

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

ANDRÉ FELIPE SILVEIRA

**DESENVOLVIMENTO DE API PARA RETROAÇÃO A ERROS E  
LINHA DO TEMPO NA FARMA.**

TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA  
2019

ANDRÉ FELIPE SILVEIRA

**DESENVOLVIMENTO DE API PARA RETROAÇÃO A ERROS E  
LINHA DO TEMPO NA FARMA.**

Projeto de 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

Coorientadores: Prof. Me. Alex Sandro de Castilho  
Universidade Tecnológica Federal do  
Paraná - Câmpus Guarapuava

Prof. Dr. Eleandro Maschio.  
Universidade Tecnológica Federal do  
Paraná - Câmpus Guarapuava

GUARAPUAVA  
2019

## RESUMO

SILVEIRA, André Felipe. Desenvolvimento de API para retroação a erros e linha do tempo na FARMA.. 2019. 35 f. Trabalho de Conclusão de Curso – Câmpus Guarapuava, Universidade Tecnológica Federal do Paraná. Guarapuava, 2019.

A FARMA é uma ferramenta que tem por objetivo o ensino da matemática e também outras matérias que envolvem expressões algébricas e/ou aritméticas através de objetos de aprendizagem (OA). Na Ferramenta o tutor cria um OA, descreve uma introdução e cria exercícios para acompanhar e avaliar seus aprendizes. Um dos focos da ferramenta é a retroação à erros onde um aprendiz pode ter acesso ao histórico de suas respostas inseridas, além de poder também rever dicas e passos que estavam presentes no momento em que uma resposta foi criada. Já para o professor, estatísticas juntamente com a retroação à erros permitem avaliar se o método de ensino aplicado tem surtido efeito para melhorar o desempenho de seus aprendizes. Atualmente, a ferramenta encontra-se em remodelagem, pois há alguns pontos de melhoria. No atual estágio da ferramenta, ainda não existe a disponibilidade de dados para a retroação a erros e linha do tempo. E assim, trabalho tem como objetivo contribuir com a remodelagem da ferramenta criando as API's para disponibilizar os dados para a retroação à erros e para a linha do tempo das interações de um aprendiz com a plataforma. Até o momento foram definidas as chamadas de APIs e o banco de dados a ser utilizado para a linha do tempo.

**Palavras-chave:** API. Retroação à erros. Contexto de Interação. Linha do tempo.

## ABSTRACT

SILVEIRA, André Felipe. Development of API for error retroaction and timeline in FARMA.. 2019. 35 f. Trabalho de Conclusão de Curso – Câmpus Guarapuava, Universidade Tecnológica Federal do Paraná. Guarapuava, 2019.

FARMA is a tool that aims to support to teach mathematics, as well as other subjects involve algebraic and / or arithmetic expressions, through learning objects (LO). In the tool, the tutor can create a LO, that contains introductions and exercises, that are used to evaluate his learners. One of the focuses of the tool is retroaction to an error, where a learner can have access to the history of his / her inserted answers, in addition to being able to review tips and steps that were present at the moment an answer was created, also it is possible to visualize a line of the time of all the interactions he performed in the tool. For the teacher, statistics together with retroaction to an error allow us to evaluate whether the applied teaching method has had an effect on improving the performance of its learners. Currently, the tool is in remodelling, as there are some improvement points. This work aims to participate in the remodelling of the tool creating the API's to make available and store the data from students errors and the timeline of the interactions of a learner with the platform.

**Keywords:** API. Retroaction to an Error. Interaction Contexto. Timeline.

## LISTA DE FIGURAS

Figura 1 – Tela de criação do OA - FARMA . . . . .	6
Figura 2 – Tela de criação da Introdução - FARMA . . . . .	7
Figura 3 – Tela de criação de Exercícios - FARMA . . . . .	8
Figura 4 – Visualização do OA - FARMA . . . . .	8
Figura 5 – Visualização de passos e dicas - FARMA . . . . .	9
Figura 6 – Arquitetura funcionalista da FARMA . . . . .	10
Figura 7 – Teclado virtual - FARMA . . . . .	12
Figura 8 – Modelagem Retroação - FARMA . . . . .	34
Figura 9 – Modelagem Completa - FARMA . . . . .	35

## LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
API	Application Programming Interface (Interface de Programação de Aplicativos)
CRUD	Create, Read, Update e Delete (Criação, Consulta, Atualização e Destruição)
FARMA	Ferramenta de Autoria para a Remediação de erros com Mobilidade na Aprendizagem
GPL	Licença Pública Geral
HTTP	Hyper Text Transfer Protocol (Protocolo de Transferência de Hipertexto)
OA	Objetos de Aprendizagem
REST	REpresentation State Transfer (Transferência de Estado Representacional)
SGBD	Sistema Gerenciador de Banco de Dados
TSDB	Time Series Database (Banco de dados de series temporais)
UTFPR	Universidade Tecnológica Federal do Paraná

## SUMÁRIO

<b>1 – INTRODUÇÃO</b>	<b>1</b>
1.1 CONTEXTO DO PROJETO	2
1.2 DEFINIÇÃO DO PROBLEMA	3
1.2.1 RETROAÇÃO À ERROS	3
1.2.2 LINHA DO TEMPO	3
<b>2 – OBJETIVOS</b>	<b>4</b>
2.1 OBJETIVO GERAL	4
2.2 OBJETIVOS ESPECÍFICOS	4
<b>3 – FARMA</b>	<b>5</b>
3.1 ERROS	5
3.2 FARMA	6
3.3 ARQUITETURA FUNCIONALISTA DA FARMA	10
3.3.1 AUTORIA	11
3.3.2 INTERAÇÃO	11
3.3.3 ACOMPANHAMENTO	11
3.3.3.1 INDEXADOR DE RESPOSTAS	12
3.3.3.2 RETROAÇÃO ÀS RESPOSTAS	13
3.3.3.3 MODELO DO APRENDIZ	13
<b>4 – METODOLOGIA DE DESENVOLVIMENTO</b>	<b>14</b>
4.1 KANBAN	14
4.2 LINGUAGEM DE PROGRAMAÇÃO RUBY	15
4.3 FRAMEWORK RAILS	16
4.4 POSTGRESQL	17
4.5 API RESTFULL	17
4.6 API RESTFULL COM RUBY ON RAILS	18
<b>5 – DESENVOLVIMENTO</b>	<b>20</b>
5.1 RETROAÇÃO À ERROS	20
5.1.1 BANCO DE DADOS	20
5.1.2 API	21
5.1.2.1 CHAMADAS DE API PARA A RETROAÇÃO NA ÁREA DO TUTOR	21
5.1.2.2 CHAMADAS DE API PARA A RETROAÇÃO NA ÁREA DO ESTUDANTE	23

5.2 LINHA DO TEMPO . . . . .	24
5.2.1 BANCO DE DADOS . . . . .	25
5.2.2 API . . . . .	25
5.2.2.1 CHAMADAS DE API PARA A LINHA DO TEMPO . . . . .	25
<b>6 – RESULTADOS PARCIAIS . . . . .</b>	<b>27</b>
<b>7 – CONSIDERAÇÕES FINAIS . . . . .</b>	<b>28</b>
<b>8 – PLANEJAMENTO DO TRABALHO . . . . .</b>	<b>29</b>
<b>Referências . . . . .</b>	<b>30</b>
<b>Anexos . . . . .</b>	<b>33</b>
<b>ANEXO A – MODELAGEM DA ÁREA DE RETROAÇÃO À ERROS NA FARMA . . . . .</b>	<b>34</b>
<b>ANEXO B – MODELAGEM COMPLETA DA FARMA . . . . .</b>	<b>35</b>



## 1 INTRODUÇÃO

Normalmente em uma aula de matemática, física ou química a interação dos aprendizes com o tutor e o conteúdo é menor que em outras matérias, e isso reflete diretamente na evolução e aprendizado. Para resolver essa falta de contato surgiram algumas ferramentas, dentre elas as ferramentas digitais com o uso de Objetos de Aprendizagem (OAs), onde por meio da computação os aprendizes podem [re]ver e se aprofundar sempre que tenham alguma dúvida ou que achar necessário. Segundo [Carneiro e Silveira \(2014\)](#) os OAs podem ter diversos formatos e formas de uso, pois ainda não há uma definição concreta sobre elas, porém uma de suas características é fomentar o aprendiz a buscar novos conhecimentos e aprimorar a aprendizagem.

A Ferramenta de Autoria para a Remediação de erros com Mobilidade na Aprendizagem (FARMA) faz uso de OAs para auxiliar no ensino. Ela foi construída com foco em matemática, mas pode ser aplicada em outras matérias que façam uso de expressões aritméticas e/ou algébricas. Na FARMA os OAs são criadas seguindo quatro passos: são eles a criação de uma OA, a descrição de uma introdução, a criação de exercícios e a criação de dicas. A Ferramenta tem como um de seus principais focos a retroação a erros, que nada mais é que um histórico de todas as respostas que um aprendiz inseriu, sendo ela correta ou não. Através da retroação a erros o aprendiz e o tutor pode voltar em qualquer uma das respostas por ele inserida e visualizar todo o contexto daquele dado momento em que a resposta foi criada ([MARCZAL; DIRENE, 2012](#)).

A FARMA encontra-se em uma fase de remodelagem, na versão atual existem alguns gargalos possibilitando pontos de melhoria. No momento na nova versão ainda não houve a implantação de algumas funcionalidades, entre elas as de retroação à erros e uma linha do tempo das interações do aprendiz com a ferramenta. Porém, existe um estudo realizado por [Correia \(2017\)](#) onde o banco de dados já foi pensado e documentado para a retroação a erros, o mesmo pode ser visualizado no Anexo A, mas não foi implementado junto a ferramenta. No Anexo B é possível verificar a visão geral da modelagem criada por [Correia \(2017\)](#).

Sem a funcionalidade de retroação a erros a FARMA perde grande valor, pois é nela que se encontram os dados analíticos que agregam recursos para a tomada de decisão na ferramenta, e que a colocam em outro patamar. Isso a torna diferente de outras ferramentas que fazem somente o uso de OAs para auxiliar no ensino e aprendizado.

Para esse projeto a codificação será realizada em forma de uma API *RESTFULL*, pois a remodelagem em andamento já utiliza essa maneira para fornecer os dados ao *framework* de *font-end*, desta forma será mantido o mesmo padrão. O segundo motivo para esse projeto trabalhar com uma API *RESTFULL* é a modularidade, visto que ao utilizar uma API duas equipes podem desenvolver separadamente, uma no *front-end* e outra no *back-end* de uma mesma funcionalidade, sem que exista nenhuma ligação.

Para realizar a implantação da retroação a erros presente no modelo de banco de dados criado por [Correia \(2017\)](#) podem existir vários pontos críticos, tais como: conhecer amplamente o sistema de funcionamento da ferramenta em seu modelo anterior e atual, conhecer e manter comunicação com todas as pessoas envolvidas na atual remodelagem da ferramenta, estudar e analisar a viabilidade e também a forma de implantação, pois o mesmo contém funcionalidades mais complexas como funções e gatilhos diretamente no banco de dados.

Portanto esse projeto tem como objetivo principal realizar o estudo e implementação da retroação a erros presente no modelo de banco de dados, a criação de um modelo de banco de dados para a linha do tempo, a criação de uma API para a retroação a erros onde os dados possam ser salvos e encontrados assim possibilitando o desenvolvimento futuro das telas na FARMA.

Como resultado parcial até o momento já estão definidas as chamadas da API, tanto para retroação a erros como para a linha do tempo. A API de retroação é a mais complexa por ter vários pontos de uso em toda aplicação, fato pelo qual foram levantadas trinta e três chamadas distintas.

