

**UNIVERSIDADE FEDERAL DO PARANÁ**

**BRUNO RICARDO SELLA  
EVANDRO LUIS MACHADO  
FELIPE MACEDO LINHARES  
YURI JUNGLES**

**NETQ – A REDE SOCIAL DOS DOCENTES**

**CURITIBA**

2015

**BRUNO RICARDO SELLA  
EVANDRO LUIS MACHADO  
FELIPE MACEDO LINHARES  
YURI JUNGLES**

**NETQ – A REDE SOCIAL DOS DOCENTES**

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Técnico em Análise e Desenvolvimento de Sistemas, do Setor de Educação profissional e Tecnológica, da Universidade Federal do Paraná.

Orientador: Msc. Razer Anthom Nizer Rojas Montaña.

**CURITIBA  
2015**

## RESUMO

tecnologia está presente cada vez mais no cotidiano de todos nós. Ela está ligada a grandes avanços na área de saúde com novas formas de tratamento, na área de segurança com o constante monitoramento via câmeras, na área educacional, planejamento urbano e muitos outros exemplos. Com o intuito de trazer a tecnologia atual mais perto da área da educação, aliada a uma demanda crescente de armazenamento em nuvem, foi desenvolvido o NETQ – A rede social dos docentes. Este sistema tem como objetivo conectar professores universitários de todo o país para que compartilhem seu conhecimento com colegas. Este compartilhamento é realizado por meio da criação de questões de provas sobre determinado assunto, de comunicação via chat e também de comentários em questões existentes. Muitas vezes o docente gasta muito tempo para montar uma prova, as vezes sem possuir um editor de texto cria-se um problema ainda maior para tal ação, com o NETQ o professor poderá criar provas com alguns cliques.

Não sendo necessária a utilização de nenhum editor complementar. Para que fosse factível tal sistema, durante todo o projeto foi utilizado metodologias e técnicas de desenvolvimento ágil de *software* permitindo uma codificação mais rápida, barata e mensurável.



## **ABSTRACT**

Every day the technology is a bit more in our lives. It's connected to the greatest advances in health studies with new treatment ways, in the security area with constant camera monitoring, in the educational area, urban planning and some other ways. With the aim of bringing the latest technology closer to the educational area, with the growing demand of cloud storing, we developed NETQ - The teachers social network. This system aims to connect college teachers from the whole country for them to share their knowledge with their colleagues. This sharing is made through questions creation for certain matters, chat communication and also comments on existing questions. Frequently teachers spend a lot of time to build a test, sometimes without having any complementary text editor. To prepare this system, along the whole project we used agile methodologies and software development techniques allowing a faster, cheaper and measurable codification.

## LISTA DE FIGURAS

Figura 1 - FASES DO <i>SCRUM</i> .....	17
Figura 2 - CSKELLY 2015 BOOSTRAP GRID 3.....	22
Figura 3 - PADRÃO MVC .....	26
Figura 4 - ESQUEMA DE UM SERVIDOR VIRTUAL PRIVADO .....	27
Figura 5 - ORGANIZAÇÃO DAS SPRINTS NO TRELLO.....	27
Figura 6 - CONTROLE DE UMA <i>SPRINT</i> DENTRO DO TRELLO UTILIZANDO KANBAN .....	28
Figura 7 - SISTEMA DE CONTROLE DE VERSÃO LOCAL .....	29
Figura 8 - SISTEMA DE CONTROLE DE VERSÃO CENTRALIZADO .....	30
Figura 9 - SISTEMA DE CONTROLE DE VERSÃO CENTRALIZADO .....	31
Figura 10 - <i>SPRINT</i> 1 .....	71
Figura 11 - <i>SPRINT</i> 2 .....	72
Figura 12 - <i>SPRINT</i> 3 .....	73
Figura 13 - <i>SPRINT</i> 4 .....	74
Figura 14 - <i>SPRINT</i> 5 .....	75
Figura 15 - <i>SPRINT</i> 6 .....	76
Figura 16 - <i>SPRINT</i> 7 .....	77
Figura 17 - <i>SPRINT</i> 8 .....	78
Figura 18 - <i>SPRINT</i> 9 .....	79
Figura 19 - <i>SPRINT</i> 10 .....	80
Figura 20 - <i>SPRINT</i> 11 .....	81
Figura 21 - <i>SPRINT</i> 12 .....	82
Figura 22 - <i>SPRINT</i> 13 .....	83
Figura 23 - <i>SPRINT</i> 14 .....	83
Figura 24 - <i>SPRINT</i> 15 .....	84
Figura 25 - FLUXO DE ACESSO AO SISTEMA NETQ.....	86
Figura 26 - TELA DE LOGIN. ....	87
Figura 27 - PÁGINA INICIAL DO PROFESSOR(A): “ <i>TIMELINE</i> ” .....	88
Figura 28 - MENU DE FUNCIONALIDADES DO PROFESSOR(A) .....	89
Figura 29 - PÁGINA INICIAL DO ADMINISTRADOR(A): “RELATÓRIOS” .....	91
Figura 30 - MENU DE FUNCIONALIDADES DO “ADMINISTRADOR(A)” .....	92

## LISTA DE QUADROS

Quadro 1 - COMPARATIVO DE <i>SOFTWARES</i> .....	33
Quadro 2 - TABELA DE PRECEDÊNCIA.....	47
Quadro 3 - VISÃO GERAL DO CRONOGRAMA DO PROJETO .....	49
Quadro 4 - CRONOGRAMA DE GERENCIAMENTO DO PROJETO .....	49
Quadro 5 - CRONOGRAMA DE DOCUMENTAÇÃO .....	49
Quadro 6 - CRONOGRAMA DA <i>SPRINT 1</i> .....	50
Quadro 7 - CRONOGRAMA DA <i>SPRINT 2</i> .....	50
Quadro 8 - CRONOGRAMA DA <i>SPRINT 3</i> .....	51
Quadro 9 - CRONOGRAMA DA <i>SPRINT 4</i> .....	51
Quadro 10 - CRONOGRAMA DA <i>SPRINT 5</i> .....	52
Quadro 11 - CRONOGRAMA DA <i>SPRINT 6</i> .....	52
Quadro 12 - CRONOGRAMA DA <i>SPRINT 7</i> .....	52
Quadro 13 - CRONOGRAMA DA <i>SPRINT 8</i> .....	52
Quadro 14 - CRONOGRAMA DA <i>SPRINT 9</i> .....	53
Quadro 15 - CRONOGRAMA DA <i>SPRINT 10</i> .....	53
Quadro 16 - CRONOGRAMA DA <i>SPRINT 11</i> .....	54
Quadro 17 - CRONOGRAMA DA <i>SPRINT 12</i> .....	54
Quadro 18 - CRONOGRAMA DA <i>SPRINT 13</i> .....	54
Quadro 19 - VISÃO GERAL DO CRONOGRAMA REALIZADO .....	56
Quadro 20 - CRONOGRAMA REALIZADO DE GERENCIAMENTO DO PROJETO .....	57
Quadro 21 - CRONOGRAMA REALIZADO DE DOCUMENTAÇÃO.....	57
Quadro 22 - CRONOGRAMA REALIZADO DA <i>SPRINT 1</i> .....	58
Quadro 23 - CRONOGRAMA REALIZADO DA <i>SPRINT 2</i> .....	58
Quadro 24 - CRONOGRAMA REALIZADO DA <i>SPRINT 3</i> .....	59
Quadro 25 - CRONOGRAMA REALIZADO DA <i>SPRINT 4</i> .....	59
Quadro 26 - CRONOGRAMA REALIZADO DA <i>SPRINT 5</i> .....	60
Quadro 27 - CRONOGRAMA REALIZADO DA <i>SPRINT 6</i> .....	60
Quadro 28 - CRONOGRAMA REALIZADO DA <i>SPRINT 7</i> .....	61
Quadro 29 - CRONOGRAMA REALIZADO DA <i>SPRINT 8</i> .....	61
Quadro 30 - CRONOGRAMA REALIZADO DA <i>SPRINT 9</i> .....	62
Quadro 31 - CRONOGRAMA REALIZADO DA <i>SPRINT 10</i> .....	62

Quadro 32 - CRONOGRAMA REALIZADO DA <i>SPRINT</i> 11 .....	63
Quadro 33 - CRONOGRAMA REALIZADO DA <i>SPRINT</i> 12 .....	63
Quadro 34 - CRONOGRAMA REALIZADO DA <i>SPRINT</i> 13 .....	64
Quadro 35 - CRONOGRAMA REALIZADO DA <i>SPRINT</i> 14 .....	64
Quadro 36 - CRONOGRAMA REALIZADO DA <i>SPRINT</i> 15 .....	64
Quadro 37 - QUADRO DE CORRELAÇÃO ENTRE OPÇÃO DO MENU E CASOS DE USO DO PROFESSOR(A). .....	90
Quadro 38 - QUADRO DE CORRELAÇÃO ENTRE OPÇÃO DO MENU E CASOS DE USO DO ADMINISTRADOR(A). .....	92



## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>12</b>
1.1 OBJETIVOS GERAIS .....	13
1.2 OBJETIVOS ESPECÍFICOS .....	13
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>15</b>
2.1 ENGENHARIA DE SOFTWARE .....	15
<b>2.1.1 Scrum</b> .....	<b>16</b>
2.1.1.1 O time de <i>Scrum</i> .....	17
2.1.1.1.1 <i>Product Owner</i> .....	18
2.1.1.1.2 <i>Scrum Master</i> .....	18
2.1.1.1.3 <i>Sprint</i> .....	18
2.1.1.1.4 Planejamento de <i>Sprint</i> .....	19
2.1.1.1.5 Reuniões Diárias .....	19
2.1.1.1.6 Revisão da <i>Sprint</i> .....	20
2.1.1.1.7 Retrospectiva da <i>sprint</i> .....	20
2.1.1.1.8 <i>Backlog</i> do produto .....	20
2.1.1.1.9 <i>Backlog da Sprint</i> .....	21
2.2 DESIGN RESPONSIVO .....	21
<b>2.2.1 Grid System</b> .....	<b>21</b>
2.3 ERGONOMIA .....	22
<b>2.3.1 Usabilidade</b> .....	<b>23</b>
2.4 UML – <i>UNIFIED MODELING LANGUAGE</i> .....	23
<b>2.4.1 Diagramas estruturais</b> .....	<b>24</b>
<b>2.4.2 Diagramas comportamentais</b> .....	<b>24</b>
2.5 IMPLEMENTAÇÃO .....	24
<b>2.5.1 MVC (<i>Model, View e Controller</i>)</b> .....	<b>25</b>
<b>2.5.2 VPS (<i>Virtual Private Server</i>)</b> .....	<b>26</b>
<b>2.5.3 Trello</b> .....	<b>27</b>
<b>2.5.4 Versionamento</b> .....	<b>28</b>
2.5.4.1 Sistemas de controle de versão Locais .....	29
2.5.4.2 Sistemas de Controle de Versão Centralizados .....	29
2.5.4.3 Sistemas de Controle de Versão Distribuídos .....	30
<b>2.5.5 JSF</b> .....	<b>31</b>

<b>3 O SOFTWARE PROPOSTO - NETQ</b> .....	<b>32</b>
3.1 DEMANDAS ENCONTRADAS.....	32
<b>3.1.1 Professor</b> .....	<b>32</b>
3.2 <i>Softwares</i> existentes .....	33
<b>4 METODOLOGIA</b> .....	<b>34</b>
4.1 MODELO DO PROCESSO DE ENGENHARIA DE <i>SOFTWARE</i> .....	34
4.2 METODOLOGIA DE IMPLEMENTAÇÃO.....	35
4.3 REQUISITOS .....	35
<b>4.3.1 Requisitos funcionais</b> .....	<b>36</b>
<b>4.3.2 Requisitos não funcionais</b> .....	<b>43</b>
4.4 PLANO DE ATIVIDADE .....	46
<b>4.4.1 Tabela de precedência</b> .....	<b>46</b>
<b>4.4.2 Cronograma</b> .....	<b>47</b>
4.4.2.1 Cronograma planejado .....	48
4.4.2.2 Cronograma realizado .....	54
4.5 PLANO DE RISCO .....	65
4.6 PLANO DE COMUNICAÇÃO .....	65
4.7 MATERIAIS .....	65
<b>4.7.1 Recursos de <i>hardware</i></b> .....	<b>66</b>
<b>4.7.2 Recursos de <i>Software</i></b> .....	<b>67</b>
4.7.2.1 <i>Softwares</i> de desenvolvimento.....	67
4.7.2.2 <i>Softwares</i> de documentação .....	68
4.8 DESENVOLVIMENTO DO PROJETO .....	68
<b>4.8.1 Definição do <i>backlog</i> do produto</b> .....	<b>69</b>
<b>4.8.2 Divisão do <i>backlog</i> do produto em <i>sprints</i></b> .....	<b>69</b>
4.8.2.1 <i>Sprint</i> 1 .....	70
4.8.2.2 <i>Sprint</i> 2 .....	71
4.8.2.3 <i>Sprint</i> 3.....	72
4.8.2.4 <i>Sprint</i> 4 .....	73
4.8.2.5 <i>Sprint</i> 5.....	75
4.8.2.6 <i>Sprint</i> 6 .....	76
4.8.2.7 <i>Sprint</i> 7 .....	76
4.8.2.8 <i>Sprint</i> 8.....	77
4.8.2.9 <i>Sprint</i> 9.....	78

4.8.2.10 <i>Sprint</i> 10 .....	79
4.8.2.11 <i>Sprint</i> 11 .....	80
4.8.2.12 <i>Sprint</i> 12 .....	81
4.8.2.13 <i>Sprint</i> 13 .....	82
4.8.2.14 <i>Sprint</i> 14 .....	83
4.8.2.15 <i>Sprint</i> 15 .....	84
<b>5 APRESENTAÇÃO DO SOFTWARE .....</b>	<b>85</b>
5.1 Interfaces e Funcionalidades.....	85
<b>5.1.1 Perfil do professor(a) .....</b>	<b>87</b>
<b>5.1.2 Perfil do administrador(a).....</b>	<b>90</b>
<b>6 CONSIDERAÇÕES FINAIS .....</b>	<b>93</b>
<b>6.1.1 CONCLUSÃO .....</b>	<b>93</b>
6.2 Implementações Futuras .....	94
<b>REFERÊNCIAS.....</b>	<b>96</b>

## 1 INTRODUÇÃO

Com a disseminação da *internet* no mundo inteiro, a interação social tecnológica entre as pessoas está cada vez mais presente no dia a dia. Esta interação se dá principalmente por meio de e-mails (gmail, hotmail, e-mails corporativos), redes sociais pessoais (facebook, hi5, google+) e também redes sociais profissionais como o LinkedIn. Usada com objetivo educacional, essa facilidade de comunicação já trouxe e continua trazendo grandes avanços. Resultados de pesquisas podem ser divulgados, novas teorias estão sendo apresentadas e o conhecimento técnico tem sido compartilhado.

Considerando a crescente utilização da tecnologia para fins educacionais associada ao aumento do armazenamento em nuvem observou-se a necessidade da criação de um sistema que possa contribuir para esta importante área por meio do desenvolvimento de um sistema web que proporcione:

- Interação social: professores poderão interagir entre si, seguir profissionais de seu interesse, vendo assim suas atualizações, bem como trocar mensagens privadas;
- Armazenamento em nuvem: cadastro e utilização de questões de provas/testes/avaliações no próprio sistema web, podendo deixá-las privadas ou públicas, assim, centralizando-as em um só lugar;
- Compartilhamento de conhecimento: cada pergunta cadastrada pode receber comentários a respeito da mesma, abrindo lugar para discussão e troca de conhecimento sobre o assunto abordado pela questão.

Este documento explica detalhadamente a vários aspectos do projeto. Este está dividido em capítulos da seguinte maneira:

O primeiro deles, visa apresentar os objetivos gerais e específicos da equipe com este projeto, introduzindo o leitor no ambiente necessário para que ele saiba o que esperar do projeto em questão.

No capítulo dois é apresentada toda a fundamentação teórica, dando um resumo de todas as técnicas e tecnologias utilizadas para o planejamento e desenvolvimento de todo o projeto, assim como das principais características adotadas da metodologia ágil de desenvolvimento de software.

No capítulo três é apresentado o software proposto, demonstrando as necessidades encontradas que nos levaram à criação deste software. Também há uma sessão que apresenta outros softwares do mercado que têm propostas parecidas ou similares, mostrando as suas diferenças em relação ao NetQ.

O quarto capítulo tem a intenção de apresentar tudo que foi feito para a geração do produto final, desde metodologia utilizada, requisitos iniciais e cronograma de atividades até as *Sprints* realizadas e recursos de *hardware* utilizados.

O capítulo de número cinco trata da apresentação do *software* desenvolvido, contendo capturas de tela de partes estratégicas do sistema para facilitar o entendimento completo das funcionalidades do mesmo.

As considerações finais se dão no capítulo 6, onde abordamos uma conclusão geral do projeto comparando o *software* proposto com o que realmente foi desenvolvido e também citando na sessão de Implementações Futuras algumas sugestões de funcionalidades que não foram implementadas neste primeiro momento mas com certeza agregarão um grande valor ao sistema como um todo caso sejam futuramente implementadas.

## 1.1 OBJETIVOS GERAIS

O objetivo desse trabalho é construir uma rede social capaz de promover a interação, cadastro de questões, criação de provas automáticas e troca de conhecimento entre professores universitários. A proposta engloba todos os docentes do ensino superior brasileiro.

## 1.2 OBJETIVOS ESPECÍFICOS

Este trabalho é composto pelos objetivos específicos:

- Cadastrar professores de ensino superior;
- Permitir a interação entre professores cadastrados, por meio de mensagens privadas;

- Compartilhar conhecimento entre os usuários por meio de perguntas cadastradas e compartilhadas pelos professores;
- Permitir a criação de avaliações a partir das perguntas que estão na base de dados de cada usuário;
- Criar modelos de provas com as questões e com as alternativas embaralhadas;
- Gerar relatórios das questões mais utilizadas para fins analíticos.

## 2 FUNDAMENTAÇÃO TEÓRICA

Serão apresentados ao longo dos tópicos as tecnologias utilizadas bem como as técnicas e as pesquisas realizadas para o desenvolvimento do projeto. As obras, artigos e buscas utilizadas para consulta também farão parte dessa documentação sendo destacadas como referencias.

### 2.1 ENGENHARIA DE SOFTWARE

Engenharia de *software* é a disciplina relacionada a todos os aspectos de produção de *software*, envolvendo questões técnicas e não técnicas como: especificação do conhecimento, técnicas de projeto e implementação e gestão de projetos (PRESSMAN, 2011). Engenharia de *Software* é o nome dado a ação que tem como objetivo definir exercitar as seguintes ações com o objetivo de atender as necessidades do cliente:

- Processos;
- Métodos;
- Ferramentas;
- Ambientes.

Segundo Sommerville (2003), “Processos de software são o conjunto de atividades e resultados associados que levam à produção de um produto de software”.

Para a produção desse produto de *software* existem alguns modelos de processos que servem para ajudar a definir a sequência que as atividades serão realizadas e também são utilizados para explicar diferentes abordagens do desenvolvimento. Para a realização desse projeto, foi utilizado o modelo de processo *Scrum*, que está explicado no item 2.1.1.

Os modelos de processos de software possuem 4 etapas principais que servem como base no processo de desenvolvimento do sistema (SCHWARTZ,1975):

- Análise e especificação de requisitos;
- Projeto de Sistema;
- Codificação;
- Verificação e Integração.

### 2.1.1 Scrum

O *Scrum* é um modelo de desenvolvimento ágil de *software*, que segundo Steffen (2012), “É uma nova forma de gestão e desenvolvimento de *software* que usa uma abordagem de planejamento e execução iterativa e incremental”. De acordo com o Beck *et al.* (2001) este modelo possui quatro premissas:

- Indivíduos e interações acima de processos e ferramentas;
- *Software* funcional acima de documentação compreensiva;
- Colaboração do cliente acima de negociação contratual;
- Resposta a mudanças acima de seguir um plano.

Este modelo tem como características aumentar o esforço no desenvolvimento e minimizar na documentação. Com entregas frequentes, que normalmente são de uma a quatro semanas, chamadas *sprints*, o cliente acaba se tornando parte da equipe de desenvolvimento, o que diminui os custos do projeto (CONH,FORD, 2003). A evolução do projeto e da equipe são discutidos diariamente, no lugar de reuniões longas e semanais ele adota reuniões rápidas, objetivas e diárias. Essas, devem começar sempre no mesmo horário e responder as três perguntas a seguir:

- O que fiz ontem?
- O que estou planejando para fazer amanhã?
- Existe algo me impedindo de realizar minha meta?



Segundo Schwaber e Sutherland (2013), todo o trabalho no *scrum* é realizado em iterações chamadas de *sprints*.

A FIGURA 1 mostra como funciona o *scrum* e onde o *product backlog* (requisitos do sistema obtidos anteriormente) é dividido em pequenas partes. Essas pequenas partes (*sprint backlog*) engloba o conteúdo que será desenvolvido naquela *sprint*, que pode durar de uma a quatro semanas. Dentro da *sprint* acontecem reuniões diárias que são chamadas de *daily meeting*.



FIGURA 1 - FASES DO SCRUM  
FONTE: CISNEIROS, 2009

#### 2.1.1.1 O time de Scrum

O time de *scrum* é a equipe de desenvolvimento, normalmente composta por poucos membros entre seis e dez pessoas. Constitui de programador, designer,

analista de testes ou arquiteto. Apesar do time de *scrum* ser responsável por atingir os objetivos da *sprint*, ele deve ter maturidade para controlar todo o processo, sendo de sua responsabilidade mostrar ao final de cada *sprint* o resultado do trabalho.

#### 2.1.1.1.1 Product Owner

O *product owner* é o cliente de um time *scrum*. Ele deve definir todas as funcionalidades que o *software* deve possuir (*product backlog*), definir e priorizar o *sprint backlog* que serão incluídos em cada *sprint*, de acordo com o valor de negócio e aceitar ou rejeitar as entregas realizadas pelo time Schwaber e Sutherland (2013).

#### 2.1.1.1.2 Scrum Master

O *scrum master* objetiva assegurar que a equipe siga as práticas do *scrum* e que não se comprometa excessivamente com relação à aquilo que é capaz de realizar durante uma *sprint*, atuando como um gerente de projeto ou líder técnico. Ele também conduz o *daily meeting* e torna-se responsável por remover obstáculos que sejam apresentados pela equipe durante essas reuniões (LIBARDI , 2010).

#### 2.1.1.1.3 Sprint

*Sprint* é o espaço de tempo que dura entre duas a quatro semanas em que um produto de valor e usável é criado. Ao final de uma *sprint* outra sempre é iniciada, até que não haja mais *backlogs* no projeto. No decorrer das semanas que englobam uma *sprint*, obrigatoriamente ocorrem quatro eventos Schwaber e Sutherland (2013) , estes estão explicados nos itens 2.1.1.1.4 , 2.1.1.1.5 , 2.1.1.1.6 e 2.1.1.1.7, são eles:

- Planejamento de *sprint* (*sprint planning*);

- Reunião diária (*daily meeting*);
- Revisão da *sprint*;
- Retrospectiva da *sprint*.

#### 2.1.1.1.4 Planejamento de *Sprint*

O planejamento de *sprint* é a primeira atividade a ser realizada. Ele é uma reunião onde estão presentes todos os envolvidos no projeto : *scrum team*, *scrum master* e *product owner*. Durante essa reunião o *product owner* descreve as funcionalidades de maior prioridade para a equipe, essa por sua vez, faz perguntas até que não restem mais dúvidas para que possam quebrar essas funcionalidades em tarefas técnicas após a reunião Schwaber e Sutherland (2013).

Após as definições das tarefas técnicas (*backlog* do produto) o *product owner* e o *time de scrum* priorizam um objetivo para um *sprint*, o sucesso será avaliado ao fim na retrospectiva Schwaber e Sutherland (2013).

#### 2.1.1.1.5 Reuniões Diárias

As Reuniões diárias também são conhecidas como *daily meeting*, acontecem todos os dias e no mesmo horário e devem ter uma duração máxima de 15 minutos. São realizadas nos primeiros minutos da jornada de trabalho. O objetivo dessa técnica é disseminar o conhecimento sobre o que foi feito no dia anterior, identificar problemas que possam estar acontecendo e priorizar no que vai ser feito no dia que se inicia Schwaber e Sutherland (2013)..

#### 2.1.1.1.6 Revisão da *Sprint*

A revisão da *sprint* é realizada sempre que o ciclo da *sprint* termina, ela acontece em uma reunião onde o *time scrum* apresenta o que foi alcançado com seu trabalho para o *product owner*, gerência e clientes. Os itens avaliados durante essa prática são os que foram definidos como requisitos durante o *sprint planning* Schwaber e Sutherland (2013)..

#### 2.1.1.1.7 Retrospectiva da *sprint*

A retrospectiva da *sprint* assim como a revisão, ocorre ao final de um ciclo, ela serve para identificar o que funcionou bem, o que pode ser melhorado e quais as ações necessárias para isso.

“Na reunião de retrospectiva da *sprint* é feita a revisão do processo de desenvolvimento, de forma a torná-lo eficaz e gratificante para a próxima iteração” (SCHWABER, SUTHERLAND, 2013).

#### 2.1.1.1.8 *Backlog* do produto

O *product backlog* ou *backlog* do produto é uma única lista que contém todos os itens necessários para o desenvolvimento ou incremento do produto (*software*). Nele estão contidos os requisitos não funcionais e funcionais, mudanças, e *bugs* encontrados durante os testes. O *product owner* é o responsável pela criação dessa lista. De acordo com Schwaber e Sutherland (2013), este artefato está em constante mudança, uma vez que seu objetivo é manter um registro concreto de todas as funcionalidades, requisitos, melhorias e atributos que envolvem esse novo incremento do sistema Schwaber e Sutherland (2013)..

#### 2.1.1.1.9 Backlog da Sprint

O *backlog* da *sprint* são as tarefas retiradas do *backlog* do produto pelo time de *scrum* para que sejam realizadas em uma *sprint*. Ele é escolhido de acordo com as prioridades definidas pelo *product owner*, assim como a percepção do tempo que será necessário para realizá-las. O *sprint backlog* não pode possuir itens pendentes ao final de uma *sprint*, ele é acompanhado e atualizado diariamente pelo *scrum master* com o intuito de mostrar quais as tarefas que já foram completadas e as que ainda não. Esse acompanhamento é de extrema importância pois com ele pode-se realizar mudanças do esforço dentro da equipe caso necessário ( PEREIRA, 2007).

## 2.2 DESIGN RESPONSIVO

O *design* responsivo adapta o leiaute da aplicação à página, dependendo do tamanho da tela do dispositivo que renderiza ou a abre. Com o aumento da utilização de dispositivos móveis, houve a necessidade de adaptar as aplicações e web sites para que pudessem ser visualizados e utilizados nesse novo tipo de tecnologia. A partir desta nova premissa, foi criado o *design* responsivo (FRAIN, 2012).

### 2.2.1 Grid System

*Grid system* é uma estrutura que permite criar leiautes de páginas com uma série de linhas verticais e horizontais, realiza a medição da tela do dispositivo utilizado para acessar a aplicação e com isso o leiaute de um sistema se adapta ao tamanho da tela. Essa técnica é utilizada originalmente no modelo do *framework bootstrap*.

A quantidade de divisões feitas na tela varia de acordo com a ferramenta utilizada, sendo que as mais populares normalmente dividem a tela em doze partes iguais que, por sua vez, são recursivamente divididas em outras doze partes (BOOTSTRAP, 2015).

Um exemplo dessa divisão pode ser vista na FIGURA 2.

