

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/255634229>

# Gerenciamento de Sistemas de Bancos de Dados XML Nativos utilizando o QuiX

## Article

CITATIONS

0

READS

11

5 authors, including:



[Leonardo Moreira](#)

Universidade Federal do Ceará

32 PUBLICATIONS 38 CITATIONS

[SEE PROFILE](#)



[Jonas Araújo Nascimento](#)

Instituto Federal de Educação, Ciência e Tec...

4 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)



[Javam Machado](#)

Universidade Federal do Ceará

59 PUBLICATIONS 113 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Traffic orchestration based on clusters in urban road networks [View project](#)



Ubiquitous and Pervasive Games [View project](#)

All content following this page was uploaded by [Jonas Araújo Nascimento](#) on 29 September 2015.

The user has requested enhancement of the downloaded file. All in-text references [underlined in blue](#) are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

# Gerenciamento de Sistemas de Bancos de Dados XML Nativos utilizando o QuiX

Leidson O. Lima<sup>3</sup>, Leonardo O. Moreira<sup>1</sup>, Jonas A. Nascimento<sup>1</sup>,  
Flávio R. C. Sousa<sup>1,2</sup>, Javam C. Machado<sup>1</sup>

<sup>1</sup> Departamento de Computação (DC)

Universidade Federal do Ceará (UFC) – Fortaleza, CE – Brasil

<sup>2</sup> Universidade Federal do Ceará em Quixadá (UFC)

Estrada do Cedro, Km 5, CEP 63900-000 – Quixadá, CE – Brasil

<sup>3</sup> Departamento de Engenharia de Teleinformática (DETI)

Universidade Federal do Ceará (UFC) – Fortaleza, CE – Brasil

leidson@gmail.com, {leoomoreira, jonasaraujo, sousa, javam}@ufc.br

**Abstract.** *Native XML DBMSs have been developed in order to manage XML data. Those systems implement common components of traditional DBMSs, like query processing, transactions and concurrence control for XML data. However, only some of them have efficient tools for user interaction, and the available ones have limitations. Web 2.0 has allowed the development of rich, interactive and user-friendly interfaces. This paper presents QuiX, a tool for managing XML data based on Web 2.0. Aiming to evaluate QuiX, some usability tests are presented.*

**Resumo.** *Para gerenciar dados XML, estão sendo desenvolvidos SGBDs XML Nativos. Estes sistemas implementam os componentes existentes nos SGBDs tradicionais, como por exemplo o processamento de consultas, transações e controle de concorrência para dados XML. Contudo, poucos desses sistemas possuem ferramentas eficientes para interação com o usuário, e essas apresentam algumas limitações. A Web 2.0 permite o desenvolvimento de interfaces visuais funcionais e interativas. Este artigo apresenta o QuiX, uma ferramenta para o gerenciamento de dados XML baseado na Web 2.0. Visando avaliar o QuiX, são apresentados alguns experimentos de usabilidade.*

## 1. Introdução

XML [W3C 2009] tem se tornado um padrão amplamente utilizado na representação e troca de dados entre aplicações. Devido a essa crescente utilização do XML, torna-se necessária a utilização de sistemas eficientes de armazenamento e recuperação de dados nesse formato. Para isso, estão sendo desenvolvidos Sistema Gerenciadores de Banco de Dados XML Nativos (SGBDXNs) [Fiebig et al. 2002]. Atualmente, há mais de 50 SGBDXNs disponíveis entre sistemas comerciais e de código livre.

SGBDXNs possuem suporte a armazenamento, indexação, processamento de consultas e transações. Além destas características, é importante que um SGBDXN possua

uma ferramenta de gerenciamento que permita a interação entre os usuários e seus componentes internos. Contudo, poucos SGBDXNs possuem ferramentas eficientes que permitam um completo gerenciamento desses sistemas. Em geral, essas ferramentas são desenvolvidas para o sistema específico, o que dificulta sua utilização por outros SGBDXNs. Isso ocorre porque não há uma padronização de tecnologias envolvidas no acesso aos SGBDXNs.

