

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM WEB E SISTEMAS DE INFORMAÇÃO

SANDRA LUZIA CORTINOVI

**Uso de XML no intercâmbio de dados
armazenados em bancos de dados relacionais**

Trabalho de conclusão de curso
apresentado como requisito parcial
para obtenção do grau de Especialista

Prof. Dr. Carlos Alberto Heuser
Orientador

Prof. Dr. Carlos Alberto Heuser
Coordenador do Curso

Porto Alegre, novembro de 2004

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Cortinovi, Sandra Luzia

Uso do XML no intercâmbio de dados armazenados em bancos de dados relacionais / Sandra Luzia Cortinovi. - Porto Alegre: Curso de Especialização em WEB e Sistemas de Informação, 2004.

51 f.:il.

Trabalho de Conclusão de Curso (especialização) - Universidade Federal do Rio Grande do Sul. Curso de Especialização em WEB e Sistemas de Informação, Porto Alegre, BR-RS, 2004. Orientador: Carlos Alberto Heuser.

1. XML. 2. Banco de Dados Relacionais. 3. Intercâmbio. 4. Estudo de Caso. I. Heuser, Carlos Alberto. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Prof. Valquiria Linck Bassani

Diretor do Instituto de Informática: Prof. Dr. Phillippe Olivier Alexandre Navaux

Chefe do Departamento de Informática Aplicada: Prof. Dr. José Valdeni de Lima

Coordenador do Curso de Especialização em WEB e Sistemas de Informação: Prof. Dr. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Inicialmente, agradeço à Diretora do CPD da UFRGS, Eng^a Jussara Issa Musse, por tornar possível minha participação no Curso de Especialização em WEB e Sistemas de Informação.

Agradeço à Vice-ProReitora de Pesquisa, Marininha Aranha da Rocha, por viabilizar junto ao CNPq a implementação dos resultados deste trabalho.

Agradeço ao analista de sistemas de informação do CNPq Cristiano Lima Kuppens, cujo apoio foi fundamental para o desenvolvimento do estudo de caso.

Agradeço especialmente a Gustavo de Abreu Sisson, por viabilizar a programação e implementação do software resultante do estudo de caso.

Agradeço aos meus colegas de curso pelo companheirismo e urbanidade que nortearam os nossos estudos, e em especial à colega de grupo de trabalho Mara Denise Coutinho da Silva.

Agradeço ao corpo docente e técnico-administrativo do Instituto de Informática pelo profissionalismo e pela competência empreendidos durante o curso.

Agradeço ao meu orientador, Prof. Dr. Carlos Alberto Heuser, pela excelente orientação conferida a este trabalho.

Por fim, agradeço ao meu esposo e à minha filha pela paciência e compreensão que tiveram pela minha dedicação ao curso e a esta monografia.

SUMÁRIO

LISTA DE FIGURAS	5
LISTA DE TABELAS	6
LISTA DE ABREVIATURAS E SIGLAS.....	7
RESUMO.....	8
ABSTRACT.....	9
1 INTRODUÇÃO	10
2 INTEGRAÇÃO XML E BANCOS DE DADOS RELACIONAIS	12
2.1 XML E BANCO DE DADOS RELACIONAIS	12
2.2 ARQUITETURA PARA TRANSFORMAÇÃO XML-BDR.....	14
2.2.1 Mapeamento baseado em tabela	15
2.2.2 Mapeamento baseado em objeto.....	16
2.3 FERRAMENTAS PARA CONSTRUÇÃO DA INTERFACE XML-BDR	19
2.3.1 XML Data Binding	19
2.3.2 Middleware	21
2.3.3 Bancos de Dados com suporte a XML	21
3 ESTUDO DE CASO – EXPORTAÇÃO PIBIC UFRGS-CNPQ NO FORMATO XML.....	23
3.1 ESTRUTURA DA INFORMAÇÃO PARA INTERCÂMBIO PIBIC	24
3.2 MODELO DE BOLSAS DE IC NA UFRGS.....	30
3.3 PROCESSO DE GERAÇÃO XML PIBIC UFRGS 2004	31
3.4 EXPORTAÇÃO XML PIBIC UFRGS	35
4 CONCLUSÃO	36
REFERÊNCIAS.....	37
ANEXO XML SCHEMA PIBIC.....	40

LISTA DE FIGURAS

Figura 2.1: Transformação entre XML e BDR	14
Figura 2.2: Exemplo de mapeamento baseado em tabela	15
Figura 2.3: Modelos de documento XML para mapeamento baseado em tabela	16
Figura 2.4: Mapeamento simples entre documento XML, objeto e tabela	17
Figura 2.5: Exemplo de transformação XML para objetos para tabelas	18
Figura 2.6: Categorias de softwares XML-BDR.....	20
Figura 3.1: XML Schema global do PIBIC CNPq.....	24
Figura 3.2: Tipo complexo DadosBasicosPessoaFisica.....	25
Figura 3.3: Tipo complexo Endereco.....	25
Figura 3.5: Digrama esquemático da representação de informações do PIBIC CNPq	27
Figura 3.6: Organização dos dados de qualquer projeto	28
Figura 3.7: Organização dos dados de qualquer bolsa	28
Figura 3.8: Organização dos dados bancários	29
Figura 3.9: Organização dos dados da bolsa	29
Figura 3.10: Modelo ER das bolsas de IC.....	30
Figura 3.11: Processo de geração do XML PIBIC UFRGS	32
Figura 3.12: Modelo de objetos PIBIC gerado pelo <i>wizard</i> XML Data Binding.....	33
Figura 3.13: Interface do software gerador XML PIBIC UFRGS	34

LISTA DE TABELAS

Tabela 2.1: Diferenças entre dados XML e BDR	13
Tabela 2.2: Exemplos de suporte a XML em BD comerciais	22
Tabela 3.1: Resultado da exportação PIBIC CNPq UFRGS	35

LISTA DE ABREVIATURAS E SIGLAS

API	Application Interface Program
BDR	Banco de Dados Relacional
CPD	Centro de Processamento de Dados da UFRGS
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DTD	Document Type Definition
DOM	Document Object Model
ER	Entidade e Relacionamento
HTML	HyperText Markup Language
IC	Iniciação Científica
LPML	Linguagem de Marcação da Plataforma Lattes
PROPESQ	Pró-Reitoria de Pesquisa
PIBIC	Programa Institucional de Bolsas de Iniciação Científica
SAX	Simple API for XML
SGBDR	Sistema Gerenciador de Banco de Dados Relacional
SQL	Structured Query Language
USP	Universidade de São Paulo
UFRJ	Universidade Federal do Rio de Janeiro
UFRGS	Universidade Federal do Rio Grande do Sul
USP	Universidade de São Paulo
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSD	W3C XML Schema

RESUMO

Com a expansão do uso da Internet e a necessidade de se manter todos os processos de negócio integrados, cresce a demanda de tecnologias que possibilitem o desenvolvimento de sistemas com capacidade de integração e interoperabilidade. É desejável que a integração desses sistemas preserve a estrutura física de suas bases de dados e disponibilize informações dispensando ajuste de modelos e sistemas. Portanto, é fundamental a padronização do método de comunicação. A linguagem XML tem sido aceita como um formato universal para o intercâmbio de dados na Internet.

O propósito deste trabalho é investigar e apresentar estratégias e ferramentas para o uso da linguagem XML no intercâmbio de dados armazenados em bancos de dados relacionais.

Para atingir este propósito, é realizado um estudo sobre um caso de dimensão e complexidade realistas, a geração no formato XML do resultado do processo de seleção de bolsistas do programa PIBIC CNPq-UFRGS 2004.

Palavras-chave: XML, Banco de Dados Relacionais, Intercâmbio de Dados, PIBIC

XML Use in Data Interchange Stored in Relational Databases

ABSTRACT

With the expansion of the use of the Internet and the necessity of keeping integrated all the business processes, grow the demand of technologies that make possible the development of systems with interoperability and integration capabilities. It is desirable that the integration of these systems preserves the physical structure of its databases and provides information avoiding adjustments of models and systems. Therefore, standardization of the communication method is basic. Language XML has been accepted as a universal format for the interchange of data in the Internet.

The intention of this work is to investigate and present strategies and tools for the use of the XML language in the interchange of data stored in relational databases.

To reach this intention, it is carried through a study on a case of realistic dimension and complexity, the generation in XML format of the result of the process of election of scholarship holders of program PIBIC CNPq-UFRGS 2004.

Keywords: XML, Databases, Data Interchange, PIBIC

1 INTRODUÇÃO

Com o crescimento espantoso no uso da Internet nos últimos anos e a necessidade de se manter todos os processos de negócio integrados, cresce a demanda de tecnologias que possibilitem o desenvolvimento de sistemas com capacidade de integração e interoperabilidade.

A interoperabilidade, definida como “a interconexão efetiva de diferentes sistemas de computador, bancos de dados ou redes com o fim de apoiar computação distribuída e/ou intercâmbio de dados” (PACHECO; KERN, 2001), costuma ser impedida por barreiras de plataformas diferentes de hardware e software, incluindo sistemas operacionais, paradigmas de programação e de modelo de dados. Uma das barreiras à interoperabilidade é a incompatibilidade dos modelos de dados subjacentes às aplicações.

É desejável que a integração desses sistemas preserve a estrutura física de suas bases de dados e disponibilize informações segundo lógica baseada em conteúdo compartilhado, dispensando ajuste de modelos e sistemas. Portanto, a padronização do método de comunicação torna-se essencial para que o processo de intercâmbio de informações seja executado de forma ágil e simples.

A Linguagem de Marcação Extensível, abreviada como XML, tem sido aceita como um formato universal para intercâmbio e publicação de dados na Internet. Pode ser aplicada a aplicações onde os dados necessitam ser representados no formato XML, tanto aplicações onde os dados iniciais estão definidos no formato XML e devem ser armazenados em um banco de dados, como em aplicações onde os dados do banco de dados devem ser encapsulados no formato XML com o intuito de intercâmbio ou publicação. A natureza auto-descritiva do XML simplifica a troca de dados transacionais entre parceiros de negócios e sistemas incompatíveis.

O propósito deste trabalho é investigar e apresentar estratégias e ferramentas para o uso de XML no intercâmbio de informações entre sistemas de diferentes plataformas. Especificamente, estratégias para geração de documentos XML a partir bancos de dados relacionais, bem como importação de dados presentes em documentos XML para bancos de dados relacionais.

Para atingir este propósito, utilizou-se um caso de dimensão e complexidade realistas, a geração no formato XML do resultado do processo de seleção de bolsistas do programa PIBIC CNPq-UFRGS 2004 conforme gramática em avaliação pela Comunidade LPML (CNPq, 2003).

A estrutura do trabalho é descrita a seguir:

O capítulo 2 apresenta uma explanação sobre as estratégias de integração de XML com bancos de dados relacionais, com enfoque no intercâmbio de dados. São

comentadas as diferenças básicas entre XML e bancos de dados relacionais, a arquitetura básica de uma interface para importação e exportação de dados no formato XML, e as alternativas de softwares existentes para construção dessa interface, são comentadas.

O terceiro capítulo descreve a solução adotada para geração e exportação dos dados dos bolsistas PIBIC selecionados na UFRGS para o CNPq no formato XML. O processo de seleção dos bolsistas CNPq-UFRGS, a gramática XML definida pelo CNPq para representação dos dados PIBIC e os resultados da aplicação dessa solução são também apresentados.

Finalmente, no capítulo 4, são apresentadas as conclusões deste trabalho e as indicações de trabalhos futuros.

2 INTEGRAÇÃO XML E BANCOS DE DADOS RELACIONAIS

A linguagem XML vem sendo reconhecida como a melhor forma de representar todos os tipos de informações, tornando possível a utilização destas informações em qualquer tipo de aplicação (DAUM; HORAK, 2000). Ela se enquadra no cenário atual de busca de padrões abertos que tiram o máximo de vantagem da arquitetura da Internet, servindo para diversos propósitos como uma linguagem de marcação mais expressiva que HTML e um formato de intercâmbio de dados (ERL, 2004).

Os sistemas gerenciadores de bancos de dados, principalmente os bancos de dados relacionais, são atualmente a melhor e mais utilizada forma de armazenamento de dados corporativos, pois fornecem controle de acesso para um grande número de usuários, otimizando as operações e oferecendo segurança a estas informações (GUARDALBEN ; SHAKU, 2004; WILLIAMS et al, 2001).

Os bancos de dados relacionais e a linguagem XML têm cada um aplicações específicas, mas quando integrados constituem uma ótima solução para intercâmbio de dados entre sistemas em diferentes plataformas (WILLIAMS et al, 2001).

A estratégia para integração de XML com bancos de dados relacionais depende do uso das informações que serão representadas na linguagem XML. Para informações provenientes de um banco de dados relacional ou cujo destino seja um banco de dados relacional é necessário a definição de uma interface de transformação de dados no formato XML para o modelo relacional e vice-versa.

Este capítulo comenta inicialmente as diferenças básicas entre XML e bancos de dados relacionais, para em seguida apresentar a arquitetura básica de uma interface para importação e exportação de dados no formato XML. Ao final, discute as alternativas de ferramentas existentes para construção dessa interface.

2.1 XML e banco de dados relacionais

Como “XML e bancos de dados relacionais são plataformas de dados fundamentalmente incompatíveis, criadas com objetivos distintos” (ERL, 2004), é importante, antes de definir qualquer estratégia de integração de dados, entender as diferenças fundamentais entre bancos de dados relacionais e a tecnologia XML, bem como os tipos de informações que podem ser representadas em XML.

