

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

LETICIA CORREIA

**Controle de Versão para Armazenamento do Contexto da Interação  
do Aprendiz com Objetos de Aprendizagem**

PROJETO DE TRABALHO DE CONCLUSÃO DE CURSO DO CURSO SUPERIOR  
DE TECNOLOGIA EM SISTEMAS PARA INTERNET

GUARAPUAVA

2016

LETICIA CORREIA

## **Controle de Versão para Armazenamento do Contexto da Interação do Aprendiz com Objetos de Aprendizagem**

Projeto do Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 1, do Curso Superior de Tecnologia em Sistemas para Internet – TSI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Guarapuava, como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Dr. Diego Marczal

GUARAPUAVA

2016

## RESUMO

CORREIA, Leticia. Controle de versão para armazenamento do contexto da interação do aprendiz com objetos de aprendizagem. 34 f. Projeto do Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2016.

A tecnologia está ganhando cada vez mais espaço e importância na educação, indo desde o uso de computadores até softwares específicos para o ensino e aprendizagem. Dentre os projetos com foco na educação, existem as ferramentas de autoria para criação e disponibilização de objetos de aprendizagem. A FARMA é uma ferramenta de autoria que tem o foco na recuperação do contexto dos erros e acertos do aprendiz, elemento fundamental para a análise da aprendizagem de um indivíduo. Para armazenamento dessas informações a FARMA gera uma grande redundância de dados, o que torna lenta a recuperação dos contextos de erros e acertos. Com isso, esse trabalho visa diminuir a redundância dos dados mantendo o acesso aos contextos de interação. Assim, será proposta uma nova modelagem do banco de dados da FARMA que forneça um controle de versão dos objetos de aprendizagem.

**Palavras-chave:** Ferramenta de Autoria. Retroação à Erros. Controle de Versão.

## ABSTRACT

CORREIA, Leticia. Version control for learner interaction context storage with learning objects. 34 f. Projeto do Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2016.

The technology is gaining more space and importance in education, going from the use of computers and specific softwares for teaching and learning. Among the projects focused on education, there are authoring tools to create and provision learning objects. The FARMA is an authoring tool that has the focus in the context of error recovery and apprentice hits, a key element for the analysis of learning of an individual. To store this information FARMA generates a large redundancy data, which slows the recovery of contexts errors and successes. So, this work aims at reducing redundancy of data while maintaining access to interaction contexts. So will be proposed a new model of FARMA database that provides a version control of learning objects.

**Keywords:** Authoring tool. Retroaction to Errors errors. Version control.

## SUMÁRIO

1. INTRODUÇÃO.....	6
1.1 OBJETIVOS.....	9
1.1.1 Objetivo geral.....	9
1.1.2 Objetivos específicos.....	9
1.2 JUSTIFICATIVA.....	10
2. A FARMA.....	11
3. ESTADO DA ARTE.....	12
3.1 REDUNDÂNCIA DE DADOS.....	13
3.2 VERSIONAMENTO DE DADOS.....	14
4. PROCEDIMENTOS METODOLÓGICOS.....	15
5. DESENVOLVIMENTO DA PESQUISA.....	16
5.1 TECNOLOGIAS ENVOLVIDAS.....	16
5.1.1 Banco de dados não relacionais.....	17
5.1.2 Banco de dados relacionais.....	18
5.1.2.1 Projeto de banco de dados relacionais.....	20
5.2 MODELAGEM DO BANCO DE DADOS ATUAL DA FARMA.....	22
5.3 ARQUITETURA DA SOLUÇÃO PROPOSTA.....	26
5.3.1 Protótipo de versionamento.....	26
6. RESULTADOS PARCIAIS.....	29
7. CRONOGRAMA DE EXECUÇÃO.....	30
8. CONSIDERAÇÕES FINAIS.....	31
REFERÊNCIAS.....	32

## 1. INTRODUÇÃO

De acordo com Blackboard (2016), há um grande movimento relacionado ao uso da tecnologia como qualificador na experiência de ensino para alunos e instituições. Como consequência, o interesse em pesquisas sobre recursos educacionais digitais tem aumentado, com o propósito de centrar esses recursos no ensino do aprendiz e no acompanhamento das suas atividades pelo professor.

Existem vários projetos voltados à educação, dentre eles estão o PROINFO<sup>1</sup>, Tablet Educacional<sup>2</sup> e EDUCA<sup>3</sup>. Da mesma forma, como softwares específicos para disponibilização de conteúdo educacional ao aprendiz pode-se citar o *Learning System Manager* (LMS) Moodle<sup>4</sup>, “um dos mais utilizados em ambientes de ensino” (MARCZAL, 2014, p.117), e o Blackboard<sup>5</sup>. Também existem repositórios de softwares educacionais como o Portal do Professor<sup>6</sup>, Banco Internacional de Objetos de Aprendizagem<sup>7</sup>, dentre outras iniciativas como o KHANACADEMY<sup>8</sup>.

Dentre os recursos educacionais, o que ganhou maior destaque nos últimos anos é o Objeto de Aprendizagem (OA). Um OA é todo material em formato físico ou digital que busca apoiar o processo de ensino e aprendizagem (GAMA, 2007). A sua criação é feita a partir da fragmentação do currículo de um curso em diversas partes, onde cada parte abrange um conteúdo específico e bem definido. Essa característica o torna flexível e de baixo custo, permitindo que seja facilmente reutilizado. De acordo com Marczal (2015), a popularidade dos OAs está crescendo tanto em pesquisas quanto em aplicações, contando em alguns casos até mesmo com a aplicação de Inteligência Artificial.

*A Ferramenta de Autoria para Objetos de Aprendizagem com Mobilidade na Aprendizagem* (FARMA), é uma ferramenta que permite a criação de OAs para o ensino de matemática. De acordo com Marczal (2015), esta tem o diferencial de

---

1 Disponível em: <http://portal.mec.gov.br/proinfo>

2 Disponível em: <http://www.fnde.gov.br/tableteducacional/inicio>

3 Disponível em: <http://www.portaleduca.com.br/>

4 Disponível em: <https://moodle.org/>

5 Disponível em: <http://blackboard.grupoa.com.br/>

6 Disponível em: <http://portaldoprofessor.mec.gov.br/index.html>

7 Disponível em: <http://objetoseducacionais2.mec.gov.br/>

8 Disponível em: <https://pt.khanacademy.org/>

facilitar a criação de OAs dinâmicos mesmo por pessoas leigas em computação e também realizar a retroação<sup>9</sup> aos erros do aprendiz, onde o professor-autor poderá identificar os equívocos do aprendiz e propor soluções para auxiliá-lo na sua superação.

