, bastaria a criação de subclasses a partir das classes componentes deste arcabouço. Tais arcabouços são tipicamente conhecidos como *frameworks*. Segundo Budd [BUD1996], um *framework* é um conjunto de classes que cooperam mutuamente umas com as outras, compondo a infraestrutura necessária ao desenvolvimento de uma aplicação específica.

Assim, o projetista ao construir um *framework* deverá definir as classes e seus relacionamentos (dependências), de tal forma que possa atender aos requisitos de determinada aplicação, diminuindo assim o tempo gasto e incrementando a reusabilidade do código.

O desenvolvedor de uma aplicação, ao utilizar um *framework*, tipicamente realiza as seguintes atividades: definição de novas classes (subclasses) necessárias para a aplicação específica; sobreposição de métodos de algumas classes do *framework* nas novas subclasses; e configuração de um conjunto de objetos, provendo parâmetros e conectando-os. Portanto, o desenvolvedor pode ampliar as funcionalidades do próprio

*framework* adicionando novas classes necessárias à sua aplicação sem fazer mudanças na estrutura interna do mesmo.

Logo, nas aplicações tradicionais que não utilizam *frameworks* e funcionam normalmente com bibliotecas que são “chamadas” pelo código criado pelo desenvolvedor, o fluxo de controle é dirigido pela aplicação. Com a utilização de *frameworks* há uma inversão deste formato, já que quando utilizados estabelecem o fluxo de controle e o desenvolvedor apenas faz modificações nas bibliotecas “chamadas” pelo *framework* (intitulados *callbacks*); isto é feito sem afetar estruturalmente o mesmo.

Maldonado *et al.* [MGB+2002] faz uma comparação entre o custo e o benefício no uso de *frameworks*, como visto na Tabela 3.2, e chega a conclusão que mesmo com todas as desvantagens, as vantagens são ainda maiores, o que viabiliza a utilização de *frameworks* pelos programadores.

Não se define a construção de um *framework* como uma atividade simples já que o projetista necessita “prever” (com a ajuda de um especialista no domínio do problema) as possíveis necessidades do usuário do *framework*, uma vez que o usuário poderá estender o *framework* - seja pela adição de novas classes ou subclasses, seja simplesmente modificando algum parâmetro para ativar uma possível funcionalidade. Neste contexto, o ciclo de vida de um *framework* pode ser apresentado, segundo Silva [SIL2000], seguindo os princípios a seguir:

- I. O ciclo de vida de um *framework* está diretamente relacionado aos componentes que o deram origem, ou que surgiram a partir deste;
- II. O projetista do *framework* não só tem como responsabilidade a definição do grau de flexibilidade que os usuários poderão ter, como tem a importância na manutenibilidade e na estrutura sobre a qual o *framework* é composto; e
- III. A construção de um *framework* deve ser direcionada para um determinado domínio de aplicações.

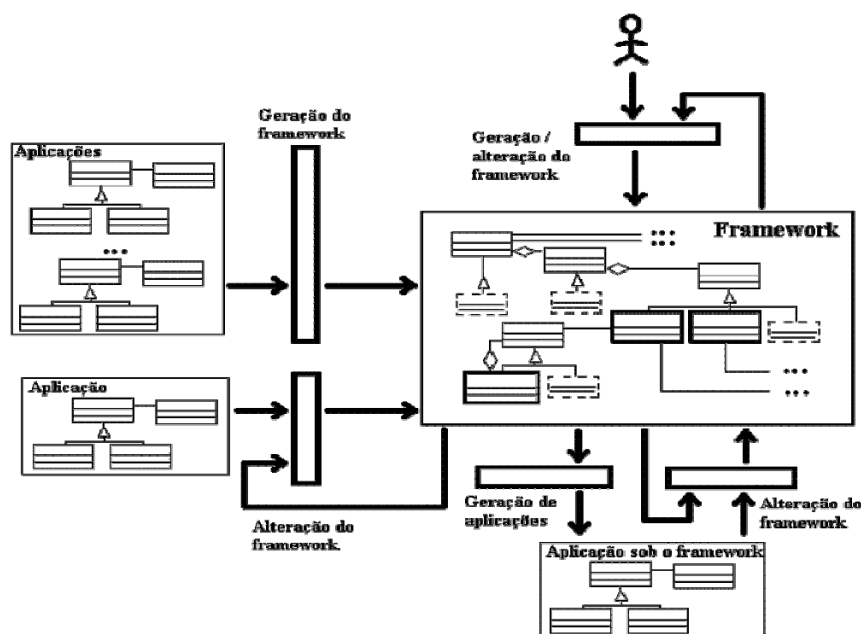


Tabela 3.2: Custos e benefícios no uso de *frameworks* [MGB+2002].

Custos	Benefícios
<ul style="list-style-type: none"> <li>• Mais esforço para construção e aprendizado;</li> <li>• Programas mais difícil de depurar; e</li> <li>• Necessária documentação, manutenção e apoio.</li> </ul>	<ul style="list-style-type: none"> <li>• Economia em longo prazo;</li> <li>• Aproveitamento da experiência de especialistas do domínio;</li> <li>• Maior consistência e integração entre aplicações;</li> <li>• Redução da manutenção: menos linhas de código nas aplicações, reparos no <i>framework</i> se propagam pelas aplicações; e</li> <li>• Melhora da produtividade: os programadores podem se concentrar nas características específicas de suas aplicações.</li> </ul>

Toda a interação existente no ciclo de vida de um *framework* pode ser ilustrada na Figura 3.4, na qual é focada desde a sua geração até a criação de aplicações baseadas no *framework*. Pode-se observar também a possibilidade de alteração do *framework* devido à existência de detalhes não considerados no momento de sua criação, mas que são necessários para o determinado domínio.

Portanto, o ciclo de vida de um *framework* só é finalizado quando todos os requisitos exigidos pelo seu domínio de aplicação forem atendidos, ou quando o domínio englobado deixa de ser utilizado. No contexto deste projeto, todas as soluções propostas serão projetadas utilizando os conceitos de *framework* para facilitar reuso, extensibilidade e adaptabilidade.

Figura 3.4: Ciclo de vida de um *framework* [SIL2000].

## 3.4. Projeto de Interface de WebApp

Tradicionalmente, considera-se que uma interface homem-máquina é um artefato que permite a um usuário controlar e avaliar o funcionamento de um sistema por meio de dispositivos sensíveis às suas ações e capazes de estimular sua percepção. A interface é tanto um meio para a interação usuário-sistema, quanto uma ferramenta que oferece os instrumentos para este processo comunicativo, ou seja, é um sistema de comunicação do usuário com a aplicação. Toda interface de usuário (IU) – quer seja projetada para uma WebApp, para uma aplicação de software tradicional, para um produto de consumo ou para um dispositivo industrial – deve exibir as seguintes características: fácil de usar, fácil de aprender, fácil de navegar, intuitiva, consistente, eficiente, livre de erros e funcional, fornecendo ao usuário uma experiência satisfatória e compensadora.

### 3.4.1 Técnicas de Interface com o Usuário

A fim de projetar interfaces que sigam as características anteriores, o projetista de interface da WebApp deve seguir um conjunto de princípios técnicos sobre interface com o usuário, enumeradas por Tognozzi [TOG2008]:

**Antecipação:** a WebApp deve ser projetada de modo a antecipar o próximo movimento do usuário, assim se um usuário solicitar um objeto de conteúdo o projetista deve antecipar que o mesmo usuário poderá solicitar um *download* de conteúdo relacionado, devendo fornecer facilidade de navegação para que isso ocorra sem que o usuário procure por essa capacidade.

**Comunicação:** deve ser comunicado pela interface da WebApp, o estado de qualquer atividade iniciada pelo usuário, o estado do usuário (por exemplo, a sua identificação) e a localização dentro da hierarquia de conteúdo desta WebApp.

**Consistência:** a utilização de controles de navegação, menus, ícones e estética deve se manter ao longo da WebApp, assim, toda característica da interface deve responder para que seja consistente com as expectativas do usuário.

**Autonomia controlada:** o movimento do usuário na WebApp deve ser facilitado pela interface, impondo convenções de navegação para a aplicação. Logo, se houverem partes restritas nesta aplicação, não deve haver mecanismos de navegação que permitam ao usuário contornar esta restrição.

**Flexibilidade:** a interface deve ser flexível de modo a permitir que enquanto alguns usuários realizem tarefas fixas, outros explorem a WebApp de modo aleatório, fornecendo funcionalidades para que possam desfazer erros e refazer caminhos de navegação mal escolhidos.

**Enfoque:** a interface da WebApp e o conteúdo nela apresentada deve permanecer concentrados nas tarefas atuais do usuário, assim o mesmo não se confundirá entre as camadas de informação de apoio e perder de vista o conteúdo original que ele requisitou.

**Alcance:** devem ser modeladas funcionalidades para movimentos rápidos, onde, com a ajuda de um apêndice em uma determinada posição inicial pode-se mover e visitar outras áreas da WebApp.

**Interface humana:** devem ser utilizados objetos de interface humana em uma WebApp, onde estes podem ser vistos, ouvidos, tocado ou percebido de algum outro modo.

**Redução de latência:** a WebApp deve permitir um ambiente multitarefa, onde o usuário poderá prosseguir seu trabalho sem ter de esperar que alguma operação interna seja completada (por exemplo, um *download*).

**Aprendizado:** a interface da WebApp deve ser projetada para minimizar o tempo de aprendizado e, uma vez aprendida, minimizar o reaprendizado. Para isso, as interfaces devem enfatizar um projeto simples, intuitivo, que organiza conteúdo e funcionalidades em categorias que sejam óbvias para o usuário.

**Metáforas:** a utilização de metáforas de interação torna o aprendizado e a utilização mais fáceis, desde que a metáfora seja apropriada para a aplicação e para o usuário. Assim, uma metáfora deve invocar imagens e conceitos da experiência do usuário, mas não precisa ser uma reprodução exata de uma experiência do mundo real.

**Integridade:** para manter a integridade do produto de trabalho em uma WebApp, devem ser fornecidos recursos que salvem o estado atual do mesmo (como por exemplo, recurso de auto-salvamento).

**Legibilidade:** as informações apresentadas na interface da WebApp deve enfatizar estilos de tipos legíveis, tamanhos de fonte e escolha de cor de fundo que aumentem o contraste, tornando assim esta informação legível para qualquer tipo de usuário (jovem, ou idoso).

**Rastrear estado:** quando apropriado, o estado da interação do usuário deve ser rastreado e armazenado para que um usuário possa sair e retornar mais tarde no ponto em que estava quando saiu.

**Navegação visível:** a interface da WebApp deve ser projetada de modo que forneça a ilusão de que os usuários estão no mesmo lugar, com um trabalho trazido até eles, e assim, a navegação não será uma preocupação do usuário.

Nielsen *et al.* [NW1996] sugere algumas diretrizes como complemento, para serem utilizadas junto às técnicas de projeto para interface com o usuário vistas anteriormente:

- A velocidade de leitura em um monitor de computador é 25% menor do que a velocidade de leitura em papel. Assim, não force o usuário a ler quantidades volumosas de texto, particularmente quando o texto explica a operação da WebApp ou apóia a navegação;
- Evite sinais “em construção”, pois eles levantam expectativas e causam um *link* desnecessário que certamente vai desapontar;
- Os usuários preferem não usar barra de rolagem, logo, informações importantes devem ser colocadas dentro das dimensões de uma típica janela de navegador;
- Menus de navegação e barras de topo devem ser projetadas consistentemente e estar disponíveis em todas as páginas que estão à disposição do usuário. O projeto não deve se apoiar nas funções do navegador para ajudar a navegação;
- A estética nunca deve superar a funcionalidade, ou seja, não se deve utilizar uma opção esteticamente agradável, mas com funcionalidade vaga;
- Opções de navegação devem ser óbvias, mesmo para o usuário eventual, ou seja, o usuário não deve ter que procurar na interface o modo de como se ligar a outro conteúdo ou serviço.

Assim, com as técnicas descritas para projeto de interface o usuário terá aumento da percepção, melhora do conteúdo ou serviços fornecidos pela WebApp, tornando-a bem estruturada e ergonomicamente eficiente.

### 3.4.2 Padrões de Interface com o Usuário

Nas últimas décadas foram propostos vários padrões para projeto de interface com o usuário. Tidwell [TID2008] fornece uma descrição completa de vários padrões e enquadra-os em dez categorias:

1. **IU Inteira** (*Whole UI*): fornece diretriz de projeto para estrutura e navegação de mais alto nível;
2. **Diagramação da página** (*Page layout*): trata da organização geral das páginas (para sites Web) ou exibições de telas distintas (para aplicações interativas);
3. **Formulários e entradas** (*Forms and inputs*): considera uma variedade de técnicas de projeto para entrada completa no nível de formulário;
4. **Tabelas** (*Tables*): fornece diretriz de projeto para criar e manipular dados tabulares de todas as espécies;
5. **Manipulação direta de dados** (*Direct data manipulation*): trata da edição, modificação e transformação de dados;
6. **Navegação** (*Navigation*): apóia o usuário na navegação por meio de menus hierárquicos, páginas Web e telas de exibição interativas;
7. **Busca** (*Searching*): habilita buscas por conteúdo específico por meio da informação mantida em um site Web ou contido em depósitos de dados persistentes aos quais se tem acesso via aplicação interativa;
8. **Elementos de página** (*Page elements*): implementa elementos específicos de uma página Web ou tela de exibição;
9. **Comércio eletrônico** (*e-Commerce*): específico de site Web, esses padrões implementam elementos recorrentes de aplicações de comércio eletrônico; e
10. **Miscelânea** (*Miscellaneous*): padrões que não são fáceis de encaixar em uma das categorias anteriores, sendo que em alguns casos, esses padrões são dependentes do domínio ou ocorrem somente para classe de usuários específicos.

