

Josafá Martins dos Santos

Módulo *Scrum* para um sistema de gestão de projetos

Picos - PI

4 de Junho de 2017

FICHA CATALOGRÁFICA
Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo

S237m Santos, Josafá Martins dos.

Módulo *Scrum* para um sistema de gestão de projetos /
Josafá Martins dos Santos.– 2017.

CD-ROM : il.; 4 ¾ pol. (54 f.)

Trabalho de Conclusão de Curso (Curso Bacharelado em
Sistemas de Informação) – Universidade Federal do Piauí,
Picos, 2017.

Orientador(A): Prof. Esp. Dennis Sávio Martins da Silva

- *Scrum*. 2. Gerenciamento de Projetos. 3. Metodologias Ágeis. I. Título.

CDD 005.133

MÓDULO SCRUM PARA UM SISTEMA DE GESTÃO DE PROJETOS

JOSAFÁ MARTINS DOS SANTOS

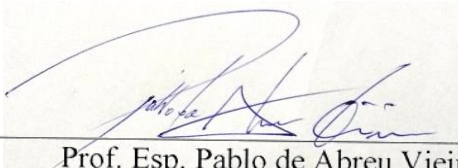
Monografia aprovada como exigência parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Data de Aprovação

Picos – PI, 20 de junho de 2017

Dennis Sávio Martins da Silva

Prof. Esp. Dennis Sávio Martins da Silva
Orientador



Prof. Esp. Pablo de Abreu Vieira
Membro

Leonardo Pereira de Sousa

Prof. Esp. Leonardo Pereira de Sousa
Membro

Josafá Martins dos Santos

Módulo *Scrum* para um sistema de gestão de projetos

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação, Campus Senador Helvídio Nunes de Barros, da Universidade Federal do Piauí como parte dos requisitos para obtenção do Grau de Bacharelado em Sistemas de Informação.

Orientador: Prof. Esp. Dennis Sávio Martins da Silva.

Universidade Federal do Piauí
Campus Senador Helvídio Nunes de Barros
Bacharelado em Sistemas de Informação

Picos - PI
4 de Junho de 2017

Agradecimentos

É difícil citar todas as pessoas que tiveram participação importante para essa conquista, mas quero aqui citar alguns poucos nomes. Primeiramente quero agradecer a Deus por mais essa conquista, e também a minha mãe (Maria da Conceição Martins dos Santos), aos meus irmãos Wandson, Josué e Moisés, e dizer que sem eles nada disso seria possível, todo o apoio emocional que me forneceram, todo carinho, foi de uma importância ímpar, e sem sombras de dúvida foi o que me deu estrutura para aguentar todas as intempéries que houveram durante essa longa caminhada.

Quero também deixar os meus mais sinceros agradecimentos aos meus amigos, em especial Jade, Guilherme, Junior, Marielson, Jailson, Kelly, Domingas, Rúbia, Amanda, João Carlos dentre outros que eu não citei que fizeram com que essa caminhada fosse bem mais prazerosa, a todos os professores, e em especial ao meu orientador professor Dennis Sávio Martins da Silva que esteve sempre disposto a me ajudar e incentivar.

E por último mas não menos importante a dona Purí, pelo melhor café na universidade.

Carl Sagan

Somos feitos de poeira de estrelas.

Resumo

Gerenciamento de projetos é uma das atividades de maior relevância dentro do processo de desenvolvimento de novos ou na manutenção de antigos projetos, um bom gerenciamento não garante o sucesso de um produto, porém o mau gerenciamento normalmente resulta em falhas no projeto: elevação do custo estimado, atraso na entrega e falha no atendimento de um ou mais requisitos do projeto. Para reduzir falhas de gerenciamento normalmente a liderança do projeto pode fazer a adoção de alguma metodologia de desenvolvimento, este trabalho apresenta um estudo de metodologias de desenvolvimento de projetos, bem como o apoio desse desenvolvimento através de ferramentas de gestão, propondo um sistema de gestão de projetos que se utilize da abordagem do método *Scrum*, bem como a apresentação dos resultados positivos que foram obtidos através dos testes executados com a ferramenta SGP (Sistema de gestão de projeto) que foi desenvolvida neste trabalho.

Palavras-chaves: Scrum, Gerenciamento de projetos, Metodologias ágeis.

Abstract

Project management is one of the most important activities within the new development process or in the maintenance of old projects, good management does not guarantee the success of a product, however, poor management usually results in project failures: Delays in delivery, and failure to meet one or more project requirements. To reduce management failures, normally the project leadership can adopt some development methodology, this work presents a study of project development methodologies, as well as the support of this development through management tools, proposing a management system of Projects using the *Scrum* method approach, as well as checking whether it offered any gain or improvement compared to tools already on the market.

Lista de ilustrações

Figura 1 – Como escolher a melhor abordagem.	15
Figura 2 – Visão geral do ciclo de atividades do método <i>Scrum</i>	22
Figura 3 – Padrão de arquitetura MVC	23
Figura 4 – Diagrama entidade relacional do sistema SGP	27
Figura 5 – Tela de funcionalidades administrativas	28
Figura 6 – Subtela product backlog	29
Figura 7 – Subtela sprint	29
Figura 8 – Visão do usuário	30
Figura 9 – Esforço empregado utilizando a ferramenta SGP	34
Figura 10 – Esforço empregado utilizando a ferramenta Redmine	34
Figura 11 – Avaliação de tempo e metodologia da ferramenta SGP	34
Figura 12 – Avaliação de tempo e metodologia da ferramenta <i>Redmine</i>	34
Figura 13 – Qualidade de desenvolvimento de atividades na ferramenta SGP	35
Figura 14 – Qualidade de desenvolvimento de atividades na ferramenta <i>Redmine</i>	35
Figura 15 – Ferramenta avaliada como mais adequada	35
Figura 16 – Avaliação da qualidade do produto atribuído ao uso da ferramenta de gestão	36
Figura 17 – Gráfico de satisfação da ferramenta SGP	36
Figura 18 – Gráfico de satisfação da ferramenta <i>Redmine</i>	36
Figura 19 – Utilização para projetos particulares ou profissionais	37

Lista de abreviaturas e siglas

API	Application Programming Interface
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
JS	JavaScript
SGBD	Sistema de Gerenciamento de Banco de Dados
SGBDOR	Sistema Gerenciador de Banco de Dados Objeto Relacional
SQL	Structured Query Language
DoD	Definition of Done

Sumário

1	INTRODUÇÃO	12
1.1	Contexto e Problema	14
1.2	Objetivos	16
1.3	Estrutura do trabalho	17
2	REFERENCIAL TEÓRICO	18
2.1	Metodologia de desenvolvimento de projetos	18
2.1.1	Metodologias Tradicionais	18
2.1.2	Metodologias ágeis	19
2.2	<i>Scrum</i>	20
2.2.1	Os papéis da equipe <i>Scrum</i>	21
2.2.2	Artefatos do <i>Scrum</i>	21
2.2.3	Visão geral dos ciclos de atividades dentro do método <i>Scrum</i>	22
2.3	Padrão de arquitetura MVC	23
3	TRABALHOS RELACIONADOS	24
4	CAPÍTULO DE DESENVOLVIMENTO	25
4.1	SGP (Sistema de gestão de projetos)	25
4.2	Arquitetura da ferramenta	25
4.3	Modelo Tecnológico	26
4.4	Visão Geral	28
5	RESULTADOS	31
5.1	Hipóteses Analisadas	31
5.2	Caracterização dos Participantes	31
5.3	Metodologia dos Testes	32
5.4	Resultados obtidos	33
5.5	Discussão dos resultados	36
6	CONCLUSÃO	38
	REFERÊNCIAS	39

APÊNDICES	42
APÊNDICE A – REPOSITÓRIO DO PROJETO	43
APÊNDICE B – TERMO DE CONSENTIMENTO	44
APÊNDICE C – QUESTIONÁRIO PÓS-EXPERIMENTO	46
APÊNDICE D – PROBLEMA AVALIADO COM AS FERRAMEN- TAS	48
APÊNDICE E – FICHA DE PERFIL DOS PARTICIPANTES	54

1 Introdução

O sucesso de um projeto pode ser determinado pelo nível de satisfação das partes interessadas (NELSON, 2005). Existem 6 critérios que podem servir para mensurar o grau de satisfação das partes: tempo (cumprimento dos prazos), custos (execução dentro do orçamento definido), produto (relativo ao atendimento aos requisitos solicitados), uso (efetividade do produto que realmente é utilizada), aprendizado (conhecimento adquirido ao longo do processo de desenvolvimento) e valor (melhoria que o produto fornece ao negócio) (ATALLA, 2013). Por outro lado, um projeto pode ser definido como fracassado caso haja o abandono total dele antes ou logo após o produto ter sido entregue (ATALLA, 2013).

O gerenciamento de projetos é uma atividade que tem por finalidade fornecer ao grupo de desenvolvimento uma maneira pragmática de executar as tarefas do projeto e aumentar a probabilidade de sucesso do mesmo. Entende-se por grupo de desenvolvimento as pessoas envolvidas direta ou indiretamente sobre um projeto em específico (ANTONIO, 2012).

Amaral (2008) afirma que projetos de *software* apresentam maior grau de dificuldade na gerência do que projetos de outras áreas da indústria como projetos de engenharia civil, e tendem a fracassar em número proporcionalmente superior. Isto remete a um cenário onde apenas 16.2% dos projetos de *softwares* são completados dentro do prazo e com o orçamento previsto, 31.1% dos projetos são descontinuados, e 56.7% dos projetos custam cerca 189% a mais do valor originalmente estipulado (STANDISH, 2014).

A dificuldade do gerenciamento pode ser oriunda de diversos fatores, como: pouca comunicação entre os membros da equipe de desenvolvimento; impossibilidade de definir completamente todos os requisitos antes do início de um novo projeto; um elevado grau de mutabilidade dentro do escopo do projeto, que implica em alterações profundas na base de decisões; aplicação inadequada de processos de gerência; não-utilização de uma metodologia de desenvolvimento adequada; dentre outros fatores que podem levar ao fracasso ou à descontinuação do projeto (BRACCO, 2008).

Uma forma de enfrentar esses problemas é adotar metodologias que facilitem o gerenciamento, detecção de problemas, diminuição de possíveis perdas e estimulem o aumento da produtividade dos desenvolvedores. Uma metodologia de gerenciamento de projetos tem como objetivo padronizar, unificar os processos e direcionar trabalhos dentro do ciclo de vida de um projeto, visando contornar ou evitar empecilhos e impedimentos dentro de um novo projeto (REIS, 2011). Um gerente de projetos pode empregar diferentes metodologias, ou até mesmo mesclar duas ou mais para melhor atender o contexto do problema que ele enfrenta. Dentre os padrões de desenvolvimento mais utilizados na indústria de software, temos as metodologias ágeis e as tradicionais, e as especificidades de cada projeto devem ser consideradas ao escolher a melhor solução para o projeto (GROFFE, 2015). O

emprego correto de metodologias adequadas pode ser um fator decisivo no sucesso de um projeto.

A engenharia ágil trata da combinação de um conjunto de diretrizes de desenvolvimento, dentre elas o encorajamento do cliente à contínua participação na tomada de decisões acerca do projeto. Agilidade em termos de processo de desenvolvimento de software significa flexibilidade, ou seja, administrar a complexidade de um produto de software de forma flexível (MARCONDES et al., 2003). A agilidade também pode ser definida como uma forma em que a equipe de desenvolvimento trabalha, visando tornar-se mais pragmática a adoção de mudanças dentro do escopo de um projeto (PRESSMAN, 2006).

Nos métodos ágeis o desenvolvimento é feito de forma incremental, e pequenas partes do produto - as novas versões ou *releases* (versão do produto de software ao qual foram adicionadas pequenas modificações, correções ou melhorias) - são desenvolvidas e disponibilizadas aos clientes em curtos intervalos de tempo. Esse tipo de abordagem permite que a equipe de desenvolvimento baseie-se na implementação e entrega do produto, e não apenas em sua concepção e documentação, adiantando o software com valor agregado para o cliente, ou seja, funcionalidades no sistema e já sendo possível receber um retorno sobre elas, o que pode diminuir custos com erros de requisitos mal formados e/ou incompletos (SOMMERVILLE, 2011).

Dentre as metodologias ágeis existe o método ágil *Scrum*, um *framework* que permite às pessoas tratar e resolver problemas complexos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível, como funcionalidades para o sistema de maior importância dentre outras (SCHAWABER, 2017). O *Scrum* é uma junção de boas práticas que podem elevar o grau de sucesso que um projeto pode eventualmente ter. O *framework Scrum* não detalha o que deve ser feito e não resolve os problemas das empresas. O seu objetivo é dar visibilidade a estes problemas e servir como guia na resolução dos mesmos, de forma pragmática por meio de rituais (conjunto de reuniões e práticas que visam elicitar fatores que tenham impacto sobre o andamento do projeto, bem como ajudar na tomada de decisões), gerando conhecimento que pode eventualmente dar origem a artefatos para o projeto (subproduto gerado através do desenvolvimento de software, como por exemplo: documentação, diagramas dentre outros) (JACQUES, 2009).