Outro resultado parcial foi para a área de linha do tempo, onde foi definido que o armazenamento de suas informações irá ficar em um banco de dados separado. Devido a grande quantidade de informações da linha do tempo o banco de dados utilizado para gerenciar será um banco para series temporais.

## 1.1 CONTEXTO DO PROJETO

A ferramenta FARMA possui grande aplicabilidade nas áreas de exatas, mais especificamente em matemática, física e química. Sua primeira versão é totalmente funcional, porém foram encontrados pontos importantes de melhoria. Alguns alunos da Universidade Tecnológica Federal do Paraná (UTFPR) Câmpus Guarapuava vem aplicando melhorias na ferramenta em seus trabalhos de conclusão de curso, criando uma nova versão. Neste mesmo passo segue esse projeto, atuando como parte importante para que essa nova versão da ferramenta possa ser viável utilização.

No processo de melhorias fizeram parte do projeto: [Santos \(2018\)](#) realizando o projeto de uma nova interface, [Goudinho \(2019\)](#) propondo um módulo de estatística, [Ramos \(2019\)](#) realizando o projeto para a visualização de um OA, [Lara \(2017\)](#) criando as aplicações de design de interface e [Correia \(2017\)](#) criando a modelagem do banco de dados atualizada.

Com o intuito de dar continuidade ao desenvolvimento da FARMA, esse projeto vem para preencher a lacuna faltante de retroação a erros e a linha do tempo. A funcionalidade de retroação a erros nesse projeto será por meio da criação de uma API de consulta, já para a linha do tempo será possível buscar e inserir dados, também por meio de uma API. A nova versão da FARMA no momento de escrita desse projeto está em uma fase onde não se é possível disponibiliza-la para uso. Esse trabalho inclusive é parte fundamental para que as funcionalidades de retroação à erros e linha do tempo possam funcionar.

## 1.2 DEFINIÇÃO DO PROBLEMA

### 1.2.1 RETROAÇÃO À ERROS

A retroação à erros é um histórico de todas as respostas incluindo o contexto em que um aprendiz realizou em sua jornada para alcançar o objetivo proposto pelo tutor. Nesse modelo o tutor pode a cada resposta correta ou principalmente nas incorretas reformular o OA e também a forma de como o conteúdo é ministrado em sala de aula.

O erro também é uma fonte de aprendizado, com isso retroação à erros é o foco principal da ferramenta FARMA, com ela é possível tanto o aprendiz como o tutor avaliarem as mudanças por eles realizadas na forma de estudo e ensino com a qual levou o aprendiz a chegar ao conhecimento desejado.

Na versão anterior da FARMA a forma como os dados de retroação eram armazenados vinham enfrentando problemas de processamento e redundância no banco de dados.

Na remodelagem do banco de dados a retroação à erros foi uma das áreas que mais sofreu mudanças, e segundo [Correia \(2017\)](#) foi um dos motivos principais para que tal mudança no banco de dados fosse proposta. Esse novo modelo de banco de dados para a retroação a erros ainda não foi implementada, o presente trabalho tem como um de seus objetivos realizar essa implantação.

### 1.2.2 LINHA DO TEMPO

A linha do tempo guardará tanto os dados da retroação a erros como cada passo que o aprendiz der na ferramenta. Com isso o tutor pode ter mais informações para criar ações que ajudem o desenvolvimento do aprendizado especificamente para um aprendiz ou generalizando para um turma.

Segundo [Paterlini \(2016\)](#) uma linha do tempo é uma exibição gráfica ou em forma de texto apresentando eventos em ordem cronológica, e é a mais utilizada como técnica para interagir com o tempo. Ela também permite ao usuário explorar as relações entre os eventos históricos.

Com o uso da linha do tempo a navegação entre dados na ferramenta será simples tanto para o tutor como para o aprendiz, quando em dado momento um dos dois julgar necessário retroceder, basta localizar visualmente em que ponto deseja retroceder e clicar sobre esse registro e o mesmo será levado ao exato momento em que os dados foram salvos na ferramenta.

No momento a ferramenta não dispõe de uma implementação nem modelo de banco de dados relacionado a linha do tempo. Esse trabalho tem como um de seus objetivos realizar a modelagem do banco de dados e a implantação de uma API para retornar os dados, para que no futuro possam ser implementada a linha do tempo na FARMA.

## 2 OBJETIVOS

### 2.1 OBJETIVO GERAL

Criar as APIs necessárias, na FARMA, para escrita e consulta de informações relacionadas a linha do tempo, assim como para consulta dos dados da retroação erros.

### 2.2 OBJETIVOS ESPECÍFICOS

1. Estudar e adaptar a área de retroação a erros do modelo de banco de dados criado por (CORREIA, 2017) para que a codificação do mesmo possa ser realizada;
2. Implementar o modelo de banco de dados referente a retroação a erros na aplicação;
3. Criar uma API na ferramenta FARMA para a leitura de dados de retroação a erros;
4. Criar um modelo de banco de dados para a funcionalidade de linha do tempo;
5. Criar uma API na ferramenta FARMA para escrita de dados da linha do tempo, baseado no modelo de dados descrito no objetivo anterior.
6. Criar uma API na ferramenta FARMA para leitura de dados da linha do tempo, baseado no modelo de dados descrito no objetivo anterior.

### 3 FARMA

Neste capítulo será apresentado de maneira detalhada a ferramenta FARMA, bem como o seu principal objetivo, que é o uso do erro como parte importante no ensino.

#### 3.1 ERROS

No contexto do aprendizado, o erro geralmente traz à mente o sentimento de fracasso. Para quem cometeu o erro remete à falta de inteligência e dedicação. Porém, em grande parte dos casos, não é verdade. A prerrogativa sobre o erro ser algo mau é antiga. Atualmente é possível extrair muitas informações sobre o erro. Métricas, remodelagem e conteúdos diferenciados são informações que podem ser facilmente analisadas sobre os erros (MOREIRA, 2005).

O erro não é um ponto final mas sim parte do aprendizado. A imagem negativa do erro influencia diretamente no aprendizado, fazendo com que o aprendiz perca o ânimo pelos estudos. Após o erro vem a cabeça do aprendiz a falta de inteligência, de aptidão, de dedicação, entre outros, quando na verdade existe um enorme número de fatores que podem ter colaborado para esse resultado. O foco não pode pairar sobre o erro, mas sobre como reverter esse cenário e encontrar formas da aprendizagem acontecer (KUTZKE, 2015).

Cannon e Edmondson (2005) nos mostra a visão de que ao deixar de lado o viés negativo do erro, é possível eliminar a desmotivação atrelada a um pensamento de falta de dom natural, e juntamente com isso desmistificar a ideologia de que o acerto é associado ao talento. Assim podendo mostrar que o aprendizado é algo que se constrói gradualmente por meio de dedicação, estudos e a prática.

No contexto da FARMA o erro é de grande importância para a plataforma, e tem seu lugar de destaque em todo o processo de aprendizado, sobre ele são levantadas métricas de ensino, disponibilizadas ao tutor para avaliar a evolução de toda a jornada de um aluno e de uma turma. A plataforma já foi estruturada para que em casos de erros, possibilitar ao tutor acrescentar dicas e passos que ajudem o aluno a superá-los.

A aprendizagem pelo erro é algo natural, realizado continuamente no cotidiano. O ser que aprende é um perceptor, ou seja, um sujeito que percebe e representa o que lhe está sendo ensinado Moreira (2005). Utilizando dessa prerrogativa na FARMA, a retroação a erros permite ao aprendiz retornar a uma resposta errada e analisar todo o contexto em que o OA se encontrava, facilitando assim assimilar toda a informação e a tomada de medidas para remediar o erro e alcançar o aprendizado.

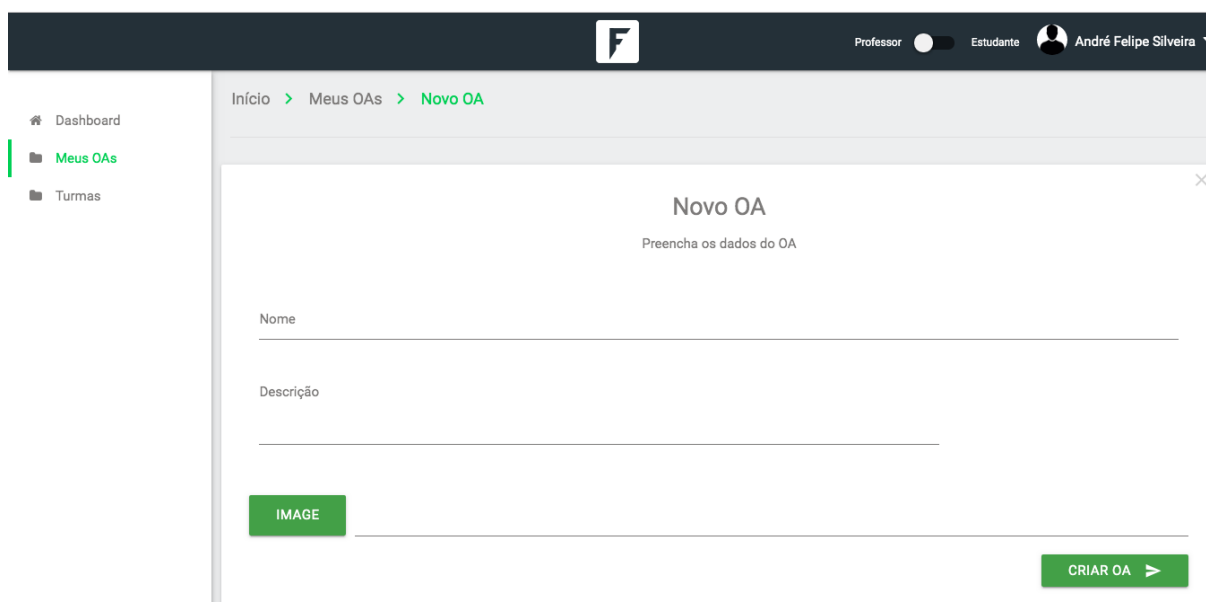
Este projeto tem ligação direta com erros no processo de aprendizagem, pois tem seu foco central na disponibilidade dos dados para consulta da retroação à erros através de uma API, porém o consumo dos dados da API e a criação de telas não serão realizados neste trabalho.

## 3.2 FARMA

A FARMA se apresenta como uma ferramenta web para auxiliar o ensino e aprendizagem por meio de OAs, para matérias de matemática, química, física e quaisquer outras que utilizem expressões aritméticas e/ou algébricas. A ferramenta ajuda a produção e disseminação de conteúdos.