Outro padrão utilizado na literatura e neste projeto é o MVC (*Model-View-Controller*, ou Modelo-Visão-Controlador) conforme o esquema descrito na Figura 3.5:

- A Visão possui elementos que representam informações de Modelo e as exibem ao usuário, que pode enviar, por meio deles, estímulos ao sistema, que são tratados pelo Controlador.
- O Controlador altera o estado dos elementos do Modelo e pode, se apropriado, selecionar outros elementos de Visão a serem exibidos ao usuário.
- O Modelo, por sua vez, notifica alterações em sua estrutura à Visão para que esta consulte novamente os dados e se atualize automaticamente.

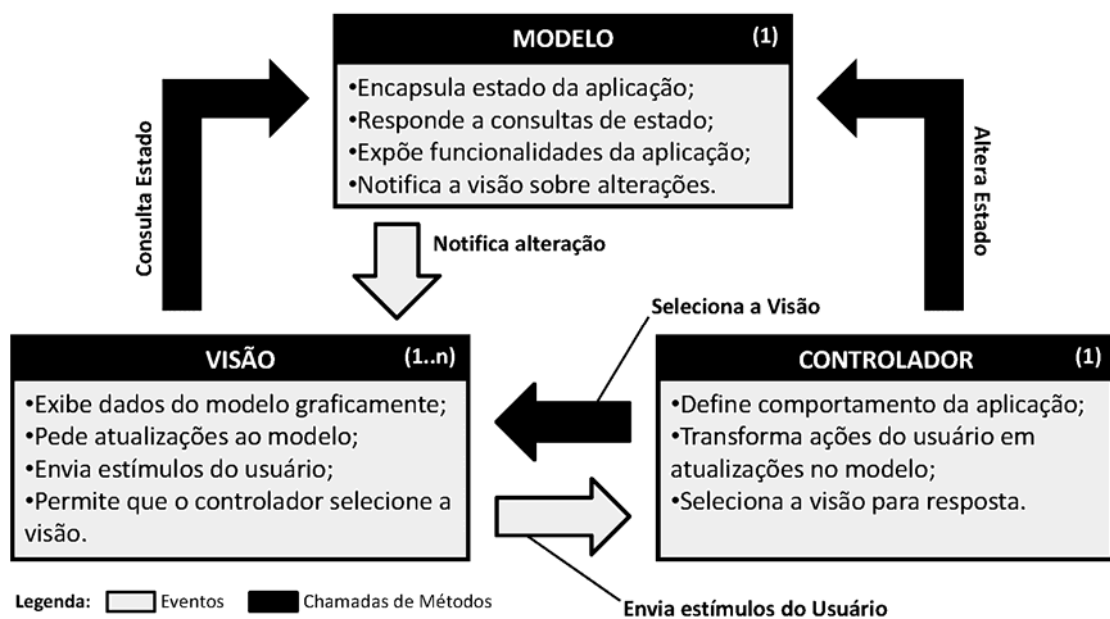


Figura 3.5: Funcionamento do padrão MVC.

### 3.5. Considerações Finais

Neste Capítulo apresentou-se o embasamento necessário sobre o desenvolvimento de aplicações Web (WebApps) com foco na reusabilidade e projeto de interface. No Capítulo seguinte descreve-se a abordagem proposta para integração de WebApps da plataforma Web-PIDE, bem como a visualização de dados, que é o objetivo final do trabalho.

# Abordagem para Integração e Visualização de Dados

### 4.1. Considerações Iniciais

Atualmente, há um grande aumento da quantidade de informações disponíveis na Internet, devido a diversos fatores, tais como avanços nas comunicações, ferramentas de governo eletrônico e o próprio crescimento da Web. Os usuários acessam essas grandes e diversificadas bases de dados e obtêm facilmente suas informações, dentre as quais muitas podem ser irrelevantes para os objetivos da tarefa que está sendo realizada. Portanto, a representação dos resultados obtidos, por meio de um mecanismo de recuperação de informações, é um dos desafios atuais.

Neste trabalho é investigada e proposta uma estratégia para integração e visualização de dados educacionais, implementando um ambiente centralizador de informações com uma camada de integração entre as diversas WebApps, facilitando as tarefas dos usuários gestores em seus fins.

Neste Capítulo é apresentada abordagem de integração e visualização utilizada para integrar as diversas WebApps que necessitem consultar os dados em avaliações educacionais, ressaltando a gestão do projeto (Seção 4.2), as funcionalidades propostas na abordagem (Seção 4.3) e a camada de integração proposta (Seção 4.4). As considerações finais sobre o Capítulo serão apresentadas na Seção 4.5.

## 4.2. Gestão de Projeto

Para facilitar a gestão dos projetos na plataforma Web-PIDE, de forma colaborativa e transparente, foi implementado o Portal Web-PIDE (<http://webpide.ledes.net> - Figura 4.1) utilizando o Titan [CT2005] como *framework* para Sistemas de Gerenciamento de Conteúdo (CMS<sup>5</sup>).



Figura 4.1: Portal Web-PIDE.

Integrados ao portal, neste mesmo contexto, três ferramentas de *Software* livre podem ser acessadas: Wiki, Redmine e Subversion.

<sup>5</sup> Devido o acrônimo do inglês *Content Management Systems*.



A **Wiki** (<http://wiki.ledes.net/index.php/WebPIDE>) permite que documentos sejam editados coletivamente com uma linguagem de marcação muito simples e eficaz apenas através da utilização de um navegador Web, sendo um recurso poderoso para documentação do projeto, pois mantém as páginas sempre atualizadas em tempo real.

O **Redmine** (<http://redmine.ledes.net/projects/show/webpide>) é uma ferramenta colaborativa para gerenciar projetos de desenvolvimento, oferecendo recursos importantes para controle de tarefas, bem como gráficos de andamento das atividades realizadas por membros da equipe.

O **Subversion** (<http://svn.ledes.net/web-pide>), ou apenas SVN, é um servidor de arquivos com controle de versões que permite a toda equipe acessar os arquivos atualizados e organizados de qualquer lugar *online*, além de restaurar facilmente as versões antigas para visualizar exatamente qual mudança gerou algum problema.

A motivação para desenvolvimento deste portal é a integração dos demais projetos desenvolvidos no âmbito do Programa Observatório de Educação, juntamente com as ferramentas já implantadas no INEP, e disponibilizar informações ao público sobre as ações executadas a partir deste projeto.

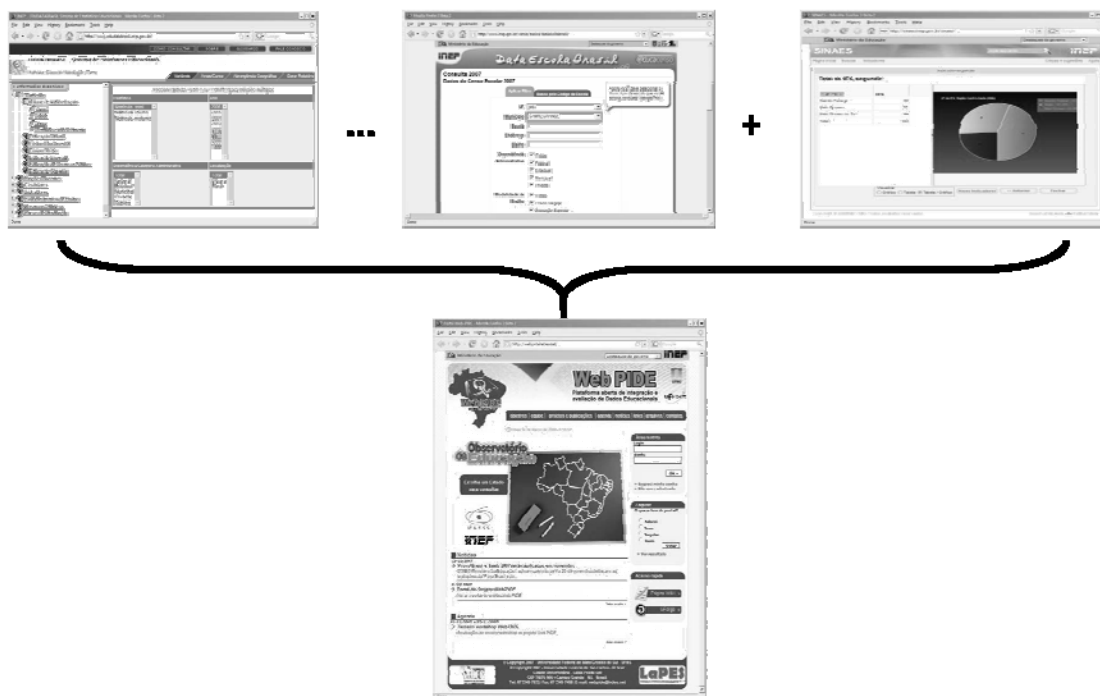


Figura 4.2: Solução integrada e centralizada no Portal Web-PIDE.

### 4.3. Funcionalidades

Analisando as especificidades apresentadas neste domínio de aplicação, focando na visualização de dados resultantes de consultas em avaliações educacionais na interface com o usuário da WebApp, pode-se especificar o seguinte caso de uso inicial sobre as funcionalidades compreendidas na abordagem proposta para desenvolver o ambiente de integração:

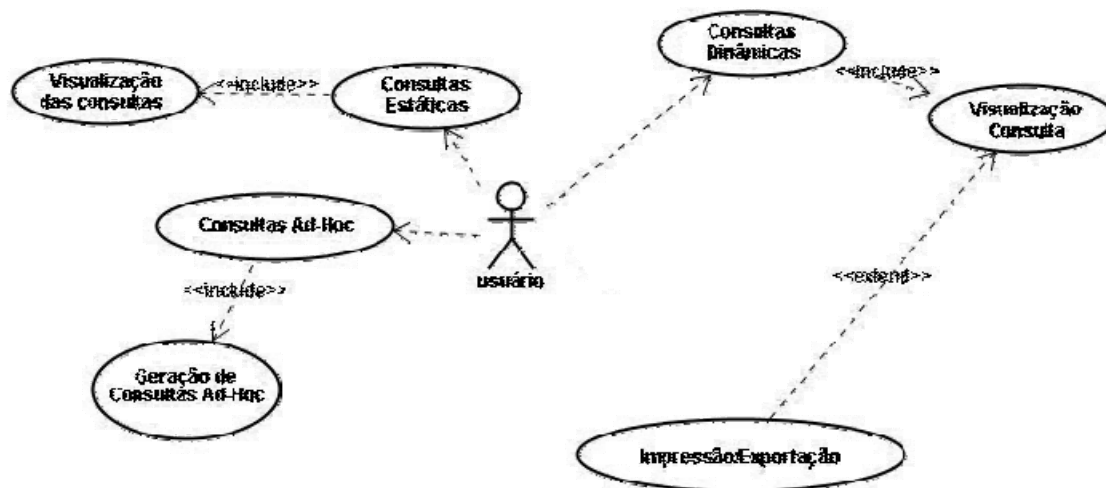


Figura 4.3: Caso de uso das funcionalidades da abordagem.

A implementação das funcionalidades definidas na abordagem é dividida em quatro etapas principais:

- Etap 1:** Levantar as fontes de dados disponíveis, bem como a estrutura e o formato dos indicadores nas diversas fontes de dados, através da utilização de uma LIDE – fornecendo uma padronização das mesmas;
- Etap 2:** Gerir recursos automatizados para especificar novas hipóteses para consulta, ou selecionar uma já existente (de uma coleção extraída de regras de associações), permitindo especificar a evolução dos dados no tempo (por ano) e geograficamente (por região, estado e município);
- Etap 3:** Visualizar os resultados da consulta (obtidas de um banco de dados geo-codificado) por meio de uma interface com o usuário que permita a extração de conhecimento; e
- Etap 4:** Exportar, através de relatórios tabulares e gráficos, os valores dos indicadores resultantes, permitindo assim, a fácil divulgação de informações extraídas ao público-alvo.

Estas etapas serão descritas detalhadamente nas seções seguintes.

### 4.3.1 Padronização das bases de dados

Como visto anteriormente, as informações das bases de dados das avaliações educacionais do INEP são disponibilizadas em formato textual de largura fixa (ASCII). Diversos são os inconvenientes do armazenamento de dados em arquivos ASCII, como a impossibilidade de realizar acesso aleatório, a inexistência de uma linguagem de consulta para extrair informações e limitações para acesso concorrente e segurança dos dados.

Portanto, é absolutamente necessário criar um mecanismo que armazene os dados das avaliações e os seus metadados. Deve-se também armazenar as correções dos dicionários, mantendo-os consistentes e promovendo seu reuso.

Para tanto, foi criado no projeto Web-PIDE a ferramenta DEAR (*Data Extractor ASCII to Relational*) [SCS2008], que atua na extração de dados e na manipulação dos metadados da extração, armazenando em um banco de dados relacional os dados oriundos de fontes sob o formato ASCII, eliminando os citados inconvenientes.

