

UNIVERSIDADE FEDERAL DE PERNAMBUCO  
Centro de Informática  
Mestrado Profissional em Ciência da Computação

Fabiano Gonçalves Matos  
Fabrício Gonçalves Matos  
Nelson Barros Cavalcante Junior  
Rogério Luiz Cardoso Silva Filho

**RELATÓRIO**  
**PROJETO REFORÇO**



Projeto de Desenvolvimento de  
*Software* SaaS como forma de  
avaliação para a disciplina de  
***Engenharia de Software***.

Professor: Vinicius Cardoso Garcia

Montes Claros – MG  
Julho / 2015

## 1 Introdução

As mudanças que vem ocorrendo na sociedade, principalmente decorrente das novas descobertas e inovações tecnológicas vêm exigindo uma reorganização nas atividades escolares e aumentando as possibilidades no mercado de trabalho educacional, principalmente quando se fala em “cursos livres”.

Na modalidade de “cursos livres” podemos caracterizar os que são organizados para serem oferecidos com cargas horárias inferiores à 200 horas/aula e, por ter o conteúdo do programa de inteira responsabilidade de seus organizadores. O Reforço Escolar objeto do nosso projeto enquadra-se na modalidade de “cursos livres”.

Com essas inovações tecnológicas têm surgido um novo modelo de comercialização de software, conhecido como *Software* como Serviço ou *Software as a Service (SaaS)*. No qual, a funcionalidade da aplicação é oferecida através de um modelo de assinatura pela Internet sem que o cliente se torne dono do *software*, ao invés disso, ele aluga ou usufrui de solução gratuitas que são oferecidas remotamente.

Tendo em vista as demandas crescentes no mercado de trabalho na área de educação, principalmente na micro-região do Norte de Minas, e as grandes possibilidades na produção de *Software* como Serviço surge o projeto REFORÇO, um sistema desenvolvido no modelo SaaS com o propósito de oferecer um serviço aos discentes, na contratação de profissionais da área de educação visando os cursos livres em especial o reforço escolar, e aos docentes que terão a possibilidade de divulgarem os seus serviços para serem contratados.

Nessa perspectiva esse projeto visa a produção de um SaaS seguindo as teorias da Engenharia de *Software* na especificação, no desenvolvimento e na manutenção de sistemas de *software*, com aplicação de tecnologias e práticas de gerência de projetos, visando organização, produtividade e qualidade.

## **2 Justificativa**

O projeto Reforço se justifica pelas necessidades do crescente mercado da educação na microrregião do Norte de Minas, e a verificação informal do anseio de docentes e discentes por um software que atenda as demandas de divulgação de cursos livres de maneira a propiciar os meios de comunicações entre professores e alunos bem como uma forma de avaliação do serviço prestado.

## **3 Gerência de Configuração e Ambiente**

Na produção do software Reforço foram utilizados diversas ferramentas para uma melhor gestão nos processos de desenvolvimento. Segundo HARRISON et al., 2000, se faz necessário o uso de coleções de ferramentas integradas que facilitam as atividades da engenharia de software, durante todo o ciclo de vida do processo, ou seja, um Ambientes de Desenvolvimento de Software (ADSs).

Nesse Ambiente de Desenvolvimento foram utilizados as ferramentas listadas abaixo com suas finalidades e o momento do uso da mesma nas etapas da produção do software Reforço.

### **3.1 Designer**

- Gimp - programa de código aberto voltado principalmente para criação e edição de imagens, e em menor escala também para desenho vetorial. - utilizado na criação do logo.
- Inkscape - aplicação completamente livre com funcionalidades muito similares a aplicações comerciais como o Illustrator, CorelDraw, Xara X, entre outras aplicações de desenho vetorial. - utilizada na criação do logo.
- Draw.io - Software online para criação de diagramas diversos.

### **3.2 Modelagem lógica e de diagramas**

- EERCASER – Ferramenta de modelagem logica de banco de dados relacional. - Usada no inicio da modelagem do domínio para se ter a logica do

negocio e os seus principais entes bem como para se obter o modelo do banco de dados a ser usado na aplicação.

- ASTAH – Ferramenta utilizada para digramas de classe – Essa ferramenta foi utilizada na modelagem dos diagramas de classes.

### 3.3 Banco de Dados

- SQLLITE - banco de dados "light", que fica armazenado em um arquivo local e pode ser transportado de um servidor/aplicação para outro livremente.
- PostgreSQL - é um sistema de gerenciamento de banco de dados objeto-relacional.

### 3.4 Linguagens de Desenvolvimento e Marcação

- Ruby - Linguagem de programação interpretada, com tipagem dinâmica e forte, orientada a objetos (versão: ruby 2.2.1p85 (2015-02-26 revision 49769) [x86\_64-linux]).
- HTML - linguagem de marcação utilizada para produção de páginas na web, que permite a criação de documentos que podem ser lidos em praticamente qualquer tipo de computador e transmitidos pela internet.
- CSS - uma "folha de estilo" composta por “camadas” e utilizada para definir a apresentação (aparência) em páginas da internet que adotam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML).
- JavaScript - é uma linguagem de programação baseada em *scripts* e padronizada pela ECMA International, utilizada neste projeto através do framework JQuery.

### 3.5 Frameworks

- Ruby on Rails – Framework de desenvolvimento web (gratuito e de código aberto) que possibilita o MVC e a construção de SaaS (versão: Rails 4.2.1).
- Bootstrap 3.3.5 – framework que contém uma estrutura pronto para criação de

páginas web, possui diversos componentes HTML, CSS e Javascript prontos para utilização.