Todo baseado em OAs, o ensino pela ferramenta FARMA tende a ser simples tanto para o tutor como para o aprendiz. A criação de uma OA segue três simples etapas, que são: a própria criação da OA (Figura 1), a descrição de uma introdução (Figura 2) e por último a criação de exercícios de fixação (Figura 3), além dessas, inserido na criação de um exercício existe mais duas etapas que são a definição de passos de solução e a inserção de dicas para cada um dos passos. Após a base da OA criada o tutor pode moldá-la segundo a sua necessidade de aprendizado. A visualização da tela completa de um OA pode ser encontrada na Figura 4 (MARCZAL; DIRENE, 2012). Na Figura 5 é possível visualizar o resultado da criação dos passos e dicas em um exercício sendo apresentados a um Aprendiz. Para um melhor entendimento do fluxo de processos realizado pela FARMA é possível acompanhar a separação e comunicação de todos os módulos, juntamente com a interação tanto do aprendiz como do tutor para com a ferramenta na Figura 6

Figura 1 – Tela de criação do OA - FARMA

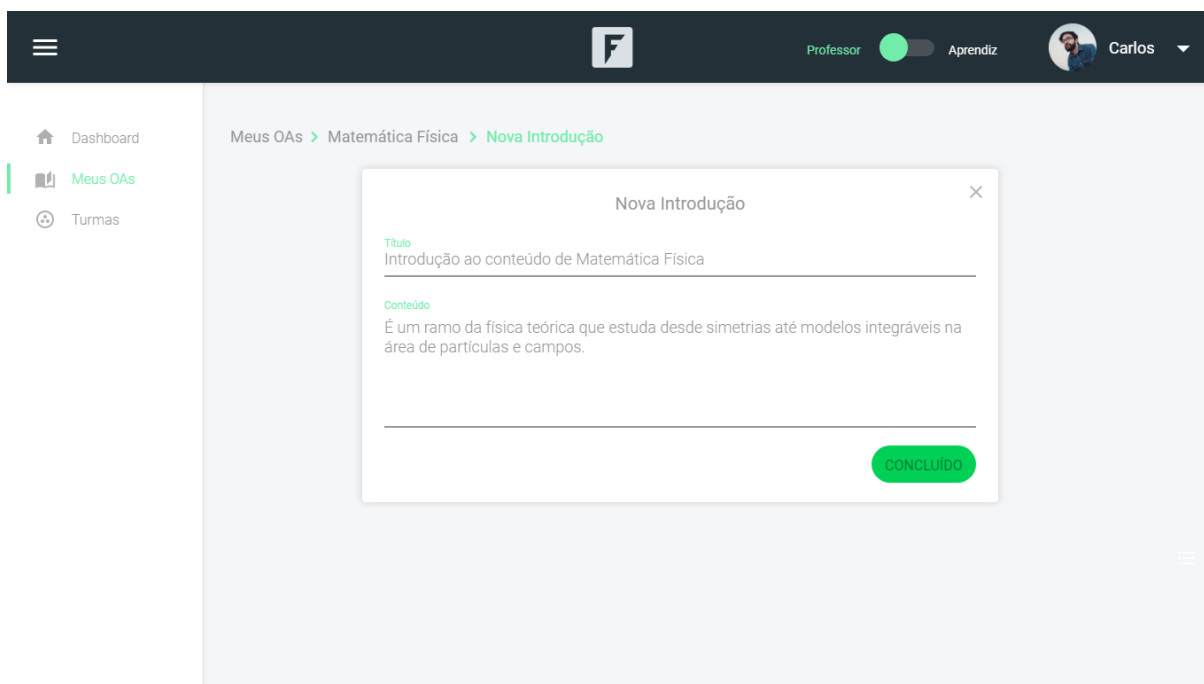


Fonte: O Autor.

Em cada resposta o aprendiz se depara com um teclado virtual (Figura 7) onde o mesmo não pode inserir respostas fora do contexto, assim sendo possível para ferramenta apresentar um feedback em tempo real para que o aprendiz possa responder da maneira que o tutor tenha planejado.

Para [Pasinoto \(2008\)](#) tão importante quanto o aprendizado são os erros, pois por eles é possível se medir a evolução, se o caminho realmente pode e levará o aprendiz ao sucesso do aprendizado. Seguindo esses preceitos a FARMA utiliza do armazenamento e apresentação dos erros em um cenário completamente congelado no tempo, isso significa que todos os dados que foram fornecidos ao aprendiz no momento que uma resposta foi inserida por ele estarão sempre salvos para que o mesmo possa consultar quando necessário.

Figura 2 – Tela de criação da Introdução - FARMA



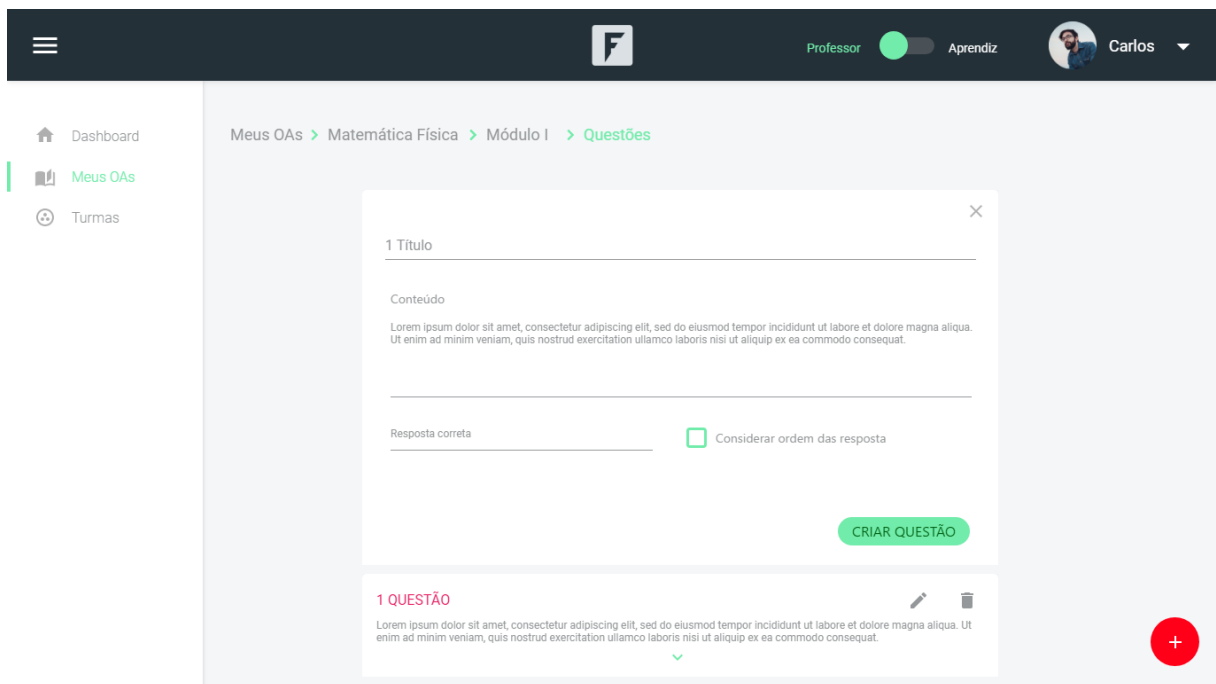
Fonte: [Lara \(2017\)](#)

Ainda sobre tentativas e erros, além da retroação a FARMA utiliza de outra abordagem que é a remediação a erros, esta funciona com o uso de dicas inseridas em passos previamente cadastrados (Figura 5), as dicas são cadastradas pelo tutor para serem apresentadas ao aprendiz após algumas tentativas sem sucesso, o ajudando a evoluir seu pensamento para alcançar a resposta correta.

A ferramenta permite a criação de turmas e com isso o tutor pode separar seus aprendizes por nível de conhecimento, com os dados fornecidos pela ferramenta através as OAs e da retroação a erros o tutor pode acompanhar e realizar suas medições sobre cada tema apresentado, assim conseguindo visualizar o desempenho de uma turma ou isoladamente de cada aprendiz.

Com todas as informações apresentadas pela ferramenta o tutor pode realizar ações baseadas no andamento de cada turma ou de cada aprendiz, dando uma atenção especial a quem julgar necessário e modificando a forma de ensino como forma de ajudar os aprendizes que estão com dificuldades. Para o aprendiz é uma maneira simples de não desistir até conseguir

Figura 3 – Tela de criação de Exercícios - FARMA



Fonte: Lara (2017)

Figura 4 – Visualização do OA - FARMA



Fonte: Lara (2017)

assimilar o assunto proposto, pois ele é livre para controlar seu tempo de estudo e também para ir e voltar em suas respostas realizando uma autoavaliação de seu aprendizado.

Resumindo, a FARMA tem por principal objetivo estender o ensinamento de conteúdos



Figura 5 – Visualização de passos e dicas - FARMA

Teorema de Pitágoras

Meu Rendimento

Carlos

Exercício 1

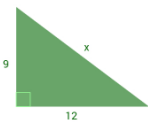
### Teorema de Pitágoras

Com o conteúdo apresentado na Introdução 1, resolva os seguintes passos.

Passo 1

#### Triângulo retângulo

Calcule o valor do segmento desconhecido no triângulo retângulo a seguir.



Clique aqui para responder

Dicas

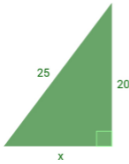
DICA

$$a^2 + b^2 = c^2$$

Passo 2

#### Encontre o valor do cateto

Calcule o valor do cateto no triângulo retângulo abaixo:



Clique aqui para responder

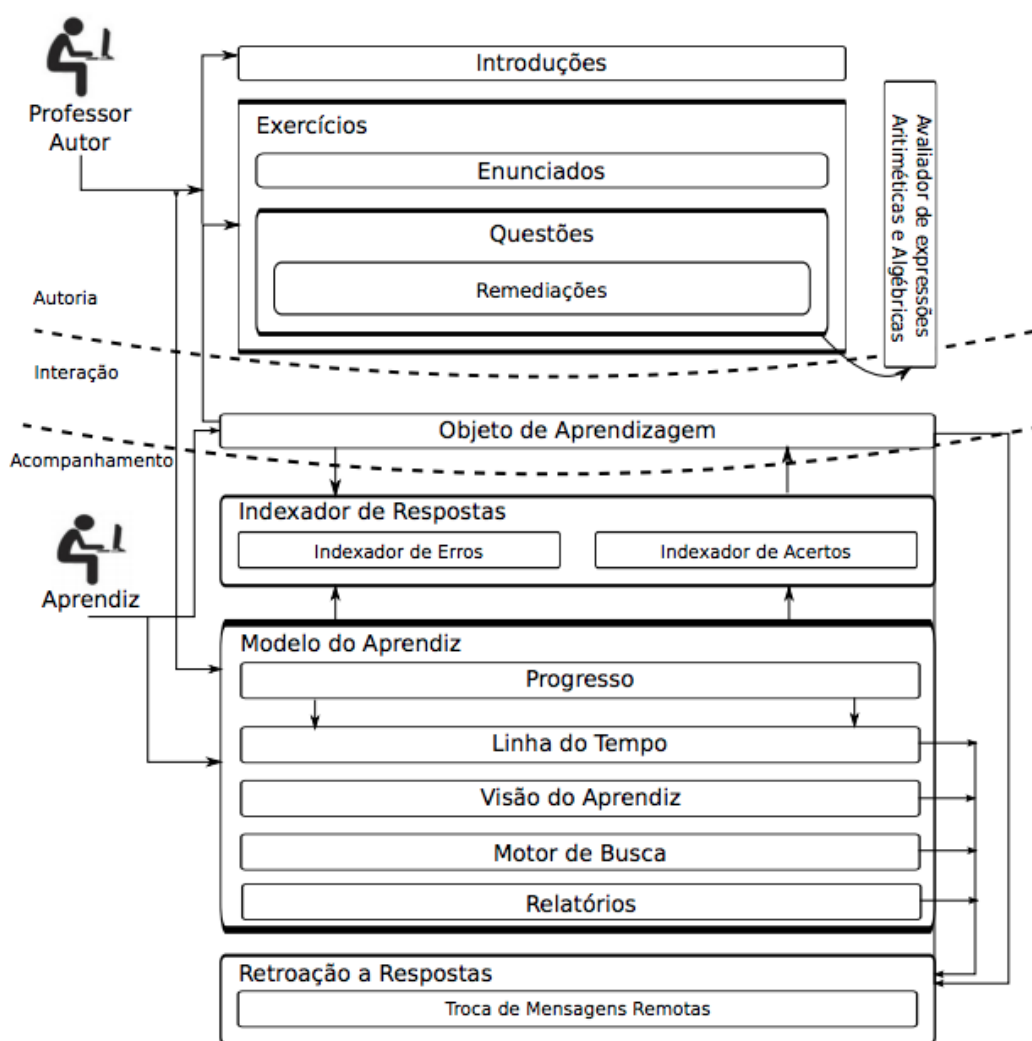
Fonte: [Lara \(2017\)](#)

de maneira simples e direta através de uma ferramenta web, armazenando, e trabalhando sobre os dados para melhorar o resultado final de um aprendizado.