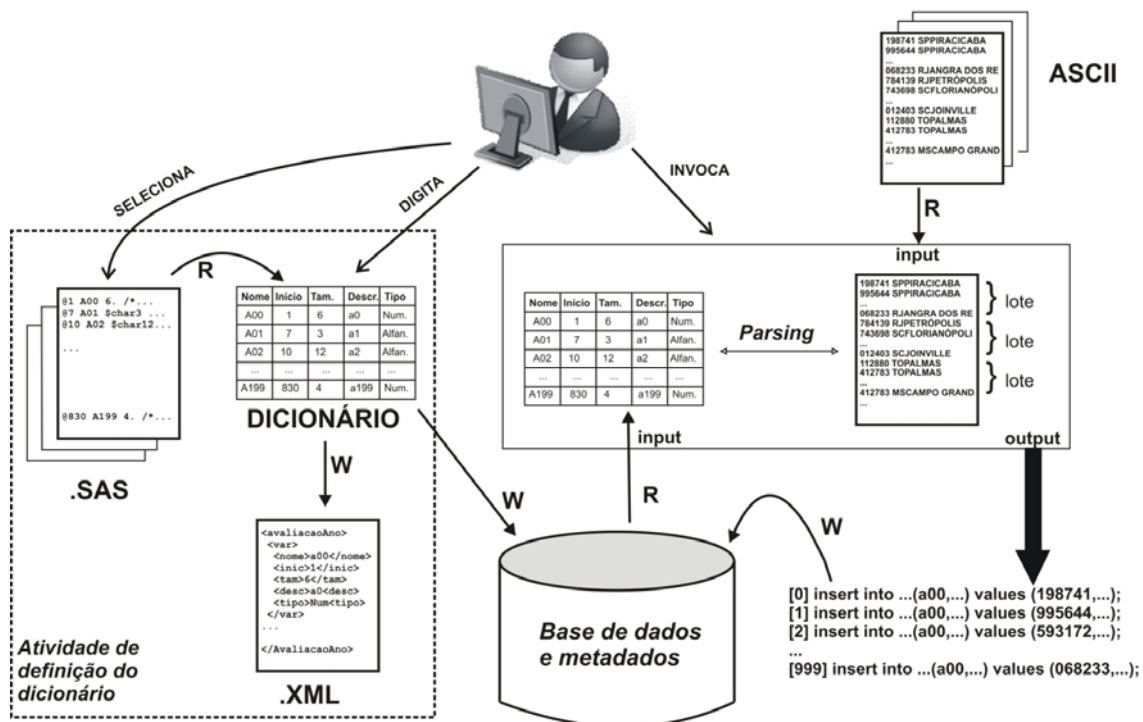


Figura 4.4: Funcionamento da ferramenta DEAR [SCS2008].

O funcionamento da ferramenta é ilustrado na Figura 4.4, onde as setas rotuladas com "W" indicam a escrita (*write*) de dados em um destino, e as rotuladas com "R" indicam a leitura (*read*) de dados a partir de uma origem.

A DEAR interpreta um arquivo ASCII segundo o dicionário de variáveis (*parsing*), que pode ser definido em tempo de execução de dois modos: pela entrada manual do usuário, ou importando o arquivo SAS correspondente. O usuário tem a opção de corrigir o dicionário que estiver inconsistente e também de selecionar apenas as variáveis de seu interesse. Depois de definido, o dicionário é armazenado em uma tabela e escrito sobre um arquivo XML, visando o seu reuso (Figura 4.5).

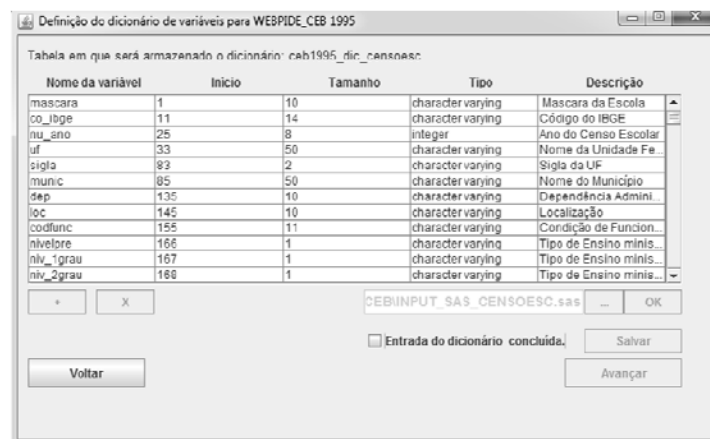


Figura 4.5: Definição de dicionários na DEAR [SCS2008]

Na medida em que se lê uma linha do arquivo de dados e interpreta as variáveis, a ferramenta cria um comando SQL (*Standard Query Language*) do tipo “INSERT INTO” (inserção) sintaticamente correto. Depois de criado, o comando é armazenado em um lote, cuja capacidade máxima é de 1000 comandos. Quando o lote fica cheio, seus comandos são então executados.

A base de dados do SAEB (uma das diversas bases de dados educacionais disponíveis na plataforma Web-PIDE), selecionada para ser utilizada neste trabalho como forma de teste e avaliação, possui uma estrutura física de banco de dados relacional, porém, a sua estrutura lógica é de um modelo multidimensional em estrela, onde o banco de dados possui uma tabela “fato” com a coleção de relacionamentos hierárquicos que são denominadas “dimensões” e campos não-chave que são denominadas “medidas” (Figura 4.6).

Esta representação lógica de um modelo multidimensional é baseada no modelo relacional denominado ROLAP (*Relational OLAP*), que é o modelo adotado como padrão neste projeto para a criação específica de um DW (*Data Warehouse*).

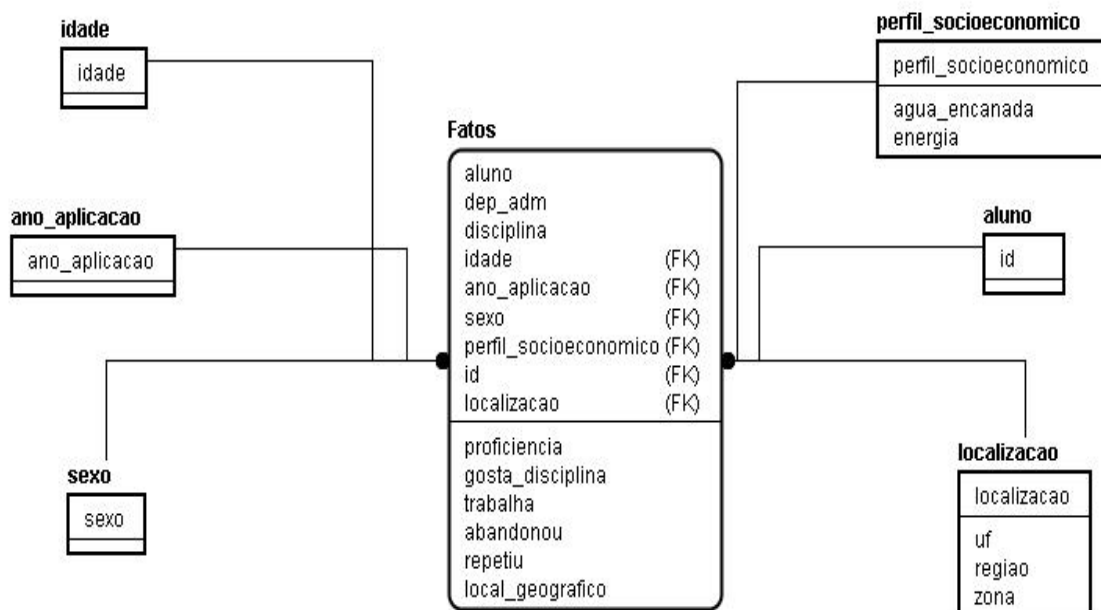


Figura 4.6: DW para base de dados do SAEB [MRC+2009].

### 4.3.2 Especificação, visualização e exportação dos dados consultados

O componente para especificação de consultas consiste em uma interface com o usuário, onde o mesmo tem acesso aos elementos das estruturas dos dados que dimensionam as informações.

Facilita a percepção das relações hierárquicas existentes entre os elementos disponíveis a se consultar, apresentando uma visão geral e, ao mesmo tempo, um detalhamento do trecho que contém o ponto de interesse ao usuário, mantendo toda a hierarquia visível.

Deste modo, pode-se realizar a especificação de uma nova consulta à base de dados por meio da seleção interativa dos elementos que filtrarão os indicadores de interesse, permitindo especificar a evolução dos dados no tempo (por ano) e geograficamente (por região, estado e município).

O sistema deve re-especificar a consulta utilizando a metalinguagem de marcação LIDE para o intercâmbio dos dados. Esta tarefa é designada para a camada de integração das WebApps.

Após a realização da consulta, o componente para visualização de dados recebe o resultado (obtidos através da camada de integração da WebApp) e disponibiliza diferentes formas de representação para estes, através de um pré-processamento à visualização, que converte uma estrutura dinâmica (em memória).

O componente deve oferecer representações dos dados em formato gráfico (em barra, linhas ou em pizza) e tabular (tabelas de listagem), o que facilita a construção de relatórios. Para o domínio de dados em avaliações educacionais, o mapeamento visual dos dados em representação geográfica é uma solução viável e muito abrangente, já que nesta forma de representação o usuário tem a noção sobre o conjunto global dos dados – permitindo assim a análise com perspectivas de suporte a decisão.

### 4.3.3 Tecnologias adotadas

Para escolha das tecnologias e ferramentas utilizadas no desenvolvimento das funcionalidades, visou-se a reutilização de *software* e adotou-se as seguintes tecnologias de *software* livre:

- PHP (<http://php.net>): utilizada para implementação da WebApp, esta linguagem foi escolhida por ser a opção em *Software* livre mais utilizada como processador de hipertextos [W3C1999];
- PostgreSQL (<http://www.postgresql.org>): a alternativa mais adotada em *Software* livre para banco de dados, pela sua robustez, seu desempenho e por sua compatibilidade com vários *Softwares* para SIG; e
- WebService: possibilita a integração de WebApps distintas, acessíveis pela rede, através de troca de mensagens baseadas em XML (*eXtensible Markup Language* [W3C1998]).

## 4.4. Camada de Integração

Para o desenvolvimento da camada de integração entre as WebApps utiliza-se o conceito de WebServices, caracterizado por integrar e conectar diferentes informações externas e internas, estejam os dados em servidores, em estações de trabalho ou até mesmo em *mainframes*. Em resumo, é uma tecnologia que permite que os dispositivos conectados à Internet troquem mensagens entre si, sem a intervenção direta dos usuários.

Um WebService é um componente que possui suas funcionalidades acessíveis pela rede através de troca de mensagens baseadas em XML (*eXtensible Markup Language* [W3C1998]). A disponibilização das operações e a descrição do serviço também ocorrem através do padrão XML. O arquivo descritor do serviço possui todas as informações necessárias para que outros componentes possam interagir com o

serviço, incluindo o formato das mensagens para as chamadas aos métodos do serviço, protocolos de comunicação e as formas de localização do serviço. Um dos maiores benefícios dessa interface é a abstração dos detalhes de implementação do serviço, permitindo que seja acessado independente da plataforma na qual foi implementado.

Como as mensagens trocadas para a comunicação são baseadas no padrão XML, também se tem flexibilidade com relação à linguagem de programação tanto na implementação do serviço quanto no componente que acessará o WebService.

Estas características permitem e motivam a implementação de WebApps baseadas em WebService por torná-las fracamente acopladas com as outras partes do código da aplicação. Com isso, as aplicações adquirem uma arquitetura componentizada e tornam-se flexíveis com relação às várias plataformas disponíveis no mercado.

Uma grande vantagem dos WebServices reside no fato de a equipe de desenvolvimento poder focar seus esforços no sistema em si, praticamente sem se preocupar com o meio de comunicação entre os processos.

Normalmente, o protocolo de troca de mensagens é o *Simple Oriented Access Protocol* (SOAP), implementado sobre o protocolo *HyperText Transfer Protocol* (HTTP) para a transferência de dados. O modelo de requisição/resposta de WebService pode ser visto na Figura 4.6.

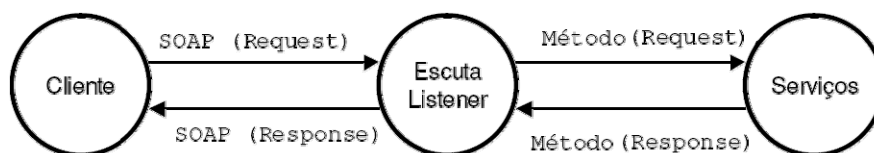


Figura 4.7: Modelo de requisição/resposta de WebService.

Para um cliente se conectar a um WebService é preciso que este disponibilize uma descrição de sua interface. Isto é feito através do documento XML de descrição de WebService, do inglês *WebService Description Language* (WSDL), no qual estão descritas informações importantes como o nome dos métodos disponíveis, parâmetros e tipos de retorno.

O WebService responsável pela interação das diversas WebApps necessita de uma especificação prévia para troca de mensagens, ou seja, a definição das mensagens possíveis para as chamadas aos métodos do serviço. Para isso, o WebService deve disponibilizar uma descrição de sua interface com estas informações.