O intuito deste trabalho é o desenvolvimento de uma ferramenta para auxiliar no gerenciamento de projetos que utilizem a metodologia ágil *Scrum*, permitindo gerir os artefatos criados no decorrer do processo de desenvolvimento do produto, organizar de forma pragmática todas as reuniões que o *framework* descreve, gerar gráficos que facilitem a visualização e identificação do estado em que o projeto se encontra e fornecer informações que ajudem na tomada de decisão por parte dos líderes do projeto.

1.1 Contexto e Problema

Para entender com clareza o que significa gerenciamento de projetos, qual sua importância e quais as dificuldades enfrentadas no ato de controlar e gerir a implementação de novos ou manutenção de projetos legados, é importante antes definir o que um projeto significa. Entende-se como projeto um empreendimento, caracterizado por uma sequência lógica não repetitiva de eventos, delimitado por um tempo pré-definido, destinado a alcançar um objetivo, e sendo conduzido por um grupo de indivíduos seguindo parâmetros previamente definidos como custo, recursos dentre outros (VARGAS, 2016).

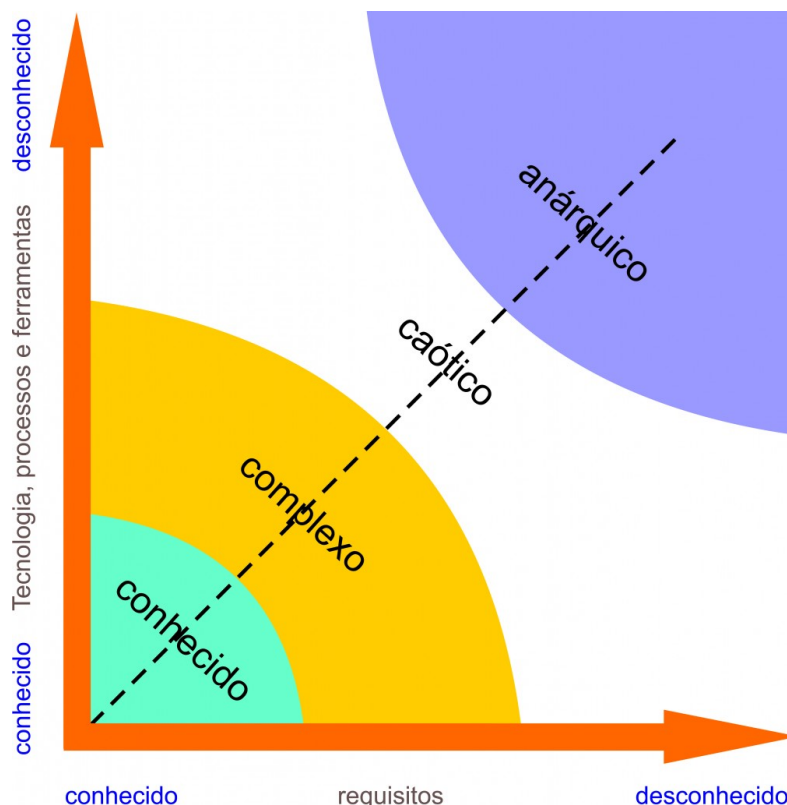
Gerenciamento de projetos é uma das atividades de maior relevância dentro do processo de desenvolvimento de novos ou na manutenção de antigos projetos, um bom gerenciamento não garante um sucesso de um produto porém o mal gerenciamento normalmente resulta em falhas no projeto: elevação do custo estimado, atraso na entrega e falha no atendimento de um ou mais requisitos do projeto (SOMMERVILLE, 2007, p.61).

Quais procedimentos e técnicas melhor se adequam para o desenvolvimento de projeto irá depender muito de qual domínio do problema ele se enquadra, grau de incertezas que o projeto tem, complexidade, dificuldade de recursos e ferramentas. A Figura 1 representa os possíveis cenários que podem ser encontrados em um projeto. Para cenários em que é possível conhecer a totalidade dos requisitos e fatores como recursos extremamente custosos, podendo ser citado como exemplo um projeto de construção de um novo edifício onde é possível fazer o levantamento de todos os requisitos, mas os recursos de execução e a realização de testes são extremamente caros, um planejamento longo é bem realizado antes da execução é o mais indicado, normalmente cenários como esse a abordagem tradicional é a melhor alternativa. Para os casos em que não é possível conhecer com detalhamento aprofundado todos os requisitos, ou casos em que há constantes mudanças no projeto, uma abordagem utilizando metodologia ágil é a mais indicada, normalmente projetos de software se enquadram neste cenário de projeto (MASSARI, 2014).

Dentro das metodologias tradicionais, o planejamento serve para prevenir mudanças ou ao menos tentar evitar o máximo delas, diferentemente as metodologias ágeis, as mudanças são vistas como oportunidades e normalmente são encorajadas e incorporadas ao projeto. Segundo Mah (2008), projetos desenvolvidos com metodologias ágeis são cerca de 16% mais produtivos em relação aos desenvolvidos com outros métodos usados pela indústria, e tem um grau maior de confiança por parte do cliente.

Informações precisas e acessíveis sobre o projeto e são tão essenciais quanto a manipulação de outros recursos como materiais e equipamentos, pois são necessários à definição de metas, planejamento e elaboração do cronograma de implementação, alocação de recursos e pessoal, dentre outras necessidades que possam surgir (SILVEIRA, 2007). Este é um dos motivos que levam muitos líderes de projetos a fazer a adoção de ferramentas de gestão dos dados do projeto, tais como: data de início e término do projeto, estado das tarefas e atividades do projeto, responsáveis pelas tarefas, estimativas de conclusão e

Figura 1 – Como escolher a melhor abordagem.



Fonte: (MASSARI, 2014).

andamento do projeto, custos, prioridades das atividades, dentre outras.

No presente estudo, foram analisadas duas ferramentas de controle de projetos que adotam como principal metodologia de desenvolvimento o *Scrum*: *Scrumwise Better Scrum*¹ e *Redmine*².

Ambas as ferramentas operam como serviços *web* e oferecem suporte ao gerenciamento de múltiplos projetos. A *Scrumwise Better Scrum* apresenta interface agradável e prática, e não necessita de instalação e configuração de infraestrutura, visto que opera como serviço fornecido pela empresa *Scrumwise copyright*. Porém, carece de adaptação linguística para outros idiomas, oferecendo como opção apenas o idioma inglês. Além disso, é um sistema *shareware* (há alguma restrição no uso, como limite de tempo de utilização gratuita ou mesmo em relação ao acesso a funcionalidades do software), e não é possível efetuar modificações no sistema, como a adição de funcionalidades com *plugins* que o usuário ache pertinentes, nem mesmo na versão paga.

A *Redmine* oferece suporte à adaptação linguística do sistema, opera sob a licença GPL (*General Public License* - Licença Pública Geral) e é multiplataforma. *Redmine* possui ainda um sistema de fórum integrado que torna possível a comunicação fácil entre *stakeholders* dos projetos, possibilitando a instalação de diversos *plugins*. Entretanto,

¹ <https://www.scrumwise.com/scrum/>

² <http://www.redmine.org/>

sua interface pouco intuitiva, sua difícil instalação para usuários não técnicos deve ser apontada.

É interessante ressaltar que ambas as ferramentas não oferecem integração nativa com sistemas de versionamento de código, como o *GitHub*³ e o *Bitbucket*⁴.

Mesmo já havendo ferramentas que sirvam para gerir projetos desenvolvidos com métodos ágeis, as características e limitações levantadas evidenciam que ainda há espaço para o desenvolvimento de soluções mais específicas para a gestão de projetos ágeis de software.

1.2 Objetivos

Propor uma ferramenta de gestão de projetos de abordagem *Scrum*, com isto será possível centralizar com maior facilidade informações que porventura possam ser de relevância para a equipe de análise e de desenvolvimento, tais como: separação e ordenação das atividades que estão ou foram desenvolvidas, bem como a geração de relatórios e estatística dos projetos gerenciados pelo sistema para o auxílio do time de desenvolvimento nas tomadas de decisões sobre o projeto.

Objetivos específicos

Este trabalho tem como objetivos específicos:

1. Estudar, analisar e compreender os conceitos de metodologias de desenvolvimento;
2. Reunir informações relevantes e necessárias para o desenvolvimento do módulo *Scrum* para a ferramenta de gestão de produtos;
3. Realizar pesquisas e comparativos de ferramentas existentes de gerenciamento de projetos que tenham como principal método de desenvolvimento o *framework Scrum*;
4. Desenvolver a interface de utilização para a ferramenta implementada;
5. Verificar o esforço empenhado para gerenciar o desenvolvimento de projetos com e sem a utilização da abordagem *Scrum* em ferramentas de gestão;
6. Analisar qualidade do desenvolvimento de projetos gerenciados com e sem a utilização da abordagem *Scrum* em ferramentas de gestão;
7. Analisar a satisfação dos usuários ao gerenciar o desenvolvimento de projetos com e sem a utilização da abordagem *Scrum* em ferramentas de gestão.

³ <https://github.com/>

⁴ <https://bitbucket.org/>

1.3 Estrutura do trabalho

Este trabalho está organizado em 6 (seis) capítulos, incluído o capítulo de introdução. O capítulo 2 contém todo o embasamento teórico necessário desenvolvimento deste trabalho, neste são apresentados os conceitos utilizado ao longo deste documento, o capítulo 3 é um apanhado de trabalhos relacionados ao tema ferramentas de gestão de projetos, nele é demonstrando as vantagens, desvantagens, condições e limitações de trabalhos anteriores e o aprimoramento que ocorreu com o implementação da ferramenta. O capítulo 4 é o desenvolvimento, onde é apresentado a construção da proposta deste trabalho, que ganhou o nome SGP (Sistema de gestão de projetos), capítulo 5 é feita uma análise dos resultados obtidos com este trabalho e capítulo 6 são apresentadas as considerações finais do autor.

2 Referencial Teórico

Neste capítulo será apresentado todo o embasamento teórico necessários para o desenvolvimento deste trabalho. O capítulo esta dividido em 3 (três) seções: a primeira Seção 2.1 aborda o assunto de metodologias de desenvolvimento de projetos, Seção 2.2 o foco será a o método *Scrum* para desenvolvimento de projetos e finalmente na Seção 2.3 irá apresentar o padrão de arquitetura MVC.

2.1 Metodologia de desenvolvimento de projetos

Soares (2004) afirma que um processo ou metologia de desenvolvimento *software* é um conjuntos de eventos e atividades que associadas auxiliam na elaboração, construção e manutenção de um sistema de *software*. As metodologias de desenvolvimento são abordagens, que organizadas, podem atingir um determinado objetivo, cumprindo-se cada etapa de procedimentos preestabelecidos no início do processo. Através das mesmas, é possível controlar as etapas de desenvolvimento de um projeto, para que estas ocorram conforme o planejado, obedecendo prazos e requisitos. Caso estas etapas sejam mal conduzidas podem pôr em risco o sucesso ou o andamento do projeto (SOMMERVILLE, 2011).

O desenvolvimento de software deixou de ser uma atividade desempenhada apenas por programadores, e passou a abranger especialistas em diversas áreas, cada membro concentrado em uma parte necessária para produzir um sistema de software. As metodologias de desenvolvimento surgiram com o objetivo de definir papéis para todos os envolvidos no processo de desenvolvimento, informando de forma clara **quem faz o que** e como essas pessoas devem proceder no andamento do projeto (SOMMERVILLE, 2011).

Existem diversas atividades e eventos que podem ser associadas ao desenvolvimento de projeto como o levantamento de requisitos, análise de requisitos e implementação, execução de testes dentro outras. Na engenharia de software as principais abordagens metodológicas para o desenvolvimento de projetos são as tradicionais e as ágeis.

2.1.1 Metodologias Tradicionais

Também conhecidas como “metodologias pesadas” ou orientadas a documentação, as metodologias tradicionais surgiram na época em que o desenvolvimento de *software* era baseado apenas em *mainframes*. O custo para se fazer alterações ou correções era bastante alto, pois o acesso aos computadores na época era bastante limitado e, além disso, não haviam depuradores e analisadores de código ou *debugs*, ferramentas que auxiliam na detecção de erros durante o desenvolvimento de *software*. Na tentativa de custos, era

necessário efetuar o planejamento de todo o *software* e realizar a documentação antes de iniciar a codificação e implementação do mesmo (KOSCIANSKI; SOARES, 2007).

As metodologias tradicionais são geralmente aplicadas em situações onde os requisitos do software são estáveis. Isso não é muito comum, pois muitos requisitos são modificados ao longo do desenvolvimento conforme as mudanças solicitadas pelos *stakeholders*, que comumente possuem dificuldade para definir o escopo do *software* que será desenvolvido (SOARES, 2004).

De acordo com Almeida (2017), dentre as características mais comuns das metodologias tradicionais, podemos citar as:

- Fases bem definidas de ciclo de vida de desenvolvimento de *software*;
- Documentação abrangente sobre o produto gerado e suas fases de desenvolvimento;
- Maior formalização da comunicação através da documentação;
- Papéis bem definidos para o time de desenvolvimento;
- Busca de padronização das etapas de desenvolvimento.