- JQuery - é uma biblioteca JavaScript cross-browser desenvolvida para simplificar os scripts “client side” que interagem com o HTML. Ela foi lançada em dezembro de 2006 no BarCamp de Nova York por John Resig. Usada por cerca de 77% dos 10 mil sites mais visitados do mundo, jQuery é a mais popular das bibliotecas JavaScript.

### **3.6 Repositório de Controle de Versão e Mudanças**

- Github - um serviço de hospedagem distribuído desenvolvido em Ruby on Rails para projetos que utilizam o controle de versão Git. Ele é um sistema de controle de versão (ou versionamento), VCS (do inglês version control system) ou ainda SCM (do inglês source code management) distribuído.

### **3.7 Sistema Operacional**

- Linux - distribuição não comercial livre, Ubuntu 14.02.

### **3.8 IDE de desenvolvimento e ambiente cloud**

- Cloud9 – IDE de desenvolvimento em nuvem. - Usada para codificação do sistema.

### **3.9 Ferramenta de Armazenamento de Arquivo**

- Google Drive – repositório de documentos e arquivos importantes do projeto. - decidido a utilização após a primeira reunião.

### **3.10 Ferramentas de Comunicação**

- WhatsApp - aplicativo de mensagens multiplataforma que permite trocar mensagens pelo celular – usado desde o início do projeto para comunicação

entre os integrantes da equipe.

- Gmail – correio eletrônico utilizado para envio de documentos e mensagens. - utilizado desde o início do projeto.
- Skype – utilizado para video conferencias e comunicação instantânea.

## **4 Requisitos**

A elicitação de requisitos foi baseada nos conhecimentos dos membros da equipe, dos quais todos já trabalharam como professores e um possuía formação docente, obtendo assim uma visão clara do que seria necessário para a construção do *software*.

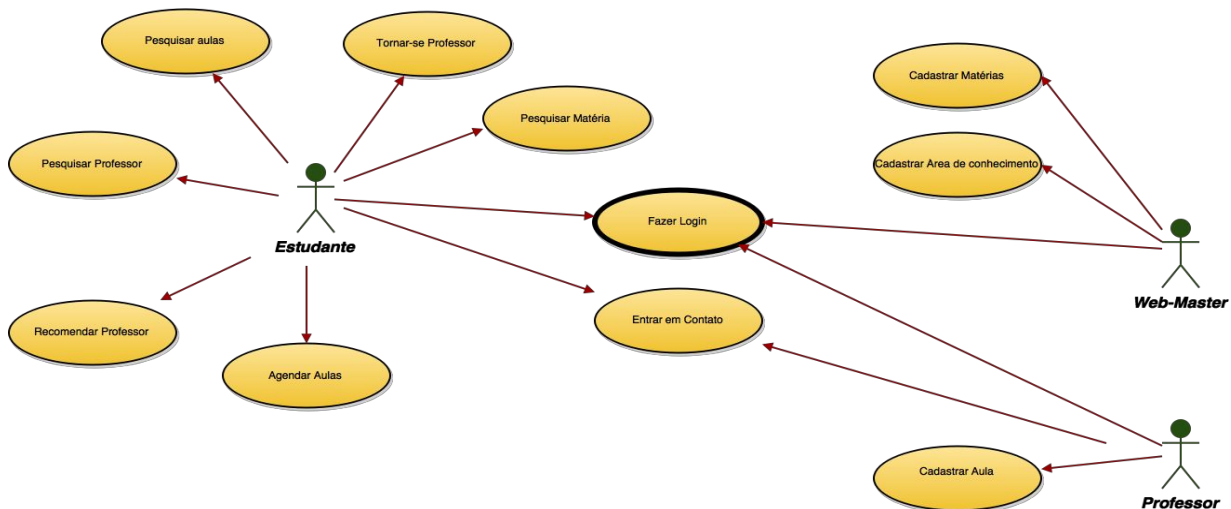
A relação de requisitos levantados no projeto é apresentada no ANEXO I - “Documento de requisitos”.

## **5 Visão de Análise e Projetos**

Para a análise e projeto foram elencados pela a equipe a necessidade da construção dos casos de uso, modelo lógico do banco de dados, dicionário de dados e a elaboração do diagrama de classes, os quais seriam o suficiente para o entendimento do domínio do sistema e o seu desenvolvimento.