Para poder viabilizar a retroação aos contextos de erros e acertos, a FARMA armazena uma cópia do contexto atual do OA para cada resposta do aprendiz. Isso permite que quando o professor ou o aprendiz acessar uma resposta, tenha exatamente a mesma visão de quando o aprendiz a respondeu, ou seja, cada resposta manterá todo o contexto mesmo que alguma parte do OA seja alterada pelo professor-autor. Devido à isso a FARMA está gerando uma grande redundância de dados. Sabe-se que a quantidade de dados no banco influencia diretamente a velocidade das consultas, principalmente quando a consulta é mais complexa, como por exemplo, uma consulta que exige a relação entre várias tabelas do banco de dados.

Atualmente a FARMA utiliza o banco de dados não relacional MongoDB. Este tipo de banco tem uma maior flexibilidade no armazenamento dos dados e conseqüentemente fácil escalabilidade em ambiente distribuído, porém apresenta um grande consumo de espaço em disco e memória, além de não possuir consistência e integridade de dados garantida (LINSTER, 2014).

A redundância de dados apresentada no salvamento do contexto da interação, acontece pela maneira com que os dados estão estruturados na modelagem atual, necessitando assim de uma remodelagem. Como o MongoDB não se apresentou adequado ao projeto atual da FARMA, devido as suas particularidades citadas acima, optou-se pela troca do banco para o modelo relacional.

Para isso, foi escolhido o Postgresql devido a sua forte consistência de dados, suporte a transações, suporte para campos JSON na versão PostgreSQL 9.4, o que significa que haverá aumento de velocidade, eficiência e flexibilidade nas

---

9 De acordo com Dicionário Online de Português: “1. Ação de voltar ao que era. 2. Ato de retornar ao estado, situação, circunstância anterior ao momento presente”. De acordo com a FARMA, retroação significa poder restaurar a sessão exata da ocorrência de um erro, incluindo todo o contexto do exercício.

consultas sem perder as normas *ACID*<sup>10</sup> que são recursos relevantes para aplicações de missão crítica. Também suporta sistemas de alto rendimento e alto processamento, por conseguir manipular uma grande quantidade de dados com vários usuários simultâneos (ENTERPRISEDB, 2014).

Adicionalmente, o estudo visa ampliar as informações salvas do aprendiz, que atualmente são somente da interação com o OA quando este responder à um exercício. Assim, será feita uma modelagem que contenha os dados relevantes para realizar a auditoria. Ressalta-se que o foco deste trabalho será no versionamento, sendo a modelagem para a auditoria considerada como um estudo adicional que será a base para pesquisas futuras, como por exemplo a modelagem do aprendiz. Segundo Silva (2015), a modelagem do aprendiz tem por finalidade prover uma avaliação eficaz do conhecimento do estudante, detectando pontos onde este tem dificuldades, desatenção e até frustração, para que de acordo com tal avaliação o sistema possa definir o conteúdo mais adequado ao nível de capacidade do aprendiz.

Considera-se a atribuição dessas informações, assim como o versionamento de um OA, como sendo o desafio do projeto, pela complexidade de criar uma solução que realize o acesso e o armazenamento dessas informações no banco de dados de maneira eficiente, sem existir grandes perdas de desempenho.

Assim a contribuição deste projeto será obter uma nova modelagem do banco de dados com um mecanismo de versionamento dos registros relacionados ao OA, de modo a viabilizar o acesso ao contexto de erros e acertos. E adicionalmente, salvar informações relevantes do aprendiz sobre sua interação com OA, para que possam vir a ser utilizadas para análises estatísticas e pesquisas futuras.

A seguir, na Seção 1.1 Objetivos, é apresentado o objetivo geral seguido dos objetivos específicos. Na Seção 1.2 Justificativa, é explicitada a relevância da proposta deste trabalho e suas contribuições. No Capítulo 2 FARMA é apresentado a ferramenta de autoria FARMA. No Capítulo 3 Estado da Arte, são apresentadas algumas ferramentas de autoria, e também são descritos os conceitos e tecnologias

---

10 ACID: Atomicity, Consistency, Isolation, and Durability.



utilizados neste trabalho. No Capítulo 4 são apresentados os procedimentos metodológicos deste trabalho. No Capítulo 5 é apresentado o desenvolvimento da pesquisa, contendo na Seção 5.2 a descrição do banco de dados atual, e na Seção 5.3 o protótipo do mecanismo de versão. No Capítulo 6 é descrito os resultados parciais deste trabalho. No Capítulo 7 é apresentado o cronograma de execução do trabalho. No Capítulo 8 são explicitadas as considerações finais deste trabalho. E por último apresenta-se as referências bibliográficas utilizadas na escrita deste projeto.

## **1.1 OBJETIVOS**

### 1.1.1 Objetivo geral

O presente trabalho tem como objetivo elaborar um mecanismo de versionamento de objetos de aprendizagem, com o intuito de diminuir a redundância dos dados e suas consequências no registro e acesso às informações do contexto de interação do aprendiz com um exercício.

### 1.1.2 Objetivos específicos

São portanto, objetivos específicos deste trabalho:

- A partir do banco de dados físico fazer a descrição da modelagem atual do banco da FARMA;

- Criar um protótipo para a nova modelagem do banco de dados, com suporte à controle de versões do contexto de uma interação do aprendiz com um exercício;
- Definir os novos dados a serem armazenados de modo a permitir a elaboração de análises estatísticas da interação do aprendiz com o OA.

## 1.2 JUSTIFICATIVA

A FARMA é uma ferramenta que busca facilitar a criação de OAs mesmo por pessoas leigas em computação, como também prover o armazenamento dos erros e acertos do aprendiz e todo seu contexto, proporcionando ao professor-autor a capacidade de realizar a análise da interação do aprendiz no OA, e poder lidar de forma apropriada com a causa do erro.

Porém atualmente o sistema apresenta uma lentidão no acesso aos dados dos contextos de interação, que pode estar associada a redundância gerada pelo armazenamento da resposta do aprendiz e todo o seu contexto. Pois quando uma resposta é enviada, é feita uma cópia de todas as informações do contexto atual da interação. São salvas todas as informações relevantes da resposta, tais como, a questão respondida junto da resposta, as dicas para aquela resposta, a data e hora da ocorrência daquela resposta, o número de tentativas, o exercício a qual pertence a questão, todas as outras questões do exercício juntamente com as suas respostas e remediações, e ainda as informações mais relevantes do OA, como título e descrição, para que as modificações futuras não interfiram o contexto da resposta já salva.

O problema é que esse procedimento é feito em cada resposta inserida, seja certa ou errada, o que resulta em uma grande quantidade de dados redundantes. Essa redundância está relacionada com a forma de como os dados estão

estruturados no banco, por isso acredita-se que a proposta deste trabalho é relevante para otimizar o registro e acesso aos contextos de interação da FARMA, e minimizar esta redundância através da remodelagem do banco de dados.

## 2. A FARMA

Nesta seção serão apresentados alguns exemplos de ferramentas de autoria para o ensino e aprendizagem em contraste com os objetivos da FARMA, e após serão detalhadas as principais funcionalidades da FARMA.