Dentre as metodologias tradicionais, destaca-se o modelo *Waterfall* que foi um modelo descrito por Dr Winston W. Royce em seu artigo "*Managin the development of large software systems*". Royce (1970) descreveu que existem duas principais etapas para o desenvolvimento de um programa computacional independente do seu tamanho ou complexidade, que são as etapas de análise e de codificação.

Para Royce (1970), o conceito do modelo em cascata era interessante, mas ele via em sua implementação riscos e acreditava que estes eram convites para falhas no desenvolvimento do projeto. Detalhes do projeto como: custos, material necessário para a implementação, documentação e arquitetura são definidos de forma precoce, e geralmente quando ainda não é possível enxergar todas enuncias e particularidades do sistema, o que pode gerar erros de análise e requisitos incompletos ou inadequados. Neste método de desenvolvimento as mudanças são fortemente indesejadas e desencorajadas, devido o grande custo que é gerado, (HONÓRIO, 2012).

2.1.2 Metodologias ágeis

As metodologias ágeis são alternativas que vem ajudando diversas equipes de desenvolvimento de *software* a encarar a imprevisibilidade de um projeto. Utilizando entregas incrementais e ciclos iterativos as metodologias ágeis tornaram-se uma alternativa aos métodos tradicionais, mencionados anteriormente. As metodologias ágeis somente se tornaram conhecidas em 2001, graças a um grupo de 17 (dezessete) especialistas em processos de desenvolvimento de *software*, que se reuniram nos EUA, com o intuito de achar métodos e processos que fossem comum aos projetos de sucesso, e que trouxesse o melhores

resultados com o mínimo de esforço possível (SOARES, 2004). Os especialistas extraíram características comuns de alguns métodos conhecidos como *Scrum* e *XP*¹ e, foi então criado o documento mais importante da comunidade ágil, o “*Manifesto Ágil*”².

A partir disso o desenvolvimento ágil descreveu as abordagens de desenvolvimento seguindo os seguintes princípios:

- Indivíduos e interação - os softwares são construídos por uma equipe, na qual seus membros trabalham juntos;
- Software Executável - Ao invés da documentação abrangente;
- Colaboração do cliente - Ter contrato é importante para definir responsabilidades e direitos, mas não deve substituir a comunicação;
- Responder rápido a mudanças - O projeto deve ter um plano, porém este deve ser flexível para comportar mudanças.

As metodologias ágeis são adaptativas, ou seja, possibilitam que mudanças possam ocorrer mais facilmente no decorrer de um projeto. Além disso, tais metodologias trabalham com constante *feedback*, o que permite eventualmente modificar e alterar os requisitos do projeto durante o processo de desenvolvimento (Barbosa, 2010).

2.2 Scrum

Dentre as metodologias ágeis podemos destacar o *Scrum*, um modelo ágil de processo. Segundo Schawaber (2017) é um dos métodos mais conhecido e possivelmente um dos mais difíceis de ser implantados dentro de uma organização que já tenha uma cultura de desenvolvimento de produtos de forma tradicional ou em cascata. A entrega forma contínua e incremental do produto, os rápidos *feedbacks* e o menor custo de manutenção das funcionalidades implementadas que foram invalidadas pelo cliente são algumas das características que podem ser vistas como vantagens dos métodos ágeis sobre as metodologias tradicionais como o método *waterfall* (CELESTINO, 2014).

A abordagem *Scrum* é um método ágil geral, mas o seu foco está no gerenciamento do desenvolvimento iterativo, ao invés das abordagens técnicas específicas da engenharia de *software* ágil (SCHAWABER, 2017). *Scrum* tem três pilares fundamentais:

Transparência Sugere que todos os *stakeholders* do projeto devem ter conhecimento sobre o andamento da implementação do sistema. Ou seja, as informações são sempre compartilhadas entre o time de desenvolvimento e demais pessoas que tem parte com o projeto.

¹ Metodologia de desenvolvimento de *software*, criada no Estados Unidos na década de 90

² Declaração dos princípios fundamentais do desenvolvimento ágil, para mais informações este documento acesse o site: <http://agilemanifesto.org/principles.html>

Inspeção É a prática que é adotada sobre o andamento do projeto, todo o tempo são inspecionadas as atividades que estão em andamento ou que já foram concluídas. Esta prática serve para verificar se as mesmas foram implementadas de acordo com os quesitos estipulados para o projeto. O *Scrum* realiza diversas reuniões para efetuar as inspeções dentro das *Sprints* (Ciclos de atividades que devem ser exercidas em um determinado tempo) do projeto, dentre os grupos de reuniões destacam-se a *Daily Scrum*³ e a *Sprint Review Meeting*⁴.

Adaptação Sugere que caso o produto desenvolvido tenha algum aspecto ou característica desviada dos limites considerados aceitáveis, o produto deve ser ajustado o mais breve possível para minimizar mais desvios.

2.2.1 Os papéis da equipe *Scrum*

Para as responsabilidades e divisões dos papéis no *Scrum* temos 3 (três) grupos principais: **Time de desenvolvimento**, o grupo de pessoas encarregadas de desenvolver o projeto; **PO** (*Product Owner*), a pessoa responsável por definir o produto, tomar as decisões relativas de negócio do desenvolvimento do projeto e fazer o intercâmbio entre o cliente e a equipe de desenvolvimento; e o ***Scrum Master*** que é um líder servidor, ou seja, ele trabalha para facilitar e potencializar o andamento do projeto, sendo responsável por efetuar a retirada de empecilhos que possam estar atrapalhando o desenvolvimento do projeto. (SABBAGH, 2013).

2.2.2 Artefatos do *Scrum*

Dentro da metodologia *Scrum*, temos originalmente a geração quatro tipos de artefatos, são eles:

Product Backlog lista de funcionalidades que se acredita que será implementada pelo time de desenvolvimento no decorrer do projeto. Essa lista é frequentemente atualizada, e ordenada de acordo com a importância e prioridade que o cliente do projeto dá para o item (SABBAGH, 2013).

Sprint Backlog lista de itens selecionados de maior prioridade do *Product Backlog* para o desenvolvimento do incremento do produto no *Sprint*, os itens serão confirmados ou não durante a construção do planejamento da *Sprint*, junto a um plano de realização desse trabalho (SABBAGH, 2013). Normalmente é utilizado um quadro de tarefas para verificar o andamento das atividades que encontra-se na *Sprint*.

³ Reunião diária feita para a resolução de problemas que por ventura a equipe esteja enfrentando.

⁴ Reunião que tem como objetivo avaliar se as metas da *Sprint*.

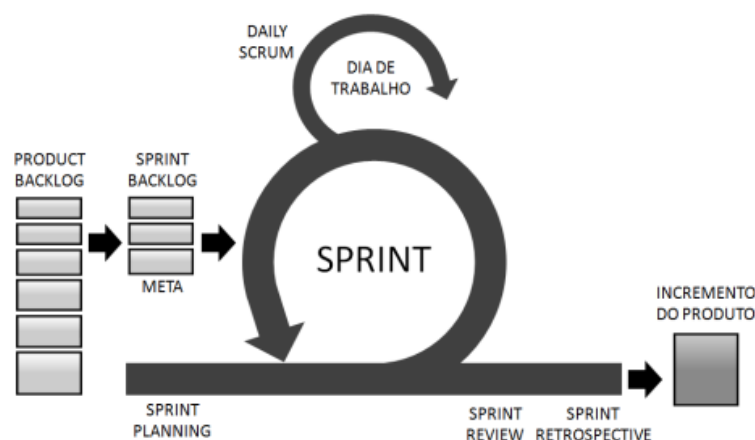
Definição de Pronto (*Definition of Done*) documento formal desenvolvido pela equipe *Scrum* que define claramente quais são os passos mínimos para a conclusão de um item e quando ele é considerado pronto. (PEDRO, 2016)

Incremento no Produto produto desenvolvido durante a *Sprint* que é potencialmente entregável ao cliente. não necessariamente é uma regra entregar em final de uma *Sprint* um produto com características e funcionalidades novas, mas é aconselhável fazer-lo porque assim é possível ter um *feedback* do cliente ou PO o mais rápido possível (SABBAGH, 2013).

2.2.3 Visão geral dos ciclos de atividades dentro do método *Scrum*

O ciclo de atividades dos projetos *Scrum* dá-se da seguinte forma (Figura 2): Primeiramente é feito pelo PO o levantamento de características e funcionalidades que irão compor o *Product Backlog*, e então é realizado o *Sprint Planning*, uma reunião que tem como objetivo definir as metas da *Sprint* e os itens do *Product Backlog* que nela serão implementados. Diariamente são realizados os *Daily Scrums*, que tem por finalidade informar ao *Scrum Master* os possíveis empecilhos a ser enfrentados no dia. Ao final da *Sprint* são realizadas duas reuniões: a *Sprint Review*, uma revisão da *Sprint* e das metas alcançadas ou não, e uma demonstração do que foi desenvolvido ao PO; e a *Sprint Retrospective*, onde é feito o levantamento dos pontos onde o time de desenvolvimento acertou e errou para que seja possível melhorar *Sprints* posteriores. Ao final de todo esse processo, o PO verifica se um produto potencialmente entregável será fornecido ao cliente (SABBAGH, 2013).

Figura 2 – Visão geral do ciclo de atividades do método *Scrum*.

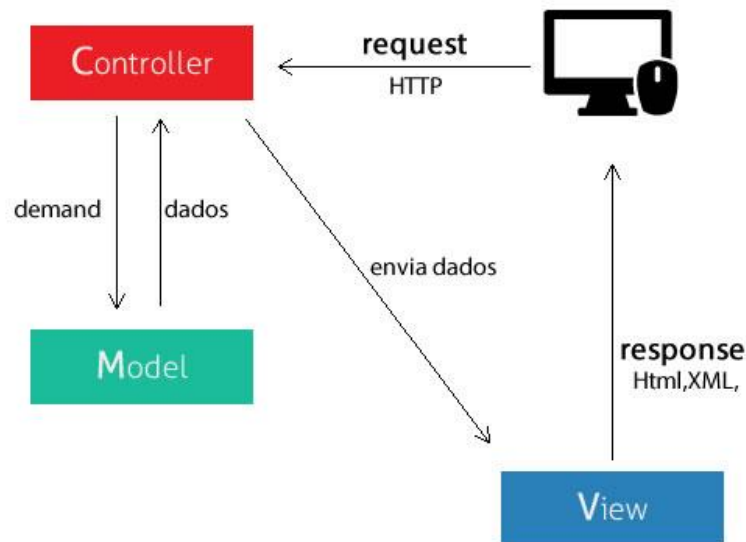


Fonte: Sabbagh (2013)

2.3 Padrão de arquitetura MVC

O padrão de arquitetura MVC (*Model View Controller*) está retratado na (Figura 3). Pode-se notar que existe uma separação clara dos componentes neste padrão. Por exemplo, os componentes da camada de *View* (visão, interface ou apresentação, a camada responsável por fazer a interação com o usuário do aplicação) não interagem diretamente com os componentes do *Model* (camada responsável pela manipulação dos dados), e sim o fluxo de informações é apenas transmitido pela camada de *Controller* (controle e gerenciamento do fluxo de informações). A utilização deste padrão pode implicar na facilidade de manutenção de um software (RAMOS, 2015).

Figura 3 – Padrão de arquitetura MVC



Fonte: (RAMOS, 2015)

3 Trabalhos Relacionados

Como já foi mencionado anteriormente o gerenciamento de projetos é uma atividade que tem como objetivo fornecer à equipe de desenvolvimento e todos os outros *stakeholders* de um projeto uma forma pragmática para a execução das atividades que venham a contribuir para o andamento do projeto e aumentar sua probabilidade de sucesso (ANTONIO, 2012). Sendo uma atividade que trabalha bastante a manipulação das informações do projeto, existe a necessidade de que essas informações estejam o mais atualizadas e acessíveis possível para que as tomadas de decisões da liderança do projeto sejam adequadas e ocorra a melhoria na qualidade do produto final e economia de recursos.

O problema de gestão de informações de um projeto já foi atacado por diversas ferramentas com diferentes metodologias de desenvolvimento. Dentre elas o *Redmine*, que de acordo com o trabalho de Moura e Nascimento (2009), foi a ferramenta utilizada para auxiliar o desenvolvimento de software no Centro de Recursos Computacionais, Universidade Federal de Goiás. Entre os benefícios citados na utilização desta se destacam: agregação de agilidade e simplicidade além da integração e melhor organização entre o time de desenvolvimento.

O trabalho de Souza, Siqueira e Freitas (2010), relatou que a "ferramenta ocasionou um melhor controle de impactos de manutenção ou de mudanças de requisitos, através da automatização dos cadastros e estabelecimento de rastreabilidade entre itens."