## 5.1 Modelo de Caso de Uso

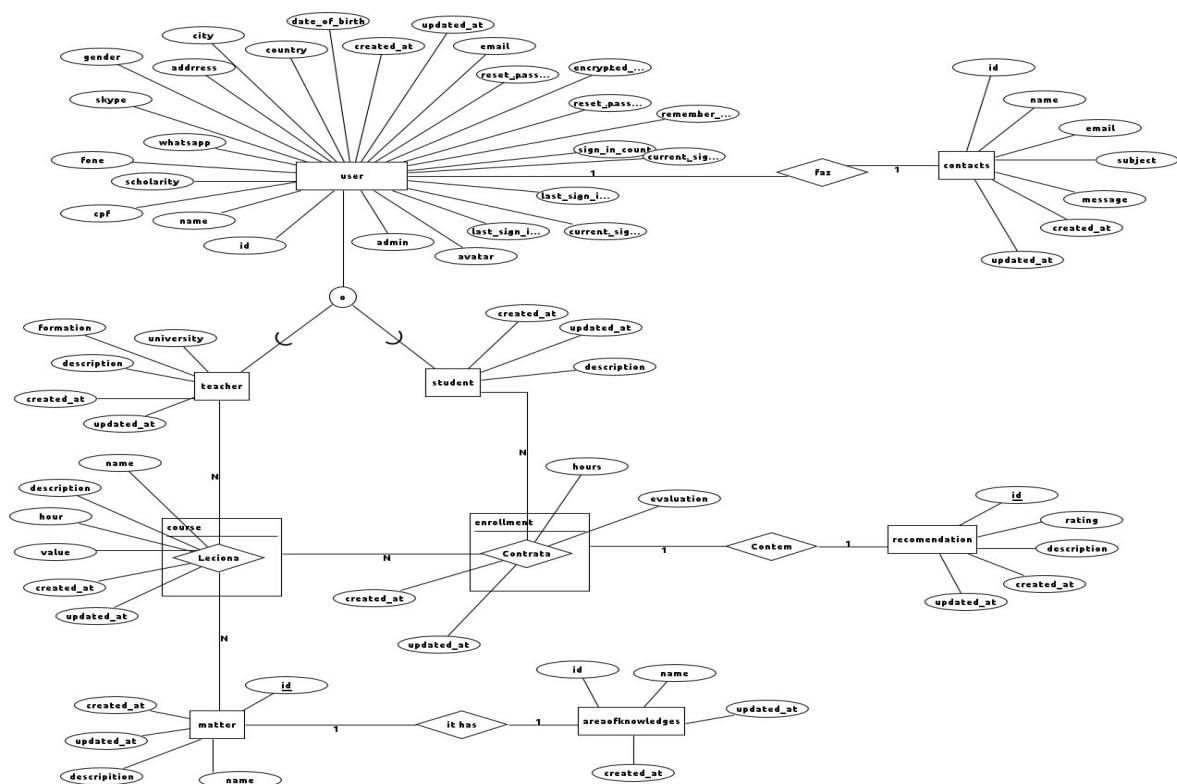
‘Neste tipo de diagrama é possível visualizar os atores (stakeholders), as principais atividades do domínio da aplicação e suas relações. Auxilia na ilustração do levantamento de requisitos.



**Figura - 1 Diagrama de Caso de Uso (ANEXO II)**

## 5.2 Modelo Lógico do banco de dados

O modelo lógico do banco de dados da *figura 1* nos determina as entidades envolvidas no sistema, os atributos de cada entidades bem como os relacionamentos existentes entre elas possibilitando assim uma análise melhor do fluxo de armazenamento dos dados.



**Figura - 2 Modelo Conceitual (ANEXO III)**

### 5.3 Dicionário de Dados

A partir do modelo lógico foi possível gerar o dicionário de dados, o qual serviu de orientação na construção dos scripts do banco de dados. Nesse dicionário de dados são descritos as tabelas do banco com seus campos, tipos e tamanho.

#### User (Usuário)

Campo	Descrição	Tipo	Default	Observação
Id	Código autoincremento.	Integer		Chave primaria
Name	Nome do usuário.	String		-



cpf	Campo para registro do CPF do usuário	String		-
scholarity	Registro da escolaridade	String		-
fone	Registro de telefone do usuário	String		-
whatsapp	Registro do whatsapp do usuário para comunicação.	String		-
skype	Rigistra o skype do usuário para comunicação.	String		-
address	Endereço do usuário, composto de rua, n, bairro e cidade.	String		-
State	Estado (unidade federativa) do usuário	String		-
Country	Pais do usuário.	String		-
date_of_birth	Data de aniversário do usuário	Date		-
created_at	Data de criação do registro de usuário	Datetime		-
updated_at	Data de atualização do registro do usuário.	Datetime		

email	Email do usuário.	string		
encrypted_password	Senha do usuário criptografada.	string		
reset_password_token	token para recuperação de senha.	string		
reset_password_sent_at	Data de envio do token.	datetime		
remember_created_at	Data de pedido do token.	datetime		
sign_in_count	Quantidade de logins registrados no sistema do usuário.	integer		
current_sign_in_at	Data do login corrente.	datetime		
last_sign_in_at	Data do último login.	datetime		
current_sign_in_ip	IP do login corrente.	string		
last_sign_in_ip	IP do último login.	string		
avatar_file_name	Nome do arquivo de imagem do perfil do usuário.	string		

avatar_content_type	Tipo de conteúdo da imagem.	string		
avatar_file_size	Tamanho do arquivo de imagem do usuário.	integer		
avatar_updated_at	Data de upload da imagem do perfil do usuário.	datetime		
admin	Campo que identifica se o usuário exerce papel de administrador no sistema.	boolean		

### Teachers (Professores)

Campo	Descrição	Tipo	Default	Observação
Id	Código autoincremento.	Integer		Chave primaria
formation	Formação do professor	String		-
university	Universidade onde o professor se formou.	String		
description	Descrição pessoal do professor.	String		
Id_user	Código de usuário	integer		Chave

				Estrangeira
created_at	Data da criação do registro de professor	Date		-
updated_at	Data de atualização do registro do usuário.	Datetime		

### Students (Alunos)

Campo	Descrição	Tipo	Default	Observação
Id	Código autoincremento.	Integer		Chave primaria
user	Código de usuário	Integer		Chave Estrangeira
created_at	Data da criação do registro de professor	Date		-
updated_at	Data de atualização do registro do usuário.	Datetime		

### Areaofknowledges (Área de Conhecimentos)

Campo	Descrição	Tipo	Default	Observação
Id	Código autoincremento.	Integer		Chave primaria

name	Nome da área de conhecimento.	String		-
created_at	Data da criação do registro de professor	Date		-
updated_at	Data de atualização do registro do usuário.	Datetime		

