

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LEONARDO FIEWSKI SOARES

**FERRAMENTA PARA GERENCIAMENTO DE DESENVOLVIMENTO
DE SISTEMAS**

TRABALHO DE DIPLOMAÇÃO

CORNÉLIO PROCÓPIO

2014

LEONARDO FIEWSKI SOARES

**FERRAMENTA PARA GERENCIAMENTO DE DESENVOLVIMENTO
DE SISTEMAS**

Trabalho de conclusão de curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de tecnólogo.

Orientador: Prof. Fabio Fernandes da Rocha
Vicente

CORNÉLIO PROCÓPIO

2014

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, por me dar à luz e a sabedoria do dia-a-dia, e por sempre estar presente em minha vida.

Ao meu orientador Prof. Fábio Vicente, por ter paciência e por ter me ajudado nesses momentos finais do curso.

Aos meus amigos e sócios, Carlos Henrique Prado Sousa e Diogo Cezar Teixeira Batista, por ter me orientado e ajudado no decorrer do trabalho.

A minha mãe, Laís, por estar sempre me mostrando o caminho para se construir uma vida feliz.

A minha namorada, Mayara Caroline, por ter entendido os meus momentos de ausência, durante o desenvolvimento desse trabalho.

A todas as pessoas, que me ajudaram e me apoiaram, diretamente ou indiretamente, para que esse trabalho ocorresse.

RESUMO

SOARES, Leonardo Fiewski. **Ferramenta para gerenciamento de desenvolvimento de sistemas**. 2014. 94 p.. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) – Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

Este trabalho apresenta o processo de análise e desenvolvimento de uma ferramenta para a gestão de desenvolvimento de sistemas. Utilizando o *framework* Scrum como base no processo de desenvolvimento. O Kanban foi usado para controlar e monitorar tarefas. O objetivo deste trabalho foi construir uma ferramenta que gerencia o processo de desenvolvimento em empresas que utilizam o Scrum e Kanban em seu desenvolvimento. No processo de desenvolvimento desta ferramenta foi utilizada estudos casos que geraram os resultados esperados.

Palavras-chave: Scrum. Kanban. Controle de desenvolvimento. Controle de Processo.

ABSTRACT

SOARES, Leonardo Fiewski. **Tools for managing systems development**, In 2014. 94 p.. Trabalho de Conclusão de Curso (Technology Analysis and Systems Development) – Federal Technological University of Paraná. Cornélio Procópio, 2014.

This work presents the process of analysis and development of a tool for managing of developments systems. The System uses the Scrum framework as basis of development process. The Kanban was used to control and monitor tasks. The objective this work was build a tool that manages the process of developing on enterprises that uses Scrum and Kanban in their development. In the process of developments of this tool, was used studies cases which generated the expected results.

Keywords: Scrum. Kanban. Control of development. Process Control.

LISTA DE FIGURAS

Figura 1 – Exemplo de quadro de kanban.....	12
Figura 2 - Kanban base.....	13
Figura 3 - Kanban com limite.	14
Figura 4 - Fluxo do Scrum.	16
Figura 5 - Diagrama de Atividades sobre as Tarefas do Projeto.....	24
Figura 6 - Modelo de Tarefa.....	25
Figura 7 - Quadro de Tarefas.....	26
Figura 8 - Tarefas em desenvolvimento.	26
Figura 9 - Tarefas finalizadas.	27
Figura 10 - Registro de tempo trabalhado.....	28
Figura 11 - Gerenciamento de Requisitos.....	29
Figura 12- Tags da Tarefa.....	30
Figura 13 - "Testado OK".....	30
Figura 14 – Padrão MVC	32
Figura 15 - Diagrama Entidade Relacionamento.....	83
Figura 16 - Exemplo tarefas agrupadas.....	87

LISTA DE TABELAS

Tabela 1 - Comparação entre softwares semelhantes.....	22
Tabela 2 - Cronograma Proposto.....	36
Tabela 3 - Cronograma Realizado.....	38

SUMÁRIO

1	INTRODUÇÃO	9
2	REFERENCIAL TEÓRICO	10
2.1	KANBAN	11
2.2	SCRUM	14
2.3	METODOLOGIA	16
3	LEVANTAMENTO PARA O DESENVOLVIMENTO	18
4	FERRAMENTA KONTROLLER	20
4.1	OBJETIVOS DA FERRAMENTA	20
4.2	ANÁLISE	21
4.2.1	Requisitos	21
4.2.2	Análise	23
4.2.2.1	Diagrama de atividades	23
4.3	DESENVOLVIMENTO	24
4.4	FUNCIONALIDADES	27
4.5	TESTES	30
4.6	CRONOGRAMA	31
5	ARQUITETURA	32
6	CONCLUSÕES	33
6.1	RESULTADOS	33
	REFERÊNCIAS	34

1 INTRODUÇÃO

Para desenvolver um sistema de qualidade, uma empresa de desenvolvimento de sistemas necessita de um mínimo de controle de processo. Por falta de planejamento ou controle das tarefas a serem realizadas, um sistema pode sair do seu escopo original ou então ter os prazos não cumprida.

Empresa que necessita controlar seu processo de desenvolvimento, pode-se recorrer à métodos já estudados e difundidos no meio da engenharia de *software* mas, nem sempre precisa seguir à risca um modelo pronto, pode-se adotar um *framework* e adapta-lo a realidade da empresa.

Segundo RAUL WAZLAWICK (2012), “Não existe um processo único e ideal para o desenvolvimento de software, porque cada sistema tem suas particularidades. Porém, usualmente, podem-se agrupar os sistemas de acordo com certas características e então definir modelos de processo mais adequados a elas.”.

Esse trabalho tem por finalidade descrever toda a construção de uma ferramenta que visa o gerenciamento do processo de desenvolvimento de sistemas, utilizando os princípios do método de controle de tarefas do Kanban e o controle de fluxo no desenvolvimento de sistemas, descritos no Scrum. Essa ferramenta foi construída utilizando tecnologias *Web* e seu escopo foi baseado no dia-a-dia da empresa Kreea Sistemas Web, para ajudá-la a controlar o seu processo de desenvolvimento.

A Kreea Sistemas Web é uma empresa situada na cidade de Cornélio Procópio, Paraná, que atua no desenvolvimento de sistemas para *Web*. Fundada em 2013, por dois alunos e um ex-aluno da UTFPR – Universidade Federal do Paraná, a empresa por ser jovem, ainda é pequena, mas em seu primeiro semestre de vida já apresentou ótimos resultados. Contando com os três sócios e dois estagiários, a empresa atua em três ramos, sendo o primeiro a venda de produtos, entre elas o sistema Kompre – para comércio eletrônico e o Kaixa – controle de caixa para bares e padarias, o segundo ramo é o desenvolvimento de *Web Sites*, terceiro e último Fábrica de Software, voltado para *Web*, onde desenvolve sistema sob demanda, através de contratos de prestação de serviços.

Devido ao desligamento da empresa Forlogic Software, o escopo deste trabalho teve que ser alterado, para atender as necessidades da empresa Kreea Sistemas Web, mas a sua temática de controle de processo, não foi alterado.

Esse trabalho foi organizado em 6 capítulos: no primeiro capítulo é descrito a introdução; o segundo capítulo refere-se a revisão bibliográfica dos temas abordados; o quarto relata sobre como foi realizado o levantamento para o desenvolvimento; a quinta seção descreve qual foi a arquitetura adotada para o projeto e no sexto capítulo os resultados obtidos com o sistema.

2 REFERENCIAL TEÓRICO

Neste capítulo encontram-se os conceitos que foram utilizados no desenvolvimento desse trabalho. Os conceitos que serão apresentados a seguir pertencem a dois métodos: Kanban e Scrum.

Na seção de Kanban encontram-se explicações sobre como surgiu o método Kanban, e um exemplo de como pode ser aplicado no desenvolvimento de sistemas. Na seção do Scrum serão apresentados conceitos sobre o *framework*, como ele é utilizado, a estrutura para o seu funcionamento e um conceito de como utiliza-lo juntamente com o Kanban. Ainda nesta seção será abordada a metodologia de desenvolvimento para este trabalho, onde serão usados os conceitos do Scrum e do Kanban.

2.1 KANBAN

O Kanban, é um método de gestão de processo, para controle de tarefas a serem desenvolvidas. Este método teve seu primeiro desenvolvimento dentro da indústria Toyota Motors, onde ficou conhecido como Toyota Way, que era sustentado por um sistema *Just In Time*, significando que, em um processo de fluxo, as partes corretas e necessárias à montagem alcançam a linha de montagem no momento em que são necessários e somente na quantidade necessária, tendo assim menos desperdício de materiais e mais foco na montagem dos veículos.

A palavra Kanban, vem da língua japonesa, que significa cartão, que é uma das ferramentas do Just In Time. A paternidade deste método se dá ao Taiichi Ohno, vice-presidente da Toyota Motors, o mesmo afirma que o “Kanban nasceu da necessidade de desenvolver um sistema de fabricação de automóveis de muitos tipos diferentes em pequenos número na mesma linha e montagem”.