### 3.3 ARQUITETURA FUNCIONALISTA DA FARMA

Nesta Seção é apresentada a arquitetura funcionalista da ferramenta FARMA, sendo um complemento ao Capítulo 3. A arquitetura funcionalista demonstra a maneira como a ferramenta foi criada para trabalhar com os fundamentos de aprendizagem. Vale ressaltar que esse projeto não irá de nenhuma maneira alterar a arquitetura do software, apenas seguir a já existente. A arquitetura é dividida em três estágios principais, que são: autoria, interação e acompanhamento (MARCZAL, 2014). Esses estágios são representados na Figura 6.

Figura 6 – Arquitetura funcionalista da FARMA



Fonte: Marczal (2014)

Segundo Marczal (2014) a ferramenta FARMA foi criada com seu uso voltado a web, onde a mesma pode ser acessada por meio de computadores ou dispositivos móveis, seu código está disponível como código livre sobre a licença GPL (Licença Pública Geral), e as tecnologias utilizadas em sua criação foram:

Linguagem Ruby (Detalhada na seção 4.2);

*Framework* Ruby on Rails (Detalhado na seção 4.3);  
Persistência de dados Mongoddb. <sup>1</sup>

### 3.3.1 AUTORIA

O módulo de autoria fornece todos os meios necessários para a criação de um OA matemático, porém como a ferramenta trabalha com expressões aritméticas ou algébricas ela pode ser utilizada para criar OAs para outras áreas de exatas. Esse módulo permite a criação tanto da parte teórica com a criação de uma introdução, como da parte prática que são os exercícios.

Na autoria existem subdivisões para construir introduções destinada a parte teórica de um OA, criar exercícios para a parte prática. O exercício, por sua vez também é subdividido em construtor de questões e dicas.

### 3.3.2 INTERAÇÃO

O módulo de interação como o próprio nome diz, é a interface de interação entre o aprendiz com um OA criado pelo professor-autor. A função do módulo de interação é juntar a introdução e os exercícios criados pelo tutor compondo assim o OA para ser apresentado ao aprendiz.

O OA é composto por páginas, e é dividido em páginas de introdução e de exercícios, assim a ferramenta disponibiliza um paginador para melhor organizar esses conteúdos. As informações são ordenadas pela mesma sequência utilizada pelo professor-autor no momento de criação do OA.

Para o aprendiz resolver as questões propostas nos exercícios a ferramenta disponibiliza um teclado virtual, neste teclado são apresentadas as mesmas variáveis utilizadas pelo professor-autor no momento de criação dos exercícios, evitando que o aprendiz tenha a oportunidade de criar expressões com erros de sintaxe. A Figura 7 retrata o design criado para o teclado virtual a ser implementado na ferramenta.

### 3.3.3 ACOMPANHAMENTO

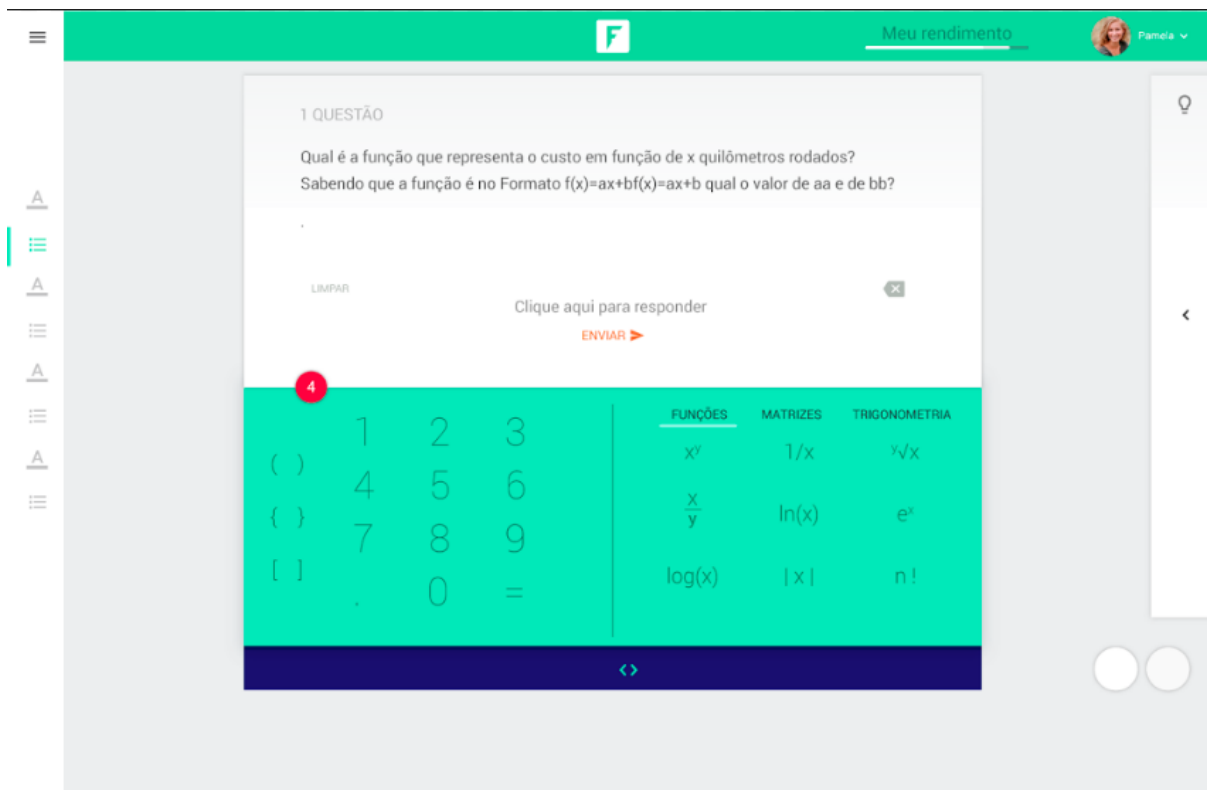
Segundo Marczal (2014) o módulo de acompanhamento é o mais importante da ferramenta FARMA, pois por meio dele o tutor pode acompanhar a evolução dos aprendizes e realizar uma avaliação formativa do aprendizado, podendo essa evolução ser calculada tanto a curto como a longo prazo.

O principal recurso para realizar as avaliações é o mecanismo de retroação a erros, por meio dele o tutor pode navegar pelos mesmos passos que o aprendiz incluindo o mesmo contexto pelo qual o aprendiz teve contato. Para o aprendiz o módulo de acompanhamento

---

<sup>1</sup>A versão atual da ferramenta deixou de utilizar como SGBD o Mongoddb e passou a utilizar o PostgreSQL.

Figura 7 – Teclado virtual - FARMA



Fonte: [Lara \(2017\)](#)

pode ser utilizado para navegar e até mesmo repetir etapas dos exercícios para entendimento da causa de seu erro.

Esse módulo é subdividido em três partes que são eles: indexador de respostas, retroação às respostas e modelo do aprendiz.

### 3.3.3.1 INDEXADOR DE RESPOSTAS

O módulo indexador de respostas é responsável por salvar todas as respostas enviadas pelo aprendiz, independente de estarem corretas ou erradas. Cada vez que uma resposta é salva, juntamente são salvas todas as informações do contexto da questão, como por exemplo:

- A questão juntamente com a resposta;
- As remediações já recebidas para a resposta;
- A data e hora em que a resposta foi enviada;
- O número de tentativas;
- O exercício ao qual a questão pertence;
- Todas as questões do exercício, com suas respostas e remediações já enviadas ao aprendiz corrente;
- Informações relevantes do OA como título e descrição.

Ao enviar uma resposta a ferramenta compara a informação do aprendiz com a

informação salva pelo tutor no momento da criação de uma questão, para isso é utilizado um módulo de comparação de expressões algébricas e aritméticas. No momento da comparação é realizado uma cópia de todo o contexto da resposta, assim caso ocorra uma alteração no OA futuramente não afetará o contexto de uma resposta já salva.

### 3.3.3.2 RETROAÇÃO ÀS RESPOSTAS

Este módulo faz uso do indexador de respostas para, por meio da recuperação dos dados salvos, permitir ao tutor e ao aprendiz retroagir ao exato contexto de uma resposta.

Ao realizar a retroação, a ferramenta remonta todo o contexto de uma questão, simulando assim a mesma visão que o aprendiz teve ao salvar a resposta. Dessa maneira o tutor pode compreender melhor os motivos que levaram o aprendiz a salvar aquela resposta. Com essas informações o tutor pode enviar um *feedback* ao aprendiz através do módulo de troca de mensagens remotas.

Além de restabelecer a visão do OA pelo aprendiz, esse módulo permite a interação com o OA da mesma maneira que antes de salvar a resposta, assim o aprendiz pode realizar novamente o envio de uma resposta e tentar perceber o motivo pelo qual o erro aconteceu das respostas passadas.

A retroação a respostas pode ser acessada por várias áreas da ferramentas, as principais são: via próprio OA onde a resposta foi submetida, pela linha do tempo do aprendiz e pelo motor de buscas. Essas áreas são acessíveis tanto pelo tutor como pelo aprendiz, porém cada qual com seu devido privilégio.

### 3.3.3.3 MODELO DO APRENDIZ

O modelo do aprendiz, ainda não foi inserido na ferramenta trata-se de um trabalho futuro que tem por objetivo a utilização da grande quantidade de dados salvos pelo módulo indexador de respostas. O projeto de [Goudinho \(2019\)](#) vem para criar parte do modelo do aprendiz. Os dados salvos pelo indexador podem ser convertidos em dados analíticos, e servirem para a tomada de decisão pelo tutor, para melhorar a qualidade do ensino e de seus OA.

Por meio de um motor de buscas ou da linha do tempo os dados armazenados pelo indexador podem apresentar ao tutor todos os passos dados na ferramenta por um aprendiz, além de fornecer facilmente a retroação para o contexto de uma dada resposta salva. O tutor também terá acesso a relatórios dos aprendizes baseados nos OAs, onde será possível visualizar em forma de tabelas as interações dos aprendizes, podendo ser agrupados por: nome do aprendiz, exercícios, questão, resposta correta ou incorreta e número de tentativas.

## 4 METODOLOGIA DE DESENVOLVIMENTO

Neste capítulo são apresentadas as tecnologias e ferramentas que serão utilizadas para o desenvolvimento desse projeto.

### 4.1 KANBAN

Criado na década de 60 pelos engenheiros da *Toyota Motors Cia* para controlar a fabricação de automóveis, o Kanban tem como objetivo tornar simples e rápidas a visualização do andamento de tarefas. O sistema Kanban funciona através do uso de cartões de sinalização que falam sobre as tarefas existentes. Entre as principais características do Kanban segundo [Leite et al. \(2004\)](#) estão:

- Melhoria total e continua do sistema de desenvolvimento;
- Regulamento do fluxo de itens, afim de executa-lo com precisão;
- Simplificação do trabalho gerencial;
- Informação sempre disponível de forma organizada, simples e rápida.

O Kanban foi criado baseado no processo de fabricação em lotes ([LEITE et al., 2004](#)). Em 2007 o método Kanban teve um considerável aumento de sua utilização para controle de processos de software, pois os resultados apresentados por Rick Garber e David J. e publicados nas conferencias "*Lean New Product Development*" e "*Agile 2007*" criaram uma grande repercussão e este método foi adotado por diversas equipes de desenvolvimento de software.