A descrição da interface de um Webservice é feita a partir de um documento no formato XML chamado *WebService Description Language* (WSDL), onde as mensagens são definidas através de métodos com parâmetros e tipos de retorno. O arquivo deste documento WSDL encontra-se no endereço “http://webpide.ledes.net/amb-pide/webservice/?wsdl” e é descrito no Anexo I.

A partir desta descrição da interface do Webservice, as WebApps terão acesso a quatro métodos:

- *getDatabases* que retorna uma listagem, em formato XML, das bases de dados disponíveis para consultas;
- *getColumnns* que retorna as informações disponíveis sobre os indicadores disponíveis em uma determinada base de dados;
- *getData* que retorna a listagem dos indicadores da base de dados selecionadas que correspondem aos dados critérios, quando especificados; e
- *getQuery* que executa uma consulta personalizada em linguagem SQL (*Standard Query Language*) para usuários avançados.

Um modelo em camadas da arquitetura de integração disponibilizada às aplicações na Plataforma Web-PIDE através do Webservice especificado é ilustrado na Figura 4.8.

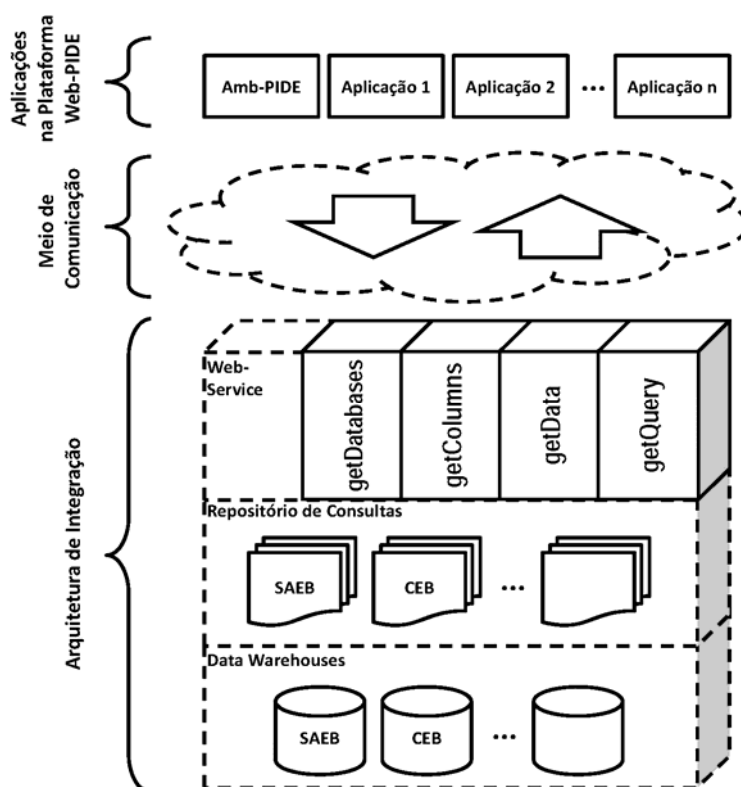


Figura 4.8: Modelo em camadas da arquitetura de integração.



#### 4.4.1 Método *getDatabases*

O método *getDatabases* é identificado pelo protocolo de troca de mensagens SOAP como “urn:amb-pide#getDatabases”<sup>6</sup>. O corpo da mensagem de requisição possui o seguinte XML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:amb-pide">
  <SOAP-ENV:Body>
    <tns:getDatabases xmlns:tns="urn:amb-pide"></tns:getDatabases>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

O *WebService* por sua vez irá responder a esta requisição com uma mensagem de resposta no seguinte formato XML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns1:getDatabasesResponse xmlns:ns1="urn:amb-pide">
      <return xsi:type="xsd:xml">
        <databases>
          <database id="saeb" text="SAEB - Sistema de Avaliação da Educação Básica"/>
          <database id="enem" text="ENEM - Exame Nacional do Ensino Médio"/>
          <database id="enade" text="ENADE - Exame Nacional de Desempenho dos Estudantes"/>
        </databases>
      </return>
    </ns1:getDatabasesResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

<sup>6</sup> Um Nome Uniforme de Recurso, URN (do inglês *Uniform Resource Name*) tem por objetivo a identificação única do recurso, de forma persistente e independente da sua localização.

Assim, a WebApp que invocou este método no Webservice terá obtido ao final da operação um XML contendo um identificador para cada base de dados disponível, bem como o rótulo da mesma.

#### 4.4.2 Método *getColumnns*

Após determinar uma base de dados utilizando o método *getDatabases*, a WebApp poderá utilizar o método *getColumnns*, que é identificado pela ação SOAP como “urn:amb-pide#getColumnns”, a fim de obter informações sobre uma dada base de dados.

Para tal, o método deve ser invocado com um parâmetro “database” do tipo texto (xsd:string) contendo o identificador do banco de dados desejado. No exemplo abaixo, o seguinte XML será enviado como mensagem de requisição passando como parâmetro “database” com o valor “saeb”:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:amb-pide">
  <SOAP-ENV:Body>
    <tns:getColumnns xmlns:tns="urn:amb-pide">
      <database xsi:type="xsd:string">saeb</database>
    </tns:getColumnns>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Como resposta, o Webservice fornecerá uma mensagem no formato XML com a definição dos dados correspondentes à base de dados especificada - neste exemplo, a base de dados do SAEB (Anexo II).

Esta mensagem XML possui em sua estrutura todas as informações necessárias para especificar uma consulta. Este passo é executado através do método *getData* do Webservice.

### 4.4.3 Método *getData*

Através do método *getData* (identificado por “urn:amb-pide#getData”), a WebApp obtém os dados disponíveis sobre um indicador na base de dados selecionada e que corresponda a determinados critérios, quando especificados.

Este método possui três parâmetros: “database” que, como no método anterior, recebe o identificador do banco de dados selecionado; “column” que recebe o identificador referente ao indicador que se deseja obter dados; e “filter”, que ao ser especificado por uma estrutura XML, filtrará os dados do indicador conforme as propriedades e os valores contidos neste parâmetro.

Portanto se, por exemplo, a WebApp necessite obter os dados disponíveis na base de dados do SAEB sobre a quantidade de alunos, na linha do tempo, que responderam à avaliação de matemática e disseram gostar desta disciplina, levando em consideração a região onde as provas foram aplicadas, o método *getData* deverá ser invocado informado o parâmetro “database” com o valor “saeb”, o parâmetro “column” com o valor “regiao” e o parâmetro “filter” com o valor “<column id='disciplina' value='Matemática' /><column id='gosta\_disciplina' value='1' />”.

A mensagem XML à ser enviada como requisição deste método, obedecendo os parâmetros do exemplo, será:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:amb-pide">
  <SOAP-ENV:Body>
    <tns:getData xmlns:tns="urn:amb-pide">
      <database xsi:type="xsd:string">saeb</database>
      <column xsi:type="xsd:string">regiao</column>
      <filter xsi:type="xsd:xml">
        <column id="disciplina" value="Matemática" />
        <column id="gosta_disciplina" value="1" />
      </filter>
    </tns:getData>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

A resposta para este método conterá uma estrutura XML com as informações necessárias sobre o indicador requisitado, respeitando os demais parâmetros. Na mensagem de resposta estão contidos os valores especificados na linha do tempo (valores totalizado por ano) de cada valor da série de valores do indicador selecionado, bem como por sua distribuição geográfica (as coordenadas indicam o centróide de uma UF). O conteúdo da mensagem de resposta para o exemplo anterior pode ser visto no Anexo III.

Estas informações são provenientes de um repositório de consultas que gerencia as requisições ao banco de dados, armazenando novas consultas e retornando consultas realizadas posteriormente, minimizando o tempo de resposta e custo de processamento. O fluxograma com a lógica empregada no repositório de consultas pode ser visto na Figura 4.9.

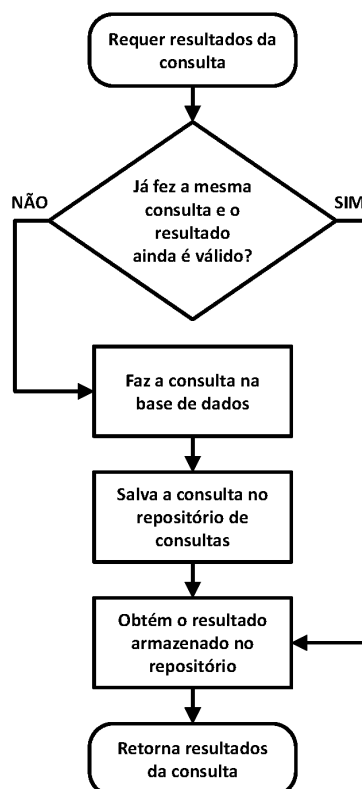


Figura 4.9: Fluxograma da lógica do repositório de consultas.

#### 4.4.4 Método *getQuery*

Através do método *getQuery* (identificado por “urn:amb-pide#getQuery”), a WebApp executará uma consulta personalizada na linguagem SQL. Este método possui apenas o parâmetros: “query” que contém a consulta no formato SQL.

#### 4.4.5 Configuração do Webservice

O Webservice possui sua configuração baseada em arquivos no formato XML, onde há um arquivo principal “databases.xml” que possui as definições de quais bases de dados estarão disponíveis para consulta. Nesta validação do Webservice, a base de dados disponível será a do SAEB, como visto anteriormente.