Documentos XML e bancos de relacionais apresentam abordagens de representação de dados opostos. Documentos XML representam uma estrutura hierárquica coesa, na qual existe sempre um elemento raiz que é composto por um ou mais elementos que podem, por sua vez, conter uma outra hierarquia de elementos. SGBDRs representam entidades tabulares (tabelas), compostas de linhas e colunas, com

os dados armazenados em um único lugar (normalização). A tabela 2.1 resume as diferenças essenciais entre XML e BDR.

Tabela 2.1: Diferenças entre dados XML e BDR

XML	SGBDR
Armazenamento como arquivo texto.	Armazenamento em ambiente altamente controlado.
Dados em uma única estrutura hierárquica	Dados em múltiplas tabelas
Nós contém elementos e/ou valores de atributos	Células contém um único valor
Elementos podem ser aninhados	Valores atômicos em células
Elementos são ordenados	Ordem linha/coluna indefinida
Esquema opcional	Esquema obrigatório
Consulta via padrões XML.	Consultas no padrão SQL

A forma como as informações estão estruturadas em um arquivo XML permite classificá-las em duas categorias (BOURRET, 2004a; BOURRET, 2004c): informações voltadas para documentos (“*document-centric documents*”) ou informações voltadas para dados (“*data-centric documents*”). Esta classificação é relevante devido ao fato de arquivos XML centrados em dados partilharem um conjunto de características distintas de arquivos XML centrados em documentos, influenciando a forma como as informações são armazenadas e recuperadas.

Informações orientadas a documento são definidas como de “consumo humano”: são produzidas diretamente por pessoas e serão consumidas também por pessoas, como livros e e-mail. Caracterizam-se por uma estrutura pouco regular, estando a menor estrutura independente de dados no nível de um elemento misturado ao conteúdo ou no próprio documento. As informações só tem significado no contexto do documento, o qual depende não apenas da representação dos dados expressa através de marcações, mas também da ordenação dos componentes dos dados.

Informações voltadas para dados se enquadram em uma estrutura bastante regular, que oferece uma granularidade mais fina, permitindo que se lide com os dados representados em XML no nível de seus elementos. O significado das informações contidas no documento depende apenas da sua estrutura, sendo irrelevante a ordem na qual esses elementos serão armazenados.

Basicamente, arquivos XML contendo informações orientadas a documentos tendem a ser mantidos em bancos de dados nativos XML¹ ou sistemas de gerência de conteúdo². Já informações oriundas de bancos de dados relacionais ou cujo destino seja um banco de dados relacional podem quase que certamente ser classificadas como orientadas a dados. Informações que se encaixam nesta categoria simplesmente fazem

¹ Bancos de dados nativos XML são bancos de dados projetados para armazenar documentos XML

² Sistemas de gerência de conteúdo são aplicações projetadas para gerência de documentos e construídas com base em XML nativo

uso das características de portabilidade da linguagem XML para o transporte de dados entre aplicações (BOURRET, 2004a; ERL, 2004).

2.2 Arquitetura para transformação XML-BDR

Para que o XML seja usado como meio intercâmbio de dados entre sistemas, especialmente entre sistemas de diferentes organizações, os seguintes pré-requisitos são essenciais (WILLIAMS et al, 2001).

- Identificação do escopo do documento, isto é, dos conceitos fundamentais (dados e relações) que serão representados no formato XML.
- Especificação não ambígua desses conceitos, definindo uma classe de documentos através de um esquema XML³ (DTD ou XSD).

A interface de exportação/importação consiste então no processo de transformação de dados entre o documento XML e o banco de dados baseado no mapeamento dos esquemas de ambas as partes.

A figura 2.1 apresenta a arquitetura do módulo de transformação de dados entre XML e SGBDR. Na exportação de dados de BDR para o formato XML, o processo utiliza o mapeamento para recuperar os dados do BD e produzir o documento XML. Na importação de dados, o processo lê o documento XML e insere ou atualiza as tabelas do banco de dados de acordo com o mapeamento.

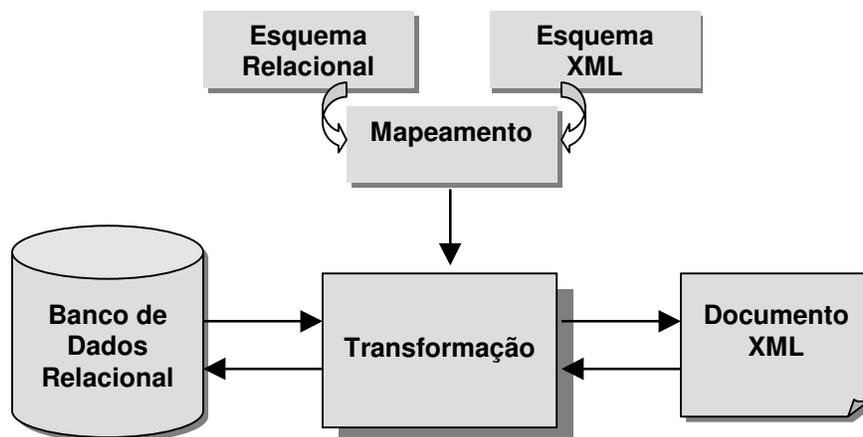


Figura 2.1: Transformação entre XML e BDR

Para mapear documentos XML para bancos de dados relacionais há métodos manuais e dois métodos automáticos principais: o mapeamento baseado em tabela e o mapeamento baseado em objeto (BOURRET, 2004a ; ERL, 2004).

³ Um esquema XML descreve uma classe de documento XML, definindo todo o vocabulário permitido para elementos e atributos, além de determinar os tipos de dados que podem ser usados, a ordem e organização hierárquica dos elementos.

Os métodos manuais consistem em se especificar em um arquivo de conversão quais elementos e atributos vão para seus respectivos depósitos. Tais métodos não são padronizados e dependem da implementação do banco de dados.

Os métodos automáticos, tanto o mapeamento baseado em tabela quanto o mapeamento baseado em objeto, realizam o mapeamento sobre tipos de elementos, atributos e textos, descartando grande parte da informação a respeito do documento XML, como a sua estrutura física, definição de entidades e a ordem em que os valores de atributos ocorrem no documento. Isto é perfeitamente aceitável, já que para a aplicação e o BD que interessa são os dados do documento XML.

2.2.1 Mapeamento baseado em tabela

O mapeamento baseado em tabela modela documentos XML como uma única tabela ou um conjunto de tabelas. A figura 2.2 exibe um mapeamento óbvio entre documento XML e tabela.

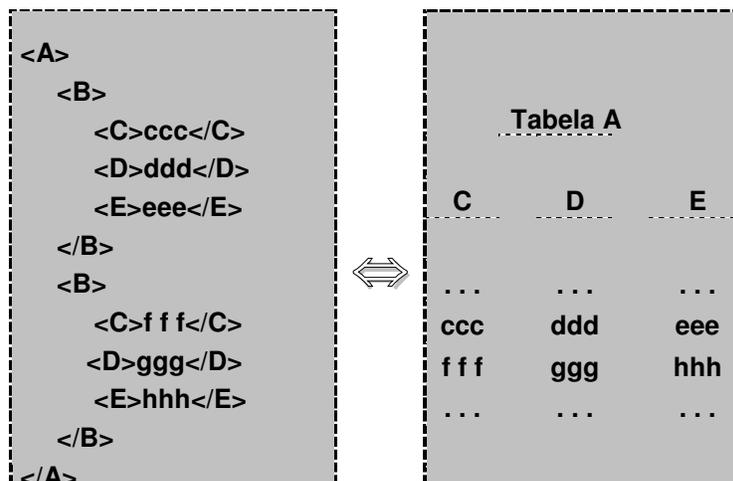


Figura 2.2: Exemplo de mapeamento baseado em tabela

Note que o termo “tabela” é interpretado livremente: na transferência de dados de BDR para XML, uma “tabela” pode ser qualquer conjunto de linhas resultantes de uma consulta SQL; na transferência de dados de XML para BDR, uma “tabela” pode ser uma visão atualizável

Para que este método possa ser utilizado a estrutura do documento XML deve obedecer a um dos modelos apresentado na figura 2.3.

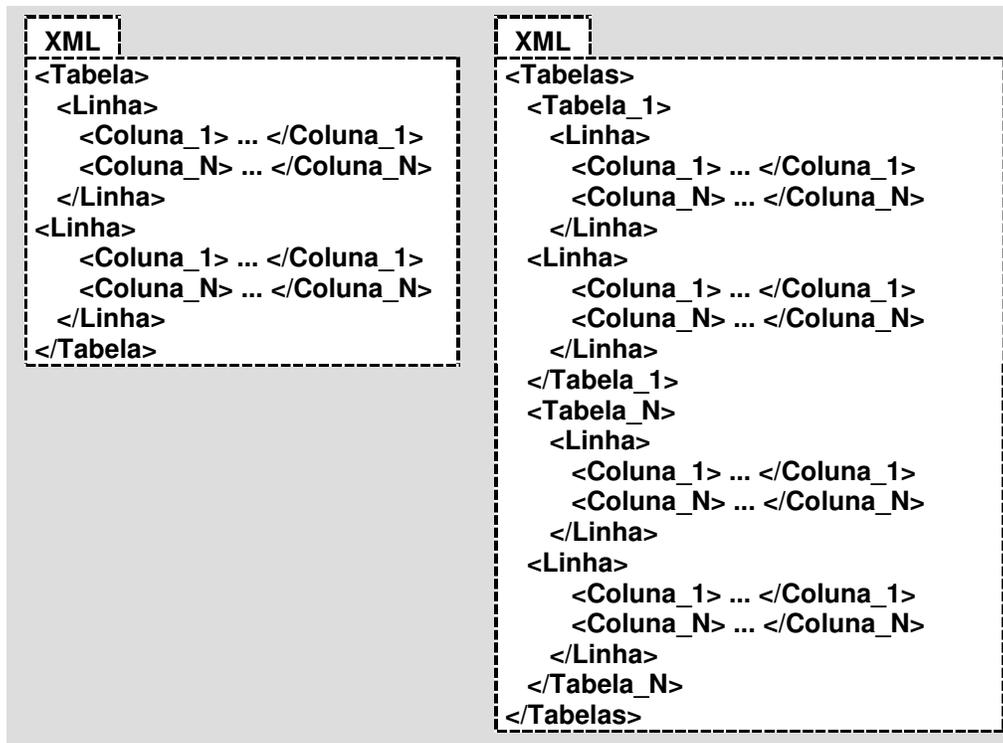


Figura 2.3: Modelos de documento XML para mapeamento baseado em tabela

O mapeamento baseado em tabela tem a vantagem da simplicidade, o que o torna útil como ferramenta de transferência de dados entre documentos XML e bancos de dados relacionais, bem como em aplicações Web que exibem resultados de consultas SQL no formato XML. Entretanto, este método aplica-se apenas ao conjunto limitado de documentos XML que seguem os padrões acima, não explorando a habilidade do XML para representar dados hierárquicos.

2.2.2 Mapeamento baseado em objeto

O mapeamento baseado em objeto, também denominado mapeamento baseado em classes ou mapeamento objeto-relacional, modela o documento XML como uma árvore de objetos e mapeia estes objetos para o banco de dados.

Cabe destacar que os objetos envolvidos são específicos de cada esquema XML e não objetos do DOM (BOURRET, 2001b). Esses objetos representam dados de uma classe de documentos XML enquanto os objetos do DOM representam as estruturas dos documentos XML.

A figura 2.4 exibe um mapeamento simples entre um documento XML, um objeto e linhas em uma tabela, com o correspondente mapeamento entre o esquema XML representado como DTD, a classe e o esquema da tabela. A figura 2.5 exemplifica um mapeamento um pouco mais complexo no qual um documento XML é convertido em objetos que por sua vez são transformados em tabelas.

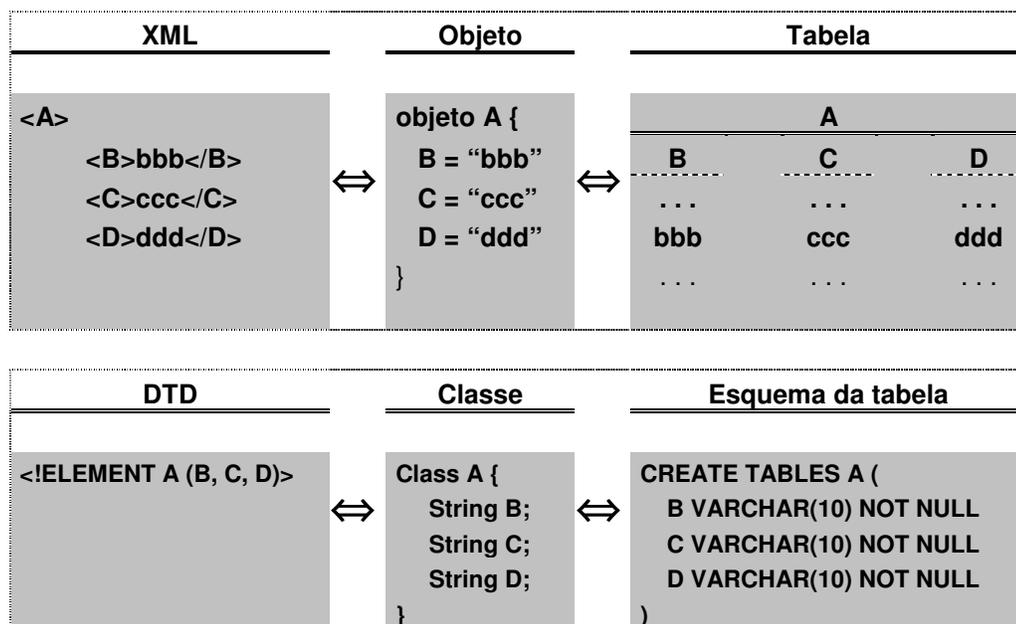


Figura 2.4: Mapeamento simples entre documento XML, objeto e tabela

O mapeamento baseado em objeto é executado em duas etapas (BOURRET, 2001b):

- O esquema XML (DTD ou XSD) é mapeado para um esquema de objetos. Esta etapa é conhecida como *XML data binding*, e consiste em relacionar tipos de elementos com tipos de dados e atributos com propriedades. Tipos simples⁴ são mapeados para tipos básicos. Tipos complexos⁵ são mapeados para classes, onde cada elemento contido no tipo complexo é mapeado para uma propriedade da classe. Referências a tipos de elementos complexos são mapeadas como ponteiros para um objeto da classe com o qual o tipo do elemento complexo foi mapeado.
- O esquema de objetos é então mapeado para um esquema relacional. Basicamente, classes são mapeadas para tabelas, propriedades básicas para colunas, e ponteiros para relacionamentos, isto é, pares chave primária – chave estrangeira.