Em seu conceito, as tarefas a serem realizadas por uma pessoa é separado e escritas nos cartões (kanban) e coladas em um quadro na onde existem três divisões, com os seguintes nomes, *To Do*, onde ficam localizadas as tarefas que precisam ser desenvolvidas. *Doing* são as tarefas que estão sendo executadas e *Done* que são as atividades que já foram executadas e finalizadas.

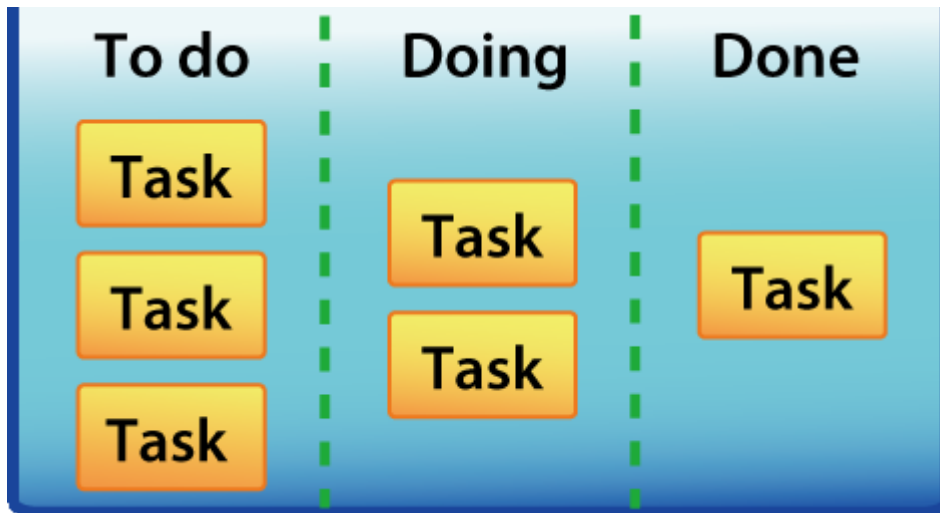


Figura 1 – Exemplo de quadro de kanban.

Fonte: <http://trichord.change-vision.com/en/basicguide.html> Acesso em: 20 set. 2013

A finalidade das divisões é para um acesso rápido, de como está o andamento do processo de trabalho que está sendo executado, para que assim possam ser tomadas decisões rápidas. Por exemplo, se existem muitas tarefas em *To Do* (á fazer) e poucas em *Done* (terminadas), e se passou muito tempo que foi iniciado o trabalho, os gerenciadores tomarão alguma atitude necessária, que poderia ser, alocação de algum outro recurso para ajudar nas tarefas.

Por ser de fácil entendimento e de simples modificação, é possível fazer alterações no método Kanban, para suprir as necessidades organizacionais de uma empresa. Ansari e Modarress (1995), Vernyi e Vinas (2005), Argenta e Oliveira (2001) fazem menção e expõem a utilização do sistema Kanban original com apenas uma modificação: a substituição dos sinalizadores físicos por sinalizadores eletrônicos, ou seja, trocar o uso manual proposto no sistema original pelo uso virtual de sinais que representam, ora ordens de produção, ora autorização para a transferência de materiais, tendo uma denominação própria sistema e-kanban.

Dentro do desenvolvimento de software, o Kanban, por se tratar de um método adaptativo, pode ser utilizados da forma que se encaixar melhor na realidade de cada empresa. A seguir será dado um exemplo básico de funcionamento do Kanban no desenvolvimento.

Neste exemplo será utilizado um quadro com cinco colunas, representando os estados das tarefas, sendo a primeira o *Backlog*, onde estarão as tarefas a serem desenvolvidas, segunda coluna Levantamento, que será feito o levantamento das funcionalidades, terceira coluna será a de Desenvolvimento, que será o desenvolvimento da tarefa levantada, a quarta coluna será a de Testes, onde estarão as tarefas já implementadas para serem testadas e a quinta e última coluna Pronto, tarefas que já foram executadas e testadas.

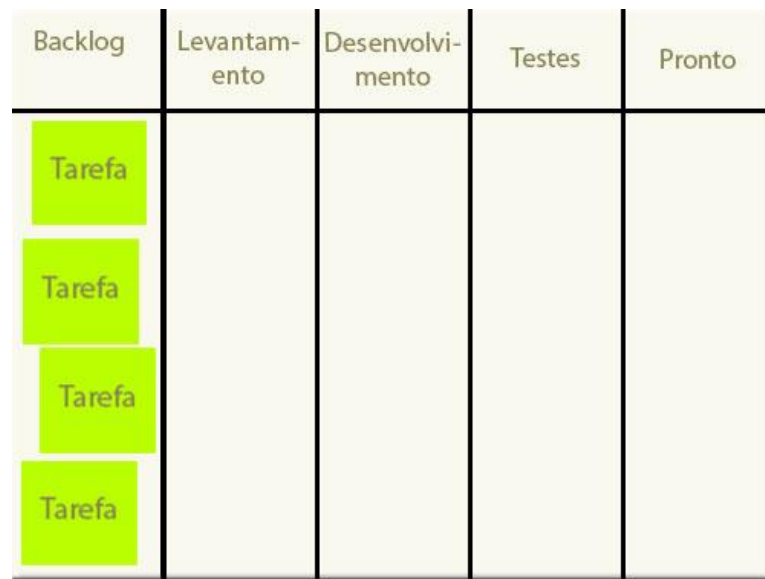


Figura 2 - Kanban base.

Em cada coluna deverá ser limitado o número de tarefas, para que não haja um gargalo. Esse limite se faz necessário para que as tarefas possam ser distribuídas igualmente. Existe também a possibilidade dos membros da equipe se destacar mais em um estado da tarefa, por exemplo, o estado de Levantamento, no qual esta pessoa se destaca no levantamento de informações para a tarefa, sendo assim ela iria fazer todos os levantamentos e ficaria ociosa logo em seguida, enquanto o próximo estado estará sobrecarregado.



Figura 3 - Kanban com limite.

2.2 SCRUM

Scrum é um *framework* desenvolvido por Ken Schwaber e Jeff Sutherland (Guia do Scrum, 2011), para desenvolver e manter produtos complexos e adaptativos. Segundo o “Guia do Scrum”, este framework está sendo utilizado para o desenvolvimento de produtos complexos desde o início de 1990. Ele não é um processo nem uma técnica para construir produto, mas sim uma ferramenta no qual pode-se empregar processos ou técnicas, sendo assim altamente adaptável a diversas realidades.

“Scrum é fundamentado nas teorias empíricas de controle de processo, ou empirismo. O empirismo afirma que o conhecimento vem da experiência e de tomada de decisões baseadas no que é conhecido. O Scrum emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos” (Guia do Scrum, 2011).

Para que o Scrum funcione corretamente, em sua documentação, a equipe responsável pelo produto trabalha unida, mas cada pessoa desenvolvendo o seu papel específico. Tendo isso em mente o *framework* separa a equipe em três funções:

- *Product Owner*: Dono do produto, responsável por maximizar o valor do produto e do trabalho da equipe de Desenvolvimento, cabe a esta pessoa expressar com clareza os itens a serem desenvolvidos, separar os itens de

maior relevância para que entrem em desenvolvimento logo no início do projeto.

- Equipe de Desenvolvimento: Consiste em profissionais que realizam o trabalho de entregar a versão pronta do produto. A equipe é auto organizada, ninguém impõe a forma de trabalhar. As organizações das tarefas são feitas, em consenso com a equipe, a própria equipe de desenvolvimento delegará o que cada participante irá desenvolver. Em relação ao tamanho da equipe, segundo o Guia do Scrum (2011), recomenda-se de três a nove integrantes. Se tiver menos que, três diminui as interações e resulta em um menor ganho a produtividade e se tiver mais do que nove, a coordenação dos membros é bastante elevada.
- *Scrum Master*: Responsável por garantir que Scrum flua e seja aplicado. É o membro de toda a equipe que irá mostrar a teoria, à prática e as regras para os demais. O *Scrum Master* trabalha tanto para a equipe de desenvolvimento, treinando-os em se auto gerenciar, tanto para o *Scrum Owner*, encontrando técnicas para o gerenciamento do produto em desenvolvimento.

Eventos prescritos são utilizados no Scrum para criar um rotina e diminuir as necessidades de reuniões não definidas. O principal evento do Scrum é a *Sprint*, que é um *time-box*, evento que possui uma duração máxima de um mês onde será desenvolvido o produto. Não é possível começar duas ou mais *Sprints* do mesmo produto. Durante a *Sprint*, não se deve fazer mudanças que afetem produto final e a equipe de desenvolvimento deve ser permanente.

Para que uma *Sprint* não saia da linha é feito uma reunião de *Sprint*, para ser definido dois tópicos: quais funcionalidades serão desenvolvidas e como cada funcionalidade será desenvolvida.

Outro evento presente no Scrum é a Reunião Diária, este *time-boxed* é executado durante a *Sprint*, onde todos os dias, logo no início da jornada de trabalho, é feita uma reunião de no máximo quinze minutos de duração, onde cada participante estará relatando os itens desenvolvidos e as dificuldades ou impedimentos encontrados. No final da reunião, se possível, deve-se repassar rapidamente o andamento do projeto como um todo e delegar novas tarefas a equipe.