Assim, o conteúdo do arquivo de definição das bases disponíveis será:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<databases>
  <database id="saeb" text="SAEB - Sistema de Avaliação da Educação Básica" />
</databases>
```

Para cada base de dados especificada no arquivo principal deve-se disponibilizar um novo arquivo com as definições dos dados disponíveis. Nesse caso será disponibilizado apenas o arquivo “saeb.xml” (vide Anexo II), contendo os dados necessários para utilização do Webservice.

Basicamente, este arquivo possui os dados dos possíveis indicadores da base de dados. Estes dados são os possíveis valores que cada indicador poderá assumir, bem como um identificador único.

#### 4.5. Considerações Finais

A abordagem para integração das WebApps na plataforma Web-PIDE e a técnica de visualização de dados educacionais utilizadas constitui uma significativa alternativa, por se tratar de uma proposta inovadora, já que proporciona uma camada de integração aos demais projetos de uma forma que o ambiente projetado forneça aos institutos de pesquisa e seus gestores, uma infraestrutura visual para minerar novos conhecimentos sobre indicadores dos dados em avaliações educacionais.

# Validação da Abordagem Proposta

### 5.1. Considerações Iniciais

Neste capítulo será apresentado o ambiente intitulado **Amb-PIDE** (Ambiente de Integração e Visualização de Dados), que é a WebApp utilizada para especificação de consultas e visualização de dados provenientes do Webservice que compõe a camada de integração entre as diversas aplicações na Plataforma Web-PIDE. Além disso, a abordagem e o ambiente Amb-PIDE serão validados por meio da visualização dos indicadores das bases de dados do INEP: SAEB e CEB (Seção 5.2). Para facilitar o processo de replicação e reuso em outras bases, na Seção 5.3 é descrito um roteiro para configuração e instanciação de novas bases de dados na aplicação. Por fim, as considerações finais na Seção 5.4.

### 5.2. Proposta de Validação

O Amb-PIDE, disponível no endereço <http://webpide.ledes.net/amb-pide>, é uma aplicação Web (WebApp) que utiliza um Webservice específico para integração das diversas WebApps na plataforma Web-PIDE (camada de integração), e tem como objetivo validar a troca de mensagens que compõem o ambiente integrador.

### 5.2.1 Interface da WebApp

A partir da interface do Amb-PIDE, acessível a partir de qualquer navegador, o gestor terá à sua disposição recursos para especificar novas hipóteses de consulta à base de dados selecionada, permitindo visualizar a evolução dos dados no tempo (por ano) e geograficamente (por região, estado e município).

Inicialmente, deve-se selecionar uma das bases de dados disponível a partir da janela inicial do sistema (Figura 5.1). Para listar as bases de dados disponíveis e oferecer a opção de escolha, a WebApp deve obter previamente estas informações do Webservice por meio do método *getDatabases*.



Figura 5.1: Janela para seleção de base de dados.

Após selecionar a base de dados, uma nova janela para consulta será aberta e é organizada em três partes principais, como visto na Figura 5.2:

- A) Seleção de um indicador disponível na base de dados selecionada;
- B) Especificação dos critérios de consulta ao indicador; e
- C) Visualização dos dados da consulta gerada.

O Amb-PIDE deve exibir uma lista de indicadores disponíveis (parte A), na qual as informações necessárias à WebApp são adquiridas a partir do método *getColumnns* do Webservice de integração. Há opção para adicionar critérios personalizados na consulta (parte B).

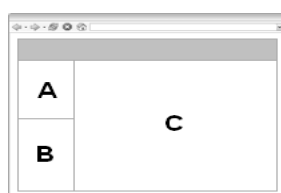


Figura 5.2: Estrutura da interface.

Destaca-se que para cada base de dados escolhida há diferentes indicadores. Como exemplo, a base de dados escolhida foi a do SAEB e os seguintes indicadores são definidos em relação ao DW projetado:

- Área;
- Disciplina;
- Série;
- Dependência administrativa;
- Estado;
- Região;
- Zona;
- Cor da pele;
- Sexo;
- Idade;
- Possui energia elétrica;
- Possui água encanada;
- Gosta da disciplina; e
- Ano de aplicação.

O que se pode visualizar e observar da hipótese a seguir: “Quais são as regiões do Brasil em que os estudantes mais responderam que gostam da disciplina de Matemática?”.

Para tal, o usuário deve selecionar na interface do Amb-PIDE, o indicador “Região” na área de seleção dos indicadores (parte A) e adicionar dois filtros (parte B): “Disciplina = Matemática” e “Gosta da disciplina = Sim”. A Figuras 5.3 ilustra o processo de especificação da consulta.



Figura 5.3: Especificação da consulta.



Após especificar a consulta desejada, o usuário deve pressionar o botão “Aplicar” para que os dados sejam gerados e visualizados na área central da WebApp (parte C). Esta ação é responsável por utilizar o Webservice para solicitar os dados referentes à consulta previamente especificada através do método *getData*.

A janela de visualização de dados do Amb-PIDE (Figura 5.4) foi desenvolvida a partir das abordagens de visualização estudadas neste trabalho, exemplificando as principais técnicas para extração de conhecimento a partir dos dados resultantes do Webservice de integração.

Esta implementação utilizou bibliotecas em linguagem JavaScript, de código aberto, e abrange as visualizações em forma tabular (DHTMLX - <http://www.dhtmlx.com>), gráfica (Highcharts - <http://www.highcharts.com>) e geográfica (Google Maps – <http://maps.google.com>).

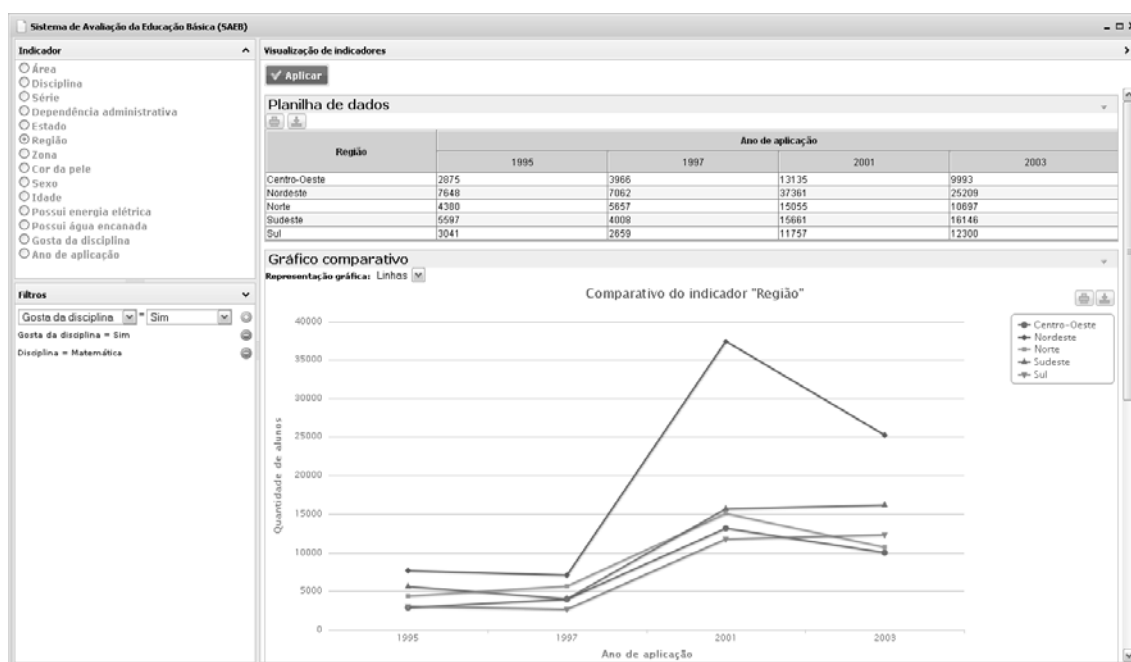


Figura 5.4: Janela de visualização da consulta.

A partir da janela de visualização da consulta pode-se abstrair as informações necessárias para responder a hipótese levantada anteriormente: a região nordeste obtém a maior quantidade de ocorrências na linha do tempo.

Os dados obtidos pela mensagem de resposta do Webservice são mapeados diretamente e visualizados em uma estrutura de tabela, dividida em duas partes: indicador e ano de aplicação (que por sua vez é subdividido em outras colunas), conforme Figura 5.5.

Região	Ano de aplicação			
	1995	1997	2001	2003
Centro-Oeste	2875	3966	13135	9993
Nordeste	7648	7062	37361	25209
Norte	4380	5657	15055	10697
Sudeste	5597	4008	15861	16146
Sul	3041	2658	11757	12300

Figura 5.5: Planilha de dados.

O formato tabular dificulta a análise de um grande número de informações, mas pode ser migrado para a forma gráfica de visualização. Plotando os valores da tabela nos eixos X (linha do tempo) e Y (série de valores do indicador), o processo de busca e compreensão de padrões é notável. Este exemplo pode ser visto ao comparar a Figura 5.5 e Figura 5.6.

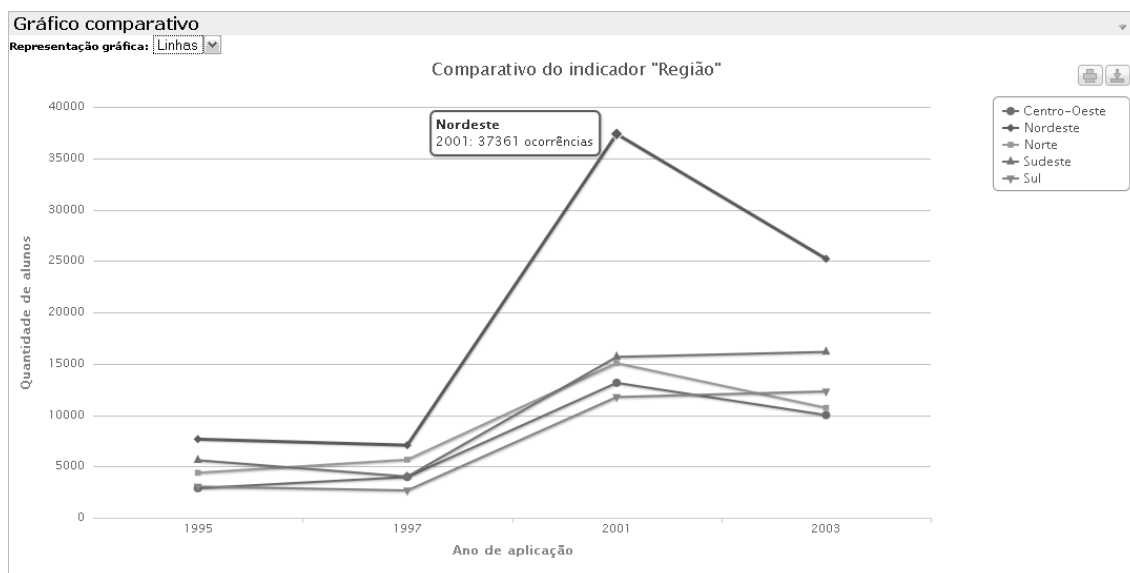


Figura 5.6: Gráfico comparativo em Linhas.

Além do gráfico em Linhas, há opção de selecionar outras formas de visualização dos valores plotados no gráfico comparativo: Barras (Figura 5.7) e Área (Figura 5.8).

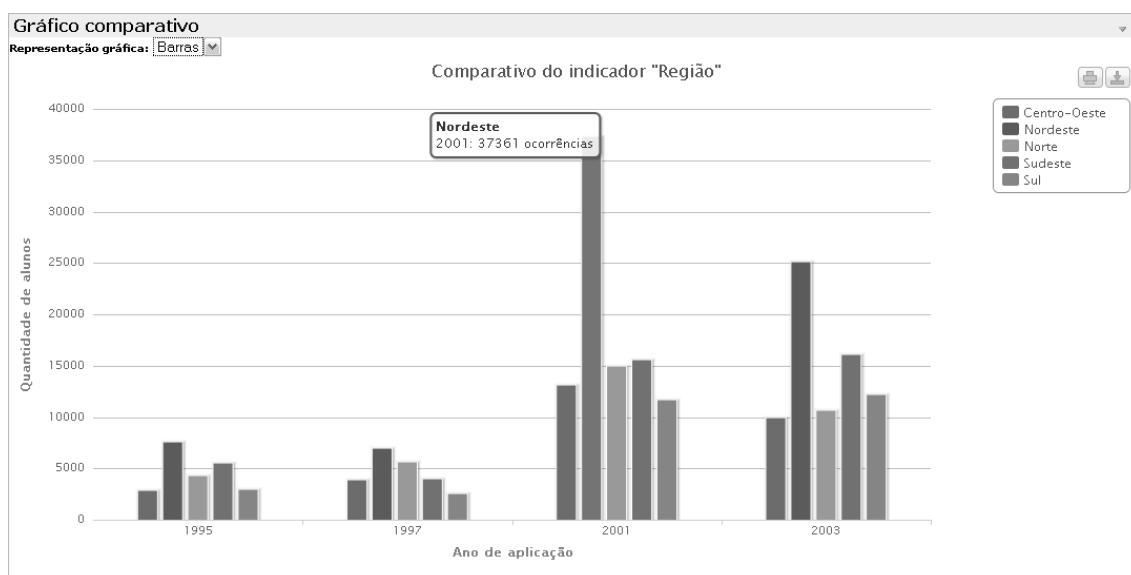


Figura 5.7: Gráfico comparativo em Barras.

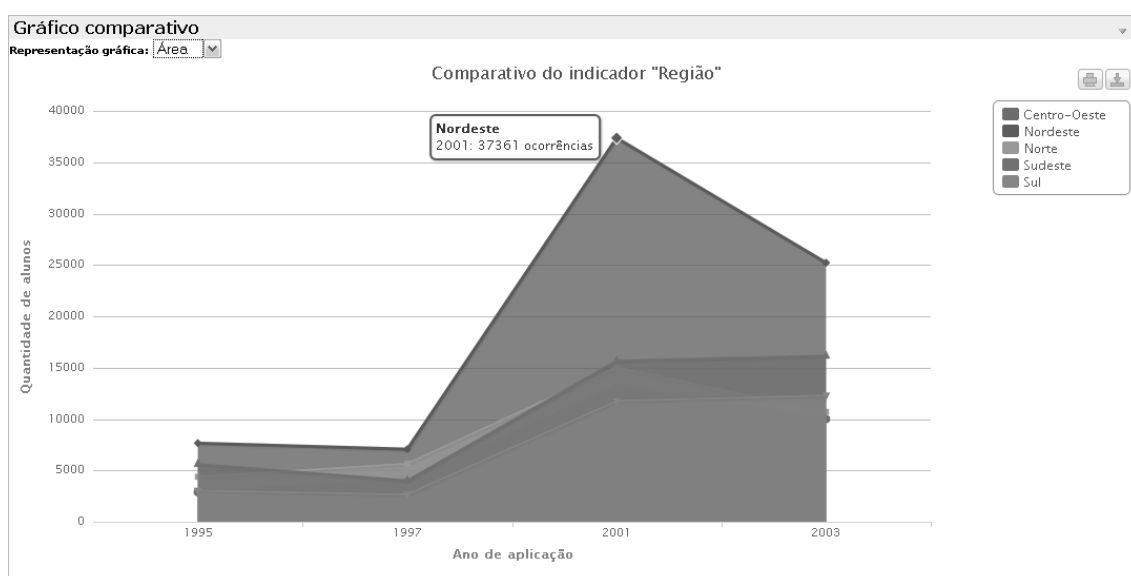


Figura 5.8: Gráfico comparativo em Área.

Os dados enviados pelo Webservice possuem uma estrutura passível de georeferenciamento. Uma forma de visualização geográfica é a inserção de “clusters” (áreas dimensionadas proporcionalmente ao valor da região pontual), permitindo uma análise mais intuitiva das aglomerações de informações (o tamanho do círculo representa a proporção numérica) e a extração de conhecimento por exploração visual (Figura 5.9).



**(3) Configuração do Webservice.** Com a modelagem do DW no padrão descrito, deve-se informar ao Webservice da camada de integração que o mesmo terá uma nova base de dados disponível. Esta operação é realizada por meio da inclusão de uma nova linha no arquivo “databases.xml” que se encontra na pasta “config” localizada na pasta raiz onde o Amb-PIDE foi instalado. Para a base de dados do Censo Escolar da Educação Básica (CEB), deve-se inserir a seguinte linha:

```
<database id="ceb" text="Censo Escolar da Educao Bsica (CEB)" />.
```

A nova linha de configuração de “database” é composta por dois atributos: “id” que corresponde a um identificador da base de dados (padronizado pela sigla correspondente ao nome da avaliação) e “text” que fornece um rótulo à base de dados disponibilizada.

As consultas ao banco de dados são realizadas por uma *view* (tabela virtual especificada a partir de uma consulta pré-escrita), que deve ser criada no esquema “dw” com o nome de “dwe” e deve conter todas as ligações necessárias entre a tabela de fatos e suas respectivas dimensões. Um exemplo de *script* para geração da *view* da base de dados do CEB pode ser visto no Anexo IV.

Os dois atributos abaixo, retornados pela *view*, devem seguir um padrão específico para que o Amb-PIDE possa interpretar, e assim, devem ser renomeados a partir da *query* utilizada na *view* através do comando “AS”:

- ano\_aplicacao: deve conter o campo responsável por categorizar os dados na linha do tempo; e
- uf: deve conter a sigla representante da Unidade Federativa que um determinado dado tem origem.

Os métodos disponibilizados pela camada de integração do Webservice necessitam de um arquivo de descrição para o novo DW (arquivo básico de LIDE) que deve ser inserido na subpasta “config/database” com o seguinte padrão de nome de arquivo: sigla.xml, onde “sigla” deve ser o conteúdo do atributo “id” da linha inserida em “databases.xml”.

O arquivo de descrição do DW deve conter uma estrutura que permita rotular os indicadores de forma legível ao usuário, no qual, por exemplo, o campo “alunos\_c\_medio” na tabela de fatos do CEB, deve ser convertido para “Quantidade de

alunos cursando o Ensino Médio”. Desta forma, a estrutura utilizada, por exemplo, para este atributo no arquivo “ceb.xml” será:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<columns>
  <column id="alunos_c_medio" text="Alunos cursando o Ensino Médio" datatype="numeric">
    <values>
      <value id="1" text="at 10"/>
      <value id="2" text="at 50"/>
      <value id="3" text="at 100"/>
      <value id="4" text="at 200"/>
      <value id="5" text="at 500"/>
      <value id="6" text="at 1.000"/>
      <value id="7" text="at 2.000"/>
      <value id="8" text="at 3.000"/>
      <value id="9" text="at 5.000"/>
      <value id="10" text="mais de 5.000"/>
      <value id="0" text="No informado"/>
    </values>
  </column>
</columns>
```

Com estas especificações, o Amb-PIDE estará apto à fornecer consultas personalizadas em sua interface.