### Matters (Matérias)

Campo	Descrição	Tipo	Default	Observação
Id	Código autoincremento.	Integer		Chave primaria
name	Nome da matéria	String		-
description	Descrição da matéria	String		-
areaOfKnowledge	Código da área de conhecimento.	Integer		Chave Estrangeira
created_at	Data da criação do registro de professor	Date		-
updated_at	Data de atualização do registro do usuário.	Datetime		

**Curses (Curso)**

<b>Campo</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Default</b>	<b>Observação</b>
Id	Código autoincremento.	Integer		Chave primaria
name	Nome do curso	String		-
teacher_id	Código do professor	Integer		Chave Estrangeira
matter_id	Código de matéria	Integer		Chave Estrangeira
description	Descrição do curso.	String		
value	Valor do curso	Float		-
created_at	Data da criação do registro de professor	Date		-
updated_at	Data de atualização do registro do usuário.	Datetime		

**Recommendations (Recomendação)**

<b>Campo</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Default</b>	<b>Observação</b>
Id	Código autoincremento.	Integer		Chave primaria
rating	Avaliação do usuário.	Integer		-

	Gostou ou não.			
Descrição	Descrição recomendação	String		-
enrollment_id	Código da matrícula (associação de um professor com matéria (curso) com aluno)	Integer		Chave Estrangeira
course_date	Data do curso	Datetime		
created_at	Data da criação do registro de professor	Date		
updated_at	Data de atualização do registro do usuário.	Datetime		

### Enrollments (Matricula)

Campo	Descrição	Tipo	Default	Observação
Id	Código autoincremento.	Integer		Chave primaria
course_id	Código do curso	Integer		Chave Estrangeira
student_id	código do aluno	Integer		Chave Estrangeira
hours	Quantidade de horas	Integer		

	contratadas pelo aluno			
evaluation	Marca se o aluno já fez avaliação do professor referente a essa matricula.	boolean		
created_at	Data da criação do registro de professor	Date		
updated_at	Data de atualização do registro do usuário.	Datetime		

### Contacts (Contatos)

Campo	Descrição	Tipo	Default	Observação
Id	Código autoincremento.	Integer	-	Chave primaria
name	Código do curso	String		Chave Estrangeira
email	Email de contato.	String		Chave Estrangeira
subject	Assunto do contato	Integer		
message	Mensagem..	String		
created_at	Data da criação do registro de professor	Date		



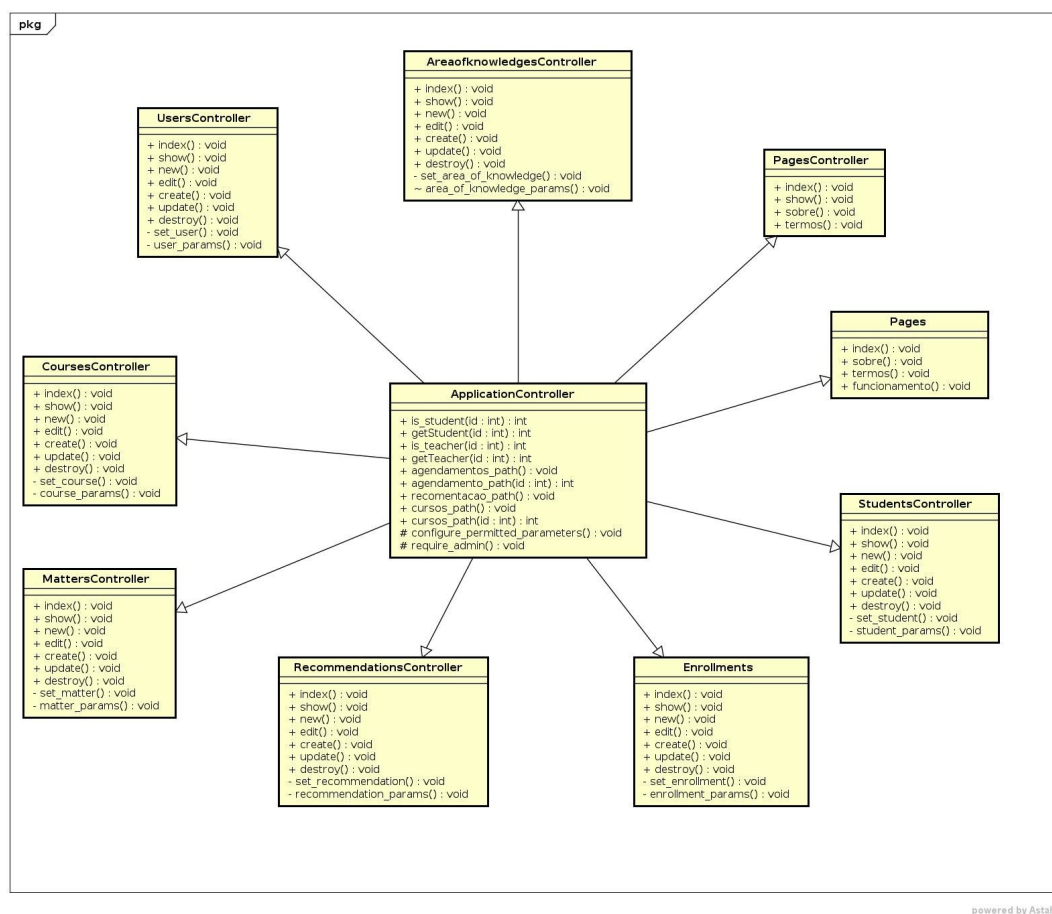
updated_at	Data de atualização do registro do usuário.	Datetime		
------------	---	----------	--	--

## 5.4 Diagrama de classes

Foram elaborados dois diagramas de classes *figura 2 e 3* na qual o primeiro é referente as classes do controller e o segundo as classes do model necessárias ao sistema Reforço mostrando as relações entre as classes, os atributos e operações.