No final da *Sprint* é feito uma reunião geral, para que os integrantes façam um levantamento do que pode ser melhorado ou mantido nas próximas *Sprints*,

também para verificar quais funcionalidades serão construídas nas próximas versões. A figura 4 exemplifica todo o processo de uma *Sprint*.

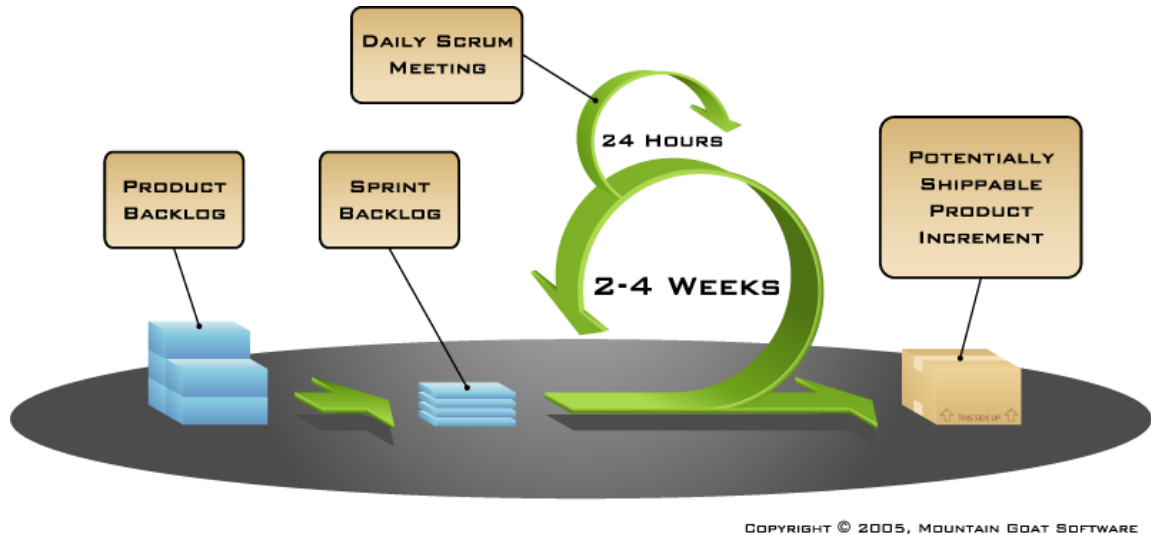


Figura 4 - Fluxo do Scrum.

Fonte:

http://epf.eclipse.org/wikis/scrumpt/Scrum/guidances/supportingmaterials/scrum_overview_610E45C2.html Acesso em: 15 out. 2013.

Tendo em mente o funcionamento do Scrum e do Kanban, é possível realizar uma união e chegar a um processo de desenvolvimento ágil e fácil, tendo o Scrum para organizar as interações do projeto e o Kanban para organizar as tarefas a serem realizadas em cada *Sprint*.

2.3 METODOLOGIA

Nas próximas seções, será relatada a metodologia de desenvolvida adotada para gerenciar o projeto.

Como metodologia de desenvolvimento da ferramenta, intitulada Kontroller, foi escolhido a junção do *framework* Scrum e do método de gestão de processos Kanban. Essa junção é utilizada atualmente na empresa Kreea Sistemas Web e foi utilizada como base para o desenvolvimento desse trabalho.

Como descrito na seção 2.2, o Scrum é um *framework* adaptável as necessidades da empresa que o utiliza. Com isso, o Scrum será utilizado para controlar o processo de desenvolvimento da seguinte forma.

Todo projeto terá um responsável (*Product Owner*), que irá controlar o andamento do projeto pelas especificações dos requisitos que serão desenvolvidos. Esse responsável também fará o papel do *Scrum Master*, e caberá a ele orientar a equipe na utilização do Scrum.

Um produto a ser desenvolvido poderá ter um ou vários projetos, mas eles não devem acontecer ao mesmo tempo (*Sprint*). Em cada projeto, cabe ao responsável do projeto selecionar os integrantes da equipe de desenvolvimento. Um projeto terá um escopo de duas a quatro semanas de desenvolvimento.

Todos os dias, no início da jornada de trabalho, toda a equipe do projeto irá fazer uma reunião, com no máximo quinze minutos de duração, para expor as atividades realizadas, as dificuldades e soluções encontradas no dia anterior.

No projeto é utilizado o Kanban, para sinalização do andamento, onde as tarefas são divididas em três estados *To Do*, *Doing*, *Done*. *To Do* são todas as tarefas que serão desenvolvidas (*Product Backlog*), *Doing* são as tarefas que estão sendo desenvolvidas e *Done* são as tarefas que foram desenvolvidas e serão testadas. Se a tarefa for reprovada no teste ela volta para *To Do*, senão ela permanece em *Done*.

Antes do início do projeto, é construído um documento de escopo, onde estarão todas as informações necessárias para o desenvolvimento do projeto. Esse documento deverá conter as funcionalidades do sistema divididas em requisitos, ou seja, um agrupamento de funcionalidades.

Para um melhor entendimento do sistema, o responsável pelo projeto também pode fazer o uso de dois diagramas, o Diagrama de Entidade Relacionamento, para especificar a estruturas dos dados e o Diagrama de Atividades, para explicação de regras que sejam mais complexas.

No final do projeto, é feita uma reunião com toda equipe. Nessa reunião é repassado o tempo total gasto no projeto, apontando os pontos positivos e negativos. Também é discutido qual será o planejamento para as próximas *Sprints* e se possível é repassado o *feedback* do cliente.

3 LEVANTAMENTO PARA O DESENVOLVIMENTO

Nos dias atuais, para desenvolver um software de qualidade é necessário um mínimo de gerenciamento, Segundo Pressman (1995), para que um projeto de software seja bem sucedido, é necessário que alguns parâmetros sejam corretamente analisados, como por exemplo, o escopo do software, os riscos envolvidos, os recursos necessários, as tarefas a serem realizadas, os indicadores serem acompanhados, os esforços e custos aplicados e a sistemática a ser seguida. A análise de todos estes parâmetros seria a função típica do gerenciamento de projetos a qual, em geral, se inicia antes do trabalho técnico e prossegue à medida que a entrega do software vai se concretizando.

Com forte necessidade de um gerenciamento no processo de desenvolvimento, a empresa Kreea Sistemas *Web*, teve a necessidade de construir um sistema para controlar e auxiliar o processo de desenvolvimento adotado por ela.

Para ter um *feedback*, o sistema foi colocado em teste durante a rotina de trabalho da empresa, para verificar se todas as necessidades para o processo de desenvolvimento foram sanadas.

3.1 EMPRESA KREEA

A Kreea Sistemas *Web* é uma microempresa constituída na cidade de Cornélio Procópio, formada por alunos da UTFPR – Universidade Tecnológica Federal do Paraná. A empresa foi fundada com o intuito de se diferir, de outras empresas do ramo, onde não há hierarquia, todos podem expor suas opiniões e soluções.

O foco de trabalho da empresa está no desenvolvimento de uma plataforma de comércio eletrônico, tendo o diferencial da customização total ou parcial das interfaces de venda, outro ramo da empresa é o desenvolvimento de software por demanda, este ramo é conhecido atualmente como fábrica de software, onde uma pessoal necessita de um sistema, e a Kreea desenvolve segundo as especificações.

A empresa atualmente possui seis funcionários, sendo quatro desenvolvedores, um vendedor e um administrador. Apesar da pequena quantidade

de colaboradores, nos três primeiros meses a empresa mostrou qualidades nos produtos, tendo sempre um *feedback* positivo seus clientes

3.2 JUSTIFICATIVA

Pelo fato da empresa ser formada por alunos do Curso de Análise e Desenvolvimento de Sistemas da UTFPR, todos sabem que é de suma importância o gerenciamento do processo de desenvolvimento de um software.

Tendo essa necessidade a empresa decidiu adotar o Kanban como gerenciamento das tarefas e o *framework* do Scrum para controlar o processo de desenvolvimento. Essas duas opções foram escolhidas por serem de fácil entendimento e por servir em um método ágil de desenvolvimento.

4 FERRAMENTA KONTROLLER

Para o desenvolvimento do Kontroller, produto final deste trabalho, procurou-se criar uma experiência de uso simples e prática para os usuários. Como a gestão de projetos nem sempre é algo fácil de se realizar, a praticidade e facilidade de uso do Kontroller deve facilitar o trabalho de seus usuários.

O Kontroller foi desenvolvido baseado no *framework* Scrum (descrito na seção 2.2), e no método de gestão de processo Kanban (descrito na seção 2.1), e possibilitará uma interação maior dos colaboradores da empresa em relação às tarefas que estão sendo desenvolvidas em cada projeto. Para isso, é essencial que os usuários utilizem assiduamente a ferramenta.

4.1 OBJETIVOS DA FERRAMENTA

O Kontroller foi desenvolvido com uma estrutura que permite que mais de uma empresa utilize a ferramenta, com seus dados sendo acessados somente pelo usuário que pertence a empresa. Para isso, um usuário administrado geral tem acesso à uma área nomeada “Administração” onde é possível gerenciar empresas e seus respectivos usuários.