O significado do termo japonês Kanban é sinal visual. A grande característica deste método é evidenciar os problemas que existem em todo o processo. Por ser a menos prescritiva, várias equipes de software o adotam. Segundo [Silva, Santos e Neto \(2010\)](#) o Kanban tem apenas 3 prescrições:

- Visualize o fluxo de trabalho atual;
- Limite o fluxo de trabalho;
- Acompanhe e gerencie o fluxo de trabalho.

Com o Kanban é possível adotar e combinar diversas ferramentas de diversos métodos até se obter o processo adequado. Ele estimula a melhoria continua do processo de forma a tornar possível respostas rápidas ao cliente.

Neste projeto o método Kanban será utilizado para acompanhar todo o processo de desenvolvimento onde os requisitos do sistema irão se tornar vários cartões de tarefas, com auxilio da ferramenta Trello<sup>1</sup>, que neste caso representará um quadro digital contento as areas: elegível a entrar em execução , em andamento e concluído, onde serão alocados os cartões com as tarefas.

---

<sup>1</sup>Trello: <https://trello.com/>

## 4.2 LINGUAGEM DE PROGRAMAÇÃO RUBY

Lançada em 1995 por Yukihiro Matsumoto Ruby é uma linguagem interpretada, multi plataforma e com muitas características em comum com outras linguagens. Baseada em Smalltalk<sup>2</sup>, Python<sup>3</sup>, Perl<sup>4</sup> e outras, porém com uma orientação a objetos muito mais completa (COLLINGBOURNE, 2006). Ruby é uma linguagem dinâmica e reflexiva, ela suporta vários paradigmas da computação como, funcionais, orientados a objetos e imperativos (PENCHEL; CANTARELA, 2015).

Ruby tem um sistema de tipagem dinâmica e forte e gerenciamento da memória automática (PENCHEL; CANTARELA, 2015). Ela é uma linguagem qualificada como simples, direto ao ponto, extensível e portátil (BRODBECK, 2007). Foi desenvolvida com diversos recursos para processar arquivos de texto e realizar tarefas de gerenciamento de sistema. Todo dado da linguagem Ruby é um objeto, e é por esse motivo que ela é considerada puramente orientada a objetos (BRODBECK, 2007).

Abaixo serão listados alguns recursos disponíveis na linguagem Ruby (BRODBECK, 2007):

- Orientação a objetos no Ruby foi criada para ser completa e editável, ou seja é possível adicionar métodos em uma classe, ou em uma instancia mesmo durante o *runtime*<sup>5</sup>;
- Herança única, pois utilizando o conceito de módulos, obtém-se uma forma mais limpa e simples de herança múltipla;
- Possui blocos de sintaxe delimitados por chaves "*f...g*" ou por instruções como "*do ... end*";
- Possui um coletor de lixo totalmente funcional, pois atua em todos os objetos do Ruby;
- Não é obrigatório o uso de um ponto e vírgula para encerrar uma instrução;
- Possui um gerenciador de pacotes o *RubyGems*, e por meio dele é possível instalar de maneira simples diversos *plugins*.

Além dos recursos já citados, a linguagem é extremamente produtiva e simples, com pouco tempo e linhas de código é possível ter um CRUD Criar, ler, editar e apagar (do inglês Create, Read, Update e Delete)<sup>6</sup> totalmente funcional. Nos últimos anos a comunidade Ruby cresceu muito e também o número de *gems*. A linguagem possui um *framework* MVC Modelo, visão e controlador (do inglês Model, View and Controller) que permite a construção de aplicações com extrema rapidez (SOUZA, 2012).

Na nova modelagem da FARMA, todo o código no lado servidor foi e será escrito em Ruby. E para acelerar e facilitar o processo de desenvolvimento será utilizado o *framework* Rails (Detalhado na subseção 4.3).

---

<sup>2</sup><http://www.smalltalk.org/>

<sup>3</sup>Python: <http://python.org.br>

<sup>4</sup>Perl: <https://www.perl.org/>

<sup>5</sup>Runtime: período em que um programa de computador permanece em execução.

<sup>6</sup>CRUD: As quatro operações básicas para se utilizar em base de dados relacionais

### 4.3 FRAMEWORK RAILS

Criado em 2003 por David Heinemeier Hansson o Rails é um *framework open source* para a linguagem Ruby. Após o seu lançamento ele ficou muito famoso e levantou junto com ele a linguagem Ruby, que anteriormente só era conhecida no Japão e em poucos lugares dos Estados Unidos. Rails trouxe com seu lançamento uma mudança no desenvolvimento web que naquele momento era cansativo e vagaroso, os *frameworks* eram complexos o que gerava aplicações difíceis de dar manutenção e algumas vezes de baixa qualidade (SOUZA, 2012).

Dentre os pilares da agilidade do *framework* Rails estão os conceitos de *Don't Repeat Yourself* que fala sobre não repetir trechos de códigos mas sim modularizá-los, ou seja, deve-se escrever trechos de códigos que serão reutilizados evitando assim a cópia e cola de código. Outro conceito é o *Convention over Configuration* que diz ser necessário fazer uma configuração somente em casos especiais. Ao utilizar os padrões fornecidos pelo Rails só é preciso configurar o que estiver fora do padrão, assim diminuindo a quantidade de escrita para a maioria das configurações (CAELUM, 2013).

O *framework* possui uma arquitetura de desenvolvimento chamada MVC que possui um conceito de separar a aplicação em três camadas. A separação do MVC ocorre entre modelos que é a área onde se encontra todas as regras de negócio da aplicação e as ações mais próximas ao banco de dados. A visão é onde se concentram todos os códigos que ficam mais próximos do usuário, ou seja, a parte em que são estruturadas as apresentações de telas. A última área é responsável por controlar as rotas da aplicação definir qual área será responsável por responder uma requisição (CAELUM, 2013).

Com Rails é possível criar uma aplicação, como por exemplo um blog, que tenha acesso ao banco de dados e validações em menos de dez minutos<sup>7</sup> e ainda conter um código bem organizado. Toda a eficiência do Rails está separada em diversos recursos e muitos deles foram criados a partir de outros. Abaixo estão listados alguns recursos do *framework* (BRODBECK, 2007):

Metaprogramação é a ação de um próprio programa criar outro programa. Ruby é a linguagem referência neste momento para a utilização desta técnica e o *framework* Rails há utiliza de uma maneira muito bem explorada;

Convenção sobre configuração é utilizado pelo Rails, assim ele localiza tudo que necessita e executa muitas tarefas automaticamente, sem ser necessário escrever linhas de código para configurar a aplicação;

Rails disponibiliza três ambientes em uma única aplicação, uma para desenvolvimento, outra para testes e uma para produção, cada uma delas possui particularidades associadas ao banco de dados para cada ambiente.

Na nova modelagem da FARMA é utilizado Ruby com o *framework* Rails que darão agilidade no processo de desenvolvimento da API e manterão todos os arquivos bem organizados,

---

<sup>7</sup><https://www.railstutorial.org/>



e isso facilitará o desenvolvimento modular. Porém uma aplicação somente com linguagem de programação não pode realizar o armazenamento de dados e para isso esse projeto utiliza um banco de dados e para gerencia-lo o sistema gerenciador de banco de dados (SGBD) em uso é o PostgreSQL.

#### 4.4 POSTGRESQL

PostgreSQL é um gerenciador de banco de dados relacional utilizado para armazenar dados de aplicações de informática (MILANI, 2008). O SGBD PostgreSQL foi criado na Universidade da Califórnia, em Berkeley, pelo departamento de ciência da computação em 1987 em um projeto chamado Postgre, seu primeiro nome foi Ingres (DOMINICO, 2013). Em 1989 foi lançada a primeira versão estável e nos anos subsequentes várias versões com correções de erros e falhas. Em 1991 seu código foi adquirido por uma empresa chamada *Illustra Information Technologies*, hoje pertence a IBM<sup>8</sup>.

Um recurso interessante do PostgreSQL é sua estabilidade pois ele foi projetado para trabalhar por tempo indefinido. Dessa forma o PostgreSQL tem sido vastamente utilizado na internet, seja em aplicações simples como sites, passando por portais e lojas virtuais e até em aplicações mais complexas (MILANI, 2008).

Existem diversas bibliotecas para várias linguagem de programação como C, Java, Ruby, PHP. Entre sistemas operacionais o PostgreSQL é extremamente portátil e está disponível tanto em Linux, Unix, Max OS X Server, Windows (MILANI, 2008). segundo Milani (2008) PostgreSQL oferece diversos recursos entre eles:

- Suporte a operações ACID (Atomicidade, Consistência, Isolamento e Durabilidade);
- Replicação;
- Clusterização;
- Multithreads;
- Segurança SSL e criptografia;
- Linguagem SQL;
- Incorporável em aplicações gratuitamente;
- Capacidade de armazenamento.

Na nova modelagem da FARMA o SGBD utilizado para armazenar os dados de maneira segura é o PostgreSQL, para esse projeto será escrito códigos SQL na própria linguagem Ruby com o framework Rails através de migrations, onde é responsável por fazer os acessos ao banco bem como todas as operações necessárias.

#### 4.5 API RESTFULL

Saudate (2013) explica que REST (REpresentation State Transfer) surgiu em um doutorado realizado por Roy Fielding, um dos coautores do protocolo HTTP (Hyper Text

---

<sup>8</sup>IBM: <http://www.ibm.com/>

Transfer Protocol), logo o protocolo REST é baseado nos preceitos do protocolo HTTP. REST é composto por um conjunto de regras simples para implementar serviços web, porém não especifica quais tecnologias devem ser utilizadas. RESTFULL é uma variação do REST onde todos os métodos do protocolo HTTP são utilizados. As regras que devem ser seguidas, baseadas no protocolo HTTP segundo [Caelum \(2013\)](#) para RESTFULL são:

GET: Obter os dados de um recurso;

POST: Criar um novo recurso;

PUT: Substituir os dados de um determinado recurso;

PATCH: Atualizar parcialmente um determinado recurso;

DELETE: Excluir um determinado recurso;

HEAD: Similar ao GET, mas utilizado apenas para se obter os cabeçalhos de resposta, sem os dados em si;

OPTIONS: Obter quais manipulações podem ser realizadas em um determinado recurso.

É possível destacar os principais pontos de uma API RESTFULL que são: Diferentes métodos de comunicação, dentre os principais estão: POST, GET, PUT e DELETE; Utilização de HEADERS, que são cabeçalhos onde são transportadas informações como formato da requisição e autorização de acesso; Definição de arquivos como recursos, ou seja cada resposta deve ter seu próprio endereço; Uso de diferentes tipos de mídias para resposta como por exemplo JSON e XML.

Utilizando o conceito de uma API, é possível disponibilizar um único tipo de serviço para diversos tipos de aplicações consumirem seus dados.

Neste projeto serão criadas APIs RESTFULL, pois a atual versão da FARMA que encontra-se em desenvolvimento, utiliza um *framework front-end* e a comunicação ocorre exclusivamente via API com a linguagem RAILS. Essa prática de comunicação via API entre *frameworks front-end* e *back-end* acelera o desenvolvimento pois divide a aplicação em duas áreas, e desta maneira é possível com que pessoas ou grupos possam trabalhar isoladamente, cada qual com a parte que tem mais conhecimento, sem que exista uma interferência entre elas ([SAUDATE, 2013](#)).

#### 4.6 API RESTFULL COM RUBY ON RAILS

Com a ajuda do Framework Rails e fazendo uso da linguagem Ruby é possível criar de maneira simples e enxuta uma API RESTFULL, onde facilmente caso haja necessidade é possível agregar mais funções a ela.

