

## **SISTEMA DE AUXÍLIO À INICIANTE NA GESTÃO DE PROJETOS DE SOFTWARE - SAIGPS**

Jonathan Oliveira

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil  
oliveira.jo@hotmail.com

Giovani Facchini

Professor Orientador

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil  
facchini@gmail.com

### **Resumo**

Este artigo apresenta os resultados de uma pesquisa experimental que teve por finalidade desenvolver um sistema Web, denominado SAIGPS. Seu objetivo principal o auxílio a iniciantes e discentes do desenvolvimento de software, aplicação de engenharia e gestão de projetos. Visando os resultados obtidos podemos ver o quanto é importante o domínio dos métodos de desenvolvimento de software, tal como sua gestão, por isso damos uma atenção especial a introdução destes ao iniciantes desta área.

**Palavras-chave:** projetos, projetos de software, metodologias tradicionais, metodologias ágeis, gestão de projetos.

### **System for Beginners in Software Project Management - SAIGPS**

#### ***Abstract***

*This article presents the results of an experimental study that aimed to develop a Web system, called SAIGPS. Main goal of SAIGPS is aid beginners and students of software development, application engineering and project management. Aiming at the results we can see how important the field of software development methods, such as management, so we give special attention to the introduction of these beginners in this area.*

**Key-Works:** *projects, software projects, traditional methodologies, agility methodologies, projects management.*

## 1. INTRODUÇÃO

É comum que as pessoas tenham muitos objetivos. Para alcançar esses objetivos necessitamos de planejamento, elaboração e execução, ou seja, importantes fases de um projeto. Seja uma viagem, cursos, uma graduação, aquisição, entre outros, é necessário empregar fases de um projeto, para que os objetivos sejam alcançados com menor possibilidade de falhas. Se não forem aplicados métodos a possibilidade de que esses projetos fracassem aumenta, por isso a importância de se conhecer a teoria básica e processos de um projeto (PARREIRAS; BAX, 2003).

Este trabalho tem por finalidade pesquisar, analisar, sintetizar e descrever os principais métodos de gerência de projetos de software e implementar um sistema denominado Sistema de Auxílio a Iniciantes na Gestão de Projetos de Software – SAIGPS, proposto para trabalho de conclusão de curso de Bacharel em Sistemas de Informação pela FACCAT. O sistema irá proporcionar aos usuários uma comparação entre um método tradicional e um ágil para que se possa identificar qual processo é o mais adequado para o tipo de projeto de software, com objetivo de introduzir os iniciantes e discentes com mais facilidade e de forma prematura no processo de gestão de projetos.

Nos projetos existem muitos erros e falhas que podem levar 31% dos projetos a serem cancelados antes do seu término (PMI, 2009). Erros como falta de recursos com as competências adequadas, de gerente de projeto experiente, de metodologia nos processos, falta de tempo para definir o escopo, de comunicação com líderes e clientes podem ser considerados motivos para a causa dos cancelamentos de projetos. (SALLES Jr et al., 2010).

Estas falhas citadas acima diminuem significativamente quando se utilizam métodos de gestão de projetos. Como forma de facilitar a implementação de métodos e processos nos projetos, torna-se importante a utilização de uma ferramenta para auxiliar na tomada de decisão. Diante de tantas ferramentas de Gerência de Projetos (GP) presentes no mercado, se mostra importante uma ferramenta intuitiva, e de fácil compreensão destas boas práticas de gestão e desenvolvimento de software, visando auxílio a discentes ou iniciantes desta área (XAVIER et al., 2005). O autor relata, a título de exemplo, a importância do entendimento referentes a estes assuntos para o melhor aproveitamento nas disciplinas do curso de Sistemas de Informação como Gerência de Projetos, Análise de Sistemas II, Sistemas Distribuídos. A partir deste cenário visualizou-se uma problemática que poderia ser explorada para criação de uma ferramenta ou sistema intuitiva de fácil utilização. Este sistema é de grande valia para a introdução as principais metodologias de gestão de projetos de software.

Com uma boa prática de GP, introduzindo alguma metodologia, se pode ter melhores resultados para o produto final, também identificar os pontos de falha mais comuns em um projeto de software. Com o objetivo de ajudar iniciantes e discentes de gestão de projetos, este trabalho propõe uma ferramenta para que as pessoas entendam e utilizem metodologias de engenharia de software, documentações relevantes, aplicação de análise, ou seja, boas práticas de gestão de projetos de software (XAVIER et al., 2005).

De forma que se pretende que seja intuitiva e prática, o iniciante ou discente visualiza alguns dos principais métodos e pode decidir qual será melhor para o projeto a ser trabalhado. Após essa escolha, o usuário pode visualizar e criar o seu projeto de acordo com a metodologia escolhida, fazendo com que o usuário tenha uma visão mais clara nos projetos implementados futuramente, sem ambigüidades e de forma a conhecer qual as características e padrões se moldam melhor a tipos específicos de projetos (SALLES Jr. et al., 2010).

Exemplificando o argumento exposto no parágrafo anterior, se o trabalho a ser executado é um projeto rápido com forte participação do cliente, possui um melhor desenvolvimento baseado em uma metodologia ágil, como *Scrum*. Caso o projeto seja desenvolvido em maior tempo, com requisitos estáticos é melhor utilizar o método tradicional, como Espiral (SOARES, 2004).

O sistema proposto denominado “Sistema de Auxílio a Iniciantes na Gestão de Projetos de Software” tem por finalidade facilitar o aprendizado e prática das principais metodologias de gestão de projetos de software aos iniciantes e discentes desta área.

## **2. REFERENCIAL TEÓRICO**

### **2.1 Processos**

Um processo é uma seqüência lógica ou conjunto seqüencial e particular de ações ou operações que transforma elementos em produtos ou serviços (XAVIER, 2005 e BOUER, CARVALHO, 2005). Um processo pode ter os mais variados propósitos, como inventar, transformar, produzir, controlar, manter e usar produtos ou sistemas.

Na administração, um processo é o conjunto de atividades realizadas na geração de resultados para o cliente, desde o início do pedido até a entrega do produto. De acordo com a sincronia entre insumos, atividades, infra estrutura e referências necessárias para adicionar valor para o ser humano (OLIVEIRA, 2009).

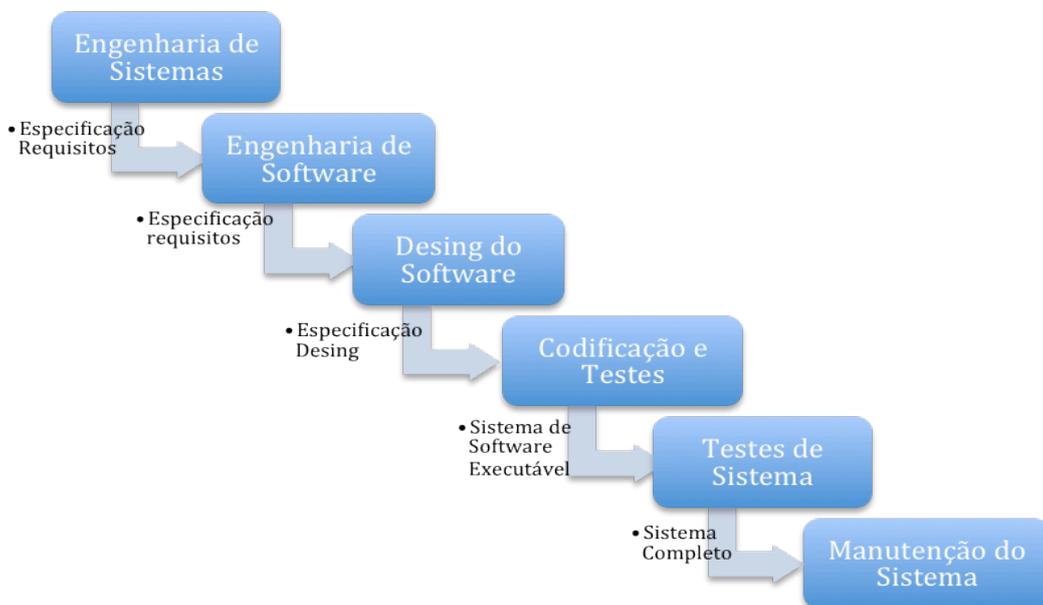
Nos sistemas operacionais (SO), o processo é um módulo executável único, que corre concorrentemente com outros módulos executáveis. Exemplificando, em um ambiente

multitarefa, como o Unix que suporta diferentes processos, um processador de texto, um tocador de áudio e um navegador, são processos separados que podem rodar concomitantemente (OLIVEIRA; CARISSINI e TOSCANI, 2010).

