

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

LETICIA CORREIA

**CONTROLE DE VERSÃO PARA ARMAZENAMENTO DO
CONTEXTO DA INTERAÇÃO DO APRENDIZ COM OBJETOS
DE APRENDIZAGEM**

TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA
2017

LETICIA CORREIA

**CONTROLE DE VERSÃO PARA ARMAZENAMENTO DO
CONTEXTO DA INTERAÇÃO DO APRENDIZ COM OBJETOS
DE APRENDIZAGEM**

Monografia de Trabalho de Conclusão de Curso de graduação,
apresentado a disciplina de Trabalho de Conclusão de Curso 2
do Curso Superior de Tecnologia em Sistemas para a Internet -
TSI da Universidade Tecnológica Federal do Paraná - UTFPR
- Câmpus Guarapuava, como requisito parcial para obtenção
do título de Tecnólogo em Sistemas para a Internet.

Orientador: Prof. Dr. Diego Marczal
UTFPR Câmpus Guarapuava

GUARAPUAVA
2017

ATA DE DEFESA DE MONOGRAFIA DE TRABALHO DE CONCLUSÃO DE CURSO DO CURSO DE TSI

No dia 21 de junho de 2017, às 15:00 horas, nas dependências da Universidade Tecnológica Federal do Paraná Câmpus Guarapuava, ocorreu a banca de **defesa da monografia** de Trabalho de Conclusão de Curso intitulada: **“Controle de Versão para Armazenamento do Contexto da Interação do Aprendiz com Objetos de Aprendizagem”** da acadêmica **Leticia Correia** sob orientação do professor **Prof. Dr. Diego Marczal** do Curso de Tecnologia em Sistemas para Internet.

Banca Avaliadora	
Membro	Nome
Orientador	Prof. Dr. Diego Marczal
Coorientador	
Avaliador 1	Prof. Dr. Paulo Henrique Soares
Avaliador 2	Profa. Me. Carolina Moreira

Situação do Trabalho	
Situação	<input checked="" type="checkbox"/> Aprovado <input type="checkbox"/> Aprovado com ressalvas <input type="checkbox"/> Reprovado <input type="checkbox"/> Não Compareceu
Encaminhamento do trabalho para biblioteca	<input checked="" type="checkbox"/> Pode ser encaminhado para biblioteca. <input type="checkbox"/> Manter sigilo para publicação ou geração de patente.

Prof. Dr. Diego Marczal

Prof. Dr. Paulo Henrique Soares

Profa. Me. Carolina Moreira

Acadêmico: Leticia Correia

Diego Marczal
Paulo H. Soares
Carolina M. Oliveira
Leticia Correia

Guarapuava, 21 de junho de 2017.

AGRADECIMENTOS

Primeiramente agradeço à Deus por toda a força e graças que tem me dado.

Aos meus pais, que me entenderam tantas vezes que tive que abrir mão de estar com eles para estudar, e também por me apoiarem a não desistir.

Ao meu orientador Diego Marczal, por toda a disposição, compromisso, atenção e por aceitar orientar este trabalho.

Aos amigos e colegas deste curso, que me proporcionaram inúmeras lembranças dessa caminhada, por tantas vezes que passamos por trabalhos, estudos, risadas, preocupações com prazo e objetivos a serem cumpridos, aqui vai os meus sinceros agradecimentos pela convivência que tivemos.

A todos os professores do curso de Tecnologia em Sistemas para Internet, que me deram tantos conhecimentos e vivências.

Vive-se apenas uma vez; ou seja: a vida de cada pessoa é única; todo ser humano é singular. Por isso não podemos perder tal oportunidade; não devemos simplesmente vegetar, mas, com olhos abertos, "percorrer o mundo", imprimindo conscientemente nele a marca de nossa vida. (GRUN, Anselm, 2011).

RESUMO

CORREIA, Leticia. Controle de Versão para Armazenamento do Contexto da Interação do Aprendiz com Objetos de Aprendizagem. 2017. 40 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2017.

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 de 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. Deste modo, este trabalho apresenta uma nova modelagem do banco de dados da FARMA, que fornece um controle de versão dos objetos de aprendizagem. Essa modelagem armazena os contextos de interação do aprendiz com um objeto de aprendizagem, e de acordo com esse contexto, garante os dados corretos para a geração das estatísticas de desempenho do aprendiz dentro de um objeto de aprendizagem e até mesmo no âmbito de uma turma. Sendo assim, este trabalho tem o diferencial de obter uma modelagem de versionamento de baixa complexidade e pouca redundância de dados.

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

ABSTRACT

CORREIA, Leticia. Version Control for Context Storage of Apprentice Interaction with Learning Objects. 2017. 40 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2017.

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. Thus, this work presents a new modeling of the FARMA database, which provides a version control of learning objects. This modeling stores the contexts of learner interaction with a learning object, and according to this context, ensures the correct data for the generation of learner performance statistics within a learning object and even within a class. Therefore, this work has the differential of obtaining a versioning model of low complexity and little data redundancy.

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

LISTA DE FIGURAS

Figura 1 – Tela inicial da FARMA	6
Figura 2 – Linha do Tempo de um Aprendiz	7
Figura 3 – Aprendiz Realizando a Retroação na FARMA	7
Figura 4 – Diagrama de Classes da Ferramenta FARMA	17
Figura 5 – Modelagem da Arquitetura Inicial.	22
Figura 6 – Fluxograma da Solução Inicial	24
Figura 7 – Retroação na Solução Inicial.	26
Figura 8 – Modelagem da Arquitetura Final	29
Figura 9 – Fluxograma da Solução Final	30
Figura 10 – Retroação na Solução Final	31
Figura 11 – Modelagem da Parte de Autoria da FARMA	38
Figura 12 – Modelagem das Estatísticas do Aprendiz	39
Figura 13 – Modelagem Completa da Solução Final	40

LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicity, Consistency, Isolation, and Durability
CVS	Concurrent Versions System
DDL	Data Definition Language
DER	Diagrama Entidade Relacionamento
DML	Data Manipulation Language
FARMA	Ferramenta de Autoria para a Remediação de erros com Mobilidade na Aprendizagem
IBM	International Business Machines
JSON	JavaScript Object Notation
MER	Modelo Entidade Relacionamento
NoSQL	Not only SQL
OA	Objeto de Aprendizagem
RI	Restrições de Integridade
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structured Query Language
STI	Sistemas Tutores Inteligentes

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 OBJETIVOS	3
1.1.1 Objetivo Geral	3
1.1.2 Objetivos Específicos	3
1.2 ORGANIZAÇÃO DO TRABALHO	3
2 – FUNDAMENTAÇÃO TEÓRICA	5
2.1 FERRAMENTAS DE AUTORIA	5
2.2 O ERRO NO ENSINO E APRENDIZAGEM	8
2.3 REDUNDÂNCIA DE DADOS	8
2.4 VERSIONAMENTO DE DADOS	9
2.5 TECNOLOGIAS ENVOLVIDAS	10
2.5.1 Banco de Dados Não Relacionais	10
2.5.2 Banco de Dados Relacionais	11
2.5.2.1 Gatilhos	12
2.5.2.2 Projeto de Banco de Dados Relacionais	12
3 – PROCEDIMENTOS METODOLÓGICOS	14
4 – DESENVOLVIMENTO	16
4.1 MODELAGEM DO BANCO DE DADOS ATUAL DA FARMA	16
4.2 ANÁLISE DA MODELAGEM ATUAL DO BANCO DE DADOS DA FARMA	19
4.3 ARQUITETURA DA SOLUÇÃO PROPOSTA	20
4.3.1 Arquitetura Inicial	21
4.3.1.1 Retroação ao contexto de interação	25
4.3.2 Arquitetura Final	27
4.3.2.1 Retroação ao contexto de interação	30
5 – RESULTADOS	32
6 – CONSIDERAÇÕES FINAIS	33
6.1 TRABALHOS FUTUROS	34
Referências	35

Apêndices	37
APÊNDICE A–MODELAGEM DA PARTE DE AUTORIA DA FARMA . . .	38
APÊNDICE B–MODELAGEM DAS ESTATÍSTICAS DO APRENDIZ . . .	39
APÊNDICE C–MODELAGEM COMPLETA DA SOLUÇÃO FINAL	40

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 PROINFO¹, Tablet Educacional² e EDUCA³. Da mesma forma, como softwares específicos para disponibilização de conteúdo educacional ao aprendiz pode-se citar o Learning System Manager (LMS) Moodle⁴, “um dos mais utilizados em ambientes de ensino” (MARCZAL, 2014, p. 117), e o Blackboard⁵. Também existem repositórios de softwares educacionais como o Portal do Professor⁶, Banco Internacional de Objetos de Aprendizagem⁷, dentre outras iniciativas como o Khan Academy⁸.

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 (2014), 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 a Remediação de erros com Mobilidade na Aprendizagem (FARMA), é uma ferramenta que permite a criação de OAs para o ensino de matemática. De acordo com Marczal (2014), esta tem o diferencial de facilitar a criação de OAs dinâmicos, mesmo por pessoas leigas em computação, e também de realizar a retroação⁹ aos erros do aprendiz, em que o professor-autor poderá identificar os equívocos do aprendiz e propor soluções para auxiliá-lo na interpretação do conteúdo.

Para viabilizar a retroação aos contextos de erros e acertos, a FARMA armazenava uma cópia do contexto atual do OA para cada resposta do aprendiz. Isso permitia que quando o professor ou o aprendiz acessasse uma resposta, tivesse exatamente a mesma visão de quando

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

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

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

⁴Disponível em: <https://moodle.org/>

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

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

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

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

⁹De 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.

o aprendiz a respondeu, ou seja, cada resposta mantinha todo o contexto mesmo que alguma parte do OA fosse alterada pelo professor-autor. Devido à isso a FARMA possui uma grande redundância de dados. Sabe-se que a quantidade de dados, em um banco de dados, 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.

A FARMA utilizava o banco de dados não relacional MongoDB. Este tipo de banco tem uma maior flexibilidade no armazenamento dos dados e consequentemente 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](#)). Outro ponto é que a redundância de dados apresentada no salvamento do contexto da interação, acontecia pela maneira com que os dados estavam estruturados na modelagem, 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.

Realizou-se então uma pesquisa para descobrir qual seria o Sistema Gerenciador de Banco de Dados (SGBD) que melhor se encaixaria para o domínio da FARMA, com enfoque no armazenamento dos dados de contexto de interação. Sendo assim, o SGBD escolhido foi o PostgreSQL, pelas seguintes características: (a) flexibilidade dos dados, através do suporte a campos JSON na versão PostgreSQL 9.4; (b) suporte a sistemas de alto rendimento e alto processamento. Essas características são relevantes para a FARMA, pois o PostgreSQL é capaz de manipular grandes quantidades de dados com vários usuários simultâneos ([ENTERPRISEDB, 2016](#)), o que significa que haverá velocidade e flexibilidade nas consultas sem perder as normas ACID¹⁰ que são recursos relevantes para aplicações de missão crítica.

Atualmente a FARMA está em processo de atualização, e um dos fatores que estão sendo analisados é 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.

Nesse sentido, este projeto buscou ampliar as informações salvas do aprendiz, que eram somente da sua interação com o OA durante a resolução de um exercício. As informações adicionadas são da pontuação do aprendiz, para definir o seu grau de desempenho referente a resolução de um passo, de um exercício, ou até mesmo do OA inteiro. O uso dessas informações serão utilizadas futuramente para identificar e apresentar ao professor-autor qual o nível de desempenho do aprendiz, como também quais os conteúdos em que o aprendiz teve maior facilidade e dificuldade, e até quais conteúdos são mais difíceis no âmbito de uma turma.