Aplicativos desenvolvidos para ambientes Web fornecem diversas características, dentre elas podem-se citar a facilidade de acesso e a distribuição de recursos computacionais. Com o crescimento da Web, esses ambientes tornam-se um atrativo para o desenvolvimento de ferramentas de administração de sistemas, pois a facilidade de acesso é importante para se obter a informação sobre o estado do sistema a qualquer instante; para isso, basta ter um navegador e acesso a Web. Já com a distribuição computacional, além de favorecer a descentralização da carga do sistema, pode-se, por meio de um aplicativo Web obter informações de diversos sistemas localizados em diferentes sítios, utilizando-se os navegadores Web.

A Web 2.0 tem sido utilizada para a construção de aplicações de simples manutenção, extensíveis, interativos e de alta usabilidade. Como o gerenciamento de dados XML envolve diversos aspectos, é interessante desenvolverem-se estratégias que facilitem a administração e utilização destes sistemas de gerenciamento.

Assim, a Web 2.0 surge como uma solução nesse contexto para assim prover uma ferramenta que possua uma interface “amigável”, dotada de componentes que facilitem a interpretação das informações, forneça extensibilidade e facilidade de uso. Este artigo apresenta o QuiX, uma ferramenta para o gerenciamento de dados XML. O QuiX utiliza Web 2.0, o que permite que ele tenha interatividade, arquitetura extensível e usabilidade.

O artigo está organizado da seguinte forma. A seção 2 apresenta os trabalhos relacionados. Na seção 3, o QuiX é apresentado e suas funcionalidades discutidas. A implementação e avaliação são apresentadas na seção 4 e, finalmente, a seção 5 apresenta as conclusões.

## **2. Trabalhos Relacionados**

Existem diversas ferramentas para consultar dados XML. Algumas delas utilizam uma abordagem estruturada, a qual é caracterizada pela ausência de flexibilidade na escrita da consulta. Nessa abordagem, as partes que compõem o documento XML são organizadas em um formulário onde o usuário seleciona as informações desejadas, formando a consulta a ser executada. Na abordagem não-estruturada, o usuário formula consultas arbitrárias de acordo com informações do documento ou de um esquema [Braga et al. 2005].

SGBDs XML Nativos como o Natix, Timber [Database 2009] possuem ferramentas para gerenciar alguns aspectos dos SGBDXN, mas não possuem interfaces Web. O eXist possui uma ferramenta *desktop*, que permite controle total do SGBDXN e outra Web que permite manipular documentos e executar consultas, não possibilitando o gerenciamento completo do SGBDXN. O Tamino possui uma ferramenta Web para gerenciar todos os aspectos do SGBDXN, mas esta ferramenta é proprietária e não pode ser utilizada por outros sistemas. O sistema BaseX [Holupirek et al. 2009] apresenta uma ferramenta

*desktop* para a visualização de dados XML, mas não contempla a administração completa de um SGBDXN.

Diferentemente dos trabalhos relacionados, o QuiX é uma aplicação Web de código livre e permite gerenciar todas as funcionalidades de um SGBDXN, como por exemplo iniciar o servidor de banco de dados, criar bases de dados, manipular coleções, índices, usuários, efetuar consultas, dentre outras.

Ferramentas administrativas baseadas na Web são extremamente úteis para fornecer acesso remoto em qualquer máquina que possua uma conexão com a Web e um navegador; além disso, não há necessidade de instalação da ferramenta para todos os usuários, nem a preocupação para manter versões idênticas da ferramenta.

Uma outra vantagem é que o QuiX foi concebido com o objetivo de ser modular, de forma a permitir que alterações mínimas e pontuais, possibilite que a ferramenta possa ser utilizada por outros SGBDXNs. A Figura 1 ilustra uma tabela comparativa, em três contextos, entre os trabalhos relacionados e o QuiX.