A CourseLab<sup>11</sup> é uma ferramenta de autoria de software livre, que permite que sejam construídos objetos de aprendizagem semelhantes às apresentações do Microsoft Power Point. Disponibiliza recursos como animações, vídeos, questionários, e permite a programação de ações entre objetos.

O exeLearning<sup>12</sup> é uma ferramenta de autoria que permite a criação de OAs em formato HTML. Disponibiliza aos usuários recursos para criação de questionários, artigos, *Wikibooks*, etc. Também fornece modelos de OAs editáveis com o intuito de facilitar a criação e edição do código HTML dos OAs.

O MyUdutu<sup>13</sup> é uma ferramenta de autoria que tem recursos similares às redes sociais. Disponibiliza recursos de imagens, animação, vídeos e questionários. Permite criar OAs no formato de sequência de páginas. Dentre as ferramentas citadas, é a única que não necessita de configuração ou instalação.

De acordo com Marczal (2014), estas ferramentas se destacam no que diz respeito a produção de arquivos digitais, mas os objetos criados por essas ferramentas tem pouca interatividade com o aprendiz, e os ambientes citados

---

11 <http://www.courselab.com/>.

12 <http://exelearning.org/>.

13 <http://www.myudutu.com/>.

também não tratam os erros cometidos pelo aprendiz.

A Ferramenta de Autoria para Objetos de Aprendizagem com Mobilidade na Aprendizagem (FARMA), é uma ferramenta que busca facilitar a criação de OAs mesmo por pessoas leigas em computação, e prover o armazenamento dos erros e acertos do aprendiz e todo seu contexto, proporcionando ao professor-autor a capacidade de realizar a análise da interação do aprendiz no OA, e poder lidar de forma apropriada com a causa do erro.

O funcionamento da FARMA se divide em três módulos principais: o módulo de autoria, o módulo de interação e o módulo de monitoramento. O módulo de autoria permite a construção de um OA pelo professor-autor, definindo as introduções, exercícios e questões. O módulo de autoria também permite a definição de regras de remediação e mensagens por parte do professor-autor, que são aplicadas para cada questão quando um aprendiz comete erro. O módulo de interação é a interface entre o aprendiz e o OA, onde as introduções, exercícios, questões e remediações são organizadas para compor o objeto de aprendizagem. O módulo de monitoramento é o que abrange a retroação a erros, onde tanto o professor-autor quanto o aprendiz podem voltar ao contexto exato da resolução de uma questão, para obter uma análise detalhada do erro.

### **3. ESTADO DA ARTE**

Neste capítulo são apresentados alguns tipos de redundância de dados, e como ela afeta o desempenho de um sistema, após é explicitada as formas de versionamento de dados, fazendo um comparativo entre as suas características mais relevantes.

### 3.1 REDUNDÂNCIA DE DADOS

A redundância de dados ocorre quando uma mesma informação está armazenada várias vezes em um sistema. Existem dois tipos bem conhecidos de redundância, a redundância controlada e a redundância não controlada.

A redundância controlada de dados ocorre quando o sistema tem conhecimento sobre as múltiplas representações e garante a sincronia entre elas, fazendo com que o usuário tenha a visão de que existe somente uma única representação daquela informação (Heuser, 2009). Esse tipo de redundância serve para melhorar a confiabilidade, pois se uma base de dados falhar, as informações redundantes serão usadas para reconstruir essa base, e também melhora o desempenho, principalmente quando se trata de um ambiente distribuído, permitindo rápido acesso à informação a partir de qualquer uma de suas máquinas.

A redundância não controlada de dados ocorre quando fica a cargo do usuário manter a sincronia entre as diversas representações de uma mesma informação (Heuser, 2009). Isso acarreta problemas como erro de transcrição de dados, pela redigitação da informação em vários lugares, e a inconsistência de dados, que ocorre quando apenas uma representação da informação é modificada, sem as demais serem atualizadas. Dessa forma, a informação aparecerá de formas diferentes no sistema, gerando inconsistência no banco de dados.

No entanto, a redundância que ocorre na FARMA é diferente das anteriores, pois ela se deve à forma de como os dados estão estruturados na modelagem atual. Ela ocorre através do armazenamento das respostas, pois para viabilizar a retroação a FARMA realiza uma espécie de versionamento de todo o contexto da resposta do aprendiz, fazendo uma cópia do exercício, da questão, do número de tentativas, dicas recebidas, etc., isso a cada vez que o aprendiz responde à uma questão. Essa redundância afeta o desempenho das consultas no banco, pois gera uma grande quantidade de dados e causa lentidão no acesso aos contextos de interação. Dessa forma se vê a necessidade da remodelagem do banco de dados para a criação de um mecanismo de controle de versões que minimize essa redundância.

## 3.2 VERSIONAMENTO DE DADOS

Atualmente, desenvolvedores utilizam um sistema de controle de versão para manter as versões dos seus arquivos, que permite reverter desde um arquivo até um projeto para um estado anterior, comparar as mudanças que foram feitas entre um intervalo de tempo, ver quem foi o último a modificar algo que pode estar causando problemas, além de conseguir trabalhar remotamente com outras pessoas que estão utilizando diferentes sistemas (CHACON; STRAUB, 2010).

Assim, pode-se definir versionamento como uma operação que cria novas versões de uma entidade a partir de um conteúdo pré-existente (JUNIOR et. al, 2011).

O conceito de versão é visto como relevante para gerenciar as diferentes representações de uma informação. De acordo com Soares et. al (1993), um mecanismo de versão tem as seguintes características:

1. Permitir aos usuários manter e manipular um histórico de alterações à sua rede.
2. Uma referência a uma entidade refere-se a uma versão específica desta, podendo o usuário acessar desde a versão mais antiga até a versão mais recente dessa entidade.
3. Utilizações concorrentes de uma mesma informação, onde a informação poderá ser compartilhada para uso simultâneo. Assim, cada cópia da informação em uso é considerada uma versão desta informação.

No desenvolvimento de softwares o controle de versão é de fundamental importância, dentre os sistemas de controle de versão mais utilizados destacam-se o CVS (*Concurrent Versions System*)<sup>14</sup>, Subversion<sup>15</sup> e o GIT<sup>16</sup>, que tem o objetivo de identificar, controlar e disseminar as mudanças feitas em um arquivo pelos seus usuários. Eles são capazes de recuperar versões específicas de um arquivo, listar

---

14 <http://www.nongnu.org/cvs/>

15 <https://subversion.apache.org/>

16 <https://git-scm.com/>

as mudanças que foram feitas e qual usuário que as fez. O GitHub<sup>17</sup>, por exemplo, é um repositório utilizado pelos desenvolvedores de software que querem compartilhar o seu projeto com outras pessoas, possibilitando obter uma versão completa deste repositório em sua máquina local, e em cada conjunto de modificações, cria-se uma nova versão dos arquivos modificados no repositório (SPINELLIS, 2012).