Ruby on Rails permite de maneira simples e prática a criação de API's RESTFULL para comunicação com aplicações de front-end, isso se dá graças a uma funcionalidade que vem desde a versão 5 do framework Rails, onde é possível eliminar todo o código desnecessário e otimizar a aplicação para responder a solicitações via API.

Na FARMA já existe vários pontos na plataforma que respondem via API, nesse projeto o objetivo é criar os meios necessários para a comunicação do front-end com as funcionalidade

de Retroação a Erros e linha do tempo, para isso será utilizado o padrão de API RESTFULL já existente na plataforma.

## 5 DESENVOLVIMENTO

### 5.1 RETROAÇÃO À ERROS

Para a disponibilidade de dados da retroação à erros na API será necessário implementar primeiramente o banco de dados modelado por [Correia \(2017\)](#), isso implica em converter a linguagem SQL para a linguagem Ruby utilizada no *framework* Ruby on Rails, essa conversão está detalhada na seção 5.1.1. Assim que o banco de dados estiver criado com as devidas regras de negócio é a hora de iniciar a API para disponibilizar a recuperação das informações sobre as retoações.

#### 5.1.1 BANCO DE DADOS

O Banco de dados já modelado como descrito na introdução deste capítulo será analisado, convertido em código Ruby e se necessário sofrerá ajustes para que sua implementação possa ser realizada. A linguagem Ruby com o *framework* Ruby on Rails utiliza de migrações para ter acesso ao banco de dados e realizar ações como a criação, edição e exclusão de tabelas, colunas, procedimentos e gatilhos.

Segundo a modelagem de [Correia \(2017\)](#) todas as informações pertinentes a retroação serão armazenadas por meio de procedimentos e gatilhos no banco de dados. Quando uma resposta é inserida na ferramenta e salva no banco de dados, o próprio banco é responsável por criar a retroação referente a essa resposta, mantendo assim o código mais limpo e livre pós processamentos.

O modelo criado por [Correia \(2017\)](#), irá evitar a redundância de dados que causada lentidão na versão anterior por meio do uso de um banco relacional, os dados de um OA serão copiados para uma turma, assim quando o tutor realizar qualquer tipo de alteração em um OA não irá afetar as respostas já inseridas por um aprendiz em uma turma.

Quando uma resposta é inserida na ferramenta o banco de dados irá popular uma tabela chamada '*Answers*' que irá conter os seguintes dados:

*id*: Identificador da linha na tabela;

*response*: Valor da resposta enviada pelo aprendiz;

*correct*: Valor booleano informando se a resposta é correta ou errada;

*attempt\_number*: Número da tentativa realizada pelo aprendiz;

*user\_id*: Identificador do aprendiz;

*team\_id*: Identificador da turma ao qual o aprendiz e o OA pertence;

*step\_version\_id*: Identificador da versão do OA no momento da resposta inserida pelo aprendiz;

*last\_answers*: Um *Array* informando todas as respostas anteriores inseridas pelo aprendiz;

As tabelas abaixo sempre serão atualizadas via gatilhos e procedimentos do banco de

dados quando o tutor realiza alguma alteração no OA, desta maneira todo o contexto de uma resposta será preservada.

*exercises\_versions* responsável por armazenar as informações sobre os exercícios de um OA;

*introductions\_versions* responsável por armazenar as informações sobre a introdução de um OA;

*los\_versions* responsável por armazenar as informações sobre um OA;

*steps\_versions* responsável por armazenar as informações sobre os passos de um OA;

*tips\_versions* responsável por armazenar as informações sobre as dicas de um OA.

### 5.1.2 API

Para a API irá existir a divisão entre a área do tutor e a área do aprendiz, pois cada um tem níveis de acesso diferentes. A API de retroação a erros terá inicialmente vinte e três chamadas para a área do tutor e dez chamadas para a área do aprendiz. Essas chamadas foram criadas baseadas na versão antiga da FARMA. Caso haja necessidade no decorrer desse projeto essas chamadas podem ser alteradas, canceladas ou até mesmo novas serem criadas. As palavras na listagem a baixo que vem seguida do uso de ':'(dois pontos) fazem referência a parâmetros que devem ser enviados para a API.

#### 5.1.2.1 CHAMADAS DE API PARA A RETROAÇÃO NA ÁREA DO TUTOR

1. **`/api/teacher/retroactions/:team_id/answers`**

Esta chamada da API é responsável por retornar todas as respostas de uma determinada turma;

2. **`/api/teacher/retroactions/:team_id/answers/correct`**

Esta chamada da API é uma variação da chamada 1, serve de filtro para todas as respostas corretas de uma turma;

3. **`/api/teacher/retroactions/:team_id/answers/wrongs`**

Esta chamada da API é uma variação da chamada 1, serve de filtro para todas as respostas erradas de uma turma;

4. **`/api/teacher/retroactions/:team_id/:lo_id/answers`**

Esta chamada da API é responsável por retornar todas as respostas de uma turma em um OA específico;

5. **`/api/teacher/retroactions/:team_id/:lo_id/answers/corrects`**

Esta chamada da API é uma variação da chamada 4, serve de filtro para todas as respostas corretas de uma turma em um OA específico;

6. **`/api/teacher/retroactions/:team_id/:lo_id/answers/wrongs`**

Esta chamada da API é uma variação da chamada 4, serve de filtro para todas as respostas erradas de uma turma em um OA específico;

7. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id`**  
Esta chamada da API é responsável por retornar todas as respostas de um aprendiz em uma turma de um OA específico;
8. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/corrects`**  
Esta chamada da API é uma variação da chamada 7, serve de filtro para todas as respostas corretas de um aprendiz em uma turma de um OA específico;
9. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/wrongs`**  
Esta chamada da API é uma variação da chamada 7, serve de filtro para todas as respostas erradas de um aprendiz em uma turma de um OA específico;
10. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id`**  
Esta chamada da API é responsável por retornar todas as respostas de um aprendiz em uma turma de um exercício em um OA específico;
11. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/corrects`**  
Esta chamada da API é uma variação da chamada 10, serve de filtro para todas as respostas corretas de um aprendiz em uma turma de um exercício em um OA específico;
12. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/wrongs`**  
Esta chamada da API é uma variação da chamada 10, serve de filtro para todas as respostas erradas de um aprendiz em uma turma de um exercício em um OA específico;
13. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id`**  
Esta chamada da API é responsável por retornar todas as respostas de um aprendiz em uma turma de um exercício em determinado passo em um OA específico;
14. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id/corrects`**  
Esta chamada da API é uma variação da chamada 13, serve de filtro para todas as respostas corretas de um aprendiz em uma turma de um exercício em determinado passo em um OA específico;
15. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id/wrongs`**  
Esta chamada da API é uma variação da chamada 13, serve de filtro para todas as respostas erradas de um aprendiz em uma turma de um exercício em determinado passo em um OA específico;
16. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id/:answer_id/history`**  
Esta chamada da API é responsável por retornar um histórico de resposta inseridas por um aprendiz em uma turma de um exercício em determinado passo em um OA específico;
17. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id/wrong/:answer_id/users_with_same_answer`**  
Esta chamada da API é responsável por retornar uma lista de aprendizes que tiveram a mesma resposta errada que um aprendiz em uma turma de um exercício em determinado

passo em um OA específico;

18. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id/corrects/:answer_id/users_with_same_answer`**

Esta chamada da API é responsável por retornar uma lista de aprendizes que tiveram a mesma resposta correta que um aprendiz em uma turma de um exercício em determinado passo em um OA específico;

19. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id/wrong/:answer_id/users`**

Esta chamada da API é responsável por retornar uma lista de aprendizes que erraram independente da resposta semelhante a um aprendiz em uma turma de um exercício em determinado passo em um OA específico;

20. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id/corrects/:answer_id/users`**

Esta chamada da API é responsável por retornar uma lista de aprendizes que acertaram independente da resposta semelhante a um aprendiz em uma turma de um exercício em determinado passo em um OA específico;

21. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id/corrects/:answer_id/users/equal_attempts_number`**

Esta chamada da API é responsável por retornar uma lista de aprendizes que tiveram o mesmo número de tentativas que um aprendiz em uma turma de um exercício em determinado passo em um OA específico;

22. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id/:answer_id/`**

Esta chamada da API é responsável por retornar o OA somente com o exercício de uma determinada resposta;

23. **`/api/teacher/retroactions/:team_id/:lo_id/answers/:user_id/:exercise_id/:step_id/:answer_id/page/:page_id`**

Esta chamada da API é responsável por páginar os resultados de um OA em uma turma de um exercício de uma determinada resposta;

#### 5.1.2.2 CHAMADAS DE API PARA A RETROAÇÃO NA ÁREA DO ESTUDANTE

1. **`/api/student/retroactions/:team_id/:lo_id/answers`**

Esta chamada da API é responsável por retornar todas as respostas de um aprendiz em uma turma de OA em específico;

2. **`/api/student/retroactions/:team_id/:lo_id/answers/corrects`**

Esta chamada da API é responsável por retornar todas as respostas corretas de um aprendiz em uma turma de OA em específico;

3. **`/api/student/retroactions/:team_id/:lo_id/answers/wrongs`**

Esta chamada da API é responsável por retornar todas as respostas erradas de um

aprendiz em uma turma de OA em específico;

4. **/api/student/retroactions/:team\_id/:lo\_id/answers/:exercise\_id**

Esta chamada da API é responsável por retornar todas as respostas de um aprendiz em uma turma de um exercício em um OA específico;

5. **/api/student/retroactions/:team\_id/:lo\_id/answers/:exercise\_id/corrects**

Esta chamada da API é responsável por retornar todas as respostas corretas de um aprendiz em uma turma de um exercício em um OA específico;

6. **/api/student/retroactions/:team\_id/:lo\_id/answers/:exercise\_id/wrongs**

Esta chamada da API é responsável por retornar todas as respostas erradas de um aprendiz em uma turma de um exercício em um OA específico;

7. **/api/student/retroactions/:team\_id/:lo\_id/answers/:exercise\_id/:step\_id**

Esta chamada da API é responsável por retornar todas as respostas de um aprendiz em uma turma de um exercício em determinado passo em um OA específico;

8. **/api/student/retroactions/:team\_id/:lo\_id/answers/:exercise\_id/:step\_id/corrects**

Esta chamada da API é responsável por retornar todas as respostas corretas de um aprendiz em uma turma de um exercício em determinado passo em um OA específico;

9. **/api/student/retroactions/:team\_id/:lo\_id/answers/:exercise\_id/:step\_id/wrongs**

Esta chamada da API é responsável por retornar todas as respostas erradas de um aprendiz em uma turma de um exercício em determinado passo em um OA específico;

10. **/api/student/retroactions/:team\_id/:lo\_id/:answers/:exercise\_id/:step\_id/:answer\_id/history**

Esta chamada da API é responsável por retornar um histórico de respostas inseridas por um aprendiz em uma turma de um exercício em determinado passo em um OA específico;

## 5.2 LINHA DO TEMPO

A linha do tempo ainda não existe na atual versão da ferramenta FARMA, essa funcionalidade irá ter um volume muito grande de dados visto que ela seria basicamente um *log* de informações e passos realizados por um aprendiz. Para que seja possível implementar a linha do tempo primeiramente é necessário escolher um banco de dados que irá trabalhar em paralelo com o banco da aplicação. Posteriormente será necessário levantar todos os dados que serão candidatos a serem armazenados.