	Timber	eXist	Natix	Tamino	BaseX	QuiX
Interface Web	Não	Limitada	Não	Sim	Não	Sim
Controle total dos recursos do SGBDXN	Limitada	Sim	Limitada	Sim	Não	Sim
Portabilidade com outros SGBDXN	Não	Não	Não	Não	Não	Sim

**Figura 1. Comparativo entre os Trabalhos Relacionados e o QuiX**

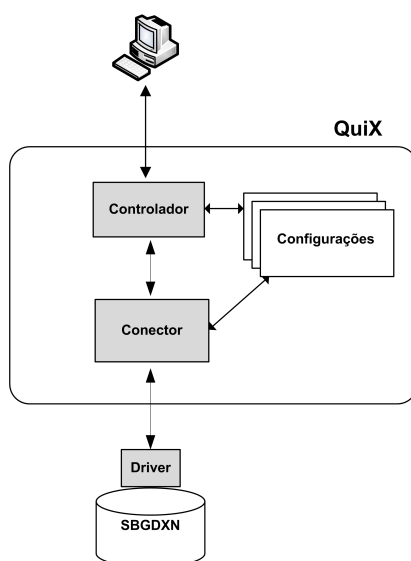
Analisando a tabela comparativa entre os trabalhos relacionados e o QuiX, por meio da Figura 1, pode-se perceber que o QuiX supre todas as necessidades requisitadas por uma ferramenta administrativa de sistema. Essas necessidades são representadas pela disponibilização de um sistema baseado na Web, que oferece recursos para o controle total de um sistema e seja portátil para ser hospedada em qualquer sistema operacional.

### 3. QuiX

O QuiX é uma aplicação Web que possibilita a administração completa de SGBDXNs. Para isso, foi definido um conjunto de *interfaces* de programação para prover a comunicação genérica com esses sistemas. A presente ferramenta foi desenvolvida para fornecer interatividade, usabilidade, extensibilidade e facilidade de acesso para os usuários destes sistemas.

Neste sentido, aplicativos Web, combinados com recursos de Web 2.0, fornecem facilidades de acesso, tornando as interfaces visuais mais atrativas, poderosas e fáceis de usar. Além disso, a Web 2.0 agrega as vantagens de *Web Services* que servem como meio de integração e disponibiliza a extensibilidade para a ferramenta. Uma visão geral da arquitetura do QuiX é apresentada na Figura 2.

O componente *Controlador* recebe a requisição AJAX do navegador do cliente, verifica as configurações definidas pelo usuário e repassa a requisição para o Componente *Conector*. Este recebe a requisição e efetua a conexão com o SGBDXN por meio do *Driver*. Após efetivar o processamento da requisição do cliente, o *Controlador* gera



**Figura 2. Arquitetura do QuiX**

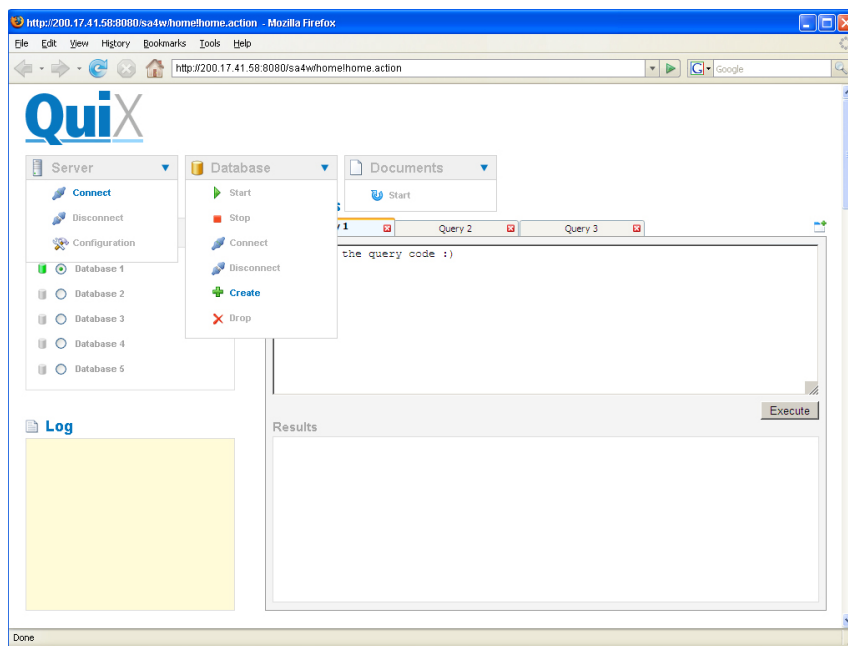
uma resposta no formato XML ou JSON (*JavaScript Object Notation*). O formato da resposta depende do tipo de requisição ou da lógica de negócio. Essa resposta é enviada ao navegador do cliente, que a processa e exibe os resultados.