Já nos Banco de Dados (BD), o processo é um conjunto de operações lógicas e matemáticas feitas sobre os dados, de acordo com instruções programadas, com objetivo de obter a informação desejada (SILBERSCHATZ; KORTH e SUDARSHAN, 1999).

Na engenharia de software processo tem como objetivo entregar um produto de software de maneira eficiente, previsível e que corresponda às necessidades de negócio, através de uma sequência coerente de prática. Estas práticas englobam as atividades de especificação, projeto, implementação, testes e caracterizam-se pela interação de ferramentas, pessoas e métodos (PRESSMAN, 2006). A Figura 1 apresenta um exemplo de um conjunto de processos da engenharia de software conhecido como método, na qual visa obter como um produto final, um *software* ou sistema, modelo denominado método tradicional cascata ou modelo cascata.

Figura 1: Modelo Cascata



Fonte: Adaptado de Pressman (2009)

## 2.2 Projetos

Um projeto é um esforço temporário para criar um produto, serviço ou resultado exclusivo, ou seja, é um esforço temporário composto de objetivos claros, medidas de resultados, datas de início e término que atendam aos requisitos negociados e explícitos das partes interessadas (XABIER et al., 2005).

Um desenvolvedor de sistemas consegue interpretar as necessidades de seu cliente através de análise e recursos propostos pela engenharia de software, encarregados de transformar os resultados da análise de requisitos em um conjunto de documentos capazes de serem interpretados no momento da codificação. Também se torna importante seguir um cronograma para que se possa estipular e cumprir metas, como prazos, custos, escopo, etc., desta forma vê-se que através de diversos processos podemos realizar o desenvolvimento de um sistema, ou seja, características de um projeto de software (PRESSMAN, 2009).

### **2.3 O Ciclo de Vida de um Projeto**

O conjunto de fases de um projeto é chamado de “ciclo de vida do projeto”. De um modo geral, as fases dos projetos apresentam as seguintes características: por (XAVIER, 2005 e BOUER; CARVALHO, 2005).

a) Preparação: definir o destino do projeto (Termo de Abertura ou Project Charter) questões como: Onde queremos chegar? Qual o objetivo (s)? Qual o destino final? Um encontro das mentes envolvidas no projeto para que todos possam entendê-lo como um todo;

b) Planejamento: Definir metas, O que melhorar? Quando vai ser feito? Quem vai participar? Onde serão os locais de trabalho? Definição de métodos (como será executado, ações e prazos, registros importantes, estrutura formal, recursos);

c) Implementação: implementar todos os métodos e métricas estipulado na fase anterior. Nem sempre se consegue seguir o caminho estipulado no planejamento, pois o mundo e pessoas mudam frequentemente, mas é de extrema importância ajustar os caminhos com foco no objetivo final;

d) Encerramento: Ao terminar o projeto, arquivar documentações e experiências para que se possa aprender com os acertos e erros para projetos futuros.

### **2.4 Gestão de Projetos**

Na gestão de projetos ou gerência de projetos (GP) se desenvolve um trabalho de planejamento, execução e controle de projetos. O GP é o ato de gerir, executar a gerência, envolvendo aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos. Sua aplicação ao longo de todo o esforço para execução de um projeto permite a avaliação do desempenho, o aprendizado contínuo e a antecipação do desempenho futuro com razoável confiabilidade (XAVIER et al., 2005).

Gerir um projeto de software é considerado diferente da gestão habitual. É difícil o planejamento antecipado, pois todo o processo de desenvolvimento envolve atividades criativas, como: arte, trabalho inovador, agilidade, ou seja, está mais ligado as ciências humanas do que exatas, diferente dos projetos tradicionais na qual vemos trabalhos relativamente fixos (MARTINS, 2004).

Projetos tradicionais possuem uma estrutura pré-definida onde o produto final é dividido em componentes, como especificações, plantas, subsistemas, dentre outros. Um processo normalmente em fases seqüenciais, com produto final definido no escopo do projeto. No desenvolvimento de software essa abordagem é normalmente falha, pois no início dos projetos de desenvolvimento sabe-se muito pouco sobre o produto final. Durante a execução dos projetos de software geralmente o contexto (escopo, requisitos, etc.) sofre mudanças durante seus ciclos de vida (SALLES, 2005).

Devido aos motivos expressados acima é de extrema importância o uso de dinâmica e iteratividade na gestão de projetos de software, com o constante monitoramentos dos requisitos, de forma a identificar as mudanças que surgem, para poder aplicar em um ciclo futuro ou até mesmo para correção no atual (XAVIER, 2010).

Gerenciamento de riscos em projetos de software é muito importante, devido a não estabilidade de requisitos e não aplicação de métodos adequados de desenvolvimento. Os riscos são muitos nos projetos de software e acarretam conseqüências vitais para essas empresas de desenvolvimento (SALLES Jr. et al., 2010).

## **2.5 Metodologias de Gestão de Projetos**

O alcance da excelência em gerenciamento de projetos não é possível sem um processo repetitivo que possa ser utilizado em cada projeto. Esse processo repetitivo é a metodologia de gerenciamento de projetos (KERNER, 2001).

Uma metodologia é um conjunto de orientações e princípios que podem ser adaptados e aplicados em uma situação específica. Em ambiente de projetos essa orientação é uma lista de coisas a fazer. Uma metodologia pode também ter uma abordagem específica, modelos, formulários e também *check lists*, usados durante o ciclo de vida do projeto. Desta forma, uma metodologia de gerenciamento de projetos é um conjunto de processos, métodos e ferramentas para o alcance dos objetivos do projeto (CHARVAT, 2003).

Existem muitos métodos de gestão de projetos, com várias ferramentas com enfoque específico. O importante é conhecer a forma de execução de cada método, tal como aspecto

pertinente a grande maioria destes modelos, por isso a importância de se estar atento a essência dos métodos de engenharia, pois a não aplicação destes modelos acabam sendo erros comuns presentes nos projetos de desenvolvimento de software (PRESSMAN, 2006).

### 2.5.1 Metodologias Tradicionais

Metodologias tradicionais, também conhecidas como orientadas a documentação ou pesadas, surgiram em um contexto totalmente diferente do nosso contexto tradicional. O desenvolvimento era baseado em *Mainframe* (computador de grande porte, dedicado normalmente a o processamento de um volume grande de dados) e terminais burros, com um custo muito elevado no desenvolvimento de sistemas, por isso o software ou sistema era todo planejado e documentado antes de ser implementado (DIAS, 2011).

### 2.5.2 Espiral

O Modelo em Espiral tenta evitar a necessidade de detalhamento e de definição dos requisitos finais do software no início do projeto. Contudo, difere do desenvolvimento iterativo por priorizar os desenvolvimentos, levando em considerações o teor risco e não as funcionalidades propriamente ditas, possuindo característica principal o formalismo e a preparação de uma extensa documentação (SOARES, 2004).

O processo de desenvolvimento é representado como uma espiral, ao invés de uma seqüência de atividades. O modelo define cinco quadrantes, nos quais as atividades (gerenciais ou técnicas) de um projeto são executadas durante o ciclo na espiral, as cito: (I) “Comunicação com Cliente” determinação dos objetivos, alternativas e restrições. (II) “Planejamento” Os objetivos específicos para a etapa são identificados e alternativas para realizar os objetivos e restrições são encontradas; (III) “Análise Risco” análise das alternativas e identificação ou resolução de riscos. Os riscos principais são identificados, analisados e buscam-se meios para reduzi-los; (IV) “Engenharia, Construção e Entrega” desenvolvimento e validação de versão corrente do produto. Um modelo apropriado para o desenvolvimento é escolhido, o qual pode ser qualquer um dos modelos do ciclo de vida. (V) “Avaliação do Cliente” o projeto é revisto e o próximo ciclo da espiral é planejado (VASCONCELO et al., 2006). A Figura 2 mostra a representação visual do modelo espiral.