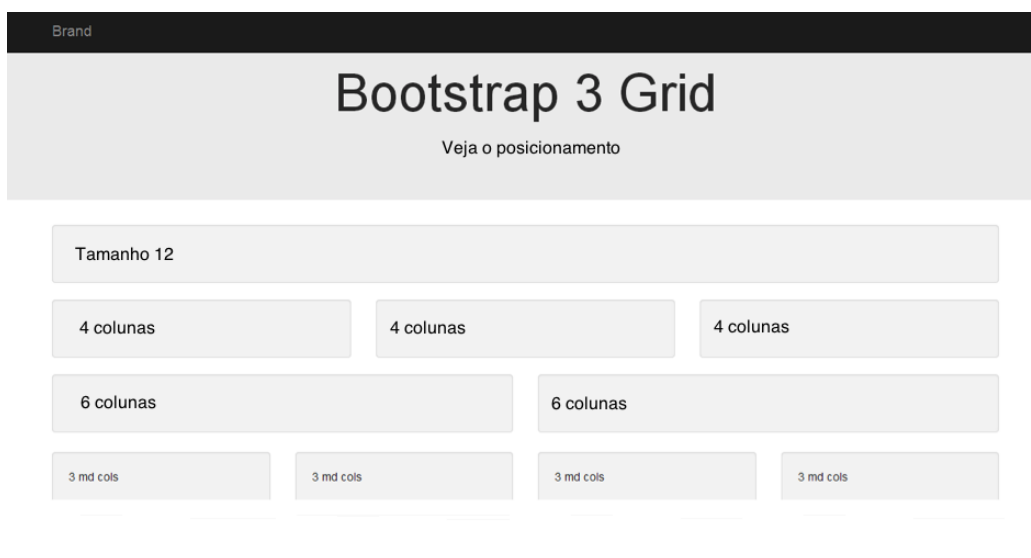


FIGURA 2 - CSKELLY 2015 BOOTSTRAP GRID 3  
FONTE: <http://www.bootstrapzero.com/bootstrap-template/grid>

## 2.3 ERGONOMIA

Ergonomia é uma disciplina científica relacionada ao entendimento das relações entre os seres humanos com outros elementos ou sistemas, e a aplicação dos princípios ergonômicos bem como seus métodos nos projetos tem como objetivo otimizar o bem estar humano e o desempenho global do sistema (IEA, 2015). Os ergonomistas contribuem para o planejamento, projeto e a avaliação de tarefas, postos de trabalho, produtos, ambientes e sistemas de modo a torná-los compatíveis com as necessidades, habilidades e limitações das pessoas.”

### 2.3.1 Usabilidade

Usabilidade é um atributo de qualidade que avalia quão fácil uma interface de usuário é de se usar. Ela também faz referência para métodos que buscam melhorar e otimizar essa interface na fase de *design*, deixando – a o mais intuitiva possível.

A usabilidade está distribuída à diversos elementos, sendo tradicionalmente associada, segundo Nielsen (1993), aos seguintes fatores:

- Facilidade de aprendizagem: o sistema deve ser fácil de assimilar pelo utilizador, para que este possa começar a trabalhar rapidamente;
- Eficiência: o sistema deve ser eficiente para que o utilizador, depois de o saber usar, possa atingir uma boa produtividade;
- Facilidade de memorização: o sistema deve ser facilmente memorizado, para que depois de algum tempo sem o utilizar, o utilizador se recorde como usá-lo;
- Segurança: o sistema deve prever erros, evitar que os utilizadores os cometam e, se o cometerem, permitir fácil recuperação ao estado anterior.
- Satisfação: o sistema deve ser usado de uma forma agradável, para que os utilizadores fiquem satisfeitos com a sua utilização.

### 2.4 UML – *UNIFIED MODELING LANGUAGE*

A linguagem unificada de modelagem, UML ou *Unified Modeling Language* é uma linguagem gráfica de modelagem utilizada para visualizar, especificar, construir e documentar os artefatos de sistemas de objetos (GILLEANES, 2007). A UML não tem como objetivo definir como o trabalho deve ser desenvolvido e sim o que deve ser feito.

Ela é subdivida em duas categorias, a de diagramas estruturais e comportamentais.

### 2.4.1 Diagramas estruturais

Os diagramas estruturais dizem respeito à arquitetura do sistema. Tem como principais objetivos identificar entidades de dados, informações de interesse e componentes englobados pelo sistema (BEZERRA, 2015).

Inclui diagrama de classe, diagrama de objeto, diagrama de componentes, diagrama de estrutura composta, diagrama de pacote e diagrama de utilização (FOWLER, 2014).

### 2.4.2 Diagramas comportamentais

Os diagramas comportamentais são diagramas que dizem respeito à interação entre os elementos que compõe o sistema e seus comportamentos, como às regras de negócio envolvidas (GUEDES, 2009). Os diagramas de comportamento são: diagrama de caso de uso, diagrama de máquina de estados e diagrama de atividades. Em sua subcategoria "Interação" estão inclusos os diagramas de sequência, comunicação, visão geral de interação e por ultimo, porém não menos importante o de temporização (GUEDES, 2009).

## 2.5 IMPLEMENTAÇÃO

Durante o desenvolvimento do produto (*software*) foram utilizadas alguns padrões, técnicas e tecnologias que estão descritos nessa seção.



### 2.5.1 MVC (*Model, View e Controller*)

MVC é um padrão de arquitetura que visa separar uma aplicação, ou até mesmo um pedaço da interface de uma aplicação em três partes: Modelo (*model*), Visão (*view*) e Controlador (*Controller*) (LIU, 2010). Tem como objetivo separar dados e lógicas de negócio (Modelo) da interface do usuário (Visão) e do fluxo da aplicação (Controlador), permitindo que uma mesma lógica de negócios possa ser acessada e visualizada através de várias interfaces.

- Modelo: é a parte do aplicativo que lida com a lógica para os dados do aplicativo, inserindo, buscando, deletando e alterando dados;
- Visão: É a interface que interage com o usuário, responsável por tudo o que o usuário visualiza, também é a camada visão que recebe os dados, informações e comandos inseridos pelo usuário e os envia para o controlador;
- Controlador: É a interface do sistema que possui as regras de negócio, ela é responsável por controlar todo o fluxo de informação que passa pelo sistema, realizando a comunicação entre a visão e o controlador.

A FIGURA 3 demonstra como funciona o padrão MVC.

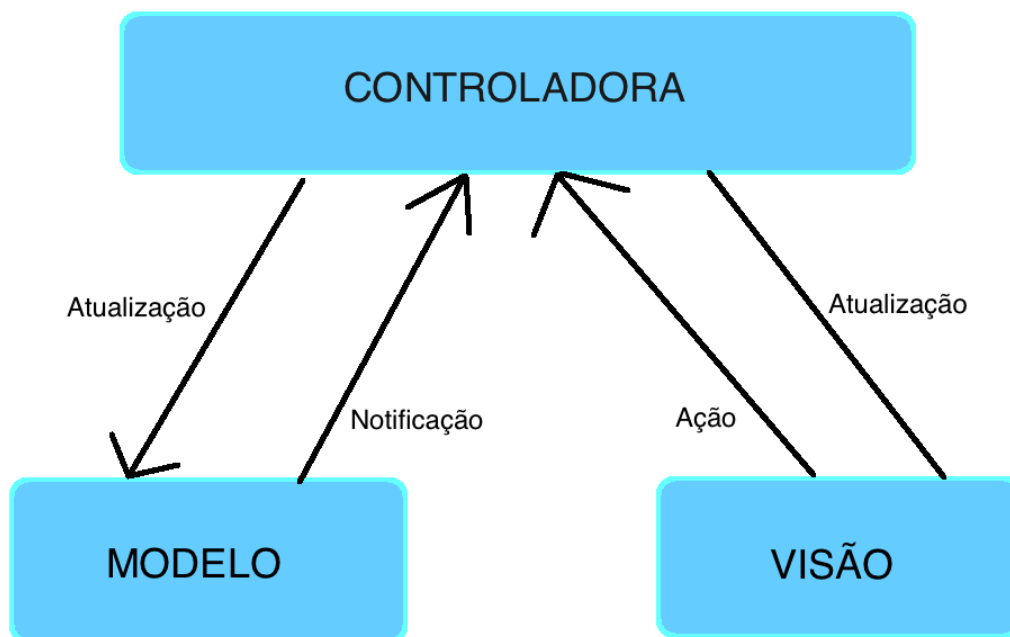


FIGURA 3 - PADRÃO MVC  
FONTE (OS AUTORES, 2015)

### 2.5.2 VPS (Virtual Private Server)

*Virtual Private Server* é o termo em inglês que significa servidor virtual privado. Trata-se do fornecimento parcial dos recursos de uma máquina em nuvem. Cria-se então, a partir destes recursos, uma máquina virtual (FIGURA 4), nesta são instalados todos os recursos necessários para que a aplicação possa ser executada. O acesso à esta máquina virtual pode ser realizado via SSH, protocolo de criptografia de rede.



FIGURA 4 - ESQUEMA DE UM SERVIDOR VIRTUAL PRIVADO  
 FONTE : VIRTIX, 2014.

### 2.5.3 Trello

O Trello é uma ferramenta de colaboração que organiza seus projetos em quadros, exatamente como a metodologia Kanban (TRELLO, 2015). O sistema Trello informa o que está sendo trabalhado, quem está trabalhando em quê, e onde algo está em um processo. A organização das *sprints* no Trello poder ser vista na FIGURA 5.

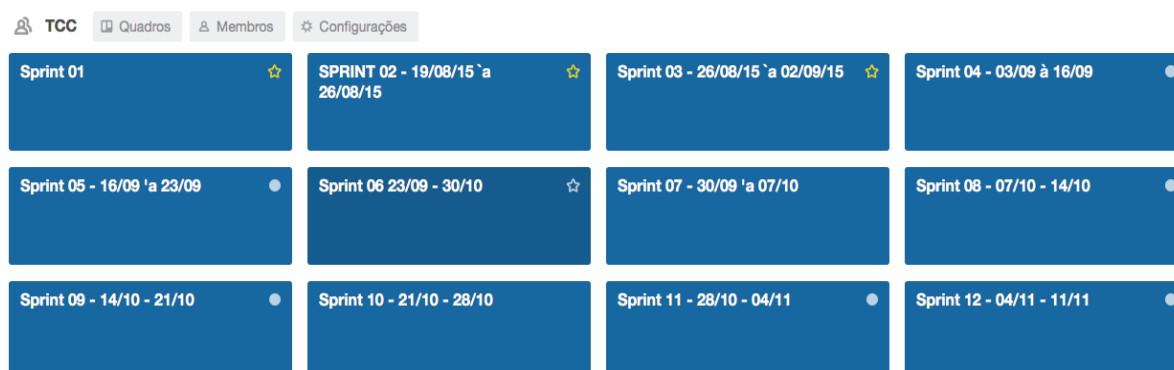


FIGURA 5 - ORGANIZAÇÃO DAS SPRINTS NO TRELLO  
 FONTE: TRELLO, 2015

O Kanban é uma simbologia visual utilizada para registrar ações. É uma prática de controle de fluxo a partir da utilização de cartões, chamado também de gestão visual (KNIBERG *et al.*, 2009) Quando refere-se ao desenvolvimento de *software*, estes cartões representam o *backlog* da *sprint*, ou seja, todos os itens que deverão ser criados naquele período de tempo estimado (*sprint*). Os cartões são colocados em quadro, esse quadro é subdividido em três etapas: A fazer (*to do*), fazendo (*doing*) e feito (*done*). No começo da *sprint* todas as atividades são colocadas no "a fazer", quando um membro da equipe começa a desenvolvê-la deve colocar o cartão na divisão do "fazendo" e, assim que a complete, colocar no "feito". Esse fluxo está demonstrado na FIGURA 6.

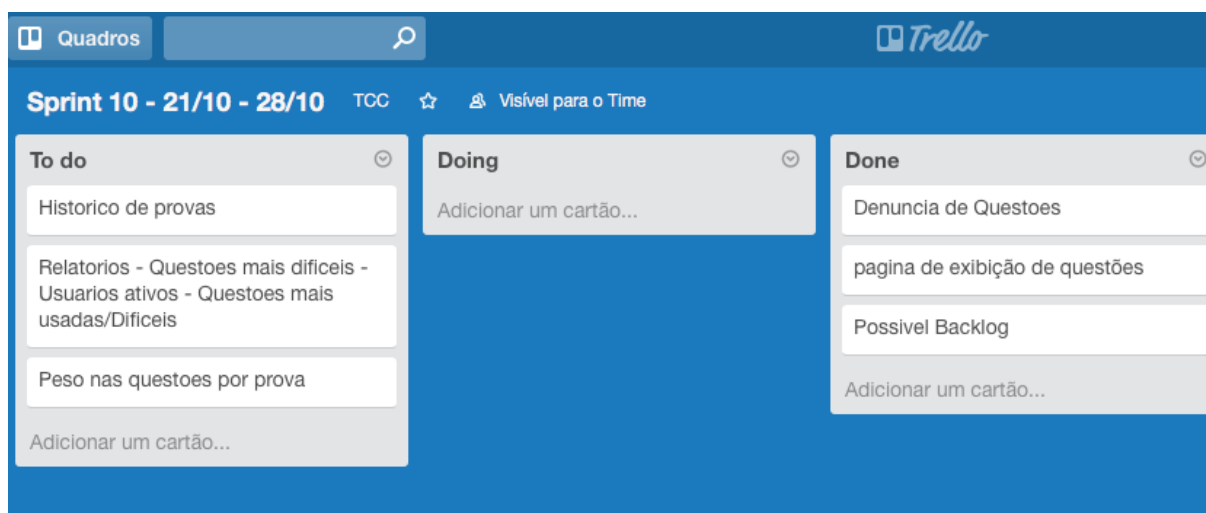


FIGURA 6 - CONTROLE DE UMA *SPRINT* DENTRO DO TRELLO UTILIZANDO KANBAN  
FONTE: TRELLO, 2015

#### 2.5.4 Versionamento

Segundo Chacon, S. e Straub, B (2014), versionamento ou sistema de controle de versões, é um *software* capaz de registrar as mudanças dos conteúdos de um arquivo ou conjunto de arquivos, permitindo que o usuário possa retornar à uma versão específica.

“(...)Ele permite reverter arquivos para um estado anterior, reverter um projeto inteiro para um estado anterior, comparar mudanças feitas ao decorrer do tempo, ver quem

foi o último a modificar algo que pode estar causando problemas, quem introduziu um bug e quando, e muito mais.” (CHACON, STRAUB, 2014)

Os sistemas de controle de versão podem locais, centralizados ou distribuídos.

#### 2.5.4.1 Sistemas de controle de versão Locais

O sistema de controle de versão local permite que o usuário consiga gerenciar as versões de um arquivo ou conjunto de arquivos em sua máquina local de como mostra a FIGURA 7 .

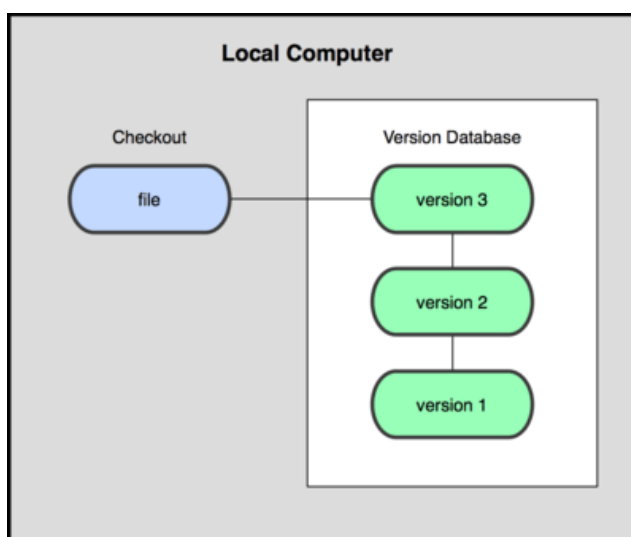


FIGURA 7 - SISTEMA DE CONTROLE DE VERSÃO LOCAL  
FONTE: GIT, 2015

#### 2.5.4.2 Sistemas de Controle de Versão Centralizados

Diferentemente do sistema de controle de versão local, o sistema de controle de versão centralizado constitui-se de um servidor contendo a base de dados de versão e vários clientes acessando este servidor para obter os arquivos. Permitindo, assim, o controle de versão para mais de uma pessoa. A FIGURA 8 apresenta o funcionamento deste sistema:

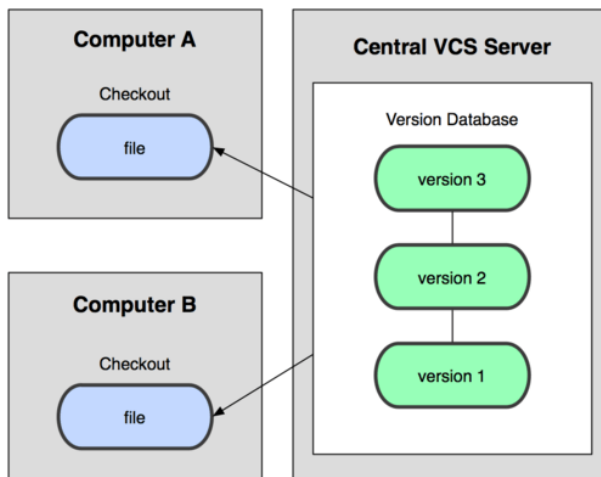


FIGURA 8 - SISTEMA DE CONTROLE DE VERSÃO CENTRALIZADO  
FONTE: GITa, 2015

#### 2.5.4.3 Sistemas de Controle de Versão Distribuídos

Em um sistema de controle de versão distribuído, não é realizado apenas a cópia dos arquivos do servidor central, mas também realiza-se uma cópia completa do repositório (FIGURA 9). Fazendo com que, caso de algum tipo de problema no servidor central ocorra, exista a possibilidade de recuperar todo o repositório. Este sistema de controle é o mais utilizado atualmente e os softwares comerciais mais conhecidos são Git, Mercurial, SVN, entre outros.

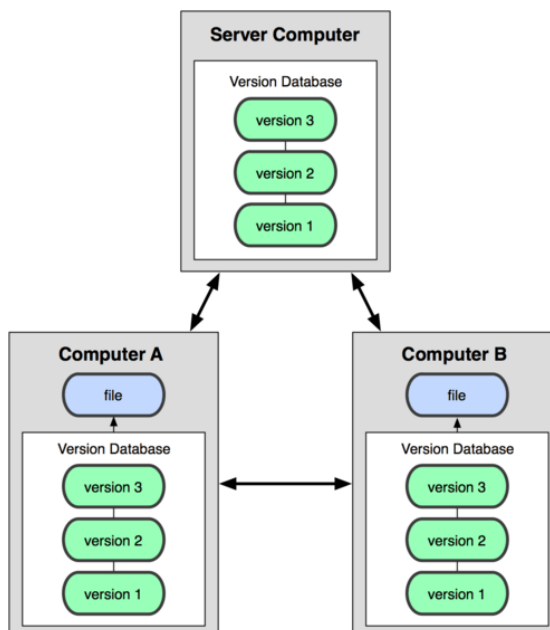


FIGURA 9 - SISTEMA DE CONTROLE DE VERSÃO CENTRALIZADO  
FONTE: GITb, 2015

### 2.5.5 JSF

Java Server Faces é um framework para desenvolvimento Java. A tecnologia JSF inclui um conjunto de bibliotecas para representar componentes de interface com o usuário e gerenciar seus estados, tratar eventos, manipulação de entrada de dados, suporte à nacionalização e acessibilidade (ORACLE, 2015).

JSF foi projetado para ser flexível e potencializar a interface com usuário existente sem limitar desenvolvedores a uma linguagem de marcação particular.

O objetivo principal do JSF é a facilidade de uso, sua arquitetura define claramente a separação entre a lógica da aplicação e de apresentação, facilitando a conexão da camada de visão com o código.

### 3 O SOFTWARE PROPOSTO - NETQ

O NETQ – A rede social dos professores, tem como objetivo atender as necessidades encontradas pelos professores universitários, conectar profissionais com os mesmos interesses além de ser um instrumento de compartilhamento de conhecimento. No cenário atual existem *softwares* que permitem o armazenamento de questões em nuvem, outros permitem uma interação social como o *Facebook*, mas nenhum possui as características combinadas como proposto pela rede social NETQ.

#### 3.1 DEMANDAS ENCONTRADAS

Durante as aulas, foi notado a dificuldade que professores encontram hoje com relação ao armazenamento de questões de provas bem como a geração de provas. Nesta seção estão apresentadas as demanda identificadas por esses profissionais.

##### 3.1.1 Professor

Professores, no geral, tem uma grande base de questões para criação de avaliações, essas normalmente armazenadas em arquivo texto. Se o professor ministra mais de uma matéria, acaba tendo um documento para cada assunto, a fim de não misturar essas questões e facilitar a sua localização. Este cenário gera a demanda de criação de questões online e organizadas por categorias.

Esses arquivos de documentos que contém as questões, devem estar sempre acessíveis ao profissional, mas sua forma de armazenamento, na maioria das vezes vezes causa dificuldade de acesso. Diante disto, o conceito do armazenamento em nuvem foi aplicado.



A criação de provas demanda muito tempo, pois requerem os seguintes passos: seleção das questões que serão colocadas na prova, seleção dos níveis de dificuldade de cada uma, ajuste dessas questões em um único documento para que possam ser impressas, eventual criação de questão na ultima hora ou busca desta na internet sem confiabilidade ou garantia da procedência.

Outra demanda encontrada foi a falta de tempo para conversar com profissionais de outras instituições de ensino para compartilhar conhecimento, bem como a falta de grupos de discussões para discutir sobre questões da área de interesse.

### 3.2 Softwares existentes

Realizando algumas pesquisas, foram encontrados *softwares* que já existem e cumprem uma ou mais das necessidades encontradas, mas nenhum deles consegue cumprir todas as necessidades expostas na sessão anterior.

Dentre todos os sistemas encontrados, vale mencionar três deles: Pral, Questões de Concursos e SGP – Sistema de Gestão de Provas.

Como mencionado anteriormente, cada um deles foca em uma ou duas das necessidades encontradas. Abaixo segue um quadro que lista as funcionalidades retiradas das necessidades indicadas e compara qual *software* à atende ou não.

Funcionalidade	Pral	Questões de Concursos	SGP	NetQ
Armazenamento de questões da nuvem	Não atende	Atende	Atende	Atende
Organização de questões por categoria	Não atende	Atende	Não atende	Atende
Criação de provas selecionando questões	Não atende	Não atende	Atende	Atende
Geração de provas com questões aleatórias com ordem e alternativas embaralhadas	Não atende	Não atende	Não atende	Atende
Integração social entre professores para compartilhamento de conhecimento.	Atende	Não atende	Não atende	Atende
Cadastro de desempenho de alunos para futura geração de relatórios.	Não atende	Não atende	Atende	Atende
Correção automática de provas	Não atende	Não atende	Atende	Não atende
Integração entre alunos e professores para compartilhamento de conhecimento	Atende	Não atende	Não atende	Não atende

QUADRO 1 - COMPARATIVO DE SOFTWARES  
 FONTE: OS AUTORES, 2015

## 4 METODOLOGIA

Devido à complexidade envolvida para o desenvolvimento da rede social e de suas funcionalidades, foi criado o *backlog* do sistema baseado no levantamento de requisitos e separado por módulos. Para atender a crescente demanda de conteúdo e acesso em dispositivos móveis, foi utilizada a tecnologia de *web design* responsivo, garantido assim a compatibilidade visual em todas as plataformas.

### 4.1 MODELO DO PROCESSO DE ENGENHARIA DE SOFTWARE

Foi utilizada a metodologia ágil *scrum*, visto que o *backlog* do sistema já estava criado com base nos requisitos obtidos, sendo necessário colocá-los em módulos e criar *backlog* da *sprint*. A seguinte sequência de trabalho foi adotada:

1. Levantamento de requisitos: feito em entrevistas com Professores universitários;
2. Fundamentação teórica: realizada com base no levantamento de requisitos;
3. Diagramas UML – Documentação : realizado após o levantamento de requisitos e discussão do projeto entre o time de *scrum*;
4. Desenvolvimento: executado por toda a equipe após os três itens supracitados estarem prontos;
5. Testes do time *scrum*: realizados em paralelo ao desenvolvimento. Em cada *sprint*, toda vez que um item do *sprint backlog* foi implementado, antes de ser movido para o status “Done”, este foi testado. Ao fim do *sprint backlog*, todos os itens foram testados para verificar que além de

funcionar individualmente, estavam funcionando corretamente de forma integrada.

## 4.2 METODOLOGIA DE IMPLEMENTAÇÃO

Baseado no *backlog* do produto e na documentação realizada, foi realizado um vasto estudo sobre os padrões, técnicas e tecnologias que suprissem com qualidade as necessidades e desafios levantados.

Para melhor estruturação e organização do projeto, foi utilizado o padrão MVC ( Model, View e Controller) – explicado na seção 2.5.1 – com isso houve ganhos significativos em tempo de desenvolvimento pois a equipe era formada por quatro integrantes.

O sistema foi desenvolvido utilizando a linguagem de programação JAVA. Ela tem a vantagem de ser orientada a objetos e pode ser utilizada em aplicações de todos os tamanhos (DEITEL; DEITEL, 2011). O SGBD (Sistema Gerenciador de Banco de Dados) escolhido foi o Postgre SQL. Ele é um potente SGBD, tem uma arquitetura sólida e boa reputação quanto a integridade dos dados e pode ser executado nos principais sistemas operacionais como Linux, Unix , MacOS e Windows (BERMIER, 2009) .

Como repositório de código-fonte, foi utilizado o Git Hub Bitbucket, que é de fácil manuseio e configuração além de ser compatível com a IDE utilizada no desenvolvimento.

A fim de obtenção de uma maior segurança, foi implementado criptografia no banco de dados. O algoritmo escolhido foi o AES, pois foi projetado para permitir a expansão da chave quando necessário, pode ser implementado tanto a nível de *hardware* quanto *software* além de ser código aberto. (TREVISAN,2013)

## 4.3 REQUISITOS

Foram realizadas reuniões com professores universitários para que fosse realizado o levantamento de requisitos, a fim de estruturar a construção do sistema.

Os requisitos foram classificados como funcionais e não funcionais e apresentado em formas de quadros de acordo com o método “Volere” que apresenta boas práticas de escrita para os requisitos, aborda o problema da granularidade/atomicidade dos requisitos funcionais, fornece um modelo de artefato para proceder com a formalização, mas se limita, na instrução de preenchimento deste artefato, a indicar que a especificação do requisito corresponde a “Um enunciado através de uma sentença acerca da finalidade do requisito” (ROBERTSON; ROBERTSON, 2007) e estão detalhados nas seções a seguir.

#### 4.3.1 Requisitos funcionais

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF1

**Nome do Requisito:** Cadastro de Professores.

**Prioridade:** Alta

**Descrição do Requisito:** O sistema deve permitir o cadastro somente de professores que tiverem convite, este sendo enviado via e-mail.

**Dependências:**RF2.

**Histórico:**

**Tipo de Requisito:** Funcional (RF)

**Código do Requisito:** RF2

**Nome do Requisito:** Edição de perfil.

**Prioridade:** Alta

**Descrição do Requisito:** Um professor pode editar e desativar seu perfil quando quiser.

**Dependências:**RF2.

**Histórico:**

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF3

**Nome do Requisito:** Convite de professores.

**Prioridade:** Alta

**Descrição do Requisito:** Um professor pode enviar convites para que outros professores também tenham acesos ao sistema.

**Dependências:** Não há.

**Histórico:**

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF4

**Nome do Requisito:** Cadastro de questões.

**Prioridade:** Alta

**Descrição do Requisito:** Um professor pode cadastrar questões. Todas as questões devem ter um tipo (Objetiva e/ou V ou F).

**Dependências:** RF1.

**Histórico:**

25/08/15 – Adicionada relação com RF1

11/11/15 – Retirada o tipo de prova Somatória pois não há tempo hábil para codificação.

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF5

**Nome do Requisito:** Características de questões.

**Prioridade:** Alta

**Descrição do Requisito:** As questões podem ser privadas ou públicas. Apenas o professor autor da questão pode editá-la ou removê-la.

**Dependências:** RF1.

**Histórico:**

25/08/15 – Adicionada relação com RF1

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF6

**Nome do Requisito:** Geração de prova.

**Prioridade:** Alta

**Descrição do Requisito:** Um professor pode gerar uma prova contendo o número de questões que achar necessário, essas questões deverão estar em sua lista de questões obrigatoriamente.

**Dependências:** RF4.

**Histórico:**

25/08/15 – Inserida relação com o RF4

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF7

**Nome do Requisito:** Geração de prova selecionada ou aleatória.

**Prioridade:** Alta

**Descrição do Requisito:** A prova pode ser gerada selecionando questões ou pode ser gerada com questões aleatórias.

**Dependências:** RF4.

**Histórico:**

25/08/15 – Inserida relação com o RF4

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF8

**Nome do Requisito:** Interação social.

**Prioridade:** Alta

**Descrição do Requisito:** Um professor pode “Seguir” outro professor e assim receberá notificações quando o professor seguido adicionar novas questões. Um professor pode “Curtir” questões e também adicionar comentários às questões. Os professores podem conversar por via de mensagens pessoais.