Para o trabalho de Vasconcelos (2014), que propôs a utilização da plataforma do *Redmine* para o auxílio de aprendizagem da disciplina de gerência de projetos. Foi obtido o relato de 92% das pessoas que usaram a ferramenta terem tido um melhor aprendizado na disciplina e 85% das pessoas relataram que adotaria o sistema gestão integrado para o uso no seu ambiente de trabalho. Ainda neste mesmo trabalho foi relatado sobre o benefício em relação ao processo de escrita manual, utilização de editores de texto e planilhas eletrônicas, sentiu-se uma maior rapidez em relação ao armazenamento e gerenciamento de informações.

O trabalho de Yoshidome et al. (2013), relatou que as ferramentas livres de gestão implicam na diminuição de custo de implantação do MPS.BR. Nos resultados dos trabalhos apresentados, notou-se que existe certa melhora no processo de gerenciamento de projetos utilizando a ferramenta *Redmine*, o sistema proposto neste trabalho difere visto que tentou adicionar características de outra metodologia que fizessem que o processo de gerenciar com a ferramenta de gestão fosse ainda mais eficiente e prático.

4 Capítulo de Desenvolvimento

Esse capítulo tem o objetivo de apresentar o desenvolvimento da ferramenta proposta. Aqui serão abordados a arquitetura utilizada para a implementação e as ferramentas escolhidas para a construção do *software*, bem como a justificativa dessas escolhas. Também serão expostos diagramas e imagens para facilitar a compreensão das informações passadas.

4.1 SGP (Sistema de gestão de projetos)

O SGP foi o sistema desenvolvido neste trabalho, e para a sua elaboração foi adicionado a ferramenta o máximo de características tidas como positivas das ferramentas anteriores, no caso a *Scrumwise Better Scrum* e a *Redmine*. As características escolhidas foram, ser uma ferramenta de código fonte aberto, para que com isso seja possível receber a contribuição de funcionalidades ou recursos de outros desenvolvedores, ser intuitiva, gratuita, e que ofereça uma boa interação entre os usuários da aplicação, e além de utilizar a metodologia *Scrum* como o método de gestão na ferramenta. Nos testes realizados com o uso dela, no que tange os aspectos de intuitividade e interação foram dados como bem sucedidos, visto que grande parte das pessoas que executaram os testes e utilizaram a ferramenta informaram que voltariam a utilizar para projetos pessoais ou profissionais, porque se mostrou fácil e prática para o desenvolvimentos das atividades.

4.2 Arquitetura da ferramenta

O desenvolvimento do sistema seguiu o padrão de arquitetura MVC. Com isso é possível desenvolver mudanças tanto na camadas de lógica de negócio do sistema (*Model*), quanto na de apresentação desses dados (*View*), sem ser necessário efetuar grandes alterações na aplicação, este foi um dos motivos que levou a escolha desse padrão de arquitetura para a ferramenta SGP.

Para o desenvolvimento da camada de interação com o usuário do *software*, optou-se pela escolha de implementação de aplicação baseada na *web*. Ou seja, qualquer usuário da aplicação que tenha acesso a um navegador *web* irá conseguir acessar os recursos do sistema sem grandes complicações. Devido à maior facilidade de manutenção e escalabilidade e ao custo de desse tipo de aplicação, viu-se esse tipo como melhor escolha (BYCHKOV, 2013).

Para as informações e os dados gerados através da ferramenta SGP, optou-se pela persistência dessas informações por meio de um sistema de banco de dados padrão SQL

(*Strutured Query Language*.), tendo em vista que é o padrão mais aceito e utilizado pela indústria de *software* atualmente (SOARES, 2017).

4.3 Modelo Tecnológico

Diversas tecnologias foram utilizadas para o desenvolvimento e materialização da ferramenta SGP. O sistema foi construído utilizando a linguagem de programação *Ruby*¹ e o *meta-framework Ruby on Rails*², este ultimo tendo como objetivo fornecer um conjunto de bibliotecas capazes de auxiliar a implementação do sistema. *Ruby On Rails* segue o padrão de arquitetura MVC e possui uma gama considerável de desenvolvedores e uma documentação bastante abrangente de suas funcionalidades, o que torna agradável e atrativa a sua adoção como ferramenta de desenvolvimento (FUENTES, 2012).

Para a construção das telas de interação do usuário com o sistema foi utilizado a biblioteca CSS (*Cascading Style Sheets*) *Ui Semantic*³ para dar maior flexibilidade e facilitar a manutenção das páginas HTML (*HyperText Markup Language*). Em conjunto com esta, também foram adicionadas algumas bibliotecas *Javascript* como *Jquery*⁴ e *Nodejs*⁵. O banco de dados escolhido para essa fase de desenvolvido foi o *SQLite3*⁶, que é um sistema de banco de dados que consegue gerenciar um número limitado de transações mas em contrapartida é de fácil instalação e *zero-configuration* (zero configuração), e para fins de desenvolvimento era extremamente capaz de suprir a demanda de persistência de informações da aplicação. Para facilitar o controle e versionamento do sistema, foi utilizada a ferramenta de versionamento de código *Git*⁷ em conjunto com a plataforma *GitHub*⁸, que possibilitou um melhor gerenciamento e distribuição do código entre a equipe de desenvolvimento.

Mediante os requisitos levantados para a implementação da ferramenta SGP, ferramenta essa que tem como finalidade ser um sistema gestão de projetos que tenha como método principal a metodologia *Scrum*, foi desenvolvido o diagrama de entidades relacional sistema (Figura 4), que representa a estrutura e os relacionamentos entre as classes que servem como modelos de objetos para a aplicação.

Podemos verificar a existência das entidades *Sprint*, *UserStore* e *UserStoryAcceptanceCriteria*s que são algumas das quais das entidades serviram para gerir os aspectos de informações do *Scrum* dentro da ferramenta.

¹ <https://www.ruby-lang.org/pt/>

² <http://rubyonrails.org/>

³ <https://semantic-ui.com/>

⁴ <https://jquery.com/>

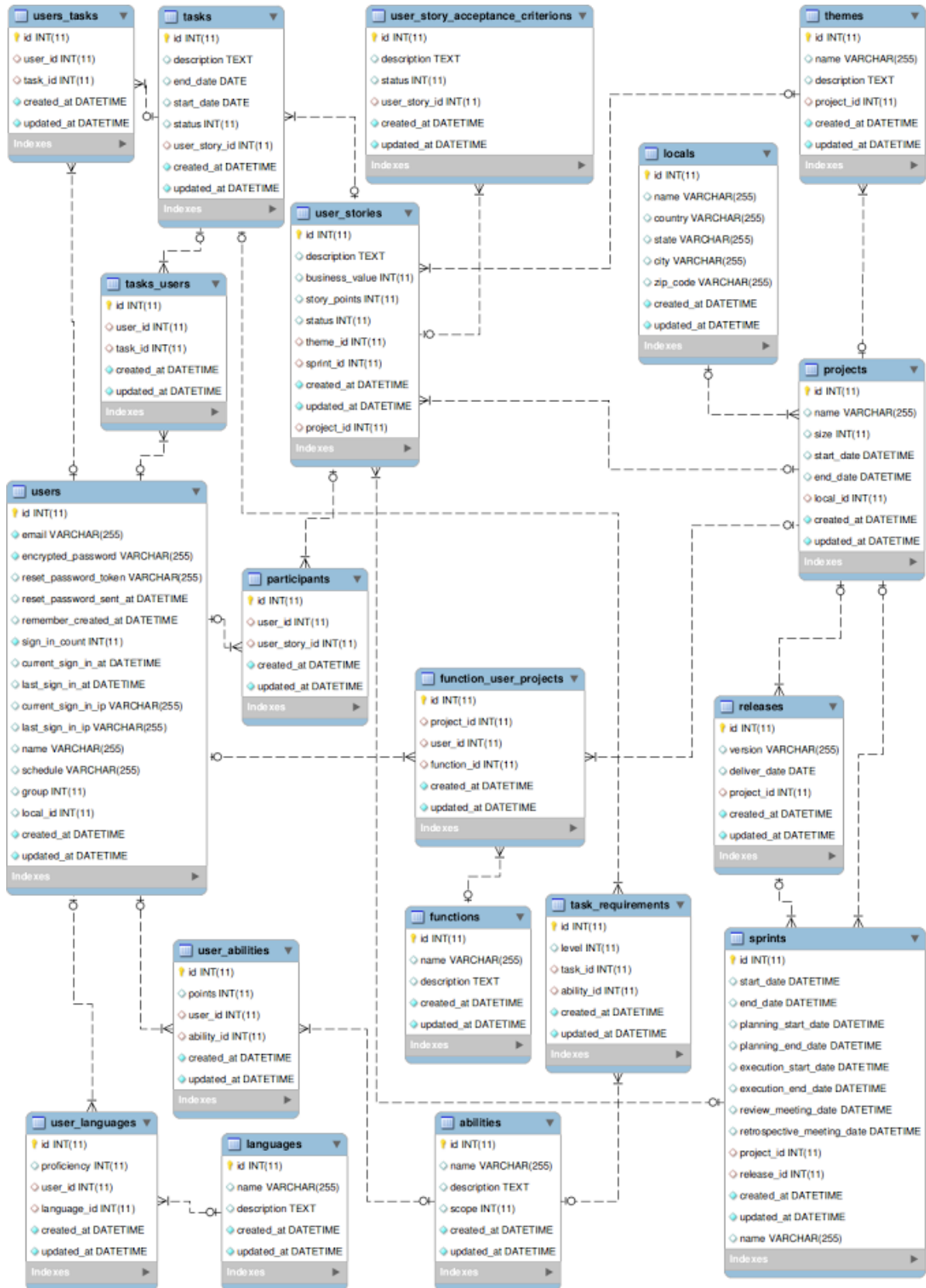
⁵ <https://nodejs.org/en/>

⁶ <https://www.sqlite.org/>

⁷ <https://git-scm.com/>

⁸ <http://github.com/>

Figura 4 – Diagrama entidade relacional do sistema SGP



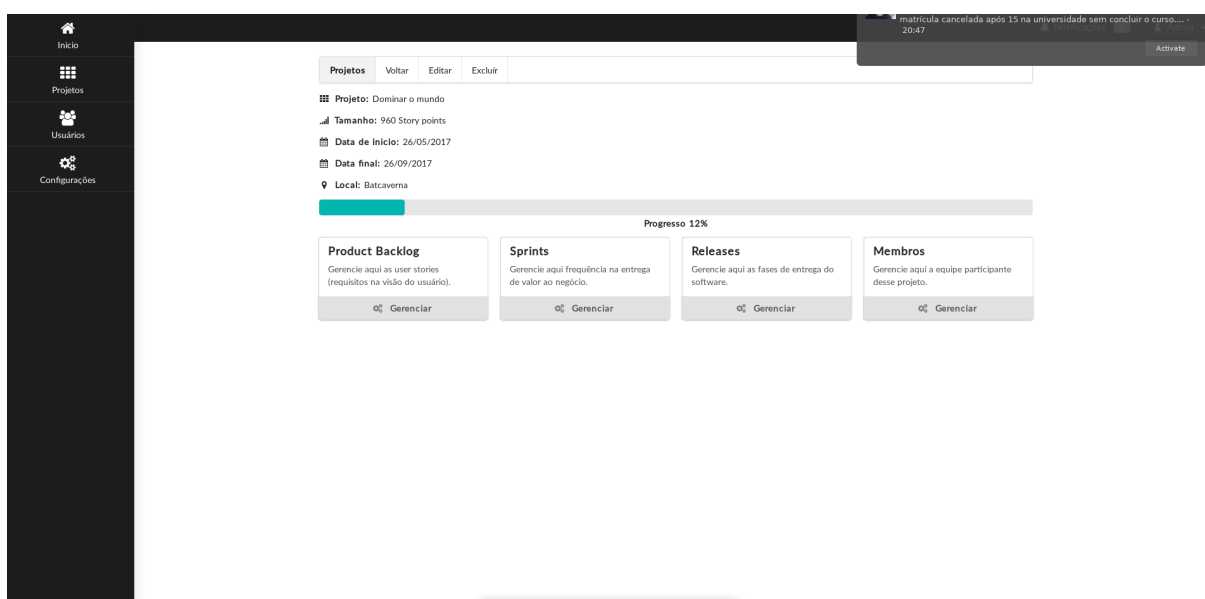
Fonte: O autor

4.4 Visão Geral

Finalizada a etapa de desenvolvimento, seguiu-se para fase de testes da ferramenta com um grupo de participantes. As análises dos testes executados encontra-se no Capítulo 5. Nas Figuras 5, 6 e 8 são apresentadas algumas das telas de interações com os usuários.

A Figura 5, exibe a tela de recursos e funcionalidades administrativas, nela é possível notar a presença das características pertinentes ao método *Scrum*, como o gerenciamento dos itens de *product backlog*, *Sprints* e *releases*, também é possível fazer o gerenciamento de membro do projeto e ver em qual a porcentagem concluída do projeto.

Figura 5 – Tela de funcionalidades administrativas



Fonte: O autor

A Figura 6, representa a subtela destinada a efetuar o gerenciamento do *product backlog* do projeto gerenciado em questão, é possível definir para cada funcionalidade adicionada um conjunto de critérios de aceitação que nada mais são do que características que devem ser cumpridas para fazer a homologação daquele item.