Figura 2: Modelo Espiral



Fonte: Adaptado de Pressman (2009)

A espiral representa o curso do projeto, onde, a cada volta um novo produto é construído (ex. Documentos de requisitos, modelo de projeto, implementação, etc.), cada volta na espiral representa uma etapa no processo de desenvolvimento. Não há etapas fixas como especificação ou projeto (conteúdo de um volta na espiral é escolhido dependendo do produto requerido pela etapa), os riscos são avaliados explicitamente e resolvidos ao longo do processo (VASCONCELOS et al., 2006).

### 2.7.3 Metodologias Ágeis

Segundo Dias (2011) Metodologias Ágeis tornaram-se populares em 2001, onde especialistas em desenvolvimento de software apresentaram alguns métodos, como o *Scrum*, *Extreme Programming* dentre outros. Devido a estes fatos foi então criado o Manifesto Ágil. Para Soares (2004) o “Manifesto Ágil” não rejeita as documentações ou processos encontrados anteriormente, apenas é feito um refinamento, mostrando que eles tem importância secundária quando comparado com indivíduos e interação. O software é executado com a colaboração e as respostas rápidas a mudanças e alterações. Conceitos que se aproximam mais das pequenas e médias organizações, assim como o contexto atual de desenvolvimento que se modifica e precisa ser moldado frequentemente.

### 2.7.4 Scrum

Um exemplo de método ágil é o *Scrum*, na qual apresenta uma abordagem empírica, que aplica algumas idéias da teoria de controle de processos industriais. Para o desenvolvimento de software, o *Scrum* trás conceitos de flexibilidade, adaptabilidade e

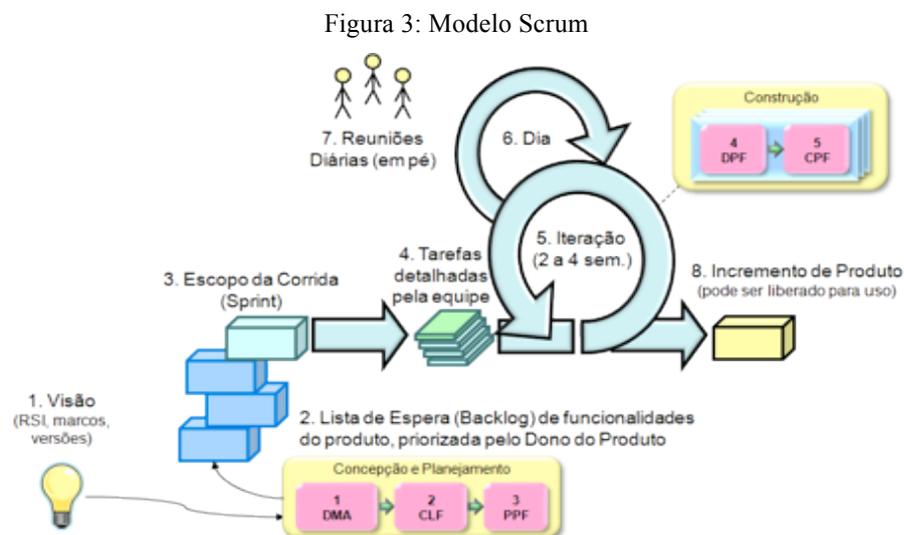
produtividade, traduzindo uma forma de trabalho dos membros da equipe para produzir sistemas de forma flexível em um ambiente com constantes mudanças (SOARES, 2004).

O foco principal do *Scrum* é que seus seguidores ou desenvolvedores envolvam muitas variáveis técnicas e do ambiente, como requisitos, recursos e tecnologia, que podem mudar durante o processo. Isso torna o processo de desenvolvimento imprevisível e complexo, requerendo flexibilidade para acompanhar as mudanças, sendo desejado como resultado um software que é realmente útil para o cliente (PICHLER, 2011).

Algumas diferenças diretas que podem ser vistas entre o *Scrum* e as metodologias tradicionais, são: indivíduos e interações ao invés de processos e ferramentas, Software executável ao invés de documentação colaboração do cliente ao invés de negociação de contratos, respostas rápidas a mudanças ao invés de seguir planos (SABBAGH, 2013).

O método *Scrum* não rejeita os processos e ferramentas, a documentação, a negociação de contrato ou o planejamento, mas simplesmente mostra que eles têm importância secundária quando comparado com os indivíduos e interações, com o software estar executável, com a colaboração do cliente e as respostas rápidas a mudanças e alterações, ou seja, o *Scrum* é uma forma adequada para as pequenas e médias organizações trabalhem e respondem a mudanças (SOARES, 2004).

Na Figura 3 está representado visualmente o funcionamento do método *Scrum*, onde os componentes de software, como requisitos, atividades, documentos, dentre outros, são incrementados diante uma iterações denominada “Sprint”, representada por um período de tempo. São realizadas reuniões com a equipe e cliente nas *Sprints* para estabelecer os artefatos e prioridades referente aos componentes de software ou sistema (SABBAGH, 2013).



Fonte: Adaptado de PICHLER (2011).

## 2.8 Linguagem de Modelagem Unificada – UML

*Unified Modeling Language* ou Linguagem de Modelagem Unificada é uma linguagem visual padrão para documentar e modelar projetos de software, podendo ser utilizada em todas as aplicações. Nos últimos anos se tornou padrão de modelagem adotada internacionalmente pela engenharia de software (MARTINS, 2004; GUEDES, 2011 e SBROCCO, 2011).

O desenvolvimento de sistemas ou softwares sem nenhuma documentação, diagrama ou esboço, gera vários problemas, como: tornar difícil a comunicação entre equipe; quando o programador sai da empresa a mesma sente uma carência de documentação para entender o que já está codificado e estruturado, consecutivamente o suporte ou continuidade do projeto acaba prejudicado; elementos como distribuição física das classes em um sistema distribuído na Web não são bem entendidos em projetos apenas com códigos, etc. (MARTINS, 2004).

Com o objetivo de tratar os problemas expostos no parágrafo anterior é importante frisar que o principal objetivos do UML esta relacionado a auxiliar os envolvidos no desenvolvimento de software a definirem as características do sistema, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica do seus processos e até mesmo a necessidade física em relação ao equipamento sobre o qual o sistema deverá ser implantado (GUEDES, 2011 e SBROCCO, 2011).

Qualquer sistema deve ser modelado antes de se iniciar sua implementação, porque os sistemas de informação freqüentemente costumam ter a propriedade de “crescer”, aumentando em tamanho, complexidade e abrangência (GUEDES, 2011).

O objetivo de haver tantos diagramas em *Unified Modeling Language* (UML) é oferecer diferentes visões do sistema a ser modelado ou como o sistema fosse modelado em camadas. Alguns diagramas enfocam o sistema de forma mais geral, apresentando uma visão de forma externa (Diagrama de Casos de Uso), outros enfocam o sistema como uma camada mais profunda, tendo uma visualização mais técnica ou características mais específicas. A utilização de diversos diagramas permite que falhas sejam descobertas, diminuindo a possibilidade da ocorrência de erros futuros (GUEDES, 2011).

Ferramenta para auxílio no desenvolvimento de diagramas UML são softwares ou sistemas que de alguma forma colaboram para a execução de uma ou mais atividades realizadas durante o processo de engenharia de software (GUEDES, 2011 e SBROCCO, 2011).

Por exemplo, o Astah é um software muito popular encontrado em *astah.net* e oferece uma edição para a comunidade, ou seja, uma versão da ferramenta que pode ser baixada gratuitamente de sua página, lógico que esta versão não suporta todos os serviços. No entanto para quem deseja praticar a UML através de seus diagramas, a edição para a comunidade é uma boa alternativa, apresenta um ambiente amigável e de fácil compreensão (CHANGE VISION, 2011).

### **3. METODOLOGIA**

Tendo em vista a problemática apresentada, levando em conta a análise dos contextos do desenvolvimento e gestão de software, foi desenvolvido um sistema Web denominado SAIGPS (Sistema de Auxílio ao Iniciante da Gestão de Projetos de Software) para auxílio na gestão e desenvolvimento de projetos de softwares focado nos iniciantes e discentes desta área.

