

SDK - Software Development Kit - v.1.0

Versionamento (nível programação)

Added by [Luís Guilherme Havryluk](#), last edited by [Zenon Garcia](#) on Feb 03, 2012 ([view change](#))

Labels

[versionamento](#), [metadados](#)

Modelagem

[Modelagem Arquitetura](#)

[Casos de Uso \(Application e Version\)](#)

Versionamento Metadados

O objetivo deste documento é descrever o funcionamento e os componentes que fazem parte da funcionalidade de versionamento metadados.

Componentes Envolvidos

Para o correto funcionamento desta inovação vários frameworks foram utilizados e colocados para trabalhar em conjunto, sendo que, cada um com seu escopo e responsabilidades bem definidas. Sendo assim segue a lista:

Framework	Descrição
Hibernate	Realiza a persistência dos objetos em base e controla as versões.
XStream	Gerador de Objetos em XML e XML em objetos.
Java Persistence	Unido ao Hibernate para persistência de dados.
Arquitetura Metadados	Classes básicas para abstrair implementações agilizando o desenvolvimento do framework.
XML	Formato das exportações versionados mantendo um padrão com os objetos que são persistidos.

Funcionamento

No atual momento em que este documento está sendo escrito existem três situações de versionamento e estas funcionam da seguinte forma.

Versão do Objeto igual a versão do Usuário logado

Neste caso, o sistema funciona normalmente e não gera nenhum tipo de versionamento. Tudo que é alterado é refletido na base sem versionar arquivos.

Home

Específicos

Flex

Getting Started

Java

Processo Datasul

Progress

Tools

Trilhas

User

Page Operations

Browse Space

Versão do Objeto *menor* a versão do Usuário logado.

Neste caso, o sistema gera a versão do objeto em formato XML e atualiza a versão do objeto na base com a versão do usuário logado. (Acontece para delete e insert). No entanto, para o delete a versão de delete também é atualizada na base.

Versão do Objeto *maior* que a versão do Usuário Logado

Neste caso, o sistema considera o objeto como "*Somente Leitura*" e se o usuário quiser verificar e executar é possível, no entanto, essa visualização e execução acontece através da leitura do XML e não pela base.

Objetos versionados - Pais e Filhos

Dentro do projeto metadados existem os objetos de persistência e cada qual é incluído em um grupo de versionamento, isto é, os pais e os filhos.

Exemplo: o objeto MDApplication é a persistência de uma aplicação. Este objeto é um Pai de outros objetos Filhos, exemplo Form e Libraries, assim, o funcionamento do versionamento, para esse caso, deve atender a alteração dos filhos. Em adição, ao alterar o Pai a arquitetura deve atender também o versionamento dele mesmo.

Desta maneira, para o caso acima, ao alterar uma aplicação (versão objeto menor versão usuário) é gerado o xml da aplicação antes da persistência das alterações realizadas. Além disso, se a atualização tenha sido feita em um filho dessa aplicação, o xml da mesma deve ser gerado e a base atualizada para esse objeto pai.

Sendo assim, podemos resumir que nunca será versionado o filho e sim o pai e o XML versionado conterá o objeto filho, mas, temos algumas exeções como APPLICATION->FORM, DATABASE->TABLE, esses relacionamentos são considerados PAI->PAI e nesse caso, os itens FORM e TABLE não são versionados no mesmo XML do pai APPLICATION ou DATABASE respectivamente.

Concluindo, toda nova entidade criada no metadados deve levar essas análises em pauta para o versionamento funcionar corretamente. **Ver as modelagens em anexo.**

Próximas Inovações

Quando este documento foi escrito era preciso uma inovação de merge de objetos e esse deve ser implementado nas próximas versões do produto.

Children (4) [Hide Children](#) | [View in Hierarchy](#)

-  [Cadastro de Versões](#)
-  [Ferramenta Gerenciador de Estados](#)
-  [Versionamento e o Gerenciador de Metadados](#)
-  [Versionamento e o Processo de Desenvolvimento](#)

 [Add Comment](#)