Figura 7, temos a subtela de gerenciamento de *Sprints* do projeto, com ela é possível efetuar modificações, adicionar novas *sprint* bem como verificar as *releases* que por ventura aquela *Sprint* possa ter.

Figura 8 representa a tela do usuário padrão do sistema, onde se apresentam as funções pertinentes aos time de desenvolvimento sem acesso administrativo. É possível notar a presença algumas características do método Kanban, onde é possível verificar a presença de separadores de estado das tarefas. São 4 (quatro) estados possíveis, no caso podemos verificar as tarefas ou atividades **para fazer**, **fazendo**, em **teste** e finalmente **concluído**.

Figura 6 – Subtela product backlog

Tema	Descrição	BV	SP	ROI	Status	Sprint	Critérios de aceitação	Tarefas	Ações
Comunicação	Dominar os meios de co...	800	600	1	Desenvolvimento	Marco zero	+ Adicionar [Remover] Documentação [Remover] Rápido e indolor [Remover] Homologação	+ Adicionar [Remover] Hackear sistema globo de comunicação [Remover] Criar API para comunicação entre sistemas [Remover] Conseguir senha internas	[Lupa] [Linha] [Linha]
Interface	Criar interface agradá...	800	350	2	Desenvolvimento	Marco zero	+ Adicionar [Remover] Semantic UI	+ Adicionar [Remover] Instalar ferramentas [Remover] Documentar ferramentas [Remover] Sem ideias para essa atividade...	[Lupa] [Linha] [Linha]
Comunicação	Acabar com programa da...	200	10	20	Pendente		+ Adicionar [Remover] Não deixar rastros	+ Adicionar [Remover] Criar uma bomba [Remover] Soltar a bomba	[Lupa] [Linha] [Linha]

Fonte: O autor

Figura 7 – Subtela sprint

Nome	Duração	Planejamento	Execução	Review meeting	Retrospective meeting	Release	Ações
Marco zero	26/05/2017 à 26/09/2017	26/05/2017 à 26/05/2017	27/05/2017 à 26/09/2017	26/09/2017	26/09/2017	Versão 1.0	[Exibir] [Editar] [Remover]

Fonte: O autor

Figura 8 – Visão do usuário

The screenshot displays a Scrumboard interface for a user named 'joseue martins'. The interface is divided into several sections:

- Top Bar:** Contains a notification bell icon with 'Notificações 1', the user's name 'joseue martins', and a dropdown arrow.
- Header:** Shows the title 'Scrumboard' and the current sprint name 'Marco zero'.
- Sprint Schedule Table:**

Duração	Planejamento	Execução	Review meeting	Retrospective meeting	Release
26/05/2017 à 26/09/2017	26/05/2017 à 26/05/2017	27/05/2017 à 26/09/2017	26/09/2017	26/09/2017	Sexy sem ser vulgar
- Progress Bar:** A teal bar indicates '16% Concluído'.
- Task Board Columns:**
 - Para fazer:** Contains one task: 'Criar API para comunicação entre sistemas'. It offers 600 XP points and lists skills: Matemática, Programação OO, and Tester Java. Buttons for 'Detalhes' and 'Fazer' are present.
 - Fazendo:** Contains two tasks:
 - 'Instalar ferramentas' (350 XP, skills: Programação OO, Designer UX) assigned to 'usuario'.
 - 'Documentar ferramentas' (350 XP, skill: Programação OO) assigned to 'israel linhares'.
 - Teste:** Contains two tasks:
 - 'Hackear sistema globo de comunicação' (600 XP, skill: Matemática) assigned to 'joseellito'.
 - 'Conseguir senha internas' (600 XP, skill: Comunitivo) assigned to 'israel linhares'.
 - Concluído:** Contains one task: 'Sem ideia para essa atividade...' (350 XP, skills: Matemática, Comunitivo, Programação OO, Designer UX, Tester Java, Quimico) assigned to 'joseue martins', completed on 01/06/17.
- Sidebar (Left):** Includes a user profile picture, name 'joseue martins', and navigation icons for 'Início', 'Projetos', 'Ranks', and 'Minhas Conquistas'.

Fonte: O autor

5 Resultados

Segundo Chibeni (2006), a concepção de conhecimento válido sobre determinado assunto é feita apenas por meio do chamado método científico. De forma simples, as etapas básicas que o método científico define são: a observação dos fenômenos, a indução de hipóteses, a verificação por meio de testes, e a validação ou não dessas hipóteses (RAZUK, 1995).

O ponto nevrálgico abordado neste trabalho é a dificuldades em gerir projetos, principalmente quando não se faz o uso de uma metodologia adequada no processo de desenvolvimento e a ausência de ferramenta capaz de suprir as necessidades de gerenciamento de projeto. No Capítulo 2, foram explanados as metodologias *Waterfall* e a *Scrum*, bem como levantadas as características de ambas e fornecidos os pontos negativos e positivos dessas duas metodologias. Fazendo-se o uso do método científico, serão analisadas algumas hipóteses levantadas neste trabalho.

Na Seção 5.1 serão apresentadas as hipóteses levantadas, será feita a descrição da caracterização dos participantes dos testes efetuados na seção 5.2. Na Seção 5.3 será descrita qual foi a metodologia utilizada para a execução dos testes, e os resultados obtidos são apresentados na Seção 5.4. as considerações sobre os resultados serão feitas na Seção 5.5.

5.1 Hipóteses Analisadas

As hipóteses levantadas e analisadas foram:

- O esforço empenhado para gerenciar o desenvolvimento de projetos é maior com ou sem a utilização da abordagem *Scrum* em ferramentas de gestão;
- A qualidade do desenvolvimento de projetos gerenciados é melhor com a utilização da abordagem *Scrum* em ferramentas de gestão;
- A satisfação dos usuários ao gerenciar o desenvolvimento de projetos é melhor com a utilização da abordagem *Scrum* em ferramentas de gestão.

5.2 Caracterização dos Participantes

O estudo iniciou com o total de 21 participantes, mas apenas 19 deles concluíram e preencheram todos os questionários corretamente. Foi entregue uma cópia do formulário de perfil de participante (Apêndice E) para cada participante que executou os testes, para que fosse possível analisar e diferenciar os resultados obtidos nos testes de acordo com nível conhecimento em metodologias de desenvolvimento de projetos de cada participante.

Também foi entregue o termo de consentimento (Apêndice B), para que todos os participantes dos testes tomassem ciência do que seria feito com as informações fornecidas por eles.

Todos os membros envolvidos nos testes realizados são acadêmicos de graduação da UFPI - Universidade Federal do Piauí. Todos os participantes informaram ter algum conhecimento ou já terem cursado alguma disciplina que abordasse os assuntos de engenharia de *software*, programação orientada a objetos e programação *web*. Cerca de 95.24% afirmaram ter conhecimento de análise de projeto de sistemas, 66.67% sobre o assunto de gerência de projetos, e 90.48% informaram já terem participado do desenvolvimento de algum projeto de *software*. Apenas 47.62% informaram ter algum conhecimento ou ter cursado disciplina que abordassem o assunto de teste de *software* e apenas 28.57% informaram terem gerenciado o desenvolvimento de algum projeto de *software*.

5.3 Metodologia dos Testes

Os testes foram executados no laboratório de informática da UFPI - Universidade Federal do Piauí do campus de Picos do curso de sistemas de informação. Os participantes foram separados em dois grupos, cada participante recebeu uma cópia de uma folha contendo um problema a ser avaliado e desenvolvido com auxílio de uma das ferramentas de gestão de produtos.

Para fins de comparação, nos testes foram utilizadas as ferramentas *Redmine* e SGP que foi a ferramenta desenvolvida neste trabalho. O conteúdo do problema avaliado pode ser verificado no Apêndice D. Vale salientar que o sistema *Redmine* foi utilizado sem o uso de *plugin Scrum*¹, ou seja, essa ferramenta foi usada apenas com abordagem *waterfall* que é o padrão da ferramenta. Apenas um dos sistemas no caso o SGP foi utilizado com a abordagem *Scrum*.

Cada um dos dois grupos ficou responsável em utilizar e desenvolver as atividades com uma das ferramentas, no caso *Redmine* ou SGP. Após concluída essa primeira etapa, os grupos inverteram a utilização das ferramentas. Ou seja, o grupo que utilizou a ferramenta *Redmine* passou a utilizar a SGP e o mesmo ocorreu com o grupo que utilizou primeiramente a ferramenta SGP, e foram refeitas as mesmas atividades desenvolvidas com mesmo problema a ser avaliado que foi utilizado anteriormente. Decorrida essa segunda etapa, foi entregue então a cada participante dos testes um questionário de avaliação (Apêndice C), tendo como objetivo fazer a avaliação das ferramentas utilizadas por cada participante. Um termo de consentimento (Apêndice B) e um formulário de perfil de participante (Apêndice E) também foi entregue cada membro que executou os testes. O termo foi utilizado para que o participante tomasse ciência do que seria feito com as informações fornecidas

¹ É possível instalar um *plugin* que ative características da metodologia *Scrum* na ferramenta *RedMine*

por ele, e o formulário de perfil de participante para que fosse possível traçar o perfil do grupo de participantes que executaram os testes.

O questionário de avaliação da ferramenta foi separado em três grupos de perguntas, e cada um desses grupos é responsável por fazer o levantamento de informações para que seja possível verificar a validade das hipóteses levantadas neste trabalho. Os grupos de perguntas são:

- Grupo A: perguntas referentes a quanto de esforço empenhado para gerenciar o desenvolvimento de projetos com a ferramenta;
- Grupo B: perguntas referentes à qualidade do desenvolvimento de atividades de gerenciamento de projetos na ferramenta;
- Grupo C: perguntas relacionadas à satisfação em gerenciar o desenvolvimento de projetos com a ferramenta.

5.4 Resultados obtidos

Serão apresentados os gráficos gerados a partir das informações colhidas nos testes e realizada a comparação das informações obtidas para as duas ferramentas analisadas.

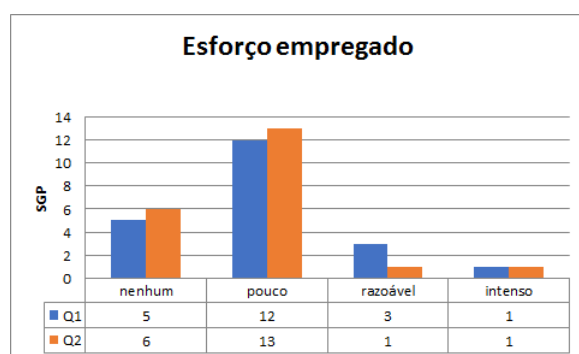
Os gráficos nas Figuras 9 e 10 apresentam as respostas fornecidas pelos participantes ao questionamento de esforço desempenhado para execução de atividades nas ferramentas, onde pode ser notado que foi necessário menos esforço para o desenvolvimento de atividades na ferramenta SGP em comparação a *Redmine*.

Os rótulos **Q1** e **Q2** nas Figuras 9 e 10 representam respectivamente a primeira e a segunda questão do **grupo A** de perguntas. Pode ser observado que para a **Q1** o total de pessoas que consideraram o esforço intenso na ferramenta SGP representa apenas 9% da quantidade que consideraram esforço intenso com a utilização da ferramenta *Redmine*. A situação mostra-se semelhante quando é feita a observação na **Q2** em ambos os gráficos.

Os rótulos **Q3** e **Q4** nos gráficos das Figuras 11 e 12, representam respectivamente a terceira e a quarta questão do **grupo A** de perguntas. **Q3** é a indagação relativa ao tempo gasto para efetuar atividades na ferramenta, **Q4** diz respeito sobre a metodologia aplicada e utilizada na ferramenta, em ambas as questões foram dadas as opções de avaliação: inadequada, razoável, bom e ótimo. Houve apenas um participante que avaliou como inadequado o tempo gasto para executar atividades na ferramenta SGP, o que representa apenas 10% da quantidades de participantes que tiveram a mesma conclusão para a ferramenta *Redmine*. Para a **Q4**, nenhum dos participantes avaliaram o sistema *Redmine* como tendo uma ótima metodologia, já para o sistema SGP houve a obtenção de cerca de 14 (quatorze) avaliações dadas como ótimo.