Os nomes utilizados no esquema XML, esquema de objetos e esquema relacional não precisam ser idênticos desde que, durante o mapeamento, sejam especificadas as relações entre os nomes (BOURRET, 2004 a). Além disso, os objetos envolvidos no mapeamento são conceituais, sendo instanciados apenas se a aplicação assim o especificar (BOURRET, 2001b).

⁴ Tipo simples define elementos com conteúdo do tipo PCDATA (XSD, 2003).

⁵ Tipo complexo define elementos que contém outros elementos ou atributos (XSD, 2003).

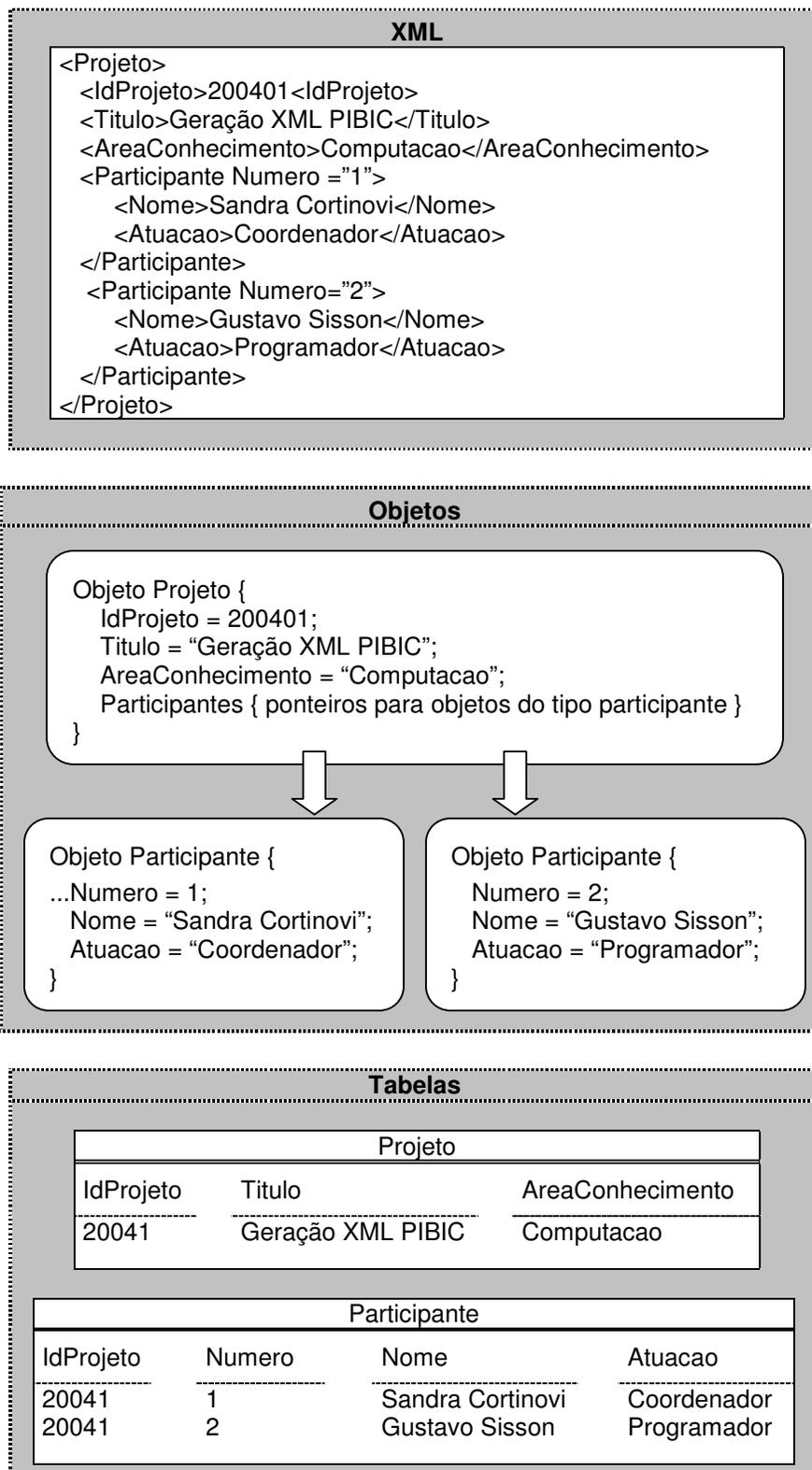


Figura 2.5: Exemplo de transformação XML para objetos para tabelas

2.3 Ferramentas para construção da interface XML-BDR

Uma interface de importação e/ou exportação de dados de uma classe de documentos XML para um BDR específico pode ser implementada por um programa escrito em qualquer linguagem de programação. Basta que tal programa implemente uma das técnicas de mapeamento discutidas na seção anterior e implemente, ou utilize, para leitura e manipulação de documentos XML um mecanismo baseado na especificação DOM ou SAX (BOURRET, 2001a; GICQUEAU, 2004).

Entretanto, tal abordagem só é viável para classes de documentos XML com estruturas simples e estáveis, isto é, que exigem apenas transformações elementares e não se alterem no decorrer do tempo. Como as classes de documentos utilizadas no intercâmbio de informações entre sistemas tendem a representar hierarquias complexas e em constante evolução, o uso de ferramentas de software para gerar o mapeamento de dados XML reduz significativamente o tempo de desenvolvimento e manutenção de aplicações de importação/exportação XML-BD (GICQUEAU, 2004).

Atualmente encontram-se disponíveis diversos softwares, tanto comerciais como *open source*, que auxiliam na construção de aplicações de transformação de dados entre documentos XML e BDR. Embora os limites de muitos desses softwares não sejam claramente definidos, é possível classificá-los como: ferramentas para *XML Data Binding*, ou *middleware* de conversão ou bancos de dados com suporte a XML. A figura 2.6 sumariza o que caracteriza cada uma dessas categorias de software.

2.3.1 XML Data Binding

O termo *XML data binding* refere-se ao mapeamento estrutural de componentes XML em um modelo de dados de programação que preserve a hierarquia lógica dos componentes, exponha o significado real dos dados e represente os componentes no formato nativo do ambiente de programação. Consiste numa solução para o acesso aos dados de um documento XML que dispensa o uso de APIs para manipulação de baixo nível, como DOM e SAX (BIRBECK, 2001).

A maioria dos softwares que realizam *XML data binding* disponibilizam o mapeamento de um esquema XML para um esquema de objetos, gerando um componente de ligação de documentos XML a objetos projetados especialmente para os dados contidos na classe de documentos (BOURRET, 2004b). Basicamente, analisam um esquema XML e geram as classes e métodos necessários para obtenção e manipulação dos dados: métodos para geração de arquivo XML a partir dos dados armazenados em um objeto e métodos para instanciação de objetos a partir da leitura de um arquivo XML.

Os produtos de software para *XML data binding* apresentam algumas limitações como, por exemplo, suporte incompleto a esquemas XML e incapacidade de extrair dados de um ou mais fragmentos de um documento XML, exporem tais dados usando esquema específico de objetos e re-escreverem aqueles fragmentos no documento, deixando o resto do documento intacto (BOURRET, 2004b).

Alguns softwares geram código exclusivamente para um ambiente de programação específico enquanto outros disponibilizam geração de código para várias linguagens de programação.

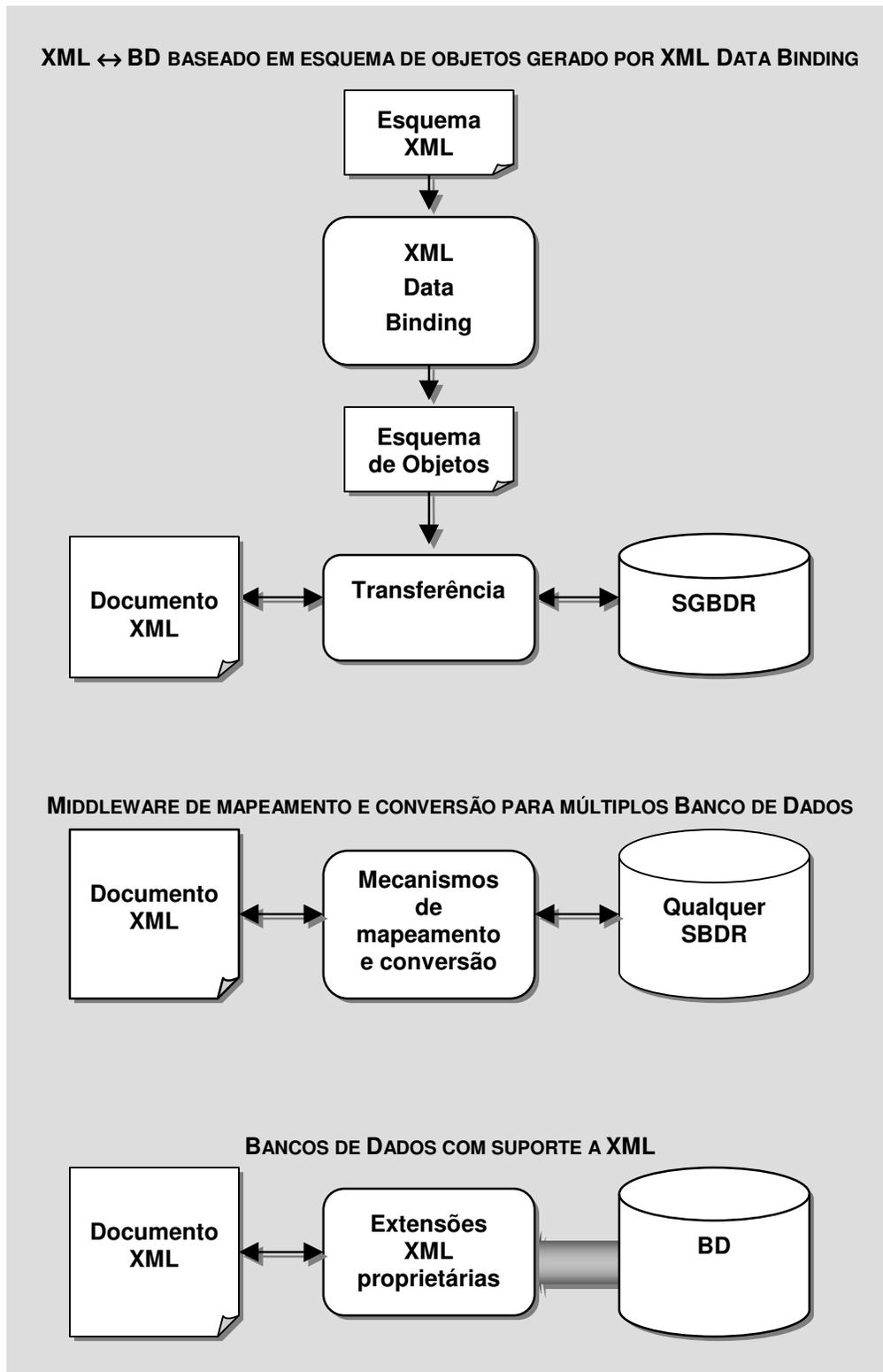


Figura 2.6: Categorias de softwares XML-BDR

2.3.2 Middleware

Produtos de *middleware* para bancos de dados são servidores independentes que atuam como camada intermediária entre sistemas de banco de dados relacionais e aplicações que necessitam utilizar dados XML.

Eles trabalham como conversores, extraindo dados de um banco de dados relacional, transformando-os para o formato XML, e passando então os dados para a aplicação XML que manipulará esses dados. No sentido inverso, ao receber de uma aplicação os dados XML a serem armazenados no banco de dados, converte-os para as instruções SQL de atualização do banco de dados.

Middlewares variam amplamente em termos de quais funcionalidades são oferecidas bem como a forma como são oferecidas. Alguns exigem que o usuário defina um arquivo de transformação que contenha o mapeamento SQL para XML, enquanto outros adotam um mapeamento padrão de resultados SQL para dados XML que não pode ser alterado. Poucos *middlewares* disponibilizam um nível de integração tal que resultados de consultas em documentos XML ou banco de dados relacionais possam ser acessados de forma idêntica pela mesma API (GUARDALBEN ; SHAKU, 2004; OBASANJO, 2004).