Assim, a contribuição deste trabalho foi criar a modelagem do banco de dados que provê a retroação ao contexto das respostas do aprendiz, tendo um mecanismo de versionamento dos contextos do OA que gera poucos dados redundantes. E, adicionalmente, salvar as informações

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

relevantes do aprendiz sobre toda a sua interação com OA, para que possam ser utilizadas em análises estatísticas e pesquisas futuras.

Considerou-se como desafio desse trabalho, a atribuição dessas informações, assim como o versionamento de um OA, pela complexidade de criar uma solução que realizasse o acesso e o armazenamento dos contextos de maneira eficiente, sem grandes perdas de desempenho no banco, e consequente rapidez de acesso na aplicação. No que se trata do desempenho, mesmo que não tenha sido comprovado por meio de um *Benchmark*¹¹, a otimização no armazenamento dessas informações é visual.

1.1 OBJETIVOS

Nesta seção serão apresentados os objetivos deste trabalho.

1.1.1 Objetivo Geral

Criar a modelagem de 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

- Fazer a descrição da modelagem atual do banco da FARMA, a partir do banco de dados físico;
- Criar uma 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;
- Elaborar a modelagem do banco de dados, que seja capaz de armazenar os contextos da interação dos aprendizes com os OAs nas turmas.

1.2 ORGANIZAÇÃO DO TRABALHO

Esta seção apresenta a organização do restante do conteúdo da monografia. O Capítulo 2 corresponde a Fundamentação Teórica, dividida em: seção 2.1, que apresenta as ferramentas de autoria, seção 2.2, que trata sobre o erro no ensino e aprendizagem, seção 2.3, que apresenta sobre a redundância de dados, seção 2.4, que apresenta sobre versionamento de dados, e seção 2.5, que apresenta as tecnologias envolvidas. O Capítulo 3 apresenta os procedimentos metodológicos deste trabalho. O Capítulo 4 apresenta todo o desenvolvimento da pesquisa.

¹¹Em computação, benchmark é o ato de executar um programa de computador, um conjunto de programas ou outras operações, a fim de avaliar o desempenho relativo de um objeto, normalmente executando uma série de testes padrões e ensaios nele.

O Capítulo 5 e 6, respectivamente, descrevem os resultados e as considerações finais deste projeto.

2 FUNDAMENTAÇÃO TEÓRICA

Apresenta-se, neste capítulo, o funcionamento de algumas ferramentas de autoria para o ensino e aprendizagem em contraste com a descrição das principais funcionalidades da FARMA, também apresenta-se sobre o erro no processo de ensino e aprendizagem, e após as formas de redundância de dados, e algumas características sobre versionamento de dados. Descreve-se também neste capítulo as tecnologias que foram estudadas para desenvolver este trabalho.

2.1 FERRAMENTAS DE AUTORIA

A CourseLab¹ 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 exeLearning² 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 MyUdutu³ 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 têm pouca interatividade com o aprendiz, e os ambientes citados também não tratam os erros cometidos pelo aprendiz.

A FARMA tem o diferencial de facilitar a criação de OAs mesmo por pessoas leigas em computação, e possibilitar que seja feita a retroação, através do armazenamento dos erros e acertos do aprendiz e todo seu contexto.

A retroação na FARMA é a recuperação de todo o contexto de uma resposta dada por um aprendiz, incluindo todos os exercícios, passos, dicas que já foram dadas, e também as respostas inseridas por esse aprendiz. A retroação na FARMA tem o objetivo de proporcionar ao professor-autor a capacidade de realizar a análise mais detalhada da interação do aprendiz no OA, e poder lidar de forma apropriada com a causa do erro.

Na Figura 1 mostra-se a tela inicial da FARMA.

¹<http://www.courselab.com/>.

²<http://exelearning.org/>.

³<http://www.myudutu.com/>.

Figura 1 – Tela inicial da FARMA



Fonte: <http://farma.educacional.mat.br/>

O funcionamento da FARMA se divide em três módulos principais, o módulo de autoria, de interação e de monitoramento.

O módulo de autoria permite a construção de um OA pelo professor-autor, definindo as introduções, exercícios e passos⁴. Este é também responsável pela definição das regras de remediação e mensagens por parte do professor-autor, que são apresentadas quando um aprendiz comete erro. O módulo de interação corresponde a interface entre o aprendiz e o OA, onde as introduções, exercícios, passos e remediações são organizadas para compor o objeto de aprendizagem. E o módulo de monitoramento abrange a retroação a erros, onde tanto o professor-autor quanto o aprendiz podem voltar ao contexto exato da resolução de um passo, para obter uma análise detalhada do erro.

Na FARMA, existem várias formas de realizar a retroação a uma questão, e uma delas é pela linha do tempo, conforme mostrada na Figura 2, e logo após, na Figura 3, é mostrada a retroação feita através da linha do tempo.

⁴Na FARMA consideramos que cada exercício é dividido em várias etapas de solução, que anteriormente tinha a nomenclatura de questão, mas foi modificada para passo de uma solução.

Figura 2 – Linha do Tempo de um Aprendiz

FARMA HOME Olá aluno-farma

Ferramenta de Autoria para a Remediação de erros com Mobilidade na Aprendizagem.
Construa Objetos de Aprendizagem para o ensino de matemática

Home / Meus Progresso / Timeline

Juros Simples

Retroceder a resposta

Suponha-se que você tenha um investimento inicial de R\$ 3.000, com uma taxa de juros de 60% em um período de 5 anos.

Passo 1

Qual o valor futuro do montante?

Resposta: Incorreto
5.000

0 comentários recebidos

Juros Simples - Passo 1
Juros Simples - Passo 2
Juros Simples - Passo 1

3:44 PM PASSO 1 3:46 PM PASSO 2

Fonte: <http://farma.educacional.mat.br/>

Figura 3 – Aprendiz Realizando a Retroação na FARMA

Retroação a resposta do aprendiz aluno-farma

Turma	OA	Exercício	Questão	Resposta	Tentativa	Ocorrida em
Algebra Financeira	Cálculo de Juros	Juros Simples	Passo 1	5.000	1	16/05/2017 15:43:51

Esta resposta possui 0 comentários

Juros Simples

Suponha-se que você tenha um investimento inicial de R\$ 3.000, com uma taxa de juros de 60% em um período de 5 anos.

Passo 1

Qual o valor futuro do montante?

Resposta: Incorreto
5.000

Passo 2

Quais são os juros totais acumulados para este montante?

Resposta:
Clique aqui para responder

Fonte: <http://farma.educacional.mat.br/>

O sistema apresentava uma lentidão no acesso aos dados dos contextos de interação, que poderia estar associada a forma como era armazenada a resposta do aprendiz e todo o seu contexto. Pois quando uma resposta era enviada, criava-se uma cópia de todo o contexto, incluindo o passo referente a resposta, as dicas para aquela resposta, o número de tentativas, o exercício a qual pertence o passo, todos os outros passos e exercícios do OA, e as suas respectivas respostas e remediações. Isso era feito para que as modificações futuras não interferissem no contexto da resposta já salva.

O problema é que esse procedimento era feito em cada resposta inserida, fosse ela certa ou errada, o que resultava em uma grande quantidade de dados redundantes. Como isso está relacionado com a modelagem do banco, acredita-se que este projeto foi relevante para otimizar o registro e o acesso aos contextos de interação da FARMA, e minimizar esta redundância através da remodelagem do banco de dados.

2.2 O ERRO NO ENSINO E APRENDIZAGEM

No ponto de vista pedagógico, o erro é considerado elemento de grande importância no processo de ensino aprendizagem. De acordo com [Silva \(2008\)](#) o erro revela a inadequação da resolução de um aluno, e mostra a necessidade da reformulação da estratégia previamente criada para resolver o problema proposto. Dessa forma, o erro possibilita a revisão e avanço, levando alunos e professores a serem os sujeitos dos seus processos de reconstrução do conhecimento.

O desvendar do erro consiste em levar os envolvidos em um processo de avaliação, de questionamento, o qual irá transformar o conhecimento, que anteriormente era restrito, para um significado construtivo. De acordo com [Silva \(2015\)](#), existem vários Sistemas Tutores Inteligentes (STI) que procuram formas de detectar e apresentar os erros dos aprendizes, e também utilizam exemplos errôneos, no qual o aprendiz terá que identificar quais passos estão incorretos, explicar a causa do erro, e corrigi-los, para que se tenha um conhecimento mais aprofundado sobre o conteúdo. Dessa forma, é importante destacar que, de acordo com [Marczal \(2014\)](#), o erro é visto dentro do contexto da FARMA como abordagem cognitivista central no processo de aprendizagem, visto que o armazenamento, a recuperação e a visualização dos erros têm grandes benefícios para o ensino e aprendizagem.

2.3 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 têm 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 diferentes formas no sistema, gerando inconsistência no banco de dados.

No entanto, a redundância na FARMA era diferente das anteriores, pois estava relacionada à modelagem do banco de dados. A redundância era gerada através do armazenamento das respostas, pois para viabilizar a retroação, a FARMA realizava uma espécie de versionamento de todo o contexto da resposta do aprendiz, fazendo uma cópia do exercício, do passo, do número de tentativas, dicas recebidas, etc., isso a cada vez que o aprendiz respondesse à um passo. Essa redundância afetava o desempenho das consultas no banco, pois gerava uma grande quantidade de dados e causava lentidão no acesso aos contextos de interação.

Dessa forma identificou-se a necessidade da remodelagem do banco de dados para a criação de um mecanismo de controle de versões a fim de minimizar essa redundância.

2.4 VERSIONAMENTO DE DADOS

Muitos desenvolvedores utilizam sistemas de controle de versão para gerenciar as versões dos seus arquivos. Tais sistemas contêm uma série recursos, tais como: reverter desde um arquivo até um projeto para um estado anterior, e comparar mudanças que foram feitas entre um intervalo de tempo, e ver quem foi o último a modificar algo que pode estar causando problemas (CHACON; STRAUB, 2014).

Assim, de acordo com Chacon e Straub (2014), controle de versão é uma operação do sistema que registra as alterações feitas em um arquivo ou conjunto de arquivos ao longo do tempo, para que se possa recuperar versões específicas posteriormente.

O conceito de versão é visto como relevante para gerenciar as diferentes representações de uma informação. De acordo com Chacon e Straub (2014), um mecanismo de versão tem as seguintes características:

1. Reverte os arquivos para um estado anterior, bem como reverter todo o projeto de volta a um estado anterior;
2. Compara as mudanças ao longo do tempo, e ver quem modificou pela última vez;
3. Recupera os dados, se acontecer algum problema, e houver perda de arquivos;
4. Todas as operações realizadas requerem pouca sobrecarga.

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)⁵, Subversion⁶ e o GIT⁷, 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 as mudanças que foram feitas e qual usuário que as fez. O GitHub⁸, 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, foi desenvolvido um mecanismo de controle de versão dos objetos de aprendizagem, com o objetivo de minimizar a redundância gerada pelo salvamento dos dados do contexto de interação da FARMA.

2.5 TECNOLOGIAS ENVOLVIDAS

Nesta seção serão apresentados os conceitos das principais tecnologias que foram utilizadas para a realização deste trabalho. Será abordado o Modelo Relacional e o Modelo Não Relacional, especificando o MongoDB e PostgreSQL, SGBDs dos modelos, respectivamente.