## **5.4. Considerações Finais**

A abordagem para integração das diversas aplicações na plataforma Web-PIDE é validada a partir da WebApp intitulada Amb-PIDE: Ambiente de Integração e Visualização de Dados, por utilizar de modo satisfatório todos os métodos disponíveis da camada de integração.

# Capítulo 6

## Conclusão

### 6.1. Considerações Iniciais

A principal motivação para este trabalho foi propor ferramentas de integração e visualização para ampliar o conhecimento sobre a realidade do sistema educacional brasileiro a partir de bases de dados do INEP.

Assim, este trabalho propõe um ambiente de integração para as diferentes WebApps da plataforma Web-PIDE do Programa Observatório da Educação, contribuindo com a visualização de indicadores de dados em avaliações educacionais em um único repositório de conhecimento.

A abordagem proposta foi validada com a especificação do Webservice de integração, que disponibiliza recursos sobre os indicadores das bases de dados educacionais, testada e validada a partir da ferramenta intitulada Amb-PIDE, que utiliza as informações obtidas no Webservice em diversos contextos: visualização de dados, data warehouse e consultas avançadas.

Como resultado, um ambiente visual facilitador para consultas aos indicadores e informações correlacionadas foi disponibilizado de forma pública e aberta à comunidade na Web por meio do portal do Programa Observatório da Educação. Os usuários potenciais do portal são gestores públicos, pesquisadores, membros de organizações da sociedade civil e cidadãos que buscam por indicadores das avaliações educacionais.

## 6.2. Contribuições

As principais contribuições deste trabalho são:

1. revisão bibliográfica sobre Engenharia Web, Visualização de Informações e Webservices;
2. levantamento das soluções, tecnologias e ferramentas existentes no portal do INEP que disponibilizam as bases de dados e possibilitam consultas aos indicadores existentes;
3. identificação das tecnologias de Engenharia de Software capazes de viabilizar a integração das técnicas de reuso;
4. especificação de uma abordagem para integração e visualização de dados educacionais;
5. especificação e implementação do WebService integrador das diversas WebApps na plataforma Web-PIDE; e
6. implementação do ambiente de integração e visualização de dados intitulado Amb-PIDE, responsável por validar a abordagem proposta.

## 6.3. Trabalhos Futuros

A partir dos resultados deste trabalho, pode-se explorar diversas pesquisas a fim de ampliar a abrangência dos resultados obtidos ou reutilizando este trabalho a fim de obter novos resultados, tais como:

1. aprimorar ou implementar novas formas para visualização dos dados obtidos através do WebService de integração, bem como combiná-las à fim de comparar os resultados obtidos;
2. representar o conhecimento das bases de dados utilizando Redes Semânticas e ontologias a fim de melhorar a busca e interpretação das informações visualizadas;
3. viabilizar a utilização do WebService de integração de WebApps a outros domínios de aplicação em Governo Eletrônico (eGov);



4. gerar novas WebApps com finalidade de obter e analisar os dados disponibilizados no ambiente de integração através de algoritmos para extração de conhecimento (KDD); e
5. disponibilizar acesso ao ambiente de integração à outras mídias de visualização, como *set-top box* (TV digital interativa) e dispositivos móveis (celulares e *handhelds*).

# Referências Bibliográficas

- [ALM2003] ALMEIDA, M. O. **Uma Ferramenta para Mineração Visual de Dados Usando Mapas em Árvores e suas Aplicações**. Dissertação de Mestrado, Núcleo de Pesquisa em Redes e Computação da Universidade Salvador, UNIFACS, Salvador, BA, Brasil, 2003.
- [AP1991] ARANGO, G.; PRIETO-DÍAZ, R. **Domain analysis concepts and research directions**. Los Alamitos: IEEE Computer Society Press, 1991.
- [BMM+1999] BOSCH, J.; MOLIN, P.; MATTSSON, M.; BENGTSSON, P.; FAYAD, M. **Framework problem and experiences**. In: M. Fayad, R. Johnson, D. Schmidt. Building Application Frameworks: Object Oriented Foundations of Framework Design. John Willey and Sons, p. 55-82, 1999.
- [BUD1996] BUDD, T. **Introduction to Object-Oriented Programming**. 2nd ed. Addison-Wesley, 1996.
- [CFP+2002] CÂMARA, G.; FERREIRA, K. R.; PAIVA, J. A. C.; QUEIROZ, G. R.; VINHAS, L. **Programação Genérica Aplicada a Algoritmos Geográficos**. Proceedings of the IV Brazilian Symposium on GeoInformatics - GeoInfo, São Paulo, SP, Brasil, 2002.
- [CMP+2004] CAMARGO, V. V.; MASIERO, P. C.; PENTEADO, R. D.; RAMOS, R. A. **Reuso da Implementação Orientada a Aspectos do Padrão de Projeto Camada de Persistência**. Anais da IV Conferência Latino-Americana em Linguagens de Padrões para Programação - SugarLoafPLOP. Porto das Dunas, CE, Brasil, 2004.
- [CMS1999] CARD, S. K.; MACKINLAY, J. D.; SHNEIDERMAN, B. **Information Visualization**. Readings in Information Visualization: Using Vision to Think, Morgan Kaufmann Publishers, p. 1-34, San Francisco, California, USA, 1999.
- [CT2005] CARROMEU, C.; TURINE, M. A. S. **Um framework de aplicações web de apoio a gestão de fomento**. Workshop de Teses e Dissertações do XI Simpósio Brasileiro de Sistemas Multimídia e Web - WebMedia. Poços de Caldas, MG, Brasil, 2005.
- [CAR2005] CARVALHO, S. T. **Frameworks e Patterns**. Jornal Opção, Suplemento de Informática, Ano XXX, Edições 1558/1559/1560, Goiânia, GO, Brasil, 2005.
- [CAV2002] CAVA, R. A. **Bifocal Tree: Uma Técnica para Visualização de Estruturas Hierárquicas**. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, UFRGS, Porto Alegre, RS, Brasil, 2002.

- [CR2001] CERQUEIRA, C. A.; RIGOTTI, J. I. R. **As bases de dados do INEP e os indicadores educacionais: conceitos e aplicações**. Proceedings of the International Union for Scientific Study of Population - IUSSP, Salvador, BA, Brasil, 2001.
- [COE2006] COELHO, A. H. **Montagem de um SIG em software livre**. Congresso Brasileiro de Cadastro Técnico Multifinalitário - COBRAC 2006. Departamento de Engenharia Civil da Universidade Federal de Santa Catarina, UFSC, Florianópolis, SC, Brasil, 2006.
- [DFH+2003] DUMOULIN, C.; FRANCISCUS, G.; HUSTED, T.; WINTERFELDT, D. **Struts in Action**. 1st ed. Connecticut, USA: Manning Publications Co, 2003.
- [HER2009] HERNANDES, E. C. M. **Um processo automatizado para tratamento de dados e conceitualização de ontologias com apoio de visualização**. Dissertação de Mestrado, Universidade Federal de São Carlos, UFScar, São Carlos, SP, Brasil, 2009.
- [FGW2002] FAYYAD, U.; GRINSTEIN, G.; WIERSE, A. **Information Visualization in Data Mining and Knowledge Discovery**. Morgan Kaufmann, 1st ed., 2002.
- [FID2005] FIDALGO, R. N. **Uma Infraestrutura para Integração de Modelos, Esquemas e Serviços Multidimensionais e Geográficos**. Tese de Doutorado, Centro de Informática da Universidade Federal de Pernambuco, UFPE, Recife, PE, Brasil, 2005.
- [FS2001] FRANCA, L. P. A.; STAA, A. **Geradores de Artefatos: Implementação e Instanciação de Frameworks**. Anais do XV Simpósio Brasileiro de Engenharia de Software - SBES, Rio de Janeiro, RJ, Brasil, 2001.
- [GM2001] GINIGE, A.; MURUGESAN, S.; **Web Engineeing: an Introduction**, Proceedings of IEEE Multimedia, v. 8, p. 14-18, 2001.
- [GMS2008] GONDIM, R. P.; MACHADO, M. C. C.; SANTOS, M. S. **VIDEWeb: Um Ambiente para Visualização de Dados Educacionais na Plataforma Web-PIDE**. Projeto de Graduação, Departamento de Computação e Estatística da Universidade Federal de Mato Grosso do Sul, UFMS, Campo Grande, MS, Brasil, 2008.
- [GOO2006] GONÇALVES, J. J.; OLIVEIRA, K. M.; OLIVEIRA, T. M. A. **Métricas de Reusabilidade para Componentes de Softwares**. Anais do VI Workshop de Desenvolvimento Baseado em Componentes - WDBC. Recife, PE, Brasil, 2006.
- [GUE2004] GUEDES, G. T. A. **UML, Uma Abordagem Prática**. Editora Novatec, 2004.
- [JAN2001] JANNUZZI, M. P. **Indicadores Sociais no Brasil: Conceitos, Fontes de Dados e Aplicações**. Alínea Editora, 2001.
- [KAP2004] KAPPEL, G. **"Web Engineering - Old Wine in New Bottles?"**. Proceedings of the International Conference on Web Engineering - ICWE, Munich, 2004.

- [KNS2004] KREUSELER, M.; NOCKE, T.; SCHUMANN, H. **A History Mechanism for Visual Data Mining**. Proceedings of the IEEE Symposium on Information Visualization - INFOVIS, Austin, TX, USA, p. 49-56, 2004.
- [LM1991] LAKATOS, E. M.; MARCONI, M. A. **Fundamentos de metodologia científica**. 3rd ed. São Paulo: Atlas, 1991.
- [MBG+2002] MALDONADO, J. C.; BRAGA, R. T. V.; GERMANO, F. S. R.; MASIERO, P. C. **Padrões e Frameworks de Software**. Notas Didáticas, Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, ICMC/USP, São Paulo, SP, Brasil, 2002.
- [MRC+2009] MOTA, F. M.; ROSSI, L. L.; CAMPELO, A. L.; ARAGÃO, A. L.; MENDES, L. P. Relatório técnico sobre Data Warehouse do SAEB. Universidade Federal do Mato Grosso do Sul - UFMS/CPCX, Coxim, MS, 2009.
- [MRW2001] MALDONADO, J.C.; ROCHA, A.R.C.; WEBER, K.C. **Qualidade de Software: Teoria e Prática**. ed. Prentice Hall, 2001.
- [NW1996] NIELSEN, J.; WAGNER, A. **User Interface Design for the WWW**. Proceedings of Computer Human Interface Conference on Human Factors in Computing Systems, ACM Press, p. 330-331, 1996.
- [OL2003] OLIVEIRA, M. C. F.; LEVKOWITZ, H. **From Visual Data Exploration to Visual Data Mining: A Survey**. Proceedings of the IEEE Transactions on Visualization and Computer Graphics - TVCG, v. 9, n. 3, p. 378-394, 2003.
- [PAE1999] PAEZ, R. A. **Un acercamiento a la reutilización en ingeniería de software**. EAFIT Magazine, Universidad de EAFIT, Medellín, Colombia, n. 114, p. 45-63, 1999.
- [PRE2005] PRESSMAN, R. S. **Software Engineering: A Practitioner's Approach**. 6th ed. New York, USA: McGraw Hill, 2005.
- [RT2001] RICCA, F.; TONELLA, P. **Analysis and Testing of Web Applications**. Proceedings of the 23rd International Conference on Software Engineering - ICSE, 2001.
- [RF2003] RIOS, E.; FILHO, T.M. **Projeto & Engenharia de Software – Teste de Software**. ed. Alta Books LTDA, 2003.
- [SIL2000] SILVA, R. P. **Suporte ao desenvolvimento e uso de frameworks e componentes**. Tese de Doutorado, Instituto de Informática da Universidade do Rio Grande do Sul, UFSC, Porto Alegre, RS, Brasil, 2000.
- [SIQ2009] SIQUEIRA, T. L. L. **SB-Index: Um Índice Espacial Baseado em Bitmap para Data Warehouse Geográfico**. Dissertação de Mestrado, Universidade Federal de São Carlos, UFScar, São Carlos, SP, Brasil, 2009.