Componentes *middleware* também variam quanto ao grau de sofisticação da interface com o usuário, indo desde praticamente inexistente, com interação feita por programa através de APIs, até interação através de sofisticadas interfaces gráficas (OBASANJO, 2004).

A opção por servidores *middlewares* tem como vantagem esconder da aplicação os detalhes do banco de dados. Entretanto, qualquer alteração na modelagem do documento XML implica em reconstruir o mapeamento XML-BDR.

2.3.3 Bancos de Dados com suporte a XML

Bancos de dados com suporte a XML são bancos de dados relacionais tradicionais ou bancos de dados objeto-relacional que incorporam extensões para tornar possível o processamento e armazenamento de dados XML (BOURRET, 2004c).

A maioria dos grandes fabricantes de bancos de dados oferece alguma funcionalidade XML em seus produtos (DAYEN, 2004; ERL, 2004; GUARDALBEN; SHAKU, 2004). Entretanto, cada um disponibiliza o acesso a dados XML de uma forma diferente e proprietária, sendo os mecanismos mais comuns os seguintes:

- Acréscimo de comandos específicos para XML em suas implementações de SQL.
- Versões próprias de especificações XML.
- Interface ou API para mapeamento de dados XML para dados relacionais.
- *Stored procedures* para manipulação de dados XML .

A tabela 2.2 resume as características de quatro bancos de dados comerciais em relação às suas estratégias para mapeamento e os meios para importação e exportação de dados em documentos XML.

Tabela 2.2: Características de suporte a XML em BD comerciais

Fabricante	Produto	Método de Mapeamento	BD → XML	XML → BD
Microsoft	SQL Server 2000	Uso de extensão SQL; função de obtenção de visão relacional de dados XML	Comando SELECT com cláusula FOR XML	Função OPENXML como origem de dados em comandos INSERT, UPDATE ou SELECT INTO
IBM	DB2 XML Extender	Uso de arquivo XML de Definição de Acesso ao Dados (DAD)	Stored procedure dxxGenXML	Stored procedure dxxShred
Oracle	XML SQL Utility (XSU)	Construção implícita do modelo de dados objeto-relacional	Classe Java OracleXMLQuery	Classe Java OracleXMLSave
Sybase	Sybase Adaptative Server (SAS)	Uso de arquivo XML Result Set contendo a estrutura e conteúdo do documento XML.	Método getXmiText : extrai documento XML de objeto da classe ResultSetXML gerado a partir de consulta SQL	Método toSqlScript : gera comandos SQL com base em objeto da classe ResultSetXML gerado a partir de documento XML

3 ESTUDO DE CASO – EXPORTAÇÃO PIBIC UFRGS-CNPq NO FORMATO XML

O Programa Institucional de Iniciação Científica do CNPq (PIBIC), juntamente com o Programa Interno de Iniciação Científica (BIC) e o Programa Institucional de Iniciação Científica da FAPERGS (PROBIC), disponibilizam quotas de bolsas de iniciação científica⁶ aos pesquisadores da UFRGS. Caracterizam-se pelo seu enfoque institucional, pois cabe a universidade a responsabilidade de administrar e operacionalizar a quota de bolsas.

Atualmente, os processos de seleção, acompanhamento e avaliação desses programas são gerenciados e implementados pela Pró Reitoria de Pesquisa (PROPESQ) com o apoio formulários Web e aplicações cliente-servidor que atualizam o banco de dados institucional da UFRGS (UFRGS, 2003).

No PIBIC, uma quota de bolsas é destinada a um pesquisador que orientará as atividades de um até três bolsistas em um projeto de pesquisa. As bolsas tem duração de 12 meses, sendo passíveis de renovação. Durante este período cabe à universidade acompanhar e avaliar o trabalho dos bolsistas, ficando sob responsabilidade do CNPq o pagamento das bolsas. É responsabilidade da instituição informar ao CNPq as eventuais substituições de projeto de pesquisa e bolsistas durante o período, bem como os resultados das avaliações.

O processo de seleção PIBIC inicia com a publicação do Edital e envolve as seguintes etapas, todas elas consultando e/ou atualizando o banco de dados da UFRGS:

- Inscrição, realizada pelo pesquisador-orientador através de formulário de solicitação de quota disponibilizado na Web.
- Distribuição das solicitações para os consultores, realizada pela PROPESQ via aplicação cliente-servidor.
- Análise das solicitações, com emissão de pareceres, realizada pelos consultores com o apoio de formulário Web.
- Priorização e implementação das solicitações aprovadas, realizada pela PROPESQ via aplicação cliente-servidor.
- Divulgação dos resultados na Web.
- Indicação dos bolsistas, realizada pelos pesquisadores-orientadores com quotas aprovadas via formulário Web.

⁶ Programas de Iniciação Científica disponibilizam bolsas a alunos de graduação para que atuem em projetos de pesquisa sob orientação de professores/pesquisadores.

- Atualização de informações cadastrais e bancárias, realizada pelos bolsistas indicados via formulário Web.

Após a conclusão do processo de seleção, cabe a universidade enviar ao CNPq a relação de quotas aprovadas. Até o ano de 2003, o CNPq disponibilizava apenas duas possibilidades para realização desse envio:

- Através do preenchimento, para cada quota aprovada, dos dados do projeto de pesquisa, pesquisador-orientador e bolsista(s) em formulários em papel remetidos pelo correio.
- Através da transmissão de um arquivo pela Web gerado por um programa específico fornecido pelo CNPq e instalado em máquina local. Esse programa exige a digitação dos dados dos projetos, orientadores e bolsistas selecionados.

Em 2004, o CNPq disponibilizou a UFRGS e a USP uma definição de arquivo XML para o envio das informações desse formulário, definição esta resultante de um projeto piloto realizado pelo CNPq junto à UFRJ em 2003 no âmbito da comunidade LPML.

A especificação e implementação da interface de geração em formato XML dos dados resultantes do processo de seleção de bolsas de iniciação científica PIBIC CNPq-UFRGS 2004 é o objeto deste estudo de caso.

A seguir apresenta-se a gramática XML para intercâmbio dos dados PIBIC, comenta-se o resultado da análise do modelo de dados para bolsas de IC na UFRGS considerando as exigências dessa gramática, e descreve-se o processo de geração do arquivo XML PIBIC UFRGS, bem como os resultados alcançados.

3.1 Estrutura da informação para intercâmbio PIBIC

A estrutura de informação do Formulário de Intercâmbio para Propostas do Programa PIBIC/CNPq é definida através de um arquivo XSD, cujo conteúdo é exibido de forma *macro* na figura 3.1 e detalhado no Anexo.

XML Schema global do PIBIC CNPq		
complexType	PessoaFisica	ann:Dados Básicos de Pessoa
element	FormularioPIBIC	ann:Formato de Intercambio para Propostas do Programa PIBIC/CNPq
complexType	Endereco	ann:Endereço residencial ou profissional
complexType	DadosBasicosPessoaFisica	ann:

Figura 3.1: XML Schema global do PIBIC CNPq

Para representar a sintaxe específica a ser respeitada nos arquivos XML com informações sobre bolsas PIBIC utiliza-se sucessivos níveis de detalhamento como os apresentados nas figuras a seguir, geradas a partir do XML SPY (ALTOVA, 2004). Esta visão esquemática é uma abstração dos documentos efetivamente usados no intercâmbio e partilhamento de informações, o arquivo XSD e arquivos de conteúdo XML.

As figuras 3.2, 3.3 e 3.4 representam, respectivamente, os tipos complexos DadosBasicosPessoaFisica, Endereco e PessoaFisica, usados na definição do elemento FormularioPIBIC.