### 5.4.1 Contollers

No diagrama de classes do controllers temos a relação dos controles existentes no sistema.



**Figura - 3 Diagrama de Classe Controller (ANEXO III)**

## 5.4.2 Model

Na representação pelo Diagrama de Classes do Model temos as classes e os seus relacionamentos.

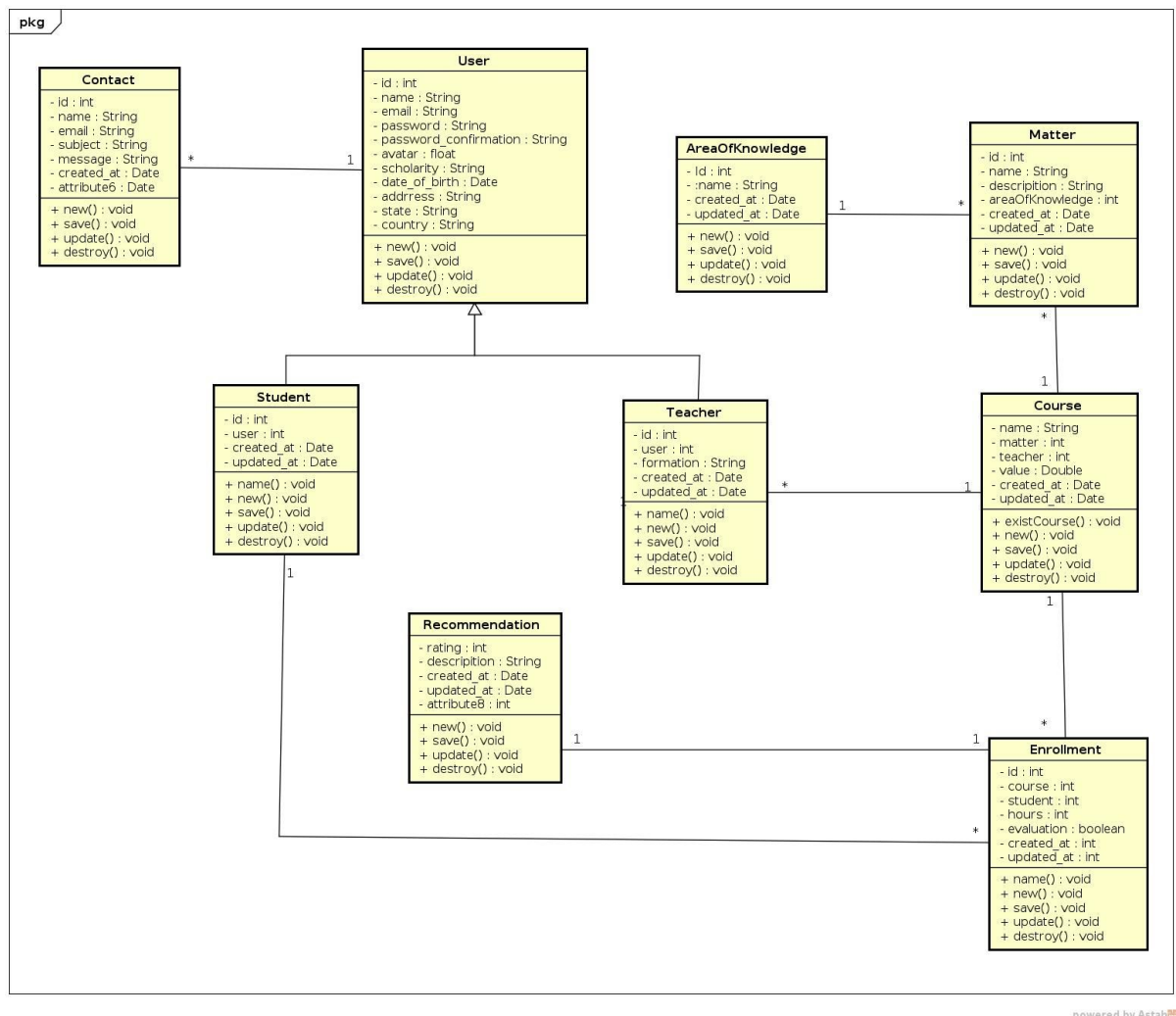
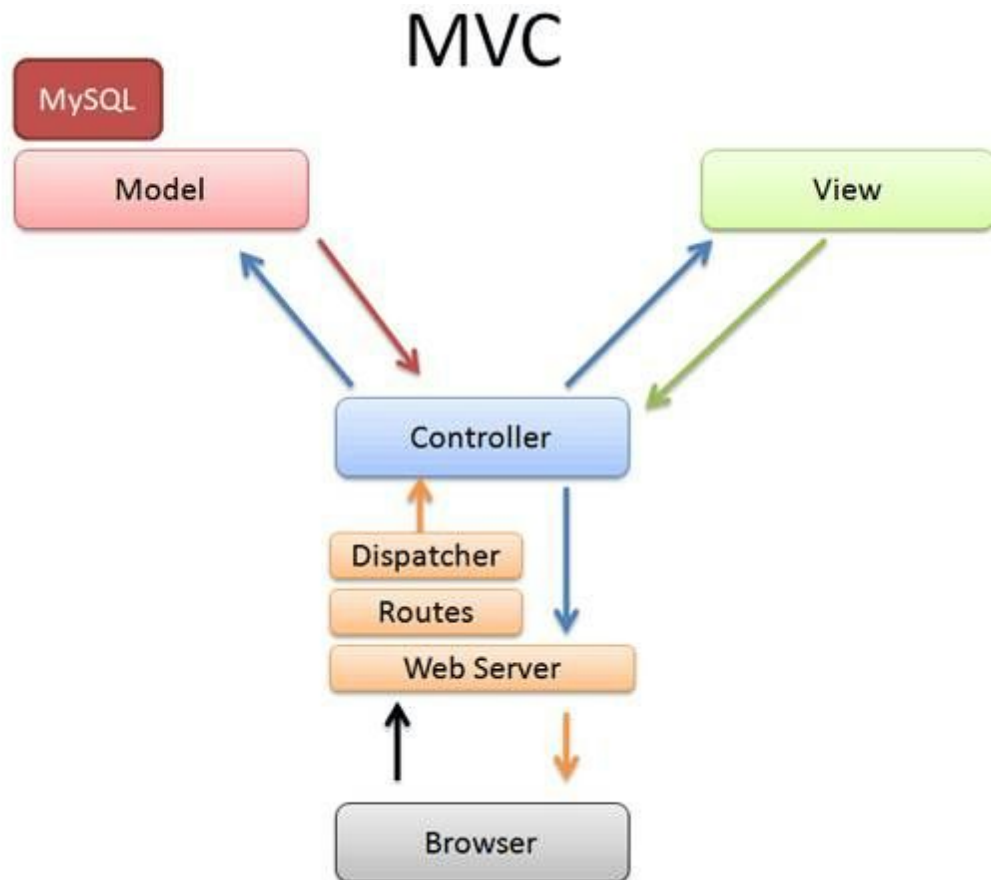