O presente projeto foi desenvolvido baseado em uma metodologia tradicional de desenvolvimento de softwares, denominado Método Espiral, visando que o projeto tem um tempo considerável e sua melhor aplicação em uma metodologia tradicional, documentada, incrementando cada volta do espiral (Documentos, modelos, protótipos, codificação, testes, implementação, etc.) com uma funcionalidade do sistema (VASCONCELO et al., 2006).

#### **3.1 Análise**

Visando a problemática apresentada no presente artigo, o sistema SAIGPS começou a ser desenvolvido através da análise do sistema, de forma a entender melhor o contexto, obtendo todas as informações relevantes e documentando-as. A Linguagem UML foi utilizada para criar os diagramas através da ferramenta Astah, de forma a auxiliar no desenvolvimento, documento de planejamento do desenvolvimento, e os processos presentes no desenvolvimento do sistema SAIGPS.

##### **3.1.1 Requisitos**

Segundo Pressman (2006) a análise de requisitos é uma etapa importante, que permite obter informações que resultam na especificação das características operacionais da aplicação. Levando em consideração esta afirmação este trabalho iniciou com a análise de requisitos.

Lista de funções adquiridas na análise de requisitos:

1. Evidentes: apresentar uma visão geral referente a projetos, das metodologias (Espiral e Scrum), Gerir Projetos de Desenvolvimento de Sistemas de acordo com um dos métodos propostos, Gerir usuários (equipe do projeto) e autenticações no sistema.
2. Escondidas: Nos projetos devemos ter informações sobre as etapas e metodologias escolhidas para gestão e criação de um projeto, tal como, breve informação sobre andamento, escalonamento de tarefas e apresentação da equipe do projeto.
3. Opcional: Gerir gráficos, gerir mensagens de avisos ou alertas por e-mail.

Lista dos atributos aplicadas ao sistema SAIPGS.

1. Referencial (projetos, metodologias e UML): de forma a apresentar um referencial dando uma visão geral de projetos, projetos de software, gestão de projetos, engenharia de software com os métodos tradicional (Espiral) e Ágil (*Scrum*) e UML;
2. Projeto (e seus componentes): gerir projetos (incluir, alterar, listar e excluir) e seus atributos como: termo de abertura, documento de requisitos, documento de riscos, gestão de tarefas e custos presentes no projeto, etc.;
3. Usuário: gerir usuários (incluir, alterar, listar e excluir) e seus atributos de conta, ou seja, manter a conta do usuário de forma a vincular com seus projetos ou projetos na qual o mesmo faz parte.

Lista dos requisitos funcionais aplicado ao sistema SAIPGS.

1. Autenticar Usuário: O sistema deve permitir a autenticação dos usuários, sempre levando em conta as regras de validação;
2. Gerir Usuário: O sistema deve permitir a gestão da conta do usuários tal como: cadastro, alteração, apresentação e exclusão;
3. Gerir Projetos O sistema deve permitir a gestão de projetos e seus atributos, tal como: cadastro, alteração, apresentação, exclusão e apresentação de relatorias de forma gráfica;
4. Gerir *Home Page*: O sistema deve permitir a gestão da *home pega*, tal como: cadastro, alteração, apresentação, exclusão e apresentação do referencial teórico para auxílio dos usuários no desenvolvimento dos seus projetos;
5. Apresentar Estrutura Projeto (gráfico de tarefas): Projetos O sistema deve apresentar relatórios em formato gráfico referente a cada projeto. Gráficos de período e de porcentagem de tarefas (à executar, executando e prontas);

Lista dos requisitos não funcionais aplicado ao sistema SAIPGS.

1. Ter Segurança: O sistema deve manter integridade consistência de danos durante os processos de gestão de projetos e usuários;

2. Possuir Usabilidade: O sistema terá interface amigável e de fácil acesso a recursos, sua utilização será de forma intuitiva, explicativa de maneira a não precisar de treinamentos.

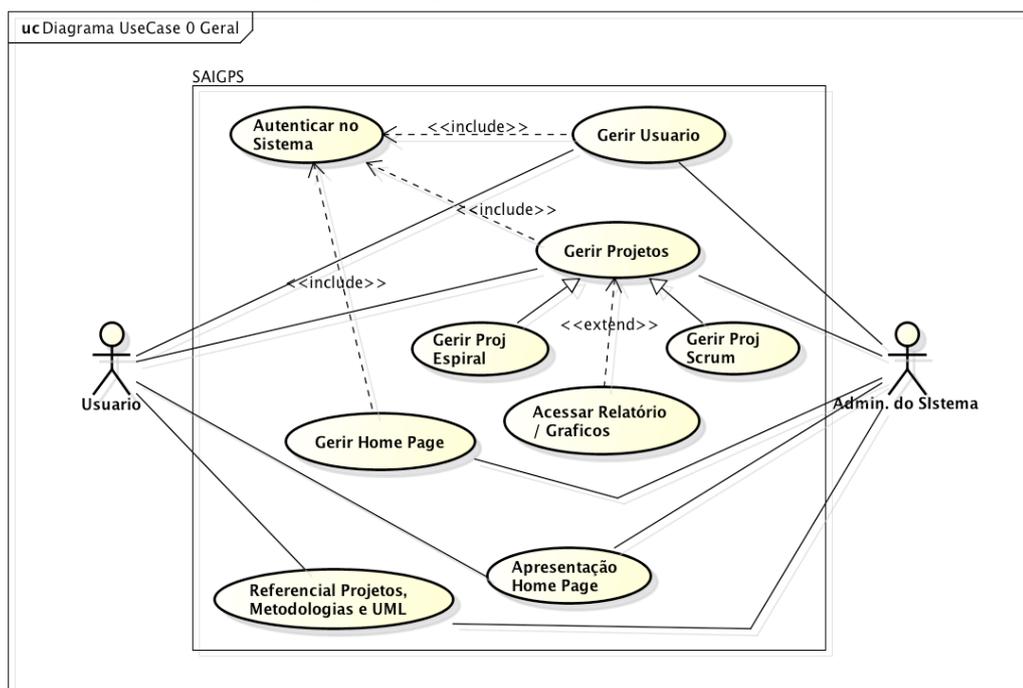
3. Ser Confiável: atualizações do sistema e melhorias serão feitas em horário planejados, a ser definido com antecedência, sempre com tempo programado, de forma a informar a possível manutenção previamente aos usuários. Sistema deve ser robusto o suficiente para poder gerir todos os projetos de forma ágil;

4. Utilizar Padrões: O padrão utilizado no projeto é o MVC que visa separar a lógica de negócios da lógica de apresentação, permitindo o desenvolvimento, testes e manutenção isolado de ambos, forte ênfase em modularização, contendo modulo banco, modulo Web, modulo projetos, etc.

### 3.1.2 Casos de Uso

O objetivo dos casos de uso é passar uma idéia geral do sistema, apresentando os atores envolvidos, serviços, ou seja, as funcionalidades que sistema disponibilizará aos atores. A Figura 4 mostra o diagrama geral de casos de uso (UML) do sistema SAIGPS.