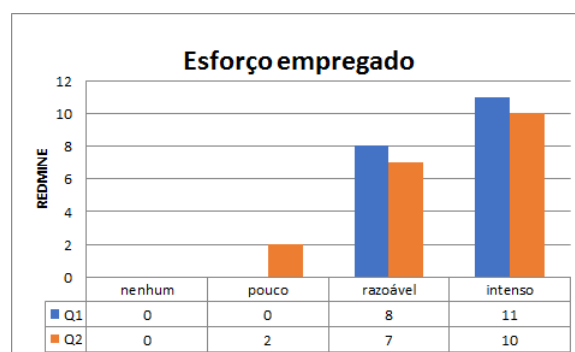
Para o **grupo B** de perguntas, os rótulos **Q1** e **Q2** nos gráficos das Figuras 13 e 14 representam a avaliação dos participantes para a primeira e a segunda questão respecti-

Figura 9 – Esforço empregado utilizando a ferramenta SGP



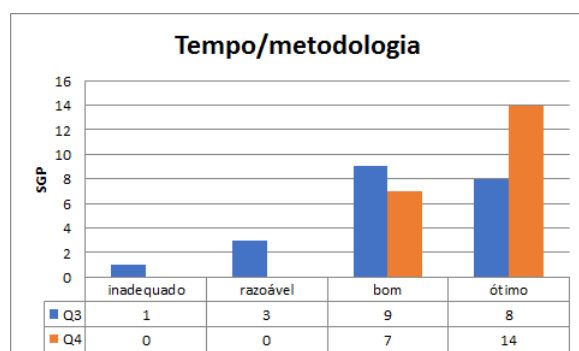
Fonte: Produzido pelo autor

Figura 10 – Esforço empregado utilizando a ferramenta Redmine



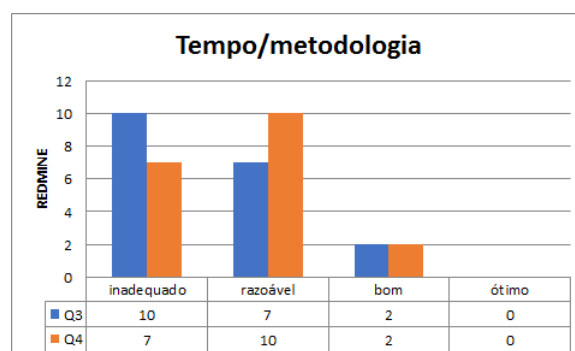
Fonte: Produzido pelo autor

Figura 11 – Avaliação de tempo e metodologia da ferramenta SGP



Fonte: Produzido pelo autor

Figura 12 – Avaliação de tempo e metodologia da ferramenta Redmine



Fonte: Produzido pelo autor

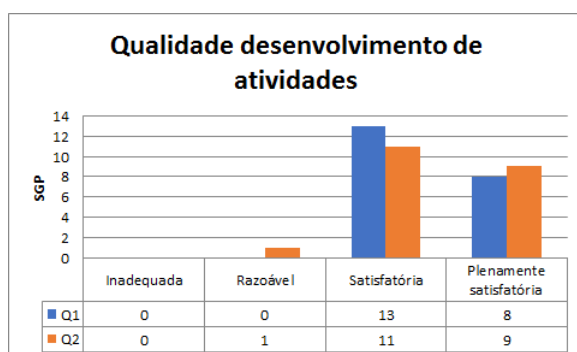
vamente. Nenhum dos participantes avaliou como plenamente satisfatória a qualidade no desenvolvimento das atividades no sistema *Redmine*, já para a outra ferramenta ocorreu essa avaliação.

O gráfico apresentado na Figura 15, representa as respostas dadas pelos participantes dos testes para a terceira questão do **grupo B** de perguntas. É feito o questionamento para o avaliador se ele acredita que a ferramenta é adequada para a gestão de produtos, 12 (doze) pessoas avaliaram o sistema SGP como extremamente adequado para esta finalidade e apenas uma pessoa avaliou o mesmo para a ferramenta *Redmine*.

Para a questão 4 do **grupo B** de perguntas, o gráfico apresentado na Figura 16 informa que nenhum dos avaliadores definiu como alta a qualidade no produto atribuída com a utilização da ferramenta *Redmine*, para o sistema SGP houve 16 (dezesesseis) pessoas avaliaram esse quesito como alto.

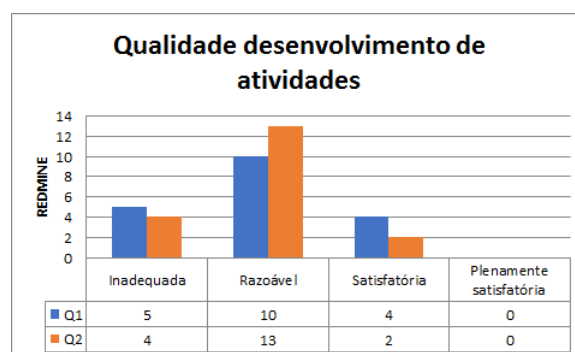
Para o **grupo C**, as Figuras 17 e 18 representam os gráficos gerados a partir das informações colhidas para a primeira e segunda questões deste grupo de perguntas, rotulados respectivamente de **Q1** e **Q2**. A primeira pergunta desse grupo indaga se o avaliador sente

Figura 13 – Qualidade de desenvolvimento de atividades na ferramenta SGP



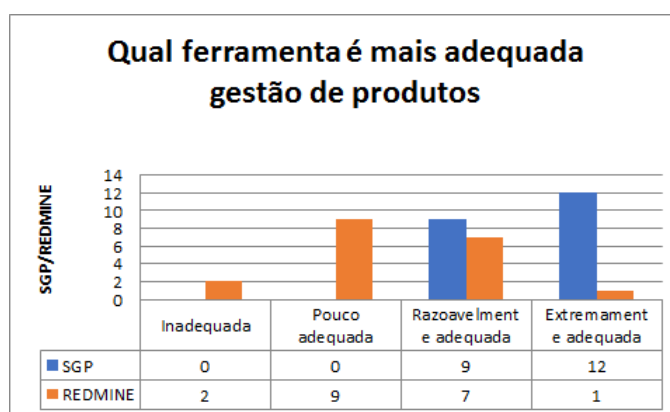
Fonte: Produzido pelo autor

Figura 14 – Qualidade de desenvolvimento de atividades na ferramenta Redmine



Fonte: Produzido pelo autor

Figura 15 – Ferramenta avaliada como mais adequada

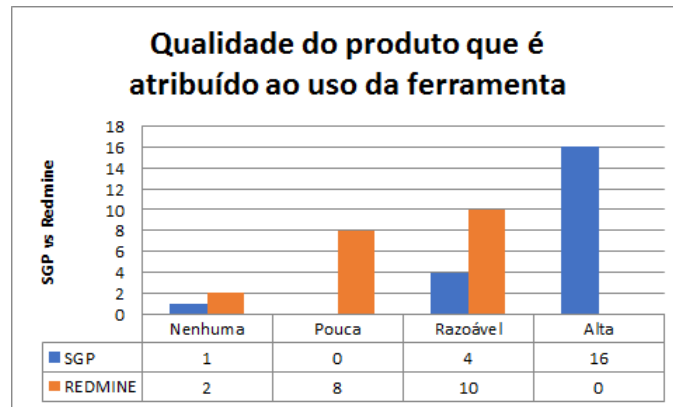


Fonte: Produzido pelo autor

que a ferramenta fornece um grau satisfação para as necessidades de gerenciamento de informações de um projeto, nenhum participante avaliou o sistema *Redmine* com um grau ótimo de satisfação para essas necessidades, e cerca de 4 pessoas avaliaram este grau como pouquíssimo, ou seja, muitos dos avaliadores não acreditam que a ferramenta satisfaça as necessidades de gerenciamento de informações a cerca de um projeto. O sistema SGP obteve cerca de 16 (dezesseis) avaliações de grau ótimo para esta questão e nenhuma em grau pouco ou pouquíssimo.

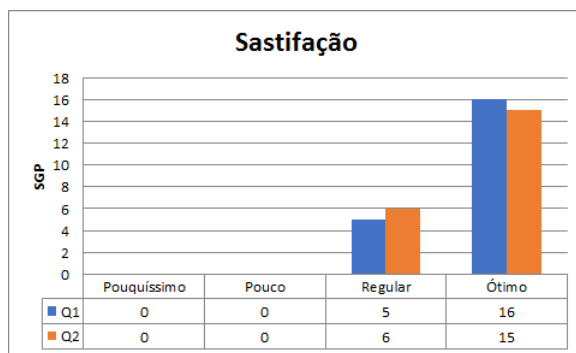
Os participantes foram questionados se utilizariam para projetos pessoais ou profissionais uma das duas ferramentas analisadas, apenas um avaliador respondeu que não utilizaria o sistema SGP para projetos pessoais, o mesmo ocorreu para projetos profissionais. Em contra-partida 18 (dezoito) participantes informaram que não utilizariam a ferramenta *Redmine* para projetos pessoais, a mesma quantidade informou que também não utilizaria para projetos profissionais.

Figura 16 – Avaliação da qualidade do produto atribuído ao uso da ferramenta de gestão



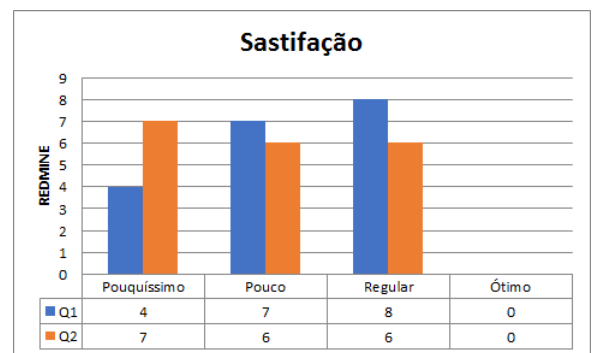
Fonte: Produzido pelo autor

Figura 17 – Gráfico de satisfação da ferramenta SGP



Fonte: Produzido pelo autor

Figura 18 – Gráfico de satisfação da ferramenta Redmine

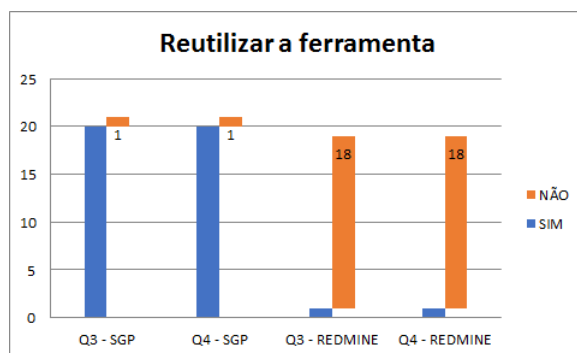


Fonte: Produzido pelo autor

5.5 Discussão dos resultados

Mediante os dados levantados com os testes, é possível constatar que houve uma melhor aceitação do avaliadores para com a ferramenta SGP, que foi o sistema desenvolvido neste trabalho. Em todos os questionamentos e comparações a ferramenta se saiu melhor avaliada do que a outra analisada, no caso a ferramenta *Redmine*. Tanto o esforço empregado, a qualidade no desenvolvimento de atividades e a satisfação dos usuários ao gerenciar projetos foram substancialmente melhores em uma ferramenta que utilizou a abordagem *Scrum*, tanto para a sua concepção quanto para a abordagem que a ferramenta utiliza projetos que serão gerenciados com ela.

Figura 19 – Utilização para projetos particulares ou profissionais



Fonte: o autor

6 Conclusão

Mesmo já havendo ferramentas que sirvam para gerir projetos desenvolvidos com métodos ágeis, as características e limitações levantadas evidenciam que ainda há espaço para o desenvolvimento de soluções mais específicas para a gestão de projetos ágeis de software. A proposta de um sistema de gerenciamento de projetos que se utilize da abordagem do método *Scrum*, bem como a verificação se este ofereceu algum ganho ou melhoria em comparação as ferramentas já existentes no mercado foi um dos assuntos que tangeram a concepção deste trabalho.

Ao longo desse trabalho, foram apresentados conceitos sobre metodologias de desenvolvimento, onde elas atuam, para que servem e qual a diferença entre as metodologias tradicionais e as ágeis. Foi feito o embasamento teórico desses conceitos, e também foram levantados alguns questionamentos em relação a qual metodologia é mais eficiente e eficaz para desenvolver um projeto.

As hipóteses levantadas no Capítulo 5 mostraram-se verdadeiras quando foram feitas as análises e comparações dos resultados obtidos através dos testes executados com duas ferramentas, no caso a SGP e a *Redmine*.

Pode ser constatado que houve benefício com o desenvolvimento desse trabalho, visto cerca de 96% dos participantes que executaram os testes informaram que voltariam a utilizar a ferramenta tanto para projetos pessoais, quanto para projetos profissionais. Vale salientar que a ferramenta ainda carece de algumas melhoras, como a possibilidade de integração com serviços de versionamento de código com a plataforma *GitHub*, ou mesmo a adição de elementos de gamificação para aumentar e melhorar o engajamento do time de desenvolvimento nos projetos.