- [SCS2008] SIQUEIRA, T. L. L.; CIFERRI, R. R.; SANTOS, M. T. P. **Projeto, Construção e Manutenção de Data Warehouses para auxiliar o Planejamento de Políticas Públicas de Educação.** In: XVI Jornadas de Jóvenes Investigadores, 2008, Montevideo, Uruguay. XVI Jornadas de Jóvenes Investigadores - Trabajos Completos, 2008. p. 1016-1025.
- [SOA2002] SOARES, V. G. **GeoVisual - Um Ambiente de Consultas Visuais para Bancos de Dados Geográficos.** Tese de Doutorado, Centro de Informática da Universidade Federal de Pernambuco, UFPE, Recife, PE, Brasil, 2002.
- [TCM+2006] TURINE, M. A. S.; CÁCERES, E. N.; MONGELLI, H.; PAIVA, D. M. B.; FABBRI, S.; ROCHA, M. G. B.; CIFERRI, R. R.; SANTOS, M. T. P. **Web-PIDE: uma plataforma aberta de integração e avaliação de dados educacionais.** São Carlos: DC-UFSCar; Campo Grande: DCT-UFMS. Observatório da Educação, Brasília, 2006. Projeto aprovado no Edital n.001/2006/INEP/CAPES.
- [TID2008] TIDWELL, J. IU Patterns and Techniques. Disponível em: <<http://time-tripper.com/uipatterns/>>. Acesso em: 15 fev. 2008.
- [TOG2008] TOGNOZZI, B. **First Principles of Interaction Design.** Disponível em: <<http://www.asktog.com/basics/firstPrinciples.html>>. Acesso em: 20 fev. 2008.
- [TLR2005] TURINE, M. A. S.; LIMA, J. E. O.; ROMÃO, R. **Uma Arquitetura Baseada em Componentes para um Sistema de Informação em Gestão Social.** XI Simpósio Brasileiro de Sistemas Multimídia e Web - WebMedia. Sociedade Brasileira de Computação, Poços de Caldas, Porto Alegre, Brasil, 2005.
- [TOI+2004] TURINE, M. A. S.; OLIVEIRA, M.; ISHY, E.; BLANES, D.; ACOSTA, A. R.; WANDERLEY, M. B. **SIGS: Uma Tecnologia Social na Web para Gestão, Monitoramento e Avaliação de Programas Sociais.** Conferência IADIS Ibero-Americana WWW/Internet 2004, v. 1, p. 437-440, Madrid, 2004.
- [W3C1998] W3C - World Wide Web Consortium XML Activity. **Extensible Markup Language (XML) 1.0 Specification, 1998.** Disponível em: <<http://www.w3.org/TR/REC-xml>>. Acesso em: 14 jan. 2008.
- [W3C1999] W3C - World Wide Web Consortium HTML Activity. **Hyper Text Markup Language 4.01 Specification, 1999.** Disponível em: <<http://www.w3.org/TR/REC-html40>>. Acesso em: 11 jan. 2008.

# **ANEXOS**

## ANEXO I - Arquivo de descrição do Webservice

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:amb-pi de"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:amb-pi de">
  <types>
    <xsd:schema targetNamespace="urn:amb-pi de">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/"
/>
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
    </xsd:schema>
  </types>
  <message name="getDatabasesRequest"></message>
  <message name="getDatabasesResponse">
    <part name="return" type="xsd:string" />
  </message>
  <message name="getColumnsRequest">
    <part name="database" type="xsd:string" />
  </message>
  <message name="getColumnsResponse">
    <part name="return" type="xsd:string" />
  </message>
  <message name="getDataRequest">
    <part name="database" type="xsd:string" />
    <part name="column" type="xsd:string" />
    <part name="filter" type="xsd:string" />
  </message>
  <message name="getDataResponse">
    <part name="return" type="xsd:string" />
  </message>
  <portType name="amb-pi dePortType">
    <operation name="getDatabases">
      <documentation>Retorna a listagem dos bancos de dados disponíveis
para consulta. </documentation>
      <input message="tns:getDatabasesRequest"/>
      <output message="tns:getDatabasesResponse"/>
    </operation>
    <operation name="getColumns">
      <documentation>Retorna a listagem das colunas disponíveis em banco
de dados especificado. </documentation>
      <input message="tns:getColumnsRequest"/>
      <output message="tns:getColumnsResponse"/>
    </operation>
    <operation name="getData">
      <documentation>Retorna os dados dos indicadores de um determinado
banco de dados. </documentation>
      <input message="tns:getDataRequest"/>
      <output message="tns:getDataResponse"/>
    </operation>
  </portType>
  <binding name="amb-pi deBinding" type="tns:amb-pi dePortType">
    <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getDatabases">
      <soap:operation soapAction="urn:amb-pi de#getDatabases"
style="rpc"/>
      <input>
        <soap:body use="encoded" namespace="urn:amb-pi de"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
      </input>
      <output>
        <soap:body use="encoded" namespace="urn:amb-pi de"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
      </output>
    </operation>
    <operation name="getColumns">
      <soap:operation soapAction="urn:amb-pi de#getColumns" style="rpc"/>
      <input>
```

```

        <soap:body use="encoded" namespace="urn:amb-pi de"
encodingStyle="http://schemas.xml soap.org/soap/encoding/" />
    </input>
    <output>
        <soap:body use="encoded" namespace="urn:amb-pi de"
encodingStyle="http://schemas.xml soap.org/soap/encoding/" />
    </output>
</operation>
<operation name="getData">
    <soap:operation soapAction="urn:amb-pi de#getData" style="rpc" />
    <input>
        <soap:body use="encoded" namespace="urn:amb-pi de"
encodingStyle="http://schemas.xml soap.org/soap/encoding/" />
    </input>
    <output>
        <soap:body use="encoded" namespace="urn:amb-pi de"
encodingStyle="http://schemas.xml soap.org/soap/encoding/" />
    </output>
</operation>
</binding>
<service name="amb-pi de">
    <port name="amb-pi dePort" binding="tns:amb-pi deBinding">
        <soap:address location="http://webpi de.ledes.net/amb-
pi de/webservice/" />
    </port>
</service>
</definitions>

```



## ANEXO II - Resposta do Webservice para a base do SAEB

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns1:getColumnsResponse xmlns:ns1="urn:amb-pide">
      <return xsi:type="xsd:xml">
        <columns>
          <column id="area" text="Área" datatype="text">
            <values>
              <value id="Exatas" text="Exatas"/>
              <value id="Humanas" text="Humanas"/>
            </values>
          </column>
          <column id="disciplina" text="Disciplina" datatype="text">
            <values>
              <value id="Matemática" text="Matemática"/>
              <value id="Português" text="Português"/>
            </values>
          </column>
          <column id="serie" text="Série" datatype="numeric">
            <values>
              <value id="3" text="Terceira série"/>
              <value id="4" text="Quarta série"/>
              <value id="8" text="Oitava série"/>
            </values>
          </column>
          <column id="dep_adm" text="Dependência administrativa"
datatype="text">
            <values>
              <value id="Não Informado" text="Não Informado"/>
              <value id="Federal" text="Federal"/>
              <value id="Estadual" text="Estadual"/>
              <value id="Municipal" text="Municipal"/>
              <value id="Particular" text="Particular"/>
            </values>
          </column>
          <column id="uf" text="Estado" datatype="text">
            <values>
              <value id="AC" text="Acre"/>
              <value id="AL" text="Alagoas"/>
              <value id="AM" text="Amazonas"/>
              <value id="AP" text="Amapá"/>
              <value id="BA" text="Bahia"/>
              <value id="CE" text="Ceará"/>
              <value id="DF" text="Distrito Federal"/>
              <value id="ES" text="Espírito Santo"/>
              <value id="GO" text="Goiás"/>
              <value id="MA" text="Maranhão"/>
              <value id="MG" text="Minas Gerais"/>
              <value id="MS" text="Mato Grosso do Sul"/>
              <value id="MT" text="Mato Grosso"/>
              <value id="PB" text="Paraíba"/>
              <value id="PE" text="Pernambuco"/>
              <value id="PI" text="Piauí"/>
              <value id="PR" text="Paraná"/>
              <value id="RJ" text="Rio de Janeiro"/>
              <value id="RN" text="Rio Grande do Norte"/>
              <value id="RO" text="Rondônia"/>
              <value id="RR" text="Roraima"/>
              <value id="RS" text="Rio Grande do Sul"/>
              <value id="SC" text="Santa Catarina"/>
              <value id="SE" text="Sergipe"/>
              <value id="SP" text="São Paulo"/>
              <value id="TO" text="Tocantins"/>
              <value id="PA" text="Pará"/>
            </values>
          </column>
        </columns>
      </return>
    </ns1:getColumnsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

</column>
<column id="regiao" text="Região" datatype="text">
  <values>
    <value id="Norte" text="Norte"/>
    <value id="Nordeste" text="Nordeste"/>
    <value id="Sul" text="Sul"/>
    <value id="Sudeste" text="Sudeste"/>
    <value id="Centro-Oeste" text="Centro-Oeste"/>
  </values>
</column>
<column id="zona" text="Zona" datatype="text">
  <values>
    <value id="Não Informado" text="Não Informado"/>
    <value id="Rural" text="Rural"/>
    <value id="Urbana" text="Urbana"/>
  </values>
</column>
<column id="cor" text="Cor da pele" datatype="text">
  <values>
    <value id="Não Informado" text="Não Informado"/>
    <value id="Amarelo" text="Amarelo"/>
    <value id="Branco" text="Branco"/>
    <value id="Índigena" text="Índigena"/>
    <value id="Pardo" text="Pardo"/>
    <value id="Negro" text="Negro"/>
  </values>
</column>
<column id="sexo" text="Sexo" datatype="text">
  <values>
    <value id="Não Informado" text="Não Informado"/>
    <value id="Feminino" text="Feminino"/>
    <value id="Masculino" text="Masculino"/>
  </values>
</column>
<column id="idade" text="Idade" datatype="numeric">
  <values>
    <value id="-1" text="Não Informado"/>
    <value id="7" text="7"/>
    <value id="8" text="8"/>
    <value id="9" text="9"/>
    <value id="10" text="10"/>
    <value id="11" text="11"/>
    <value id="12" text="12"/>
    <value id="13" text="13"/>
    <value id="14" text="14"/>
    <value id="15" text="15"/>
    <value id="16" text="16"/>
    <value id="17" text="17"/>
    <value id="18" text="18"/>
    <value id="19" text="19"/>
    <value id="20" text="20"/>
    <value id="21" text="21"/>
    <value id="22" text="22"/>
    <value id="23" text="23"/>
    <value id="24" text="24"/>
    <value id="25" text="25"/>
    <value id="26" text="26"/>
    <value id="27" text="27"/>
    <value id="28" text="28"/>
  </values>
</column>
<column id="energia" text="Possui energia elétrica"
datatype="boolean">
  <values>
    <value id="-1" text="Não Informado"/>
    <value id="1" text="Sim"/>
    <value id="0" text="Não"/>
  </values>
</column>
<column id="agua_encanada" text="Possui água encanada"
datatype="boolean">
  <values>
    <value id="-1" text="Não Informado"/>
    <value id="1" text="Sim"/>
    <value id="0" text="Não"/>
  </values>
</column>