2.5.1 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 (LINSTER, 2014). 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.

⁵<http://www.nongnu.org/cvs/>

⁶<https://subversion.apache.org/>

⁷<https://git-scm.com/>

⁸<https://github.com/>

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.

[Howe, 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.

2.5.2 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; GEHRKE, 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:

- 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.
- Linguagem de manipulação de dados - Data Manipulation Language (DML), usada para formular consultas e inserir, excluir e consultar os registros do banco de dados.
- 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; GEHRKE, 2008](#)).

2.5.2.1 Gatilhos

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:

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

A utilização das triggers foi um recurso muito favorável para FARMA, pois o gerenciamento dos contextos ficou à cargo do banco de dados, que utilizou apenas uma conexão para fazer o armazenamento do contexto da resposta, o que seria muito mais pesado se ficasse a cargo da aplicação, pois demandaria o uso de várias conexões para realizar o armazenamento do contexto. Dessa forma, toda a lógica para o salvamento e recuperação dos contextos foi feita no banco de dados, deixando a aplicação mais leve, e sendo responsável apenas pela renderização dos contextos.

2.5.2.2 Projeto de Banco de Dados Relacionais

De acordo com [Mannino \(2008\)](#), 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 (MANNINO, 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 Mannino (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 Mannino (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.

3 PROCEDIMENTOS METODOLÓGICOS

Este capítulo apresenta de forma sequencial os passos que foram desenvolvidos neste trabalho, e uma descrição de como foi realizado cada passo.

Estudo da ferramenta de autoria FARMA. Para entendimento do objetivo e das principais funcionalidades da FARMA, foi realizado um estudo teórico por meio de textos publicados sobre a mesma. Além disso, foram realizados testes na própria ferramenta para conhecimento do seu funcionamento e identificação de pontos lentos. Com isso, foi identificado que o sistema apresenta lentidão, no momento em que é inserida uma resposta, devido a forma com que os dados estão estruturados no banco de dados.

Criação da modelagem do banco de dados atual. Este banco de dados não possuía uma modelagem, para tanto, foi realizada uma modelagem da base de dados atual a partir do banco de dados físico. Primeiramente esta modelagem foi feita automaticamente, através de uma *gem* do Rails, e depois foi traduzida para um diagrama de classes utilizando a ferramenta Astah¹. Foi utilizado o Astah como ferramenta case de apoio, para que a modelagem pudesse representar com maior fidelidade a estrutura do banco de dados atual, que era um tipo de banco NoSQL.

Análise do banco de dados atual. A partir da modelagem realizada no item anterior, foi realizada a análise do banco a fim de identificar as principais ligações e em que parte do banco estavam os pontos lentos. Após a análise, identificou-se que ao salvar uma resposta, salva-se uma cópia do OA completo para aquela única resposta, o que demanda processamento e memória, que afetam a velocidade dos registros e das consultas dos contextos do OA no banco de dados.

Estudo sobre banco de dados PostgreSQL. Foi feita uma pesquisa sobre o PostgreSQL, onde foram identificadas suas principais vantagens e características favoráveis para as especificações da FARMA, dentre elas a capacidade de conseguir manipular uma grande quantidade de dados, e suporte para campos JSON² e *Array*. Desta forma, optou-se por utilizar o SGBD PostgreSQL, que possui atributos do modelo relacional, assim como características voltadas ao modelo híbrido.

¹Astah Community é um software gratuito para modelagem UML. Mais informações disponíveis em: <http://astah.net/>.

²JSON (JavaScript Object Notation - Notação de Objetos JavaScript) é uma formatação leve de troca de dados. Fonte: <http://www.json.org/json-pt.html>

Pesquisa sobre versionamento dados. Foi pesquisado como é feito o versionamento de dados, e também como é a estrutura do banco para armazenar os dados versionados, afim de ter o entendimento do funcionamento de um sistema de versão. A partir desta pesquisa, pode-se perceber a importância do versionamento de arquivos nos sistemas atuais, dando uma maior relevância para o tema deste trabalho. Nesta etapa identificou-se que nenhum dos exemplos de sistema de versionamento de dados encontrados se encaixavam nas particularidades da FARMA, e que foi necessário criar um sistema de versionamento exclusivo para a FARMA.

Estudo das tecnologias utilizadas. Foram pesquisadas e estudadas algumas tecnologias para serem utilizadas no projeto. Dentre elas, destaca-se a ferramenta pgTAP³, usada para realizar os testes nas triggers no banco de dados, a ferramenta Astah, para realizar a modelagem do banco atual, a ferramenta MySQL Workbench⁴, para criar a nova modelagem do banco de dados, o SGBD PostgreSQL, o qual teve foco de pesquisa principalmente na definição das triggers e funções utilizadas para o salvamento dos contextos de interação.

Modelagem do novo banco de dados da FARMA. A partir dos dados da análise, esta etapa consistiu em elaborar o modelo lógico, definindo as tabelas necessárias e os seus devidos relacionamentos. Esse modelo passou por várias modificações, a fim de realizar o salvamento dos dados necessários de forma mais simples, e que pudesse atender aos requisitos da ferramenta FARMA sem uma grande complexidade nas consultas do banco de dados. Essa modelagem será detalhada na Seção 4.3.

³pgTAP é um software para realizar testes em psql no banco de dados. Mais informações em: <http://pgtap.org/>.

⁴Software utilizado para elaborar modelagens do banco de dados. Mais informações: <https://www.mysql.com/products/workbench/>

4 DESENVOLVIMENTO

Neste capítulo apresenta-se como ocorreu todo o processo de desenvolvimento deste trabalho, iniciando pela análise do banco de dados da FARMA, após a identificação da redundância de dados e pontos lentos, o processo de criação e implementação da solução proposta, e por último, quais as modificações realizadas nela até chegar na solução final.

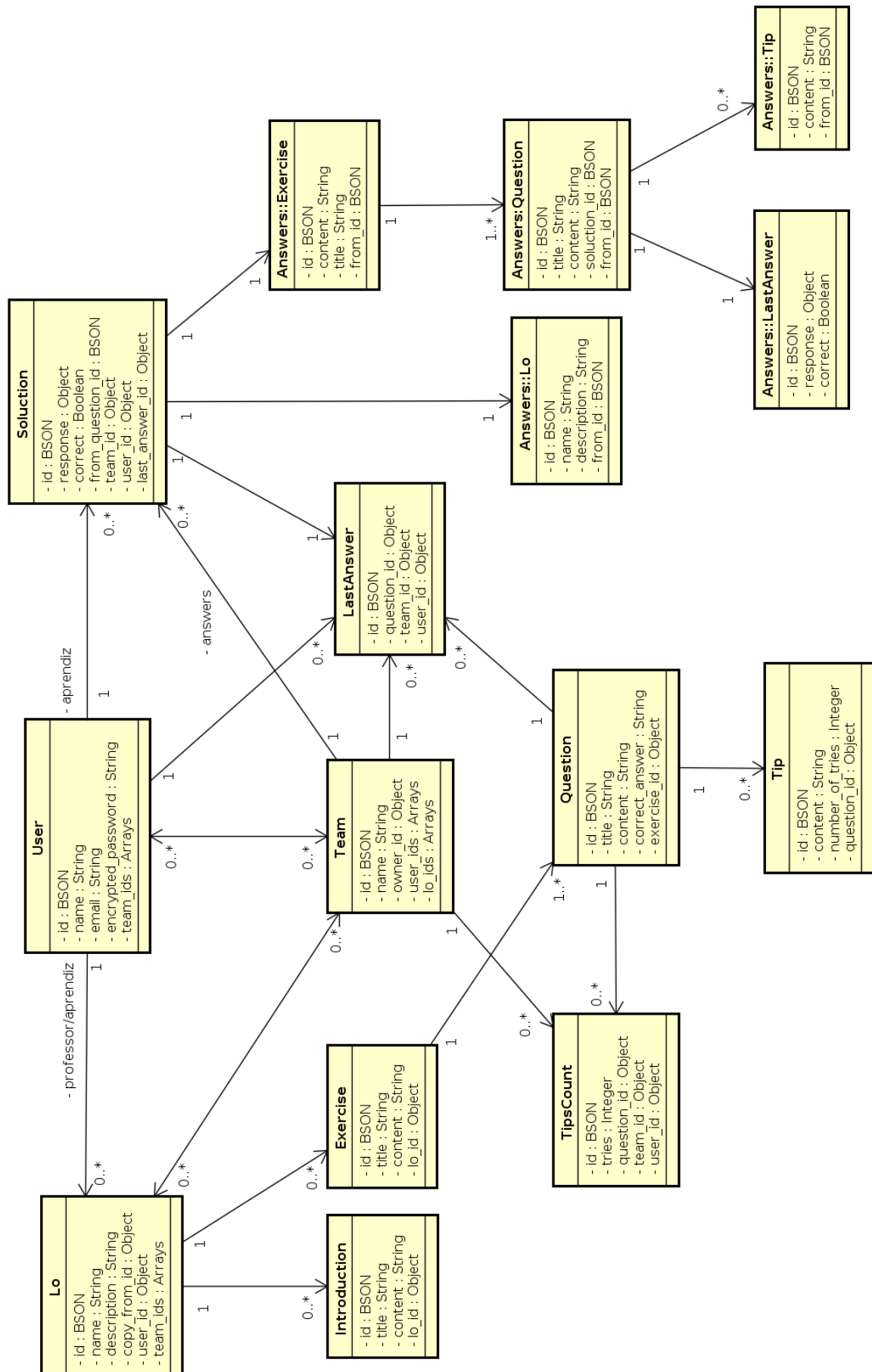
4.1 MODELAGEM DO BANCO DE DADOS ATUAL DA FARMA

Como o banco de dados não tinha modelagem, utilizou-se a *Rubygem RailRoady*¹ para gerar uma espécie de diagrama de classes do banco. Porém era difícil de visualizar as relações nessa modelagem, bem como compreender a estrutura. Então, a partir dela foi criada outra modelagem do banco atual, através de um Diagrama de Classes utilizando a ferramenta case Astah, que permite que a estrutura de um banco de dados NoSQL (que é a partir do modelo orientado a objeto) seja representada de forma mais fiel. Destaca-se que foram representadas nessa modelagem somente as classes que são relevantes para o versionamento dos contextos de interação.

Essa fase do trabalho possibilitou uma melhor visualização das ligações do banco, sendo a base de análise da estrutura atual do banco, e tendo o foco no armazenamento dos contextos de interação do aprendiz com o OA na FARMA. A modelagem feita com a ferramenta Astah é apresentada na [Figura 4](#).

¹<https://github.com/preston/railroady>

Figura 4 – Diagrama de Classes da Ferramenta FARMA



A partir dessa modelagem, foi feita uma descrição do banco de dados para melhor compreensão do domínio do problema deste projeto, especificando cada classe que compõe a modelagem apresentada anteriormente.

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 conterá conceitos, fórmulas, imagens e vídeos.

Classe Exercise. Um exercício compreende a parte prática do OA, conterá 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 - Soluction e Answers::Lo Cada resposta conterà uma versão exclusiva do OA, e este OA terá a referência do original.

Associação A10 - Soluction e Answers::Exercise. Cada resposta conterà 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 conterà 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 conterà a cópia da última resposta que foi dada à ela pelo aprendiz.

Associação A13 – Answers:: Question e Answers::Tip. Também cada versão da questão conterà a cópia de todas as dicas que foram dadas à ela.