Por essa mesma área Administração, os usuários do sistema também conseguem gerenciar outros dados necessários no sistema, como Associados – que é representado por cliente, fornecedores e parceiros, Requisitos Padrão – requisitos que estão presente nas maiorias do projeto e Perfis de Usuário – perfil de ações que o usuário pode executar no sistema.

O sistema também conta com uma estrutura de permissões onde cada perfil de usuário é configurado com as ações que pode ser realizado no sistema. Para cada usuário é definido um perfil em questão que limitará suas ações no sistema. Somente usuários habilitados podem definir as regras de permissão, tornando assim o sistema seguro, de acordo com as regras de cada empresa que o utiliza.

O Kontroller possui ainda as áreas de Agenda e Projetos. Em agenda os usuários da empresa podem realizar registros de compromissos gerais da empresa e na área Projetos são gerenciados todos os projetos da empresa, assim como suas tarefas.

4.2 ANÁLISE

Para realizar a análise para o desenvolvimento da ferramenta Kontroller foram levantados os requisitos necessários de acordo com o dia a dia de trabalho da empresa Kreea Sistemas Web. Com os requisitos definidos, estes foram documentados, gerando assim artefatos como o documento de escopo, diagrama de entidade e relacionamento e diagrama de atividades. Os futuros usuários da ferramenta, atualmente colaboradores da Kreea Sistemas Web, também opinaram sobre as funcionalidades e ajudaram a definir o fluxo de funcionamento da ferramenta.

4.2.1 Requisitos

Para levantar os requisitos da ferramenta Kontroller foram utilizadas as mesmas técnicas que são utilizadas na Kreea Sistemas Web. O *framework* Scrum e o método Kanban foram utilizados com a ajuda de um quadro, onde são colocadas as tarefas a serem executadas em cada projeto, juntamente com seu estado de execução.

A partir do quadro de tarefas, foram levantados os requisitos iniciais para a ferramenta, que são o controle de Projetos, Requisitos e Tarefas. Para a gestão administrativa do sistema foram levadas em conta as necessidades atuais da empresa, como Controle de Compromissos. O sistema ainda conta com gerenciamento de Usuários, gerenciamento de Perfis de Usuário e o gerenciamento de Clientes, Fornecedores, Parceiros e Futuros Clientes, chamada de Associados.

Todos os detalhes de cada requisito foi registrado em um documento chamado Documento de Escopo. Juntamente com outros desenvolvedores da Kreea Sistemas Web, cada requisito foi discutido e com base na experiência de cada colaborador, foi realizada uma estimativa em horas para o desenvolvimento de cada requisito. No apêndice E encontra-se o Documento de Escopo gerado com os detalhes de cada requisito da ferramenta.

Na etapa de levantamento de requisitos foram analisados alguns sistemas semelhantes ao Kontroller, para comparar as funcionalidades e tentar garantir o

sucesso do resultado final da ferramenta. As ferramentas analisadas foram o Trac¹, e a ferramenta TeamPulse².

A ferramenta Trac trata-se de um projeto *open-source* que permite o gerenciamento de projetos e suas tarefas, utilizando conceitos de Scrum em seu desenvolvimento. O Trac pode ser implantado e configurado em um servidor da empresa e pode ser integrado com tecnologias como *subversion* e github. Ele ainda permite que os usuários gerenciem as tarefas de um projeto, controlando o tempo utilizado para cada tarefa, estado de execução, cronograma de tarefas e controle de erros.

A ferramenta TeamPulse é fornecida pela empresa Telerik para gerenciamento de requisitos, projetos, tarefas e erros de projetos ágeis. Ele conta com funcionalidades utilização de Scrum e Kanban, controle de tempo utilizado nas tarefas, relatórios e controle de erros. O TeamPulse trata-se de uma ferramenta privada, sendo possível realizar um cadastro para utilização de trinta dias gratuitos. Na tabela 1 encontra-se uma comparação entre as ferramentas citadas e seus requisitos. Na última coluna, encontra-se as funcionalidades do sistema Kontroller.

Requisitos/Ferramentas	TRAC	TeamPulse	Kontroller
Empresas	x	x	x
Usuários	x	x	x
Associados		x	x
Requisitos			x
Agenda		x	x
Integração com Versionadores	x		
Projetos			x
Tarefas	x		x
Controle de Tempo			x
Relatórios			

Tabela 1 - Comparação entre *softwares* semelhantes.

Algumas das ferramentas possuem integração com aplicativos que realizam o versionamento e controle de projetos, como o *Subversion* e Microsoft TFS – *Team*

¹ Informações em: <<http://trac.edgewall.org/>>

² Informações em: <<http://www.telerik.com/agile-project-management-tools/>>

Foundation Server. Por se tratar de uma versão inicial e não estar diretamente ligado ao código fonte dos projetos, o Kontroller não realizará a integração com estas ferramentas.

4.2.2 Análise

O *framework* Scrum, por ser uma das metodologias ágeis, permite que o sistema a ser desenvolvido não necessite de etapas de análise longas. Devido a esse motivo, a análise do projeto foi realizada com apenas alguns diagramas afim de facilitar o entendimento de alguns requisitos complexos.

4.2.2.1 Diagrama de atividades

Segundo Gudwin (2011), os diagramas de atividades são utilizados com o objetivo de especificar os requisitos de um sistema. Apenas um diagrama de atividades foi elaborado, mostrando as atividades principais do sistema Kontroller, exemplificando também, como será seu funcionamento. A figura 5 mostra o diagrama de atividades da requisito de tarefas da ferramenta Kontroller. Ele exemplifica todo o fluxo em que uma tarefa passa no sistema.

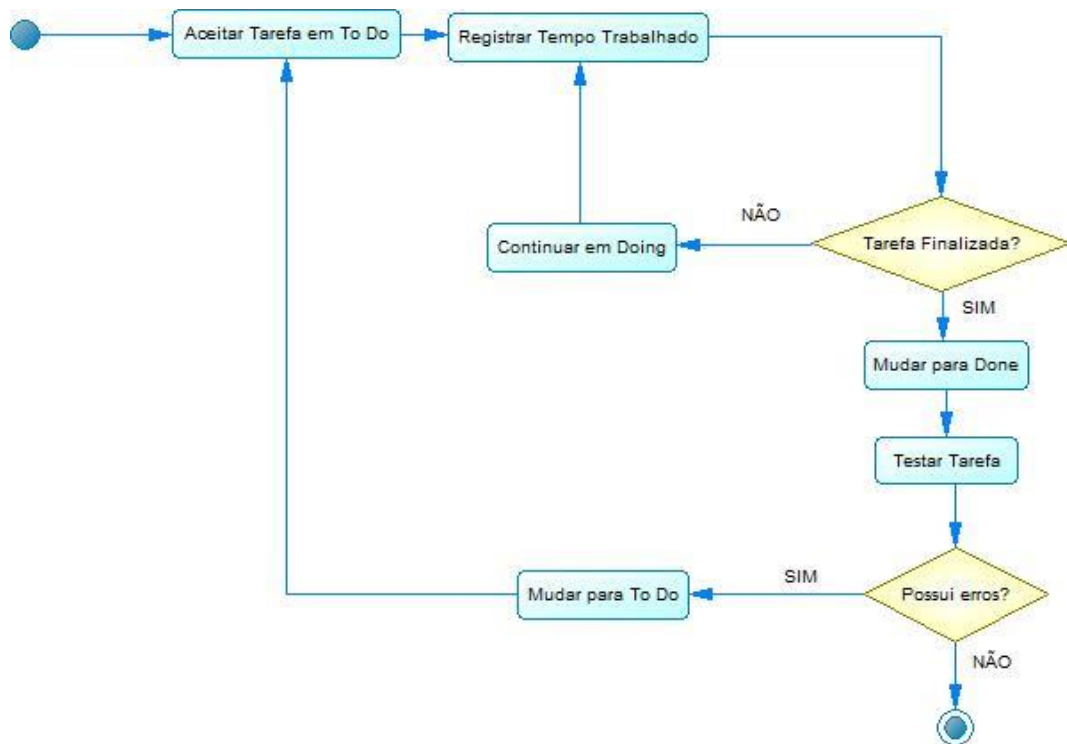


Figura 5 - Diagrama de Atividades sobre as Tarefas do Projeto.

4.3 DESENVOLVIMENTO

O desenvolvimento da ferramenta Kontroller foi gerenciado utilizando um quadro de tarefas. Para isso, foram criadas tarefas a partir dos requisitos que seriam desenvolvidos, tendo cada uma as horas estimadas para realizar, uma breve descrição e uma prioridade. A figura 6 exibe um modelo de tarefa que foi utilizado no quadro de tarefas.