Neste trabalho, será desenvolvido uma espécie de mecanismo de controle de versão dos registros dos dados do contexto de interação, para minimizar a redundância gerada pelo salvamento do contexto de interação da FARMA.

#### **4. PROCEDIMENTOS METODOLÓGICOS**

Este capítulo descreve de forma sequencial os passos que serão feitos para o desenvolvimento do projeto apresentado neste trabalho. A seguir, estão os itens que serão realizados neste projeto:

**1. Estudar a ferramenta de autoria FARMA para entendimento das suas principais funcionalidades.**

Realizar um estudo teórico e testes na própria ferramenta para identificar os pontos mais lentos na aplicação.

**2. Modelar o banco de dados atual.**

Será feita uma modelagem da base de dados atual, utilizando o Astah<sup>18</sup> como ferramenta case de apoio.

**3. Organizar um estudo sobre banco de dados PostgreSQL.**

Realizar uma pesquisa sobre o PostgreSQL para identificar as suas principais

---

<sup>17</sup> <https://github.com/>

<sup>18</sup> Astah Community é um software gratuito para modelagem UML. Mais informações disponíveis em: <http://astah.net/>.

vantagens, e características favoráveis para as especificações da FARMA.

#### **4. Organizar um estudo sobre banco de dados da FARMA.**

Fazer a modelagem do banco atual, para ter uma visão geral do banco e suas principais ligações.

Serão feita uma análise no banco de dados atual para identificar os pontos lentos da aplicação. Será considerado também o armazenamento de novos dados que poderão ser necessários no banco para uma melhor análise do perfil do aprendiz.

#### **5. Pesquisar sobre versionamento de informações da aplicação.**

Pesquisar como é feito o versionamento de informações em sistemas, como é a estrutura do banco para armazenar os dados versionados.

#### **6. Estudo das tecnologias a serem utilizadas.**

Serão estudadas as tecnologias a serem utilizadas no projeto, como por exemplo, ferramentas para modelagem e outras que venham ser necessárias.

#### **7. Modelar o novo banco de dados da ferramenta.**

A partir dos dados da análise, elaborar modelo lógico, definir quais tabelas serão necessárias e os relacionamentos que deverão possuir entre as mesmas.

## **5. DESENVOLVIMENTO DA PESQUISA**

### **5.1 TECNOLOGIAS ENVOLVIDAS**

Nesta seção será apresentado os conceitos das principais tecnologias que



serão utilizadas para a realização deste projeto. Será abordado o Modelo Relacional e o Modelo Não Relacional, especificando o MongoDB e PostgreSQL, SGBDs dos modelos, respectivamente.

### 5.1.1 Banco de dados não relacionais

Bancos de dados NoSQL ou não relacionais são sistemas de armazenamento que vieram para suprir necessidades onde os bancos de dados relacionais são ineficazes, foi consequência de um projeto que tinha o intuito de flexibilizar a forte estrutura utilizada pelo modelo relacional (LINDER, 2016). Este tipo de banco de dados não tem conceitos de tabelas, esquemas, SQL ou linhas. Não há transações, nem conformidade com a ACID.

Dentro do modelo não relacional, existe o conceito de banco de dados orientado a documentos, com a característica de trabalhar facilmente com dados complexos, como em casos relacionados a web analytics, e postagens de blogs e comentários. “O MongoDB é um banco de dados orientado a documentos, otimizado quanto a velocidade e escalabilidade, e que pode ser executado em quase todos os ambientes que se deseja usá-lo” (HOWS; MEMBREY; PLUGGE, 2015).

O MongoDB utiliza como modelo de dados a notação JSON (**Java Script Object Notation**), apresentando-se sempre em pares compostos de chave e valor. Hows, Membrey e Plugge (2015) destacam que essa forma armazena um conteúdo estruturado de maneira fácil de entender e de usar. Assim, os dados são armazenados em um único documento, e não há a necessidade de especificar a estrutura do banco de antemão. Esse tipo de banco é chamado de *schemaless* ou sem esquema.

Este tipo de banco de dados permite que sejam adicionados itens como uma lista dentro de um documento, conhecido como *embedded document* (documento embutido). Por exemplo, um documento Person pode ter vários documentos Address embutidos.

Hows, Membrey e Plugge (2015) destacam que um recurso significativo do MongoDB é o *sharding*, muito utilizado em sistemas de larga escala, no qual permite que os dados sejam distribuídos em vários computadores, cada qual estará responsável por uma parte dos dados. Como as queries no MongoDB procuram chaves e valores específicos em um documento, cada servidor verificará o conteúdo e retornará o resultado, garantindo que os dados sejam enviados para o servidor correto, e que as queries sejam combinadas de maneira eficiente.

O MongoDB é um banco de dados dinâmico, em que se pode definir coleções sem predefinir uma estrutura para cada documento. Um documento é constituído de pares chave-valor, e as coleções são como um contêiner onde armazena um conjunto de documentos. Comparando com o modelo relacional, um documento equivale a uma linha, e uma coleção equivale a uma tabela do banco de dados.

### 5.1.2 Banco de dados relacionais

O modelo relacional é considerado o mais utilizado atualmente, sendo a base para os principais SGBDs do mercado, tais como DB2 da IBM, Oracle, Access, SQLServer, etc. No modelo relacional, o banco de dados é definido como uma coleção de uma ou mais relações, que são representadas na forma de tabelas com linhas e colunas. Essa forma facilita a compreensão de como os dados estão dispostos até mesmo por pessoas inexperientes, e possibilita o uso de linguagens de alto nível para realizar consultas complexas nesse tipo de banco (RAMAKRISHNAN, 2008).

Os SGBDs relacionais utilizam uma linguagem padrão para gerenciar o banco de dados, chamada de *Structured Query Language* (SQL). Essa linguagem

possui diversos aspectos, dentre eles:

- Contém a linguagem de manipulação de dados - *Data Definition Language* (DML), usada para formular consultas e inserir, excluir e consultar os registros do banco de dados.
- Contém a linguagem de definição de dados - *Data Definition Language* (DDL), que provê a criação, exclusão e modificação das definições das tabelas. É onde são definidas as restrições de integridade e os índices.
- Provê suporte a gatilhos (triggers) – que são ações que são executadas pelo SGBD sempre que ocorrem alterações no banco de dados que satisfazem as condições especificadas pelo gatilho.

Uma tabela ou relação é composta por um esquema e por uma instância. No esquema da relação é onde se define o nome da tabela, o nome de cada campo ou coluna, e o domínio, que define o tipo do campo. No domínio também são feitas as chamadas restrições de domínio, que impõem condições que devem ser satisfeitas e restringe os valores que podem aparecer nesse campo. A instância é o conjunto de tuplas ou registros, que contém os valores associados a cada campo definido na tabela.