4.2 ANÁLISE DA MODELAGEM ATUAL DO BANCO DE DADOS DA FARMA

Nesta etapa realizou-se uma análise da modelagem do banco atual, focando principalmente no armazenamento dos contextos de interação do aprendiz. A partir disso, identificou-se alguns problemas, destacados a seguir.

Observou-se que havia uma grande redundância de dados associada ao armazenamento das respostas. Pois para cada resposta inserida pelo aprendiz, era salva uma cópia de todo o conteúdo do OA, o que corresponde a todas as introduções, exercícios, passos do exercício, as dicas de cada passo, as respostas anteriores do aprendiz e também os *feedbacks* recebidos. O objetivo dessa ação era salvar todo o contexto da interação do aprendiz com o OA, para que ao acessar qualquer resposta, o aprendiz ou o professor-autor pudesse ter a mesma visão do OA no momento em que o aprendiz a inseriu.

Este problema estava relacionado com a forma de como os dados estavam estruturados no banco, pois todos os itens do OA estavam armazenados em documentos separados, e os relacionamentos entre os dados eram feitos pela referência de um documento em outro. Por exemplo, no relacionamento em que um exercício pode ter várias questões, o documento *Question* tinha a referência do documento *Exercise*, da mesma forma, no relacionamento de que uma questão pode ter várias dicas, o documento *Tip* armazenava a referência do documento

Question o qual pertencia. Diante disso, percebeu-se que mesmo utilizando o MongoDB, o banco foi pensado de modo relacional, e sabe-se que para utilizar de maneira eficiente um banco de dados NoSQL, deve-se mudar a maneira de pensar sobre a estrutura de armazenamento dos dados.

Como havia muitos relacionamentos no banco, era necessário realizar várias consultas para recuperar as informações, e também havia um grande número de documentos, que comprometia o desempenho das consultas no MongoDB. Existe o atributo *embedded*, no qual os dados são incorporados dentro de outro, com o objetivo de permitir que aplicações possam recuperar ou manipular os dados relacionados utilizando apenas uma operação. Porém neste tipo de atributo a consulta é dificultada, por exemplo, se os passos estivessem associados ao exercício em forma de documento *embedded*, para acessar um passo isolado seria necessário acessar primeiramente o exercício deste passo. E também, se este tipo de documento continua a crescer muito em tamanho, pode afetar o desempenho da leitura e gravação, e levar à fragmentação de dados (MONGODB, 2017).

Nesse sentido, pelo fato da FARMA possuir muitos relacionamentos, que são necessários para salvar os contextos de interação, identificou-se que este não é um cenário ideal para utilizar o documento *embedded*, pois ocasionaria muitos níveis de dados neste documento, e portanto, decidiu-se não utilizá-lo. Além disso, o banco de dados MongoDB tem suas características voltadas para o ambiente distribuído, e a FARMA é uma aplicação centralizada, o que é um fator negativo na utilização deste banco. Por esse e os motivos anteriores, optou-se pela troca do MongoDB para um SGBD relacional.

Para tanto, foi pesquisado o SGBD PostgreSQL, a fim de identificar se as características eram favoráveis ao banco de dados da FARMA. O PostgreSQL possui um recurso denominado gatilho (*trigger*), que é um procedimento chamado automaticamente pelo SGBD em resposta a alterações especificadas no banco de dados. Esse recurso se mostrou muito útil para a FARMA, pois através dessa funcionalidade era possível configurar ações no banco em determinadas situações, como por exemplo, após a ocorrência de uma inserção ou modificação.

Outra característica importante do PostgreSQL é que este contém novos tipos de dados, que contribuem para que seja criada uma estrutura do banco mais flexível, pois possibilitam que seja armazenado um conjunto de dados em um único campo. Alguns tipos de dados considerados relevantes para a FARMA são o *JSON*, *Array* e *HStore*, pois a FARMA possui características híbridas, requer tanto recursos do modelo relacional como do modelo não relacional. E também, o PostgreSQL tem o suporte a sistemas de alto rendimento e de alto processamento (ENTERPRISEDB, 2016). Essas características foram decisivas para a escolha do PostgreSQL como o novo SGBD para o banco de dados da FARMA.

4.3 ARQUITETURA DA SOLUÇÃO PROPOSTA

Nesta seção é apresentado como foi realizado o processo de criação da modelagem do banco de dados, e também as etapas de desenvolvimento do mecanismo de versionamento até

chegar na solução final.

4.3.1 Arquitetura Inicial

Considerando somente a parte de autoria do OA, foi criada uma primeira versão da nova modelagem do banco de dados da FARMA. A autoria está relacionada com a criação e a disponibilização do OA nas turmas, que envolve as tabelas *users*, *los*, *introductions*, *exercises*, *steps*, e *tips*, e *tips_count*. Esta modelagem foi feita no modelo relacional, utilizando a ferramenta *MySQL Workbench*, e é apresentada no [Apêndice A](#).

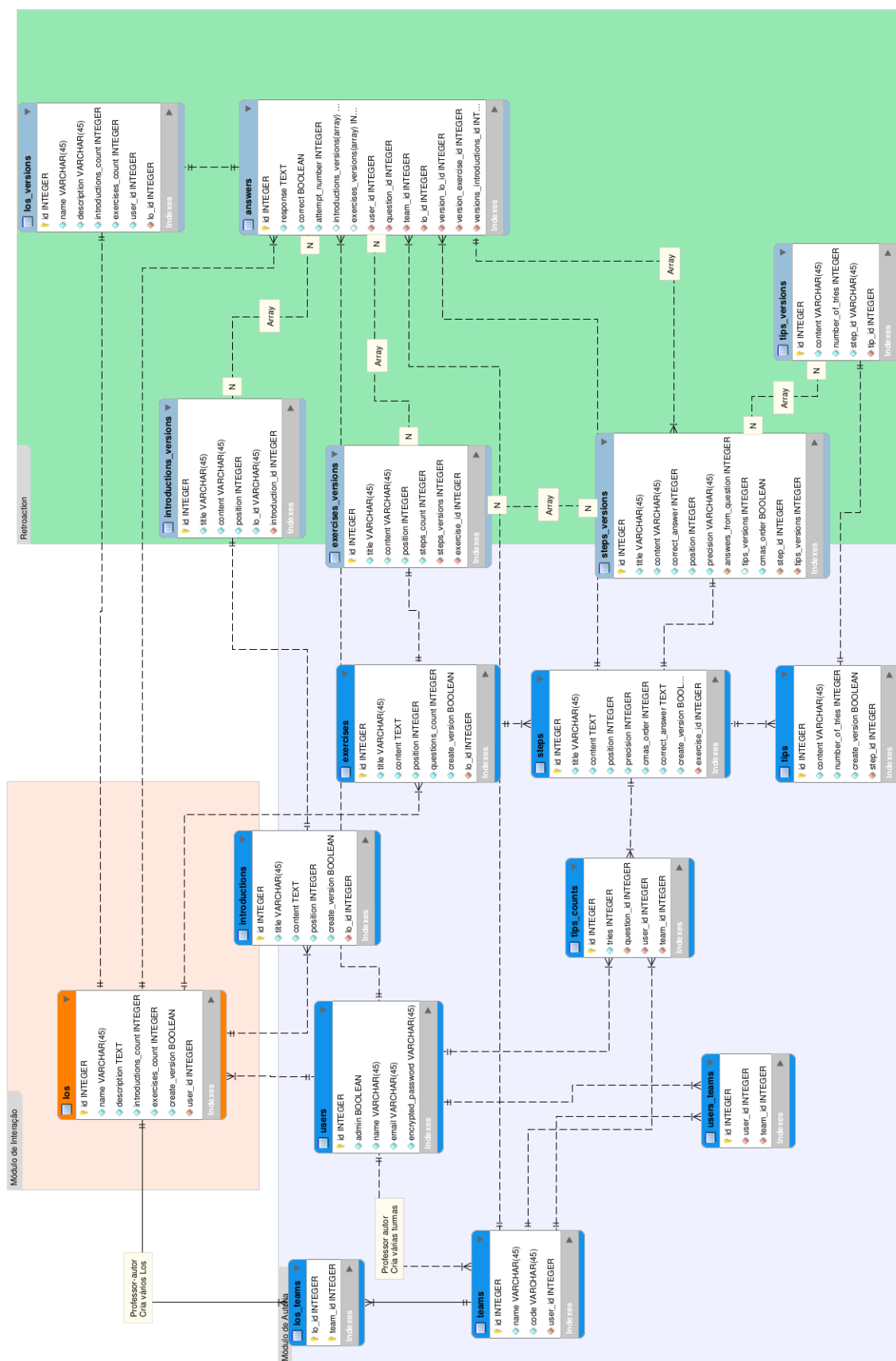
Durante a realização deste trabalho, está sendo desenvolvida uma nova versão da FARMA, sendo assim, esta modelagem já vem sendo utilizada desde o início deste processo, o que é um fator relevante para este trabalho, visto que a modelagem do banco de dados e a criação de um mecanismo de versão são partes essenciais para o desenvolvimento da nova versão da FARMA.

Foi criado também um mecanismo de versionamento dos conteúdos do OA, para garantir que ao retroceder a uma resposta, o aprendiz ou o professor-autor pudesse acessar o contexto correto da interação com o OA. Inicialmente foi criado um protótipo do mecanismo de versionamento, que foi analisado e expandido para a solução completa. Na modelagem foi adicionada uma tabela de versão para cada tabela correspondente a um item do OA (introduções, exercícios, passos, dicas). Porém, as tabelas de versão tinham um relacionamento diferente das tabelas originais, pois como se tratavam de versões do OA para serem adicionadas ao contexto de uma resposta, elas estavam sendo associadas diretamente na resposta.

A associação com a resposta foi feita através de chave estrangeira, e nos casos em que deveria salvar as referências das versões de vários exercícios do OA em uma resposta, foi utilizado o atributo *array* para armazenar essas referências. Pois este tipo de atributo reduz a quantidade de consultas, visto que todos os *ids* já estão na própria entidade, e também poupa o uso de muitas tabelas. Neste caso, devido ao fato da FARMA utilizar tais tipos de dados como *array*, identifica-se que a FARMA possui características híbridas, pois requer tanto recursos do modelo relacional, como do modelo não relacional.

Também foi atribuído um campo do tipo booleano em todas as tabelas correspondentes aos conteúdos do OA, denominado *modified*, para ser usado como uma *flag*, que irá indicar se é necessário ou não criar uma nova versão do respectivo item do OA. Essa modelagem pode ser visualizada na [Figura 5](#).

Figura 5 – Modelagem da Arquitetura Inicial.