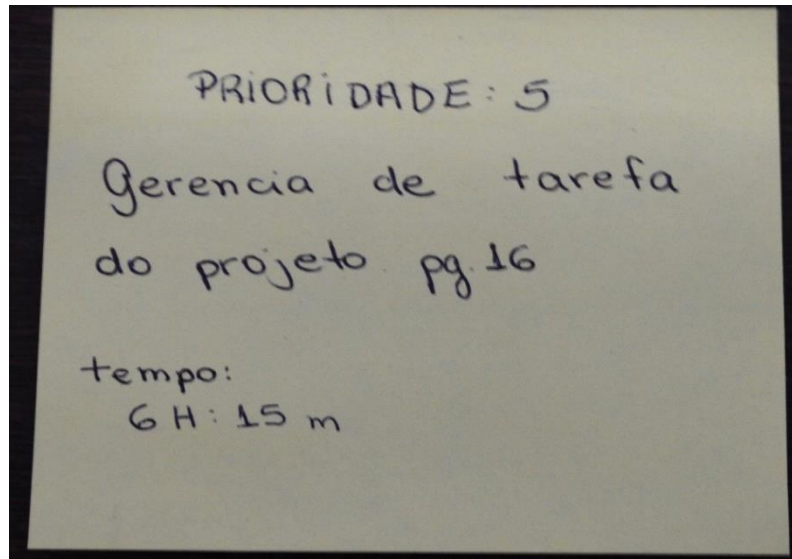


Figura 6 - Modelo de Tarefa.

Para utilização do quadro de tarefas, foi aplicado o *framework* Scrum e as técnicas de Kanban. O quadro foi dividido em três colunas para alocação das tarefas, sendo que cada coluna representa um estado de execução. Com isso, as tarefas foram separadas nos três estados de execução, sendo *To Do*, *Doing* e *Done*. *To Do* são as tarefas que estão em espera para serem desenvolvidas (*Backlog*), *Doing* são as atividades que estão em desenvolvimento e *Done* são as atividades que já foram finalizadas. Na figura 7 encontra-se uma imagem do quadro de tarefas no início do projeto.

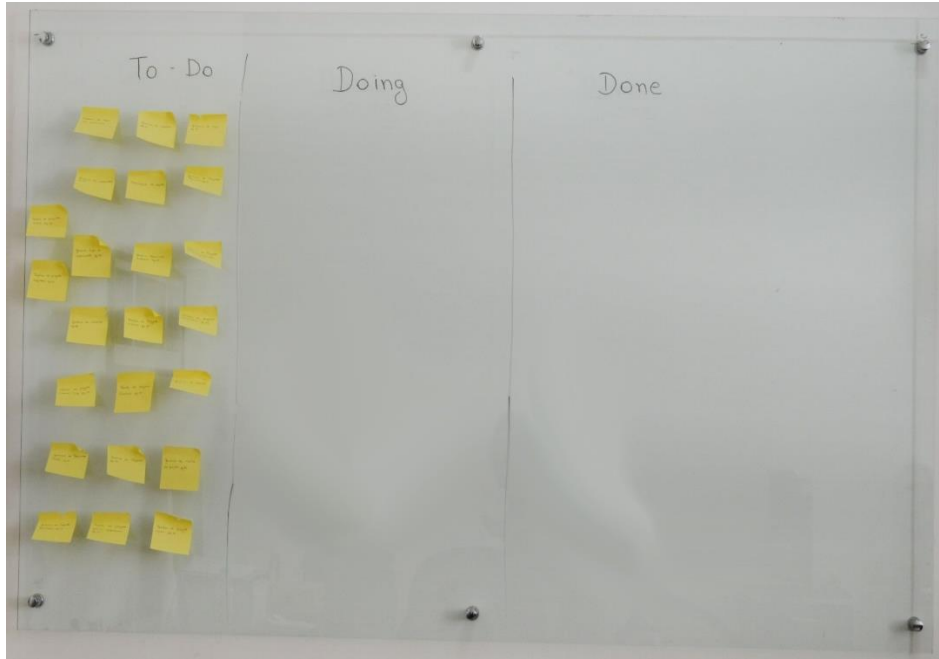


Figura 7 - Quadro de Tarefas.

Conforme as tarefas foram iniciadas, os *kanbans* eram passadas para o estado de *Doing*, tendo em vista que havia somente um desenvolvedor, o número de *kanban* em *Doing*, seria reduzido, como mostrado na figura 8 logo abaixo.

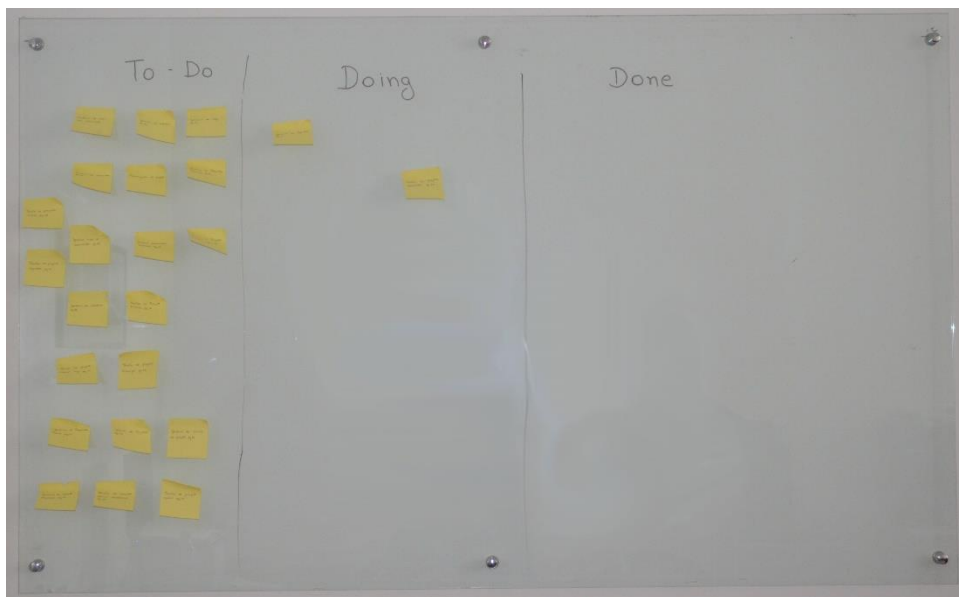


Figura 8 - Tarefas em desenvolvimento.

Assim que as tarefas foram terminadas, elas passavam para o estado de Done, como mostra a figura 9, mas uma tarefa só estava realmente terminada se ela fosse aprovada no testes.

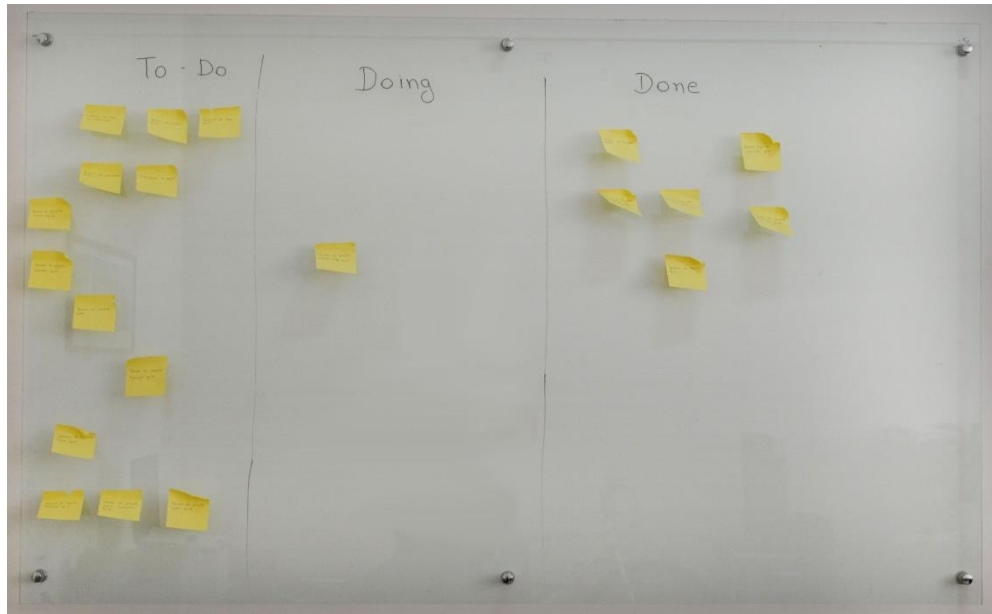
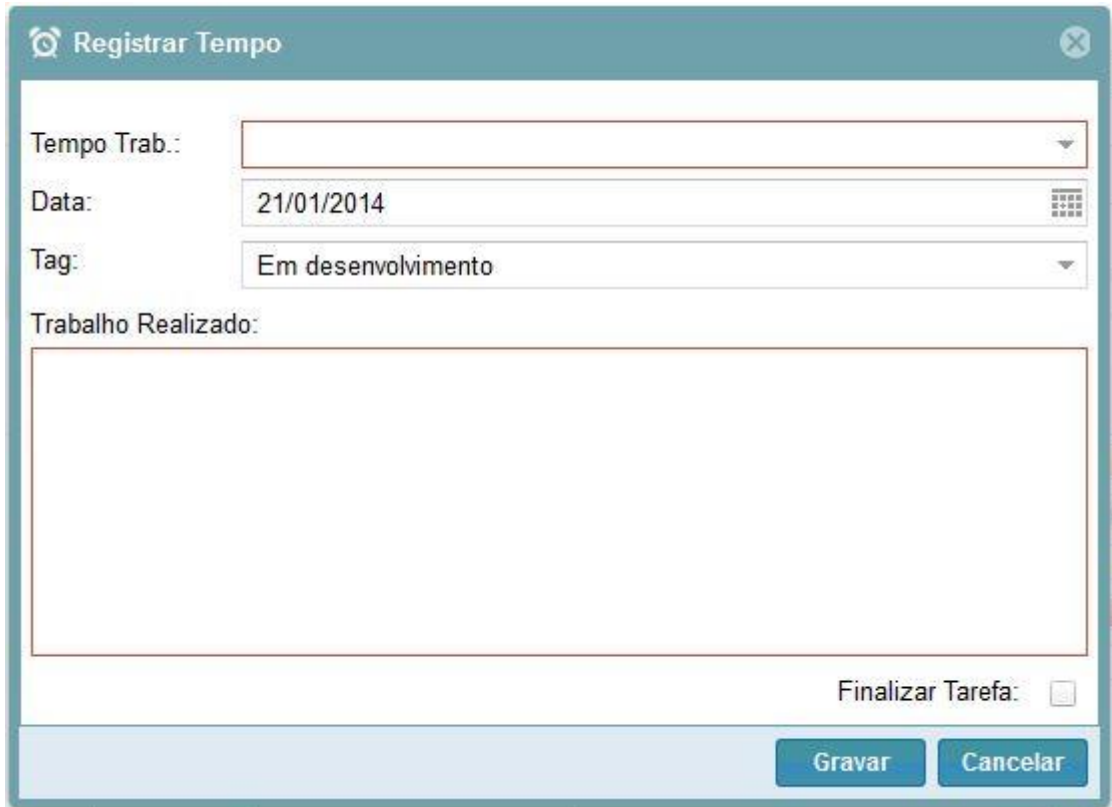