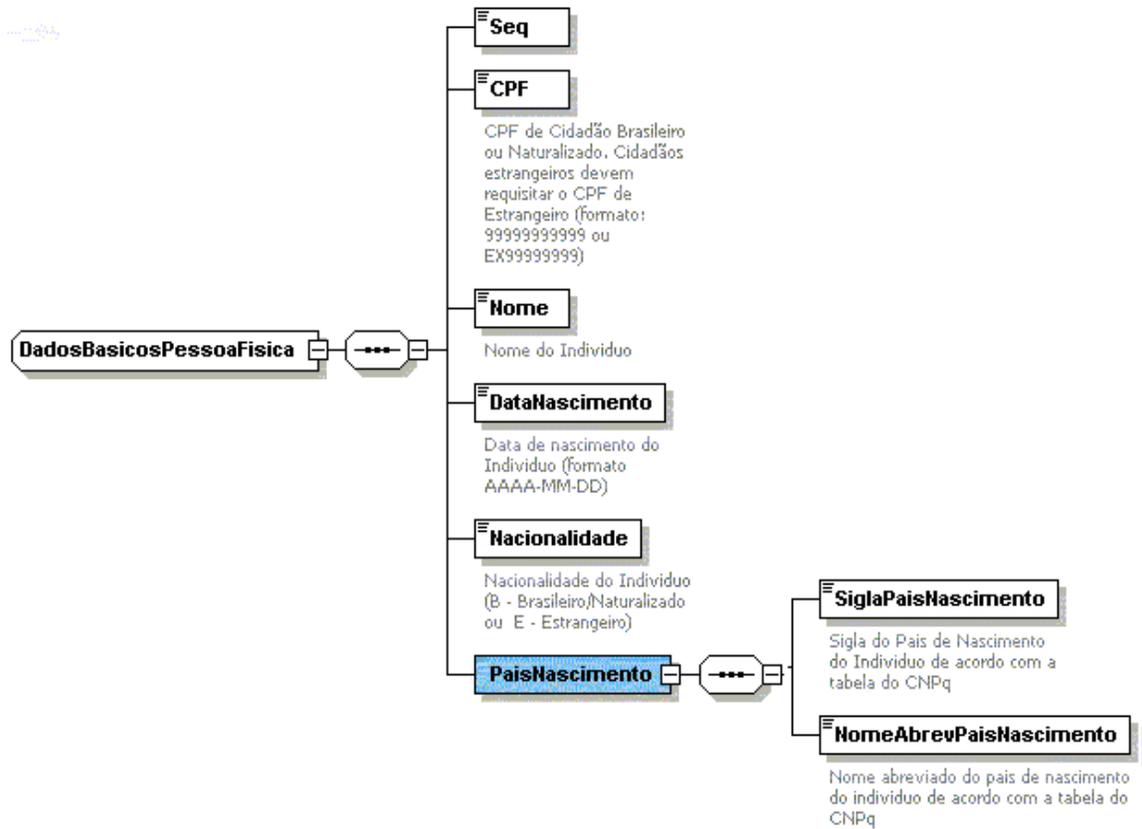


Figura 3.2: Tipo complexo DadosBasicosPessoaFisica

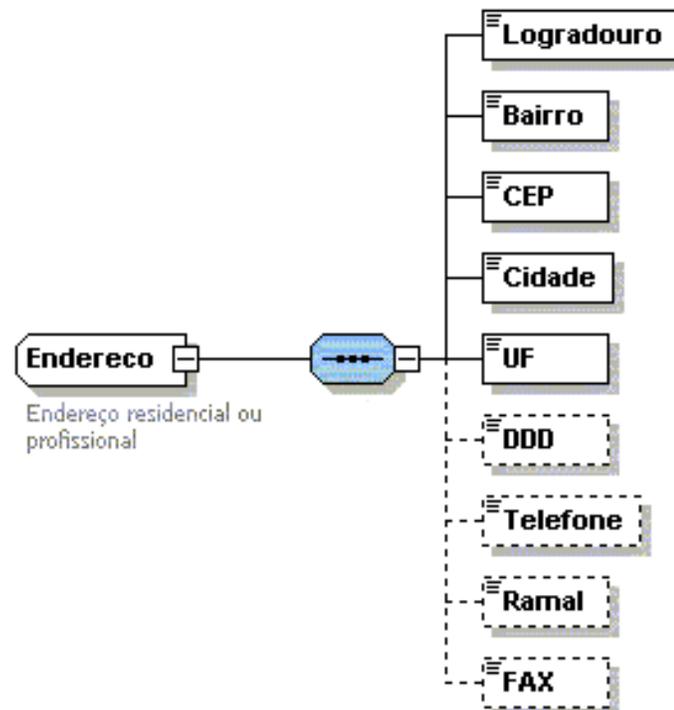


Figura 3.3: Tipo complexo Endereco

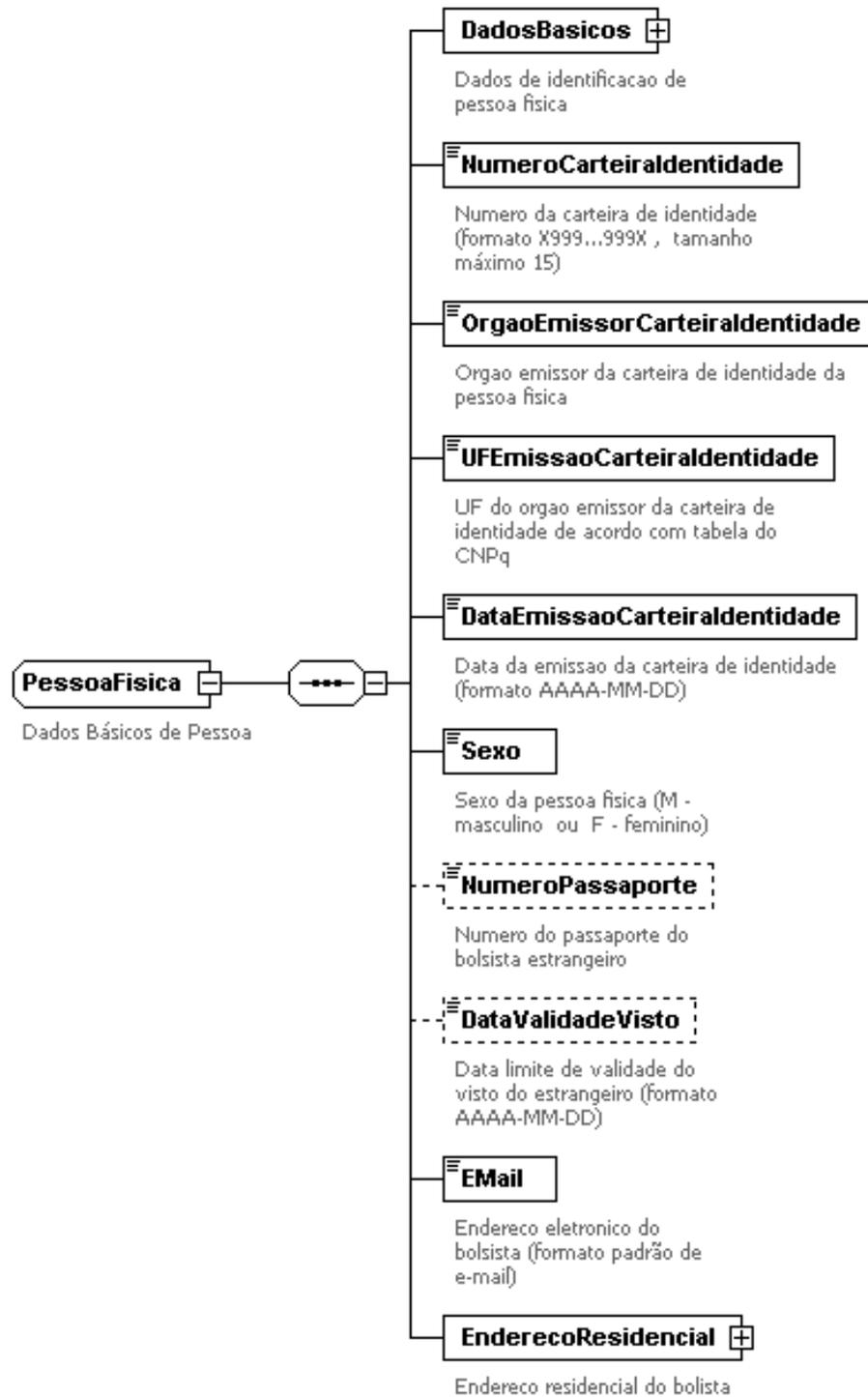


Figura 3.4:Tipo complexo PessoaFisica

A figura 3.5 ilustra, de forma esquemática, a estrutura de informação do Formulário de Intercâmbio para Propostas do Programa PIBIC/CNPq. Segundo o esquema, as informações necessárias são número do processo da instituição no programa, identificação da instituição, identificação do responsável pelo programa na instituição, e do conjunto de projetos de pesquisa com cotas aprovadas.

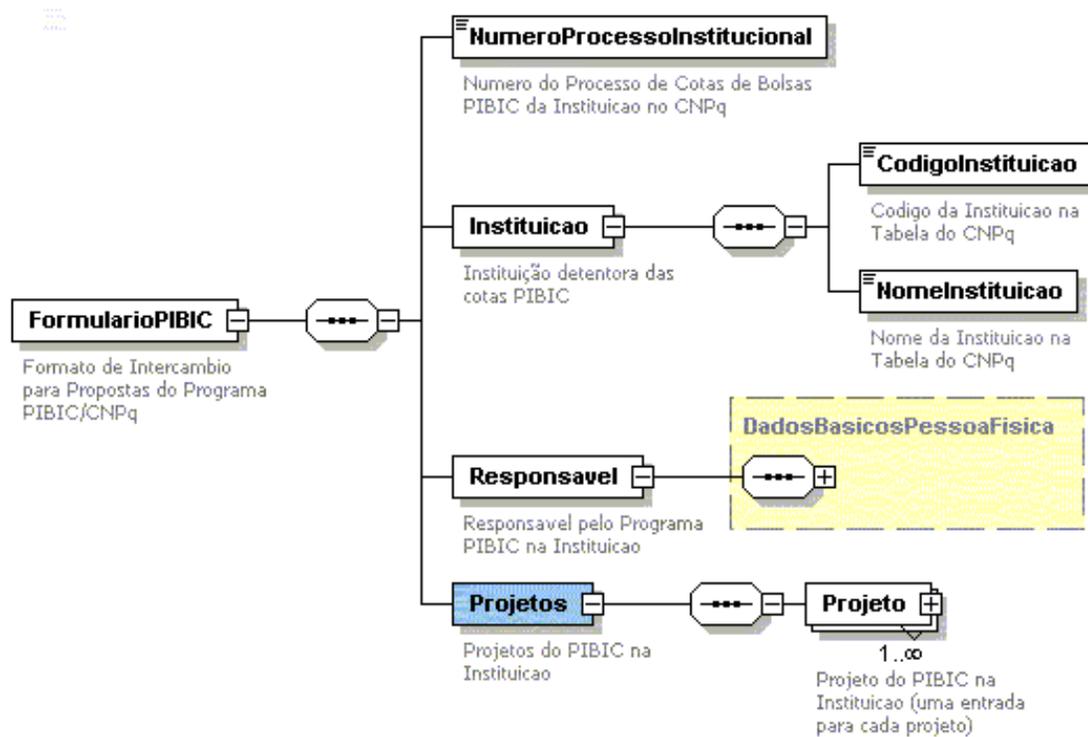


Figura 3.5: Digrama esquemático da representação de informações do PIBIC CNPq

O elemento Projeto, cuja estrutura de informação é exibida na figura 3.6, representa a quota de bolsa destinada a um projeto de pesquisa sob a responsabilidade de um pesquisador-orientador. É raiz de uma hierarquia que informa:

- a identificação do pesquisador-orientador (elemento Orientador);
- a identificação do projeto de pesquisa: título (elemento TituloProjeto), área do conhecimento (elemento AreaConhecimento) e um conjunto com no mínimo 1 e no máximo 6 palavras-chave (elemento PalavrasChaveProjeto); e
- o conjunto de bolsistas (elemento Bolsas) que receberão bolsa para atuar no projeto de pesquisa sob a orientação do pesquisador.

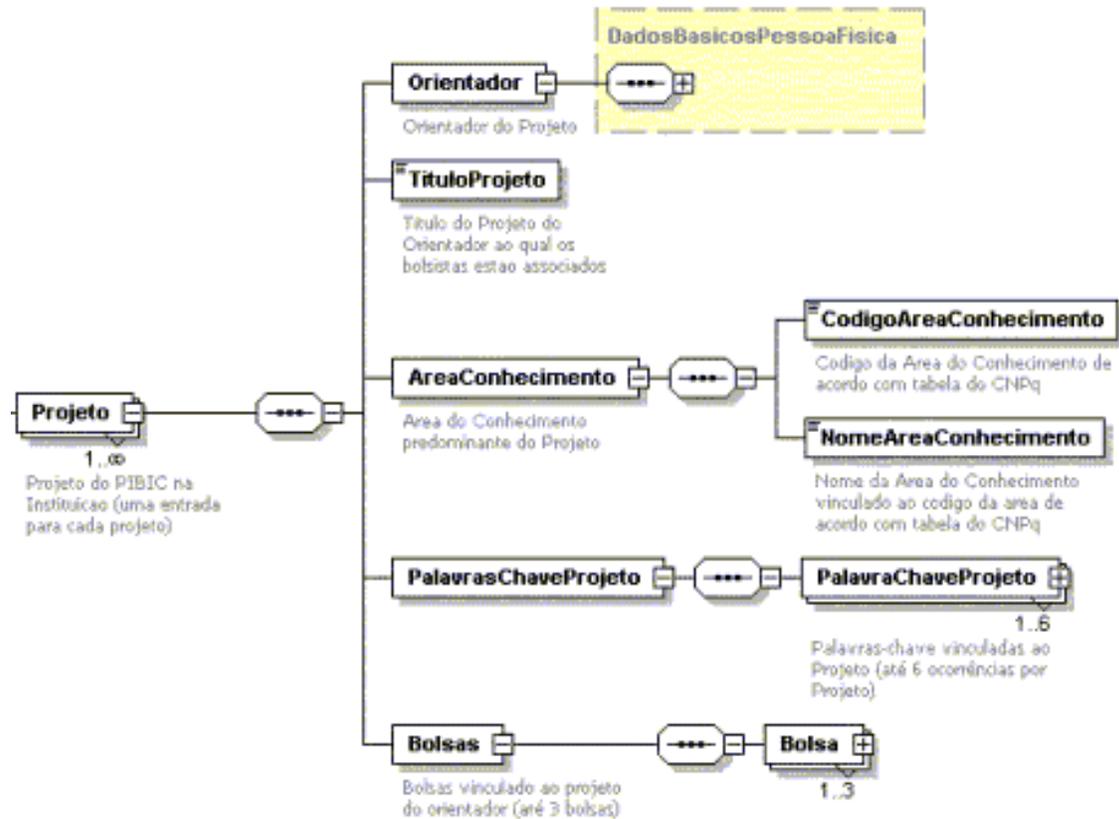


Figura 3.6: Organização dos dados de qualquer projeto de pesquisa

A figura 3.7 mostra como a informação referente a uma “Bolsa” é organizada. Esta descreve a ocupação da cota de bolsa: os dados do bolsista, os dados bancários necessários ao pagamento (figura 3.8) e os dados da bolsa quanto à situação do bolsista e período de concessão (figura 3.9).