Figura - 4 Diagrama de Classe *Model* (ANEXO IV)

## 6 Visão de implantação

Sendo o software Reforço uma aplicação SaaS, e como afirma Velte et al. (2010), é uma aplicação hospedada em um servidor remoto e acessada pela Internet.



A implantação do sistema será a disponibilização através de um ambiente cloud para a utilização na web não sendo necessário instalação no computador do usuário.

### 6.1 Release Notes

Nessa cessão buscamos mostrar as funcionalidades disponível na versão atual do sistema e as implementações futuras bem como os erros conhecidos.

### **6.1.1 Funcionalidades disponibilizadas - Versão 1.0**

- Área administrativa do portal
- Controle de Alunos;
- Controle de Professores;
- Áreas de Conhecimento;
- Matérias;
- Busca de professores;
- Oferta de cursos por professores (aulas);
- Agendamento de aula
- Sistema de Recomendações dos professores;
- Entre em contato conosco
- Estatísticas

### **6.1.2 Implementações futuras**

- Login com facebook e google;
- Chat para contato entre os usuários do sistema;
- Alertas via e-mail;

### **6.1.3 Erros Conhecidos**

Os erros conhecidos do sistema seguem listados abaixo;

- Falta de validação javascript de alguns campos;
- A pesquisa na barra superior do sistema não esta funcionando;

## **7 Visão de Uso**

Para uma melhor visão de uso do sistema foi criado um manual do usuário que está disponível no ANEXO V desse documento.

O manual contém todas as informações necessárias para que o usuário consiga utilizar de todas as funcionalidades do sistema, seja ele um visitante do site, um aluno, um professor ou o administrador do sistema.

## **8 Revisão do Projeto**

Nessa seção do relatório descreveremos o processo de desenvolvimento, principais problemas e tomadas de decisão em relação ao projeto, as atribuições das atividades e técnicas de gerenciamento, monitoramento e controle.