**Dependências:** Não há.

**Histórico:**

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF9

**Nome do Requisito:** Curtir e comentar questões.

**Prioridade:** Alta

**Descrição do Requisito:** Um professor pode “curtir” as questões que achar interessante e adicionar comentários às questões.

**Dependências:** Não há.

**Histórico:**

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF10

**Nome do Requisito:** Conversa via *Chat*.

**Prioridade:** Alta

**Descrição do Requisito:** Os professores podem conversar por via de mensagens pessoais ( *Chat* ), em tempo real.

**Dependências:** Não há.

**Histórico:**

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF11

**Nome do Requisito:** Modelos de prova.

**Prioridade:** Média

**Descrição do Requisito:** Um professor pode personalizar o cabeçalho de um modelo de prova.

**Dependências:**RF5.

**Histórico:**

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF12

**Nome do Requisito:** Tipos de prova.

**Prioridade:** Média

**Descrição do Requisito:** Um professor pode optar pelos seguintes tipos de prova: Todas as questões iguais para todos, apenas ordem de questões embaralhada ou questões e alternativas embaralhadas.

**Dependências:**RF5.

**Histórico:**

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF13

**Nome do Requisito:** Relatório de questões mais difíceis.

**Prioridade:** Alta

**Descrição do Requisito:** O sistema pode gerar um relatório com as questões mais difíceis (mais erradas pelos alunos) dentre a sua lista de questões.

**Dependências:**RF10.

**Histórico:**

**Tipo de Requisito:** Funcional (RF)

**Código do Requisito:** RF14

**Nome do Requisito:** Histórico de provas

**Prioridade:** Alta

**Descrição do Requisito:** O sistema deve armazenar e mostrar todas as provas criadas pelo professor.

**Dependências:**RF5.

**Histórico:**

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF15

**Nome do Requisito:** Denúncia de questões



**Prioridade:** Média

**Descrição do Requisito:** Ao visualizar uma questão e encontrar um erro, um professor pode acionar o botão “Denunciar Questão”. Em seguida, o sistema abre uma janela para que o professor entre com um texto explicando qual o erro.

**Dependências:**RF4.

**Histórico:**

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF16

**Nome do Requisito:** Avaliação de questões.

**Prioridade:** Baixa

**Descrição do Requisito:** Um professor pode avaliar o nível de qualidade de questões de outros professores dando notas de 0 a 5 “Estrelas”.

**Dependências:**RF4.

**Histórico:**

25/08/15 – Alteração da prioridade de Alta para Baixa e inserida relação com RF4

11/11/15 - Requisito não pode ser implementado, falta de tempo hábil para a codificação.

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF17

**Nome do Requisito:** Modelos de prova.

**Prioridade:** Baixa

**Descrição do Requisito:** Um professor pode escolher um modelo de prova por instituição dentre os modelos pré-definidos ou então cadastrar um modelo de prova para uso próprio.

**Dependências:**RF5.

**Histórico:**

11/11/15 - Requisito não pode ser implementado, falta de tempo hábil para a codificação.

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF18

**Nome do Requisito:** Peso nas questões por prova.

**Prioridade:** Baixa

**Descrição do Requisito:** Ao gerar uma prova, um professor pode definir um peso de 0 a 100 para cada questão, assim o aluno saberá quanto cada uma vale.

**Dependências:**RF5.

**Histórico:**

25/08/15 – Alteração da prioridade de Alta para média.

30/09/2015 – Alteração no valor máximo do peso da prova.

11/11/15 - Requisito não pode ser implementado, falta de tempo hábil para a codificação.

**Tipo de Requisito:** Funcional(RF)

**Código do Requisito:** RF19

**Nome do Requisito:** Geração de prova baseada em desempenho anterior dos alunos.

**Prioridade:** Baixa

**Descrição do Requisito:** Um professor pode optar por gerar uma prova baseado em uma ou mais provas anteriores, assim o sistema irá gerar a prova com questões relacionadas ao tema que a turma teve mais dificuldade na(s) prova(s) anterior(es).

**Dependências:** RF5, RF10.

**Histórico:**

11/11/15 - Requisito não pode ser implementado, falta de tempo hábil para a codificação.

#### 4.3.2 Requisitos não funcionais

**Tipo de Requisito:** Não Funcional (RNF)

**Código do Requisito:** RNF01

**Nome do Requisito:** Atualização de tela por AJAX na área de mensagens.

**Prioridade:** Média

**Descrição do Requisito:** A tela das mensagens deve ser atualizada a cada 10 segundos via AJAX.

**Dependências:** RF01.

**Histórico:**

**Tipo de Requisito:** Não Funcional (RNF)

**Código do Requisito:** RNF02

**Nome do Requisito:** Servidor disponível durante 99,99% do tempo.

**Prioridade:** Média

**Descrição do Requisito:** O servidor deve ficar disponível durante 99,99% do tempo durante a execução da aplicação. A garantia deste requisito encontra-se no Apêndice I – Acordo de nível de Serviço.

**Dependências:**

**Histórico:** 12/11/2015 – Adicionado referência ao Apêndice I.

**Tipo de Requisito:**Não Funcional (RNF)

**Código do Requisito:** RNF03

**Nome do Requisito:**Utilizar padrão MVC.

**Prioridade:** Alta

**Descrição do Requisito:** A aplicação deve ser desenvolvida no padrão MVC (*Model, View, Controller*).

**Dependências:**

**Histórico:**

**Tipo de Requisito:**Não Funcional (RNF)

**Código do Requisito:** RNF04

**Nome do Requisito:**Senha criptografada.

**Prioridade:** Média

**Descrição do Requisito:**A senha deve ser criptografada, em AES.

**Dependências:**

**Histórico:**

15/11/15 – Criptografia alterada de md5 para AES.

**Tipo de Requisito:**Não Funcional (RNF)

**Código do Requisito:** RNF05

**Nome do Requisito:**Validação dos campos.

**Prioridade:** Alta

**Descrição do Requisito:** Os campos dos formulários devem ser validados tanto no lado do cliente quanto no lado do servidor.

**Dependências:**

**Histórico:**

11/11/15 – Os campos dos formulários devem ser validados no lado do servidor, não mais no lado cliente.

**Tipo de Requisito:**Não Funcional (RNF)

**Código do Requisito:** RNF06

**Nome do Requisito:**Redirecionamento de portas para acesso ao servidor.

**Prioridade:** Média

**Descrição do Requisito:** O servidor de aplicação deve estar ouvindo na porta 8080 e para ser acessado, deve ser redirecionado por outro servidor ouvindo na porta 80.

**Dependências:**

**Histórico:**

**Tipo de Requisito:** Não Funcional (RNF)

**Código do Requisito:** RNF07

**Nome do Requisito:** Desenvolvimento do sistema com Java e framework JSF.

**Prioridade:** Média

**Descrição do Requisito:** Deve-se utilizar a linguagem Java para o desenvolvimento do sistema e framework JSF para confeccionar as telas.

**Dependências:**

**Histórico:**

#### 4.4 PLANO DE ATIVIDADE

As atividades realizadas durante o desenvolvimento do projeto foram planejadas e baseadas nos seguintes documentos:

- Tabela de Precedência;
- Cronograma.

##### 4.4.1 Tabela de precedência

Visando o sucesso do projeto, foi definida uma ordem para a realização das principais atividades. A definição dessas atividades foi baseada nos levantamentos de requisitos junto ao cliente adicionada da experiência técnica da equipe de desenvolvedores. Essa ordem está representada no QUADRO 1.

Nº Identificador	Atividade	Precedência
1	Acesso ao sistema	
2	Convite de professores	1
3	Manter professores	2
4	Conexão com Facebook	2
5	Manter questões	3
6	Geração de prova	3
7	Interação social	5
8	Avaliação de questões	5
9	Modelos de prova	3
10	Tipos de prova	3
11	Cadastro de desempenho	6
12	Peso nas questões por prova	6
13	Geração de prova baseada em desempenho anterior dos alunos	6,11
14	Relatório de questões mais difíceis	6

QUADRO 2 - TABELA DE PRECEDÊNCIA  
 FONTE: OS AUTORES, 2015

#### 4.4.2 Cronograma

Para o desenvolvimento do projeto foi utilizada a metodologia ágil *scrum* que é baseada em *sprints*. O espaço de tempo das entregas foi definido para ser semanal, sempre começando e terminando em quarta-feira. O cronograma do projeto foi elaborado visando demonstrar o que teria que ser realizado em cada uma das 13 *sprints* planejadas e 15 realizadas. Para demonstrar o tempo de execuções das atividades, foram montados dois tipos de cronogramas: o cronograma planejado, que contém as atividades e datas planejadas antes de começar o desenvolvimento do produto e o cronograma realizado, que mostra o que realmente foi feito em cada data, permitindo uma comparação.

A utilização de cronograma tem como objetivo o planejamento de todas as atividades necessárias para a confecção do *software* e da sua documentação, assim como descrever a ordem cronológica em que serão realizadas.

Para uma melhor organização das tarefas, foi definido como necessário a utilização das precedências antes de colocar em prática a distribuição em *Kanbans*

entre a equipe. Isso ocorreu por entendermos que seria mais correto e otimizado se cada um soubesse o que deveria ser feito e em qual ordem, antes das atribuições de tarefas. Essa foi uma decisão que coube exclusivamente para a realização desse projeto, assim não alterando as metodologias escolhidas mas sim buscando o aperfeiçoamento das mesmas.

#### 4.4.2.1 Cronograma planejado

Nessa seção estão demonstrados os cronogramas planejados antes do início do desenvolvimento do *software*. O QUADRO 2 representa a visão geral do cronograma, e os QUADROS 3 a 35 apresentam uma visão mais detalhada.

ID	Nome da Tarefa	Trabalho (horas)	Início	Termino
1	Sistema NetQ	1132	Segunda 09/08/2015	Terça 01/12/2015
2	Gerenciamento do Projeto	110	Quarta 12/08/2015	Quarta 26/08/2015
7	Documentacao	200	Quarta 19/08/2015	Quarta 16/09/2015
17	SPRINT 1	30	Quarta 12/08/2015	Quarta 19/08/2015
	SPRINT 2	52	Quarta 19/08/2015	Quarta 26/08/2015
	SPRINT 3	60	Quarta 26/08/2015	Quarta 02/09/2015
	SPRINT 4	60	Quarta 02/09/2015	Quarta 09/09/2015
	SPRINT 5	60	Quarta 09/09/2015	Quarta 16/09/2015
	SPRINT 6	60	Quarta 16/09/2015	Quarta 23/09/2015
	SPRINT 7	90	Quarta 23/09/2015	Quarta 30/09/2015
	SPRINT 8	90	Quarta 30/09/2015	Quarta 07/10/2015
	SPRINT 9	90	Quarta 07/10/2015	Quarta 14/10/2015
	SPRINT 10	90	Quarta 14/10/2015	Quarta 21/10/2015
	SPRINT 11	90	Quarta 21/10/2015	Quarta 28/10/2015
	SPRINT 12	90	Quarta 28/10/2015	Quarta 04/11/2015



	SPRINT 13	90	Quarta 04/11/2015	Quarta 11/11/2015
--	-----------	----	-------------------	-------------------

QUADRO 3 - VISÃO GERAL DO CRONOGRAMA DO PROJETO  
 FONTE: OS AUTORES, 2015

ID	Nome da tarefa	Trabalho (horas)	Início	Termino	Nomes dos recursos
4	Tabela de Precedencia	16	Quarta 19/08/2015	Quarta 26/08/2015	Yuri
5	Plano de Risco	51	Quarta 19/08/2015	Quarta 26/08/2015	Bruno
6	Plano de Comunicação	51	Quarta 19/08/2015	Quarta 26/08/2015	Felipe

QUADRO 4 - CRONOGRAMA DE GERENCIAMENTO DO PROJETO  
 FONTE: OS AUTORES, 2015

ID	Nome da Tarefa	Trabalho (horas)	Início	Termino	Nome dos recursos	Prede-cessoras
8	Requisitos Funcionais	30	Quarta 12/08/2015	Quarta 19/08/2015	Felipe	
9	Aperfeiçoamento dos Requisitos Funcionais	21	Quarta 19/08/2015	Quarta 26/08/2015	Yuri	8
10	Requisitos não funcionais	8	Quarta 19/08/2015	Quarta 26/08/2015	Evandro	
11	Diagrama de Casos de Uso	8	Quarta 26/08/2015	Quarta 02/09/2015	Bruno	9
12	Especificação de Caso de Uso	20	Quarta 26/08/2015	Quarta 02/09/2015	Felipe	11
13	Diagrama de Classe	20	Quarta 26/08/2015	Quarta 02/09/2015	Bruno	9
14	Diagrama de Entidade e Relacionamento	20	Quarta 26/08/2015	Quarta 02/09/2015	Yuri	9
15	Diagrama de Seqüência	20	Quarta 02/09/2015	Quarta 09/09/2015	Felipe	12

QUADRO 5 - CRONOGRAMA DE DOCUMENTAÇÃO

FONTE: OS AUTORES, 2015

<b>Id</b>	<b>Nome da Tarefa</b>	<b>Trabalho (horas)</b>	<b>Inicio</b>	<b>Termino</b>	<b>Nomes dos Recursos</b>	<b>Predecessoras</b>
18	Cronograma	30	Quarta 12/08/2015	Quarta 26/08/2015	Bruno	
19	Prototipação das Telas	15	Quarta 12/08/2015	Quarta 19/08/2015	yuri	9
20	Trello - Atualizar	1	Quarta 12/08/2015	Quarta 19/08/2015	Bruno	18
21	Análise de Requisitos	15	Quarta 12/08/2015	Quarta 19/08/2015	Felipe	
22	Infraestrutura (servidor)	40	Quarta 12/08/2015	Quarta 19/08/2015	Evandro	
23	Infraestrutura IDE, BD	3	Quarta 12/08/2015	Quarta 19/08/2015	Todos	

QUADRO 6 - CRONOGRAMA DA SPRINT 1

FONTE: OS AUTORES, 2015

<b>Id</b>	<b>Nome da Tarefa</b>	<b>Trabalho (horas)</b>	<b>Inicio</b>	<b>Termino</b>	<b>Nomes dos Recursos</b>	<b>Predecessoras</b>
10	Requisitos não funcionais	8	Quarta 19/08/2015	Quarta 26/08/2015	Bruno	9
24	Plano de Risco	20	Quarta 19/08/2015	Quarta 26/08/2015	Felipe	11
25	Protótipo no Servidor	20	Quarta 19/08/2015	Quarta 26/08/2015	Evandro	9
26	Tabela de precedência	20	Quarta 19/08/2015	Quarta 26/08/2015	Yuri	9
27	Plano de comunicação	1	Quarta 19/08/2015	Quarta 26/08/2015	Bruno	
19	Aperfeiçoamento dos requisitos funcionais	15	Quarta 19/08/2015	Quarta 26/08/2015	Yuri	9

QUADRO 7 - CRONOGRAMA DA SPRINT 2

FONTE: OS AUTORES, 2015

ID	Nome da Tarefa	Trabalho (horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
11	Finalizar prototipação das telas	8	Quarta 26/08/2015	Quarta 02/09/2015	Bruno	9
12	Diagramas de Casos de uso	20	Quarta 26/08/2015	Quarta 09/09/2015	Felipe	11
13	Especificações de caso de uso	20	Quarta 26/08/2015	Quarta 02/09/2015	Evandro	9
14	Diagrama de Classe	20	Quarta 26/08/2015	Quarta 02/09/2015	Yuri	9
22	Trello - Atualizar	1	Quarta 26/08/2015	Quarta 02/09/2015	Bruno	

QUADRO 8 - CRONOGRAMA DA *SPRINT* 3  
 FONTE: OS AUTORES, 2015

ID	Nome da Tarefa	Trabalho (horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
12	Finalizar Especificação de Caso de Uso	20	Quarta 02/09/2015	Quarta 09/09/2015	Felipe	11
14	Iniciar diagrama de sequencia	20	Quarta 02/09/2015	Quarta 09/09/2015	Bruno	9
22	Trello - Atualizar	1	Quarta 02/09/2015	Quarta 09/09/2015	Bruno	
24	Definir arquitetura	15	Quarta 02/09/2015	Quarta 09/09/2015	yuri	9
25	Prototipar primeira tela funcional	15	Quarta 02/09/2015	Quarta 09/09/2015	yuri	9
26	Configurar Servidor	15	Quarta 02/09/2015	Quarta 09/09/2015	Evandro	9

QUADRO 9 - CRONOGRAMA DA *SPRINT* 4  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
27	Prototipar todas as telas do sistema	20	Quarta 16/09/2015	Quarta 23/09/2015	Felipe	9
28	Consolidar diagramas e Especificação na documentação	20	Quarta 16/09/2015	Quarta 23/09/2015	Yuri	11,12,13, 14

22	Trello - Atualizar	1	Quarta 16/09/2015	Quarta 23/09/2015	Bruno	
29	Programar todas as classes e atributos	15	Quarta 16/09/2015	Quarta 23/09/2015	yuri	12,14

QUADRO 10 - CRONOGRAMA DA SPRINT 5  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Início	Termino	Nomes dos Recursos	Predecessoras
30	Criação do Script do BD	20	Quarta 23/09/2015	Quarta 30/09/2015	Felipe	32
31	Iniciar programação dos DAOS	20	Quarta 23/09/2015	Quarta 30/09/2015	Yuri	29
22	Trello - Atualizar	1	Quarta 23/09/2015	Quarta 30/09/2015	Bruno	
32	Criação diagrama lógico do BD	15	Quarta 23/09/2015	Quarta 30/09/2015	yuri	13

QUADRO 11 - CRONOGRAMA DA SPRINT 6  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Início	Termino	Nomes dos Recursos	Predecessoras
33	CRUD de instituição	20	Quarta 30/09/2015	Quarta 07/10/2015	Felipe	29,31
34	CRUD de questões	20	Quarta 30/09/2015	Quarta 07/10/2015	Yuri	29,31
22	Trello - Atualizar	1	Quarta 30/09/2015	Quarta 07/10/2015	Bruno	
35	CRUD de professores	15	Quarta 30/09/2015	Quarta 07/10/2015	yuri	29,31

QUADRO 12 - CRONOGRAMA DA SPRINT 7  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Início	Termino	Nomes dos Recursos	Predecessoras
36	Seguir Professor	20	Quarta 07/10/2015	Quarta 14/10/2015	Felipe	35
37	Login, Sessão e Área completa do professor	20	Quarta 07/10/2015	Quarta 14/10/2015	Yuri	35
22	Trello - Atualizar	1	Quarta 07/10/2015	Quarta 14/10/2015	Bruno	
38	Convite de Professor	15	Quarta 07/10/2015	Quarta 14/10/2015	yuri	35
39	Tela e codificar - Mensagem	15	Quarta 07/10/2015	Quarta 14/10/2015	yuri	27,35

QUADRO 13 - CRONOGRAMA DA SPRINT 8

40	Módulo de mensagens entre professores	15	Quarta 07/10/2015	Quarta 14/10/2015	yuri	35
----	---------------------------------------	----	----------------------	----------------------	------	----

FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecesoras
41	Criação de prova - Modelo de prova	20	Quarta 14/10/2015	Quarta 21/10/2015	Felipe	34
42	Comentário nas questões	20	Quarta 14/10/2015	Quarta 21/10/2015	Yuri	34,35
22	Trello - Atualizar	1	Quarta 14/10/2015	Quarta 21/10/2015	Bruno	
43	Like das questões	15	Quarta 14/10/2015	Quarta 21/10/2015	yuri	34,35

QUADRO 14 - CRONOGRAMA DA SPRINT 9

FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecesoras
44	Historico de Provas	20	Quarta 21/10/2015	Quarta 28/10/2015	Felipe	41
45	Criação dos Relatorios	20	Quarta 21/10/2015	Quarta 28/10/2015	Yuri	Todos
22	Trello - Atualizar	1	Quarta 21/10/2015	Quarta 28/10/2015	Bruno	
46	Peso nas questões por prova	15	Quarta 21/10/2015	Quarta 28/10/2015	yuri	-
47	Denuncia de questões	15	Quarta 21/10/2015	Quarta 28/10/2015	yuri	34,35
48	Página de exibição das questões	15	Quarta 21/10/2015	Quarta 28/10/2015	yuri	34

QUADRO 15 - CRONOGRAMA DA SPRINT 10

FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecesoras
49	Cadastro de desempenho	20	Quarta 28/10/2015	Quarta 04/11/2015	Felipe	34,35,41
50	Avaliação das questões	20	Quarta 28/10/2015	Quarta 04/11/2015	Yuri	34
22	Trello - Atualizar	1	Quarta 28/10/2015	Quarta 04/11/2015	Bruno	
51	Tipo de prova	15	Quarta 28/10/2015	Quarta 04/11/2015	yuri	34,35,41

52	Conexão com o facebook	15	Quarta 28/10/2015	Quarta 04/11/2015	yuri	-
53	Modelo de prova	15	Quarta 28/10/2015	Quarta 04/11/2015	yuri	41

QUADRO 16 - CRONOGRAMA DA SPRINT 11  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
54	Ajustar documentação	20	Quarta 04/11/2015	Quarta 11/11/2015	Felipe	28
55	Iniciar Power Point da apresentação	20	Quarta 04/11/2015	Quarta 11/11/2015	Yuri	28
22	Trello - Atualizar	1	Quarta 04/11/2015	Quarta 11/11/2015	Bruno	
56	Geração de prova baseada em desempenho dos alunos	15	Quarta 04/11/2015	Quarta 11/11/2015	yuri	41

QUADRO 17 - CRONOGRAMA DA SPRINT 12  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
57	Possível backlog	20	Quarta 02/09/2015	Quarta 09/09/2015	Felipe	Todos
58	Terminar documentação	20	Quarta 02/09/2015	Quarta 09/09/2015	Yuri	54
22	Trello - Atualizar	1	Quarta 02/09/2015	Quarta 09/09/2015	Bruno	
59	Finalizar apresentação	15	Quarta 02/09/2015	Quarta 09/09/2015	yuri	55
60	Praticar apresentação	15	Quarta 02/09/2015	Quarta 09/09/2015	yuri	59
61	Testes	15	Quarta 02/09/2015	Quarta 09/09/2015	yuri	

QUADRO 18 - CRONOGRAMA DA SPRINT 13  
 FONTE: OS AUTORES, 2015

#### 4.4.2.2 Cronograma realizado

Nessa seção serão demonstrados os mesmos quadros apresentados na seção 4.4.2.1, com os dados informando as datas que as atividades foram realmente realizadas.

ID	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino
1	Sistema NetQ	1132	Segunda 09/08/2015	Terca 01/12/2015
2	Gerenciamento do Projeto	110	Quarta 12/08/2015	Quarta 26/08/2015
7	Documentação	200	Quarta 19/08/2015	Quarta 16/09/2015
17	<i>SPRINT 1</i>	30	Quarta 12/08/2015	Quarta 19/08/2015
	<i>SPRINT 2</i>	52	Quarta 19/08/2015	Quarta 26/08/2015
	<i>SPRINT 3</i>	60	Quarta 26/08/2015	Quarta 02/09/2015
	<i>SPRINT 4</i>	60	Quarta 02/09/2015	Quarta 09/09/2015
	<i>SPRINT 5</i>	60	Quarta 09/09/2015	Quarta 16/09/2015
	<i>SPRINT 6</i>	60	Quarta 16/09/2015	Quarta 23/09/2015
	<i>SPRINT 7</i>	60	Quarta 23/09/2015	Quarta 30/09/2015
	<i>SPRINT 8</i>	90	Quarta 30/09/2015	Quarta 07/10/2015
	<i>SPRINT 9</i>	90	Quarta 07/10/2015	Quarta 14/10/2015
	<i>SPRINT 10</i>	90	Quarta 14/10/2015	Quarta 21/10/2015
	<i>SPRINT 11</i>	90	Quarta 21/10/2015	Quarta 28/10/2015
	<i>SPRINT 12</i>	90	Quarta 28/10/2015	Quarta 04/11/2015
	<i>SPRINT 13</i>	90	Quarta 04/11/2015	Quarta 11/11/2015
	<i>SPRINT 14</i>	90	Quarta 11/11/2015	Quarta 18/11/2015
	<i>SPRINT 15</i>	90	Quarta 18/11/2015	Quarta 25/11/2015

QUADRO 19 - VISÃO GERAL DO CRONOGRAMA REALIZADO  
 FONTE: OS AUTORES, 2015



<b>Id</b>	<b>Nome da tarefa</b>	<b>Trabalho (Horas)</b>	<b>Inicio</b>	<b>Termino</b>	<b>Nomes dos recursos</b>
4	Tabela de Precedencia	16	Quarta 19/08/2015	Quarta 26/08/2015	Yuri
5	Plano de Risco	51	Quarta 19/08/2015	Quarta 26/08/2015	Bruno
6	Plano de Comunicação	51	Quarta 19/08/2015	Quarta 26/08/2015	Felipe

QUADRO 20 - CRONOGRAMA REALIZADO DE GERENCIAMENTO DO PROJETO  
 FONTE: OS AUTORES, 2015

<b>Id</b>	<b>Nome da Tarefa</b>	<b>Trabalho (Horas)</b>	<b>Inicio</b>	<b>Termino</b>	<b>Nomes dos Recursos</b>	<b>Predecesoras</b>
8	Requisitos Funcionais	30	Quarta 12/08/2015	10/10/2015	Felipe	
9	Aperfeiçoamento dos Requisitos Funcionais	21	Quarta 19/08/2015	10/10/2015	Yuri	8
10	Requisitos nao funcionais	8	Quarta 19/08/2015	26/08/2015	Evandro	
11	Diagrama de Casos de Uso	8	Quarta 26/08/2015	02/09/2015	Bruno	9
12	Especificação de Caso de Uso	20	Quarta 26/08/2015	20/11/2015	Felipe	11
13	Diagrama de Classe	20	Quarta 26/08/2015	02/09/2015	Bruno	9
14	Diagrama de Entidade e Relacionamento	20	Quarta 26/08/2015	02/09/2015	Yuri	9
15	Diagrama de Sequencia	20	Quarta 02/09/2015	18/11/2015	Felipe	12

QUADRO 21 - CRONOGRAMA REALIZADO DE DOCUMENTAÇÃO  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
18	Cronograma	30	Quarta 12/08/2015	26/08/2015	Bruno	
19	Prototipação das Telas	15	Quarta 12/08/2015	16/09/2015	yuri	9
20	Trello - Atualizar	1	Quarta 12/08/2015	19/08/2015	Bruno	18
21	Análise de Requisitos	15	Quarta 12/08/2015	10/10/2015	Felipe	
22	Infraestrutura (servidor)	40	Quarta 12/08/2015	19/08/2015	Evandro	
23	Infraestrutura IDE, BD	3	Quarta 12/08/2015	22/08/2015	Todos	

QUADRO 22 - CRONOGRAMA REALIZADO DA SPRINT 1

FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
11	Definição de requisitos não funcionais	8	Quarta 26/08/2015	02/09/2015	Bruno	9
12	Plano de Risco	20	Quarta 26/08/2015	02/09/2015	Felipe	11
13	Protótipo no Servidor	20	Quarta 26/08/2015	02/09/2015	Evandro	9
14	Tabela de precedência	20	Quarta 26/08/2015	02/09/2015	Yuri	9
22	Trello - Atualizar	1	Quarta 26/08/2015	02/09/2015	Bruno	
19	Plano de comunicação	15	Quarta 12/08/2015	02/09/2015	yuri	9

QUADRO 23 - CRONOGRAMA REALIZADO DA SPRINT 2

FONTE: OS AUTORES, 2015

<b>Id</b>	<b>Nome da Tarefa</b>	<b>Trabalho (Horas)</b>	<b>Inicio</b>	<b>Termino</b>	<b>Nomes dos Recursos</b>	<b>Predecessoras</b>
11	Diagrama de Casos de Uso	8	Quarta 26/08/2015	02/09/2015	Bruno	9
12	Especificação de Caso de Uso	20	Quarta 26/08/2015	20/11/2015	Felipe	11
13	Diagrama de Classe	20	Quarta 26/08/2015	02/09/2015	Evandro	9
14	Diagrama de Entidade e Relacionamento	20	Quarta 26/08/2015	02/09/2015	Yuri	9
22	Trello - Atualizar	1	Quarta 26/08/2015	02/09/2015	Bruno	