Figura 4: Diagrama Geral de Casos de Uso



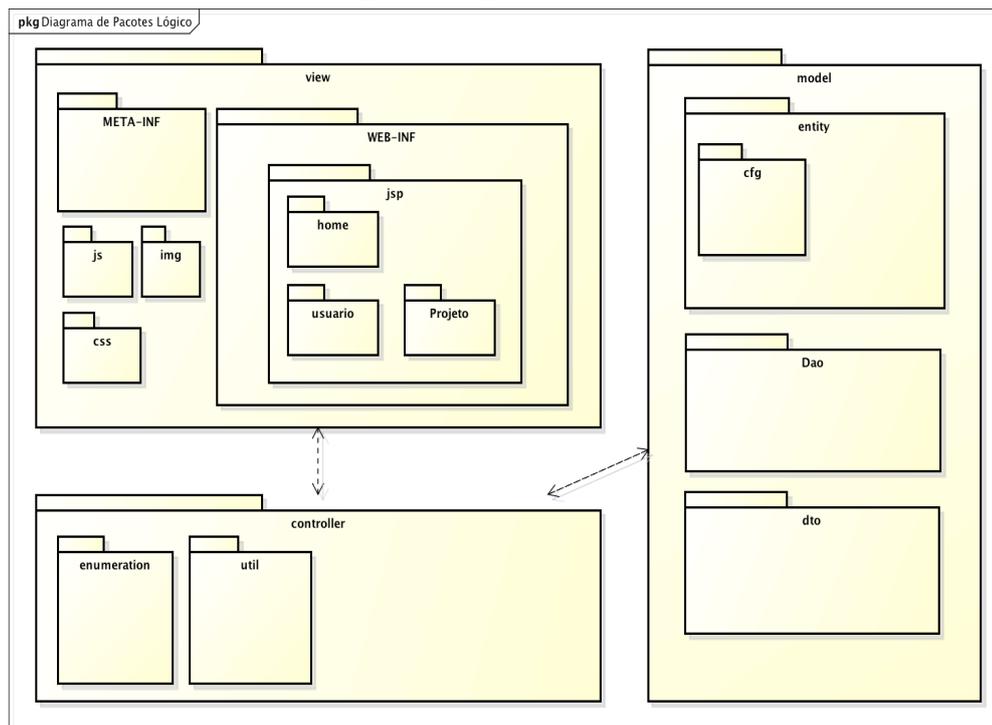
powered by Astah

Fonte – Autor

### 3.2 Modelagem do sistema

O sistema desenvolvido foi baseado em um modelo de arquitetura de software denominado *Model View Controller* (MVC) “modelo visão controle”, buscando isolar a lógica da interação do usuário. O *Model* “modelo” consiste nos dados presentes no sistema, regras de negócios, lógica e funções. Uma *View* “visão” é conhecida como a saída ou representação dos dados, como uma tabela ou um diagrama, é possível ter várias visões dos mesmos dados, um exemplo o *Navegador*. Já no *Controller* “Controle” fica toda a parte de tratamento de entrada, convertendo-a em comandos para o modelo ou visão. A idéia principal do MVC é a poder deixar seu código reusável e com separação de conceitos. Podemos ver melhor a estrutura MVC na Figura 5, denominado diagrama de pacotes (UML) do sistema SAIGPS.

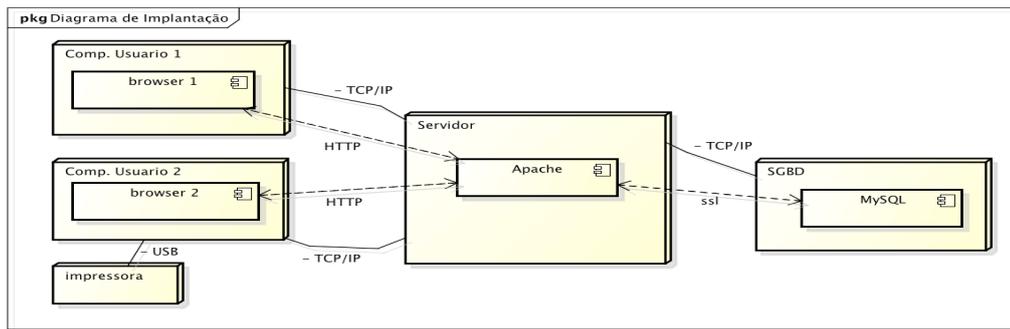
Figura 5: Diagrama de Pacotes



Fonte – Autor

A Figura 6 apresenta a arquitetura do sistema SAIGPS, uma arquitetura Web, que faz uso de um servidor Apache (APACHE, 2013a), tornando o sistema multiplataforma, ou seja, para utilizar o presente sistema, o usuário apenas precisa de um navegador e acesso a Web, não sendo necessário nenhum outro aplicativo. SAIGPS pode ser acessado de qualquer dispositivo móvel que possua um navegador e acesso a Web, por ser um sistema responsivo (se adequa a qualquer tipo ou tamanho de tela).

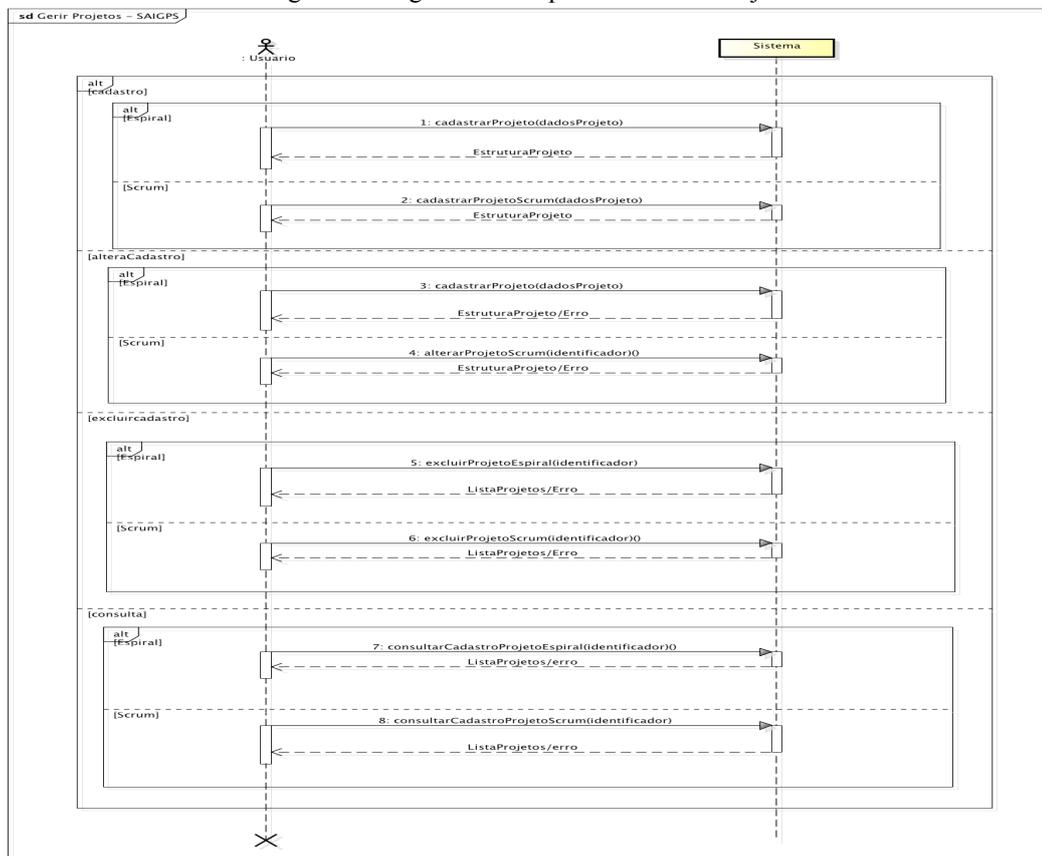
Figura 6: Diagrama de Implantação



Fonte – Autor

Tendo em vista que o sistema SAIGPS possui uma complexidade referente a projetos, surge a necessidade de se entender melhor os processos envolvidos, tal como sua seqüência e mensagens passadas entre objetos. Para descrever a maneira como os grupos de objetos colaboram em algum comportamento ao longo do tempo utilizou-se o diagrama de seqüência (UML). Em síntese este diagrama é usado para representar interações entre objetos de um cenário, realizadas através de operações ou métodos. A Figura 7 mostra o diagrama de seqüência do processo gerir projetos do sistema de SAIGPS.

Figura 7: Diagrama de Seqüência – Gerir Projeto



Fonte – Autor



SAIGPS foi desenvolvido com auxílio do *Integrated Development Environment* (IDE) ou ambiente integrado de desenvolvimento denominado NetBeans, na qual é gratuito e de código aberto para desenvolvedores de software. O IDE é executado em muitas plataformas, como Windows, Linux, Solaris e MacOS. O NetBeans IDE oferece aos desenvolvedores ferramentas necessárias para criar aplicativos profissionais de desktop, empresariais, móveis e Web conforme utilizado no sistema SAIGPS (NETBENS, 2013a).