Figura 9 - Tarefas finalizadas.

4.4 FUNCIONALIDADES

As funcionalidades do Kontroller foram projetadas para serem semelhantes ao quadro de tarefas. Para isso a listagem de tarefas foi agrupada pelos estados, como mostra a figura no anexo A.

Para o gerenciamento do tempo gasto nas tarefas, o usuário responsável pela sua realização fica encarregado de registrar todo o tempo trabalhado nela, assim como registrar também o que foi feito e suas dificuldades, caso houverem. A figura 10 mostra como os usuários registram o tempo trabalhado em uma tarefa.



Registrar Tempo

Tempo Trab.:

Data: 21/01/2014

Tag: Em desenvolvimento

Trabalho Realizado:

Finalizar Tarefa:

Gravar Cancelar

Figura 10 - Registro de tempo trabalhado.

O Kontroller ainda conta com a funcionalidade de gerenciamento dos requisitos do projeto, onde estes são cadastrados e exibem ainda uma estatística de progresso, possibilitando ao responsável pelo projeto saber quanto de cada requisito está finalizado. A figura 11 mostra a funcionalidade de gerenciamento de requisitos.

Código	Name	Projeto	Criado em	Estimativa	Andamento	Remove
R0001	Administração	Kontrole Fuin	16/01/2014	15 hora(s)	20.00%	
R0002	Estoque	Kontrole Fuin	16/01/2014	25 hora(s)	50.00%	
R0003	Vendas	Kontrole Fuin	16/01/2014	90 hora(s)	0.00%	
R0004	Administrativo	Kontrole Fuin	16/01/2014	0 hora(s)	80.00%	

Nome:

Estimativa:

Descrição:

Figura 11 - Gerenciamento de Requisitos.

Para sinalizar alguma ação ou informação a uma tarefa, foram criadas *Tags* de marcação. No cadastro de *tags*³ são atribuídas cores para facilitar a identificação das tarefas. Assim que uma tarefa for marcada pela *tag*, ela recebe uma marcação com a cor cadastrada na *tag*. Essa funcionalidade é utilizada, por exemplo, para controlar as tarefas que precisam ser testadas, como mostra a figura 12.

³ Tags são classificação indicativas, para informar características das tarefas.

Código	Nome	Requisito	Resumo
- Situação: Done			
T0005	Gráfico de Projetos/Status	Projetos	Gráfico de Projetos/Status
T0002	Gerencia de Status	Status	Criação da gerencia de status,
T0004	Criação da base de dados	Estrutura	Criação da estrutura da base d
Aguardando Testes			
- Situação: To Do			
T0006	Gerencia de Perfis de Usuário	Usuário	Criação da gerencia de perfis
T0004	Gerencia de Usuarios	Usuário	
T0003	Gerencia de Projetos	Projetos	Desenvolvimento da gerencia

Figura 12- Tags da Tarefa.

4.5 TESTES

Após o termino do desenvolvimento de algumas tarefas, são realizados os testes das funcionalidades. Para realizar esse controle nos testes, utilizamos um documento descrevendo os erros encontrados. Se a tarefa testada estivesse de acordo com o proposta em seu *kanban*, recebia um "Testado OK", então ela realmente estava terminada. Se fosse encontrado alguma anormalidade, era anotada no documento de erros, e o *kanban* voltava para o estado de *To Do*.

Para que não houvesse problemas, os testes foram executados pelos colaboradores e usuários da Kreea Sistemas Web. Nenhum teste foi executado pela pessoa que codificou. Na figura 13, mostra um *kanban* com o teste realizado.

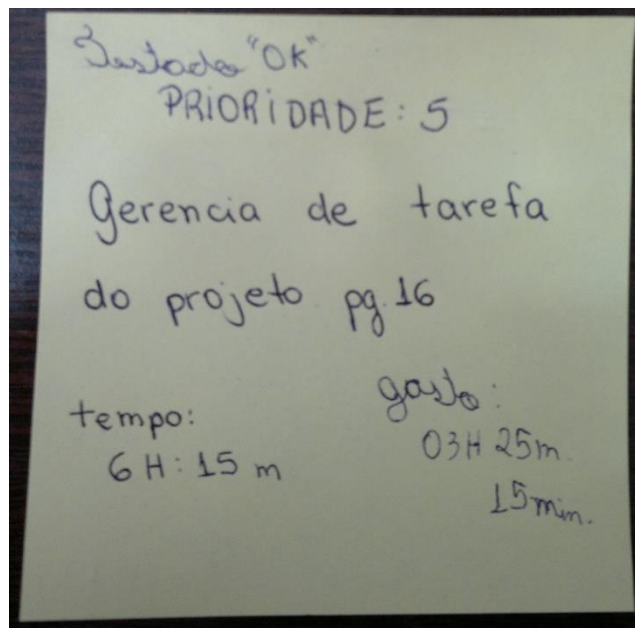


Figura 13 - "Testado OK".

4.6 CRONOGRAMA

Desde o início do projeto, até a implantação da ferramenta na empresa, procurou-se seguir o cronograma. Durante a execução do trabalho houveram impedimentos que acarretaram em uma mudança no cronograma.

Entre esses impedimentos, está a recém abertura da empresa Kreea Sistemas *Web*, que necessitava de uma dedicação especial a sua infraestrutura, aos seus primeiros clientes e ao seu funcionamento em geral para definir também os requisitos do sistema.

O cronograma realizado encontra-se no apêndice A, e o proposto no apêndice B.

5 ARQUITETURA

A ferramenta Kontroller, irá manter o mesmo padrão de projeto utilizado na Kreea Sistemas *Web*. A arquitetura utilizada será o MVC (*Model-View-Controller*), essa escolha foi feita, por conta da tecnologia utilizada no desenvolvimento, ASP.net MVC, onde a estrutura e a interação entre as divisões foram baseada na arquitetura do MVC.

Para GAMMA o “MVC é composto por três tipos de objetos. O modelo é o objeto de aplicação, a vista é a apresentação na tela e o controlador define a maneira como a interface do usuário reage às entradas do mesmo. Antes do MVC, os projetos de interface para o usuário tendiam em agrupar esses objetos. MVC para aumentar a flexibilidade e a reutilização.” (GAMMA, 2000).

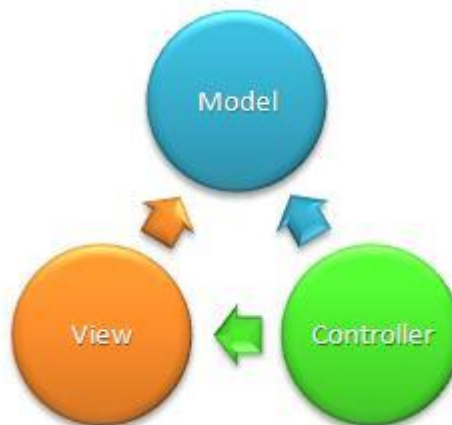


Figura 14 – Padrão MVC

6 CONCLUSÕES

O resultado esperado com a implantação do sistema *Kontroller* no setor de desenvolvimento da *Kreea Sistemas Web* é uma ferramenta que fosse capaz de gerenciar as tarefas e o processo de um projeto. Para verificar se os resultados foram obtidos, foi realizada uma avaliação com os usuários do sistema durante o período de implantação. Nas próximas seções são descritos os detalhes sobre a avaliação dos usuários.

6.1 RESULTADOS

Em uma avaliação feita com os usuários que tiveram contato com a ferramenta foram elaboradas questões relacionadas ao objetivo do sistema. Os avaliados foram questionados sobre as funcionalidades, sobre a facilidade em controlar as tarefas na ferramenta e sobre questões de melhoria do desempenho da equipe. Foram questionados também sobre sugestões e reclamações.