Como essa funcionalidade irá trabalhar conversando com a aplicação atual mas em um banco de dados separado, existe uma maior liberdade para sua criação. Posterior a escolha do banco de dados e dos campos que deverão ser salvos, a funcionalidade retorna sua ligação total com a ferramenta no momento da criação da API. Para a API dessa funcionalidade serão necessários filtros no código da aplicação que realizem um disparo de eventos onde seja possível o armazenamento dos passos do aprendiz.



### 5.2.1 BANCO DE DADOS

Inicialmente foi pensado na utilização do banco de dados MongoDB<sup>1</sup> por trabalhar bem com grande quantidade de dados e por sua adaptação aos tipos de informações que podem ser armazenadas por ele, também foi cogitada a utilização em paralelo do banco de dados Elasticsearch<sup>2</sup> por trabalhar muito bem com a velocidade na entrega de resultados e também se adaptar aos dados armazenados (RODRIGUES, 2016).

Nos SGBDs relacionais e não relacionais existem alguns gargalos devido a grande quantidade de dados e com buscas por uma inserção de dois minutos atrás até cinco anos (DUNNING; FRIEDMAN, 2014). Para solucionar esses problemas foi escolhido a utilização de um banco de dados TSDB (Time Series Database). Inicialmente foi escolhido o banco de dados opentsdb<sup>3</sup> por ser *software* livre e com uma licença de uso comercial.

Os bancos de dados TSDB trabalham de maneira diferenciada, onde cada informação é uma sequencia de valores, cada um com um valor de tempo indicando quando foi registrado. Como esse tipo de dado normalmente não sofre alterações mas são frequentemente solicitados a otimização nesses tipo de SGBD é totalmente voltada a responder prontamente uma consulta independente de seu tamanho (DUNNING; FRIEDMAN, 2014).

### 5.2.2 API

Apesar deste trabalho ter o objetivo de salvar e consultar os dados de linha do tempo a API somente terá chamadas para a consulta. Isso se deve ao fato que salvar os dados da linha do tempo será algo imperceptível na ferramenta, pois existirão procedimento internos no código que irão disparar ações após cada passo do aprendiz realizando o armazenamento dessas informações.

A linha do tempo terá seis chamadas para área do tutor onde por padrão são apresentados dados de uma semana, mas é possível especificar o período desejado. Caso haja necessidade no decorrer desse projeto essas chamadas podem ser alteradas, canceladas ou até mesmo novas serem criadas. As palavras na listagem a baixo que vem seguida do uso de ':' (dois pontos) fazem referência a parâmetros que devem ser enviados para a API.

#### 5.2.2.1 CHAMADAS DE API PARA A LINHA DO TEMPO

1. **api/teacher/:user\_id/**

Esta chamada da API é responsável apresentar todos os dados de linha do tempo de um aprendiz;

2. **api/teacher/:user\_id/:team\_id**

Esta chamada da API é responsável apresentar todos os dados de linha do tempo de um aprendiz em determinada turma;

---

<sup>1</sup><https://www.mongodb.com/>

<sup>2</sup><https://www.elastic.co/pt/>

<sup>3</sup><http://opentsdb.net/>

**3. api/teacher/:user\_id/:team\_id/:lo\_id**

Esta chamada da API é responsável apresentar todos os dados de linha do tempo de um aprendiz em determinada turma de um OA em específico;

**4. api/teacher/:user\_id/:team\_id/:lo\_id/answers**

Esta chamada da API é responsável apresentar todos os dados de linha do tempo de uma resposta de um aprendiz em determinada turma de um OA em específico;

**5. api/teacher/:user\_id/:team\_id/:lo\_id/answers/correts**

Esta chamada da API é responsável apresentar todos os dados de linha do tempo de uma resposta correta de um aprendiz em determinada turma de um OA em específico;

**6. api/teacher/:user\_id/:team\_id/:lo\_id/answers/wrongs**

Esta chamada da API é responsável apresentar todos os dados de linha do tempo de uma resposta errada de um aprendiz em determinada turma de um OA em específico;

## 6 RESULTADOS PARCIAIS

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

**Estudo do modelo banco de dados Correia (2017).** Nesta etapa foram realizados estudos e testes no modelo de banco de dados proposto por Correia (2017), os códigos do modelo foram executados e os resultados obtidos foram analisados. Mostrou-se satisfatório os dados coletados nos testes e foi confirmado a possibilidade da implantação do modelo sem nenhuma alteração na ferramenta.

**Criação das chamadas de APIs.** Nesta etapa foram realizadas reuniões com o professor orientador e com Ramos (2019) para debater quais seriam as chamadas necessárias para que o desenvolvimento da funcionalidade de retroação à erros estivesse disponível na ferramenta, para dar sequencia no desenvolvimento das telas.

**Definição do banco de dados para a linha do tempo.** Nesta etapa foram estudados diversos bancos de dados incluindo modelos relacionais e não relacionais, em alguns testes foram realizados em outros foi constatado em artigos acadêmicos que seu uso não satisfaz as exigências da quantidade de dados para uma linha do tempo. Foi adotado então um modelo TSDB mais especificamente o Opentsdb por ser um *software* de código aberto e de documentação bem detalhada.

## 7 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho é criar APIs para a ferramenta FARMA, para que a consulta de dados da retroação à erros e a consulta e gravação de informações da linha do tempo estejam disponíveis. Seguindo o modelo proposto por [Correia \(2017\)](#) e em sincronia com o corrente trabalho de [Ramos \(2019\)](#) foram levantado todos os passos que devem ser seguidos para a o desenvolvimento das APIs de retroação à erros bem como a implantação do banco de dados no atual andamento da ferramenta.

Para a linha do tempo, como não existem precedentes na ferramenta, o principal objetivo era a definição de como os dados seriam armazenados. Com o estudo de vários SGBDs chegou-se a conclusão que o melhor caminho para arquivar a grande quantidade de dados seria o uso de um banco de dados de series temporais.

A solução para o problema geral deste trabalho será a criação das APIs sobre os bancos de dados de linha do tempo e de retroação à erros. A criação dessas APIs são de total importância para a ferramenta, servindo todos os dados necessários para remediação à erros nas funcionalidades de retroação e linha do tempo. As Chamadas para essas APIS já estão definidas, bastando agora dar sequencia no projeto iniciando o desenvolvimento na ferramenta.



## Referências

- BRODBECK, M. H. J. **Comparando ambientes de desenvolvimento Java e Ruby on Rails**. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, 2007. Citado 2 vezes nas páginas 15 e 16.
- CAELUM. **Desenvolvimento Ágil para Web 2.0 com Ruby on Rails**. 2013. Disponível em: <<https://www.caelum.com.br/apostila-ruby-on-rails/>>. Citado 2 vezes nas páginas 16 e 18.
- CANNON, M. D.; EDMONDSON, A. C. Failing to learn and learning to fail (intelligently):how great organizations put failure to work to innovate and improve. **Long Range Planning**, v. 38, p. 299–319, 2005. Disponível em: <[https://www.researchgate.net/publication/228630786\\_Failing\\_to\\_Learn\\_and\\_Learning\\_to\\_Fail\\_Intelligently\\_How\\_Great\\_Organizations\\_Put\\_Failure\\_to\\_Work\\_to\\_Innovate\\_and\\_Improve/download](https://www.researchgate.net/publication/228630786_Failing_to_Learn_and_Learning_to_Fail_Intelligently_How_Great_Organizations_Put_Failure_to_Work_to_Innovate_and_Improve/download)>. Acesso em: 31 de maio de 2019. Citado na página 5.
- CARNEIRO, M. L. F.; SILVEIRA, M. S. Objetos de aprendizagem como elementos facilitadores na educação a distância. **Educar em Revista**, n. 4, p. 235–260, 2014. Disponível em: <<http://www.scielo.br/pdf/er/nspe4/0101-4358-er-esp-04-00235.pdf>>. Acesso em: 03 de Abril de 2019. Citado na página 1.
- COLLINGBOURNE, H. **The Little Book Of Ruby**. [S.l.]: Dark Neon Ltd, 2006. Citado na página 15.
- CORREIA, L. **CONTROLE DE VERSÃO PARA ARMAZENAMENTO DO CONTEXTO DA INTERAÇÃO DO APRENDIZ COM OBJETOS DE APRENDIZAGEM**. [S.l.], 2017. Disponível em: <[https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/93/GP\\_COINT\\_2017\\_1\\_22\\_LETICIA\\_CORREIA\\_MONOGRAFIA\\_COM\\_ATA.pdf?1499722594](https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/93/GP_COINT_2017_1_22_LETICIA_CORREIA_MONOGRAFIA_COM_ATA.pdf?1499722594)>. Acesso em: 21 de Abril de 2019. Citado 9 vezes nas páginas 1, 2, 3, 4, 20, 27, 28, 34 e 35.
- DOMINICO, S. **Tuning: Um Estudo sobre a Otimização de Desempenho de Sistemas Gerenciadores de Banco de Dados Relacionais sob Carga de Trabalho de Suporte a Decisão**. 2013. Monografia (Tecnólogo em Sistemas para Internet), UTFPR (Universidade Tecnológica Federal do Paraná), Guarapuava, Brasil. Citado na página 17.
- DUNNING, T.; FRIEDMAN, E. **Time Series Databases**. [S.l.]: O'Reilly Media, 2014. ISBN 978-1-491-91702-2. Citado na página 25.
- GOUDINHO, A. **DESENVOLVIMENTO DO MODULO DE ESTATÍSTICAS DA FERRAMENTA DE AUTORIA DE OBJETOS DE APRENDIZAGEM FARMA**. [S.l.], 2019. Disponível em: <[https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/154/GP\\_COINT\\_2019\\_1\\_PROPOSTA\\_AGUINALDO\\_GOUDINHO\\_DOS\\_SANTOS.pdf?1559243519](https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/154/GP_COINT_2019_1_PROPOSTA_AGUINALDO_GOUDINHO_DOS_SANTOS.pdf?1559243519)>. Acesso em: 18 de Junho de 2019. Citado 2 vezes nas páginas 2 e 13.
- KUTZKE, A. R. Doutorado em Ciência da Computação, **Informática educacional e a mediação do erro na educação : um estudo teórico-crítico e uma proposta de instrumento computacional**. Paraná: [s.n.], 2015. 197 f. Disponível em: <<https://acervodigital.ufpr.br/bitstream/handle/1884/40907/R-T-ALEXANDERROBERTKUTZKE.pdf>>. Acesso em: 31 de maio de 2019. Citado na página 5.

LARA, D. **APLICAÇÃO DE DESIGN DE INTERFACE NA ÁREA DE ANÁLISE DE APRENDIZAGEM DA FARMA**. [S.l.], 2017. Disponível em: <[https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/95/GP\\_COINT\\_2017\\_1\\_DARLAN\\_LARA\\_DE\\_ALMEIDA\\_PROJETO.pdf?1500473380](https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/95/GP_COINT_2017_1_DARLAN_LARA_DE_ALMEIDA_PROJETO.pdf?1500473380)>. Acesso em: 18 de Junho de 2019. Citado 5 vezes nas páginas 2, 7, 8, 9 e 12.

LEITE, M. O. et al. Aplicação do sistema kanban no transporte de materiais na construção civil. **Revista Produção Online**, v. 4, n. 4, 2004. Citado na página 14.

MARCZAL, D. Doutorado em Ciência da Computação, **FARMA: UMA FERRAMENTA DE AUTORIA PARA OBJETOS DE APRENDIZAGEM DE CONCEITOS MATEMÁTICOS**. Paraná: [s.n.], 2014. 196 f. Disponível em: <[http://repositorio.utfpr.edu.br/jspui/bitstream/1/2152/1/UFPR\\_PPGINF\\_%20Marczal%2C%20Diego\\_2014.pdf](http://repositorio.utfpr.edu.br/jspui/bitstream/1/2152/1/UFPR_PPGINF_%20Marczal%2C%20Diego_2014.pdf)>. Acesso em: 10 de junho de 2019. Citado 2 vezes nas páginas 10 e 11.