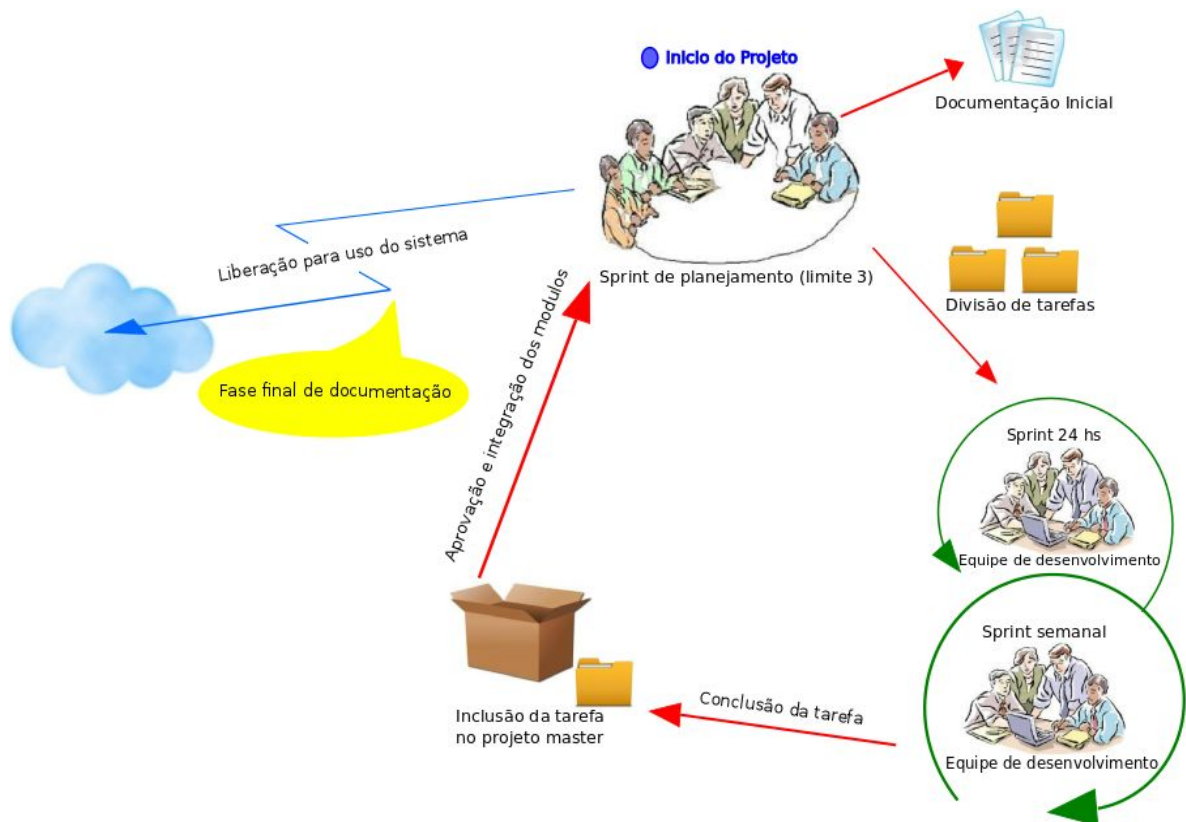
### **8.1 Processo de Desenvolvimento**

Para esse projeto a equipe optou por trabalhar com processo de desenvolvimento ágil uma vez que se dispunha de pouco tempo para a realização do mesmo.

Tendo em vista o processo de desenvolvimento ágil o método usado como base foi o Scrum buscando dividir bem as tarefas e determinando pequenos prazos para as conclusões das mesmas. No processo também se utilizou de técnicas de outras metodologias como a Extreme Programming (XP), uma vez que cada membro da equipe tinha acesso a toda e qualquer parte do código e, acima de tudo, todos tinham o direito de alterar qualquer parte do sistema, a qualquer momento, sem ter que pedir permissão a ninguém, informando as mudanças nas reuniões.

As reuniões foram definidas de modo que acontecesse três grandes reuniões de Sprints de Planejamento onde as mesmas teriam atas, e pequenas reuniões diárias no decorrer do desenvolvimento usando a técnica Stand up meeting (reuniões rápidas).

Na *figura 3* temos a representação do fluxo do processo Scrum utilizado pela equipe.



**Figura - 5** Fluxo do processo *Scrum* adotado pela equipe

Podemos caracterizar as fases do projeto como sendo:

- Elicitação de requisitos
- Análise e validação dos requisitos e preparo da documentação inicial
- Projeto de Sistema e Software
- Lançamento e teste de módulos
- Integração, testes finais e finalização da documentação
- Publicação e operação

## **8.2 Principais Problemas e Tomadas de Decisão**

Três problemas principais foram encontrados durante o decorrer do projeto:

- Criação de uma área administrativa;
- Mecanismo de Controle de autenticação;
- Implementação de layout padrão na aplicação;

Diante do curto prazo para implementar o sistema e do tempo gasto no desenvolvimento dos requisitos funcionais do sistema, foi tomada a decisão de utilizar uma API externa para a criação da área administrativa. Com a “gem rails-admin”, algumas configurações e adaptações conseguimos resolver esta demanda com um terço do tempo previsto para a tarefa.

Quanto a autenticação e controle de sessão optamos por utilizar a gem “Devise” que permite a criação de uma base de usuários ou ainda, que especifiquemos uma entidade existente em nosso modelo para ser utilizada como repositório de usuários para autenticação.

Por fim, tínhamos que desenvolver um layout padrão a ser implementado em todas as views do sistema. Utilizamos o scaffold para geração do código base das views, e por fim, reescrevemos seus estilos utilizando a biblioteca Bootstrap.

## **8.3 Atribuição das Atividades**

Na reunião iniciais foram distribuídas as atividades conforme as habilidades dos membros da equipe ficando registrado nas atas de reuniões de planejamentos. A cada Stand up meeting se verificava o progresso de cada membro podendo se fazer nova alocação das tarefas caso se verificasse a impossibilidade de execução da mesma por algum membro.

Uma tabela com o cronograma do projeto contendo as atividades distribuídas por cada membro da equipe segue no ANEXO V desse relatório.

## **8.4 Técnicas de Gerenciamento**

Por se tratar de um desenvolvimento ágil o gerenciamento das atividades foi feito pelos relatórios de atividades do github e nas reuniões de Sprint semanal, tendo um membro da equipe encarregada por garantir a execução das tarefas no tempo planejado e cobranças de toda equipe.

## **9 Revisão Individual**

### **9.1 Fabiano Matos**

Vejo este projeto como uma oportunidade única para todos os membros da equipe. Neste trabalho tivemos a oportunidade de aplicar o conhecimento e experiência de todos para implementar um sistema web a partir do zero.

Após um início difícil com reuniões, levantamento de requisitos, e definições das tecnologias e metodologias a serem adotadas podemos ver o projeto tomando forma e se transformar em algo concreto.

Trabalhando com scrum como metodologia e o framework RoR - Ruby on Rails para desenvolvimento web ágil, vivenciamos os prós e contras em cada fase do projeto. Dentre as principais dificuldades encontradas posso citar o aprendizado das tecnologias utilizadas, e a ausência formal de um gerente do projeto, prejudicando a definição dos rumos a serem seguidos pelos membros da equipe.

Assim, avalio como positivos os resultados, pois fomos capazes de concluir o projeto apresentando um sistema que contempla os requisitos iniciais, com uma interface simples e amigável, dentro do prazo estabelecido.