Com relação às respostas dos usuários o resultado foi considerado satisfatório. Mesmo com o número reduzido de colaboradores, foi notado uma melhora na questão do controle das tarefas, e de tempo gastos nas tarefas. Os itens contidos no questionário foram: Usabilidade do sistema, eficiência no controle de tarefas, busca por tarefas pendentes e verificar o andamento do projeto.

Nos itens que se tratavam de usabilidade e interação, não houve respostas negativa, apenas uma observação. O item de controle de projeto, os entrevistados não apresentaram nenhuma dificuldade em cadastrar projeto e nem controlar as tarefas.

De acordo com resultados do questionário, foi possível perceber que os usuários não tiveram nenhuma dificuldade em utilizar o sistema, tendo em vista esse resultado também se deve a participação deles no levantamento. O modelo das perguntas encontra-se no apêndice C, e o resultado com as respostas no anexo B.

REFERÊNCIAS

- GUDWIN, R. R. **Diagrama de Atividade e Diagramas de Estado**. Campinas: Dca-feec-unicamp, 2011. 13 p.
- GAMMA, Erich et al. **Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos**. Porto Alegre: Bookman, 2000. 366 p.
- JUNIOR, M.L.; FILHO, M.G., **Adaptações ao sistema *kanban*: revisão, classificação, análise e avaliação**. Gest. Prod., São Carlos, v. 15, n.1, p. 173-188, jan.-abr. 2008.
- KNIBERG, H.; SKARIN, M., **Kanban and Scrum: making the most of both**. USA: C4Media Inc, 2010. 104 p.
- LADAS, C. **Scrumban: Essays on Kanban Systems for Lean Software Development**. USA: Modus Cooperandi Inc, 2008. p.177.
- PRESSMAN, R.S., **Engenharia de Software: Uma Abordagem Profissional**. 7ªed. Porto Alegre: AMGH Editora Ltda, 2011. 780 p.
- SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum: As regras do jogo**. 2011. SHARIQ, S. Z. **Knowledge management: an emerging discipline**. The Journal of Knowledge Management, v. 1, n. 1, set. 1997.
- SUGIMORI, Y.; KUSUNOKI, K.; CHO, F.; UCHIKAWA S., **Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system**: International Journal of Production Research. Japan:1977, 6 ed. 15 v.

APÊNDICE A – Cronograma Proposto

	Fevereiro	Março	Abril	Maio	Junho	Julho	Agosto	Setembro
Proposta do Trabalho	█	█	█					
Concepção			█					
Elaboração				█				
Construção				█	█	█	█	
Transição							█	
Escrita da Proposta				█	█	█	█	█
Apresentação da Proposta								█

Tabela 2 -Cronograma Proposto

APÊNDICE B – Cronograma Realizado

Ano	2013												2014	
Tarefa/Mês	Janeiro	Fevereiro	Março	Abril	Maió	Junho	Julho	Agosto	Setembro	Outubro	Novembro	Dezembro	Janeiro	Fevereiro
Escolha do Tema	■		■		■		■		■		■		■	
Levantamento	■	■	■		■		■		■		■		■	
Revisão Bibliográfica	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Redação da Proposta	■	■	■	■	■		■		■		■		■	
Análise e Projeto	■		■		■	■	■	■	■		■		■	
Codificação	■		■		■		■	■	■	■	■		■	
Testes gerais e Implantação	■		■		■		■		■	■	■	■	■	
Redação do TD	■		■		■	■	■	■	■	■	■	■	■	
Defesa do TD	■		■		■		■		■		■		■	■

Tabela 3 - Cronograma Realizado.

APÊNDICE C – Modelo do Questionário

AVALIAÇÃO DE USABILIDADE E
FUNCIONALIDADE DE SISTEMA KONTROLLER

NOME:

IDADE:

CARGO:

ROTEIRO DE TAREFAS:

- 1 - ACESSAR O SISTEMA;
- 2 - ACESSAR O MENU DE PROJETOS;
- 3 - CADASTRAR UM NOVO PROJETO;
- 4 - CADASTRAR UM NOVO REQUISITO;
- 5 - CADASTRAR 3 NOVAS TAREFAS NO PROJETO RECÉM CRIADO;
- 6 - ACEITAR UMA NOVA TAREFA;
- 7 - REGISTRAR TEMPO GASTO EM UMA TAREFA;

QUESTIONÁRIO

- 1 – Você teve alguma dificuldade em acessar o sistema?
- 2 – O menu de projetos estava em fácil localização?
- 3 – Você teve alguma dificuldade em cadastrar um novo projeto? Se sim, quais foram as dificuldades?
- 4 – Você teve alguma dificuldade em cadastrar um novo requisito? Se sim, quais foram as dificuldades?
- 5 - Teve alguma dificuldade em cadastrar uma nova tarefa? Se sim, quais foram as dificuldades e quais sugestões você poderia dar para melhorar?
- 6 - Alguma dificuldade em encontrar o botão para aceitar uma tarefa?
- 7 - Você teve alguma dificuldade em registrar o tempo gasto em uma tarefas? Se sim, quais foram elas e quais melhorias você sugere?
- 8 - Depois de ter utilizado, você teria alguma sugestão ou crítica sobre o sistema?

APÉNDICE D – Documento de Escopo

DOCUMENTO DE ESCOPO KONTROLLER

KE012/2013

SUMÁRIO

DESCRIÇÃO DO DOCUMENTO	44
1. Padrões	45
2. Empresas	46
3. Usuários	47
4. Tipos de Associados.....	48
5. Associados	48
6. Tags	51
7. Requisito Padrão	52
8. Agenda	52
9. Projetos	54
Requisito	55
10. Tarefas do Projeto.....	57
Registrar.....	57
Aceitar	58
Reabrir	58
Finalizar.....	58
Anexo	58
Comentário.....	59
Alterar Tag	59
Definir Responsável	60

DESCRIÇÃO DO DOCUMENTO

Neste documento estão relatadas as funcionalidades do projeto definidos por nosso time. Este documento inclui regras, protótipos, diagramas e explicações textuais.

1. Padrões

1. Padrão de Layout

Todas a gerencias, do sistema deverá seguir um modelo de layout, onde cada gerencia deverá ter um grid listando os seus itens, tendo tambem a opção de pesquisa, e os botão necessários para realização das ações. Segue um protótipo:

Novo	Editar	Remover	Pesquisar	Pesquisar
Column 1	Column 2			
Cell Content 1	Cell Content 1			
Cell content 2	Cell Content 2			
Cell content 3	Cell Content 3			

2. Permissões

Todas as gerencias e funcionalidades devem conter permissões, salve as funcionalidades que tenham regras especificas, das quais são relatadas no próprio requisito. E durante a programação todas as permissões devem ser validadas.

No projeto em questão, as permissões serão dadas por empresa, para que assim, em cada empresa o usuário possa ter permissões diferentes.

2. Empresas

Esta gerencia, será responsável pelo cadastro das empresas, ela deve conter os seguintes campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Nome	Campo Texto	Sim	1 - 150
WebSite	Campo Texto	Não	1-150
E-mail	Campo Texto	Sim	1-100
Responsável p/ Contato	Campo Texto	Sim	1-120
CNPJ	Campo Texto	Não	1-18
Endereço	Campo Texto	Não	1-30
Estado	ComboBox (Listar Estados presente no banco de Dados)	Não	1-1
Cidade	ComboBox (Listar Cidades de acordo com o estado selecionado)	Não	1-1
Número	Campo Numérico	Não	1-1
Bairro	Campo Texto	Não	1-100
Complemento	Campo Texto	Não	1-100
Telefone	Campo Texto	Não	1-15
Limite de Usuário	Campo Numérico	Não	1-99999
Empresa Ativa	CheckBox	Não	-
Observações	Campo Texto Longo	Não	1-500

Esta gerencia possuirá botões, Novo, Editar e Remover, e só devem ser habilitados se o usuário logado no sistema tiver permissões.

Nesta gerencia também existirá uma função associar o usuários a uma empresa específica, para que no momento do login, o usuário possa ver informações da empresa associada. Essa funcionalidade deverá abrir uma janela, listando os usuários cadastrados no sistema, para que possa ser selecionados os usuários associados.

Protótipo:

3. Usuários

Esta gerencia será responsável por cadastrar os usuários no sistema, e deve conter os seguintes campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Nome	Campo Texto	Sim	1 - 150
E-mail	Campo Texto	Sim	1-1
Empresa	ComboBox (Listar empresas cadastradas)	Sim	1-1
Foto	Campo Upload	Não	1-1
CPF	Campo Texto	Não	1-14
Senha	Campo Texto (Formatado para Senha)	Sim	1-30

Esta gerencia possuirá botões, Novo, Editar e Remover, e só devem ser habilitados se o usuário logado no sistema tiver permissões.