Outra característica importante do modelo de dados são as restrições de integridade (RIs). As restrições de integridade permitem que o SGBD rejeite operações que possam causar dano a integridade dos dados (RAMAKRISHNAN, 2008).

Um recurso importante nos SGBDs Relacionais são os gatilhos. Um gatilho (trigger) é um procedimento chamado automaticamente por um SGBD em resposta a alterações especificadas no banco de dados. Um banco de dados que possui um conjunto de gatilhos associados é considerado um banco de dados ativo. Um gatilho é composto por três itens:

1. Evento: onde é designada qual a alteração no banco de dados que ativa o gatilho;
2. Condição: uma consulta ou teste que é executado quando o gatilho é ativado;

3. Ação: Um procedimento que é executado quando o gatilho é ativado e a condição for verdadeira.

A ação de um gatilho pode examinar as respostas da consulta da parte da condição do gatilho, referenciar os valores antigos e novos das tuplas modificadas pelo comando ativador do gatilho, executar novas consultas e fazer alterações no banco de dados. Uma ação pode até mesmo executar uma série de comandos de definição de dados, como por exemplo, criar novas tabelas.

#### 5.1.2.1 Projeto de banco de dados relacionais

De acordo com Maninno (2008), o processo de desenvolvimento de banco de dados consiste em produzir um banco de dados operacional para um sistema de informação, e é dividido em fases para a execução do mesmo. A princípio é definido três esquemas (externo, conceitual e interno), para após popular o banco de dados. Os itens externo e conceitual, estão relacionados com aspectos de informação do sistema, enquanto os itens interno e popular já referem-se à aspectos de implementação e eficiência desse banco.

#### **Modelagem de Dados Conceitual**

Primeiramente são identificados os requisitos do sistema, ou seja, identificar quais dados serão armazenados, quais as operações que serão mais frequentes, e quais as necessidades dos usuários no banco de dados. O resultado é expresso na modelagem de dados conceitual, que produz diagramas-entidade-relacionamento (DERs) para o esquema conceitual e para cada esquema externo (MANINNO, 2008). Os requisitos de dados podem ser formulários, relatórios, salvamento e edição de conteúdos. Sendo assim, o esquema conceitual deve representar todos os

requisitos do sistema, e o esquema externo deve representar apenas um requisito particular do banco de dados, como por exemplo a inscrição de um aprendiz em uma turma.

Esses dois esquemas seguem as regras do Modelo Entidade-Relacionamento, que é um tipo de representação gráfica dos elementos que compõem o sistema (entidades) e relacionamentos entre essas entidades. A entidade conterá atributos ou propriedades, e um destes atributos será o identificador único para cada instância dessa entidade, chamado de chave primária.

### **Projeto Lógico**

De acordo com Maninno (2008), essa fase utiliza o modelo conceitual para fazer um modelo mais refinado dos dados, em que possibilita a sua implementação em um SGBD. O refinamento relacionado ao modo como as informações serão distribuídas no banco, ocorre através de duas fases: conversão e normalização. A conversão irá transformar o DER em projetos de tabelas, que inclui colunas, chaves primárias, chaves estrangeiras, e outras propriedades. A normalização irá retirar as redundâncias usando dependências entre colunas.

### **Projeto Físico**

Para Maninno (2008), nessa fase duas coisas consideradas importantes são os índices e a disposição física dos dados. Um índice é um arquivo auxiliar, que serve para melhorar o desempenho nas recuperações de dados. Quanto a disposição física, refere-se a tomada de decisão em deixar os dados em apenas um disco ou mantê-los em cluster. Também nessa fase preocupa-se com o desempenho do banco de dados, utilizando métricas como tempo de resposta, e menor uso de recursos, como espaço em disco e memória principal.

### **Projeto de Visões**

O desenvolvimento de um banco de dados de tamanho moderado é um

processo intensivo que demanda tempo. Para administrar tal complexidade, divide-se o problema em problemas menores, permitindo que este seja resolvido independentemente, e essas soluções menores serão então combinadas para resolver o problema inteiro.

### **Projeto e integração de visões**

Neste projeto, uma visão é construída para trabalhar em uma parte específica do sistema, cobrindo apenas parte do banco de dados. Após é feito o processo de integração dessas visões em um esquema conceitual completo.

A seguir, será apresentada a modelagem conceitual do protótipo de versionamento proposto neste projeto.

## **5.2 MODELAGEM DO BANCO DE DADOS ATUAL DA FARMA**

Nesta seção será descrita uma parte do banco de dados atual da FARMA, representada através de um Diagrama de Classes utilizando a ferramenta case Astah, devido ao fato da natureza do banco de dados atual ser o MongoDB. Foram representadas nessa modelagem somente as classes que são relevantes para o versionamento dos contextos de interação.