### 3.3.1 Tecnologias Utilizadas

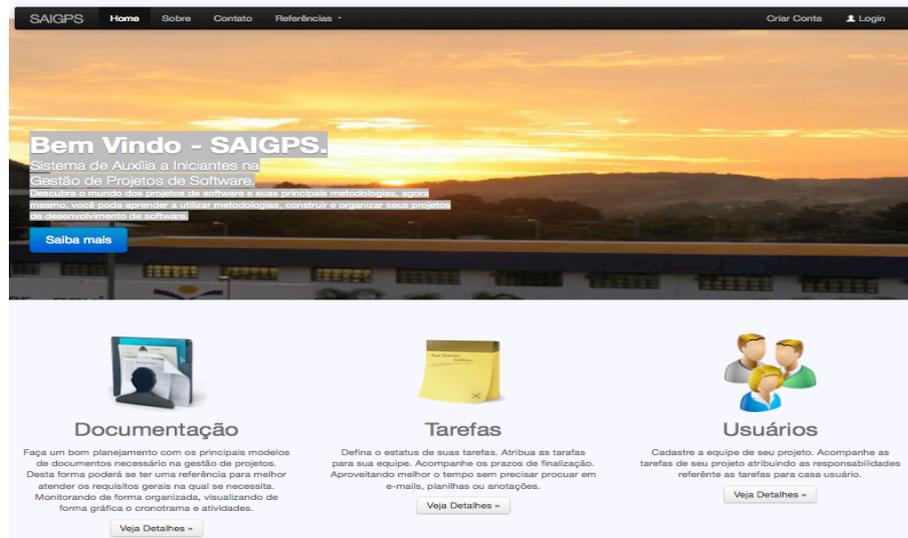
1. SAIGPS gera páginas dinâmicas pois é uma aplicação Java Web, onde contém linguagem de marcação como HTML, XML, dentre outras, e conteúdo dinâmico. É composto por componentes Web, como: JPS, Servlets e JavaBeans para modificar e armazenar dados temporariamente, interagir com SGBDs e Web Services de forma a processar o conteúdo como resposta às solicitações do cliente (ORACLE CORPORATION, 2013a).

2. Como forma de agilizar no momento de codificação de queries SQL e no respectivo código JDBC, SAIGPS usou uma ferramenta para auxiliar nesta tarefa, na qual são conhecidas como ferramentas de mapeamento objeto-relacional (ORM). Hibernate é uma ferramenta ORM *open source* (código livre) e é a líder de mercado (REDHAT, 2013a).

3. SAIGPS é desenvolvido com auxílio da ferramenta Bitbucket, um controlador de versão distribuído, um serviço de hospedagem de projetos controlados através do Mercurial. *Bitbucket* utilizando em conjunto com o GIT um sistema de gerenciamento de código fonte (ATLASSIAN, 2013a).

4. SAIGPS utiliza um framework chamado Bootstrap para o desenvolvimento de *front-end* (*view*). Em linhas gerais, o Twitter Bootstrap é uma ótima opção para desenvolvedores que buscam um framework completo para *front-end* contendo recursos CSS, HTML5, Javascript para desenvolvimento rápido, vale ressaltar que o mesmo possui recursos responsivos, ou seja, possibilitando a apresentação do lado cliente (*view*), redimensionando as páginas para que possa ser apresentada em qualquer dispositivo que possua um navegador com acesso a Web, como: *tablets, smartphones, notebooks, desktops*, etc. (TWITTER, 2013a). A Figura 9 e 10 apresentam a *home page* de SAIGPS, na qual faz uso dos recursos presentes no Bootstrap, como CSS e HTML em seu visual.

Figura 9: SAIGPS Home Page Superior



Fonte – Autor

Figura 10: SAIGPS Home Page Inferior

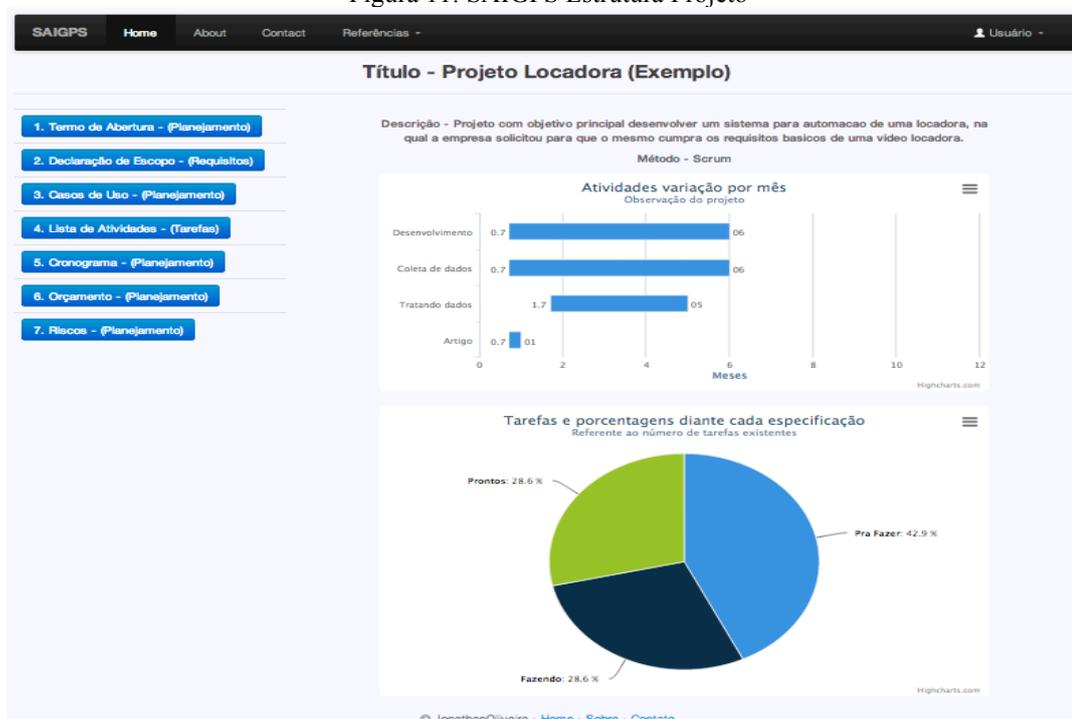


Fonte – Autor

5. A figura 11 apresenta a tela de estrutura de projeto com seus informativos, passos e gráficos (exemplificando o uso do Highcharts). Após a criação de um projeto o usuário é remetido a tela “estrutura do projeto”, onde vemos o título e uma breve descrição, logo abaixo no lado esquerdo podemos ver o menu principal do projeto, onde vê-se a documentação e gestão de tarefas, em passos. SAIGPS mostra também dois gráficos com porcentagem de tarefas concluídas e um com diagrama visual do tempo do projeto. Para construção destes gráficos é utilizado uma biblioteca em *HTML5* e *JavaScript* denominado Highcharts, que oferece uma maneira fácil para adicionar gráficos interativos em seu site ou

aplicação Web. Funciona em todos os navegadores modernos, incluindo o Safari presente no iPhone e Internet Explorer a partir da versão 6. (HIGHSOFT, 2013a).

Figura 11: SAIGPS Estrutura Projeto



Fonte – Autor

6. Para organizar as tarefas e eventos do desenvolvimento de SAIGPS, é utilizado o auxílio da ferramenta de gestão denominada como Trello. Uma ferramenta que ajuda no gerenciamento de projetos, sua organização pode ser feita de maneira a englobar o período inteiro de um projeto. Foi inspirado na metodologia *Scrum*, processo de desenvolvimento para gerenciar projetos de software, de forma a auxiliar na organização das atividades do desenvolvimento de software, tal como dividir as tarefas em equipe e período de tempo (CREK, 2013a).

#### 4. RESULTADOS

A pesquisa e desenvolvimento tem como resultado um sistema Web denominado SAIGPS, um sistema de criação e gestão de projetos de softwares, tendo como foco principal iniciantes e discentes.

Através da URL do sistema SAIGPS os usuários podem acessar a *home page* do sistema, onde podem obter um referencial teórico sobre projetos e metodologias de desenvolvimento (tradicional e ágil). SAIGPS traz também em sua *home page* informações sobre como utilizar o sistema nos itens de menu: documentação, tarefas, usuários, passo à

passo de como desenvolver um projeto no SAIGPS e traz detalhes do seu desenvolvimento no item de menu “Tecnologias”.