Referências

- ALMEIDA, G. A. M. de. Fatores de escolha entre metodologias de desenvolvimento de software tradicionais e ágeis. 2017. Citado na página 19.
- AMARAL, F. Por que projetos de software falham? 2008. Disponível em: <<http://www.linhadecodigo.com.br/artigo/2054/por-que-projetos-de-software-falham.aspx>>. Acesso em: 4 de julho de 2017. Citado na página 12.
- ANTONIO, A. Gerenciamento de projetos: Uma análise da importância da estrutura de gerenciamento de projetos no alcance dos objetivos estratégicos. 2012. Citado 2 vezes nas páginas 12 e 24.
- ATALLA, F. Por que os projetos de ti fracassam? 2013. Disponível em: <http://www.techoje.com.br/site/techoje/categoria/detalhe_artigo/1532>. Acesso em: 4 de julho de 2017. Citado na página 12.
- BRACCO, P. D. C. Sucesso e fracasso nos projetos. 2008. Disponível em: <<http://gerenciamentodeprojetos-npj.blogspot.com.br/2008/02/sucesso-e-fracasso-nos-projetos.html>>. Acesso em: 4 de julho de 2017. Citado na página 12.
- BYCHKOV, D. Desktop vs. web applications: A deeper look and comparison. 2013. Disponível em: <<http://www.seguetech.com/desktop-vs-web-applications/>>. Acesso em: 4 de julho de 2017. Citado na página 25.
- CELESTINO, A. L. Scrum: A eficiência de uma sprin. 2014. Disponível em: <<https://www.profissionaisiti.com.br/2014/03/scrum-a-eficiencia-de-uma-sprint/>>. Acesso em: 4 de julho de 2017. Citado na página 20.
- CHIBENI, S. S. *Algumas observações sobre o “método científico”*. [S.l.], 2006. Citado na página 31.
- FUENTES, V. B. *Coloque sua aplicação web nos trilhos*. São Paulo: 2012, 2012. Citado na página 26.
- GROFFE, R. Projetos de software: uma comparação entre metodologias ágeis e tradicionais. 2015. Disponível em: <<http://netcoders.com.br/comparacao-metodologias-ageis-e-tradicionais/>>. Acesso em: 4 de julho de 2017. Citado na página 12.
- HONÓRIO, M. C. Algumas diferenças entre scrum e modelo waterfall. 2012. Disponível em: <<https://www.portaleducacao.com.br/conteudo/artigos/informatica/algumas-diferencas-entre-scrum-e-modelo-waterfall/14137>>. Acesso em: 4 de julho de 2017. Citado na página 19.
- JACQUES, D. V. *Implantando o Scrum em um ambiente de desenvolvimento de produtos para internet*. Rio de Janeiro: [s.n.], 2009. Citado na página 13.
- KOSCIANSKI, A.; SOARES, M. dos S. *Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software*. 2. ed. São Paulo: 2007, 2007. 391 p. Citado na página 19.

- MAH, M. How agile projects measure up, and what this means to you. 2008. Disponível em: <http://www.researchandmarkets.com/reports/660321-/how_agile_projects_measure_up_and_what_this.pdf>. Acesso em: 4 de julho de 2017. Citado na página 14.
- MARCONDES, F. S. et al. Agilidade em desenvolvimento de software. 2003. Disponível em: <<http://bt.fatecsp.br/system/articles/574/original/bt1748.pdf>>. Acesso em: 4 de julho de 2017. Citado na página 13.
- MASSARI, V. Gerenciamento Ágil versus gerenciamento tradicional: quem ganha? 2014. Disponível em: <<https://www.profissionaisti.com.br/2014/11/gerenciamento-agil-versus-gerenciamento-tradicional-quem-ganha/>>. Acesso em: 4 de julho de 2017. Citado 2 vezes nas páginas 14 e 15.
- MOURA, M. H. D. de; NASCIMENTO, H. A. D. do. Gerenciamento de projetos com o redmine. 2009. Disponível em: <http://consuni_26-03.pdf.cercomp.ufg.br/up/18/o/redmine-article.pdf>. Acesso em: 4 de julho de 2017. Citado na página 24.
- NELSON, R. R. Project retrospectives: Evaluating project success, failure, and everything in between. 2005. Disponível em: <<http://misqe.org/ojs2/index.php-/misqe/article/view/85>>. Acesso em: 4 de julho de 2017. Citado na página 12.
- PEDRO, D. *Definition of Done: 7 Passos Simples para sua DoD*. [S.l.], 2016. Disponível em: <<http://www.mindmaster.com.br/definition-of-done/>>. Acesso em: 4 de julho de 2017. Citado na página 22.
- PRESSMAN, R. S. *Engenharia de software*. 6. ed. [S.l.]: 2006, 2006. 720 p. Citado na página 13.
- RAMOS, A. Mvc - afinal, é o que? 2015. Disponível em: <<https://tableless.com.br/mvc-afinal-e-o-que/>>. Acesso em: 4 de julho de 2017. Citado na página 23.
- RAZUK, D. P. C. O método científico. 1995. Disponível em: <<http://wwwp.feb.unesp.br/jcandido/metodologia/Apostila/CAP02PG.pdf>>. Acesso em: 4 de julho de 2017. Citado na página 31.
- REIS, G. Gerenciamento de projetos: sua importância e seu valor estratégico. 2011. Disponível em: <<http://www.taskblog.com.br/02/gerenciamento-de-projetos-sua-importancia-e-seu-valor-estrategico/>>. Acesso em: 4 de julho de 2017. Citado na página 12.
- ROYCE dr Winston w. *Managin the development of large software systems*. [S.l.], 1970. Citado na página 19.
- SABBAGH, R. *Scrum 360 - Um guia completo e prático de agilidade*. São Paulo: [s.n.], 2013. Citado 2 vezes nas páginas 21 e 22.
- SCHAWABER, K. *The Definitive Guide to Scrum: The Rules of the Game*. [S.l.], 2017. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us-.pdf>>. Acesso em: /today. Citado 2 vezes nas páginas 13 e 20.

SILVEIRA, P. G. da. O gerenciamento da informação no contexto de projetos: estudo de práticas em empresas brasileiras. 2007. Disponível em: <<http://portaldeperiodicos.eci.ufmg.br/index.php/pci/article/view/503>>. Acesso em: 4 de julho de 2017. Citado na página 14.

SOARES, J. Lista de banco de dados mais utilizados no mercado – 2016/2017. 2017. Disponível em: <<https://codigosimples.net/2017/01/18/lista-de-banco-de-dados-mais-utilizados-no-mercado-20162017/>>. Acesso em: 4 de julho de 2017. Citado na página 26.

SOARES, M. dos S. Comparação entre metodologias Ágeis e tradicionais para o desenvolvimento de software. 2004. Citado 3 vezes nas páginas 18, 19 e 20.

SOMMERVILLE, I. *Engenharia de software*. 8. ed. São Paulo: 2007, 2007. 556 p. Citado na página 14.

SOMMERVILLE, I. *Engenharia de software*. 9. ed. São Paulo: 2011, 2011. 526 p. Citado 2 vezes nas páginas 13 e 18.

SOUZA, D. G. M. de; SIQUEIRA, K. A.; FREITAS, R. L. de. Integração da gerência de requisitos com a plataforma redmine. 2010. Disponível em: <<http://bd.centro.iff.edu.br/bitstream/123456789/428/3/Documento.pdf>>. Acesso em: 4 de julho de 2017. Citado na página 24.

STANDISH, G. I. *Chaos Report*. [S.l.], 2014. Disponível em: <<https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>>. Acesso em: 4 de julho de 2017. Citado na página 12.

VARGAS, R. V. *Gerenciamento de projetos: Estabelecendo diferencias competitivas*. 8. ed. Rio de Janeiro: 2016, 2016. 529 p. Citado na página 14.

VASCONCELOS, S. A ferramenta gestão integrada como auxilio no processo de ensino e aprendizagem. 2014. Disponível em: <http://www.tise.cl/volumen10/TISE2014-tise2014_submission_344.pdf>. Acesso em: 4 de julho de 2017. Citado na página 24.

YOSHIDOME, E. Y. C. et al. Uma implementação do processo de gerência de projetos usando ferramentas de software livre. 2013. Disponível em: <http://www.softex.br/wp-content/uploads/2013/10/27_WAMPS-2010_Uma-Implementa%C3%A7%C3%A3o-do-Processo-de-Ger%C3%Aancia-de-Projetos-Usando-Ferramentas-de-Softwarvre.pdf>. Acesso em: 4 de julho de 2017. Citado na página 24.

Apêndices

APÊNDICE A – Repositório do projeto

Endereço do projeto: <https://github.com/jdruk/sgp>

APÊNDICE B – Termo de consentimento

Termo de Consentimento Livre e Esclarecimento

Baseado nas Diretrizes Contidas na Resolução CNS N°466/2012, MS.

Eu, _____, convidado(a) a participar voluntariamente do estudo experimental do trabalho intitulado “**Módulo Scrum para um sistema de gestão de projetos**”, recebi do graduando **Josafá Martins dos Santos**, responsável pela execução do experimento, as seguintes informações que me fez compreender sem objeções e dúvidas as seguintes informações:

- Esse estudo é destinado a avaliar o gerenciamento do desenvolvimento de projetos que utilizam uma ferramenta de gestão de abordagem tradicional ou ágil;
- Que a importância deste estudo é a de possibilitar a disponibilização de uma ferramenta que facilite o processo de gestão de desenvolvimento de sistemas de computacionais;
- Que almeja alcançar os seguintes resultados:: i) O esforço empenhado para gerenciar o desenvolvimento de projetos com e sem a utilização da abordagem Scrum em ferramentas de gestão; ii) A qualidade do desenvolvimento de projetos gerenciados com e sem a utilização da abordagem Scrum em ferramentas de gestão; iii) A satisfação dos usuários ao gerenciar o desenvolvimento de projetos com e sem a utilização da abordagem Scrum em ferramentas de gestão;
- Que, minha participação nesse estudo não implicará em nenhum risco à minha saúde física ou mental;
- Que, em qualquer momento deste estudo, eu poderei me recusar a continuar participando deste estudo;
- Que os resultados obtidos por meio de minha participação não possibilitará a minha identificação, exceto ao responsável por este estudo;
- Que o estudo realizado iniciou-se às ____:____ na data ____/____/____ e seu término às ____:____ na data ____/____/____;

Considerando, que fui informado(a) dos objetivos e da relevância do estudo proposto, e de como será minha participação, dos procedimentos de que não há riscos decorrentes deste estudo, declaro o meu consentimento em participar da pesquisa, como também concordo que os dados obtidos por meio desse sejam utilizados para fins científicos (publicações). Estou ciente de que receberei uma via desse documento.

Assinatura do Participante ou Responsável Legal

APÊNDICE C – Questionário pós-experimento

Questionário pós-experimento

Ferramenta: SGP Redmine

Quanto ao esforço empenhado para gerenciar o desenvolvimento de projetos

Esforço necessário para efetuar atividades dentro na ferramenta, pode ser considerado?	<input type="checkbox"/> Nenhum <input type="checkbox"/> Pouco <input type="checkbox"/> Razoável <input type="checkbox"/> Intenso
Esforço para o acesso a informações na ferramenta pode ser considerado?	<input type="checkbox"/> Nenhum <input type="checkbox"/> Pouco <input type="checkbox"/> Razoável <input type="checkbox"/> Intenso
O tempo gasto para efetuar tarefas no sistema.	<input type="checkbox"/> Inadequado <input type="checkbox"/> Razoável <input type="checkbox"/> Bom <input type="checkbox"/> Ótimo
A metodologia que a ferramenta fornece para gerir projetos pode ser considerada	<input type="checkbox"/> Inadequado <input type="checkbox"/> Razoável <input type="checkbox"/> Boa <input type="checkbox"/> Ótima

Quanto a qualidade do desenvolvimento de atividades de gerenciamento de projetos na ferramenta

À qualidade no desenvolvimento de atividades na ferramenta pode ser considerada?	<input type="checkbox"/> Inadequada <input type="checkbox"/> Razoável <input type="checkbox"/> Satisfatória <input type="checkbox"/> Plenamente satisfatória
À qualidade de disposição de informações pode ser considerado?	<input type="checkbox"/> Inadequada <input type="checkbox"/> Razoável <input type="checkbox"/> Satisfatória <input type="checkbox"/> Plenamente satisfatória
A ferramenta mostrou-se adequada para a atividade de gestão de projetos?	<input type="checkbox"/> Inadequada <input type="checkbox"/> Pouco adequada <input type="checkbox"/> Razoavelmente adequada <input type="checkbox"/> Extremamente adequada
De maneira geral, o quanto você atribui a qualidade do produto desenvolvido com a utilização desta ferramenta?	<input type="checkbox"/> Nenhuma <input type="checkbox"/> Pouca <input type="checkbox"/> Razoável <input type="checkbox"/> Alta

Quanto a satisfação em gerenciar o desenvolvimento de projetos na ferramenta

Qual grau você acredita que a ferramenta satisfaça as necessidades de gerenciamento de informação de um projeto?	<input type="checkbox"/> Pouquíssimo <input type="checkbox"/> Pouco <input type="checkbox"/> Regular <input type="checkbox"/> Ótimo
Qual grau de satisfação você teve com a ferramenta?	<input type="checkbox"/> Pouquíssimo <input type="checkbox"/> Pouco <input type="checkbox"/> Regular <input type="checkbox"/> Ótimo
Você utilizaria esta ferramenta para projetos pessoais?	<input type="checkbox"/> Sim <input type="checkbox"/> Não
Você utilizaria esta ferramenta para projetos profissionais?	<input type="checkbox"/> Sim <input type="checkbox"/> Não

Sugestões, críticas ou comentários sobre a ferramenta: _____

APÊNDICE D – Problema avaliado com as ferramentas

Problema explorado com as ferramentas

1. Definição do Problema

Com a crescente demanda de criação de páginas *web* por empresas e pessoas, muitas tecnologias foram desenvolvidas para agilizar esse processo, como a criação de CMSs (*Content Manger System* – Sistema Gerenciador de Conteúdo) populares como *Wordpress* para blogs, *Joomla* para portais, *Magento* para *web-Commerce* (Comércio eletrônico), etc. Porém essas ferramentas não dão completo apoio as necessidades dos desenvolvedores, os *layouts* são poucos personalizáveis e a curva de aprendizado para construção de *templates* é longa, outra dificuldade enfrentada é que esses CMSs vem consigo todos os recursos que o desenvolvedor precisa, e os recursos contruídos pelo desenvolvedor é de difícil integração com esses CMSs, como por exemplo: um programador tem a necessidade adicionar a sua página um sistema de sorteio e integra-lo ao site construído com algum CMS; ou um *webdesign* ao construir um *template* precisa integrar ao gerenciador de conteúdo.