Para estabelecer a conexão com cada SGBDXN, o componente *Conector* contém as definições de todas as *interfaces* de programação para comunicação com o SGBDXN, de forma a estabelecer a conexão com cada sistema. As *interfaces* de programação contém uma série de métodos que devem ser implementados de acordo com os respectivos *drivers* para o acesso ao SGBDXN.

Com o objetivo de desenvolver uma ferramenta funcional e eficaz, as operações e funcionalidades do SGBDXN foram dispostas em uma tela integrada. A Figura 3 mostra a tela principal do QuiX, que é composta por um *menu* superior com uma barra de atalhos e quatro áreas: navegação baseada na estrutura do documento, edição de consultas, exibição de resultados e informações sobre o *status* das operações executadas.

Para descrever essas funcionalidades, adotamos um fluxo de execução. Inicialmente as informações sobre o SGBDXN a ser utilizado são configuradas. Após isso, o usuário pode iniciar o SGBDXN e criar ou inicializar uma ou mais instâncias de bancos de dados. Para adicionar novos documentos, o usuário seleciona o documento e algum esquema associado como DTD ou XSD, caso deseje validar o documento, como mostra a Figura 4(a). É possível adicionar documentos sem esquema e, nesse caso, uma estrutura de *DataGuide* [McHugh and Widom 2009] é gerada, e então pode-se navegar pela representação da estrutura gráfica do documento. Esta estrutura é hierárquica e permite a visualização das informações, de acordo com o tipo do dado, como por exemplo a raiz, os elementos e os atributos. Outra possibilidade disponível é criar coleções e índices associados a um documento XML.

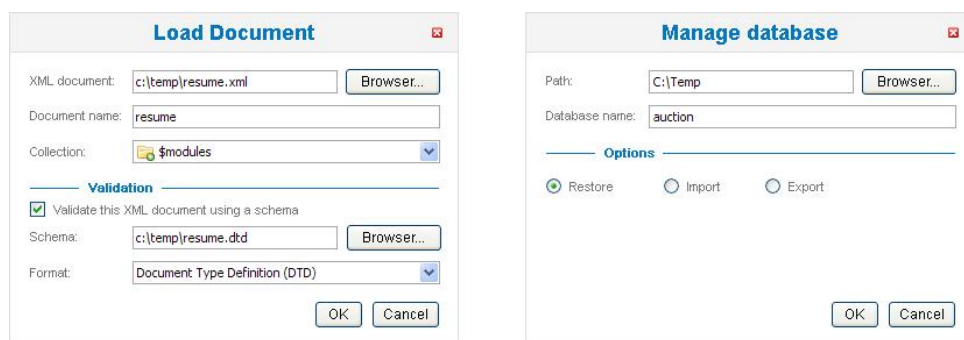
O usuário pode formular suas consultas com base nas informações disponibilizadas pela estrutura hierárquica. No momento em que as consultas estão sendo formuladas, as palavras reservadas da linguagem de consulta são reconhecidas, auxiliando no enten-



**Figura 3. Tela Principal do QuiX**

dimento da sintaxe.

Ao submeter a consulta, caso esta seja válida, os resultados são exibidos no formato XML, sendo informada a duração da execução da mesma no SGBDXN. Caso ocorra algum problema na execução da consulta, as informações recuperadas do SGBDXN são exibidas, de forma destacada, na área inferior esquerda da tela principal. Caso o usuário deseje submeter uma consulta novamente, uma lista com as últimas consultas executadas estará disponível. O sistema de submissão de consultas da ferramenta é baseado em “abas”. Com isso, podem-se formular diversas consultas ao mesmo tempo.