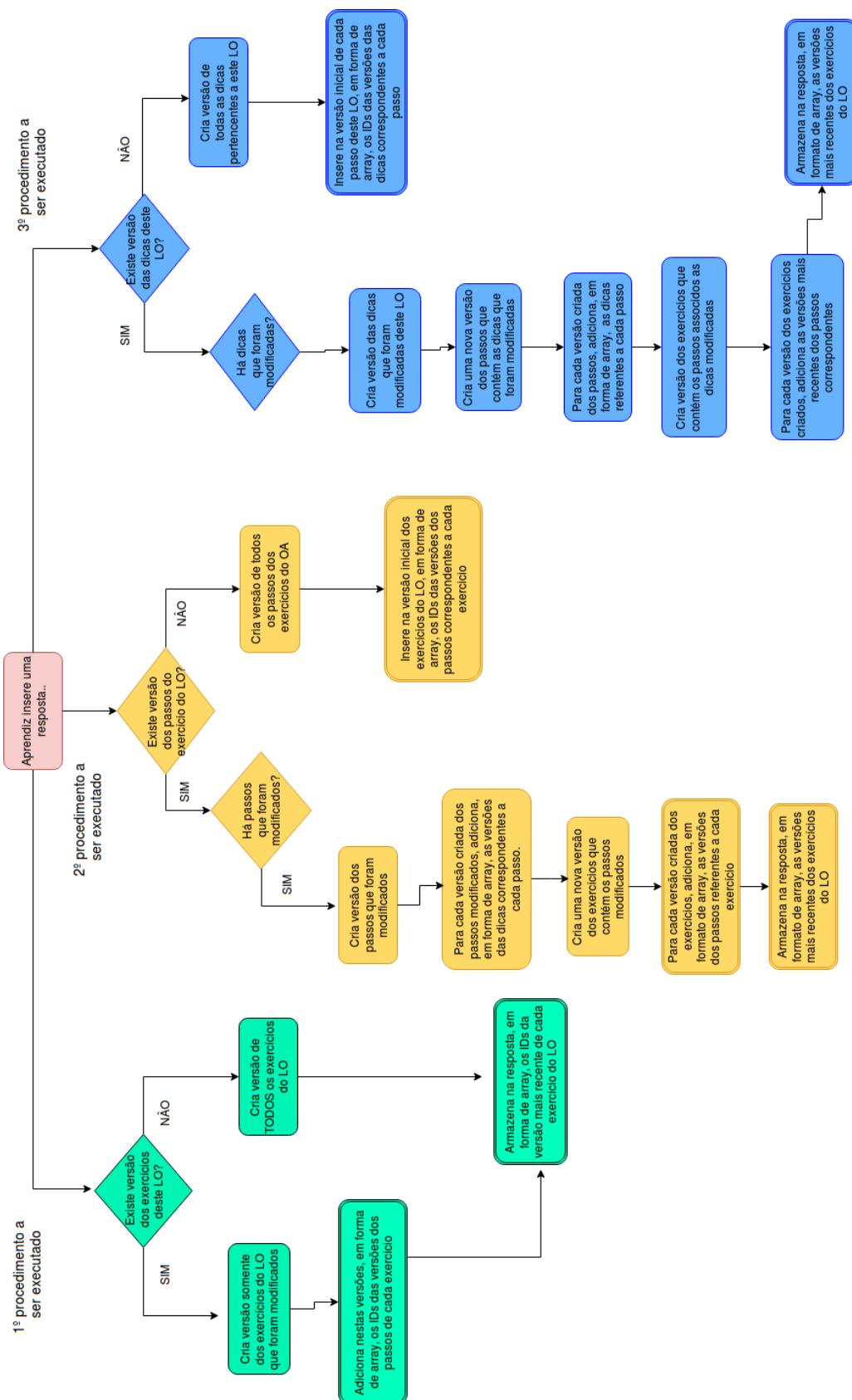
Fonte: Autora

Também foi criado um *script* no banco de dados, no qual foi desenvolvida toda a lógica de funcionamento deste mecanismo. O armazenamento dos contextos já existia na estrutura anterior do banco de dados da FARMA, e funcionava da seguinte forma: quando uma resposta era inserida no banco, criava-se uma cópia do OA, contendo uma cópia dos exercícios e dos passos pertencentes a cada exercício, uma cópia das dicas de cada passo, e também uma cópia da última resposta enviada pelo aprendiz para cada passo. Neste sentido, todas essas cópias eram associadas somente a esta resposta, e na próxima resposta inserida, criava-se novamente a cópia dessas entidades, o que resultava em uma grande repetição dos dados.

Nessa solução, o funcionamento do versionamento pode ser dividido em dois casos principais: o primeiro caso é quando for inserida a primeira resposta no OA, ou seja, logo após este ser adicionado na turma. No momento em que um aprendiz inserir a primeira resposta, o SGBD dispara um conjunto de *triggers*, que verificam nas tabelas de introduções, exercícios, passos, e dicas se existe versão dos mesmos no banco. Como neste caso não há nenhuma versão, cria-se então uma versão de cada um dos conteúdos do OA, que serão associadas a resposta, por meio de chaves estrangeiras, e *arrays* com as referências dos IDs.

O segundo caso é quando forem inseridas outras respostas, as *triggers* serão novamente disparadas, e no entanto, já existe versão de cada conteúdo do OA. Então, verifica se existe alguma modificação relevante de que necessita criar versão, a qual é identificada através da *flag*. Se a *flag* de algum dos conteúdos estiver marcada como *true*, então será criada uma nova versão do respectivo conteúdo, e será referenciada na resposta a versão mais recente deste conteúdo, seja uma introdução, um exercício, um passo ou uma dica. E se não houver modificação no conteúdo do OA, é somente referenciada a versão mais recente na resposta. O funcionamento do mecanismo de versionamento está apresentado no Fluxograma da [Figura 6](#).

Figura 6 – Fluxograma da Solução Inicial



Fonte: Autora

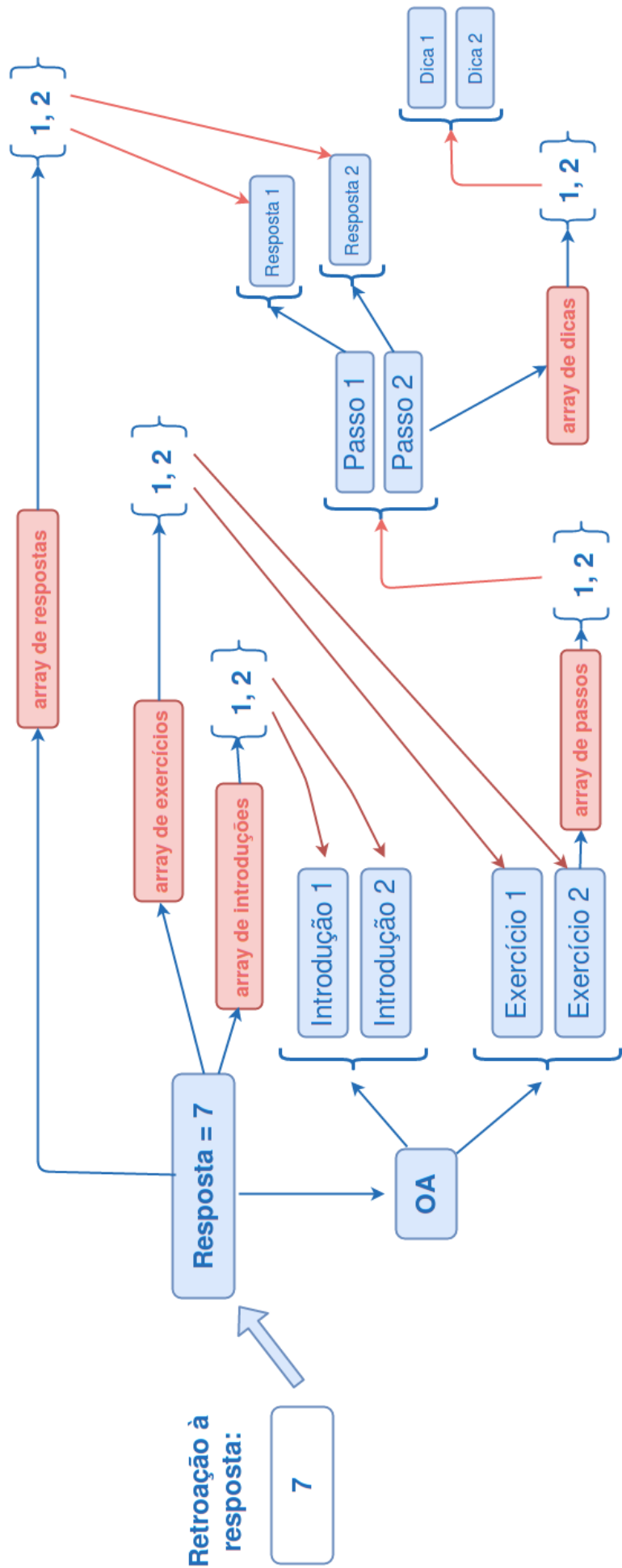
Esse mecanismo garante que seja realizada a retroação ao contexto de interação de uma resposta, através do armazenamento das versões do OA, e associando a referência dessas versões nas respostas. O fato de armazenar somente as referências está relacionado com a diminuição da quantidade de dados redundantes, o que compromete o desempenho das consultas. Pois ao invés de criar uma cópia do conteúdo inteiro do OA todas as vezes que inserir uma resposta, agora a cópia é feita somente se necessário, por meio da verificação da *flag* em cada um dos conteúdos do OA. Caso não seja necessário criar uma nova versão, é associada na resposta a versão mais recente daquele conteúdo. Tal solução resultou na diminuição da redundância dos dados, pois ao invés de ter a versão do “objeto inteiro” associado exclusivamente a uma resposta, agora uma versão pode ser compartilhada entre várias respostas.

Porém, se é feita uma alteração em um passo, será necessário criar a versão do exercício referente a esse passo, assim como quando há a modificação de uma dica, é necessário criar uma nova versão do passo referente a essa dica, e também do exercício referente a esse passo. Isso é feito para garantir que uma modificação não afete o contexto de outras interações. Considera-se este um problema, visto que ainda resulta em uma quantidade de dados redundantes, além de demandar um grande processamento no momento da inserção das respostas.

4.3.1.1 Retroação ao contexto de interação

Nesta solução a recuperação dos contextos é feita da seguinte forma: quando o aprendiz retroceder ao contexto de uma resposta, é feita uma busca no banco de dados a partir da própria resposta, que contém a referência da versão do OA a qual pertence, e também a referência das versões das introduções e dos exercícios. Cada um destes exercícios terá as referências dos passos correspondentes, que serão buscados na tabela de versões de passos e retornados para cada exercício correspondente. Da mesma forma, cada passo terá a referência das dicas, que será buscada na tabela de versões de dicas. E também, a resposta contém as referências das últimas respostas dadas por aquele aprendiz em todo o OA, que serão buscadas na tabela de respostas e associadas a cada passo correspondente. O fluxograma da retroação pode ser visto na [Figura 7](#).

Figura 7 – Retroação na Solução Inicial.



O problema desta solução, é que um OA poderia estar associado a várias turmas simultaneamente, através do relacionamento N:N. Devido a isso, quando um professor realizasse uma modificação no OA, esta modificação passaria a ser apresentada em todas as turmas nas quais o OA estava presente. E isso iria afetar o cálculo das estatísticas dos aprendizes das outras turmas, como por exemplo, em uma turma onde praticamente todos os aprendizes já estivessem respondido a todos os passos do exercício, e fosse feita uma correção através de outra turma nesse OA, modificando o valor de referência da resposta correta de um passo, após a alteração todos os aprendizes estariam com suas respostas atreladas a um contexto antigo, ou seja, com uma referência de resposta incorreta, e assim, teriam suas estatísticas avaliadas de forma incoerente.

Neste caso, pensou-se na possibilidade de notificar os aprendizes quando surgisse uma alteração no OA, indicando que o passo do exercício foi modificado, para que o aprendiz pudesse voltar a responder novamente o exercício, e dessa forma, seria salva com a nova resposta um novo contexto. Também, para que a pontuação dos aprendizes não fossem afetadas, o sistema deveria excluir esse contexto incorreto, ou seja, todas as versões que tivessem essa referência de resposta correta, e as respostas atreladas a esse contexto também seriam excluídas. Porém isso envolveria um sistema de notificação, que necessitaria de uma implementação e uma interface mais complexa.