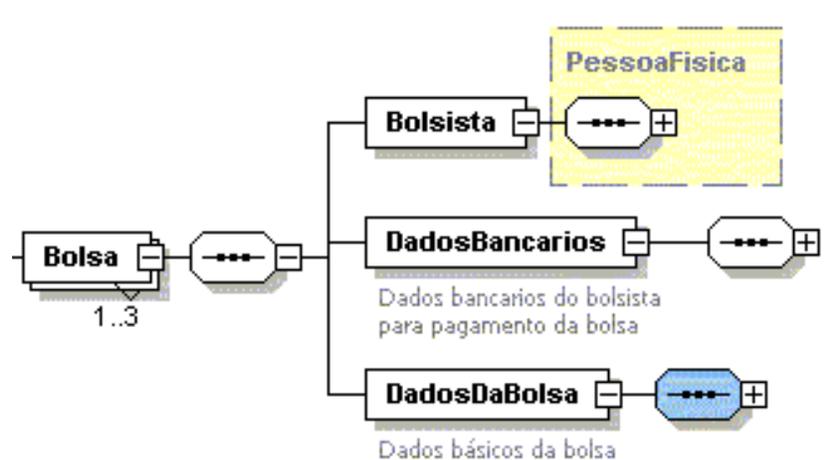


Figura 3.7: Organização dos dados de qualquer bolsa

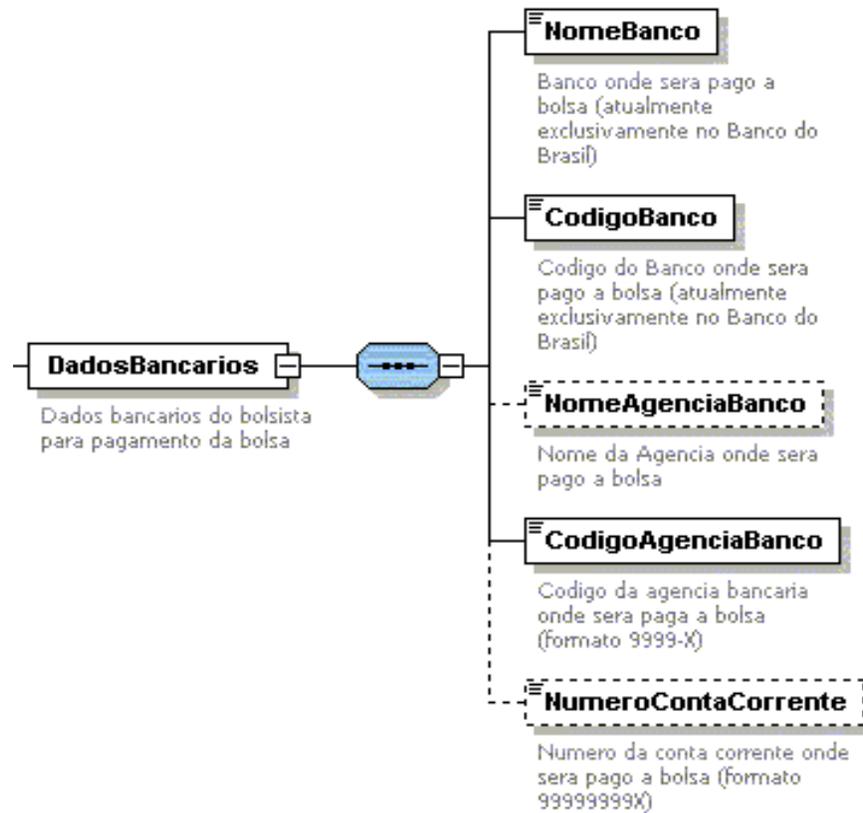


Figura 3.8: Organização dos dados bancários

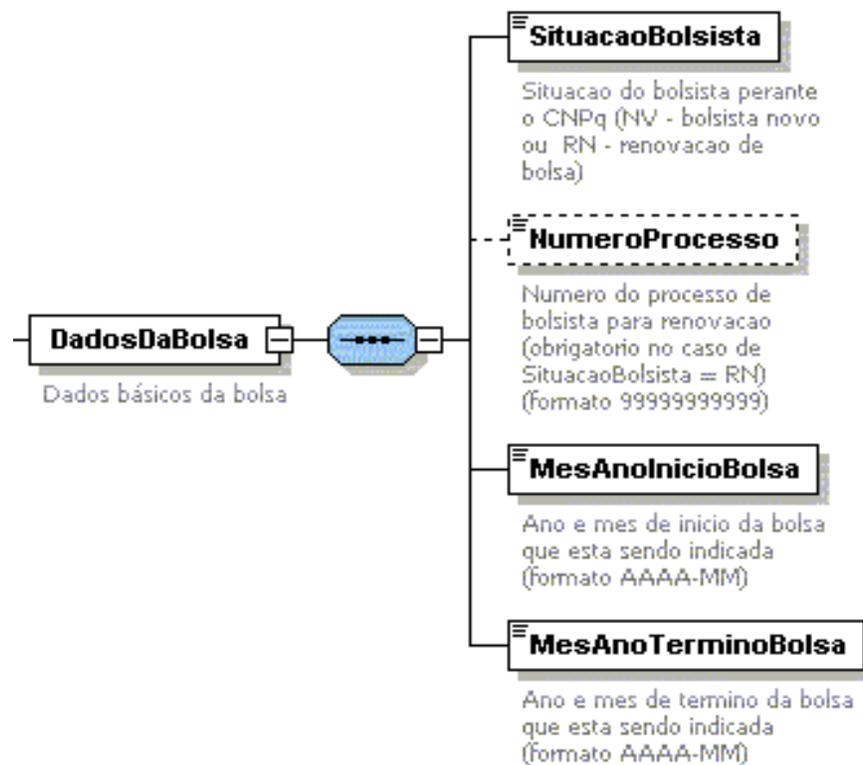


Figura 3.9: Organização dos dados da bolsa

3.2 Modelo de bolsas de IC na UFRGS

O banco da UFRGS integra Ensino, Pesquisa, Extensão e Área Administrativa. Está implementado em um servidor de banco de dados Sybase, e é resultado do Projeto de Democratização da Informação na UFRGS (UFRGS, 2004).

O modelo de dados implementado neste banco de dados suporta diversas modalidades de bolsas presentes na UFRGS, todas baseadas no conceito de quota de bolsas. Este conceito é representado pela entidade BOLSA, a qual é oferecida por um PROGRAMABOLSAS, existe em função de um conjunto de solicitações aprovadas registradas em HISTORICOBOLSA, e contém $n > 0$ quotas, cada quota destinada a uma PESSOA que recebe bolsa, o bolsista.

Os relacionamentos com outras entidades são opcionais, existindo em função da modalidade. A implementação das peculiaridades de cada modalidade, bem como dos programas de bolsas, é de responsabilidade de regras de negócio externas ao banco de dados.

O modelo ER apresentado na figura 3.10, construído através do DBDesigner4 (fabFORCE.net, 2004), exhibe as entidades e relacionamentos principais da modalidade Iniciação Científica.

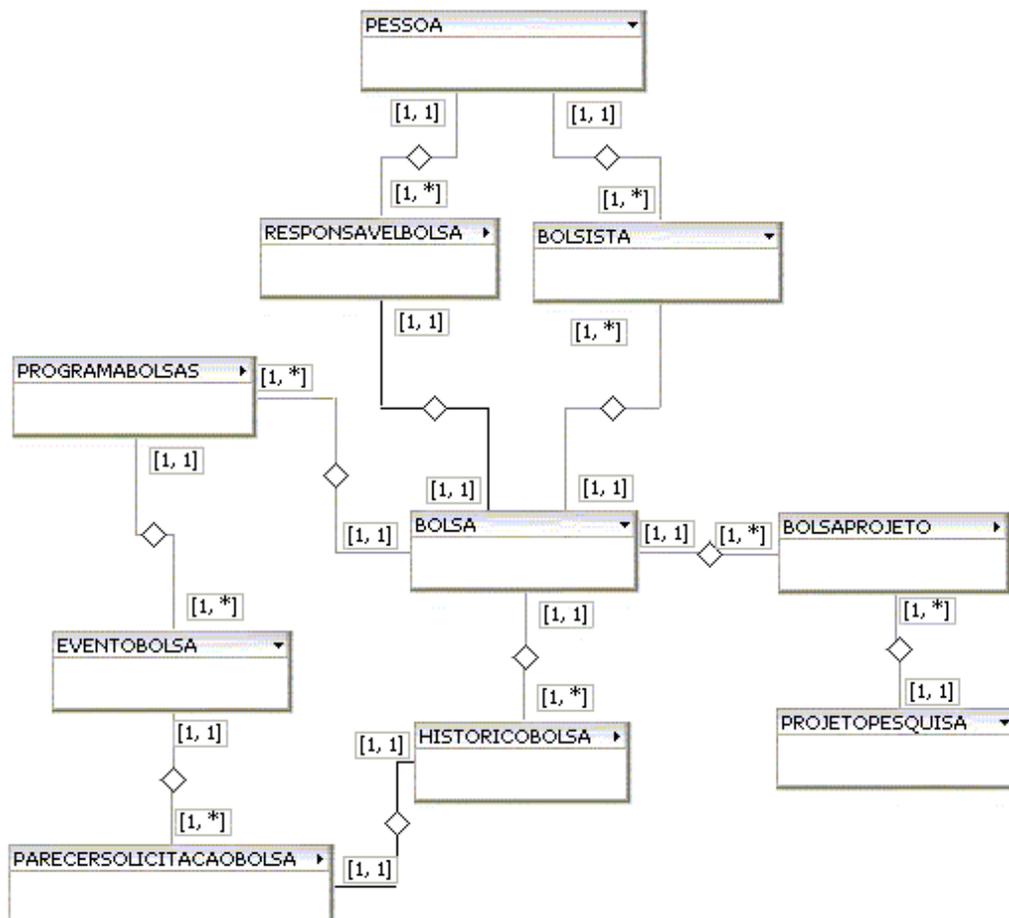


Figura 3.10: Modelo ER das bolsas de IC na UFRGS

Nesta modalidade, uma bolsa é responsabilidade de uma pessoa específica, o orientador do(s) bolsista(s), e está vinculada a um projeto de pesquisa no qual o orientador exerce atividade de pesquisa. No decorrer do tempo, poderão existir diversos projetos vinculados a uma bolsa de IC, mas nunca a bolsa estará associada a mais de um projeto de pesquisa simultaneamente. Da mesma forma, cada uma das quotas da bolsa poderá ser ocupada por vários bolsistas, mas nunca por mais de um bolsista simultaneamente..

As informações mantidas nesse modelo de dados atendem todas as necessidades dos processos de seleção, acompanhamento e avaliação dos programas de IC. Entretanto, para o processo de geração do resultado do processo PIBIC nos moldes definidos pelo CNPq, observou-se que duas informações não estavam disponíveis no banco de dados da UFRGS:

- a informação NumeroProcesso, obrigatória em DadosDaBolsa (figura 3.9) quando de renovação de bolsista, e
- a informação PaisNascimento na forma SiglaPaisNascimento e NomeAbrevPaisNascimento conforme tabela do CNPq, obrigatória em DadosBásicosPessoaFisica.

Em função disso, foram realizados os seguintes ajustes no banco de dados:

- Em relação a informação NumeroProcesso, acrescentou-se à entidade OCUPACAOBOLSAS o atributo NumeroProcessoCNPq, e procedeu-se à carga desse atributo a partir de arquivo enviado pelo CNPq.
- Em relação a informação PaisNascimento, a solução encontrada consistiu em criar uma tabela de mapeamento entre a tabela PAIS presente no banco de dados da UFRGS e a entidade PAIS presente no banco de dados do Lattes Institucional.

3.3 Processo de geração XML PIBIC UFRGS 2004

A interface de geração do XML PIBIC UFRGS foi construída no ambiente de desenvolvimento Delphi versão 7 (BORLAND SOFTWARE CORPORATION, 2002). A escolha deste produto deve-se, principalmente, ao fato das aplicações cliente-servidor de Ensino, Pesquisa, Extensão e Recursos Humanos da UFRGS terem sido desenvolvidas neste ambiente.

O suporte a XML disponibilizado pelo ambiente Delphi fundamenta-se no componente TXMLDocument, que representa a estrutura de um documento XML. Este componente implementa a interface IXMLDocument para acesso aos elementos de um documento XML (SWART, 2004a).

Além de permitir trabalhar com um documento XML usando somente o componente TXMLDocument e a interface IXMLDocument para percorrer ou gerar a estrutura do documento, o Delphi permite gerar código para manuseio de documentos XML através do uso de um *wizard* XML Data Binding (SWART, 2004b) bem como criar *pacotes de dados* a partir de documentos XML baseados em arquivos de transformação XML-BDR através do utilitário XML Mapper (SWART, 2004c).

Para a geração do XML PIBIC optou-se pela solução mais simples em função das características do problema, que consiste:

- no uso do *wizard* XML Data Binding para gerar o modelo de objetos do esquema PIBIC; e
- na construção de uma aplicação cliente-servidor que usa as classes e métodos desse modelo de objetos.

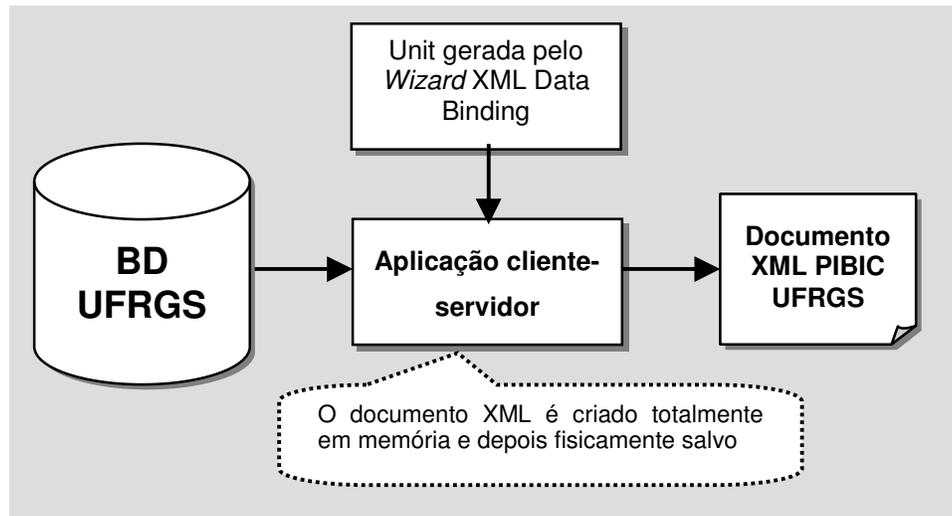


Figura 3.11: Processo de geração do XML PIBIC UFRGS

Para gerar *unit* Delphi contendo a interface *IXMLDocument* que descreve o documento XML através do *wizard* XML Data Binding foram realizados os seguintes procedimentos:

1. Especificação do esquema XML para o qual se desejava gerar as interfaces. O *wizard* XML Data Binding aceita um documento XML exemplo, um arquivo DTD (.dtd), um arquivo XML schema (.xsd), ou um arquivo Reduced XML Data (.xdr). O arquivo usado foi o arquivo XSD enviado pelo CNPq, FormPIBIC_d_Seq.xsd.
2. Examinou-se a forma como o *wizard* representou cada elemento e definiu os métodos *get* e *set*, o mapeamento de tipos XML ↔ Delphi, os prefixos das interfaces, classes e elementos, efetuando-se os ajustes necessários.

A *unit* resultante do processo de mapeamento do esquema PIBIC gerou o modelo de objetos cujas principais classes e atributos estão representados no diagrama exibido na figura 3.12, o qual foi construído através do software Poseidon for UML Standard Edition 2.6 (GENTLEWARE AG, 2004).