**Figura 4. (a) Carregar documento e (b) Backup**

Teclas de atalho podem ser usadas para facilitar o acesso às funcionalidades desejadas. Para facilitar as configurações do SGBDXN, o QuiX disponibiliza uma tela onde podem ser configuradas informações específicas, como a porta de comunicação, senhas, etc. Podem-se criar usuários e definir permissões para eles. Com relação à parte de *backup*, QuiX apresenta a opção de importar, exportar e restaurar uma base de dados, de acordo com a Figura 4(b).

#### 4. Implementação e Avaliação

A fim de prover portabilidade ao QuiX optou-se por utilizar a plataforma Java para seu desenvolvimento. Java possui componentes já existentes para a manipulação de XML, tornando a aplicação modular e manutenível, o código fonte legível e simples para a adição de novas funcionalidades às aplicações. O QuiX utiliza em seu projeto a biblioteca DOM do W3C para realizar a validação de documentos XML, segundo seus respectivos esquemas.

Os *Rest Web Services* [Pautasso et al. 2008] são utilizados para prover o fluxo de controle nas requisições e envio dos resultados. Estas tecnologias agregam uma série de vantagens tais como a utilização de padrões de projeto, facilidade de implementação e manutenção. Além disso, disponibilizam o recurso para integração de novos serviços, fornecendo a extensibilidade da ferramenta.

Com *Rest Web Services*, é possível agregarem-se novas funcionalidades mesmo que estas estejam fisicamente disponíveis. Com isso, podem-se distribuir os componentes da ferramenta, representados por serviços, em diversos nós do sistema distribuído. Isso só pode ser feito em serviços que não necessitem obter acesso local ao SGBDXN. Os *Rest Web Services*, no contexto da ferramenta, são responsáveis por receber as requisições dos clientes, tratar, realizar computações e devolver resultados em formatos JSON ou XML a serem processados pelos navegadores Web dos clientes.

A arquitetura do QuiX exige processamento tanto no lado cliente quanto servidor. Isso é vantajoso sob o ponto de vista de eliminar processamento desnecessário no lado servidor, fornecendo assim um melhor desempenho da ferramenta. Para isso, críticas de entradas de dados, formatação e organização das respostas são feitas no lado cliente por meio da linguagem JavaScript.

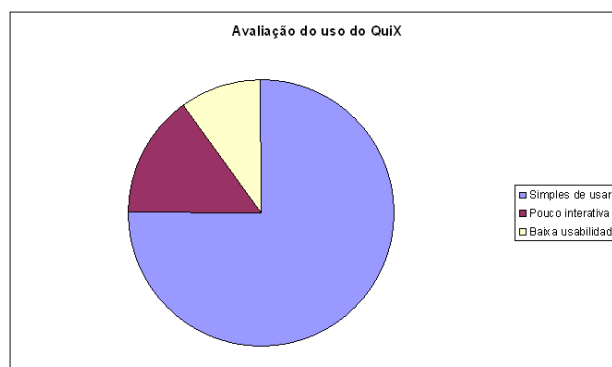
Todas as requisições que partem dos clientes são realizadas por meio de requisições AJAX. A utilização de recursos AJAX fornece interfaces visuais mais amigáveis, rápidas e poderosas, além de eliminar os carregamentos desnecessários de partes ou códigos das páginas Web.

O fato de QuiX ser executado e gerenciado por um Java *Web Container*, surgem algumas vantagens, como por exemplo a responsabilidade pelo ciclo de vida dos componentes Web, otimizações, uma série de serviços adicionais e facilidades de instalação e atualização dos aplicativos. O QuiX pode ser gerenciado por qualquer Java *Web Container* [Sun 2009b], como por exemplo o Glassfish [Sun 2009a] e o Apache Tomcat [Apache 2009].

A ferramenta acompanha, por padrão, a implementação do Conector para o Sedna, e está disponível para Windows, Linux e Mac OSX. Para avaliar o QuiX, primeiramente, foram implementadas *interfaces* de programação definidas para o SGBDXN Sedna [Fomichev et al. 2006], de forma a gerenciar este SGBDXN. O Sedna é de código livre desenvolvido pelo ISPRAS/Rússia que possui todas as características de um SGBDXN, desde o armazenamento, indexação, controle de concorrência, dentre outras.