```

```

        <column id="gosta_disciplina" text="Gosta da disciplina"
datatype="boolean">
        <values>
            <value id="-1" text="Não Informado"/>
            <value id="1" text="Sim"/>
            <value id="0" text="Não"/>
        </values>
    </column>
    <column id="ano_aplicacao" text="Ano de aplicação"
datatype="numeric">
        <values>
            <value id="1995" text="1995"/>
            <value id="1997" text="1997"/>
            <value id="1999" text="1999"/>
            <value id="2001" text="2001"/>
            <value id="2003" text="2003"/>
        </values>
    </column>
</columns>
</return>
</ns1:getColumnsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## ANEXO III - Resposta do Webservice para uma consulta

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns1:getDataResponse xmlns:ns1="urn:amb-pide">
      <return xsi:type="xsd:xml">
        <data>
          <column id="regiao" text="Região"/>
          <series>
            <serie id="Centro-Oeste" text="Centro-Oeste">
              <timeline>
                <value id="1995" total="2875"/>
                <value id="1997" total="3966"/>
                <value id="2001" total="13135"/>
                <value id="2003" total="9993"/>
              </timeline>
              <geo>
                <value id="DF" text="Distrito Federal"
total="4569" coordinates="-15.68, -47.91"/>
                <value id="GO" text="Goiás" total="8212"
coordinates="-15.87, -50.34"/>
                <value id="MS" text="Mato Grosso do Sul"
total="9245" coordinates="-20.40, -54.65"/>
                <value id="MT" text="Mato Grosso" total="7943"
coordinates="-12.83, -55.94"/>
              </geo>
            </serie>
            <serie id="Nordeste" text="Nordeste">
              <timeline>
                <value id="1995" total="7648"/>
                <value id="1997" total="7062"/>
                <value id="2001" total="37361"/>
                <value id="2003" total="25209"/>
              </timeline>
              <geo>
                <value id="AL" text="Alagoas" total="7907"
coordinates="-9.70, -36.54"/>
                <value id="BA" text="Bahia" total="10894"
coordinates="-12.31, -41.53"/>
                <value id="CE" text="Ceará" total="8935"
coordinates="-4.93, -39.55"/>
                <value id="MA" text="Maranhão" total="9280"
coordinates="-5.85, -45.57"/>
                <value id="PB" text="Paraíba" total="8623"
coordinates="-7.16, -36.76"/>
                <value id="PE" text="Pernambuco" total="9381"
coordinates="-8.42, -38.14"/>
                <value id="PI" text="Piauí" total="8035"
coordinates="-7.64, -42.60"/>
                <value id="RN" text="Rio Grande do Norte"
total="7297" coordinates="-5.70, -36.61"/>
                <value id="SE" text="Sergipe" total="6928"
coordinates="-10.65, -37.42"/>
              </geo>
            </serie>
            <serie id="Norte" text="Norte">
              <timeline>
                <value id="1995" total="4380"/>
                <value id="1997" total="5657"/>
                <value id="2001" total="15055"/>
                <value id="2003" total="10697"/>
              </timeline>
              <geo>
                <value id="AC" text="Acre" total="3905"
coordinates="-8.88, -70.80"/>
                <value id="AM" text="Amazonas" total="6948"
coordinates="-4.95, -64.62"/>
              </geo>
            </serie>
          </series>
        </data>
      </return>
    </ns1:getDataResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

    coordi nates="1. 48, -51. 75" />      <val ue id="AP" text="Amapá" total ="3841"
    coordi nates="-11. 04, -63. 02" />    <val ue id="RO" text="Rondônia" total ="5998"
    coordi nates="1. 74, -61. 22" />    <val ue id="RR" text="Roraima" total ="2392"
    coordi nates="-10. 74, -48. 27" />   <val ue id="TO" text="Tocantins" total ="4094"
    coordi nates="-5. 57, -52. 47" />    <val ue id="PA" text="Pará" total ="8611"
    </geo>
  </serie>
  <serie id="Sudeste" text="Sudeste">
    <time line>
      <val ue id="1995" total ="5597" />
      <val ue id="1997" total ="4008" />
      <val ue id="2001" total ="15661" />
      <val ue id="2003" total ="16146" />
    </time line>
    <geo>
      <val ue id="ES" text="Espírito Santo"
total ="7977" coordi nates="-19. 64, -40. 58" />
      <val ue id="MG" text="Mina s Gerai s"
total ="12065" coordi nates="-17. 76, -44. 30" />
      <val ue id="RJ" text="Rio de Janei ro"
total ="9585" coordi nates="-22. 18, -42. 30" />
      <val ue id="SP" text="Sao Paul o" total ="11785"
coordi nates="-22. 10, -48. 98" />
    </geo>
  </serie>
  <serie id="Sul " text="Sul ">
    <time line>
      <val ue id="1995" total ="3041" />
      <val ue id="1997" total ="2659" />
      <val ue id="2001" total ="11757" />
      <val ue id="2003" total ="12300" />
    </time line>
    <geo>
      <val ue id="PR" text="Paraná" total ="10313"
coordi nates="-24. 60, -51. 86" />
      <val ue id="RS" text="Rio Grande do Sul "
total ="10374" coordi nates="-29. 57, -53. 50" />
      <val ue id="SC" text="Santa Catari na"
total ="9070" coordi nates="-27. 13, -50. 65" />
    </geo>
  </serie>
</series>
</data>
</return>
</ns1: getDataResponse>
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

## ANEXO IV - Script para gerar a view do CEB

```
CREATE VIEW "dw"."dwe" AS
SELECT pe.pre_escola, ef.ens_fundamental, em.ens_medio, g.regiao, g.uf,
       g.zona_escola, e.dep_admin, e.cond_func, t.anocenso AS ano_aplicacao,
CASE
  WHEN f.funcionarios > 0 AND f.funcionarios <= 10 THEN 1
  WHEN f.funcionarios > 10 AND f.funcionarios <= 50 THEN 2
  WHEN f.funcionarios > 50 AND f.funcionarios <= 100 THEN 3
  WHEN f.funcionarios > 100 AND f.funcionarios <= 200 THEN 4
  WHEN f.funcionarios > 200 AND f.funcionarios <= 500 THEN 5
  WHEN f.funcionarios > 500 AND f.funcionarios <= 1000 THEN 6
  WHEN f.funcionarios > 1000 AND f.funcionarios <= 2000 THEN 7
  WHEN f.funcionarios > 2000 AND f.funcionarios <= 5000 THEN 8
  WHEN f.funcionarios > 5000 AND f.funcionarios <= 10000 THEN 9
  WHEN f.funcionarios > 10000 THEN 10
  ELSE 0
END AS funcionarios,
CASE
  WHEN f.professores > 0 AND f.professores <= 10 THEN 1
  WHEN f.professores > 10 AND f.professores <= 50 THEN 2
  WHEN f.professores > 50 AND f.professores <= 100 THEN 3
  WHEN f.professores > 100 AND f.professores <= 200 THEN 4
  WHEN f.professores > 200 AND f.professores <= 500 THEN 5
  WHEN f.professores > 500 AND f.professores <= 1000 THEN 6
  WHEN f.professores > 1000 AND f.professores <= 2000 THEN 7
  WHEN f.professores > 2000 AND f.professores <= 5000 THEN 8
  WHEN f.professores > 5000 AND f.professores <= 10000 THEN 9
  WHEN f.professores > 10000 THEN 10
  ELSE 0
END AS professores,
CASE
  WHEN f.matriculasp_re > 0 AND f.matriculasp_re <= 10 THEN 1
  WHEN f.matriculasp_re > 10 AND f.matriculasp_re <= 100 THEN 2
  WHEN f.matriculasp_re > 100 AND f.matriculasp_re <= 500 THEN 3
  WHEN f.matriculasp_re > 500 AND f.matriculasp_re <= 1000 THEN 4
  WHEN f.matriculasp_re > 1000 AND f.matriculasp_re <= 5000 THEN 5
  WHEN f.matriculasp_re > 5000 AND f.matriculasp_re <= 10000 THEN 6
  WHEN f.matriculasp_re > 10000 AND f.matriculasp_re <= 50000 THEN 7
  WHEN f.matriculasp_re > 50000 AND f.matriculasp_re <= 100000 THEN
8
  WHEN f.matriculasp_re > 100000 AND f.matriculasp_re <= 200000 THEN
9
  WHEN f.matriculasp_re > 200000 THEN 10
  ELSE 0
END AS matriculas_pre,
CASE
  WHEN f.alunos_c_fund > 0 AND f.alunos_c_fund <= 10 THEN 1
  WHEN f.alunos_c_fund > 10 AND f.alunos_c_fund <= 50 THEN 2
  WHEN f.alunos_c_fund > 50 AND f.alunos_c_fund <= 100 THEN 3
  WHEN f.alunos_c_fund > 100 AND f.alunos_c_fund <= 200 THEN 4
  WHEN f.alunos_c_fund > 200 AND f.alunos_c_fund <= 500 THEN 5
  WHEN f.alunos_c_fund > 500 AND f.alunos_c_fund <= 1000 THEN 6
  WHEN f.alunos_c_fund > 1000 AND f.alunos_c_fund <= 2000 THEN 7
  WHEN f.alunos_c_fund > 2000 AND f.alunos_c_fund <= 3000 THEN 8
  WHEN f.alunos_c_fund > 3000 AND f.alunos_c_fund <= 5000 THEN 9
  WHEN f.alunos_c_fund > 5000 THEN 10
  ELSE 0
END AS alunos_c_fund,
CASE
  WHEN f.alunos_a_fund > 0 AND f.alunos_a_fund <= 10 THEN 1
  WHEN f.alunos_a_fund > 10 AND f.alunos_a_fund <= 50 THEN 2
  WHEN f.alunos_a_fund > 50 AND f.alunos_a_fund <= 100 THEN 3
  WHEN f.alunos_a_fund > 100 AND f.alunos_a_fund <= 200 THEN 4
  WHEN f.alunos_a_fund > 200 AND f.alunos_a_fund <= 500 THEN 5
  WHEN f.alunos_a_fund > 500 AND f.alunos_a_fund <= 1000 THEN 6
  WHEN f.alunos_a_fund > 1000 AND f.alunos_a_fund <= 2000 THEN 7
  WHEN f.alunos_a_fund > 2000 AND f.alunos_a_fund <= 3000 THEN 8
  WHEN f.alunos_a_fund > 3000 AND f.alunos_a_fund <= 5000 THEN 9
  WHEN f.alunos_a_fund > 5000 THEN 10
  ELSE 0
END AS alunos_a_fund,
CASE
```

```

        WHEN f.alunos_r_fund > 0 AND f.alunos_r_fund <= 10 THEN 1
        WHEN f.alunos_r_fund > 10 AND f.alunos_r_fund <= 50 THEN 2
        WHEN f.alunos_r_fund > 50 AND f.alunos_r_fund <= 100 THEN 3
        WHEN f.alunos_r_fund > 100 AND f.alunos_r_fund <= 200 THEN 4
        WHEN f.alunos_r_fund > 200 AND f.alunos_r_fund <= 500 THEN 5
        WHEN f.alunos_r_fund > 500 AND f.alunos_r_fund <= 1000 THEN 6
        WHEN f.alunos_r_fund > 1000 AND f.alunos_r_fund <= 2000 THEN 7
        WHEN f.alunos_r_fund > 2000 AND f.alunos_r_fund <= 3000 THEN 8
        WHEN f.alunos_r_fund > 3000 AND f.alunos_r_fund <= 5000 THEN 9
        WHEN f.alunos_r_fund > 5000 THEN 10
        ELSE 0
    END AS alunos_r_fund,
    CASE
        WHEN f.alunos_c_medio > 0 AND f.alunos_c_medio <= 10 THEN 1
        WHEN f.alunos_c_medio > 10 AND f.alunos_c_medio <= 50 THEN 2
        WHEN f.alunos_c_medio > 50 AND f.alunos_c_medio <= 100 THEN 3
        WHEN f.alunos_c_medio > 100 AND f.alunos_c_medio <= 200 THEN 4
        WHEN f.alunos_c_medio > 200 AND f.alunos_c_medio <= 500 THEN 5
        WHEN f.alunos_c_medio > 500 AND f.alunos_c_medio <= 1000 THEN 6
        WHEN f.alunos_c_medio > 1000 AND f.alunos_c_medio <= 2000 THEN 7
        WHEN f.alunos_c_medio > 2000 AND f.alunos_c_medio <= 3000 THEN 8
        WHEN f.alunos_c_medio > 3000 AND f.alunos_c_medio <= 5000 THEN 9
        WHEN f.alunos_c_medio > 5000 THEN 10
        ELSE 0
    END AS alunos_c_medio,
    CASE
        WHEN f.alunos_a_medio > 0 AND f.alunos_a_medio <= 10 THEN 1
        WHEN f.alunos_a_medio > 10 AND f.alunos_a_medio <= 50 THEN 2
        WHEN f.alunos_a_medio > 50 AND f.alunos_a_medio <= 100 THEN 3
        WHEN f.alunos_a_medio > 100 AND f.alunos_a_medio <= 200 THEN 4
        WHEN f.alunos_a_medio > 200 AND f.alunos_a_medio <= 500 THEN 5
        WHEN f.alunos_a_medio > 500 AND f.alunos_a_medio <= 1000 THEN 6
        WHEN f.alunos_a_medio > 1000 AND f.alunos_a_medio <= 2000 THEN 7
        WHEN f.alunos_a_medio > 2000 AND f.alunos_a_medio <= 3000 THEN 8
        WHEN f.alunos_a_medio > 3000 AND f.alunos_a_medio <= 5000 THEN 9
        WHEN f.alunos_a_medio > 5000 THEN 10
        ELSE 0
    END AS alunos_a_medio,
    CASE
        WHEN f.alunos_r_medio > 0 AND f.alunos_r_medio <= 10 THEN 1
        WHEN f.alunos_r_medio > 10 AND f.alunos_r_medio <= 50 THEN 2
        WHEN f.alunos_r_medio > 50 AND f.alunos_r_medio <= 100 THEN 3
        WHEN f.alunos_r_medio > 100 AND f.alunos_r_medio <= 200 THEN 4
        WHEN f.alunos_r_medio > 200 AND f.alunos_r_medio <= 500 THEN 5
        WHEN f.alunos_r_medio > 500 AND f.alunos_r_medio <= 1000 THEN 6
        WHEN f.alunos_r_medio > 1000 AND f.alunos_r_medio <= 2000 THEN 7
        WHEN f.alunos_r_medio > 2000 AND f.alunos_r_medio <= 3000 THEN 8
        WHEN f.alunos_r_medio > 3000 AND f.alunos_r_medio <= 5000 THEN 9
        WHEN f.alunos_r_medio > 5000 THEN 10
        ELSE 0
    END AS alunos_r_medio
FROM dw.fato_ensino_escola f
JOIN dw.pre_escolar pe ON f.pre_escolar = pe.id_pre_escolar
JOIN dw.ens_fundamental ef ON ef.id_ens_fundamental = f.ensino_fundamental
JOIN dw.ens_medio em ON em.id_ens_medio = f.ensino_medio
JOIN dw.geografia g ON g.id_geografia = f.geografia
JOIN dw.escola e ON e.id_escola = f.escola
JOIN dw.tempo t ON t.id_tempo = f.tempo;

```