Muitos desenvolvedores optam por construir seu próprio gerenciador de conteúdo para ter o total controle do site criado, apesar dos icríveis recursos desses *frameworks* que aceleram o processo de desenvolvimento construir um CMS é um processo que pode demorar, resultando na insatisfação dos clientes.

2. Descrição da Solução

O sistema gerenciador de conteúdo em *Ruby on Rails* baseado em *web service* permitirá a comunicação com outras aplicações através do formato de dados *json*. Este gerenciador possuirá um *template/layout* padrão de página que poderá ser usado pelo usuário, *webdesign* ou programador. O administrador poderá gerenciar páginas, postagens, categorias, e configurações do site, como definir título, sub-título, número de postagens por página, e se o site estará disponível ou não, na sessão de postagens o administrador terá os seguintes camops disponíveis: título, resumo, e conteúdo, onde ele poderá escrever e formatar sua postagem e também realizar o upload de imagens, e definir se a postagem será pública, VIP para os visitantes cadastrador ou como rascunho (condúdo para publicação futura). Na sessão de gerencimanto de páginas o administrador poderá definir um título e um conteúdo estático que será exibido para os visitantes.

Os usuários do site feito com o CMS em *Ruby on Rails* além de poderem navegar pelas postagens e páginas estáticas também poderão comentar nas publicações. Para os usuários terem acesso as postagens VIPs, estes deverão credenciar-se no site, onde poderão logar-se e ter acesso ao conteúdo exclusivo e também poder publicar comentários nas postagens.

Além do módulo do adminstrador que é utilizado para genrenciar o conteúdo na página, a aplicação *web* também disponibilizará seu conteúdo em formato *JSON* no módulo de *web service*, permitindo assim que desenvolvedores e *webdesigns* integrem suas próprias aplicações ou *layouts/templates* com este CMS.

3. Arquitetura do Sistema

3.1 Diagrama UML de classes:

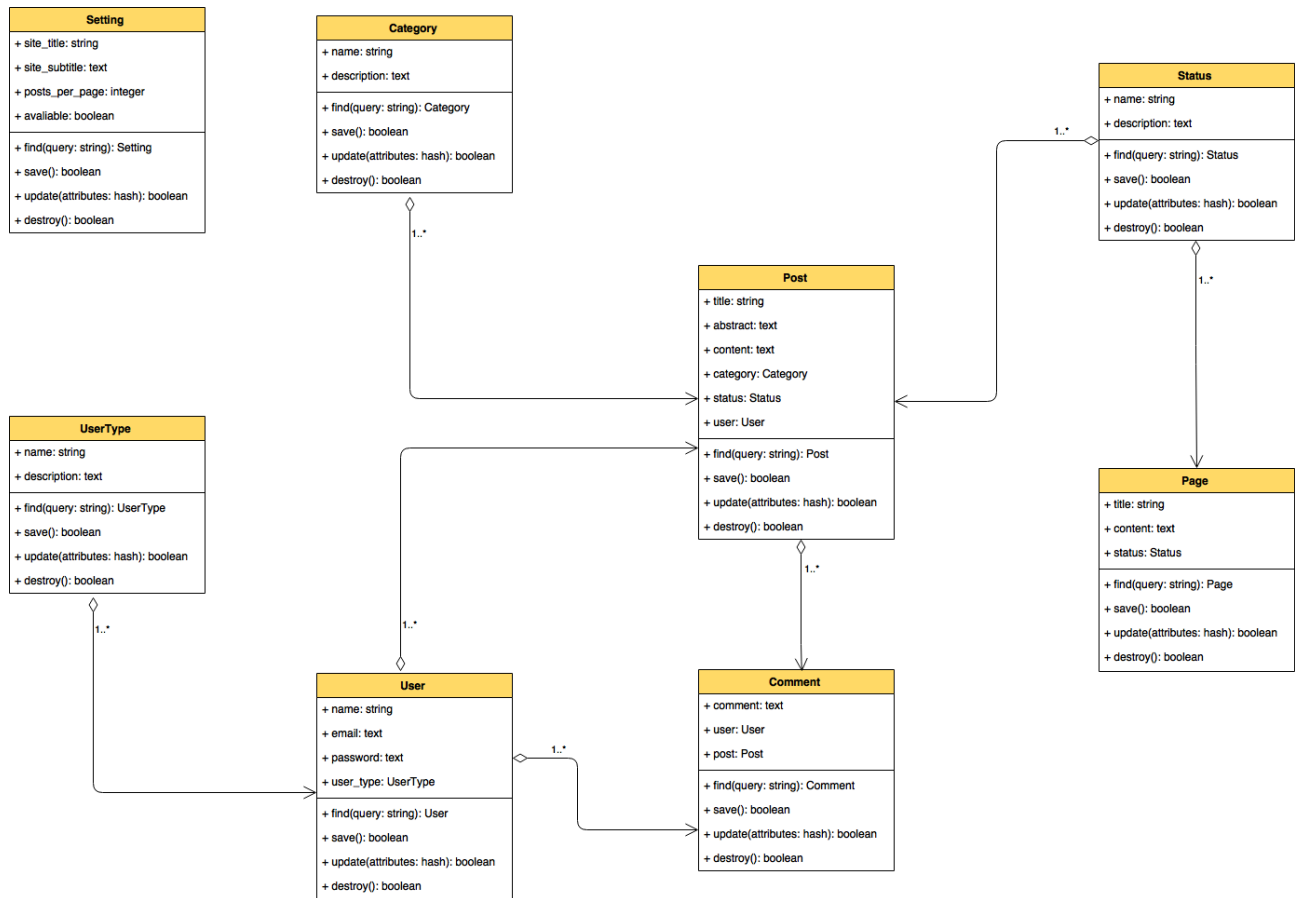


Figura 2 – Diagrama UML de classes

3.2 Product backlog e sprints backlog:

Product backlog:

IBL	TEMA	USER STORY	CRITÉRIOS DE ACEITAÇÃO	VALOR DE NEGÓCIO	STORY POINTS	ROI (VN/SP)	SPRINT
1	Visitante	Como visitante do site, gostaria de poder visualizar as postagens publicadas.	Exibir as postagens publicadas.	100	1	100	1
2	Visitante	Como visitante do site, gostaria de poder navegar pelas páginas do site.	Exibir barra de navegação e páginas de conteúdo estático.	100	2	50	1
3	VIP	Como visitante frequente da página, gostaria de poder me cadastrar no site para me tornar um visitante VIP.	Realizar o cadastro do visitante VIP.	80	4	20	1
4	VIP	Como usuário VIP, gostaria de poder visualizar as postagens para assinantes do site.	Exibir as postagens exclusivas para os usuários VIPs que estejam autenticados no sistema.	80	4	20	2
5	VIP	Como usuário VIP, gostaria de poder publicar comentários nas postagens.	Armazenar e exibir os comentários dos usuários autenticados no sistema.	80	4	20	2
6	Admin	Como administrador do site gostaria de poder, alterar as configurações do site, como título, números de posts por página e a disponibilidade do site.	Armarzenas as informações de configuração do site, exibir mecanismos para manipulação das configurações	100	5	20	2
7	Admin	Como administrador do site, gostaria de gerenciar as publicações de posts.	Gerenciar as publicações dos posts	200	5	40	3
8	Admin	Como administrador do site gostaria de poder gerenciar as páginas estáticas do site.	Gerenciar as páginas estáticas do site.	200	5	40	3
9	Admin	Como administrador do site, gostaria de poder gerenciar as categorias das publicações.	Gerenciar as categorias das publicações do site.	200	5	40	3
10	Dev	Como desenvolvedor que utiliza o CMS, gostaria de acessar as informações do CMS no formato JSON para que eu possa integrar	Disponibilizar as informações publicadas em formato JSON por meio de URLs.	300	6	50	4

		as minhas aplicações.					
11	Dev	Como desenvolvedor que utiliza o CMS, gostaria de poder integrar os conteúdos publicados aos meus templates de páginas web.	Disponibilizar as informações publicadas em formato JSON por meio de URLs.	300	6	50	4

Sprints backlog:

IBL	USER STORY	CRITERIOS DE ACEITAÇÃO	STATUS (POR FAZER, EM ANDAMENTO, FEITO)	RESPONSÁVEL
1	Como visitante do site, gostaria de poder visualizar as postagens publicadas.	Exibir as postagens publicadas.	FEITO	[GF]
2	Como visitante do site, gostaria de poder navegar pelas páginas do site.	Exibir barra de navegação e páginas de conteúdo estático.	FEITO	[RR]
3	Como visitante frequente da página, gostaria de poder me cadastrar no site para me tornar um visitante VIP.	Realizar o cadastro do visitante VIP.	FEITO	[RF]

Sprint backlog 1

IBL	USER STORY	CRITERIOS DE ACEITAÇÃO	STATUS (POR FAZER, EM ANDAMENTO, FEITO)	RESPONSÁVEL
4	Como usuário VIP, gostaria de poder visualizar as postagens para assinantes do site.	Exibir as postagens exclusivas para os usuários VIPs que estejam autenticados no sistema.	FEITO	[GF]
5	Como usuário VIP, gostaria de poder publicar comentários nas postagens.	Armazenar e exibir os comentários dos usuários autenticados no sistema.	FEITO	[RR]
6	Como administrador do site gostaria de poder, alterar as configurações do site, como título, números de posts por página e a disponibilidade do site.	Armarzenas as informações de configuração do site, exibir mecanismos para manipulação das configurações	FEITO	[RR]

Sprint backlog 2

IBL	USER STORY	CRITERIOS DE ACEITAÇÃO	STATUS (POR FAZER, EM ANDAMENTO, FEITO)	RESPONSÁVEL
7	Como administrador do site, gostaria de gerenciar as publicações de posts.	Gerenciar as publicações dos posts	FEITO	[GF]
8	Como adminstrador do site gostaria de	Gerenciar as páginas	FEITO	[RR]

	poder gerenciar as páginas estáticas do site.	estáticas do site.		
9	Como administrador do site, gostaria de poder gerenciar as categorias das publicações.	Gerenciar as categorias das publicações do site.	FEITO	[RF]

Sprint backlog 3

IBL	USER STORY	CRITERIOS DE ACEITAÇÃO	STATUS (POR FAZER, EM ANDAMENTO, FEITO)	RESPONSÁVEL
10	Como desenvolvedor que utiliza o CMS, gostaria de acessar as informações do CMS no formato JSON para que eu possa integrar as minhas aplicações.	Disponibilizar as informações publicadas em formato JSON por meio de URLs.	FEITO	[GF]
11	Como desenvolvedor que utiliza o CMS, gostaria de poder integrar os conteúdos publicados aos meus templates de páginas web.	Disponibilizar as informações publicadas em formato JSON por meio de URLs.	FEITO	[RR]

Sprint backlog 4

APÊNDICE E – Ficha de perfil dos participantes

Caracterização de Participante

Nome:	
Email:	Curso:

1 - Qual seu nível de ensino?

- Técnico
 Graduação
 Pós-graduação

() Concluído () Cursando () Abandonei

2 - Sua atuação?

- Acadêmico
 Mercado profissional

3 - Já cursou ou tem conhecimento de alguma dessas disciplinas?

Engenharia de Software	() sim () não
Análise e Projeto de Sistemas	() sim () não
Programação OO	() sim () não
Programação WEB	() sim () não
Teste de Software	() sim () não
Gerência de Projetos	() sim () não

4 - Já participou do desenvolvimento de algum projeto de software?

- () sim
() não

5 - Já gerenciou o desenvolvimento de algum projeto de software?

- () sim
() não

OBS: Esse questionário tem como objetivo caracterizar os participantes desse experimento e auxiliar na compreensão dos resultados desta pesquisa, então é importante que seja preenchido de maneira fiel.



TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
"JOSÉ ALBANO DE MACEDO"

Identificação do Tipo de Documento

- Tese
 Dissertação
 Monografia
 Artigo

Eu, JOSAFÁ MARTINS DOS SANTOS,
autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de
02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar,
gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação
MÓDULO SCRUM PARA UM SISTEMA DE GESTÃO
DE PROJETOS
de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título
de divulgação da produção científica gerada pela Universidade.

Picos-PI 7 de JULHO de 2017

Josafá Martins dos Santos
Assinatura
Josafá Martins dos Santos
Assinatura