Em seguida, com o objetivo de avaliar a usabilidade do QuiX, foram disponibilizados para 40 alunos no início e final da graduação, a ferramenta QuiX, um tutorial sobre as funcionalidades desta e um questionário contendo 10 perguntas objetivas.

De acordo com o resultado dos questionários, 75% dos usuários afirmaram que o QuiX é simples de usar, 15% que a ferramenta auxilia no gerenciamento dos dados XML, mas poderia ser mais interativa, e 10% tiveram dificuldade na utilização da ferramenta. Neste último resultado, os alunos consultados eram do início da graduação. Acredita-se que isso ocorreu porque esses alunos tinham poucos conhecimentos de XML. A Figura 5 ilustra de forma simplificada os resultados da avaliação.



**Figura 5. Resultados da Avaliação do QuiX**

## 5. Conclusões

Este trabalho apresentou o QuiX, uma ferramenta para o gerenciamento de SGBDXNs, concebida para ser integrada a qualquer SGBDXN, e que permite a administração completa do sistema, como funcionalidades para criar um banco de dados, executar consultas, gerenciar usuários, navegar e validar documentos. Para avaliar a ferramenta, foi conduzido um estudo de usabilidade da ferramenta. Os resultados iniciais deste estudo mostraram que o QuiX é fácil de usar e eficaz para a maioria dos usuários.

Como trabalhos futuros, pretende-se adicionar a técnica de complemento automático de consultas, melhorando ainda mais a interação com o usuário. O QuiX foi avaliado com o Sedna, mas pode ser utilizado por qualquer SGBDXN, visto que é necessário implementar apenas as *interfaces* de programação definidas pelo componente *Conector*. Nesse contexto, é interessante realizar uma ampla divulgação da ferramenta esperando que os desenvolvedores sintam-se motivados a implementar *interfaces* de programação de forma que o QuiX suporte diversos SGBDXNs.

Alguns sistemas fornecem informações sobre o plano de execução das consultas; outro direcionamento seria exibir este plano para execução das consultas e criar um *plugin* associado a uma *Integrated Development Environment* (IDE) para facilitar a utilização da ferramenta pelos desenvolvedores.

## Agradecimentos

Os autores agradecem ao ISPRAS/Rússia pela colaboração neste trabalho.

## Referências

Apache (2009). *Apache Tomcat*. <http://tomcat.apache.org/>.

Braga, D., Campi, A., Ceri, S., and Raffio, A. (2005). Xqbe: a visual environment for learning xml query languages. In *SIGMOD '05*, pages 903–905. ACM.



Database, X. (2009). *XML Database*. [http://en.wikipedia.org/wiki/XML\\_database](http://en.wikipedia.org/wiki/XML_database).

Fiebig, T., Helmer, S., Kanne, C.-C., Moerkotte, G., Neumann, J., Schiele, R., and Westmann, T. (2002). Anatomy of a native xml base management system. *The VLDB Journal*, 11(4):292–314.

Fomichev, A., Grinev, M., and Kuznetsov, S. (2006). Sedna: A native xml dbms. In *SOFSEM 2006*, volume 3831 of *Lecture Notes in Computer Science*, pages 272–281.

Holupirek, A., Grün, C., and Scholl, M. H. (2009). Basex and deepfs joint storage for filesystem and database. In *EDBT '09*, pages 1108–1111, New York, NY, USA. ACM.

McHugh, J. and Widom, J. (2009). Query optimization for xml. In *VLDB'99*.

Pautasso, C., Zimmermann, O., and Leymann, F. (2008). Restful web services vs. "big" web services: making the right architectural decision. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 805–814, New York, NY, USA. ACM.

Sun (2009a). *Glassfish Application Server*. <https://glassfish.dev.java.net/>.

Sun (2009b). *Java EE at a Glance*. <http://java.sun.com/javaee/>.

W3C (2009). *Extensible Markup Language*. <http://www.w3.org/XML/>.