Além disso, para restringir as alterações por turma, seria necessário ter uma quantidade maior de cópias, o que teria uma complexidade em gerenciá-las, pois haveriam várias versões de uma mesma introdução, exercício, passo, dica, necessárias para salvar o contexto exato de todo o OA, ao mesmo tempo, teria que fazer muitas consultas em várias tabelas para recuperar a versão correta do OA, o que demandaria um processamento mais pesado. Identificou-se que essa solução fornecida para o armazenamento de contextos era muito complexa, e devido a isso, buscou-se uma forma mais simples de garantir o contexto para cada turma.

4.3.2 Arquitetura Final

Sendo assim, buscou-se uma solução que pudesse garantir que as correções feitas em um OA não afetem todas as aplicações já realizadas desse OA, mas somente para aquela turma que o professor-autor deseja incluir tais alterações, ou mais especificamente, para uma aplicação individual do OA na turma.

Outro ponto a ser considerado também é de garantir que o professor-autor possa fazer alterações no OA para aplicar nas próximas turmas, mas que estas alterações não interfiram no contexto das aplicações atuais deste OA. Dessa forma, será possível garantir a integridade das estatísticas, pois quando um aprendiz inserir uma resposta no OA, o cálculo das estatísticas do aprendiz e da turma será feito com base somente no contexto de uma aplicação específica do OA.

Foi criada uma nova solução para garantir os contextos do OA e a integridade das estatísticas, na qual cria-se uma cópia do OA para a turma, a cada vez que o professor-autor

adicionar este OA na turma. Essa forma garante que as modificações feitas pelo professor-autor afetem apenas uma aplicação em específico do OA, sem afetar outras turmas ou até mesmo outras aplicações deste OA na turma. Essa forma também dispensa a interação com o professor-autor no momento da geração da versão, pois anteriormente, ele deveria tomar uma decisão a respeito de criar uma nova versão do OA ou não. E a ideia da FARMA é ser simples, e que o professor precise tomar decisões sobre ações na FARMA.

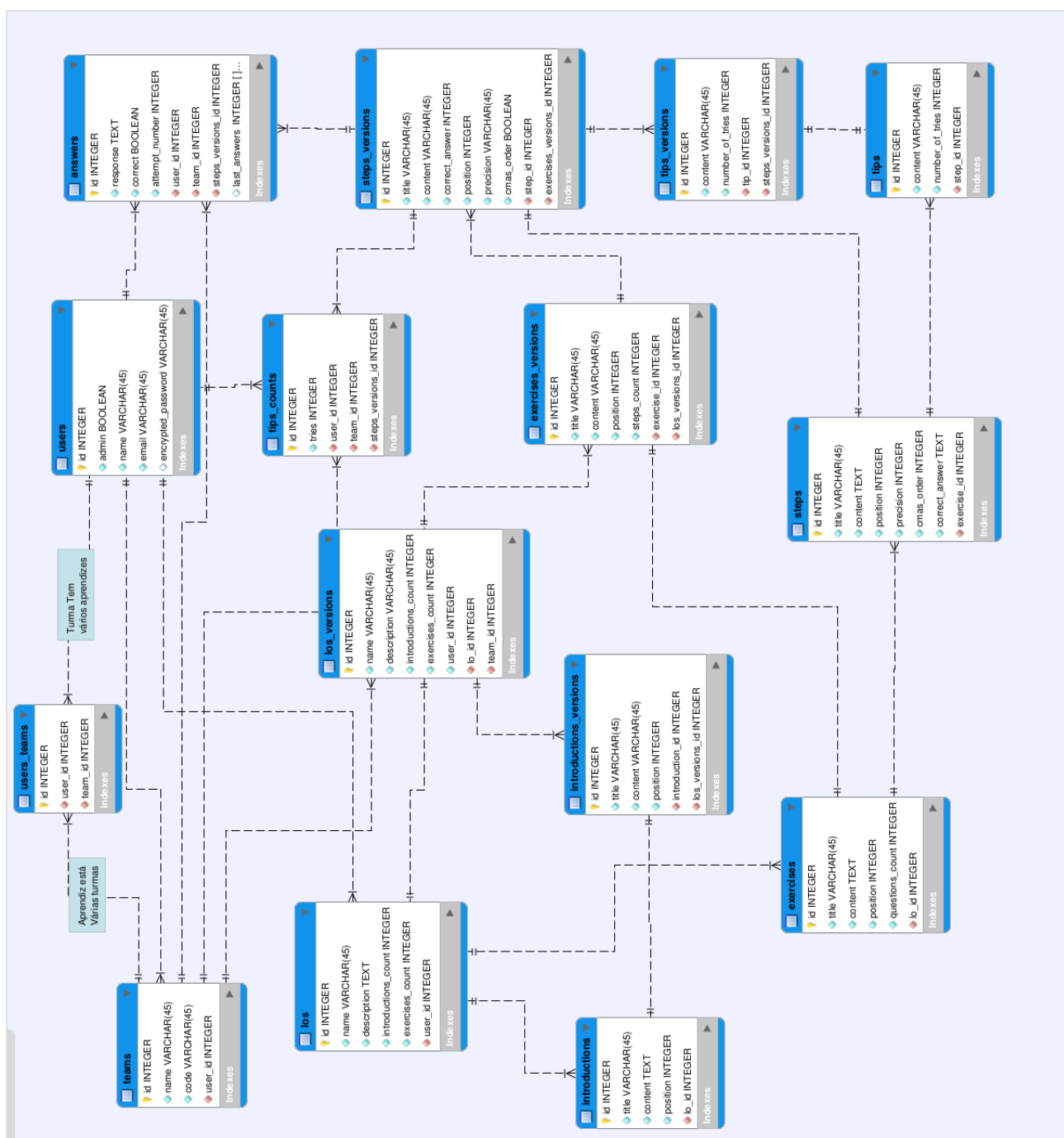
Também, optou-se por restringir as modificações do professor-autor no OA da turma, permitindo a este fazer apenas modificações leves, tais como, adicionar uma nova dica, alterar o enunciado de um passo, mudar o título de um exercício. Quando o professor-autor quiser fazer uma modificação mais abrangente no OA, como adicionar um exercício, passos, dicas ou um passo, este terá uma área exclusiva para alterar o OA.

No desenvolvimento dessa solução, algumas modificações foram feitas na modelagem. O relacionamento de N:N entre as tabelas *teams* e *los*, representado pela tabela intermediária *los_teams* foi retirado, e a tabela *teams* passou a ter um relacionamento de 1:N com a tabela *versions_los*, para que cada cópia do OA seja exclusiva de uma turma. As tabelas de versão dos conteúdos do OA foram mantidas, porém as referências que existiam dessas versões diretamente na resposta foram retiradas. Agora, a resposta armazena apenas a referência do passo correspondente à ela, o qual pertence a versão dos conteúdos do OA desta turma. A modelagem desta solução é apresentada na [Figura 8](#), esta contém apenas alguns atributos de cada tabela, e a modelagem completa, com todos os atributos está representada no [Apêndice C](#).

Para realizar a ação de criação de cópia, foi criada uma *trigger* no SGBD, que é disparada no momento em que o professor-autor insere um OA na turma, e esta cria uma cópia de todos os itens do OA, e associa esta cópia do OA na turma. Dessa forma, foi dispensado o uso de várias versões, sendo utilizada apenas uma única versão do OA para cada aplicação, e através dessa versão que os aprendizes irão interagir dentro da turma. Isso torna mais fácil o gerenciamento dos contextos do OA, sem precisar de uma busca complexa para salvar o contexto correto para cada resposta do aprendiz.

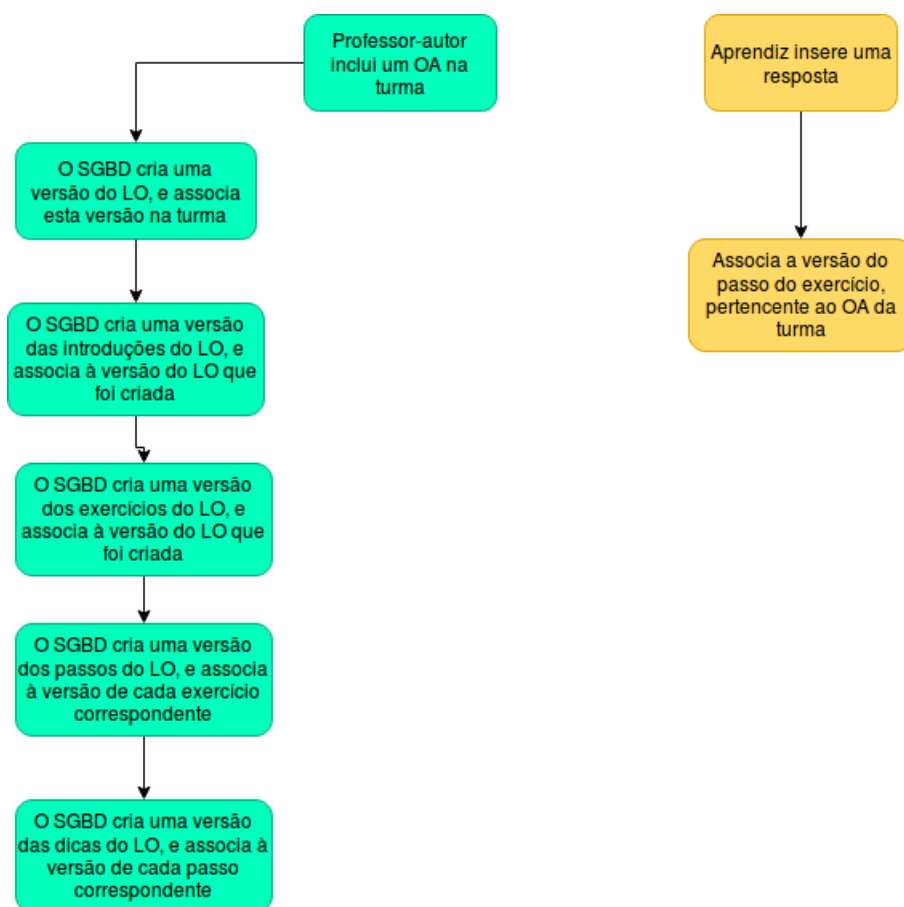
Essa modelagem contém um pouco mais de redundância, no sentido de criar uma cópia do OA para cada turma, mas que é necessária para garantir o contexto de interação dos aprendizes com o OA específico da turma, e assim, garantir a integridade das estatísticas. E também, nessa solução, a criação da versão do OA não dependerá da decisão do professor-autor, e esta ação se realizará sem o conhecimento do professor-autor, pois a FARMA tem o objetivo de ter uma interação simples com o usuário. O funcionamento dessa solução está representando no Fluxograma da [Figura 9](#).