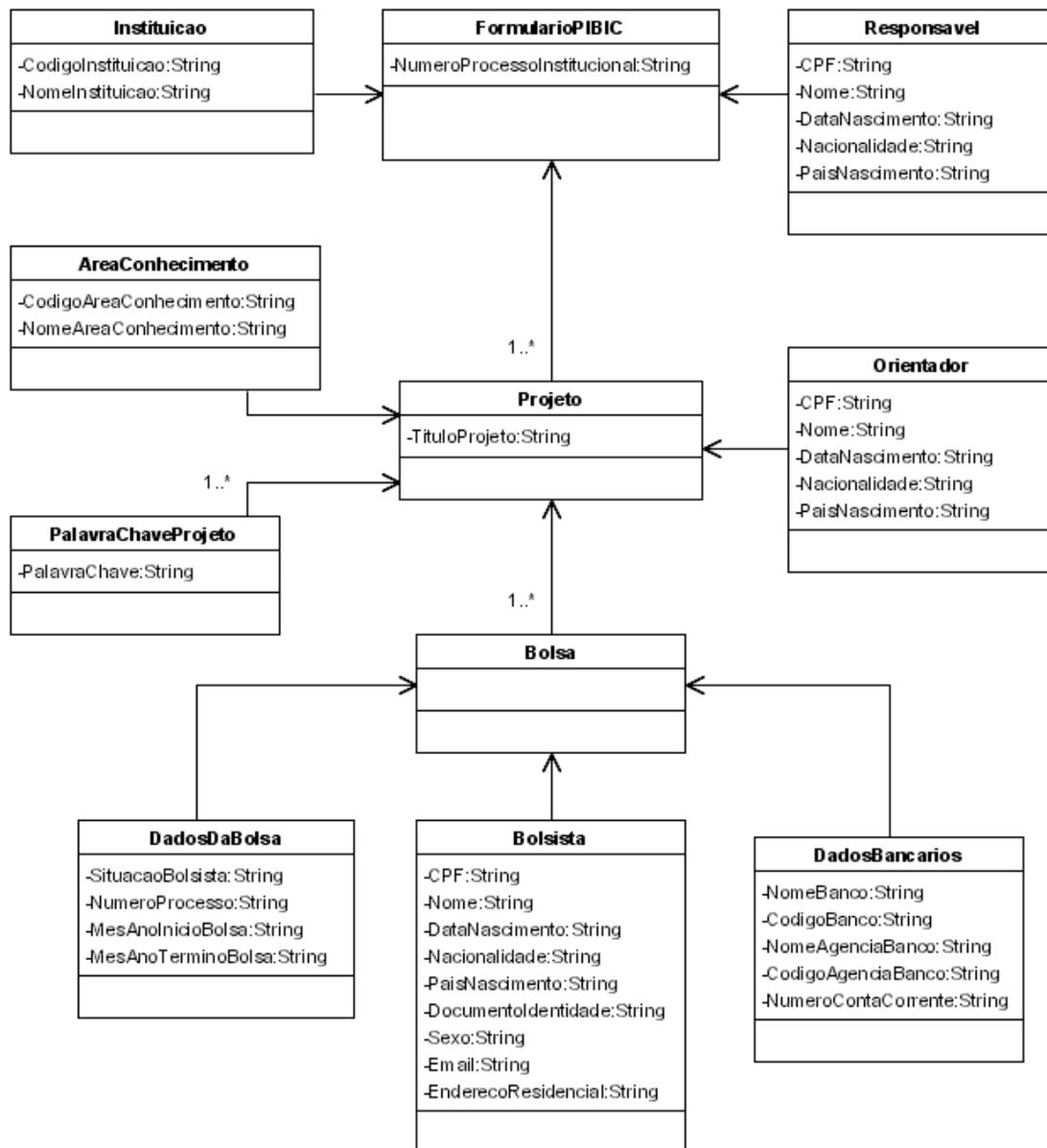


Figura 3.12: Modelo de objetos PIBIC gerado pelo *wizard* XML Data Binding

A aplicação cliente servidor incorpora a *unit* gerada pelo XML Data Binding e instancia um objeto da classe FormularioPIBIC. Realiza, então, consultas ao banco de dados com base no Evento de Solicitação de Bolsas selecionado pelo usuário, utilizando os resultados dessas consultas como argumentos dos métodos *set*. Ao final, salva o documento XML produzido com o nome e local indicado pelo usuário através do uso do método *SaveToFile* do componente *TXMLDocument*.

Durante o processo de geração do documento XML a aplicação exibe as quotas de bolsas recuperadas (projeto e orientador), bem como os bolsistas e palavras-chave associados, como exibido na figura 3.13 .

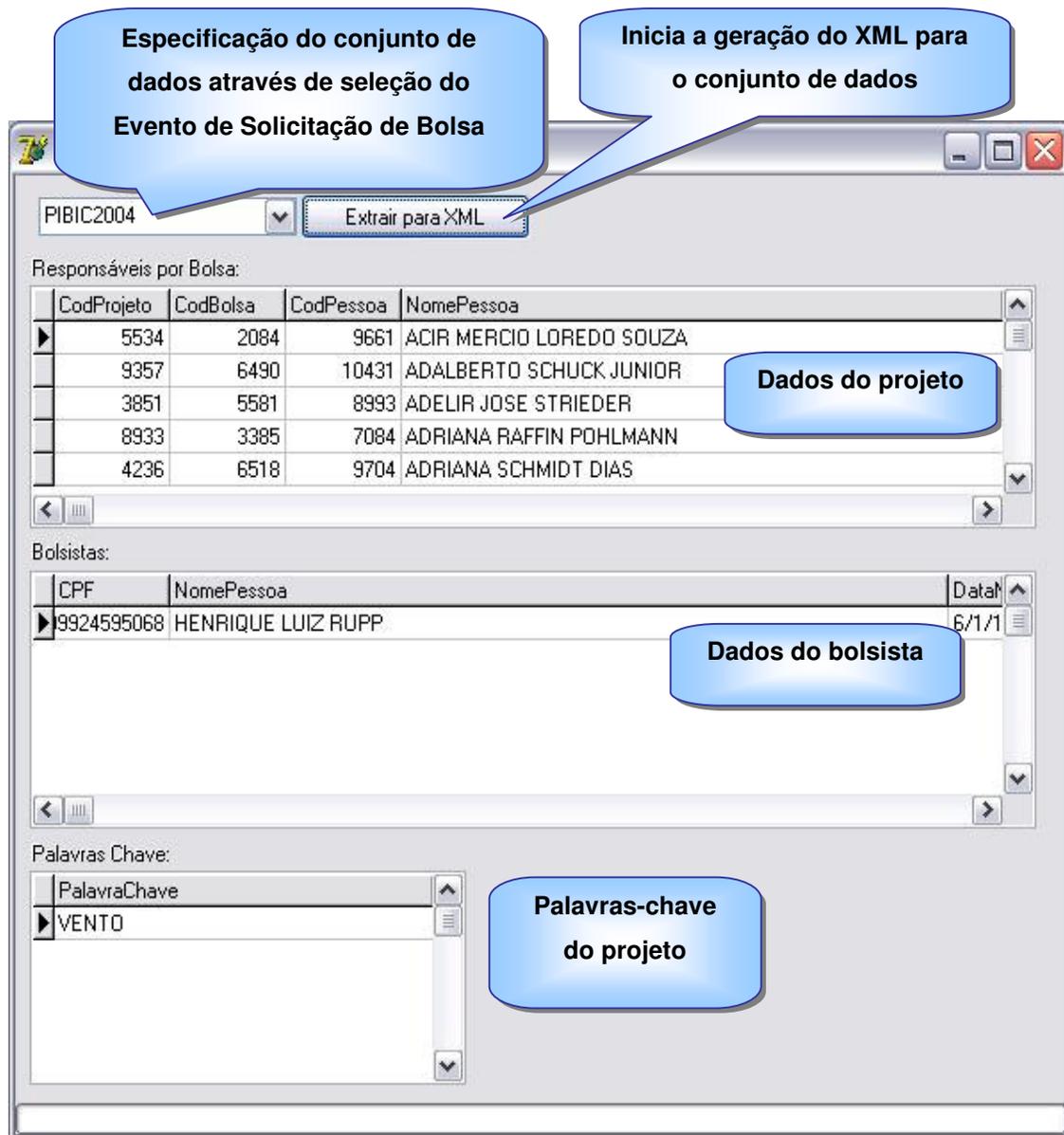


Figura 3.13: Interface do software gerador XML PIBIC UFRGS

Durante a etapa de desenvolvimento da aplicação realizou-se um conjunto de teste com dados fictícios, nos quais os arquivos XML gerados foram analisados através do XML SPY (ALTOVA, 2004).

A validação final consistiu na geração e envio de um arquivo XML gerado em ambiente de teste ao analista de sistemas de informação do CNPq Cristiano Lima Kuppens. O resultado do processamento deste arquivo no ambiente do CNPq possibilitou a realização dos ajustes necessários para que a aplicação executasse no ambiente de produção.

3.4 Exportação XML PIBIC UFRGS

O software gerador do XML PIBIC UFRGS foi executado em ambiente de produção para as 608 quotas aprovadas no evento de solicitação de bolsas PIBIC 2004. Em um tempo de execução de 30 segundos gerou o arquivo PIBIC_UFRGS_2004.xml com informação de 599 quotas. As quotas desconsideradas enquadram-se na situação de dados de bolsistas exigidos pelo CNPq omissos, como informações bancárias e CPF.

Como o do Formulário de Intercâmbio para Propostas do Programa PIBIC/CNPq no formato XML ainda é um projeto piloto, o arquivo XML foi encaminhado ao analista de sistemas de informação do CNPq Cristiano Lima Kuppens via e-mail.

No CNPq, o arquivo foi processado, validado e foram aceitos 597 bolsistas. O motivo da recusa de três bolsistas reside no escopo exclusivo do CNPq: dois na categoria de bolsistas com indicação de bolsa de IC no Programa Projeto Integrado do CNPq e o terceiro com registro de inadimplência em relação ao CNPq. A tabela 3.1 resume os resultados desse processo.

Tabela 3.1: Resultado da exportação PIBIC CNPq UFRGS

	Projetos / Orientadores	Bolsistas
Aprovados UFRGS	599	608
Enviados ao CNPq	591	600
Aceitos pelo CNPq	588	597

Entre a geração e envio do arquivo PIBIC_UFRGS_2004.xml e a resposta do CNPq decorreram apenas 24 horas. Este fato é altamente significativo, especialmente quando se considera que o processo utilizado nos anos anteriores exigiria duas semanas para digitação dos dados das quotas aceitas.

Como o envio do arquivo XML foi feito dois dias antes do prazo final estipulado pelo CNPq e como dos 608 bolsistas indicados apenas onze apresentaram problemas, houve tempo suficiente para a PROPESQ tentar obter os dados faltantes e/ou remanejar as quotas para outros solicitantes na fila de espera, atualizando esses casos diretamente no formulário do CNPq disponibilizado na Web.

4 CONCLUSÃO

Esta monografia apresentou um estudo de estratégias e ferramentas para o uso do XML no intercâmbio de dados armazenados e/ou destinados a bancos de dados relacionais. Nem toda a gama de estratégias foi abrangida, mas apenas aquelas consideradas mais importantes para a meta principal do trabalho.

O trabalho começa com uma introdução aos aspectos de integração XML e bancos de dados relacionais, enfocando as diferenças fundamentais entre essas tecnologias. Após, descreve uma arquitetura para transformação de dados entre XML e SGBDR, bem como as categorias de softwares disponíveis para sua implementação. Por fim, é apresentado um estudo de caso para exemplificar a utilização dessa arquitetura, o processo de geração no formato XML do resultado da seleção de bolsistas PIBIC CNPq UFRGS.

A autora acredita que este trabalho teve dois produtos importantes, além do requisito essencial para formatura no curso de especialização WEB e Sistemas de Informação, que é a monografia propriamente dita.

O primeiro foi possibilitar uma maior compreensão da autora sobre a tecnologia XML e as ferramentas existentes para seu uso na representação e intercâmbio de informações, com potencial utilização profissional desses conhecimentos.

O segundo produto é o próprio estudo de caso, pois seu resultado é um software que reduz sensivelmente o tempo de preparação dos dados a serem enviados ao CNPq, liberando os recursos humanos da PROPESQ para atividades mais relevantes que digitação de dados. Além disso, permitiu que se estendesse o prazo para as etapas de indicação de bolsista e atualização de dados cadastrais o máximo possível.

Por fim, a autora considera que o estudo aqui encerrado pode contribuir para a criação de uma cultura XML no âmbito do CPD da UFRGS, aplicada a projetos como a exportação da produção bibliográfica da UFRGS para a plataforma Lattes Institucional, importação dos dados de Grupos de Pesquisa do CNPq para os bancos de dados da UFRGS, entre outros.

REFERÊNCIAS

ALTOVA. **2004 Release 4 of XMLSPY 2004 Enterprise Edition**. Disponível em: <http://www.altova.com/products_ide.html> . Acesso em: out. 2004.

BIRBECK, M. et al. XML Data Binding. In: BIRBECK, M. **Professional XML (Programer to Programer)**: 2nd ed. Birmingham: Wrox Press, 2001. p. 671-721. Disponível em: <<http://www.drbob42.com/wrox/index.htm>>. Acesso em: out. 2004.

BORLAND SOFTWARE CORPORATION. **Borland Delphi Enterprise Version 7.0**. [S.l], 2002.

BOURRET, R. **Data Transfer Strategies**: Transferring data between XML documents and relational databases. 2001a. Disponível em: <<http://www.rpbouret.com/xml/DataTransfer.htm>>. Acesso em: out. 2004.

BOURRET, R. Mapping DTDs to Databases. **XML.com**, May 2001b. Disponível em: <<http://www.xml.com/lpt/a/2001/05/09/dtdtodbs.html>> . Acesso em: out. 2004.

BOURRET, R. **XML and Databases**. 2004a. Disponível em: <<http://www.rpbouret.com/xml/XMLAndDatabases.htm>>. Acesso em: out. 2004.

BOURRET, R. **XML Data Binding Resources**. 2004b. Disponível em: <<http://www.rpbouret.com/xml/XMLDataBinding.htm>>. Acesso em: out. 2004.

BOURRET, R. **XML Database Products**. 2004c. Disponível em: <<http://www.rpbouret.com/xml/XMLDataBinding.htm>>. Acesso em: out. 2004.