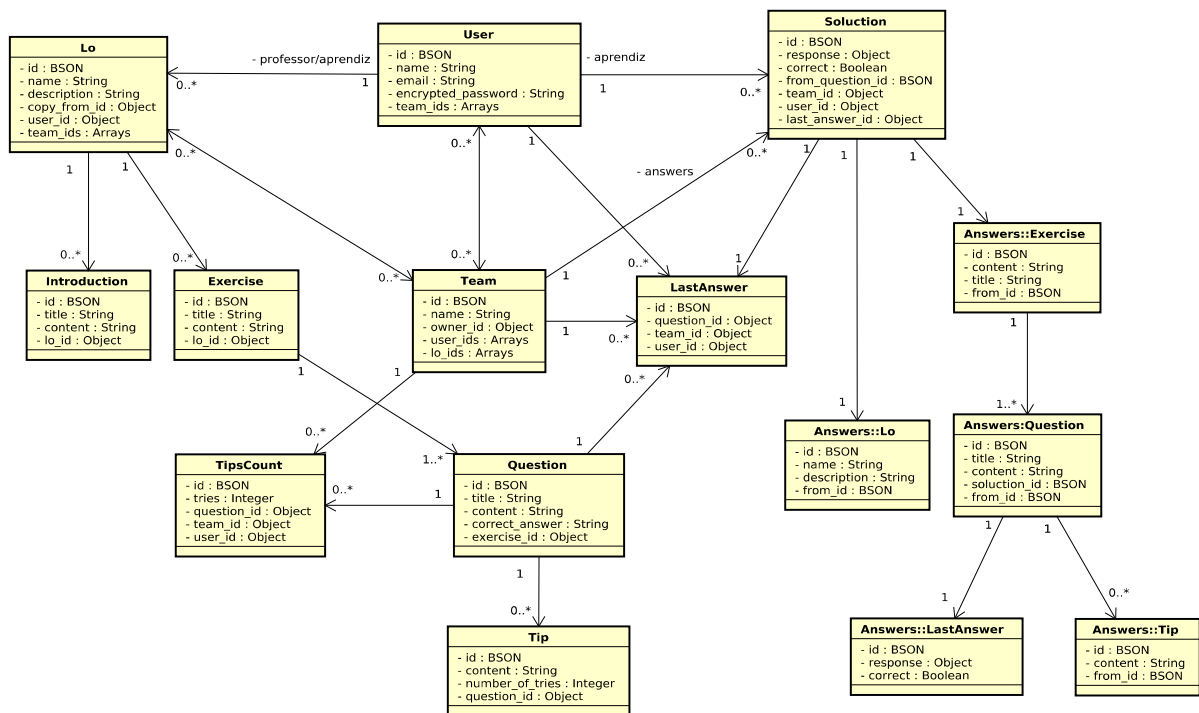


Figura 1: Diagrama de Classes da Ferramenta FARMA

Fonte: Autora

## Classe User

Esta classe contém as informações referentes ao cadastro de usuário, a qual está associada as classes responsáveis para realizar a criação de OAs e turmas, como também para interação com os objetos de aprendizagem de uma turma.

### Associação A1 – User e Lo

Representa o papel de professor-autor, quando cria objetos de aprendizagem, e também representa o papel de aprendiz quando ele interage com um OA.

### Associação A2 – User e Team

Representa o papel do professor-autor, quando cria turmas, e o papel do aprendiz quando se inscreve em uma turma. Porém, um usuário não pode entrar em uma turma que ele criou.

## Classe Lo

Representa um objeto de aprendizagem, que é criado por um professor-autor, que o atribui à uma turma para ser utilizado. Um objeto de aprendizagem é composto por

introduções, exercícios e questões.

### **Classe Introduction**

Toda introdução pertence à um objeto de aprendizagem. Em uma introdução contém a parte teórica do objeto de aprendizagem, geralmente conterà conceitos, fórmulas, imagens e vídeos.

### **Classe Exercise**

Um exercício compreende a parte prática do OA, conterà um título e um conteúdo, para apresentar o problema para o aprendiz. Um exercício poderá conter várias questões, a serem constituídas como parte da solução de um exercício. A ordem das questões é definida pelo professor-autor.

### **Classe Question**

Uma questão compreende a resolução parcial de um exercício, cada questão contém um título, enunciado e uma definição de resposta correta, para verificar quando uma resposta é inserida nela.

### **Classe Tip**

Representa uma dica que pertencerá a uma questão específica. A dica terá um conteúdo, que será exibido ao aprendiz após o número de tentativas limite for atingido. Cabe ao professor-autor definir quantas tentativas devem ser realizadas para exibir a dica.

### **Classe TipsCount**

Essa classe irá armazenar o número de tentativas que um aluno fez à uma questão.

### **Classe LastAnswer**

Essa classe representa a última resposta dada à uma questão, e sua associação com a turma e o usuário que a respondeu.

### **Classe Soluction**

Representa a resposta do aprendiz, seja ela correta ou incorreta. Uma solução está associada à uma última resposta.

### **Associação A8 – Soluction e LastAnswer**

Uma resposta está associada a uma última resposta, pois a cada vez que um aprendiz responde, esta passa a ser a última resposta daquela questão e turma.

### **Versionamento atual do contexto de uma resposta**

O versionamento está sendo feito da seguinte maneira: para cada resposta, é



armazenada uma cópia do OA, do exercício, contendo todas as questões e respostas, e também uma cópia da questão e da última resposta dessa questão.

A seguir, serão descritas as classes do versionamento de uma resposta:

#### **Classe Answers::Lo**

Esta classe mantém a cópia do OA, e está associada a uma resposta e ao OA original.

#### **Classe Answers::Exercise**

Essa classe mantém a cópia do exercício original, e está associada também a cópia do OA pertencente a resposta.

#### **Associação A9 - Solution e Answers::Lo**

Cada resposta conterá uma versão exclusiva do OA, e este OA terá a referência do original.

#### **Associação A10 - Solution e Answers::Exercise**

Cada resposta conterá uma versão do exercício, juntamente com todas as questões associadas à ele.

#### **Classe Answers::Question**

Essa classe representa a cópia da questão associada a resposta, e contém a referência para a questão original da resposta.

#### **Associação A11 – Answers::Exercise e Answers::Question**

A versão do exercício conterá todas as questões associadas a ele, que também serão cópias das questões originais.

#### **Classe Answers::LastAnswer**

Representa a cópia da última resposta dada pelo respectivo aprendiz para cada questão copiada.

#### **Classe Answers::Tip**

Esta classe mantém a cópia das dicas dadas para cada questão.

#### **Associação A12 – Answers::Question e Answers::LastAnswer**

Cada questão versionada conterá a cópia da última resposta que foi dada à ela pelo aprendiz.

#### **Associação A13 – Answers:: Question e Answers::Tip**

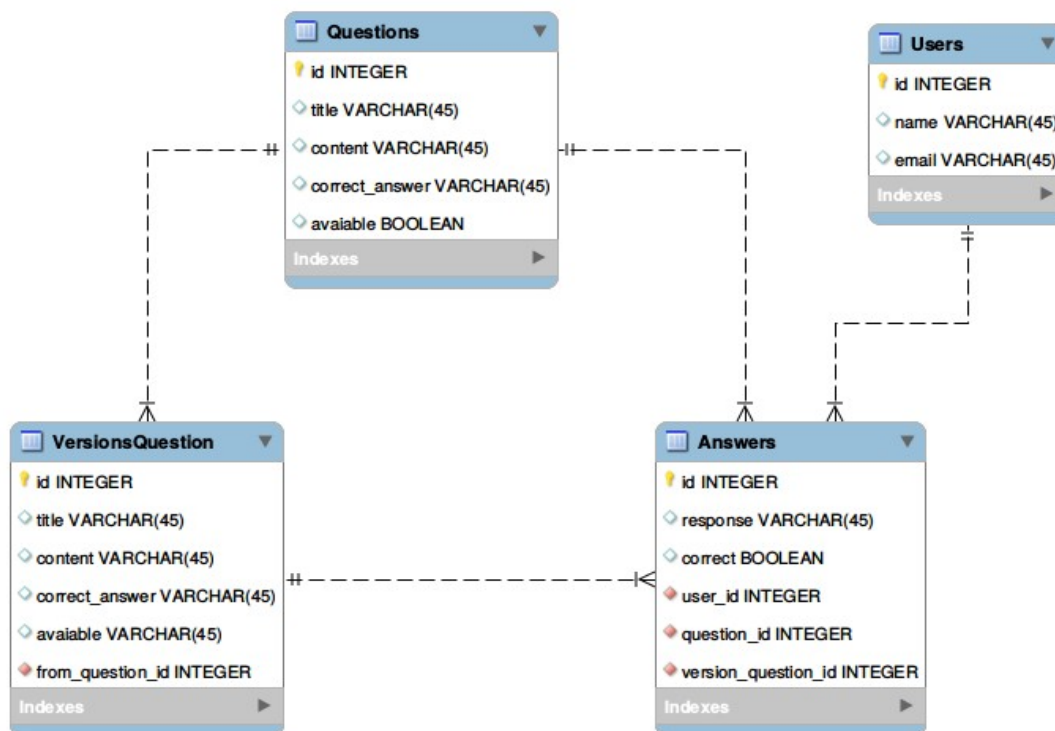
Também cada versão da questão conterá a cópia de todas as dicas que foram dadas à ela.