QUADRO 24 - CRONOGRAMA REALIZADO DA *SPRINT 3*  
 FONTE: OS AUTORES, 2015

<b>Id</b>	<b>Nome da Tarefa</b>	<b>Trabalho (Horas)</b>	<b>Inicio</b>	<b>Termino</b>	<b>Nomes dos Recursos</b>	<b>Predecessoras</b>
12	Finalizar Especificação de Caso de Uso	20	Quarta 02/09/2015	09/09/2015	Felipe	11
14	Iniciar diagrama de sequencia	20	Quarta 02/09/2015	20/10/2015	Yuri	9
22	Trello - Atualizar	1	Quarta 02/09/2015	09/09/2015	Bruno	
24	Definir arquitetura	15	Quarta 02/09/2015	09/09/2015	yuri	9
25	Prototipar primeira tela funcional	15	Quarta 02/09/2015	09/09/2015	yuri	9
26	Configurar Servidor	15	Quarta 02/09/2015	09/09/2015	Evandro	

QUADRO 25 - CRONOGRAMA REALIZADO DA *SPRINT 4*  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
27	Prototipar todas as telas do sistema	20	Quarta 16/09/2015	23/09/2015	Felipe	9
28	Consolidar diagramas e Especificação na documentação	20	Quarta 16/09/2015	23/09/2015	Yuri	11,12,13,14
22	Trello - Atualizar	1	Quarta 16/09/2015	23/09/2015	Bruno	
29	Programar todas as classes e atributos	15	Quarta 16/09/2015	23/09/2015	yuri	12,14

QUADRO 26 - CRONOGRAMA REALIZADO DA SPRINT 5

FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
30	Criação do Script do BD	20	Quarta 23/09/2015	30/09/2015	Felipe	32
31	Iniciar programação dos Daos	20	Quarta 23/09/2015	30/09/2015	Bruno	29
22	Trello - Atualizar	1	Quarta 23/09/2015	30/09/2015	Evandro	
32	Criação diagrama lógico do BD	15	Quarta 23/09/2015	30/09/2015	yuri	13
	Possível Backlog - Correção de Bugs				Bruno/Felipe/Evandro/Yuri	

QUADRO 27 - CRONOGRAMA REALIZADO DA SPRINT 6

FONTE: OS AUTORES, 2015

<b>Id</b>	<b>Nome da Tarefa</b>	<b>Trabalho (Horas)</b>	<b>Inicio</b>	<b>Termino</b>	<b>Nomes dos Recursos</b>	<b>Predecessoras</b>
33	CRUD de instituição	20	Quarta 30/09/2015	07/10/2015	Felipe	29,31
34	CRUD de questões	20	Quarta 30/09/2015	07/10/2015	Bruno/Evandro	29,31
22	Trello - Atualizar	1	Quarta 30/09/2015	07/10/2015	Evandro	
35	CRUD de professores	15	Quarta 30/09/2015	07/10/2015	yuri	29,31
	Possível Backlog - Correção de Bugs				Bruno/Felipe/Evandro/Yuri	

QUADRO 28 - CRONOGRAMA REALIZADO DA *SPRINT 7*

FONTE: OS AUTORES, 2015

<b>Id</b>	<b>Nome da Tarefa</b>	<b>Trabalho (Horas)</b>	<b>Inicio</b>	<b>Termino</b>	<b>Nomes dos Recursos</b>	<b>Predecessoras</b>
36	Seguir Professor	20	Quarta 07/10/2015	14/10/2015	Felipe	35
37	Login, Sessão e Área completa do professor	20	Quarta 07/10/2015	14/10/2015	Yuri	35
22	Trello - Atualizar	1	Quarta 07/10/2015	14/10/2015	Bruno	
38	Convite de Professor	15	Quarta 07/10/2015	14/10/2015	yuri	35
39	Tela e codificar - Mensagem	15	Quarta 07/10/2015	14/10/2015	Bruno/Evandro	27,35
40	Módulo de mensagens entre professores	15	Quarta 07/10/2015	14/10/2015	Bruno/Evandro	35
	Possível Backlog - Correção de Bugs				Bruno/Felipe/Evandro/Yuri	

QUADRO 29 - CRONOGRAMA REALIZADO DA *SPRINT 8*

FONTE: OS AUTORES, 2015

<b>Id</b>	<b>Nome da Tarefa</b>	<b>Trabalho (Horas)</b>	<b>Inicio</b>	<b>Termino</b>	<b>Nomes dos Recursos</b>	<b>Predecessoras</b>
41	Criação de prova	20	Quarta 14/10/2015	30/11/2015	Bruno	34
42	Comentário nas questões	20	Quarta 14/10/2015	10/11/2015	Yuri	34,35
22	Trello - Atualizar	1	Quarta 14/10/2015	21/10/2015	Felipe	
43	Like das questões	15	Quarta 14/10/2015	21/10/2015	Evandro	34,35
	Possível Backlog - Correção de Bugs				Bruno/Felipe/Evandro/Yuri	

QUADRO 30 - CRONOGRAMA REALIZADO DA *SPRINT 9*  
 FONTE: OS AUTORES, 2015

<b>Id</b>	<b>Nome da Tarefa</b>	<b>Trabalho (Horas)</b>	<b>Inicio</b>	<b>Termino</b>	<b>Nomes dos Recursos</b>	<b>Predecessoras</b>
44	Historico de Provas	20	Quarta 21/10/2015	30/11/2015	Yuri	41
45	Criação dos Relatorios	20	Quarta 21/10/2015	20/11/2015	Yuri	Todos
22	Trello - Atualizar	1	Quarta 21/10/2015	28/10/2015	Evandro	
46	Peso nas questões por prova	15	Quarta 21/10/2015	Não Realizado	Evandro	-
47	Denuncia de questões	15	Quarta 21/10/2015	28/10/2015	Bruno	34,35
48	Página de exibição das questões	15	Quarta 21/10/2015	28/10/2015	Felipe	34
	Possível Backlog - Correção de Bugs				Bruno/Felipe/Evandro/Yuri	

QUADRO 31 - CRONOGRAMA REALIZADO DA *SPRINT 10*  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Início	Termino	Nomes dos Recursos	Predecessoras
49	Cadastro de desempenho	20	Quarta 28/10/2015	24/11/2015	Yuri	34,35,41
50	Avaliação das questões	20	Quarta 28/10/2015	04/11/2015	Bruno	34
22	Trello - Atualizar	1	Quarta 28/10/2015	04/11/2015	Felipe	
51	Tipo de prova	15	Quarta 28/10/2015	24/11/2015	Evandro	34,35,41
52	Conexão com o facebook	15	Quarta 28/10/2015	Funcionalidad e Retirada	Evandro	-
53	Modelo de prova	15	Quarta 28/10/2015	04/11/2015	Yuri	41
	Possível Backlog - Correção de Bugs				Bruno/Felipe/Evandro/Yuri	

QUADRO 32 - CRONOGRAMA REALIZADO DA SPRINT 11  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Início	Termino	Nomes dos Recursos	Predecessoras
54	Ajustar documentação	20	Quarta 04/11/2015	24/11/2015	Bruno/Felipe/Evandro/Yuri	28
55	Iniciar Power Point da apresentação	20	Quarta 04/11/2015	26/11/2015	Yuri/Bruno	28
22	Trello - Atualizar	1	Quarta 04/11/2015	11/11/2015	Felipe	
56	Geração de prova baseada em desempenho dos alunos	15	Quarta 04/11/2015	Funcionalidad e Retirada	Yuri	-
	Possível Backlog - Correção de Bugs				Bruno/Felipe/Evandro/Yuri	

QUADRO 33 - CRONOGRAMA REALIZADO DA SPRINT 12  
 FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
57	Testes	20	Quarta 11/11/2015	29/11/2015	Bruno/Felipe/Evandro/Yuri	Todos
58	Terminar documentação	20	Quarta 11/11/2015	24/11/2015	Bruno/Felipe/Evandro/Yuri	54
22	Trello - Atualizar	1	Quarta 11/11/2015	18/11/2015	Evandro	
59	Finalizar apresentação	15	Quarta 11/11/2015	29/11/2015	Bruno	55
60	Praticar apresentação	15	Quarta 11/11/2015	29/11/2015	Bruno/Felipe/Evandro/Yuri	59
	Possível Backlog - Correção de Bugs				Bruno/Felipe/Evandro/Yuri	

QUADRO 34 - CRONOGRAMA REALIZADO DA SPRINT 13

FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
	Possível Backlog - Correção de Bugs	80	18/11/2015	25/11/2015	Bruno/Felipe/Evandro/Yuri	Todos

QUADRO 35 - CRONOGRAMA REALIZADO DA SPRINT 14

FONTE: OS AUTORES, 2015

Id	Nome da Tarefa	Trabalho (Horas)	Inicio	Termino	Nomes dos Recursos	Predecessoras
	Possível Backlog - Correção de Bugs	80	25/11/2015	01/12/2015	Bruno/Felipe/Evandro/Yuri	Todos

QUADRO 36 - CRONOGRAMA REALIZADO DA SPRINT 15

FONTE: OS AUTORES, 2015



#### 4.5 PLANO DE RISCO

No plano de risco situado no apêndice D, foram descritas as condições de risco que poderiam ter efeito no desenvolvimento do trabalho, suas consequências e o plano de ação que deveria ser empregado caso houvesse necessidade. Também foi descrito quais eram os responsáveis pelo monitoramento de cada risco, sua probabilidade de ocorrência e classificação (BOEHM,1991).

#### 4.6 PLANO DE COMUNICAÇÃO

O plano de comunicação, situado no apêndice G foi criado com base nas necessidades de compartilhamento de comunicações e informações. Ele responde às seguintes questões:

- Como as informações serão fornecidas;
- Por quem serão fornecidas;
- Quem precisa das informações;
- Quando as informações são necessárias.

O atraso na entrega das mensagens, problemas de comunicação de informações confidenciais para o público incorreto ou a falta de comunicação entre as partes interessadas é resultado de um planejamento incorreto de um plano de comunicação (PMI,2008)

#### 4.7 MATERIAIS

Foram utilizados recursos de *software* e de *hardware* para a confecção do projeto. Alguns desses *softwares* são gratuitos, outros possuem licença de uso paga porém com uso grátis permitido para 30 dias. Nessa seção estão explicados cada um desses recursos.

#### 4.7.1 Recursos de *hardware*

Os recursos de *hardware* utilizados pelos integrantes da equipe foram dois computadores portáteis, dois computadores de mesa, sendo todos pessoais e uma máquina virtual alugada pela equipe.

##### 1. COMPUTADOR 01

- a. Nome da Máquina: YURI-PC
- b. Proprietário: Yuri Jungles
- c. Sistema Operacional: Windows 10
- d. Memória RAM: 8 GB
- e. Processador: Intel Core i5 3.2GHz
- f. Espaço de armazenamento: 500GB
- g. Placa de vídeo: Nvidia GTX760

##### 2. COMPUTADOR 02

- a. Nome da Máquina: Evandro-BC
- b. Proprietário: Evandro Machado
- c. Sistema Operacional: Windows 7
- d. Memória RAM: 16 GB DD3
- e. Processador: i5 -3.4GHZ
- f. Espaço de armazenamento: 1TB
- g. Placa de vídeo: Nvidia GeForce GTX 650TI

##### 3. COMPUTADOR 03

- a. Nome da Máquina: Felipe
- b. Proprietário: Felipe Linhares
- c. Sistema Operacional: Windows 10
- d. Memória RAM: 8 GB
- e. Processador: i7 2.4 GHz
- f. Espaço de armazenamento: 1 TB
- g. Placa de vídeo: Nvidia GeForce GTX756m

#### 4. COMPUTADOR 04

- a. Nome da Máquina: Bruno-PC
- b. Proprietário: Bruno Ricardo Sella
- c. Sistema Operacional: MAC OS EL CAPITAN
- d. Memória RAM: 4 GB
- e. Processador: i5 2.3 GHz
- f. Espaço de armazenamento: 500 GB
- g. Placa de vídeo: Intel HD Graphics 3000 384 MB

#### 5. COMPUTADOR 05

- a. Nome da Máquina: Máquina Virtual - VM
- b. Proprietário: Digital Ocean
- c. Sistema Operacional: Linux
- d. Memória RAM: 512 MB
- e. Processador: Um núcleo
- f. Espaço de armazenamento: 20 GB SSD

#### 4.7.2 Recursos de *Software*

Os recursos de *software* utilizados no projeto foram subdivididos em duas categorias: *Softwares* de desenvolvimento e *softwares* de documentação.

##### 4.7.2.1 *Softwares* de desenvolvimento

A seguir estão listados os *softwares* utilizados para o armazenamento de dados, codificação e funcionamento referentes ao desenvolvimento do sistema.

1. Postgres: Sistema de gerenciamento de banco de dados utilizado para gerenciar os dados gerados, armazená-los e controlá-los;

2. GitHub: *Repositório on-line* utilizado para armazenar os dados referentes a todas as versões geradas durante o projeto;
3. Netbeans: Interface utilizada por todos os membros da equipe para a codificação, estruturação e arquitetura da aplicação;
4. WildFly: Servidor código aberto da comunidade JBoss que permite implantações de aplicações JEE.

#### 4.7.2.2 Softwares de documentação

Nesta seção estão listados os *softwares* utilizados para a criação da documentação do sistema.

1. Astah Community: Utilizado para criação dos diagramas UML;
2. Pacote Microsoft Office 2010: Utilizado para a criação e edição de planilhas, documento de texto e cronogramas;
3. MySQL Workbench: Utilizado para a criação do diagrama de entidade e relacionamento;
4. IBM Architect: *Software* utilizado para a criação dos diagramas de sequência.

## 4.8 DESENVOLVIMENTO DO PROJETO

O desenvolvimento do projeto teve como base a utilização do método ágil *scrum*. Este método foi adaptado para as realidades do grupo, as reuniões diárias

foram realizadas remotamente por *chat* através da ferramenta de comunicação Skype®.

Os papéis desempenhados pelos membros da equipe foram ajustados onde todos atuaram como donos do produto (*product owner*) e um dos membros como *scrum master*. Outra adaptação foi em relação à duração das *sprints*, que ao invés de durar de duas a quatro semanas, passou a ser apenas uma semana, quando ao final de cada semana foi realizado a demonstração dos itens do *backlog* da *Sprint* ao orientador.

Em casos de impossibilidade de comparecimento à reunião semanal, foi convencionado que a *sprint* duraria duas semanas. Foi decidido que o início da *sprint* se daria ao fim da reunião semanal, colocando novos itens do *backlog* ao *backlog* do produto na *sprint* como citado no item 4.9.2. O final de cada *sprint* aconteceu quando os itens do *backlog* do produto eram apresentados ao orientador.

#### 4.8.1 Definição do *backlog* do produto

O *backlog* do produto foi definido a partir da análise de requisitos, todos estes foram divididos em requisitos menores. Estes pequenos requisitos são o que formam o *backlog* do produto desenvolvido, constituído de uma lista de requisitos. Nesta lista pode-se incrementar mais itens ao se perceber a necessidade de alguma funcionalidade adicional ou algum *bug* de maior importância durante as *sprints*. Os itens desta lista podem ser adicionados apenas pelo dono do produto.

#### 4.8.2 Divisão do *backlog* do produto em *sprints*

O *backlog* do produto foi dividido em onze *sprints* logo no início do projeto (encontrado na seção 2.1.3 – Cronograma), este fato gerou alguma dificuldade para definir com acurácia qual *sprint* deveria conter cada parte necessária para chegar no requisito. Sendo assim, algumas atividades foram reposicionadas no decorrer do processo de desenvolvimento do projeto. Caso a equipe não atingisse os objetivos

de determinada *sprint*, as atividades incompletas retornavam para o backlog do produto, sendo, posteriormente, adicionados novamente a uma *Sprint* posterior. As *sprints* do projeto estão descritas a seguir.

Tendo em vista a organização dos itens a serem desenvolvidos, a otimização do tempo de trabalho e a programação de maneira mais efetiva, as sprints apresentam os passos que foram realizados para chegar no requisito proposto, deixando exemplificados os procedimentos e não somente o objetivo final. Essa estratégia de organização de trabalho foi a escolhida por vários motivos, tendo como o mais relevante deles o fato de assim poder ter uma visão realista do trabalho que seria realizado durante cada *sprint*. As *sprints* do projeto estão descritas a seguir.

#### 4.8.2.1 *Sprint* 1

Objetivos da *sprint*:

- Configuração do ambiente de desenvolvimento, instalação de todos os *softwares* necessários;
- Prototipação das telas do sistema;
- Configuração da infraestrutura do servidor da aplicação;
- Inserção de histórico de prioridades na análise de requisitos;
- Criação do repositório de códigos-fonte;
- Definição do cronograma do projeto;
- Divisão do *backlog* do produto nas *sprints*.

Data de início: 12/08/2015

Data de entrega: 19/08/2015

Time box: 1 semana

Demonstrado na FIGURA 10.

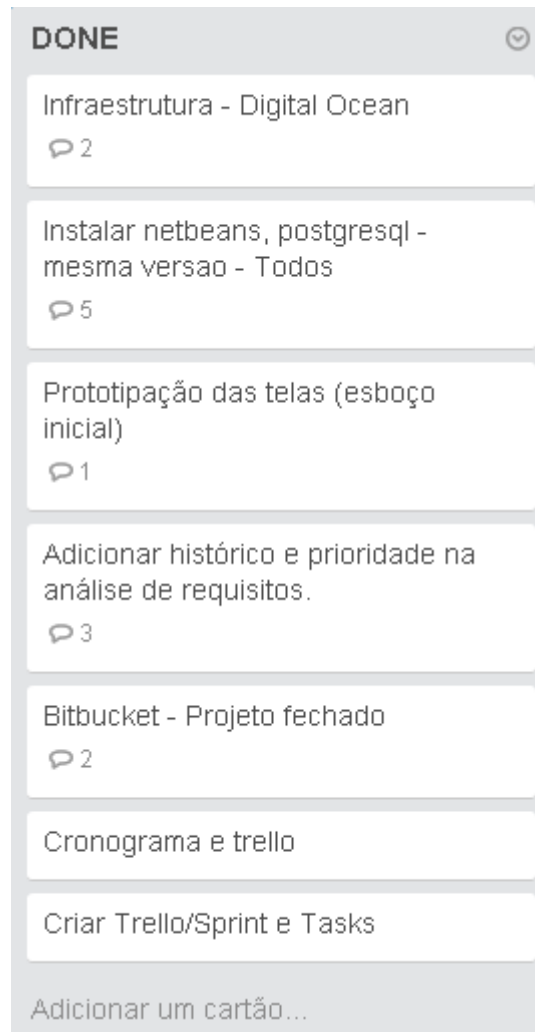


FIGURA 10 - *SPRINT 1*  
 FONTE: OS AUTORES, 2015

#### 4.8.2.2 *Sprint 2*

##### Objetivos da *Sprint*:

- Criação dos requisitos não funcionais;
- Documentação do plano de risco;
- Protótipo de CRUD rodando no servidor;
- Criação de tabela de precedência;
- Criação de plano de comunicação;
- Aperfeiçoamento dos requisitos funcionais.

Data de início: 19/08/2015

Data de entrega: 26/08/2015

Time box: 1 semana

Demonstrado na FIGURA 11.

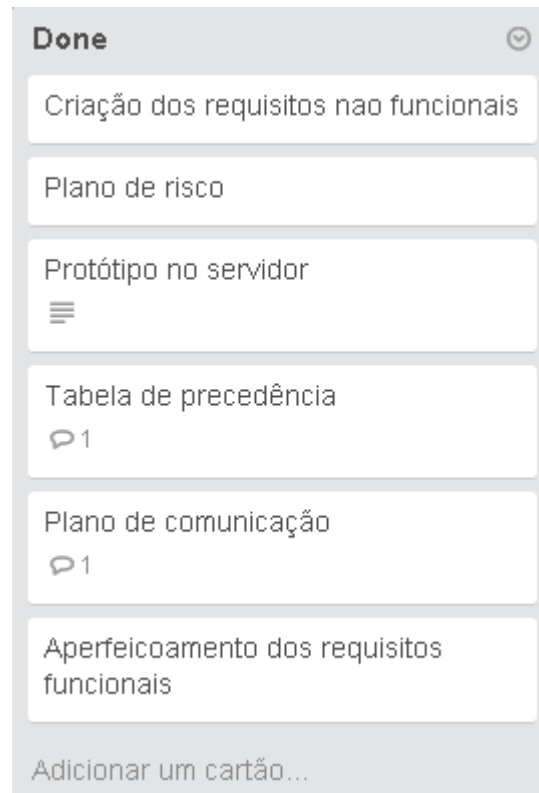


FIGURA 11 - *SPRINT 2*  
FONTE: OS AUTORES, 2015

#### 4.8.2.3 *Sprint 3*

Objetivos da *Sprint*:

- Finalização do protótipo das telas;
- Criação dos diagramas de casos de uso;
- Criação das especificações de casos de uso;
- Criação do diagrama de classes;
- Organização do trello.

Data de início: 26/08/2015

Data de entrega: 03/09/2015



Time box: 1 semana e 1 dia

Demonstrado na FIGURA 12

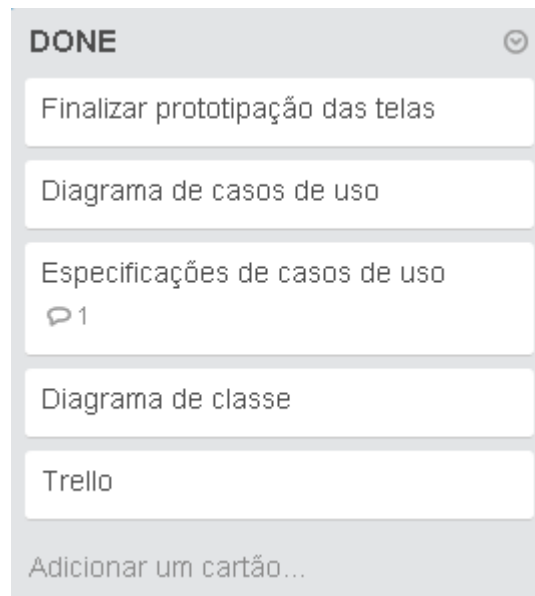


FIGURA 12 - SPRINT 3  
 FONTE: OS AUTORES, 2015

#### 4.8.2.4 Sprint 4

Objetivos da Sprint:

- Prototipação das telas não funcionais;
- Definição da arquitetura do sistema;
- Finalização da confecção das especificações de casos de uso;
- Configurações do servidor;
- Correção dos diagramas de casos de uso;
- Correção dos diagramas de classe;
- Prototipação da primeira tela funcional;
- Criação de todas as sprints no Trello;
- Inserção de requisito “Denunciar Questão” nos diagramas;
- Correção do documento de análise de requisitos.

Data de início: 03/09/2015

Data de entrega: 16/09/2015

Time box: 1 semana e 6 dias

Demonstrado na FIGURA 13.

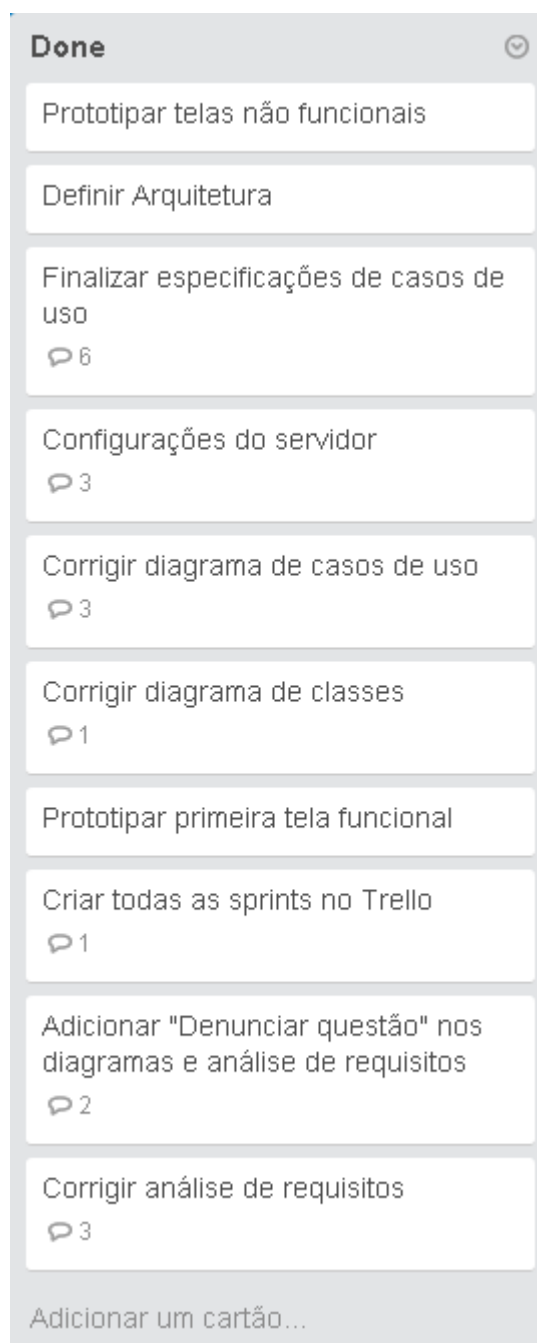


FIGURA 13 - SPRINT 4  
FONTE: OS AUTORES, 2015

#### 4.8.2.5 Sprint 5

Objetivos da Sprint:

- Prototipação de todas as telas do sistema;
- Consolidação dos diagramas em um único documento para correção;
- Criação de todas as classes e atributos;
- Implementação de possível backlog.

Data de início: 16/09/2015

Data de entrega: 23/09/2015

Time box: 1 semana

Demonstrado na FIGURA 14.

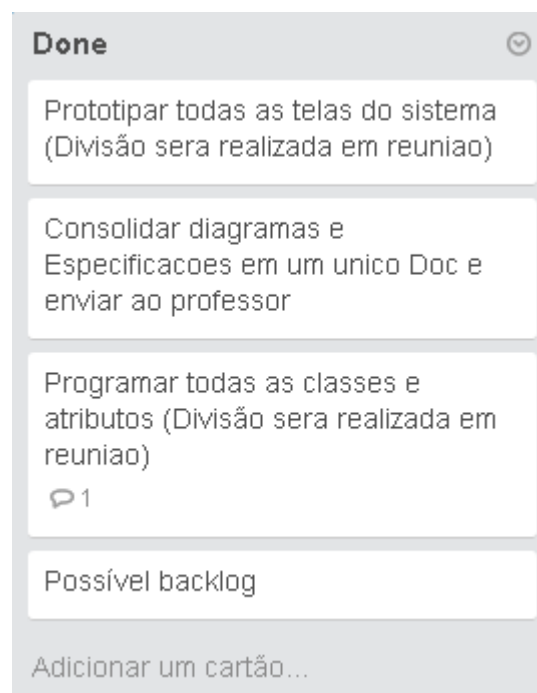


FIGURA 14 - SPRINT 5  
FONTE: OS AUTORES, 2015

#### 4.8.2.6 *Sprint 6*

Objetivos da Sprint:

- Criação do script de banco de dados;
- Programação dos objetos de acesso a dados;
- Criação do diagrama lógico do banco de dados;
- Implementação de possível backlog.

Data de início: 23/09/2015

Data de entrega: 30/09/2015

Time box: 1 semana

Demonstrado na FIGURA 15.

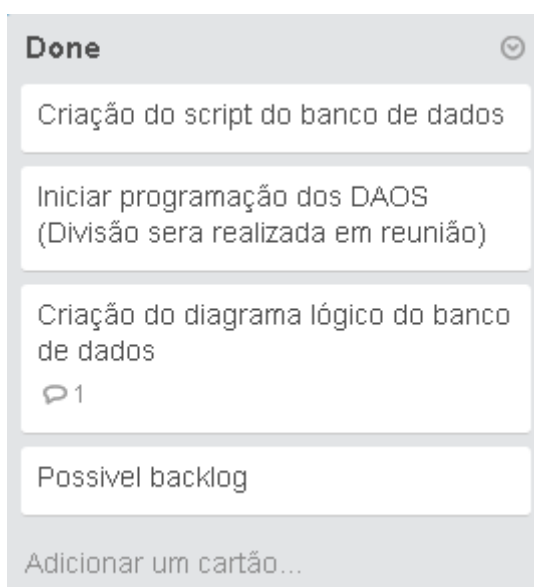


FIGURA 15 - *SPRINT 6*  
FONTE: OS AUTORES, 2015

#### 4.8.2.7 *Sprint 7*

Objetivos da Sprint:

- Criação do CRUD de professores;

- Criação do CRUD de questões;
- Criação do diagrama de sequência;
- Implementação de possível backlog.