CNPq. **Comunidade LPML**. Disponível em: <<http://lattes.cnpq.br/lmpl/>>. Acesso em: nov. 2004.

DAYEN, I. **Storing XML in Relational Databases**. Disponível em: <<http://www.xml.com/pub/a/2001/06/20/databases.html>>. Acesso em: out. 2004.

DAUM, B; HORAK, C. **The XML Shockwave**. Germany: Software AG, 2000.

FABFORCE.NET. **Fabulous Force Database Desinger 4 for Windows**. Disponível em: <<http://www.fabForce.net>>. Acesso em: nov. de 2004

ERL, T. Integrating XML and databases. In: ERL, T. **Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services**. 2004. Disponível em: <<http://www.xmltc.com/fieldguide/chapter7.pdf>>. Acesso em: out. 2004.

GENTLEWARE AG. **Poseidon for UML**. Disponível em: <<http://www.gentleware.com>> Acesso em: out. 2004.

GUARDALBEN, G.; SHAKU, A. Integrating XML and Relational Database Technologies: A Position Paper. In: HIT SOFTWARE. **White Papers**. Disponível em: <http://www.hitsw.com/products_services/whitepapers/integrating_xml_rdb/integrating_xml_white_paper.pdf>. Acesso em: out. 2004.

GICQUEAU, A. Importing XML documents to Relational Databases using Java. In: HIT SOFTWARE: **White Papers**. Disponível em: http://www.hitsw.com/products_services/whitepapers/importing_xml/ . Acesso em: out 2004.

GRUBER, T.R. A Translation Approach to Portable Ontology Specifications. **Knowledge Acquisition**, [S.l.], v.5, n.2, p.199-220, 1993.

MARCHAL, B. **XML conceitos e aplicações**. São Paulo: Berkeley, 2000.

OBASANJO, D. **XML in Database Management Systems**. Disponível em: <<http://www.25hoursaday.com/StoringAndQueryingXML.html>>. Acesso em: out. 2004.

PACHECO, R. C. S.; KERN, V. M. Uma ontologia comum para a integração de bases de informações e conhecimento sobre ciência e tecnologia. **Ciência da Informação**, [S.l.],v. 30, n.3, p. 56-63, 2001.

SWART, B. XML Document Programming. In: SWART, B. **XML Kylix 2+ BizSnap - XML, SOAP & WebServices**. Disponível em: <<http://www.drbob42.com/kylix/BizSnap1.htm>>. Acesso em: jul. 2004a.

SWART, B. XML Data Binding. In: SWART, B. **XML Kylix 2+ BizSnap - XML, SOAP & WebServices**. Disponível em: <<http://www.drbob42.com/kylix/BizSnap2.htm>>. Acesso em: jul. 2004b.

SWART, B. XML Mapping Tool. In: SWART, B. **XML Kylix 2+ BizSnap - XML, SOAP & WebServices**. Disponível em: <<http://www.drbob42.com/kylix/BizSnap3.htm>>. Acesso em: jul. 2004c.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. PRÓ-REITORIA DE PESQUISA. **A iniciação científica na UFRGS:** um projeto institucional. Porto Alegre:UFRGS, 2003. 98p.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. CENTRO DE PROCESSAMENTO DE DADOS. **A travessia digital:** do processamento de dados para a tecnologia da informação. Porto Alegre:CPD/UFRGS, 2004. 168p.

W3C (WORLD WIDE WEB CONSORTIUM). **Extensible Markup Language (XML) 1.0.** [Feb 4, 2004]. Disponível em: <<http://www.w3.org/TR/2004/REC-xml-20040204/#NT-document>>. Acesso em: jun. 2004.

W3C (WORLD WIDE WEB CONSORTIUM). **XML Schema Part 0: Primer.** [May 2, 2001]. Disponível em: <<http://www.w3.org/TR/xmlschema-0/>>. Acesso em: jun. 2004.

WILLIAMS, K. et al. XML Structures for Existing Databases. In: WILLIAMS, K. **Professional XML Databases.** Birmingham: Wrox Press, 2001. p. 47-66. Disponível em: <http://www-106.ibm.com/developerworks/xml/library/x-struct/> . Acesso em: out. 2004.

ANEXO XML SCHEMA PIBIC

Este anexo exhibe o conteúdo do arquivo XSD que contém a gramática XML para intercâmbio de dados PIBIC.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by CNPq (CNPq) -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
            attributeFormDefault="unqualified">

<xs:complexType name="PessoaFisica">
  <xs:annotation>
    <xs:documentation>Dados Básicos de Pessoa</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="DadosBasicos" type="DadosBasicosPessoaFisica">
      <xs:annotation>
        <xs:documentation>Dados de identificacao de pessoa fisica</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="NumeroCarteiraIdentidade">
      <xs:annotation>
        <xs:documentation>Numero da carteira de identidade (formato X999...999X , tamanho
                           máximo 15)
        </xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="15"/>
          <xs:pattern value="([A-Z]|[0-9])([0-9]{1,13})([A-Z]|[0-9])"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="OrgaoEmissorCarteiraIdentidade">
      <xs:annotation>
        <xs:documentation>Orgao emissor da carteira de identidade da pessoa fisica
        </xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

```

</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:maxLength value="8"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="UFEmissaoCarteiraldentidade">
  <xs:annotation>
    <xs:documentation>UF do orgao emissor da carteira de identidade de acordo com
      tabela do CNPq
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="2"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="DataEmissaoCarteiraldentidade" type="xs:date">
  <xs:annotation>
    <xs:documentation>Data da emissao da carteira de identidade (formato AAAA-MM-DD)
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Sexo">
  <xs:annotation>
    <xs:documentation>Sexo da pessoa fisica (M - masculino ou F - feminino)
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="1"/>
      <xs:enumeration value="M"/>
      <xs:enumeration value="F"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="NumeroPassaporte" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Numero do passaporte do bolsista estrangeiro</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="20"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="DataValidadeVisto" type="xs:date" minOccurs="0">
  <xs:annotation>

```

```

    <xs:documentation>Data limite de validade do visto do estrangeiro
        (formato AAAA-MM-DD)
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="EMail">
  <xs:annotation>
    <xs:documentation>Endereco eletronico do bolsista (formato padrão de e-mail)
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="70"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="EnderecoResidencial" type="Endereco">
  <xs:annotation>
    <xs:documentation>Endereco residencial do bolista</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:element name="FormularioPIBIC">
  <xs:annotation>
    <xs:documentation>Formato de Intercambio para Propostas do Programa PIBIC/CNPq
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NumeroProcessoInstitucional">
        <xs:annotation>
          <xs:documentation>Numero do Processo de Cotas de Bolsas PIBIC da Instituicao no
              CNPq
          </xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{11}"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Instituicao">
        <xs:annotation>
          <xs:documentation>Instituição detentora das cotas PIBIC</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="CodigoInstituicao">

```

```

<xs:annotation>
  <xs:documentation>Codigo da Instituicao na Tabela do CNPq
</xs:documentation>
</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:length value="12"/>
    <xs:pattern value="\d\d\d\d\d\d\d\d\d\d"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="NomeInstituicao">
  <xs:annotation>
    <xs:documentation>Nome da Instituicao na Tabela do CNPq
  </xs:documentation>
</xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="75"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Responsavel" type="DadosBasicosPessoaFisica">
  <xs:annotation>
    <xs:documentation>Responsavel pelo Programa PIBIC na Instituicao
  </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Projetos">
  <xs:annotation>
    <xs:documentation>Projetos do PIBIC na Instituicao</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Projeto" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Projeto do PIBIC na Instituicao (uma entrada para cada
            projeto)
          </xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Orientador" type="DadosBasicosPessoaFisica">
              <xs:annotation>
                <xs:documentation>Orientador do Projeto</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="TituloProjeto">
  <xs:annotation>
    <xs:documentation>Titulo do Projeto do Orientador ao qual os bolsistas
      estao associados
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="255"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="AreaConhecimento">
  <xs:annotation>
    <xs:documentation>Area do Conhecimento predominante do Projeto
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CodigoAreaConhecimento">
        <xs:annotation>
          <xs:documentation>Codigo da Area do Conhecimento de acordo
            com tabela do CNPq
          </xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="\d\d\d\d\d\d\d\d"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="NomeAreaConhecimento">
        <xs:annotation>
          <xs:documentation>Nome da Area do Conhecimento vinculado ao
            codigo da area de acordo com tabela do CNPq
          </xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="85"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="PalavrasChaveProjeto">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PalavraChaveProjeto" maxOccurs="6">

```

```

<xs:annotation>
  <xs:documentation>Palavras-chave vinculadas ao Projeto (até 6
    ocorrências por Projeto)
  </xs:documentation>
</xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element name="PalavraChave" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Bolsas">
  <xs:annotation>
    <xs:documentation>Bolsas vinculado ao projeto do orientador
      (até 3 bolsas)
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Bolsa" maxOccurs="3">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Bolsista" type="PessoaFisica"/>
            <xs:element name="DadosBancarios">
              <xs:annotation>
                <xs:documentation>Dados bancarios do bolsista para
                  pagamento da bolsa
                </xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="NomeBanco" type="xs:string"
                    fixed="Banco do Brasil">
                    <xs:annotation>
                      <xs:documentation>Banco onde sera pago a bolsa
                        (atualmente exclusivamente no Banco do Brasil)
                      </xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="CodigoBanco" type="xs:string"
                    fixed="001">
                    <xs:annotation>
                      <xs:documentation>Codigo do Banco onde sera pago a
                        bolsa (atualmente exclusivamente no Banco do Brasil)
                      </xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="NomeAgenciaBanco" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Nome da Agencia onde sera pago a
      bolsa
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="20"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="CodigoAgenciaBanco">
  <xs:annotation>
    <xs:documentation>Codigo da agencia bancaria onde
      sera paga a bolsa (formato 9999-X)
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="5"/>
      <xs:pattern value="([0-9]{4})(X|[0-9])"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="NumeroContaCorrente" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Numero da conta corrente onde sera
      pago a bolsa (formato 99999999X)
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="9"/>
      <xs:pattern value="([0-9]{1,8})(X|[0-9])"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="DadosDaBolsa">
  <xs:annotation>
    <xs:documentation>Dados básicos da bolsa
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SituacaoBolsista">
        <xs:annotation>

```

```

        <xs:documentation>Situacao do bolsista perante o
        CNPq (NV-bolsista novo ou RN-renovacao de bolsa)
    </xs:documentation>
</xs:annotation>
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:length value="2"/>
        <xs:enumeration value="NV"/>
        <xs:enumeration value="RN"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="NumeroProcesso" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Numero do processo de bolsista
        para renovacao (obrigatorio no caso de
        SituacaoBolsista = RN) (formato 99999999999)
    </xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:length value="11"/>
            <xs:pattern value="[0-9]{11}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="MesAnoInicioBolsa"
    type="xs:gYearMonth">
    <xs:annotation>
        <xs:documentation>Ano e mes de inicio da bolsa que
        esta sendo indicada (formato AAAA-MM)
    </xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="MesAnoTerminoBolsa"
    type="xs:gYearMonth">
    <xs:annotation>
        <xs:documentation>Ano e mes de termino da bolsa que
        esta sendo indicada (formato AAAA-MM)
    </xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

```

        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:complexType name="Endereco">
  <xs:annotation>
    <xs:documentation>Endereço residencial ou profissional</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Logradouro">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="180"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Bairro">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="20"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="CEP">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="9"/>
          <xs:pattern value="d\d\d\d\d-\d\d\d"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Cidade">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="45"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="UF">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:length value="2"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

    </xs:simpleType>
  </xs:element>
  <xs:element name="DDD" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="5"/>
        <xs:pattern value="\d{1,5}"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Telefone" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="10"/>
        <xs:pattern value="\d{1,10}"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Ramal" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="5"/>
        <xs:pattern value="\d{1,5}"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="FAX" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="10"/>
        <xs:pattern value="\d{1,10}"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="DadosBasicosPessoaFisica">
  <xs:sequence>
    <xs:element name="Seq" type="xs:string" default="1"/>
    <xs:element name="CPF">
      <xs:annotation>
        <xs:documentation>CPF de Cidadão Brasileiro ou Naturalizado.
          Cidadãos estrangeiros devem requisitar o CPF de Estrangeiro
          (formato: 99999999999 ou EX99999999)
        </xs:documentation>
      </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="11"/>

```

```

        <xs:pattern value="[0-9]{11}|(EX[0-9]{8})"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Nome">
    <xs:annotation>
        <xs:documentation>Nome do Individuo</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
</xs:element>
<xs:element name="DataNascimento" type="xs:date">
    <xs:annotation>
        <xs:documentation>Data de nascimento do Individuo (formato AAAA-MM-DD)</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="Nacionalidade">
    <xs:annotation>
        <xs:documentation>Nacionalidade do Individuo (B - Brasileiro/Naturalizado ou
            E - Estrangeiro)
        </xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:length value="1"/>
            <xs:enumeration value="B"/>
            <xs:enumeration value="E"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="PaisNascimento">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SiglaPaisNascimento">
                <xs:annotation>
                    <xs:documentation>Sigla do Pais de Nascimento do Individuo de acordo com a
                        tabela do CNPq
                    </xs:documentation>
                </xs:annotation>
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:length value="3"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="NomeAbrevPaisNascimento">
                <xs:annotation>
                    <xs:documentation>Nome abreviado do pais de nascimento do individuo de

```

```
                                acordo com a tabela do CNPq
</xs:documentation>
</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>
```