### 5.3 ARQUITETURA DA SOLUÇÃO PROPOSTA

Será apresentado o protótipo do mecanismo de versionamento dos contextos de interação do aprendiz com o OA, bem como os resultados parciais.

#### 5.3.1 Protótipo de versionamento

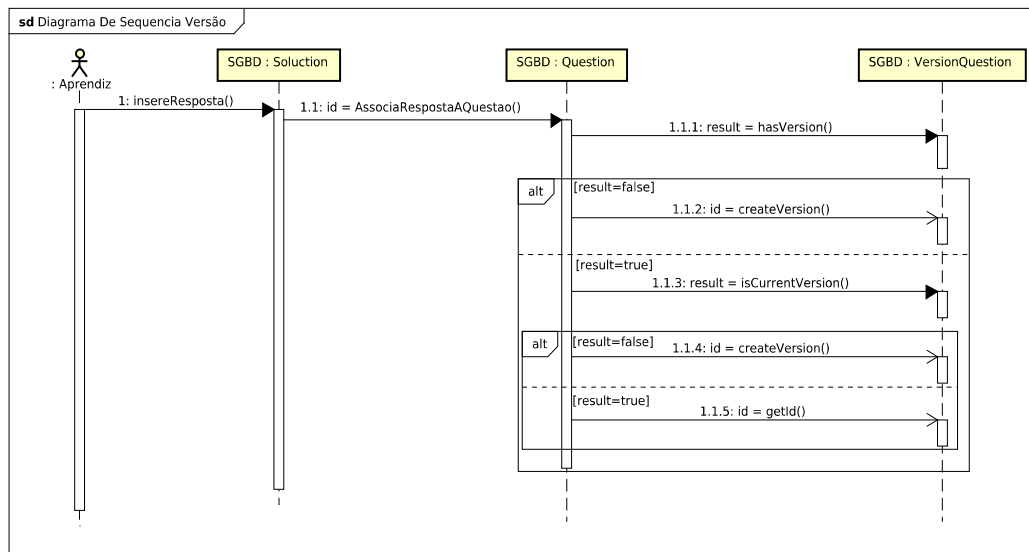
Nesta Seção é apresentado o protótipo do esquema de versionamento, considerando apenas a extremidade mais interna do contexto de respostas, que envolve a classes de resposta e questão. A importância deste protótipo é analisar a viabilidade do mecanismo de versão, bem como identificar necessidades e maiores detalhes a serem considerados para a expansão do versionamento para o restante do contexto das respostas. A seguir, na Figura 2, é representada a modelagem conceitual do protótipo de versionamento.



**Figura 2: Modelagem do protótipo do versionamento.**

Fonte: Autora.

Neste protótipo foi considerado apenas o essencial para simular o versionamento, ocultando outros campos existentes do banco de dados e outras tabelas. Foi criada a tabela VersionsQuestions que conterà as versões de uma questão, e adicionado o campo `version_question_id`, que associará a versão da questão à uma resposta no momento em que esta for inserida. A ação a ser executada no banco de dados quando uma resposta é inserida é representada através do Diagrama de Sequência ilustrado na Figura 3 seguida da sua descrição.



**Figura 3: Diagrama de Sequência do Protótipo**

Fonte: Autora

O funcionamento deste protótipo tem por base o uso de triggers. Sendo assim, o SGBD fará a seguinte verificação no momento que o aluno responder à uma questão: se existe ou não uma versão daquela questão. Se não existir, considera-se que é a primeira vez que ocorreu uma interação com esta questão, então cria-se uma versão inicial desta da questão e a atribui na resposta. Se já existir uma versão desta questão, verifica se é a versão mais recente, ou seja, se não contém diferenças da questão original. Caso não for a versão mais recente, então cria uma nova versão e a atribui na resposta, caso contrário, apenas atribui a versão mais recente na resposta.

## 6. RESULTADOS PARCIAIS

Neste Capítulo são retomadas as atividades que já foram desenvolvidas com base no cronograma, para apresentação dos resultados parciais.

**Estudo do banco de dados da ferramenta.** Nesta etapa inicial, foi analisada a regra de negócio do sistema, identificando as principais ações do professor-autor e aprendiz, como também o modo de armazenamento das respostas e principalmente como é feito atualmente o armazenamento dos dados de contexto.

**Descrição do banco de dados atual da ferramenta.** Nesta etapa, foram identificadas as associações entre as classes do sistema relacionadas com o contexto das respostas, que foram representadas através de um Diagrama de Classe expresso na Figura 1.

**Protótipo do versionamento dos contextos de respostas.** Nesta etapa foi criado um protótipo de versionamento considerando apenas as tabelas mais internas do contexto de interação, Answer e Question. Este protótipo será a base para a criação do mecanismo de versionamento de todo o contexto de interação.

## 7. CRONOGRAMA DE EXECUÇÃO

Na Figura 4 está apresentado o cronograma das atividades deste trabalho, alterando entre cores mais claras as atividades que já foram concluídas, e para as cores mais escuras as atividades pendentes.

Atividades	TCC 1						TCC 2			
	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1. Escrita da Proposta de TCC	█	█								
2. Defesa da Proposta de TCC		█								
3. Revisão dos apontamentos da banca		█								
4. Revisão bibliográfica		█	█							
5. Estudo do banco de dados atual da ferramenta			█	█						
6. Descrição do banco de dados atual da ferramenta			█	█						
7. Elencar pontos lentos da aplicação			█	█						
8. Redação do projeto de TCC			█	█						
9. Defesa do projeto de TCC				█						
10. Elencar alterações no banco de dados (novos dados)				█	█					
11. Modelar o novo banco de dados						█	█	█		
12. Migrar o banco de dados para a nova modelagem								█	█	
13. Testes e análise dos dados								█	█	
14. Redação da Monografia de TCC							█	█	█	
15. Elaboração da apresentação final									█	
16. Defesa final do TCC									█	

Figura 4: Cronograma do desenvolvimento do trabalho.