Data de início: 30/09/2015

Data de entrega: 07/10/2015

Time box: 1 semana

Demonstrado na FIGURA 16.

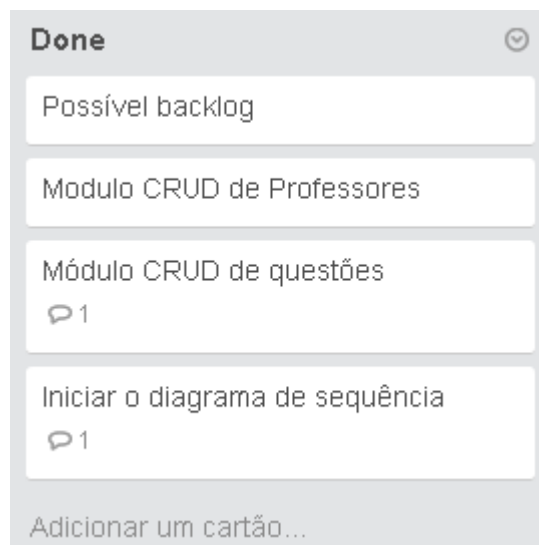


FIGURA 16 - SPRINT 7  
FONTE: OS AUTORES, 2015

#### 4.8.2.8 Sprint 8

Objetivos da Sprint:

- Implementação de login, sessão e área do professor;
- Implementação de convite a professores;
- Implementação da tela de mensagens e suas funcionalidades;
- Implementação do módulo de mensagens;
- Implementação do módulo de seguir professor;
- Separação de membros responsáveis pela implementação de outras classes;
- Implementação de possível backlog.

Data de início: 07/10/2015

Data de entrega: 14/10/2015

Time box: 1 semana

Demonstrado na FIGURA 17.

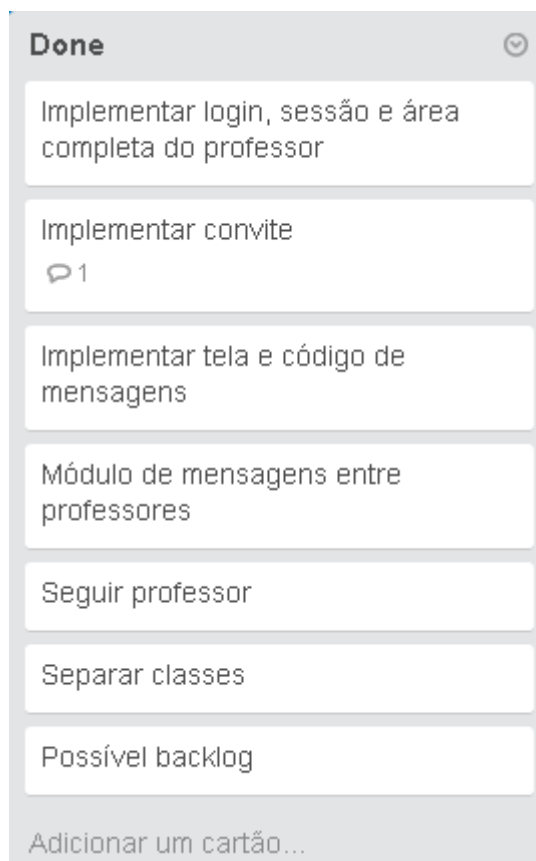


FIGURA 17 - SPRINT 8  
FONTE: OS AUTORES, 2015

#### 4.8.2.9 Sprint 9

Objetivos da Sprint:

- Implementação de comentários na questão;
- Implementação de interação entre professores;
- Implementação de “likes” nas questões;
- Implementação de possível backlog.

Data de início: 14/10/2015

Data de entrega: 21/10/2015

Time box: 1 semana

Demonstrado na FIGURA 18.

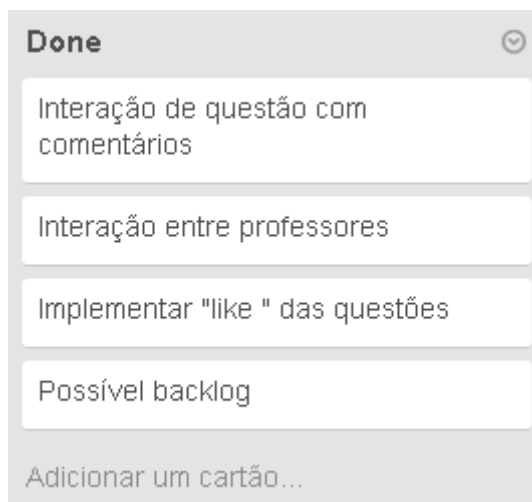


FIGURA 18 - SPRINT 9  
FONTE: OS AUTORES, 2015

#### 4.8.2.10 Sprint 10

Objetivos da Sprint:

- Implementação de denúncia de questões;
- Implementação de peso nas questões por prova;
- Implementação de histórico de provas;
- Implementação de geração de relatório de questões mais usadas e mais difíceis;
- Implementação de página para gerenciar questões;
- Implementação de possível backlog.

Data de início: 21/10/2015

Data de entrega: 28/10/2015

Time box: 1 semana

Demonstrado na FIGURA 19.

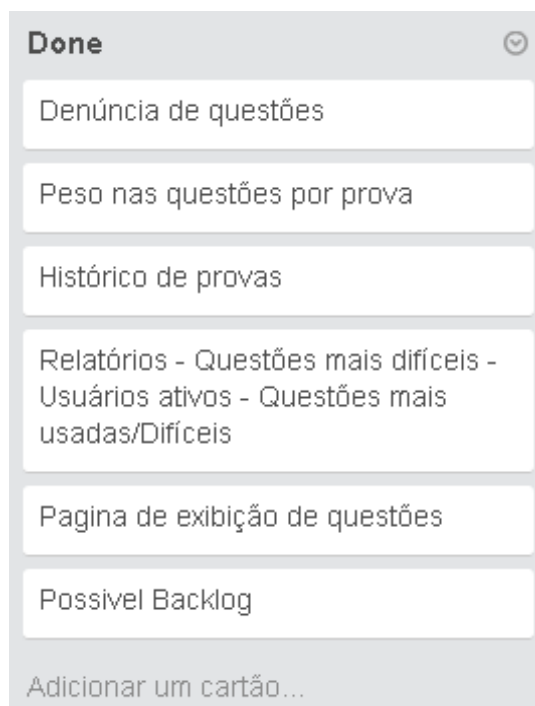


FIGURA 19 - *SPRINT 10*  
FONTE: OS AUTORES, 2015

#### 4.8.2.11 *Sprint 11*

Objetivos da Sprint:

- Implementação de tipo de prova;
- Implementação de cadastro de desempenho;
- Implementação de modelos de prova;
- Implementação de possível backlog.

Data de início: 28/10/2015

Data de entrega: 04/11/2015

Time box: 1 semana

Demonstrado na FIGURA 20.



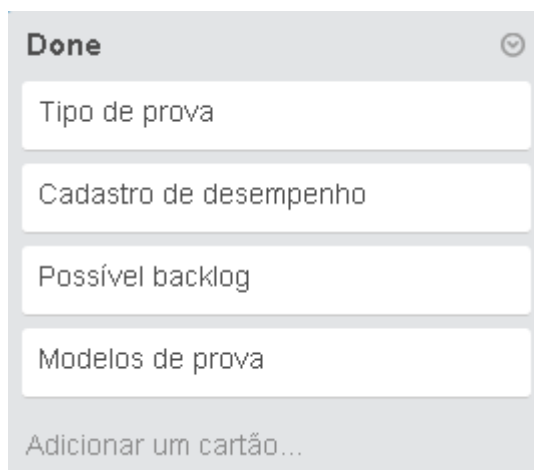


FIGURA 20 - SPRINT 11  
FONTE: OS AUTORES, 2015

#### 4.8.2.12 *Sprint 12*

Objetivos da Sprint:

- Implementação de geração de prova baseada em desempenho anterior dos alunos;
- Ajustes na documentação;
- Criação do script de apresentação do sistema;
- Implementação de possível backlog.

Data de início: 04/11/2015

Data de entrega: 11/11/2015

Time box: 1 semana

Demonstrado na FIGURA 21.

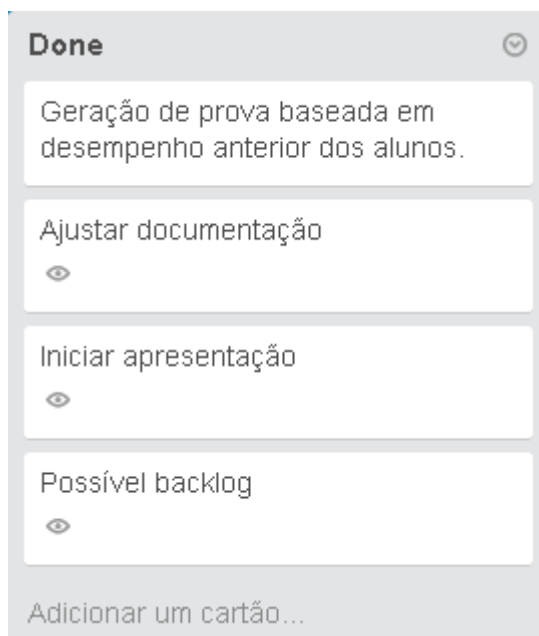


FIGURA 21 - *SPRINT 12*  
FONTE: OS AUTORES, 2015

#### 4.8.2.13 *Sprint 13*

Objetivos da Sprint:

- Finalização dos ajustes na documentação;
- Finalização do script apresentação;
- Realização do treinamento de apresentação do sistema;
- Implementação de possível backlog.

Data de início: 11/11/2015

Data de entrega: 18/11/2015

Time box: 1 semana

Demonstrado na FIGURA 22.

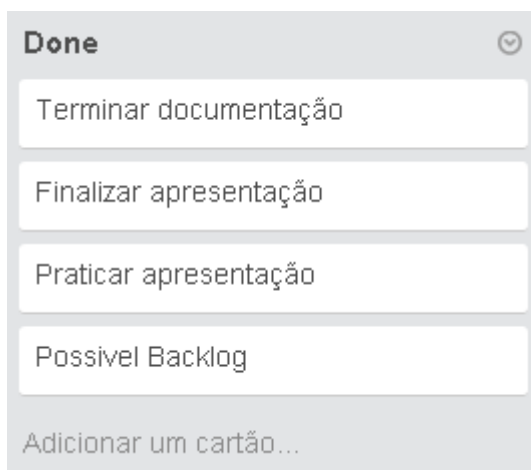


FIGURA 22 - *SPRINT 13*  
FONTE: OS AUTORES, 2015

#### 4.8.2.14 *Sprint 14*

Objetivos da Sprint:

- Implementação de possível backlog.

Data de início: 18/11/2015

Data de entrega: 25/11/2015

Time box: 1 semana

Demonstrado na FIGURA 23.

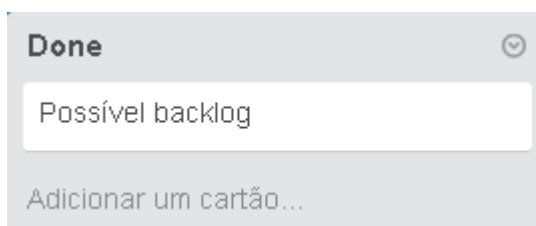


FIGURA 23 - *SPRINT 14*  
FONTE: OS AUTORES, 2015

#### 4.8.2.15 *Sprint 15*

Objetivos da Sprint:

- Implementação de possível *backlog*.

Data de início: 25/11/2015

Data de entrega: 02/12/2015

Time box: 1 semana

Demonstrado na FIGURA 24.

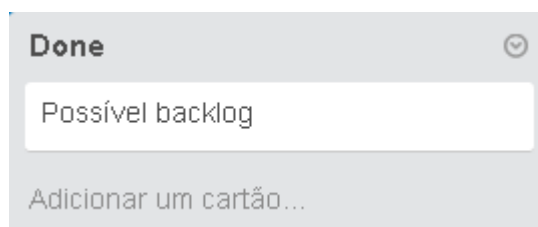


FIGURA 24 - *SPRINT 15*  
FONTE: OS AUTORES, 2015

## 5 APRESENTAÇÃO DO SOFTWARE

Os próximos itens fornecerão uma visão mais detalhada do funcionamento do *software* proposto através das interfaces relacionadas a cada perfil (professor ou administrador).

### 5.1 Interfaces e Funcionalidades

A utilização do sistema se dá por uma interface de login (FIGURA 26) em que ao usuário ser autenticado, caso ele seja um professor, o sistema o redireciona para o painel com as funcionalidades de professor, caso contrário, o sistema o redireciona para o painel de administrador. A FIGURA 25 demonstra o funcionamento de acesso ao sistema.

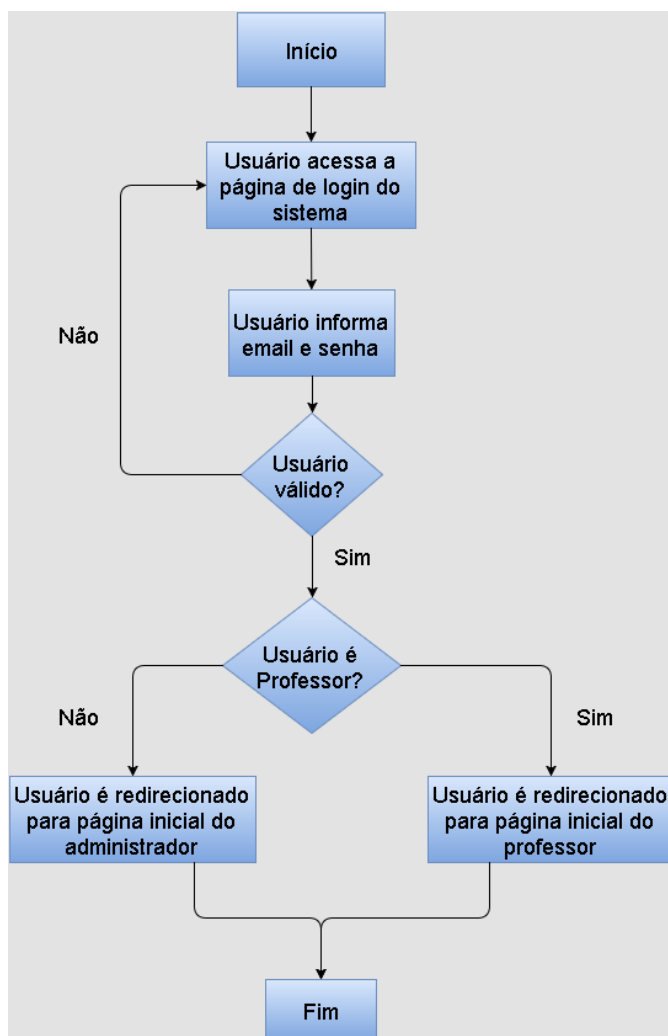


FIGURA 25 - FLUXO DE ACESSO AO SISTEMA NETQ  
FONTE: OS AUTORES, 2015

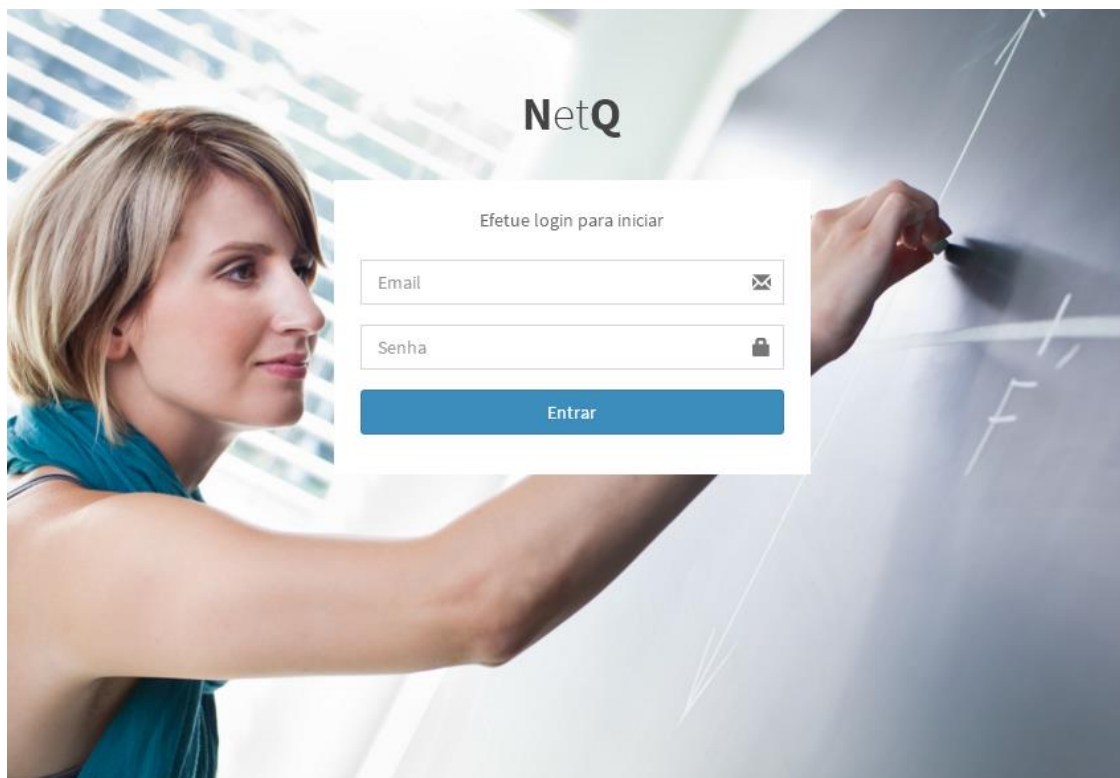


FIGURA 26 - TELA DE LOGIN.  
FONTE: OS AUTORES, 2015

### 5.1.1 Perfil do professor(a)

Depois da autenticação do login no sistema, caso o usuário seja um professor, o sistema direciona o usuário para a página inicial que então é apresentada como mostrado na FIGURA 27.

Nesta página, que se denomina linha do tempo, são apresentadas as ações dos professores os quais o usuário logado segue, tais como:

- Professor seguido criou questão;
- Professor seguido curtiu questão;
- Professor seguido comentou questão;
- Professor seguido seguiu alguém.

Estas funcionalidades são descritas no Apêndice B, com o título de “UC05 – Ver *timeline*”.

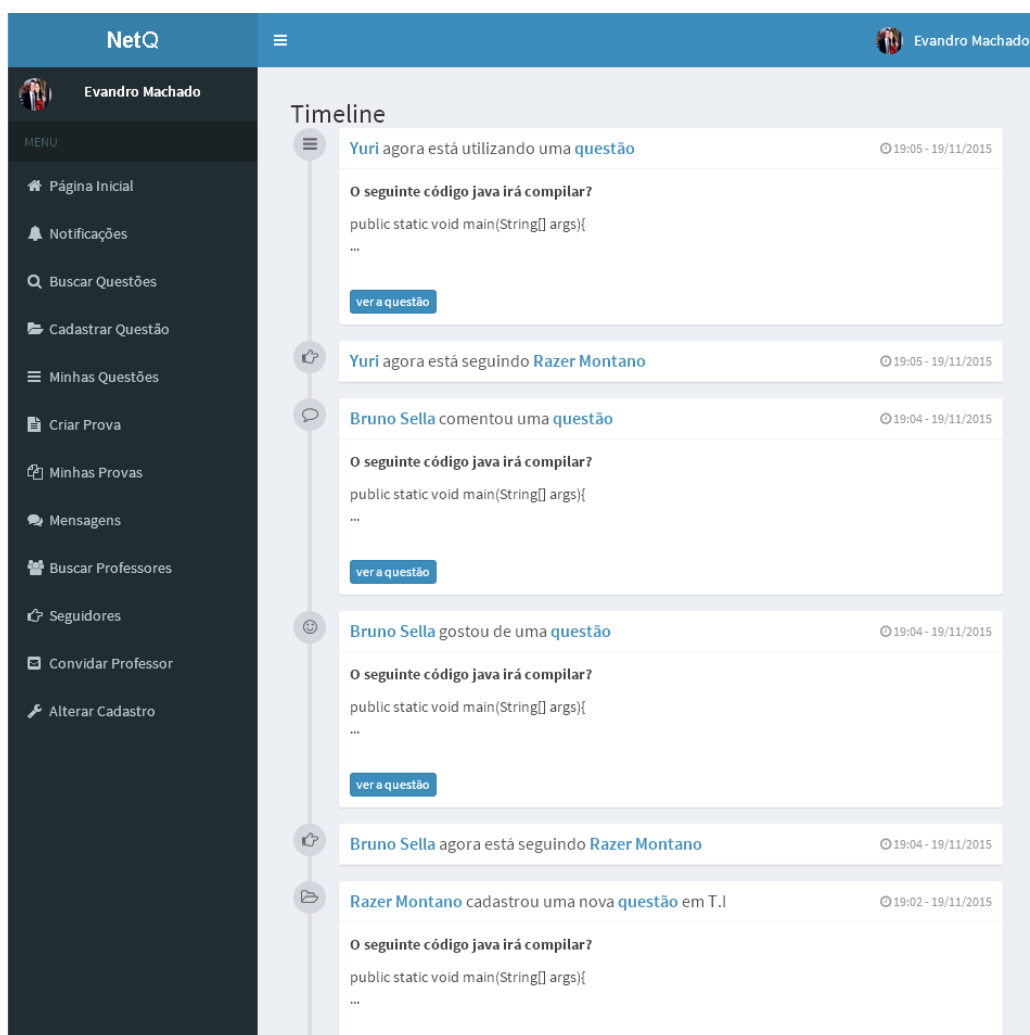


FIGURA 27 - PÁGINA INICIAL DO PROFESSOR(A): “TIMELINE”  
 FONTE: OS AUTORES, 2015

O menu lateral apresenta as principais funcionalidades fornecidas ao usuário com perfil de professor conforme mostra a FIGURA 28. O quadro 36 mostra a correspondência entre as funcionalidades apresentadas pela FIGURA 28 e seus respectivos casos de uso, os quais são encontrados no Apêndice A e suas especificações no Apêndice B.





FIGURA 28 - MENU DE FUNCIONALIDADES DO PROFESSOR(A)  
FONTE: OS AUTORES, 2015

<b>Opção de Menu</b>	<b>Caso de Uso Relacionado</b>
Página inicial	N/A
Notificações	N/A
Buscar Questões	UC08 - Buscar Questão
Cadastrar Questão	UC11 - Gerenciar Questões
Minhas Questões	UC02 - Visualizar Questão
Criar Prova	UC09 - Gerenciar Prova
Minhas Provas	UC09 - Gerenciar Prova
Mensagens	UC07 - Enviar Mensagem Privada
Buscar Professores	UC13 - Gerenciar Professores
Seguidores	UC13 - Gerenciar Professores
Convidar Professor	UC05 - Convidar Professor
Alterar Cadastro	UC12 - Gerenciar Perfil

QUADRO 37 - QUADRO DE CORRELAÇÃO ENTRE OPÇÃO DO MENU E CASOS DE USO DO PROFESSOR(A).  
 FONTE: OS AUTORES, 2015

### 5.1.2 Perfil do administrador(a)

Depois da autenticação do login no sistema, caso o usuário seja um administrador, o sistema direciona o usuário para a página inicial que então é apresentada como mostrado na FIGURA 29.

Nesta página, chamada “Painel do Administrador”, são apresentados alguns relatórios do sistema, tais como:

Gerar relatório de questão mais difícil; Gerar relatório de quantos usuários se cadastraram no mes atual; Gerar Relatório de quantidade de questões existentes; Gerar relatórios de quantas provas foram geradas;

- Questões mais difíceis;
- Quantidade de novos usuários cadastrados;
- Quantidade de questões cadastradas;
- Quantidade de provas geradas;

Estas funcionalidades são descritas no Apêndice B, com o título de “UC05 – Ver timeline”.

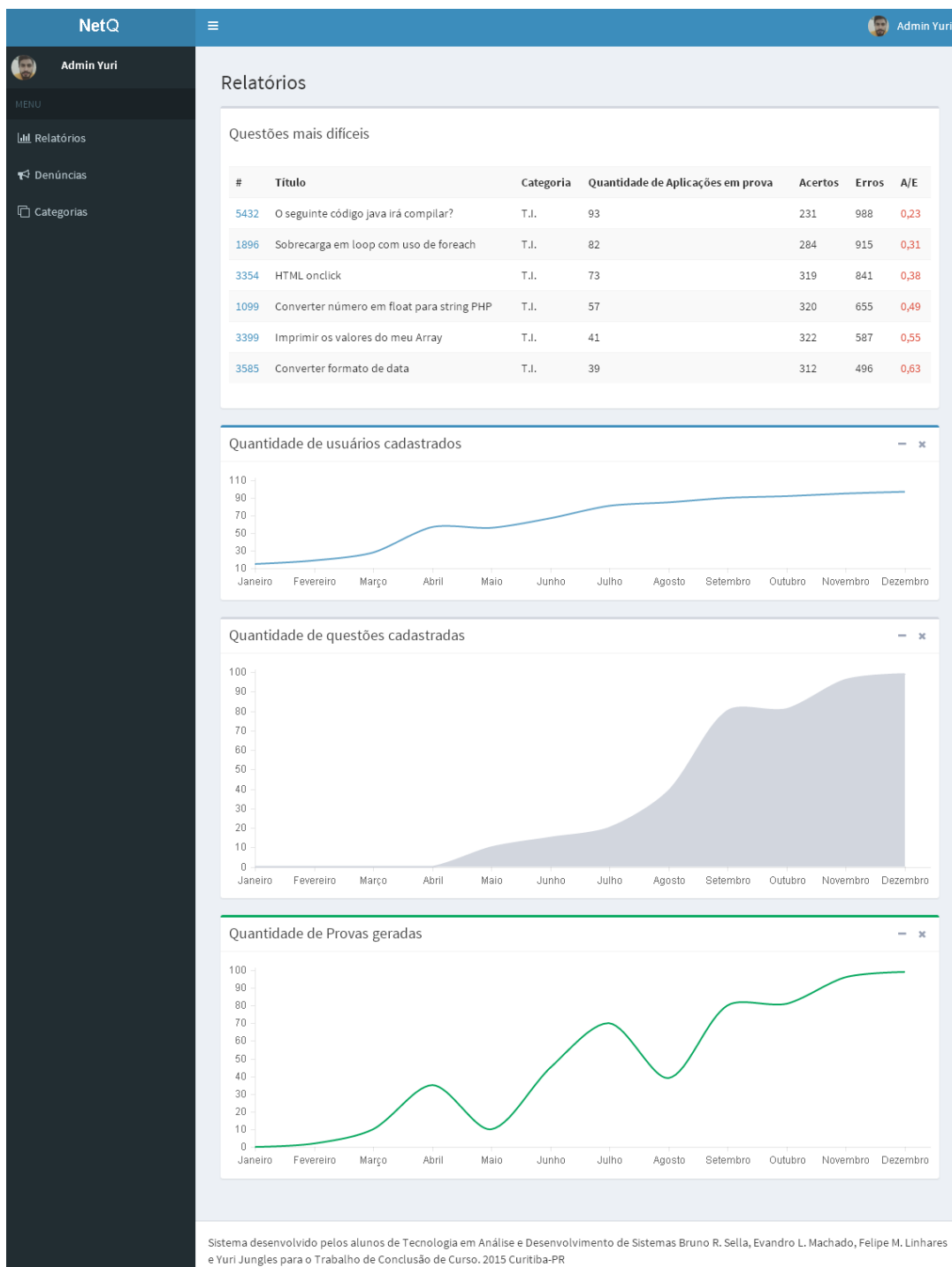


FIGURA 29 - PÁGINA INICIAL DO ADMINISTRADOR(A): “RELATÓRIOS”  
 FONTE: OS AUTORES, 2015

O menu lateral apresenta as principais funcionalidades fornecidas ao usuário com perfil de administrador conforme mostra a FIGURA 30. O quadro 37 mostra a correspondência entre as funcionalidades apresentadas pela FIGURA 30 e seus respectivos casos de uso, os quais são encontrados no Apêndice A e suas especificações no Apêndice B.