MARCZAL, D.; DIRENE, A. Farma: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos. 2012. Citado 2 vezes nas páginas 1 e 6.

MILANI, A. **PostgreSQL - Guia do Programador**. [S.l.]: Novatec, 2008. ISBN 978-85-7522-157-0. Citado na página 17.

MOREIRA, M. A. Aprendizagem significativa: da visão clássica à visão crítica. **I Encontro Nacional de Aprendizagem Significativa**, p. 15, 2005. Disponível em: <<https://www.if.ufrgs.br/~moreira/visaoclasicavisaocritica.pdf>>. Acesso em: 31 de maio de 2019. Citado na página 5.

PASINOTO, R. **O ERRO NO PROCESSO DE ENSINO-APRENDIZAGEM**. [S.l.], 2008. Citado na página 7.

PATERLINI, R. Uma proposta para o sistema de linha do tempo da utfpr - londrina. 2016. Citado na página 3.

PENCHEL, M. C.; CANTARELA, R. Sistema web de anúncio de vagas em imóveis: Um exemplo de desenvolvimento ágil com ruby on rails. 2015. Citado na página 15.

RAMOS, J. H. **TECLADO VIRTUAL PARA EXPRESSOES ARITMÉTICAS E ALGÉBRICAS**. [S.l.], 2019. Disponível em: <[https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/120/GP\\_COINT\\_2018\\_1\\_JEFFERSON\\_HENRIQUE\\_RAMOS\\_PROPOSTA.pdf?1528470476](https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/120/GP_COINT_2018_1_JEFFERSON_HENRIQUE_RAMOS_PROPOSTA.pdf?1528470476)>. Acesso em: 18 de Junho de 2019. Citado 3 vezes nas páginas 2, 27 e 28.

RODRIGUES, M. D. **Análise Sobre Bases de Dados Para Armazenamento e Consulta de Dados Não Estruturados e Semiestruturados**. 2016. Monografia (bacharel em Ciência da Computação) Centro de Informática da Universidade Federal de Pernambuco. Citado na página 25.

SANTOS, E. L. D. **PROJETO DE UMA NOVA INTERFACE GRÁFICA PARA A FERRAMENTA DE AUTORIA DE OBJETOS DE APRENDIZAGEM MATEMÁTICOS FARMA**. [S.l.], 2018. Disponível em: <[https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/135/GP\\_COINT\\_2018\\_2\\_02\\_ERIKA\\_LEAO\\_DOS\\_SANTOS\\_MONOGRAFIA.pdf?1545308065](https://tcc.tsi.gp.utfpr.edu.br/attachments/approvals/135/GP_COINT_2018_2_02_ERIKA_LEAO_DOS_SANTOS_MONOGRAFIA.pdf?1545308065)>. Acesso em: 18 de Junho de 2019. Citado na página 2.

SAUDATE, A. **REST, Construa API's inteligentes de maneira simples**. [S.l.]: Casa do Código, 2013. Citado 2 vezes nas páginas 17 e 18.

SILVA, D. V. d. S.; SANTOS, F. A. d. O.; NETO, P. S. Os benefícios do uso de kanban na gerência de projetos de manutenção de software. **The Standish Group: CHAOS Summary for**, 2010. Citado na página 14.

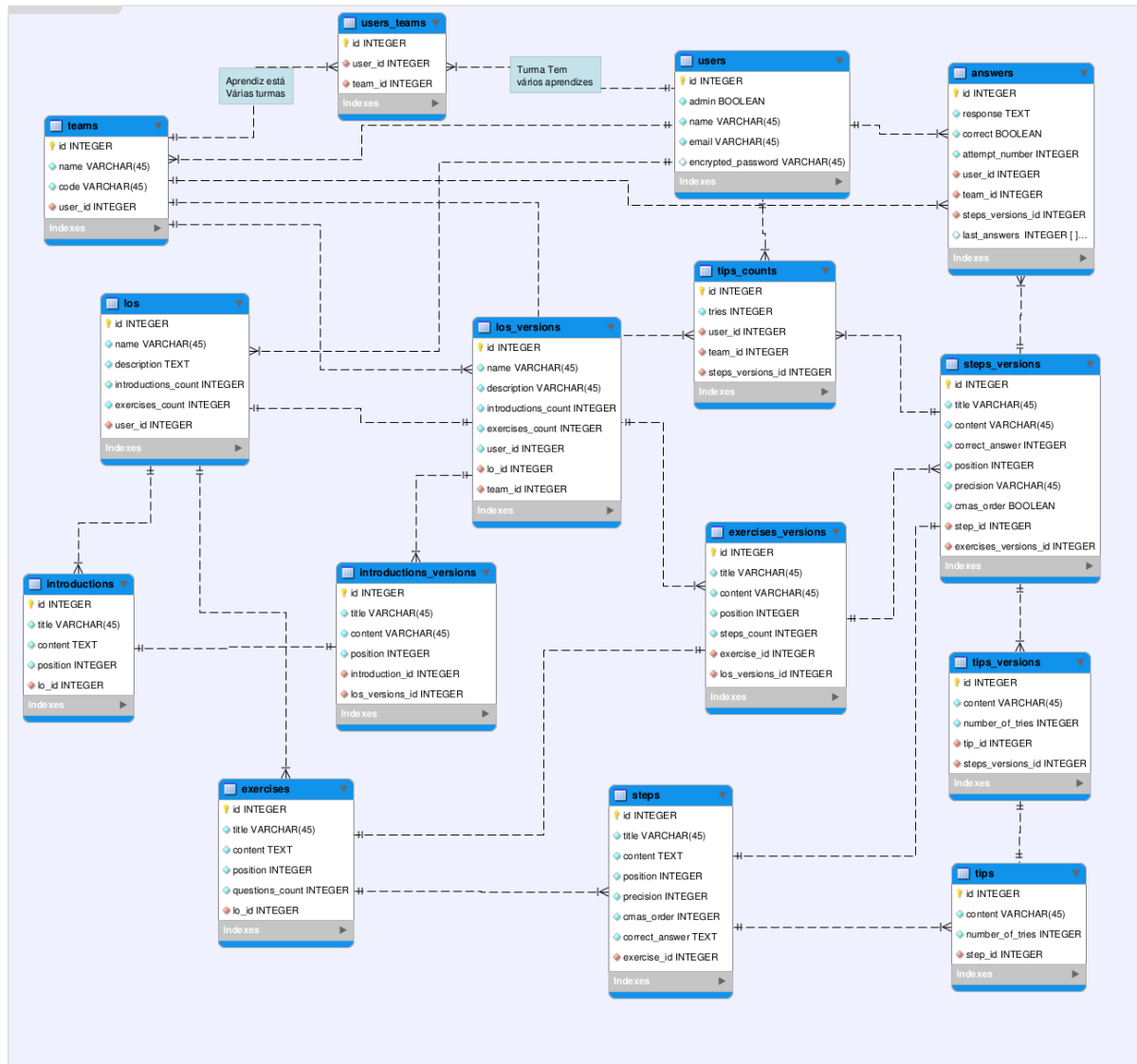
SOUZA, L. **Ruby aprenda a programar na linguagem mais divertida**. [S.l.]: Casa do Código, 2012. ISBN 978-85-66250-24-4. Citado 2 vezes nas páginas 15 e 16.



Anexos

## ANEXO A – MODELAGEM DA ÁREA DE RETROAÇÃO À ERROS NA FARMA

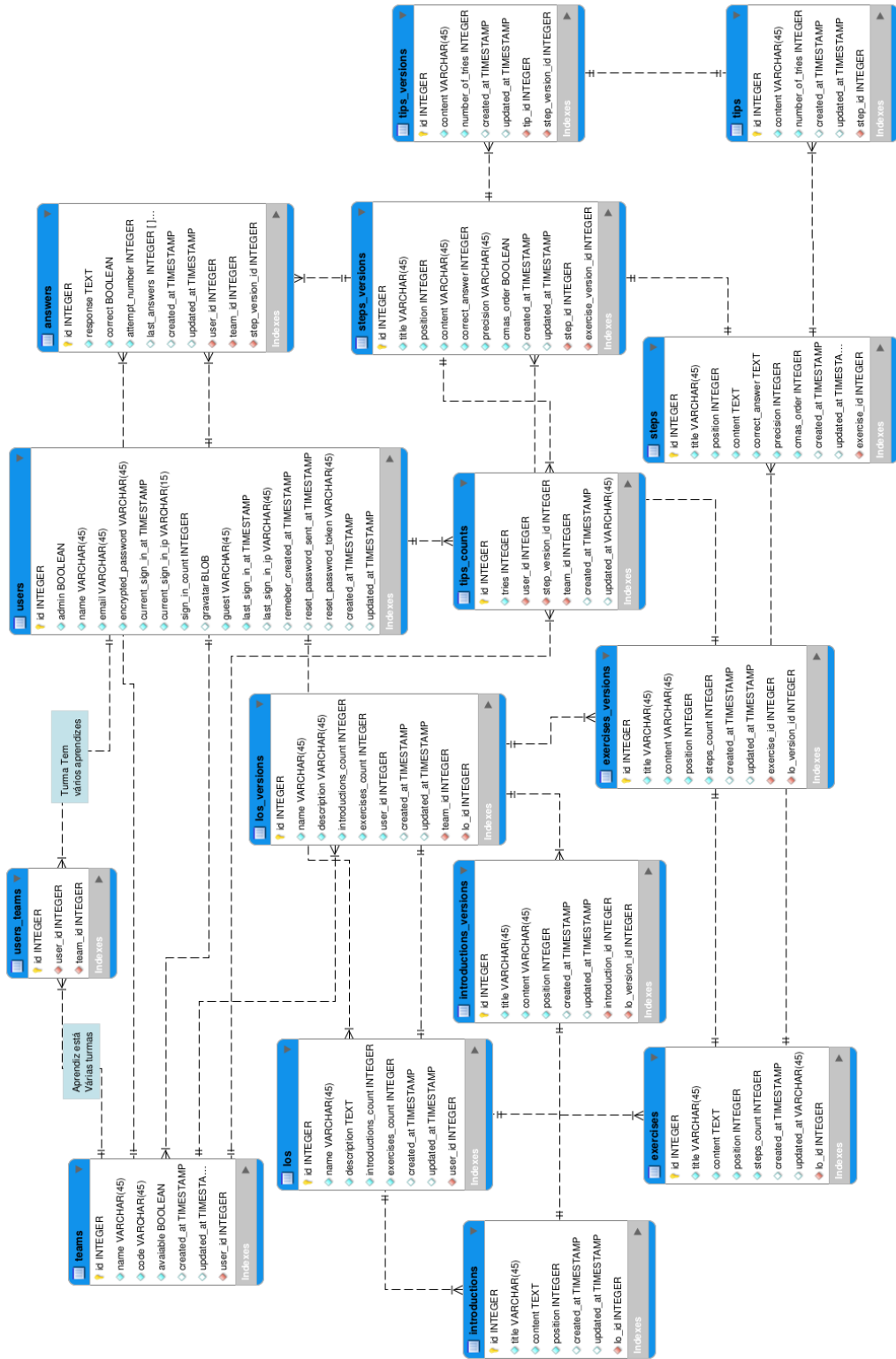
Figura 8 – Modelagem Retroação - FARMA



Fonte: Correia (2017)

## ANEXO B – MODELAGEM COMPLETA DA FARMA

Figura 9 – Modelagem Completa - FARMA



Fonte: Correia (2017)