O usuário de SAIGPS pode acessar um referencial teórico sobre projetos, metodologias de engenharia de software (Espiral e *Scrum*), gestão e UML. Desta forma se o usuário precisar de informações para embasar seu conhecimento ou projeto, o mesmo poderá acessá-lo a qualquer momento em um menu que ficará sempre presente em todas as telas de sistema. A Figura mostra 12 mostra o menu do referencial teórico e conteúdo parcial deste item.

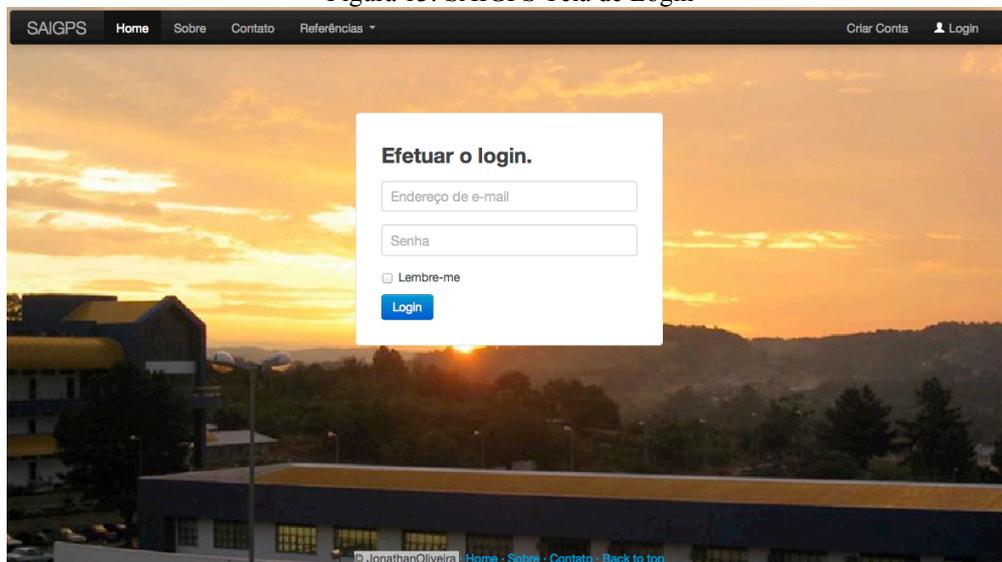
Figura 12: SAIGPS Referencial



Fonte – Autor

Após preencher o formulário de cadastro no sistema SAIGPS, o usuário é direcionado para a tela de *login*, onde deve informar suas credenciais. Logo após terá acesso a recursos como: meus projetos e minha conta. A Figura 13 apresenta a tela de *login* de SAIGPS.

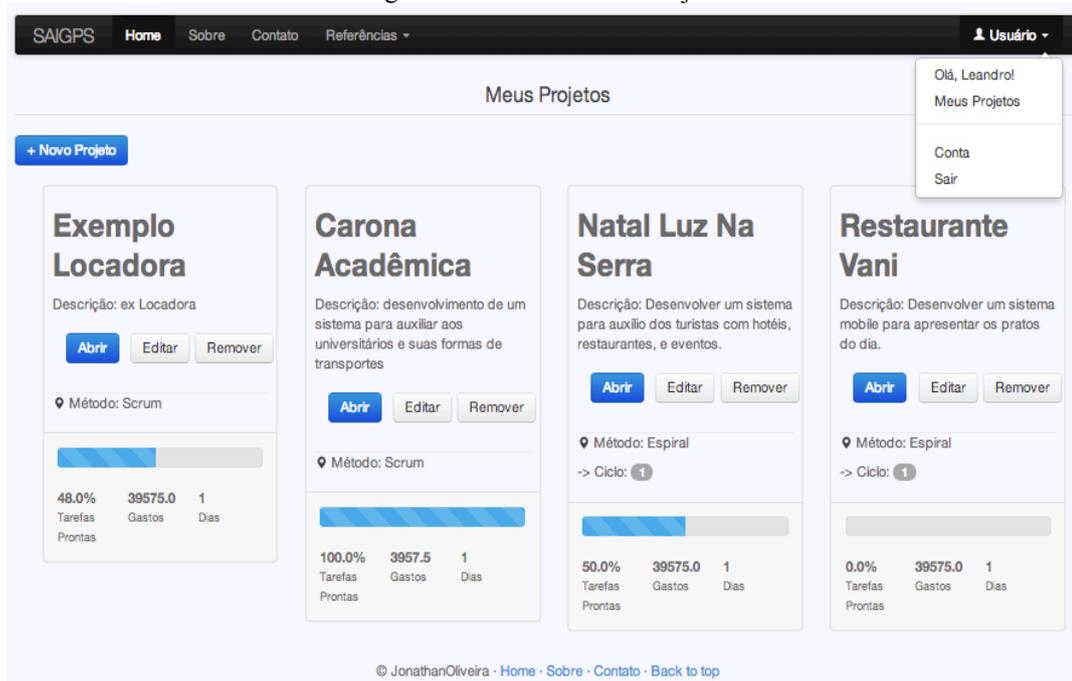
Figura 13: SAIGPS Tela de Login



Fonte – Autor

O item “meus projetos” é um local destinado às práticas dos métodos Espiral e *Scrum* e gestão, destinado a seguir um passo a passo, exemplificado as etapas de projetos de software baseado nas metodologias tradicional ou ágil (Espiral e *Scrum*). A Figura 14 apresenta o menu principal no superior da página e no lado direito o sub-menu principal do usuário, onde vemos as opções para acessar sua conta e listagem de projetos (Meus Projetos, apresentada na imagem 14).

Figura 14: SAIGPS Meus Projetos



Fonte – Autor

Após clicar no menu “Meus Projetos”, a página descrita na figura 14 aparece. Ao clicar no botão “Novo Projeto” o usuário entra com 3 informações básicas, primeira: nome do projeto, segunda: uma breve descrição e terceira o método que deseja aplicar a este projeto. Logo após estes passos o usuário será remetido a estrutura do projeto, com um os passos e documentos exemplificados para o desenvolvimento deste projeto (menu a esquerda). Desta forma a interação com os principais documentos presentes será muito mais fácil e intuitiva visando que o mesmo só precisa selecionar no menu um documento (apresentado em uma ordem) e alterar o exemplo que foi dado. A Figura 15 apresenta o documento “Termo de Abertura” com o menu lateral, onde podemos ver os passos para obter os principais documentos do projeto ou a gestão das atividades ou tarefas baseado no método Espiral ou *Scrum*.

Figura 15: SAIGPS Meus Projetos

Fonte – Autor

A gestão de tarefas proposta por SAIGPS ajuda a controlar a equipe através da criação e gestão destas atividades, ao clicar em “Nova Tarefa” o usuário é remetido para o formulário de cadastro da tarefa onde é preciso entrar com dados como: nome, descrição, quem é responsável, data de entrega e se está nas modalidades “à executar”, “executando” ou “concluídas”. Desta forma se pode ter o melhor controle das atividades relacionadas ao projeto e equipe participante do projeto. A Figura 16 mostra a tela de Tarefas ou Lista de Atividades de um projeto executado pelo método *Scrum*.

Figura 16: SAIGPS Tarefas Método *Scrum*

Fonte – Autor

A Figura 17 apresenta a tela de Tarefas ou Lista de Atividades de um projeto executado pelo método Espiral, apresentando um referencial ao ciclo que o projeto se encontra. Após a conclusão das atividades relacionadas a um ciclo de projeto o mesmo pode ser incrementado e se inicia uma nova fase de projeto.

Figura 17: SAIGPS Tarefas Método Espiral

SAIGPS Home Sobre Contato Referências ▾ Usuário ▾

### Tarefas

Projeto Criado em: 24/10/2013  
Método Espiral  
Ciclo -> 1

Código	Tarefa	Responsável	Situacao	Excluir
65	Ex.: Criar Entidade	Ex.: Exemplo Nome	à executar	Excluir
66	Criar Endidade Cliente	Jonathan	concluída	Excluir
71	Criar Endidade Item	Daniel	concluída	Excluir
75	Criar Repositório	Leandro	executando	Excluir

© JonathanOliveira · Home · Sobre · Contato · Back to top

Fonte – Autor

#### 4.1 Trabalhos Relacionados

SAIGPS possui um foco específico, é destinado a prática de projetos de softwares, focado nos iniciantes e discentes desta área. Existe sistemas que possuem algumas similaridades com SAIGPS, porem nenhum com mesmo foco.