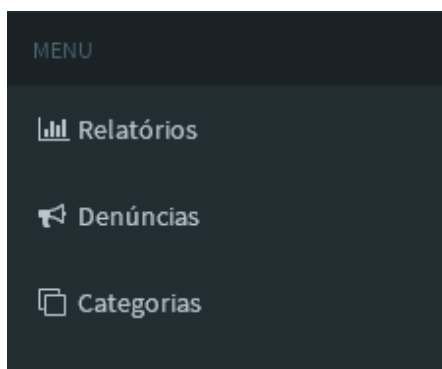


FIGURA 30 - MENU DE FUNCIONALIDADES DO “ADMINISTRADOR(A)”  
 FONTE: OS AUTORES, 2015

Opção de Menu	Caso de Uso Relacionado
Relatórios	UCXX - Gerar Relatórios por YZ
	UCXX - Gerar Relatórios por YZ
	UCXX - Gerar Relatórios por YZ
	UCXX - Gerar Relatórios por YZ
Denúncias	UC02 - Visualizar Questão
Categorias	UC09 - Gerenciar Prova

QUADRO 38 - QUADRO DE CORRELAÇÃO ENTRE OPÇÃO DO MENU E CASOS DE USO DO ADMINISTRADOR(A).  
 FONTE: OS AUTORES, 2015

## 6 CONSIDERAÇÕES FINAIS

Nas próximas seções serão apresentadas as dificuldades encontradas durante todo o processo de desenvolvimento do *software*, o aprendizado obtido e também os requisitos e melhorias determinados como implementações futuras no sistema existente.

### 6.1.1 CONCLUSÃO

Este trabalho apresentou todas as etapas de desenvolvimento do sistema NetQ, durante este processo encontraram-se algumas dificuldades, tais como: curva de aprendizagem da tecnologia JSF, conflito de horários do grupo gerando impossibilidade de reuniões diárias presenciais, o prazo para entrega foi de aproximadamente quatro meses o que, para a quantidade de requisitos propostos, pode ser considerado bastante curto. Também pode se observar que a metodologia utilizada para a realização da documentação não foi muito efetiva, pois a tentativa de construir grande parte dos diagramas e especificações de caso de uso logo na primeira etapa do desenvolvimento, fez com que no final do projeto houvesse a necessidade atualização de toda a documentação gerando retrabalho necessário. Outra dificuldade foi em questão de organização da estrutura do banco de dados, pois a cada alteração era necessário atualizar a estrutura das tabelas no computador de cada membro.

Infelizmente, devido à escassez de tempo, não houve a possibilidade de realizar testes com o usuário final. Porém foi realizado a inserção de oitenta questões no banco de questões e vários testes foram realizados pela equipe de desenvolvimento.

Os resultados destes testes foram bastante positivos, pois foi possível observar que tanto o processo de criação de questões, de provas e correção das mesmas foi bastante otimizado, permitindo grande economia de tempo e esforço por parte do professor.

Pode se afirmar que o trabalho foi bem sucedido pois todos os objetivos específicos foram alcançados. O único quesito que não foi possível validar foi o de verificar se houve efetivamente troca de conhecimento entre professores universitários, esta impossibilidade se dá pelo fato de que não houve validação do produto com o usuário final como já dito anteriormente.

O processo de desenvolvimento do sistema forneceu à equipe várias novas experiências, os aprendizados mais importantes observados foram os seguintes: a tecnologia JSF que inicialmente dificultou, de certa forma, o início do desenvolvimento pela inexperiência da equipe, porém, após o aprendizado, ela potencializou vários aspectos na qualidade do código como: menor quantidade de código para realizar operações, gerenciamento do escopo dos objetos, código mais fácil de se compreender e realizar manutenção, aumento da produtividade e velocidade devido as questões anteriores supracitadas. O padrão MVC também contribuiu muito para a organização do projeto pois permitiu a clara divisão das responsabilidades do projeto, como telas (visão), beans gerenciados (controle), beans (modelo). Outro aprendizado foi a utilização de *software* para controle de versão, pois praticamente não houve problemas na questão de versão do código. Outro aprendizado foi em relação à forma como foi realizada a documentação, para um próximo trabalho consideramos que esta poderia ser realizada de forma incremental, junto ao desenvolvimento, para evitar retrabalhos e deixando o processo mais fiel à metodologia ágil.

## 6.2 Implementações Futuras

Ao longo da confecção do sistema pode se observar a necessidade de algumas funcionalidades adicionais às inicialmente propostas. Algumas outras funcionalidades também acabaram por não ser implementadas, a maioria por falta de tempo hábil, tendo em vista que a duração do projeto foi de aproximadamente quatro meses. Todas estas requisitos novos e os penderes ficam como sugestão para implementações futuras, os já existentes são:

- RF16 – Avaliação de questões;
- RF17 – Modelos de prova;
- RF18 – Peso nas questões por prova;

- RF19 – Geração de prova baseada em desempenho anterior dos alunos.

As novas funcionalidades sugeridas são:

- Correção das provas através de foto do gabarito;
- Postagem de ações realizadas pelo professor no Facebook;
- Relatório de quantidade de questões por área de conhecimento;
- Possibilidade de gerar testes online para os alunos.

## REFERÊNCIAS

BECK, K.; BEEDLE, M.; BENNEKUM, A.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN R. C.; MELLO, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. **Manifesto para Desenvolvimento Ágil de Software**, 2001. Disponível em: <<http://agilemanifesto.org/iso/ptbr/>> . Acesso em: 02/11/2015.

BERMIER,R. Total security in a Postgre SQL database. IBM, 2009. Disponível em: <<http://ibm.com/develiperworks/library/os-postgresecurity>>.Acesso em: 05/11/2015.

BEZERRA, E. **Princípios De Análise E Projeto De Sistemas Com Uml**, 3ª Ed. Elsevier: Brasil, 2015.

BOEHM, Barry W. Software risk management: principles and practices. **Software, IEEE**, v. 8, n. 1, p. 32-41, 1991.

PRAL. Disponível em :< <http://www.pral.com.br>>. Acesso em : 11/12/2015.

SGP. Disponível em :< <http://starlinetecnologia.com.br/sgp>>. Acesso em : 11/12/2015.

QUESTOES DE CONCURSOS. Disponível em :< <https://www.qconcursos.com>>. Acesso em : 11/12/2015.

BOOTSTRAP. Disponível em :< <http://www.bootstrapzero.com/bootstrap-template/grid>>. Acesso em : 28/10/2015.

CHACON,S; STRAUB,B. **Pro Git (2nd Edition)**, Apress, 2014

CISNEIROS, D. **Modelo de Desenvolvimento Ágil . SCRUM**, 2009. Disponível em: < <http://www.devin.com.br/modelo-scrum/>>. Acesso em: 10/11/2015.

COHN, M.; FORD, D. Introducing an agile process to an organization. **Computer**, n. 6, p. 74-78, 2003.

DEITEL, P.; DEITEL, H. **Java How to program**. Prentice Hall Press, 2011.

FRAIN, B. **Responsive web design with HTML5 and CSS3**. Packt Publishing Ltd, 2012.

FOWLER, M. **UML Essencial: um breve guia para linguagem padrão**. Bookman, 2014.



GIT. Disponível em: <<https://git-scm.com/figures/18333fig0101-tn.png>> Acessado em: 10/11/2015.

GITa. Disponível em: <<https://git-scm.com/figures/18333fig0102-tn.png>>. Acessado em: 10/11/2015.

GITb. Disponível em: <<https://git-scm.com/figures/18333fig0103-tn.png>>. Acessado em: 10/11/2015.

GUEDES, G. TA. **UML 2: uma abordagem prática**. São Paulo: Novatec, v. 485, p. 3, 2009.

IEA. **Definition and Domains of ergonomics**. Disponível em: <<http://www.iea.cc/whats/index.html>>. Acesso em: 19/11/2015.

KNIBERG, H; SKARIN, M. **Kanban e Scrum - obtendo o melhor de ambos**. Estados Unidos: C4Media Inc, 2009.

LIBARDI, Paula LO; BARBOSA, Vladimir. **Métodos Ágeis**. 2010.

LIU, L. *et al.* Design and actualization of universal MVC pattern [J]. **Journal of University of Science and Technology of China**, v. 6, p. 017, 2010.

NIELSEN, J. **Usability Engineering**. New Jersey: Academic Press, 1993.

PEREIRA, Paulo; TORREÃO, Paula; MARÇAL, Ana Sofia. Entendendo Scrum para gerenciar projetos de forma ágil. **Mundo PM**, v. 1, p. 3-11, 2007.

PMI. **Um Guia do Conhecimento em Gerenciamento de Projetos**. Guia PMBOK®. Quarta Edição – EUA: Project Management Institute, 2008.

PRESSMAN, Roger S. **Engenharia de software**. McGraw Hill Brasil, 2011.

ORACLE. Disponível em: <<http://www.oracle.com/technetwork/java/javaee/overview-140548.html>>. Acessado em: 26/10/2015.

SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum: Um guia definitivo para o Scrum: As regras do jogo**. [SI], 2013.

SCHWARTZ, J. I. **Construction of software, Problems and Practicalities**. Addison-Wesley, Reading, MA, 1975.

SOMMERVILE, Ian. **“Engenharia de Software”**, Addison Wesley, São Paulo, 2003.

STEFFEN, J. B. O que são essas tais de metodologias ágeis? IBM, 2012 . Disponível em: <[https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/mas\\_o\\_que\\_s\\_c3\\_a3o\\_essas\\_tais\\_de\\_metodologias\\_\\_c3\\_a1geis?lang=en](https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/mas_o_que_s_c3_a3o_essas_tais_de_metodologias__c3_a1geis?lang=en)>. Acessado em: 13/11/2015.

TRELLO. Disponível em: <<http://help.trello.com/article/708-what-is-trello>>. Acessado em: 07/11/2015.

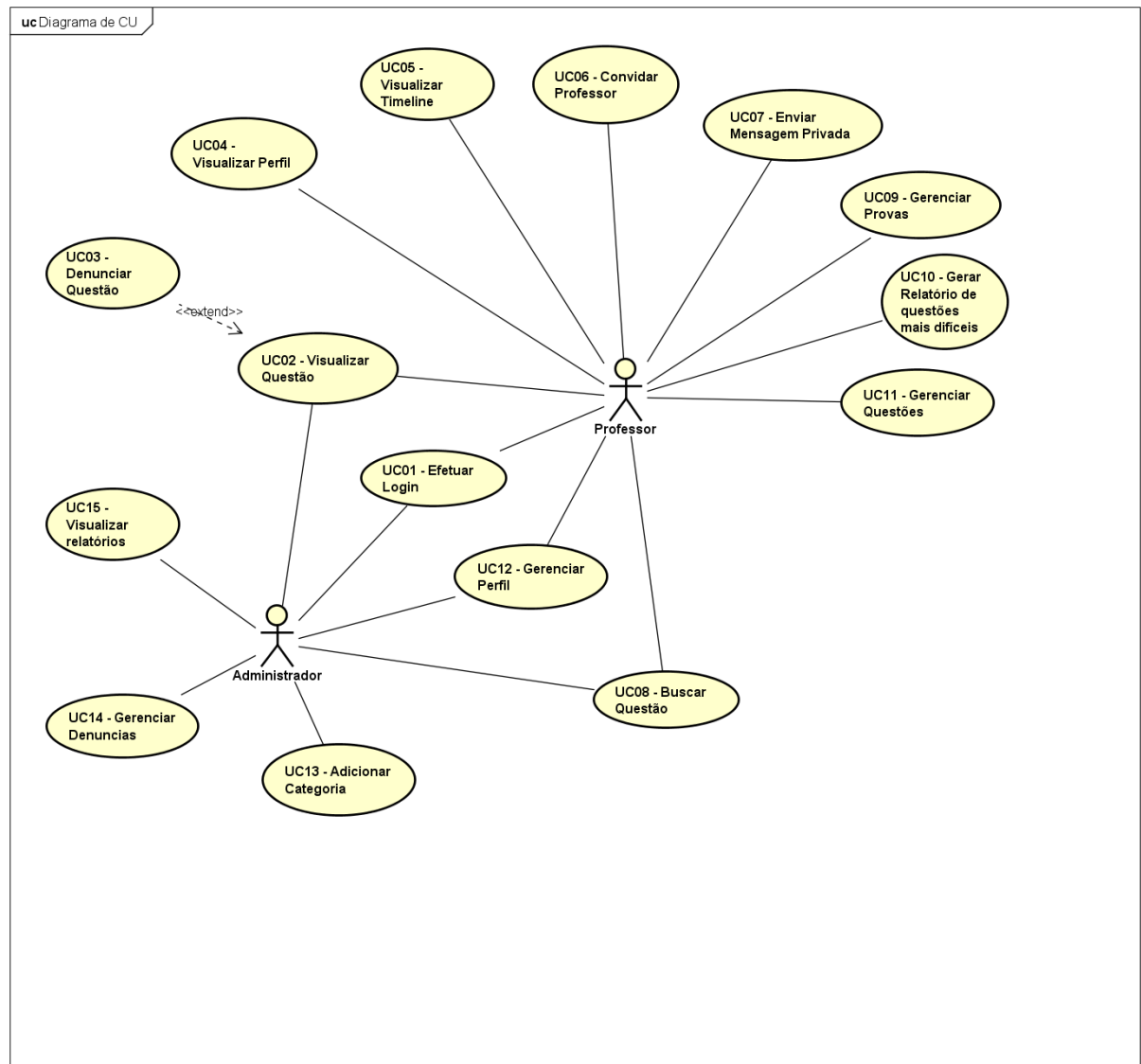
TREVISAN, Diogo Fernando; DA SILVA SACCHI, Rodrigo P.; SANABRIA, Lino. Estudo do Padrão Avançado de Criptografia AES–Advanced Encryption Standard. **Revista de Informática Teórica e Aplicada**, v. 20, n. 1, p. 13-24.

VIRTX. Soluções em TI . Virtx, 2014. Disponível em: <<http://virtx.com.br/wp-content/uploads/2014/12/vps.png>>. Acessado em: 06/11/2015.

## LISTA DE APÊNDICES

APÊNDICE A – DIAGRAMA DE CASOS DE USO .....	96
APÊNDICE B – ESPECIFICAÇÕES DE CASOS DE USO .....	97
APÊNDICE C – DIAGRAMA DE CLASSES.....	135
APÊNDICE D – PLANO DE RISCO.....	136
APÊNDICE E – DIAGRAMA ENTIDADE RELACIONAMENTO .....	137
APÊNDICE F – WBS .....	138
APÊNDICE G – PLANO DE COMUNICAÇÃO.....	139
APÊNDICE H – DIAGRAMAS DE SEQUÊNCIA.....	139
APÊNDICE I – ACORDO DE NÍVEL DE SERVIÇO.....	144
APÊNDICE J – DIAGRAMA DE CLASSES DE PROJETO.....	145

## APÊNDICE A – DIAGRAMA DE CASOS DE USO

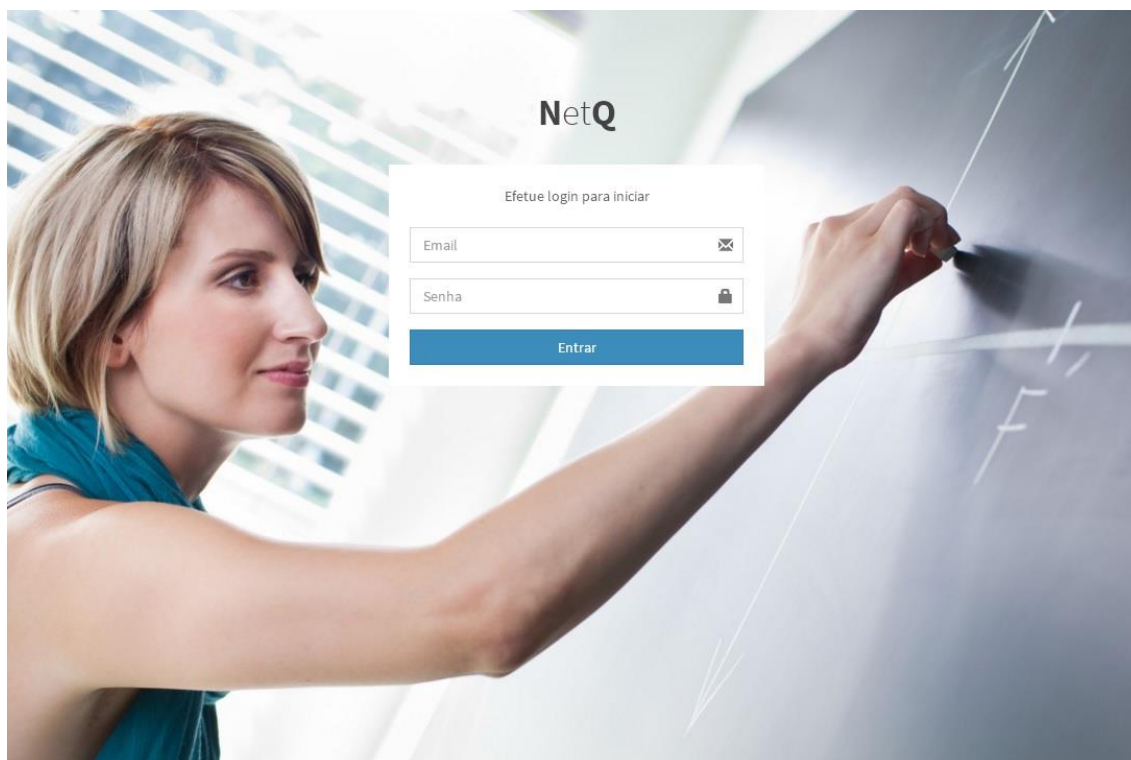


## APÊNDICE B – ESPECIFICAÇÕES DE CASOS DE USO

**UC01 – Efetuar login**

Descrição: Este caso de uso serve para que um Professor ou Administrador efetue login no sistema.

**Data View:** DV01 – Tela Login.png.



DV01 – Tela Login.png.

**Pré-condições**

Não há pré-condições.

**Pós-condições**

Após o fim normal deste caso de uso o sistema deve:

1. Permitir ao usuário acesso ao sistema.

**Ator Principal**

Professor ou Administrador.

**Fluxo de Eventos Principal**

1. O sistema apresenta a tela. (DV01)
2. O usuário insere seu email no campo email.
3. O usuário insere sua senha no campo Senha.
4. O usuário clica no botão Entrar.
5. O sistema autentica o login do usuário. (R01) (E01)
6. O sistema inicia a sessão com o login efetuado para o respectivo perfil e o redireciona para sua tela inicial. (R02)

**Fluxos Alternativos**

Nenhum fluxo alternativo.

**Fluxos de Exceção**

E01. Email e senha inválidos:

1. O sistema recarrega a pagina para que o usuário digite suas credenciais novamente.
2. O Caso de Uso é reiniciado.

**Regras de Negócio**

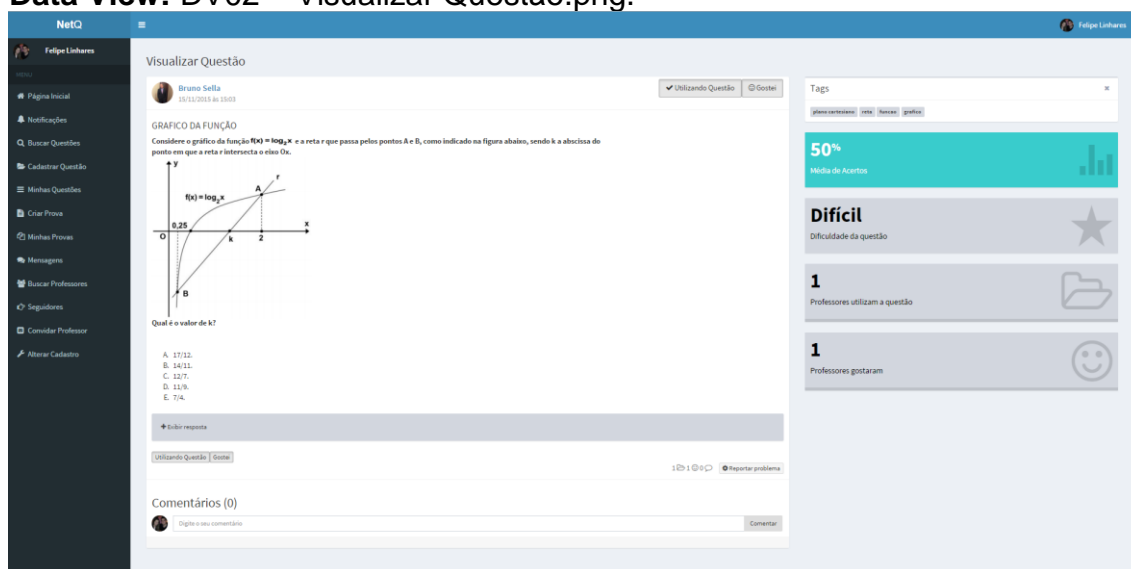
R01. Para autenticar o login, o sistema faz uma busca pelo email no banco de dados, descriptografa a respectiva senha do banco de dados e a compara com a senha digitada.

R02. Caso as credenciais autenticadas pertençam a um perfil de professor, o sistema redireciona para o UC05 – Visualizar Timeline. Caso pertençam a um gerente, o redirecionamento ocorrerá para o UC15 – Visualizar Relatórios.

## UC02 – Visualizar Questão

Descrição: Este caso de uso serve para que um usuário visualize, curta, comente ou adicione uma questão a sua lista de questões.

Data View: DV02 – Visualizar Questão.png.



DV02 – Visualizar Questão.png.

### Pré-condições

Este caso de uso pode iniciar somente se:

1. Usuário estiver autenticado no sistema.
2. O usuário deve ter acesso a questão. (R01)

### Pós-condições

Não há nenhuma pós-condição obrigatória

### Ator Principal

Professor

Administrador

### Fluxo de Eventos Principal

1. O sistema carrega os dados.
2. O sistema apresenta a tela. (DV02)
3. O usuário clica no botão "Utilizar questão". (A01) (A02) (A03) (A04) (A05)
4. O sistema adiciona a questão à aba de "Questões em minha lista" da lista de questões do professor.
5. O caso de uso é encerrado.

## Fluxos Alternativos

A01: Botão “Gostei” é pressionado.

1. O sistema contabiliza um voto ao total de “gostei” da questão; (R03)
2. O caso de uso é reiniciado.

A02: “Exibir Resposta” é pressionado

1. O sistema exibe a(s) resposta(s) da questão.

A04: O usuário redige um comentário e pressiona o botão “Comentar”

1. O sistema insere o texto do campo comentário aos comentários da questão.

A05: Botão “Reportar Problema” é pressionado

1. O caso de uso é redirecionado para o UC03 – Denunciar Questão;
2. O caso de uso é encerrado.

## Regras de Negócio

R01. A questão só pode ser visualizada mediante cumprimento de pelo menos uma das seguintes condições:

- O usuário for um Administrador;
- O usuário for o professor autor da questão;
- O questão for pública.

R02. Após acionado, o botão “Utilizar Questão” fica “pressionado” e seu texto muda para “Utilizando Questão” demonstrar que a ação já aconteceu e o contador de utilizações é atualizado. Caso o usuário acione o botão já pressionado a ação de utilizar é desfeita.

R03. Após acionado, o botão “Gostei” fica “pressionado” e seu texto muda para “Gostei” para demonstrar que a ação já aconteceu e o contador de pessoas que gostaram é atualizado. Caso o usuário acione o botão já pressionado a ação de gostar é desfeita.



## UC03 – Denunciar Questão

Descrição: Este caso de uso serve para que um Professor possa fazer a denúncia de alguma questão que contenha algum erro ou conteúdo impróprio.

**Data View:** DV03 – Tela Denúncia.png.

Reportar Problema X

Qual é o problema?

A Questão possui erros

A Questão está na categoria errada

A Questão é ofensiva

Outros

Descreva o problema:

Fechar Enviar Problema

DV03 – Tela Denúncia.png.

### Pré-condições

Este caso de uso pode iniciar somente se:

1. O usuário possuir cadastro no sistema.
2. O usuário acionar o botão “Reportar Problema” do UC02.

### Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Cadastrar no banco de dados a denúncia da questão e notificar o Administrador sobre a nova denúncia.

### Ator Principal

Professor

Administrador

### **Fluxo de Eventos Principal**

1. O sistema apresenta a tela. (DV03)
2. O usuário escolhe um dos principais problemas listados. (R01) (A01)
3. O usuário redige uma descrição do problema no campo “Descreva o problema”.
4. O usuário aciona o botão “Enviar Problema”. (E01)
5. O sistema cadastra a denúncia e notifica o administrador que há uma nova denúncia.

### **Fluxos Alternativos**

A01: Botão “Fechar” é pressionado.

1. O sistema fecha a pop-up.

### **Fluxos de Exceção**

E01. Dados insuficientes:

1. O usuário não escolheu um dos principais problemas listados ou não descreveu o problema.
2. O sistema exibe uma mensagem abaixo do campo em questão enfatizando que o mesmo é mandatório.

### **Regras de Negócio**

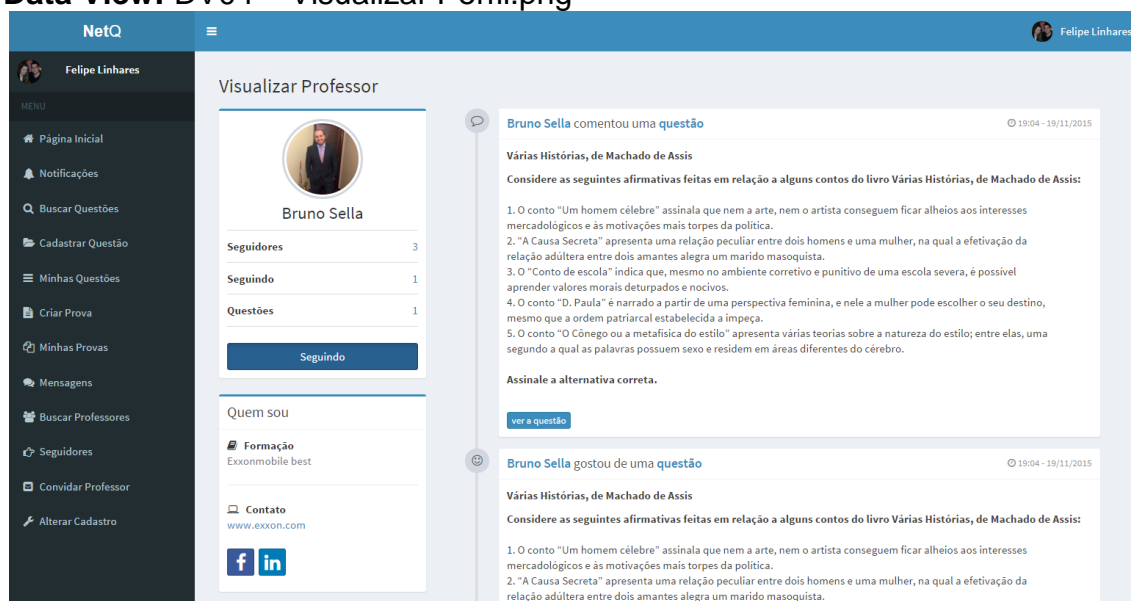
R01. Os principais problemas listados são:

- A questão possui erros;
- A questão está na categoria errada;
- A questão é ofensiva;
- Outros

## UC04 – Visualizar Perfil

Descrição: Este caso de uso serve para que um Professor visualize o perfil de outro Professor e o siga.

### Data View: DV04 – Visualizar Perfil.png



DV04 – Visualizar Perfil.png

### Pré-condições

Este caso de uso pode iniciar somente se:

1. O professor estiver autenticado no sistema.

### Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Guardar a informação de que o professor foi seguido.

### Ator Principal

Professor

### Fluxo de Eventos Principal

1. O sistema carrega os dados. (R01)
2. O sistema apresenta a tela. (DV04)
3. O usuário clica no botão “seguir”. (R02) (A01) (A02)
4. O sistema adiciona o usuário à lista de “seguindo” do usuário.
5. O caso de uso é encerrado.

### Fluxos Alternativos

A01: *Link* “questão” é acionado.

1. O sistema redireciona o professor para a página da respectiva questão.

A02: *Link* com o nome de um professor é acionado.