Figura 8 – Modelagem da Arquitetura Final



Fonte: Autora

Figura 9 – Fluxograma da Solução Final



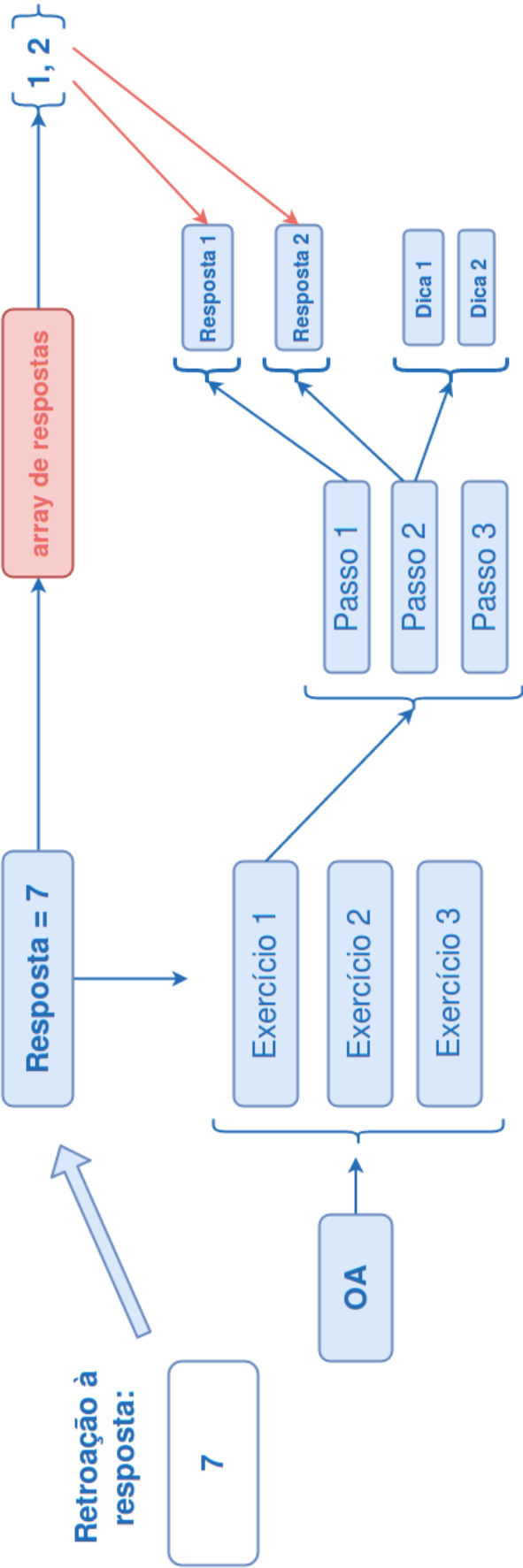
Fonte: Autora

4.3.2.1 Retroação ao contexto de interação

Na segunda solução, a retroação aos contextos é feita da seguinte forma: a partir da própria resposta, que contém a referência da versão do passo a qual pertence, é retornado o OA referente à resposta, bem como as introduções e os exercícios deste OA. A partir da versão de cada exercício são retornadas as versões dos passos correspondentes. Da mesma forma, a partir da versão de cada passo, é buscada a versão das dicas associadas ao passo. Por fim, a resposta contém as referências das últimas respostas dadas por aquele aprendiz em todo o OA, que serão buscadas na tabela de respostas e associadas a cada passo correspondente.

Nessa solução, a existência de uma cópia do OA exclusiva por turma, bem como a restrição a apenas as modificações leves foram pensadas com o intuito de não comprometer o contexto do OA dentro de uma turma, e também as estatísticas dos aprendizes e da turma. Outro ponto é que o armazenamento dos contextos será facilitado, o qual passará a ser salvo apenas uma única cópia do OA, sem precisar realizar uma busca complexa para retornar o contexto exato da interação de cada aprendiz com o OA. O fluxograma da retroação a partir dessa segunda solução está representado na [Figura 10](#).

Figura 10 – Retroação na Solução Final



5 RESULTADOS

De acordo com os objetivos expostos na Seção 1.1, apresenta-se uma sumarização com os resultados alcançados pelo trabalho, a saber:

1. Descrição do banco de dados atual da FARMA, com enfoque dado a parte de armazenamento dos contextos de interação. A partir desta modelagem pode-se fazer uma análise do banco atual, e identificou-se que havia uma redundância desnecessária, e que estava associada com a forma com que os dados estavam estruturados, e além disso, tal estrutura comprometia o desempenho das consultas, por estarem dispersas em diversos documentos.
2. Alteração do SGBD MongoDB para o PostgreSQL. Identificou-se que o banco de dados MongoDB não era o mais apropriado para os requisitos da FARMA, sendo então substituído pelo SGBD relacional PostgreSQL, por este ter características de modelo híbrido. Dessa forma, foi feita uma nova modelagem da parte de autoria do OA, para ser utilizada no desenvolvimento da nova versão da FARMA;
3. Elaboração do mecanismo de versionamento dos contextos de interação do OA. Foi criada a modelagem e desenvolvimento do mecanismo de versionamento dos OAs. Esse mecanismo possibilita o armazenamento de versões de partes do OA, para serem referenciadas no contexto de cada resposta inserida, assim era possível garantir o acesso aos contextos da interação dos aprendizes. Porém esse mecanismo tinha um problema, no qual as versões do mesmo OA eram compartilhadas entre diferentes turmas, e dessa forma, alterações feitas em uma turma eram reproduzidas nas outras que continham o mesmo OA.
4. Criação de uma nova arquitetura de solução para o armazenamento de contextos. Para que as modificações de um OA não afetassem outras turmas, foi criada uma nova modelagem, na qual era feita uma cópia exclusiva do OA para cada turma, que seria a base para todas as interações dos aprendizes. Nessa arquitetura, as modificações do professor-autor afetariam somente uma turma em específico, e também a recuperação dos contextos foi simplificada, pois não era mais necessário uma grande quantidade de consultas no banco para fazer a retroação.
5. Foram adicionadas na modelagem alguns dados para armazenar as informações das estatísticas. Na forma de uma pontuação, que irá ser salva de acordo com a interação do aprendiz com o OA. Essas informações possibilitam salvar a pontuação de uma aprendiz em relação a um passo de um exercício, a um exercício, e a um OA específico. Mais detalhes podem ser visto no [Apêndice B](#).

6 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi criar a modelagem de um mecanismo de versionamento dos objetos de aprendizagem para a FARMA, para que fosse possível garantir o acesso aos contextos da interação dos aprendizes com o OAs nas turmas, de forma que houvesse pouca redundância de dados.

A primeira solução tinha o intuito de salvar todas as modificações consideradas importantes do OA, para que pudesse ser salvo o contexto atual da interação do aprendiz com maior fidelidade. Nessa modelagem, a redundância dos dados foi minimizada, pois continha menos cópias do OA, visto que era salva a cópia de apenas “parte do OA”, e somente se necessário. Além disso, as cópias eram associadas por referência, podendo ser referenciadas também em outras instâncias.

Porém, o problema dessa modelagem era que um mesmo OA poderia ser compartilhado em várias turmas, sendo assim, quando houvessem alterações nesse OA, todas as turmas que tivessem aquele OA iriam receber as modificações, o que comprometeria as estatísticas do aprendiz e da turma como um todo. Nesse sentido, buscou-se criar uma nova solução, com o objetivo de restringir as modificações feitas no OA pelo professor-autor, para que não afetassem as outras turmas.

Pensou-se na possibilidade de criar um mecanismo de notificação, para notificar todos os aprendizes que tivessem respondido ao passo ou exercício associado a modificação, para que estes respondessem novamente. Além disso, o sistema deveria apagar todas as respostas associadas ao contexto antigo, para que não afetasse as estatísticas do aprendiz e da turma. Porém, criar tal mecanismo e gerenciar esses contextos demandaria uma implementação muito complexa, que não seria viável de desenvolver.

Sendo assim, optou-se por restringir as cópias do OA para uma única cópia por turma, na qual será a referência para todas as interações dos aprendizes, e também, o professor-autor não pode realizar modificações muito abrangentes, para que as estatísticas não sejam afetadas. Dessa forma, foi feita uma alteração em todo o mecanismo de versionamento, visando simplificar o gerenciamento dos contextos de interação.

A contribuição deste trabalho foi elaborar uma solução de dados que fosse capaz de armazenar os contextos corretos de interação do aprendiz, e garantir a integridade das estatísticas de cada aprendiz, para que não fossem afetadas por outros contextos. Além disso, outra contribuição foi a substituição do banco de dados não relacional para um relacional com características híbridas, para que os relacionamentos fossem melhor estruturados no banco. E a partir da solução final, foi obtida uma modelagem que provê a garantia dos contextos e das estatísticas com uma menor complexidade nas consultas, e uma menor redundância nos dados.

6.1 TRABALHOS FUTUROS

Nesta seção apresenta-se algumas sugestões de pesquisas e trabalho que podem ser feitos com base neste trabalho, a destacar:

1. Realização de testes de desempenho. Visto que o foco principal deste trabalho não foi na melhoria de desempenho, mas apenas no armazenamento e recuperação dos contextos corretos de cada interação do aprendiz com o OA, não foram feitos teste de carga para identificar onde o banco está tendo maior rapidez ou lentidão. Dentro disso, podem ser feitos em trabalhos futuros testes para avaliar o desempenho do banco de dados, utilizando algumas técnicas como Teste de Carga, Teste de Stress, que servirão de base para justificar quantitativamente onde podem ser feitas melhorias no banco de dados.
2. Otimização. Sabe-se que tão importante quanto a disponibilidade de um conteúdo, é a velocidade com que esse conteúdo é entregue. Dessa forma, pode ser feito uma otimização na FARMA, buscando melhorar o desempenho da aplicação e do banco de dados, para que os conteúdos sejam salvos e recuperados com uma maior velocidade.