Fonte: Autora

## **8. CONSIDERAÇÕES FINAIS**

Este trabalho tem como objetivo propor uma forma de versionamento dos contextos de interação de um aprendiz com um OA na FARMA, a fim de diminuir a redundância nas cópias dos dados. A partir do banco de dados da ferramenta, foi possível identificar que a forma como os dados estão estruturados tem contribuído para que exista uma redundância excessiva, comprometendo as consultas no banco. Sendo assim, acredita-se que a migração do banco de dados para um SGBD relacional irá facilitar a criação do mecanismo de versão, e também melhorar a forma como os dados estão dispostos no banco para diminuir a redundância dos mesmos. O protótipo criado neste trabalho será posteriormente analisado e expandido para todo o contexto de interação do aprendiz com um OA, fornecendo assim uma solução para o problema da redundância dos dados.

## REFERÊNCIAS

CourseLab. Ferramenta de autoria CourseLab, 2012. <http://www.courselab.com>.

BLACKBOARD. **Hospedagem Gerenciada da Blackboard - Blackboard Brasil**. Disponível em <<http://blackboard.grupoa.com.br/servicos/hospedagem-gerenciada-da-blackboard/>>. Acesso em: 06 abr. 2016.

BRITO, R. W. **Bancos de Dados NoSQL x SGBDs Relacionais: Análise Comparativa**. Vila Mariana - São Paulo: [s.n.], 2012.

CHACON, Scott; STRAUB, Ben. **Pro Git**. 2. ed. Nova York: Apress, 2014.

DIANA, Mauricio de; GEROSA, Marco Aurélio. NOSQL na Web 2.0: Um Estudo Comparativo de Bancos Não-Relacionais para Armazenamento de Dados na Web 2.0. **Laboratório de Banco de Dados**. São Paulo, 2010. Disponível em: <[http://www.lbd.dcc.ufmg.br/colecoes/wtddb/2010/sbbd\\_wtd\\_12.pdf](http://www.lbd.dcc.ufmg.br/colecoes/wtddb/2010/sbbd_wtd_12.pdf)>. Acesso em: 02 set. 2015.

ENTERPRISEDB. **NoSQL for the Enterprise**. Disponível em <<http://www.enterprisedb.com/nosql-for-enterprise>>. Acesso em: 29 mar. 2016.

GAMA, Carmem Lúcia Graboski da. **Método de construção de objetos de aprendizagem com aplicação em métodos numéricos**. 2007. 210 f. Tese (Doutorado em Métodos Numéricos em Engenharia) – Universidade Federal do Paraná, Curitiba, 2007.

GIMENES, Gabriel Augusto. **Proposta para a Implementação do Cadastro de um Log de Auditoria Baseada em Padrões de Projeto**. 2013. 50 f. Trabalho de Conclusão de Curso (Bacharelado) - Faculdade Campo Limpo Paulista, Campo Limpo, 2013.

HOWS, David; MEMBREY, Peter; PLUGGE, Eelco. **Introdução ao MongoDB**. 1. ed. São Paulo: Novatec, 2015.

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. 6. ed. Porto Alegre: Bookman, 2009. Rio de Janeiro, 1993.



JUNIOR, Corneli Gomes F. et. al. Reuso e versionamento de conteúdos didáticos digitais em reuso interinstitucional compartilhado: Modelagem e validação por Redes de Petri Coloridas. In: **Anais do Simpósio Brasileiro de Informática na Educação**. 22., 2011, Aracaju. **Anais eletrônicos...** Belo Horizonte: Laboratório de Banco de Dados, 2011. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbie/2011/0096.pdf>>. Acesso em: 17 mar. 2016

KOSEKI, Michihiko; et. al. **Data processing system with mechanism for restoring file systems based on transaction logs**. US 09/501,568, 30 mar. 1999, 9 fev. 2000.

LIMA, Claudio de; MELLO, Ronaldo S. Um Estudo sobre Modelagem Lógica para Bancos de Dados NoSQL. **Modelagem Lógica para NoSQL**. Santa Catarina, v. 2.2, 2015.

LINSTER, Marc. **Postgres Outperforms MongoDB and Ushers in New Developer Reality**. Disponível em <<http://www.enterprisedb.com/postgres-plus-edb-blog/marc-linster/postgres-outperforms-mongodb-and-ushers-new-developer-reality>>. Acesso em: 29 mar. 2016.

MACDOWALL, R. D. Validation of Spectrometry Software-Audit Trails for Spectrometer Software. **Spectroscopy-Springfield then Eugene then Duluth**, v. 22, n. 4, p. 14-20, abr. 2007.

MARCZAL, Diego. **FARMA: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos**. 2014. 194 f. Tese (Doutorado) - Universidade Federal do Paraná, Curitiba, 2014.

MANNINO, Michael. **Projeto, Desenvolvimento de Aplicações e Administração de Banco de Dados**. 3. ed. São Paulo: AMGH, 2008.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de Banco de Dados**. 3. ed. São Paulo: McGraw-Hill, 2008.

RORATTO, Rodrigo; DIAS, Evandro Dotto. **Security information in production and operations: a study on audit trails in database systems**. In: JISTEM - Journal of Information Systems and Technology Management, vol. 11, n. 3, São Paulo, dez. 2014. Disponível em: <<http://www.jistem.fea.usp.br/index.php/jistem/article/view/2445>>. Acesso em: 29 abr. 2016.

SILVA, Rômulo César. **Sequenciamento Adaptativo de Exercícios Baseado na Correspondência entre a Dificuldade da Solução e o Desempenho Dinâmico do Aprendiz.** 2015. 82 f. Tese (Doutorado) – Universidade Federal do Paraná, Curitiba, 2015.

SIMON, Fernando; SANTOS, Aldri L. dos; HARA, Carmem S. Um Sistema de Auditoria baseado na Análise de Registros de Log. **Escola Regional de Banco de Dados (ERBD'2008)**, 2008.

SOARES, Luiz Fernando G.; CASANOVA, Marco Antonio.; RODRIGUEZ, Noemi L.R. **Um Modelo Conceitual Hipermídia com Nós de Composição e Controle de Versões.** Disponível em <[https://www.researchgate.net/profile/Luiz\\_Fernando\\_Soares/publication/228804559\\_Um\\_Modelo\\_Conceitual\\_Hipermdia\\_com\\_Ns\\_de\\_Composio\\_e\\_Control\\_e\\_de\\_Versoes/links/02e7e53adaf8902554000000.pdf](https://www.researchgate.net/profile/Luiz_Fernando_Soares/publication/228804559_Um_Modelo_Conceitual_Hipermdia_com_Ns_de_Composio_e_Control_e_de_Versoes/links/02e7e53adaf8902554000000.pdf)>. Acesso em: 12 jan. 2016.

SPINELLIS, Diomidis. *Git*. **IEEE SOFTWARE - Tools of the Trade.** Los Alamitos, v. 29, n. 3, p.100-101, mai. 2012.

PAZINATTO, Eder. Auditoria em Banco de Dados com Utilização de Regras. **Revista de Ciências Exatas e Tecnologia.** Valinhos, v. 5, n. 5, p. 91-100, 2010.