1. O sistema redireciona o professor para a página do perfil do respectivo professor.

### **Regras de Negócio**

R01. Os dados carregados são:

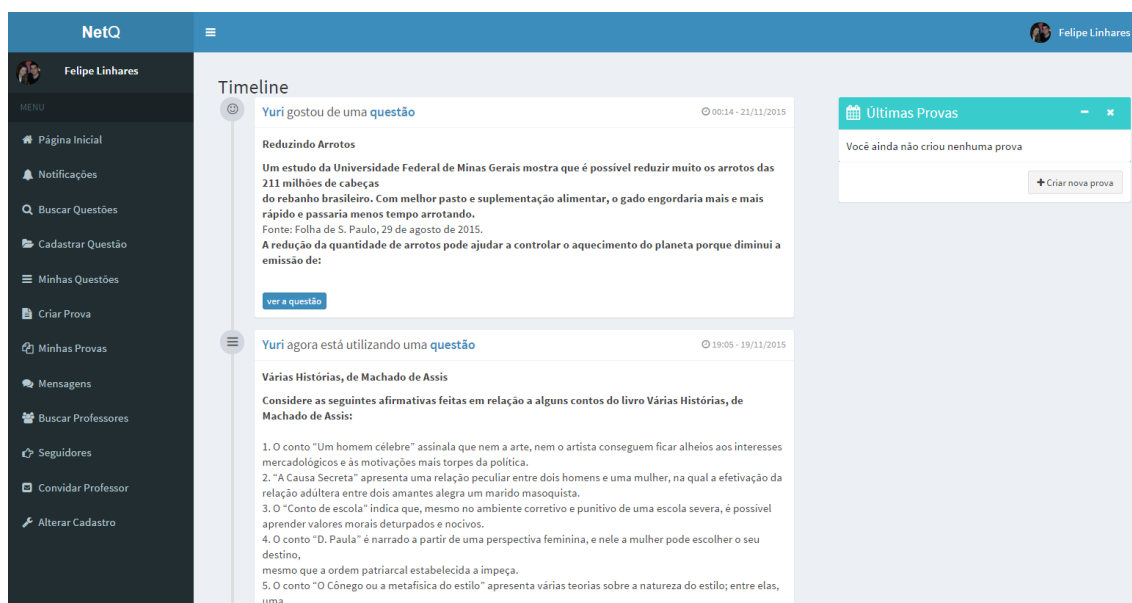
- Dados básicos do professor (foto, nome, formação e contato);
- Quantidade de professores que aquele professor segue;
- Quantidade de professores que seguem aquele professor;
- Ações mais recentes (questões adicionadas, curtidas, seguidas).

R02. Após acionado, o botão “Seguir” fica “pressionado” e seu texto muda para “Seguindo” demonstrar que a ação já aconteceu e o contador de seguidores é atualizado. Caso o usuário acione o botão já pressionado a ação de seguir é desfeita.

## UC05 – Visualizar Timeline

Descrição: Este caso de uso serve para que professor visualize sua timeline que contém as ultimas ações dos professores seguidos e suas ultimas provas.

### Data View: DV05 – Visualizar Timeline.png



### DV05 – Visualizar Timeline.png

### Pré-condições

Este caso de uso pode iniciar somente se:

1. O professor estiver autenticado no sistema

### Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Redirecionar o professor para a página de criação de prova.

### Ator Principal

Professor

### Fluxo de Eventos Principal

1. O sistema carrega os dados.

2. O sistema apresenta a tela; (DV05) (R01)
3. O usuário clica no botão “Criar uma prova”; (A01) (A02) (A03)
4. O sistema redireciona o professor para o a página de criação de provas; (DV09-2)
5. O caso de uso é encerrado.

### **Fluxos Alternativos**

A01: *Link* “questão” é acionado.

1. O sistema redireciona o professor para a página da respectiva questão.

A02: *Link* com o nome de um professor é acionado.

O sistema redireciona o professor para a página do perfil do respectivo professor.

A03: O botão “Ver a questão” é acionado.

1. O sistema redireciona o professor para a página da respectiva questão.

### **Regras de Negócio**

R01. Os dados carregados são:

- Ações recentes dos professores seguidos (questões adicionadas, curtidas, seguidas);
- Últimas provas adicionadas;

## UC06 – Convidar Professor

Descrição: Este caso de uso serve para que um Professor convide um professor a participar da rede social

Data View: DV06 – Convidar Professor.png

The screenshot displays the 'Convidar Professor' (Invite Professor) interface in the NetQ system. The page title is 'Convidar Professor' and the user is identified as 'Felipe Lihares'. The main content area contains a form with the following elements:

- A header: 'Convide professores conhecidos para fazerem parte da sua rede'.
- An 'Email' input field containing 'exemplo@email.com'.
- A 'Mensagem' (Message) text area with the placeholder text 'Adicione uma mensagem ao seu convidado...'. The text area is currently empty.
- Two buttons at the bottom: 'Cancelar' (Cancel) and 'Convidar' (Invite).

The left sidebar menu includes options such as 'Página Inicial', 'Notificações', 'Buscar Questões', 'Cadastrar Questão', 'Minhas Questões', 'Criar Prova', 'Minhas Provas', 'Mensagens', 'Buscar Professores', 'Seguidores', 'Convidar Professor', and 'Alterar Cadastro'. The footer contains the text 'Copyright 2015 Company. All rights reserved.' and 'Anything you want'.

DV06 – Convidar Professor.png

### Pré-condições

Este caso de uso pode iniciar somente se:

1. O Professor estiver autenticado no sistema

### Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Ter enviado o convite para o e-mail inserido pelo Usuário

### Ator Principal

Professor

### Fluxo de Eventos Principal

1. O sistema apresenta a tela; (DV06)
2. O usuário insere o e-mail do Professor que deseja convidar; (A01)
3. O usuário insere uma mensagem opcional;
4. O usuário pressiona o botão “Convidar”; (E01)
5. O sistema envia o convite; (R01)
6. O caso de uso é encerrado.

### **Fluxos Alternativos**

A01: Botão “Cancelar” é pressionado.

1. O sistema redireciona o professor para o UC05 – Visualizar Timeline;
2. O caso de uso é encerrado.

### **Fluxos de Exceção**

E01. Endereço de email inválido:

1. O usuário digitou um endereço de email inválido;
2. O sistema exibe a mensagem “Favor informar um email válido” em vermelho;
3. O caso de uso é reiniciado.

### **Regras de Negócio**

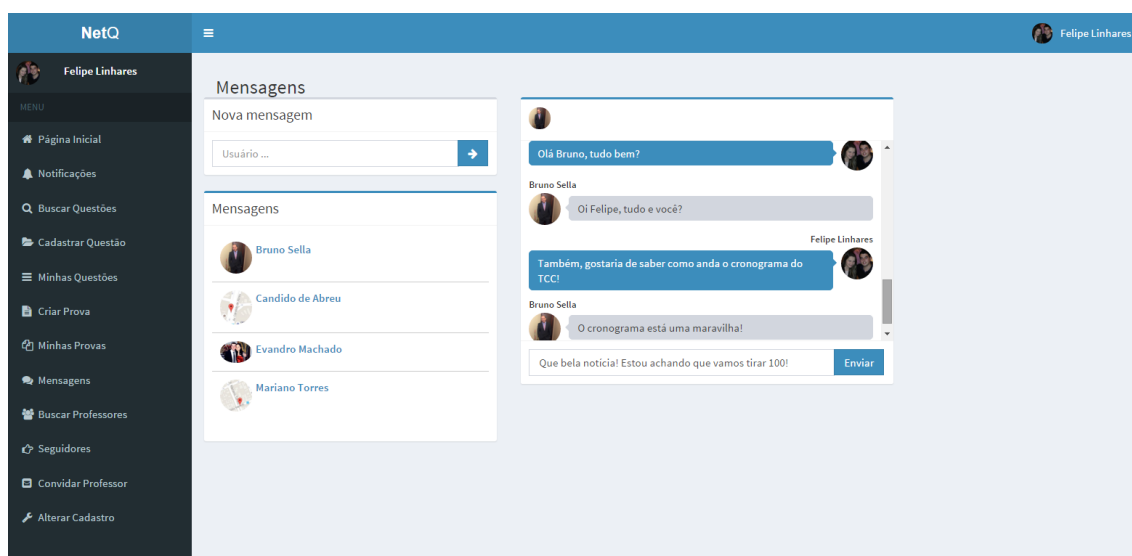
R01. O email enviado deve conter um *link* para a página de cadastro de professores e este link deve conter um código gerado aleatoriamente para que o professor tenha acesso a página de cadastro.



## UC07 – Enviar Mensagem Privada

Descrição: Este caso de uso serve para que um Professor envie uma mensagem a outro professor.

Data View: DV07 – Mensagens.png



DV07 – Mensagens.png

### Pré-condições

Este caso de uso pode iniciar somente se:

1. O professor estiver autenticado no sistema

### Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Ter entregue a mensagem ao professor destinatário

### Ator Principal

Professor.

### Fluxo de Eventos Principal

1. O sistema carrega mensagens recentes;
2. O sistema apresenta a tela; (DV07)
3. O usuário preenche o campo de destinatário; (A01)

4. O usuário pressiona o botão que contém uma seta para prosseguir;
5. O sistema pesquisa e exibe os resultados com o usuário pesquisado; (E01)
6. O usuário seleciona o resultado correspondente;
7. O sistema exibe a caixa de diálogo;
8. O usuário digita a mensagem e pressiona o botão “Enviar”;
9. O sistema envia a mensagem ao destinatário;
10. O caso de uso é encerrado.

### **Fluxos Alternativos**

A01: Destinatário recente selecionado.

1. O usuário seleciona um destinatário da lista recente.
2. O sistema exibe a caixa de diálogo
3. O usuário digita a mensagem e pressiona o botão “Enviar”
4. O sistema envia a mensagem ao destinatário.
5. O caso de uso é encerrado.

### **Fluxos de Exceção**

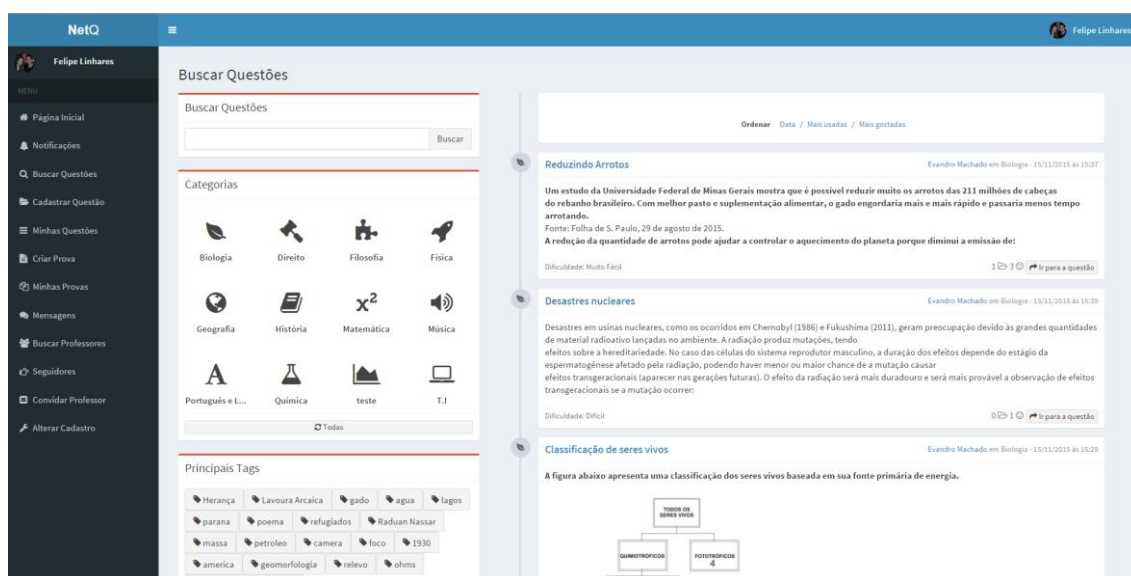
E01. Nenhum destinatário encontrado para a pesquisa

1. O sistema não exibe nenhum contato na lista de contatos.
2. O Use Case é reiniciado.

## UC08 – Buscar Questão

Descrição: Este caso de uso serve para que um usuário busque questões cadastradas.

Data View: DV08 – Buscar Questão.png



DV08 – Buscar Questão.png

### Pré-condições

Este caso de uso pode iniciar somente se:

1. O usuário estiver autenticado no sistema.

### Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Ter redirecionado o usuário para a página da questão selecionada.

### Ator Principal

Professor ou Administrador.

### Fluxo de Eventos Principal

1. O sistema carrega questões recentes;

2. O sistema apresenta a tela; (DV08)
3. O usuário digita as palavras chave de pesquisa e pressiona “Buscar”; (A01)(E01)
3. O sistema exibe os resultados da busca; (E01)
4. O usuário clica no “Ir para a questão”;
5. O sistema direciona para o caso de uso “UC02 – Visualizar Questão”;
6. O caso de uso é encerrado.

### **Fluxos Alternativos**

A01: Categoria selecionada.

1. O usuário clica sobre uma categoria;
2. O sistema exibe as questões da respectiva categoria.

A02: Ordenação ativada

1. O usuário seleciona um tipo de ordenação; (R01)
2. O sistema exibe as questões ordenadas.

### **Fluxos de Exceção**

E01. Nenhum resultado encontrado:

1. O sistema mostra a lista de questões vazia.
2. O Use Case é reiniciado.

### **Regras de Negócio**

R01. As ordenações respeitarão critérios:

1. Por data;
2. Popularidade - Questões mais “gostadas”;
3. Mais utilizadas – Questões mais utilizadas pelos professores em suas listas.

## UC09 – Gerenciar Prova

Descrição: Este caso de uso serve para que um Professor crie , visualize, edite, imprima uma prova e cadastre o desempenho dos alunos por questão.

**Data View:** DV09-1 – Minhas provas.png , DV09-2 – Criar Prova.png, DV09-3 – Escolher questoes.png, DV09-4 – Finalizacao e impressao.png, DV09-5 Prova gerada.png, DV09-6 Cadastro desempenho.png .

NetQ

Evandro Machado

Minhas Provas

Criar nova prova

Titulo	Criada em	Corrigida em	Status	PDF	Editar
prova 2 tteste			Gerada		
prova 1 teste			Em produção		

Copyright 2015 Company. All rights reserved. Anything you want

DV09-1 – Minhas provas.png

NetQ

Felipe Linhares

Criar Prova

**Deixe-me escolher**

Selecione questões a partir da sua lista para utilizar nesta prova

Nome da Prova

Digite um nome...

\* O nome servirá apenas para identificação e não aparecerá na prova

Próximo passo

**Gerar aleatoriamente**

O sistema seleciona automaticamente as questões para você

Número de Questões

Categoria: Biologia

Somente da minha seleção

Qualquer

Próximo passo

DV09-2 – Criar Prova.png

**NetQ** Felipe Linhares

**Criar Prova** Seleção do tipo da Prova > Seleção de questões > Finalização e Impressão

Nome da Prova: Prova Teste

Questões a serem selecionadas (22)

Questões selecionadas (2)

**No fluxo e refluxo da maré** +selecionar

**Unificação alemã** x remover

**Enunciado** O soneto "No fluxo e refluxo da maré encontra o poeta incentivo pra recordar seus males", de Gregório de Matos, apresenta características marcantes do poeta e do período em que ele o escreveu:

Seis horas enche e outras tantas vaza  
A maré pelas margens do Oceano,  
E não larga a tarefa um ponto no ano,  
Depois que o mar rodela, o sol abrasa  
Desde a esfera primeira opaca, ou rasa  
A Lua com impulso soberano  
Engole o mar por um secreto cano,  
E quando o mar vomita, o mundo arrasa.  
Muda-se o tempo, e suas temperanças.  
Até o céu se muda, a terra, os mares,  
E tudo está sujeito a mil mudanças.  
Só eu, que todo o fim de meus pesares  
Eram de algum mirguante as esperanças,  
Nunca o mirguante vi de meus azares.

**Alternativas**

De acordo com o poema, é correto afirmar:

A. A temática barroca do desconcerto do mundo está representada no poema, uma vez que as coisas do mundo estão em desarmonia entre si.  
B. A transitoriedade das coisas terrenas está em oposição ao caráter

**Unificação alemã**

**Enunciado** A unificação alemã foi articulada pelo reino da:

**Alternativas**

A. Prússia, após a derrota da Comuna de Paris na Guerra Franco-Prussiana, apoiado em uma aliança com a aristocracia austríaca e a burguesia prussiana.  
B. Áustria, devido à sua superioridade industrial e militar dentro da Confederação Germânica, apoiado em uma aliança com a aristocracia prussiana.  
C. Áustria, como resposta à ameaça prussiana de unificação após a Instituição do Zollverein na Confederação Germânica, apoiado em uma aliança com a aristocracia austríaca.  
D. Prússia, devido ao seu poderio militar e força econômica dentro da Confederação Germânica, apoiado em uma aliança entre a aristocracia e a alta burguesia.  
E. Prússia, devido à mobilização nacionalista da Confederação Germânica durante a Guerra Franco-Prussiana, apoiado em uma aliança com a grande burguesia austríaca.

**Categoria** História  
**Dificuldade** Difícil  
**Acertos** 50%

**Media de idades** x remover

DV09-3 – Escolher questoes.png

**NetQ** Felipe Linhares

**Criar Prova** Seleção do tipo da Prova > Seleção de questões > Finalização e Impressão

Embaralhar ordem das Questões?  
Embaralhar ordem das Alternativas?

**Gerar Prova**

**Cabeçalho**

Normal B I [Rich Text Editor Icons] Código-Fonte

**UNIVERSIDADE FEDERAL DO PARANÁ**

Aluno: \_\_\_\_\_ Matrícula: \_\_\_\_\_  
Curso: \_\_\_\_\_ Professor: \_\_\_\_\_  
Disciplina: \_\_\_\_\_ Data: \_\_\_\_/\_\_\_\_/\_\_\_\_ Valor: \_\_\_\_\_

**body p**

**Questões**

1 - A unificação alemã foi articulada pelo reino da:

A. Prússia, após a derrota da Comuna de Paris na Guerra Franco-Prussiana, apoiado em uma aliança com a aristocracia austríaca e a burguesia prussiana.  
B. Áustria, devido à sua superioridade industrial e militar dentro da Confederação Germânica, apoiado em uma aliança com a aristocracia prussiana.  
C. Áustria, como resposta à ameaça prussiana de unificação após a Instituição do Zollverein na Confederação Germânica, apoiado em uma aliança com a aristocracia austríaca.  
D. Prússia, devido ao seu poderio militar e força econômica dentro da Confederação Germânica, apoiado em uma aliança entre a aristocracia e a alta burguesia.  
E. Prússia, devido à mobilização nacionalista da Confederação Germânica durante a Guerra Franco-Prussiana, apoiado em uma aliança com a grande burguesia austríaca.

2 - Em um grupo de 6 pessoas, a média das idades é 17 anos, a mediana é 16,5 anos e a moda é 16 anos. Se uma pessoa de 24 anos se juntar ao grupo, a média e a mediana das idades do grupo passarão a ser, respectivamente:

A. 17 anos e 17 anos.  
B. 18 anos e 17 anos.  
C. 18 anos e 16,5 anos.  
D. 20,5 anos e 16,5 anos.  
E. 20,5 anos e 20,25 anos.

DV09-4 – Finalizacao e impressao.png



### **Pré-condições**

Este caso de uso pode iniciar somente se:

1. O professor possuir cadastro no sistema.

### **Pós-condições**

Após o fim normal deste caso de uso o sistema deve:

1. Cadastrar a prova no sistema.

### **Ator Principal**

Professor.

### **Fluxo de Eventos Principal**

1. O sistema apresenta a tela que lista todas as provas. (DV09-1)
2. O professor aciona o botão “Criar nova prova”; (A01) (A02) (A04)
3. O sistema apresenta a tela de criação de prova; (DV09-2)
4. O professor escolher “Deixe-me escolher” como tipo de prova e preenche o campo de nome; (R01)(A03)
5. O professor aciona o botão “Próximo passo”;
6. O sistema redireciona o professor à página de escolha de questões; (DV09-3)
7. O professor seleciona as questões que ele quer na prova e aciona o botão “Próximo passo”;
8. O sistema redireciona o professor à página de finalização e impressão; (DV09-4)
9. O professor edita o cabeçalho da prova se julgar necessário;
10. O professor escolhe se vai ou não utilizar as opções de embaralhamento de questões ou alternativas; (R02)
11. O professor aciona o botão “Gerar prova”;
12. O sistema gera a prova e redireciona o professor para a página de listagem de provas; (DV09-1)
13. O caso de uso é encerrado.

### **Fluxos Alternativos**



A01: O professor aciona o botão de visualização de uma prova já gerada.

1. O sistema redireciona o professor para a tela de visualização de prova; (DV09-5)
2. O professor aciona o botão “Imprimir”;
3. O sistema manda a prova para impressão;
4. O caso de uso é encerrado.

A02: O professor aciona o botão de edição de uma prova existente.

1. O sistema redireciona o professor para a tela de escolha de questões; (DV09-3)
2. O professor adiciona ou remove as questões desejadas e aciona o botão “Próximo passo”;
3. O sistema redireciona o professor à página de finalização e impressão; (DV09-4)
4. O professor edita o cabeçalho da prova se julgar necessário;
5. O professor escolhe se vai ou não utilizar as opções de embaralhamento de questões ou alternativas; (R02)
6. O professor aciona o botão “Gerar prova”;
7. O sistema gera a prova e redireciona o professor para a página de listagem de provas; (DV09-1)
8. O caso de uso é encerrado.

A03: O professor opta por uma prova do tipo “Gerar aleatória”:

1. O professor preenche o número de questões desejado;
2. O professor escolhe a categoria a qual as questões deverão pertencer;
3. O professor escolhe se dejesa apenas questões em sua lista ou todas as questões do sistema para compor esta prova;
4. O professor aciona o botão “Próximo passo”;
5. O sistema redireciona o professor para a página de finalização e impressão, já exibindo as questões que foram aleatoriamente escolhidas baseando-se nos critérios informados; (DV09-4)
6. O professor edita o cabeçalho da prova se julgar necessário;
7. O professor escolhe se vai ou não utilizar as opções de embaralhamento de questões ou alternativas; (R02)
8. O professor aciona o botão “Gerar prova”;
9. O sistema gera a prova e redireciona o professor para a página de listagem de provas; (DV09-1)
10. O caso de uso é encerrado.

A04: O professor pressiona o botão de cadastro de desempenho:

1. O sistema redireciona o professor à página de cadastro de desempenho; (DV09-6)

2. O professor informa quantos alunos fizeram a prova e quantos acertaram cada questão;
3. O professor aciona o botão “Cadastrar desempenho”;
4. O sistema guarda os dados de desempenho no banco de dados;
5. O caso de uso é encerrado.

### **Fluxos de Exceção**

E01. Dados obrigatórios não preenchidos:

1. O sistema retorna a mensagem “Favor preencher os campos marcados como preenchimento obrigatório (\*)”.
2. O professor aciona o botão “Ok”.
3. O sistema retorna à pagina de criação de prova (DV09-2).

### **Regras de Negócio**

R01. Os tipos de prova são:

1. Deixe-me escolher: Neste tipo de geração de prova o professor escolhe manualmente uma a uma todas as questões que vão compor sua prova.
2. Gerar Aleatoriamente: Neste tipo de geração de prova o professor escolhe quantas questões ele quer na prova, qual a categoria das questões a serem selecionadas e se as questões serão somente as de sua lista ou todas do sistema. Após tal seleção o sistema gera a prova com questões aleatórias e alternativas embaralhadas.

R02. A opção de embaralhar questões garante que a ordem das questões da prova será embaralhada. A opção de embaralhar alternativas garante que as alternativas de todas as questões serão embaralhadas.

## UC10 – Gerar Relatório de questões mais difíceis

Descrição: Este caso de uso serve para que um Professor gere um relatório das questões mais difíceis.

Data View: DV10 – Geração de relatório.png

#	Título	Categoria	Quantidade de Aplicações em prova	Acertos	Erros	A/E
5432	O seguinte código java irá compilar?	T.I.	93	231	988	0,23
1896	Sobrecarga em loop com uso de foreach	T.I.	82	284	915	0,31
3354	HTML onclick	T.I.	73	319	841	0,38
1793	Buscando dados em várias tabelas num único Select	T.I.	69	339	739	0,46
1099	Converter número em float para string PHP	T.I.	57	320	655	0,49
3399	Imprimir os valores do meu Array	T.I.	41	322	587	0,55
3585	Converter formato de data	T.I.	39	312	496	0,63
1912	Session Java	T.I.	21	228	331	0,69
489	trigger de inserção	T.I.	18	142	200	0,71
775	lista com push AngularJS	T.I.	17	77	107	0,72

DV10 – Geração de relatório.png

### Pré-condições

Este caso de uso pode iniciar somente se:

1. O professor possuir cadastro no sistema.
2. O professor tiver cadastrado o desempenho dos alunos de pelo menos uma prova.

### Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Gerar relatório de questões mais difíceis.

**Ator Principal**

Professor.

**Fluxo de Eventos Principal**

1. O sistema carrega os dados das questões mais difíceis; (R01) (E01)
2. O sistema apresenta a tela; (DV10)
3. O caso de uso é encerrado.

**Fluxos de Exceção**

E01. Nenhum desempenho cadastrado:

1. O sistema retorna a mensagem “Você ainda não cadastrou o desempenho de seus alunos em nenhuma prova!”;
2. O caso de uso é reiniciado.

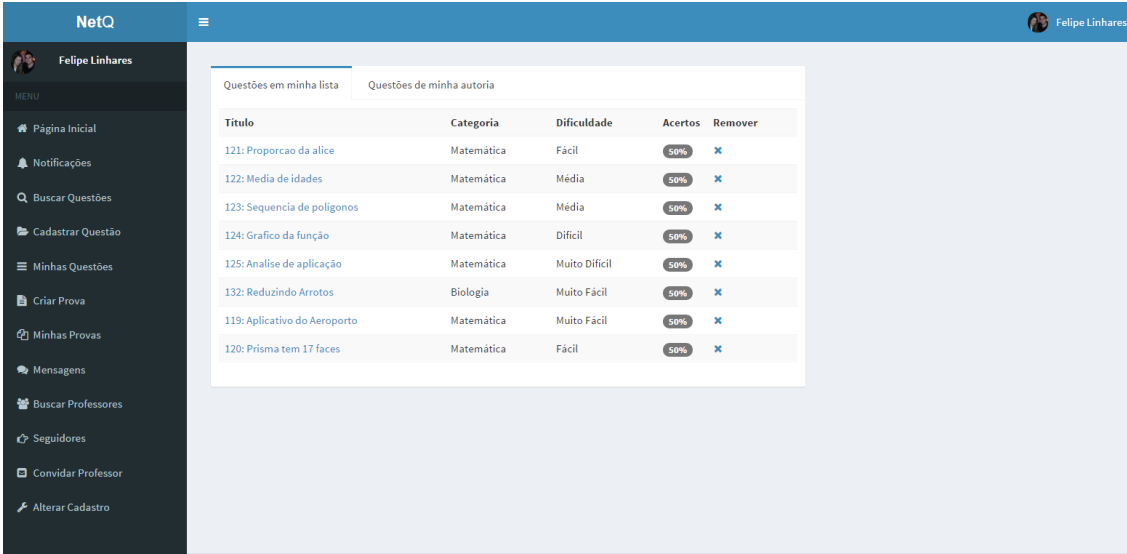
**Regras de Negócio**

R01. O sistema considera mais difíceis as questões que são erradas pela maior porcentagem de alunos quando o professor cadastra o desempenho nas provas.

## UC11 – Gerenciar Questões

Descrição: Este caso de uso serve para que um Professor gerencie suas questões.

**Data View:** DV11-1 – Listar questões.png, DV11-2 – Editar questão.png, DV11-3 – Cadastrar questão.png



The screenshot displays the NetQ application interface. On the left is a dark sidebar menu with the user's name 'Felipe Linhares' and a list of navigation options: Página Inicial, Notificações, Buscar Questões, Cadastrar Questão, Minhas Questões, Criar Prova, Minhas Provas, Mensagens, Buscar Professores, Seguidores, Convidar Professor, and Alterar Cadastro. The main content area is titled 'Questões em minha lista' and 'Questões de minha autoria'. It features a table with the following data:

Título	Categoria	Dificuldade	Acertos	Remover
121: Proporcao da alice	Matemática	Fácil	50%	X
122: Media de idades	Matemática	Média	50%	X
123: Sequencia de poligonos	Matemática	Média	50%	X
124: Grafico da função	Matemática	Difícil	50%	X
125: Analise de aplicação	Matemática	Muito Difícil	50%	X
132: Reduzindo Arrotos	Biologia	Muito Fácil	50%	X
119: Aplicativo do Aeroporto	Matemática	Muito Fácil	50%	X
120: Prisma tem 17 faces	Matemática	Fácil	50%	X

At the bottom of the interface, there is a copyright notice: 'Copyright 2015 Company. All rights reserved.' and a placeholder text: 'Anything you want'.