Referências

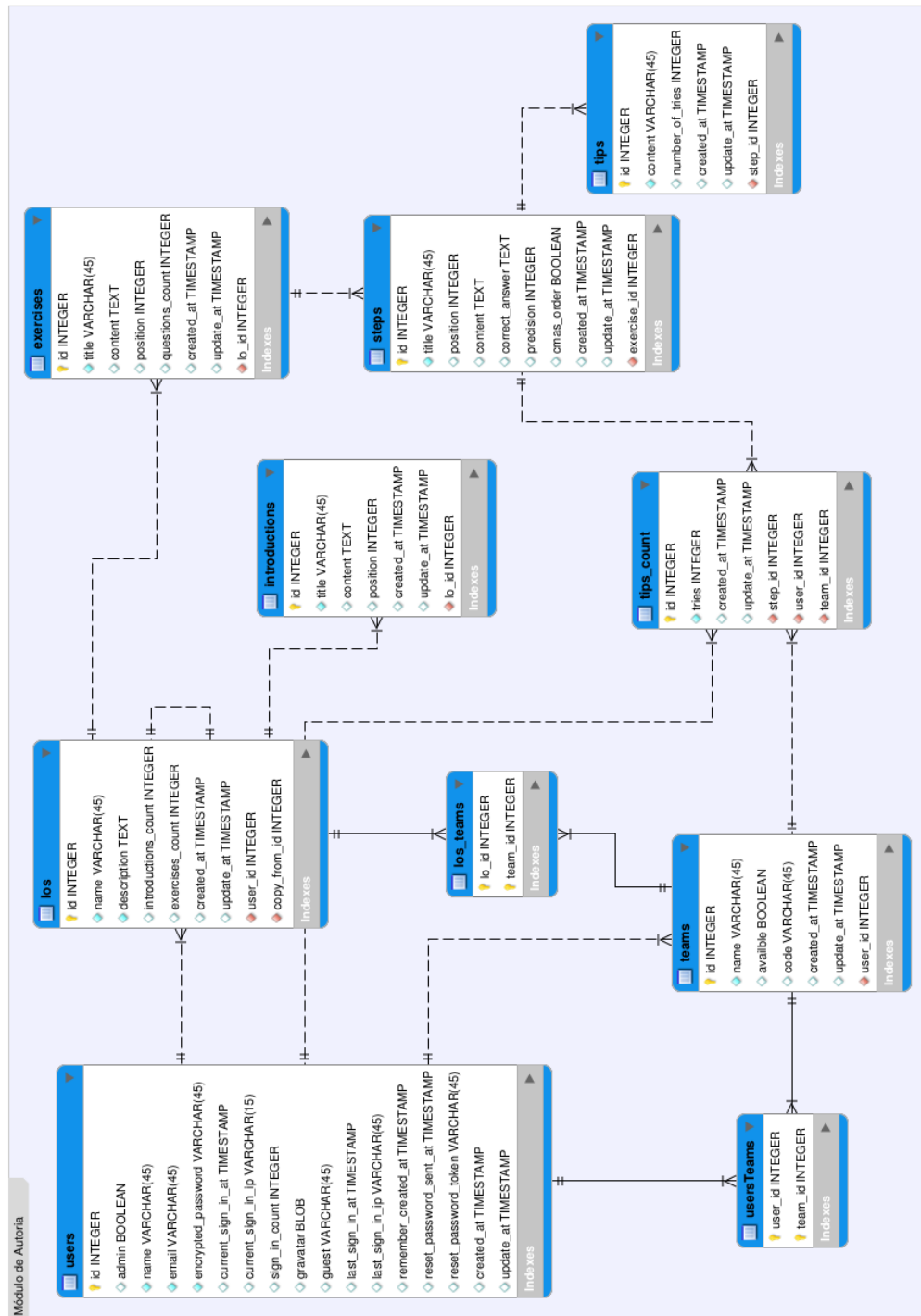
- BLACKBOARD. **Hospedagem Gerenciada da Blackboard - Blackboard Brasil**. 2016. Disponível em: <<http://blackboard.grupoa.com.br/servicos/hospedagem-gerenciada-da-blackboard/>>. Acesso em: 6 de abril de 2016. Citado na página 1.
- CHACON, S.; STRAUB, B. **Pro Git**. 2. ed. Nova York: Apress, 2014. Citado na página 9.
- ENTERPRISEDB. **NoSQL for the Enterprise**. 2016. Disponível em: <<http://www.enterprisedb.com/nosql-for-enterprise>>. Acesso em: 29 de março de 2016. Citado 2 vezes nas páginas 2 e 20.
- GAMA, C. L. G. da. **Método de construção de objetos de aprendizagem com aplicação em métodos numéricos**. 210 f. Tese (Doutorado em Métodos Numéricos em Engenharia) — Universidade Federal do Paraná, Curitiba, 2007. Citado na página 1.
- HEUSER, C. A. **Projeto de Banco de Dados**. 6. ed. Porto Alegre: Bookman, 2009. Citado 2 vezes nas páginas 8 e 9.
- HOWS, D.; MEMBREY, P.; PLUGGE, E. **Introdução ao MongoDB**. 1. ed. São Paulo: Novatec, 2015. Citado 2 vezes nas páginas 10 e 11.
- LINSTER, M. **Postgres Outperforms MongoDB and Ushers in New Developer Reality**. 2014. Disponível em: <<http://www.enterprisedb.com/postgres-plus-edb-blog/marc-linster/postgres-outperforms-mongodb-and-ushers-new-developer-reality>>. Acesso em: 29 de março de 2016. Citado 2 vezes nas páginas 2 e 10.
- MANNINO, M. **Projeto, Desenvolvimento de Aplicações e Administração de Banco de Dados**. 3. ed. São Paulo: AMGH, 2008. Citado 2 vezes nas páginas 12 e 13.
- MARCZAL, D. **FARMA: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos**. 194 f. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba, 2014. Citado 3 vezes nas páginas 1, 5 e 8.
- MONGODB. **Data Model Design**. 2017. Citado na página 20.
- RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de Banco de Dados**. 3. ed. São Paulo: McGraw-Hill, 2008. Citado 2 vezes nas páginas 11 e 12.
- SILVA, E. M. D. da. A virtude do erro: uma visão construtiva da avaliação. **Estudos em Avaliação Educacional**, v. 19, p. 91–111, 2008. Disponível em: <<http://www.fcc.org.br/pesquisa/publicacoes/ae/arquivos/1420/1420.pdf>>. Acesso em: 08 de Junho de 2016. Citado na página 8.
- SILVA, R. C. **Sequenciamento Adaptativo de Exercícios Baseado na Correspondência entre a Dificuldade da Solução e o Desempenho Dinâmico do Aprendiz**. 82 f. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba, 2015. Citado 2 vezes nas páginas 2 e 8.

SPINELLIS, D. Git - tools of the trade. **IEEE SOFTWARE**, v. 29, p. 100–101, 2012. Disponível em: <<https://www.dmst.aueb.gr/dds/pubs/jrnl/2005-IEEESW-TotT/html/v29n3.pdf>>. Acesso em: 14 de Abril de 2016. Citado na página 10.

Apêndices

APÊNDICE A – MODELAGEM DA PARTE DE AUTORIA DA FARMA

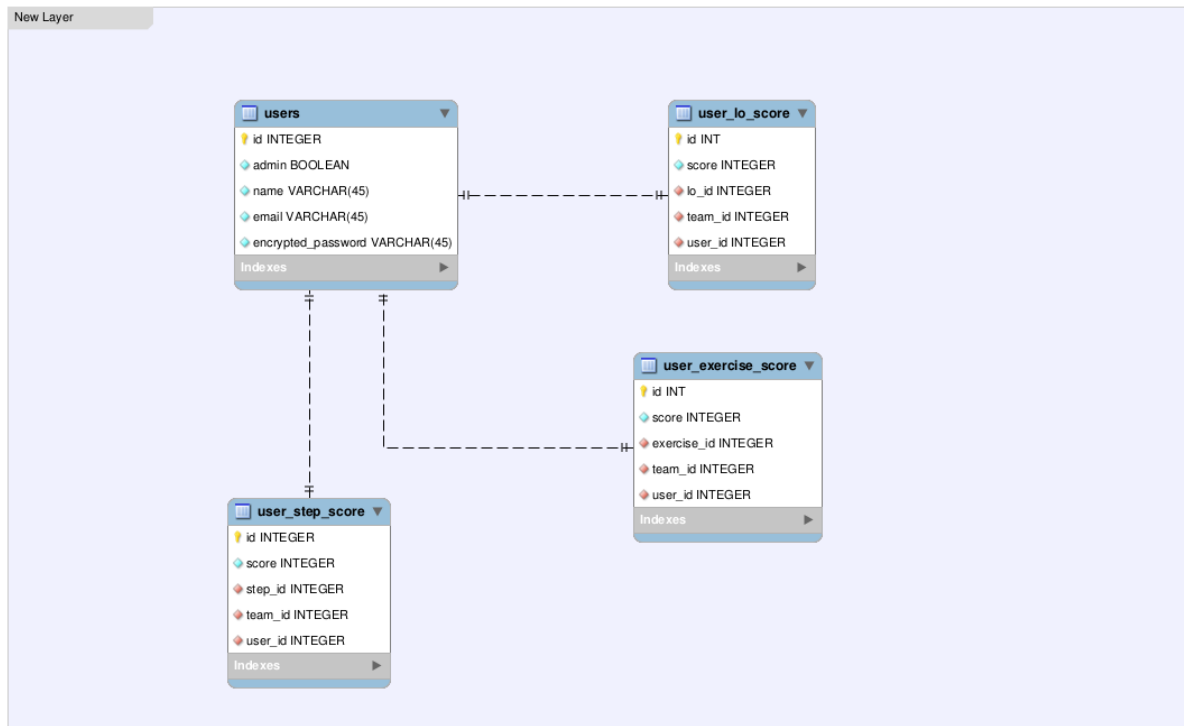
Figura 11 – Modelagem da Parte de Autoria da FARMA



Fonte: Autora

APÊNDICE B – MODELAGEM DAS ESTATÍSTICAS DO APRENDIZ

Figura 12 – Modelagem das Estatísticas do Aprendiz



Fonte: Autora

The ER diagram illustrates the database structure for 'Projeto de Banco'. It includes the following tables and their attributes:

- teams**: id INTEGER (PK), name VARCHAR(45), code VARCHAR(45), available BOOLEAN, created_at TIMESTAMPTZ, user_id INTEGER (FK).
- users_teams**: user_id INTEGER (FK), team_id INTEGER (FK).
- users**: id INTEGER (PK), admin BOOLEAN, name VARCHAR(45), email VARCHAR(45), encrypted_password VARCHAR(45), current_sign_in_at TIMESTAMPTZ, sign_in_count INTEGER, gravatar BLOB, guest_sign_in_at TIMESTAMPTZ, last_sign_in_ip VARCHAR(45), remember_created_at TIMESTAMPTZ, reset_password_sent_at TIMESTAMPTZ, reset_password_token VARCHAR(45), created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ.
- los_versions**: id INTEGER (PK), name VARCHAR(45), description VARCHAR(45), introductions_count INTEGER, exercises_count INTEGER, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, team_id INTEGER (FK).
- los**: id INTEGER (PK), name VARCHAR(45), description TEXT, introductions_count INTEGER, exercises_count INTEGER, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, user_id INTEGER (FK).
- introductions_versions**: id INTEGER (PK), title VARCHAR(45), content VARCHAR(45), position INTEGER, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, lo_id INTEGER (FK).
- introductions**: id INTEGER (PK), title VARCHAR(45), content TEXT, position INTEGER, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, lo_id INTEGER (FK).
- exercises_versions**: id INTEGER (PK), title VARCHAR(45), content VARCHAR(45), position INTEGER, steps_count INTEGER, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, exercise_id INTEGER (FK).
- exercises**: id INTEGER (PK), title VARCHAR(45), content TEXT, position INTEGER, steps_count INTEGER, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, lo_id INTEGER (FK).
- tips_versions**: id INTEGER (PK), title VARCHAR(45), position INTEGER, content VARCHAR(45), correct_answer TEXT, precision VARCHAR(45), cmask_order INTEGER, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, step_id INTEGER (FK).
- tips**: id INTEGER (PK), title VARCHAR(45), position INTEGER, content TEXT, correct_answer TEXT, precision VARCHAR(45), cmask_order INTEGER, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, exercise_id INTEGER (FK).
- steps_versions**: id INTEGER (PK), title VARCHAR(45), position INTEGER, content VARCHAR(45), correct_answer INTEGER, precision VARCHAR(45), cmask_order BOOLEAN, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, step_id INTEGER (FK).
- steps**: id INTEGER (PK), title VARCHAR(45), position INTEGER, content TEXT, correct_answer TEXT, precision VARCHAR(45), cmask_order INTEGER, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, exercise_id INTEGER (FK).
- answers_versions**: id INTEGER (PK), response TEXT, attempt_number INTEGER, last_answers INTEGER[], created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, user_id INTEGER (FK), team_id INTEGER (FK), step_id INTEGER (FK).
- answers**: id INTEGER (PK), response TEXT, attempt_number INTEGER, last_answers INTEGER[], created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, user_id INTEGER (FK), team_id INTEGER (FK), step_id INTEGER (FK).

Relationships are defined as follows:

- teams** to **users_teams**: One-to-many relationship.
- users_teams** to **users**: One-to-many relationship.
- los_versions** to **los**: One-to-many relationship.
- introductions_versions** to **introductions**: One-to-many relationship.
- exercises_versions** to **exercises**: One-to-many relationship.
- tips_versions** to **tips**: One-to-many relationship.
- steps_versions** to **steps**: One-to-many relationship.
- answers_versions** to **answers**: One-to-many relationship.

Fonte: Autora