### **9.2 Fabrício Matos**

Na minha opinião, esse projeto foi um grande desafio. Sempre fui apaixonado por programação e fui monitor de programação durante minha graduação. O poder de criar as coisas sempre me fascinou e por isso sempre



foi a área que tive mais interesse e afinidade.

Com o passar do tempo, por motivos profissionais, fui obrigado a abandonar as atividades de programador para entrar em outras áreas: Primeiro o suporte e manutenção de equipamentos, depois a gerência de redes e infraestrutura de servidores e por último a gestão de coordenação de equipes de T.I., além do planejamento de compras e contratações da unidade onde trabalhei por cinco longos anos.

Durante todo esse nunca pude retomar as atividades de desenvolvimento até agora, e por esse motivo tal tarefa se tornou especial para mim. Desde a graduação que eu não passava tanto tempo programando como os últimos dois meses e as dificuldades encontradas nas primeiras semanas foram muitas, pois depois de tanto tempo devo confessar que estava bem enferrujado, mais com o passar do tempo as coisas foram se encaixando e tudo voltando ao mesmo ponto de quando me apaixonei por programar.

A nova linguagem de programação trouxe mais desafios, mais também tornou muito mais satisfatório o prazer de construir um sistema todo do zero novamente. Sistema pequeno é claro, mais que me permitiu exercitar vários dos conhecimentos da época de graduação e consolidar muitos outros aprendidos nesta cadeira no curso de Mestrado Profissional do Cin.

Vários projetos pessoais tomam forma agora e tudo graças a motivação e coragem adquiridos e necessários para concluir esse trabalho dentro do prazo. Ao professor Vinicius Garcia somente agradecimentos por me proporcionar essa maravilhosa experiência de auto-conhecimento.

### **9.3 Nelson Barros Cavalcante**

O projeto Reforço foi um bom desafio para nossa equipe, a qual pode aplicar os conceitos de Engenharia de Software aprendidos na disciplina e vividos por todos

nas suas carreiras estudantis e profissionais.

Os trabalhos para o desenvolvimento do *software* foi sem duvida bastante intensos e valiosos para todos, intenso no que diz respeito ao pouco tempo para aprendermos uma nova linguagem de programação, uma ferramenta de serviço de hospedagem e controle de versionamento bem como a criação de uma *software* a serviço (Saas) que não era de domínio de todos os membros da equipe e valioso devido a todo os conhecimentos e experiencias adquiridos.

A aprendizagem no decorrer do desenvolvimento do projeto foi importante para consolidar as teorias de Engenharia de *Software*, pois foi possível passarmos por todas as etapas da construção de um *software* desde sua concepção, passando pelo levantamento dos requisitos, pela elaboração dos documentos iniciais, estudo e construção do ambiente de desenvolvimento, o próprio desenvolver da aplicação até a conclusão do projeto.

O bom desempenho da equipe para o sucesso do projeto atribuo a heterogeneidade dos conhecimentos e pela forma de trabalho individual e coletivo de todos os membros que estavam comprometidos com o propósito de entregar um bom produto com as documentações bem estruturadas.

## **9.4 Rogério**

A experiência na concepção do projeto Reforço foi bastante valiosa no que tange ao conhecimento e experiência na área de Engenharia de Software. Foi possível praticar conceitos de forma real desde a parte da análise de requisitos à implementação e implantação do sistema.

Para o desenvolvimento do projeto, devido ao pouco tempo disponível, optou-se por métodos ágeis, que além de permitir uma precocidade na entrega do produto, trouxe um maior dinamismo nas adaptações necessárias durante a fase de codificação. Tais mudanças são inerentes às mudanças de alguns requisitos, bem como a identificação de novos. Outra grande vantagem desse tipo de metodologia,

foram as frequentes reuniões entre os membros, que muito auxiliou na socialização acerca do andamento e do desenvolvimento das tarefas do projeto. .

Ainda sim, talvez à pouca experiência do grupo e também ao curto espaço de tempo, observou-se como pontos a serem melhorados o desperdício de tempo, causado devido à duplicidade na implementação de alguns casos de uso por diferentes membros da equipe. Nesse sentido, foi possível perceber a necessidade de uma maior atenção na divisão de tarefas, que poderiam ter sido feitas de forma mais clara e com um maior detalhamento.

De toda forma, avalio o desenvolvimento do projeto, como um todo, de forma positiva, pois foi possível alcançar um produto final satisfatório em relação aos requisitos e arquitetura outrora planejados.

## **10 Entrega**

Todos os artefatos do projeto estão disponíveis para análise da seguinte maneira:

- Código fonte e Documentações do sistema: disponível no GitHub no link:
  - <https://github.com/fgm2/reforco>
- Publicado no Heroku:
  - <http://reforco.herokuapp.com/>

## 11 Bibliografia

BECK, Kent. **Programação Extrema (XP) explicada: acolha as mudanças**. In: Porto Alegre: Bookman, 2004.

ELSEN, PETER, R.; VELTE, A. T.; VELTE, T. J. **Cloud Computing A Practical Approach**. FL, United States: Osborne-mcgraw-hil, 2010.352 p.

HARRISON, W., OSSHER, H., TARR, P. **Software Engineering Tools and Environments: A Roadmap**. In Proc. of The Future of Software Engineering, ICSE'2000, Ireland, 2000.

Rails. **Desenvolvimento web sem dor**. <http://www.rubyonrails.com.br/> Acessado em 27/06/2015.

SOARES, Michel S. **Comparação Entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software**. Revista de Ciência da Computação, Lavras, v. 3, n. 1, p 8-13, nov. 2004.