DV11-1 – Listar questões.png

**NetQ** Felipe Linhares

### Editar Questão

**Título**  
No fluxo e refluxo da maré

**Enunciado**

O soneto "No fluxo e refluxo da maré encontra o poeta incerto pra recordar seus males", de Gregório de Matos, apresenta características marcantes do poeta e do período em que ele o escreveu:

Seis horas enche e outras tantas vaza  
A maré pelas margens do Oceano,  
E não larga a tarefa um ponto no ano,  
Depois que o mar rodeia, o sol abraça,  
Desde a esfera primeira opaca, ou rasa  
A Lua com impulso soberano

**Categoria**: Português e Literatura **Dificuldade**: Difícil **Privacidade**: Pública **Tags**: poema, Gregório de Matos

Discursiva | **Múltipla Escolha / Verdadeiro ou Falso**

**Atenção!**  
Ao assinalar mais de uma alternativa como correta, a questão será automaticamente do tipo "Verdadeira ou Falso".

**Alternativas**

A temática barroca do desconcerto do mundo está representada no poema, uma vez que as coisas do mundo estão em desarmonia entre si.  Correta  Incorreta

A transitoriedade das coisas terrenas está em oposição ao caráter imutável do sujeito, submetido a uma concepção fatalista do destino humano.  Correta  Incorreta

A concepção de um mundo às avessas está figurada no soneto através da clara oposição entre o mar que tudo move e a lua imutável.  Correta  Incorreta

A clareza empregada para exposição do tema reforça o ideal de simplicidade e bucolismo da poesia barroca, cujo lema fundamental era a aurea mediocritas.  Correta  Incorreta

A sintonia entre a natureza e o eu-poético embasa as personificações de objetos inanimados aliadas às hipérboles que descrevem o sujeito.  Correta  Incorreta

**Atualizar**

DV11-2 – Editar questão.png

**NetQ** Felipe Linhares

### Criar Questão

**Título**  
Título

**Enunciado**

**Categoria**: Biologia **Dificuldade**: Muito Fácil **Privacidade**: Pública **Tags**

Discursiva | **Múltipla Escolha / Verdadeiro ou Falso**

**Resposta**  
Resposta

**Cadastrar**

DV11-3 – Cadastrar questão.png

### **Pré-condições**

Este caso de uso pode iniciar somente se:

1. O usuário possuir cadastro no sistema.

### **Pós-condições**

Após o fim normal deste caso de uso o sistema deve:

1. Atualizar a questão no sistema.

### **Ator Principal**

Professor.

### **Fluxo de Eventos Principal**

1. O sistema carrega os dados de questões próprias e utilizadas; (R01)
2. O sistema apresenta a tela; (DV11-1)
3. O professor aciona a aba “Questões de minha autoria”; (A01) (A02) (A03)
4. O sistema exibe as questões próprias;
5. O professor escolhe uma questão própria e aciona o botão de edição; (A04)
6. O sistema redireciona o professor para a página de edição de questão com os dados pré-carregados; (DV11-2) (R02)
7. O professor altera os dados desejados da questão;
8. O professor aciona o botão “Atualizar”; (E01)
9. O sistema armazena as atualizações dos dados da questão;
10. O sistema redireciona o professor para o UC02 – Visualizar Questão para que o professor visualize sua questão alterada.

### **Fluxos Alternativos**

A01: O professor clica no título de alguma questão listada.

1. O sistema redireciona o professor para o UC02 – Visualizar Questão da respectiva questão;
2. O caso de uso é encerrado.

A02: O professor aciona o botão “X” para remover uma questão de sua lista.

1. O sistema remove a questão da lista;
2. O caso de uso é encerrado.

A03: O professor aciona o botão “Cadastrar Questão” do menu lateral.

1. O sistema redireciona o professor para a tela de cadastro de questão; (DV11-3)
2. O professor digita os dados da questão;
3. O professor aciona o botão “Cadastrar”; (E01)
4. O sistema cadastra a questão no banco de dados;
5. O sistema redireciona o professor para a UC02 – Visualizar Questão da questão recém criada.
6. O caso de uso é encerrado.

A04: O professor aciona o botão “X” para remover uma questão.

1. O sistema remove a questão;
2. O caso de uso é encerrado.

### **Fluxos de Exceção**

E01. Dados obrigatórios não preenchidos:

1. O sistema exibe uma mensagem adequada em vermelho junto ao campo que precisa ser preenchido.
2. O professor preenche o necessário e aciona o botão novamente.
3. O sistema prossegue com a requisição.

### **Regras de Negócio**

R01. A aba “Questões em minha lista” deve ser populada com as questões que não são de autoria própria mas foram adicionadas à lista por meio do botão “Utilizar Questão” encontrado em demais páginas. A aba “Questões de minha autoria” deve ser populada com as questões adicionadas pelo usuário.

R02. Nas páginas de adicionar ou editar questão a resposta pode ser digitada na primeira aba caso seja uma questão discursiva. Caso a questão seja objetiva ou “V ou F”, as alternativas devem ser preenchidas utilizando a segunda aba.



R03. Caso haja apenas uma alternativa marcada como correta, a questão será considerada objetiva. Caso haja zero ou mais do que uma alternativa marcada como correta, a questão será considerada do tipo “V ou F”.

## **UC12 – Gerenciar Cadastro**

Descrição: Este caso de uso serve para que um Professor se cadastre, altere ou apague seu cadastro.

**Data View:** DV12-1 – Cadastrar Professor.png, DV12-2 – Alterar cadastro.png

**NetQ**

### Cadastrar Professor

**Foto**  Nenhum arquivo selecionado  
Formato .jpg ou .jpeg de tamanho máximo 1MB

**Nome**

**Email**

**Senha**

**Data de Nascimento**

**Formação**

**Instituição**


**Site Pessoal**

**LinkedIn**

DV12-1 – Cadastrar Professor.png

**NetQ** Felipe Linhares

### Alterar Cadastro

**Foto**   Nenhum arquivo selecionado  
Formato .jpg ou .jpeg de tamanho máximo 1MB

**Nome**

**Email**

**Senha**

**Data de Nascimento**

**Formação**

**Instituição**

**Site Pessoal**

**LinkedIn**

DV12-2 – Alterar cadastro.png

## Pré-condições

Este caso de uso pode iniciar somente se:

1. O professor estiver acessando a página de novo cadastro utilizando um link válido proveniente de um convite efetuado por um professor já cadastrado;

2. O professor estiver autenticado no sistema e quiser atualizar ou apagar seu cadastro.

### **Pós-condições**

Após o fim normal deste caso de uso o sistema deve:

1. Cadastrar ou atualizar as informações do professor no banco de dados.

### **Ator Principal**

Professor.

### **Fluxo de Eventos Principal**

1. O usuário aciona o *link* que lhe foi enviado por email pelo UC06 – Convidar Professor; (A01)
2. O sistema valida o código do *link*; (E01)
3. O sistema apresenta a página de cadastro; (DV12-1)
4. O usuário preenche suas informações; (R01)
5. O usuário aciona o botão “Cadastrar”; (E02)
6. O sistema cadastra os dados do novo professor na base de dados;
7. O sistema apaga o código do convite para que não seja mais possível acessar a página de cadastro com aquele código.
8. O caso de uso é encerrado.

### **Fluxos Alternativos**

A01: O professor já é cadastrado no sistema e aciona o botão “Alterar Cadastro” do menu lateral.

1. O sistema busca os dados do professor;
2. O sistema exibe a página de alteração de cadastro; (DV12-2)
3. O usuário preenche suas informações; (R01) (A02)
4. O usuário aciona o botão “Atualizar”; (E02)
5. O sistema altera os dados do professor na base de dados;
6. O caso de uso é encerrado.

A02: O professor aciona o botão “Apagar perfil”.

1. O sistema inativa o perfil em questão;

2. O professor é redirecionado para a página de login UC01 – Efetuar Login.

### **Fluxos de Exceção**

E01. Código inválido:

1. O sistema não encontra o código do *link* no banco de dados;
2. O sistema redireciona o usuário para o UC01 – Efetuar Login.
3. O caso de uso é encerrado.

E03. Dados inválidos:

1. O usuário deixa de preencher um campo obrigatório ou o preenche com dados inválidos;
2. O sistema exibe uma mensagem em vermelho junto ao campo explicando qual o problema.
3. O caso de uso é reiniciado.

### **Regras de Negócio**

R01. O professor deve subir uma foto por meio do botão “Escolher arquivo”. A foto deve estar no formato .jpg ou .jpeg e deve pesar no máximo 1MB.

## UC13 – Adicionar Categoria

Descrição: Este caso de uso serve para que o Administrador cadastre uma nova categoria.

Data View: DV13 – Adicionar categoria.png

The screenshot displays the 'Cadastrar Categoria' (Register Category) interface in the NetQ system. The interface includes a dark sidebar with navigation options: 'Relatórios', 'Denúncias', and 'Categorias'. The main content area features a form with a 'Titulo' (Title) input field and an 'Imagem' (Image) field currently showing 'Nenhuma imagem selecionada' (No image selected). A blue 'Cadastrar' (Register) button is positioned to the right of the form. Below the form, a section titled 'Selecione uma imagem' (Select an image) presents a grid of 48 icons for selection, arranged in 8 rows and 6 columns. The icons represent various categories such as location, anchor, document, building, plus, minus, person, asterisk, at-sign, scales, no, bar chart, list, menu, battery, notification, and others.

### Pré-condições

Este caso de uso pode iniciar somente se:

1. O administrador estiver autenticado no sistema.

### Ator Principal

Administrador.

### Fluxo de Eventos Principal

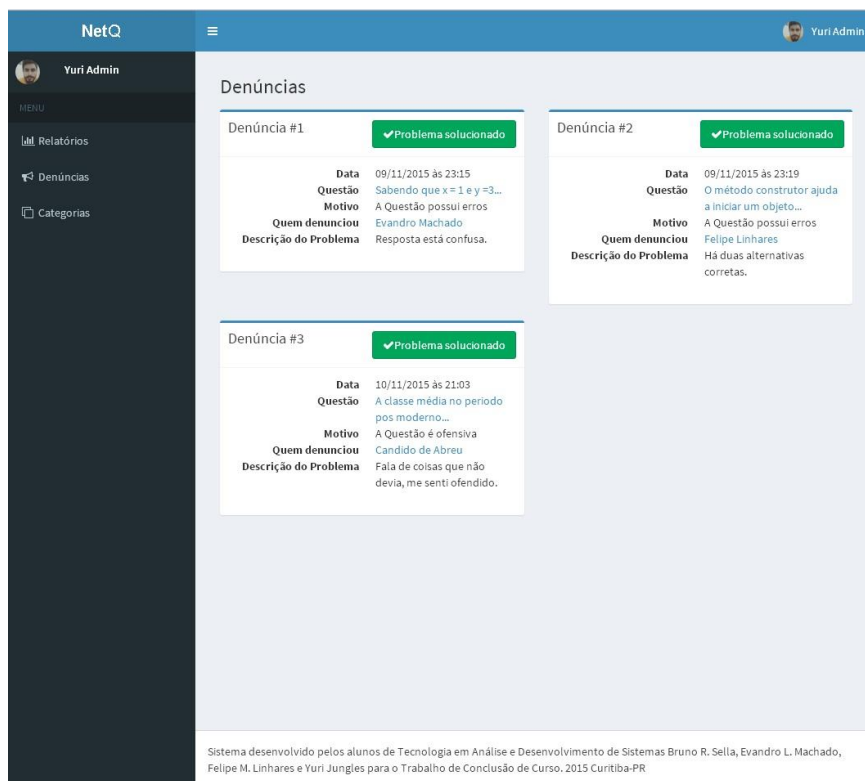
1. O sistema carrega as possíveis imagens à serem utilizadas.
2. O sistema apresenta a tela; (DV13)

3. O administrador preenche o campo “Título” da categoria e escolhe uma imagem a ser utilizada;
4. O administrador aciona o botão “Cadastrar”;
5. O sistema cadastra a nova categoria e a partir de agora ela estará disponível para os professores;
6. O caso de uso é encerrado.

## UC14 – Gerenciar Denuncias

Descrição: Este caso de uso serve para que o Administrador possa gerenciar as denúncias realizadas pelos usuários.

Data View: DV14 – Gerenciar Denuncias.png



DV 14 - Gerenciar Denuncias

### Pré-condições

Este caso de uso pode iniciar somente se:

1. O administrador estiver autenticado no sistema.

### Ator Principal

Administrador.

### Fluxo de Eventos Principal

1. O sistema carrega as denúncias realizadas pelos usuários e as lista; (E01)
2. O sistema apresenta a tela; (DV14)
3. O administrador escolhe uma denúncia e entra em contato com o autor da questão para que o problema seja solucionado;

4. O professor autor da questão denunciada edita a questão e avisa o administrador;
4. O administrador verifica que a questão foi arrumada e aciona o botão “Problema Solucionado” da respectiva denúncia;
5. O sistema deleta a denúncia;
6. O Sistema carrega as denúncias não verificadas;
7. O caso de uso é encerrado.

### **Fluxos de Exceção**

E01. Nenhuma denuncia é encontrada:

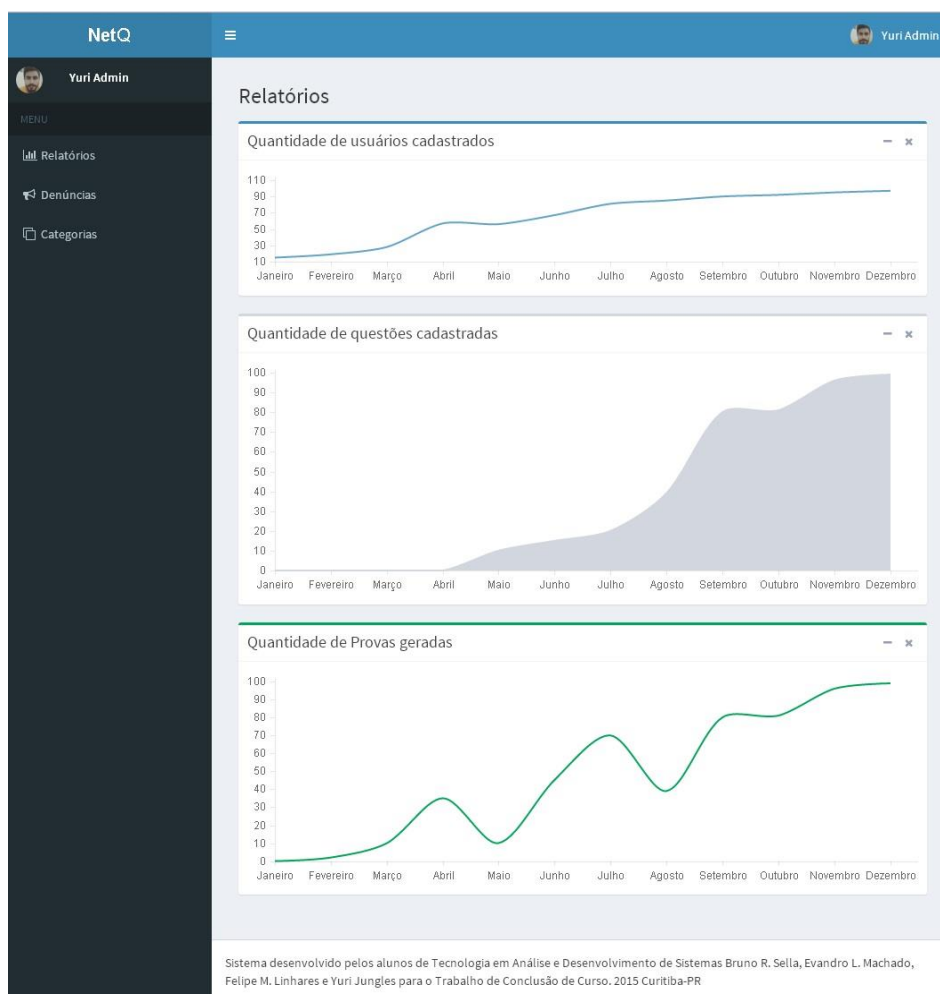
1. O sistema retorna a mensagem “Não há denúncias”;
2. O caso de uso é reiniciado.



## UC15 – Visualizar Relatórios

Descrição: Este caso de uso serve para que um Administrador visualize os relatórios gerados pelo sistema.

Data View: DV15 – Visualizar Relatórios.png



### Pré-condições

Este caso de uso pode iniciar somente se:

1. O administrador estiver autenticado no sistema.

### Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Ter exibido ao administrador os relatórios.

### **Ator Principal**

Administrador

### **Fluxo de Eventos Principal**

1. O sistema carrega os dados dos relatórios; (E01) (R01)
2. O sistema exibe a página com os relatórios; (DV15)
3. O caso de uso é encerrado.

### **Fluxos de Exceção**

E01. Não há dados:

1. O sistema não exibe a mensagem “Não há dados para o relatório”;
2. O caso de uso é encerrado.

### **Regras de Negócio**

R01. Os relatórios serão todos exibidos em forma de gráficos e são eles:

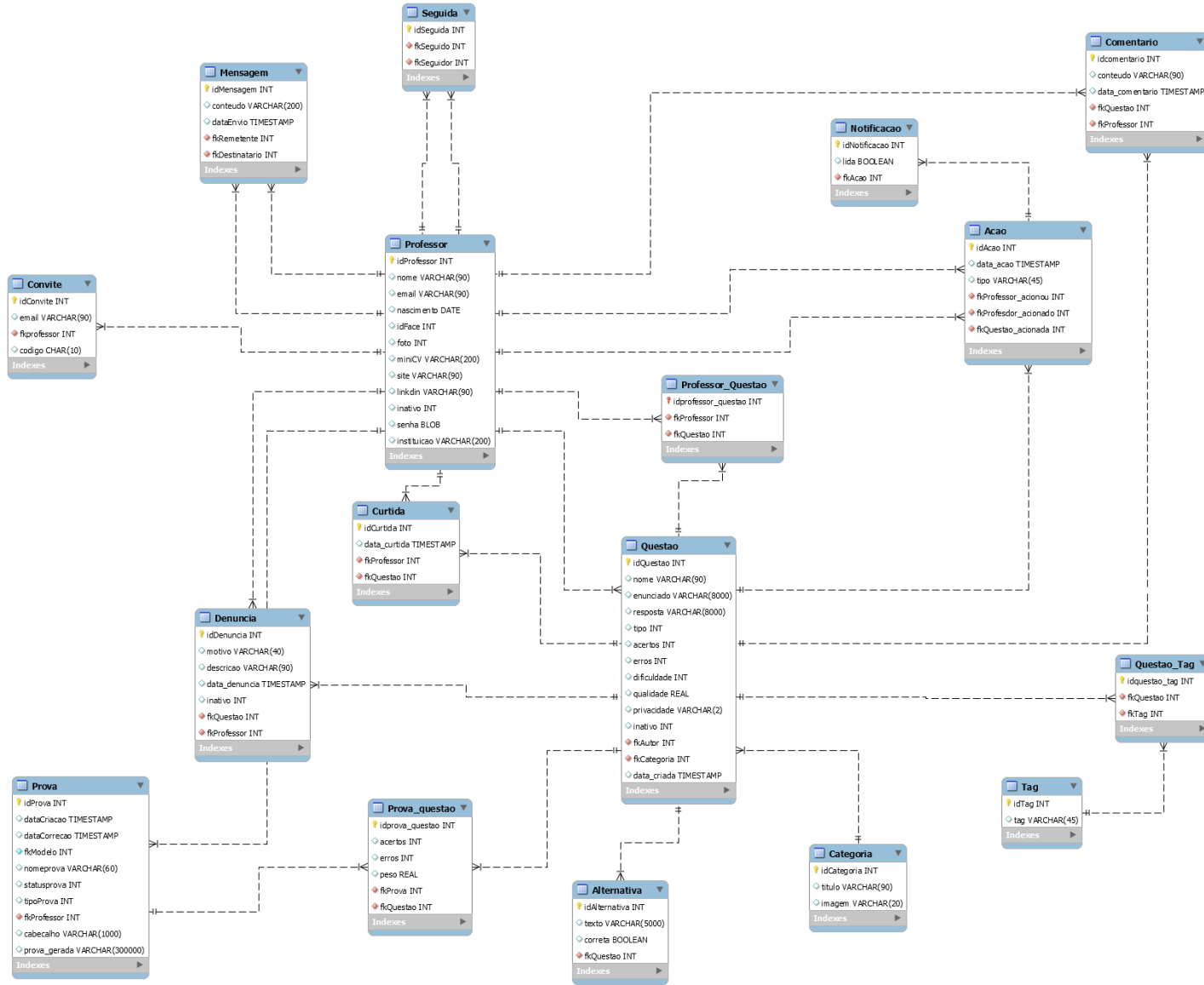
- Quantidade de usuários cadastrados: Informa a quantidade de novos cadastros efetuados por mês;
- Quantidade de questões cadastradas: Informa a quantidade de novas questões adicionadas por mês;
- Quantidade de provas geradas: Informa a quantidade de novas provas geradas por mês.



## APÊNDICE D – PLANO DE RISCO

PLANO DE RISCO								
ID	Condição	Data Limite	Consequencia	Ação	Monitoramento	Probabilidade	Impacto	Classificação
1	Perda de integrante	Sem Limite	Sobrecarga de trabalho e atraso no projeto	Escolher colegas que já participaram em outros trabalhos junto, manter sempre a verdade e sinceridade como base da relação profissional	Toda Equipe	Baixíssima	Altíssimo	4
2	Falta de conhecimento dos processos adotados	19/09/15	Atraso do projeto	Realizar acompanhamento semanal e seguir cronograma previamente determinado	Toda Equipe	Pequena	Alto	4
3	Falta de conhecimento da Tecnologia empregada	01/10/15	Atraso do projeto	Estudar as tecnologias que serão utilizadas no desenvolvimento antes da data destinada ao início da programação	Toda Equipe	Moderada	Alto	6
4	Problema de comunicação	Sem data	Retrabalho	Utilizar plano de comunicação, documentação sólida e ferramentas para gerenciamento das atividades.	Toda Equipe	Baixo	Baixo	4
5	Variação dos requisitos	19/09/15	Retrabalho e desvio do objetivo	Tomar como base os requisitos validados, verificar real necessidade de adição e modificação	Toda Equipe	Moderada	Moderado	5

# APÊNDICE E – DIAGRAMA ENTIDADE RELACIONAMENTO



## APÊNDICE F – WBS

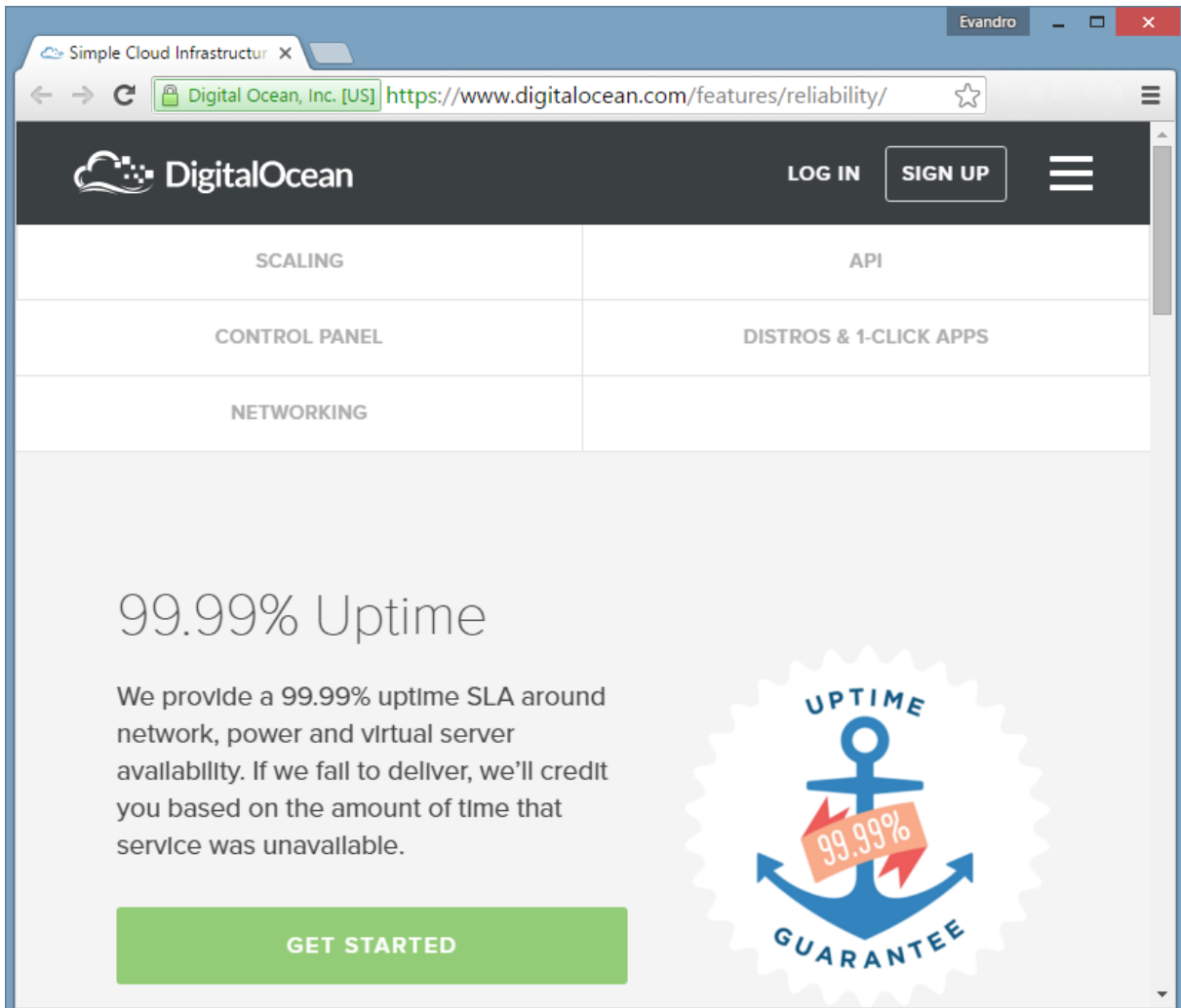
## APÊNDICE G – PLANO DE COMUNICAÇÃO

Grupo de interessados	Foco	O que este grupo precisa saber?	Método	Quando?
<b>Informações internas ao projeto</b>				
Equipe do projeto	Informações sobre o andamento do projeto	Andamento das atividades, dificuldades, impedimentos	Reunião	Diariamente
Equipe do projeto	Auxílio em atividades	Compartilhar conhecimentos para o desenvolvimento de atividades muito grandes ou difíceis	Reunião	Quando necessário
Equipe do projeto	Repasse de informações dos usuários	Compartilhar informações recebidas de futuros usuários do sistema	Reunião ou conversa em grupo no facebook	Quando necessário
<b>INFORMAÇÕES EXTERNAS AO PROJETO</b>				
Equipe do projeto e professor orientador	Informações sobre o andamento do projeto	Andamento do projeto	Reunião	Semanalmente

## APÊNDICE H – DIAGRAMAS DE SEQUÊNCIA



## APÊNDICE I – ACORDO DE NÍVEL DE SERVIÇO



The screenshot shows a web browser window with the URL <https://www.digitalocean.com/features/reliability/>. The page features the DigitalOcean logo and navigation links for "LOG IN" and "SIGN UP". A menu of features is displayed in a grid:

SCALING	API
CONTROL PANEL	DISTROS & 1-CLICK APPS
NETWORKING	

The main content area highlights a "99.99% Uptime" guarantee. The text states: "We provide a 99.99% uptime SLA around network, power and virtual server availability. If we fail to deliver, we'll credit you based on the amount of time that service was unavailable." A green "GET STARTED" button is positioned below the text. To the right, a circular seal features a blue anchor with a red banner across it displaying "99.99%". The words "UPTIME" and "GUARANTEE" are written around the seal's perimeter.

# APÊNDICE J – DIAGRAMA DE CLASSES DE PROJETO