O sistema de gerencia de projetos DotProject possui similaridade na documentação dos projetos, na fase de planejamento. Já o sistema Trello possui similaridade com a gestão de tarefas, baseada no método *Scrum*.

Estes e outros sistemas possuem similaridades, porem trazem um foco diferente. SAIGPS traz exemplos de documentações e gestão de tarefas em cada projeto criado, além de um referencial teórico, fazendo com que o usuário possa moldar o exemplo de acordo com suas necessidades de projeto.

#### 5. CONCLUSÃO

Este trabalho de pesquisa e desenvolvimento consiste na criação de um sistema Web destinado a gestão de projetos de softwares, denominado SAIGPS. Tem como foco principal os iniciantes e discentes do desenvolvimento de sistemas ou softwares. Através de uma

interface Web os usuários podem utilizar este sistema sem precisar muita prática ou conhecimento com métodos de engenharia de software ou maior conhecimento em gestão de projetos.

Tendo em vista os pontos apresentados no parágrafo anterior e os resultados obtidos no presente trabalho, SAIGPS auxilia na criação e gestão de projetos destinado aos iniciantes e discentes desta área. Por apresentar todo um referencial e um passo a passo de como se executar um projeto e sua gestão, trazendo exemplos de documentação e gestão de recursos e listas de tarefas presente em um projeto de desenvolvimento de software.

SAIGPS cumpre os requisitos definidos no seu planejamento, ou seja, aplicar as etapas de gestão e desenvolvimento de software, trazendo um referencial teórico e aplicação prática da metodologia Espiral e *Scrum* na gestão das tarefas do projeto. Porém no seu desenvolvimento notou-se que a aplicação de novas metodologias e recursos em uma versão futura é interessante, pois amplia mais o conhecimento e aplicabilidade destes métodos em todo o desenvolvimento de conhecimento desta área.

### **Histórico de Versões**

- a) Versão 01: versão onde foram apresentadas a intenção do artigo e um arcabouço;
- b) Versão 02: implementações de “Referências” e aprimorando a apresentação na Versão 01 e iniciando a aplicação do “Referencial Teórico”;
- c) Versão 03: correções do texto anterior e complemento do referencial teórico.
- d) Versão 04: desenvolvimento da justificativa;
- e) Versão 05: unificação da intenção do artigo com a justificativa para formar a introdução;
- f) Versão 06: complemento referencial teórico e correções na introdução;
- g) Versão 07: implementação metodologia, análise, desenvolvimento e correções no referencial teórico;
- h) Versão 08: complemento do desenvolvimento e correções;
- i) Versão 09: resultados e conclusão.
- j) Versão 10: incremento resultados, conclusão e correções.

### **REFERÊNCIAS**

ATLASSIAN. **Bitbucket**. 2013. Disponível em: <<http://bitbucket.org/>>. Acesso em: 08 setembro. 2013.

APACHE SOFTWARE FOUNDATION. **Apache HTTPD Server Project**. 2013. Disponível em: <<http://httpd.apache.org>>. Acesso em: 08 agosto. 2013.

BANIN, Sérgio Luiz. **Simulação de Gerencia de Projetos: uma ferramenta de ensino e aprendizado**. São Paulo: teses.usp.br, 2011.

BOUER, Ruy; CARVALHO, Marley Monteiro. **Metodologia Singular de Gestão de Projetos: condição suficiente para a maturidade em gestão de projetos?** Revista Produção, 2005.

CAELUM. **VRaptor3**. 2013. Disponível em: <<http://caelum.com.br/pt/docs>>. Acesso em: 18 setembro. 2013.

CHANGE VISION. **Astah Community**. 2013. Disponível em: <<http://astah.net>>. Acesso em: 22 setembro. 2013.

CHARVAT, Jason. **Project Management Methodologies**. Ed.1. New Jersey: John Wiley & Sons, 2003.

CREK, **Trello**. 2013. Disponível em: <<http://trello.com/docs/>>. Acesso em: 08 setembro. 2013.

DIAS, Marisas Villas Bôas. **Um Novo Enfoque Para o Gerenciamento de Projetos de Desenvolvimento de Software**. São Paulo: teses.usp.br, 2011.

FOWLER, Martin. **UML Essencial – Um breve guia para a linguagem-padrão de modelagem de objetos**. Ed.3. Porto Alegre: Bookman, 2005.

GUEDES, Gilleanes Thorwald Araujo. **UML 2 Uma Abordagem Prática**. Ed. 2. São Paulo: Editora Novatec, 2011.

HIGHSOFT. **Highcharts Js**. 2013. Disponível em: <<http://www.highcharts.com/docs/>>. Acesso em: 10 outubro. 2013.

KERZNER, Harold. **Project Management: A system approach to planning scheduling and controlling**. Ed.7. New Jersey: John Wiley & Sons, 2001.

MARTINS, José Carlos Cordeiro. **Gerenciando Projetos de desenvolvimento de Software com PMI, RUP e UML**. Rio de Janeiro: Brasport, 2004.

ORACLE CORPORATION. **NetBeans**. 2013. Disponível em: <<http://netbeans.org>>. Acessado em: 25. agosto. 2013.

ORACLE CORPORATION. **Java EE Platform Specification**. 2013. Disponível em: <<http://java.net/projects/javaee-spec/>>. Acessado em: 11. agosto. 2013.

OLIVEIRA, Djalma de Pinto Rebouças. **Introdução À Administração**. Ed. Compacta. São Paulo: Alhas, 2009.

OLIVEIRA, Rômulo Silva; CARISSIMI, Alexandre da Silva; TOSCANI, Simão Sirineo. **Sistemas Operacionais**. Ed.4, Porto Alegre: Bookman: Instituto de Informática da UFRGS, 2010.

PARREIRAS, Fernando S.; BAX, Marcello P.; **A Gestão de Conteúdo no Apoio à Engenharia de Software**. São Paulo: SBGC – Sociedade Brasileira de Gestão de Conhecimento, 2003.

PICHLER, Roman. **Gestão de Produtos com Scrum – Implementando Métodos Ágeis na Criação e Desenvolvimento de Produtos**. Ed.1. Rio de Janeiro: Campus, 2011.

PRESSMAN, Roger. **Software Engineering - A Practitioner's Approach**. Ed.6. Mc Graw Hill, 2006.

REDHAT. **Hibernate**. 2013. Disponível em: <<http://www.hibernate.org/docs>>. Acessado em: 04. setembro. 2013.

SABBAGH, Rafael. **Scrum – Gestão Ágil para Projetos de Sucesso**. Ed.1. Casa do Código, São Paulo, 2013.

SALLES JR, Carlos Alberto Corrêa; SOLER, Alonso Mazini; VALLE, José Angelo Santos; RABECHINI JR, Roque. **Gerenciamento de Riscos em Projetos**. Ed. 2. Rio de Janeiro: Editora FGV, 2010.

SBROCCO, José Henrique Teixeira de Carvalho. **UML 2.3: Teoria e Prática**. São Paulo, Editora Érica, 2011.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de Dados**. Ed.6. São Paulo: Makron Books, 1999.

SOARES, Michel dos Santos. **Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software**. Gigante: Revista Eletrônica de Sistemas de Informação, 2004.

SOARES, Michel dos Santos. **Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software**. Gigante: Journal of Computer Science, 2004.

TONSIG, Sérgio Luiz. **Engenharia de Software Análise e Projeto de Sistemas**. Ed. 2. Rio de Janeiro: Editora Ciência Moderna Ltda., 2008.

TWITTER, **Bootstrap**. 2013. Disponível em: <<http://getbootstrap.com>>. Acessado em: 26. setembro. 2013.

XAVIER, Carlos Magno da Silva. **Gerenciamento de Projetos: Como definir e controlar o escopo do projeto**. São Paulo: Saraiva, 2005.

XAVIER, Carlos Magno da Silva; VIVACQUA, Flávio Ribeiro; MACEDO, Otualp Sarmiento; XAVIER, Luiz Fernando da Silva. **Metodologia de Gerenciamento de Projetos Methodware: Abordagem prática de como iniciar, planejar, executar, controlar e fechar projetos**. Rio de Janeiro: Brasport, 2005.