Protótipo:

Novo Usuário

Nome:

E-mail:

Empresa:

Foto: Selecionar 

CPF:

Senha:

Conf. a Senha:

Gravar Cancelar

4. Tipos de Associados

Esta gerencia tem como finalidade o cadastro de tipos de associados, e deve conter os seguinte campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Nome	Campo Texto	Sim	1 - 150
Ação	ComboBox (Itens Fixos – Entrada, Saída, Saída Entrada)	Sim	1-1
Observação	Campo Texto Longo	Não	1-500

Esta gerencia possuirá botões, Novo, Editar e Remover, e só devem ser habilitados se o usuário logado no sistema tiver permissões.

5. Associados

Esta gerencia tem como finalidade cadastrar os associados da empresa, os associados podem ser clientes, funcionários, fornecedores. Esta gerencia deve conter o seguintes campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Identificar Pessoa	RadioButton (Pessoa Física, Pessoa Jurídica)	Sim	1 - 1
Tipo de Cliente	ComboBox (Carregar os itens cadastrados na Gerencia de Tipos de Associados)	Sim	1-1
Nome	Campo Texto	Sim	1-120
Razão Social	Campo Texto	Não	1-120
Responsável pela Empresa	Campo Texto	Não	1-120
E-mail	Campo Texto	Não	1-120
CPF	Campo Texto	Não	1-14
CNPJ	Campo Texto	Não	1-18
Telefone	Campo Texto	Não	1-80
Telefone Comercial	Campo Texto	Não	1-80
Celular	Campo Texto	Não	1-80

Esta gerencia terá uma regra na qual, se o Identificador de Pessoa for do tipo Pessoa Física os campos Razão Social, Responsável pela Empresa, CNPJ devem ser ocultados.

Protótipos:

O protótipo mostra uma janela intitulada "Novo Cliente" com uma barra de título azul e um ícone de fechar no canto superior direito. O formulário contém os seguintes elementos:

- Seleção:** Dois botões de opção: "Pessoa Física" (selecionado) e "Pessoa Jurídica".
- Tipo do Cliente:** Um campo de lista suspensa.
- Nome:** Um campo de texto.
- E-mail:** Um campo de texto.
- CPF:** Um campo de texto.
- RG:** Um campo de texto.
- Data Nasc.:** Um campo de data com um ícone de calendário.
- Telefone:** Um campo de texto.
- Tel. Comercial:** Um campo de texto.
- Celular:** Um campo de texto.

Na base da janela, há dois botões: "Gravar" e "Cancelar".



Ainda nesta gerencia será possível cadastrar os endereços dos associados. Para esta gerencia será necessário os seguintes campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Endereço Padrão	CheckBox	Não	0 - 1
Identificação	Campo Texto	Sim	1-150
Logradouro	Campo Texto	Sim	1-120
Numero	Campo Numérico	Sim	1-9999
Bairro	Campo Texto	Sim	1-120
Estado	ComboBox (Listar os estados presente em banco)	Sim	1-1
Cidade	ComboBox (Listar as cidades presente em banco)	Sim	1-1
CEP	Campo Texto	Não	1-10
Complemento	Campo Texto	Não	1-150
Observação	Campo Texto Longo	Não	1-500
Celular	Campo Texto	Não	1-80

7. Requisito Padrão

Esta gerencia tem como finalidade o cadastro de Requisitos Padrão, e deve conter os seguinte campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Nome	Campo Texto	Sim	1 - 150
Estimativa	Campo Numérico	Sim	1-99999
Descrição	Campo Texto Longo	Não	1-500

Protótipo:

Esta gerencia possuirá botões, Novo, Editar e Remover, e só devem ser habilitados se o usuário logado no sistema tiver permissões.

8. Agenda

Esta gerencia tem como finalidade o cadastro de eventos que precisam ser realizados pela empresa, uma agenda, e deve conter os seguinte campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Nome	Campo Texto	Sim	1 - 150
Associado	ComboBox (Listar os associados cadastrados)	Não	1-1
Projeto	ComboBox (Listar os projetos de acordo com o associado selecionado)	Não	1-1
Data	Campo de Data	Sim	-
Horário	Campo de Horário	Sim	-
Descrição	Campo Texto Longo	Sim	1-500
Estrada Pessoal	CheckBox	Não	0-1
Participantes	Grid (Listar usuários cadastrados c/ checkbox p/ selecionar as pessoas que irão participar do evento)	Não	-

Se a opção de Entrada Pessoal estiver marcada, ela só deverá aparecer para a pessoa que a cadastrou.

Protótipo:

Nova Entrada na Agenda

Agendamento | Participantes

Nome:

Cliente:

Projeto:

Data: 21/01/2014 Horário:

Descrição:

Entrada Pessoal?

Gravar Cancelar

Nova Entrada na Agenda

Agendamento | Participantes

<input type="checkbox"/>	Miniatura	Nome
<input type="checkbox"/>		Carlos Henrique Prado Sousa
<input type="checkbox"/>		Diogo Cezar Teixeira Batista
<input type="checkbox"/>		Paola Caroline Kishi Fredegoto
<input type="checkbox"/>		Leonardo Fiewski Soares
<input type="checkbox"/>		Edilson Nabhen Filho

Entrada Pessoal?

Gravar Cancelar

Ainda nesta gerencia existirá uma opção de Resultado, onde o usuário poderá apresentar o resultado do evento, para isso os seguintes campos se fazem necessários:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Feedback	Campo Texto Longo	Sim	1 - 500
Deseja Retorno	CheckBox	Não	0-1

Protótipo:

Se a opção de “Deseja Retorno”, estiver selecionado, a janela de evento deve abrir, para cadastrar um novo evento, e deve ser salvo também o id do evento que a originou.

9. Projetos

Esta gerencia tem por finalidade o cadastro e manutenção de projetos presente na empresa, para isso se faz necessário os seguintes campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Nome	Campo Texto	Sim	1 - 150
Responsável	ComboBox (Deve listar os usuários cadastrados)	Sim	1-1
Associado	ComboBox (Deve listar os associados cadastrados)	Sim	1-1
Preço/Hora	Campo Numérico	Sim	1-9999999
Desconto	Campo Numérico	Não	1-9999999
Tempo Limite	Campo de Data	Sim	-
Previsão	Campo de Data	Sim	-
Status	ComboBox (Deve listar 4 itens fixos, Projeto, Em Desenvolvimento, Em Manutenção, Finalizado)	Sim	1-1
Observações	Campo Texto Longo	Não	1-500

Esta gerencia terá as opções de Novo, Editar, Remover, mas para que elas possam ser executadas o usuário logado deverá ter permissão.

Protótipo:

The image shows a software window titled "Novo Projeto" (New Project). It contains the following fields and controls:

- Nome:** A text input field.
- Responsável:** A dropdown menu.
- Cliente:** A dropdown menu.
- Preço/Hora:** A text input field.
- Desconto:** A text input field.
- Tempo Limite:** A text input field with a small grid icon to its right.
- Previsão:** A text input field with a small grid icon to its right.
- Status:** A dropdown menu.
- Observações:** A large text area for notes.
- Buttons:** "Gravar" (Save) and "Cancelar" (Cancel) buttons at the bottom right.

Na gerencia Será necessário um botão que só será ativo quando um projeto for selecionado, com o nome de “Abrir”, e assim que for clicado, deve ser redirecionado para a página de tarefas do projeto, as opções desta página será descrita logo abaixo no item 10.

3. Requisito

Ainda nesta gerencia, terá a opção de cadastro de requisitos do projeto, para que seja cadastrado um requisito, para que seja cadastrado um requisito será necessário os seguintes campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Nome	Campo Texto	Sim	1 - 150
Estimativa	Campo Numérico	Sim	1-99999
Descrição	Campo Texto Longo	Não	1-500

Protótipo:

Requisitos do Projeto

Novo Editar Remover

Código	Nome	Projeto	Criado em	Estimativa	Andamento	Remover
R0001	Administração	Kontrolle Fuin	16/01/2014	15 hora(s)	20.00%	
R0002	Estoque	Kontrolle Fuin	16/01/2014	25 hora(s)	50.00%	
R0003	Vendas	Kontrolle Fuin	16/01/2014	90 hora(s)	0.00%	
R0004	Administrativo	Kontrolle Fuin	16/01/2014	0 hora(s)	80.00%	

Nome:

Estimativa:

Descrição:

Gravar Cancelar

Fechar

Para que seja possível cadastrar, editar e remover o usuário terá que possuir as respectivas permissões.

Para que não seja necessário, ficar reescrevendo requisitos, nesta gerencia terá que existir uma opção de importar os Requisitos Padrões (item 7);

Para que se possa importar os requisitos será necessário uma grid listando os Requisitos Padrão, com checkboxes, para selecionar os requisitos que necessita importar.

Protótipo:

Importar Requisitos Padrão

<input type="checkbox"/> Nome	Estimativa
<input type="checkbox"/> Usuários	3 hora(s)
<input type="checkbox"/> Recorte do Layout	10 hora(s)
<input type="checkbox"/> Banco de Dados	5 hora(s)
<input type="checkbox"/> Login	3 hora(s)
<input type="checkbox"/> Planejamento do Projeto	5 hora(s)

Importar Cancelar

10. Tarefas do Projeto

Essa gerencia tem por finalidade o gerenciamento das tarefas de um projeto, para que possa cadastrar ou editar uma tarefa se faz necessário os seguintes campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Nome	Campo Texto	Sim	1 - 150
Responsável	ComboBox (Listar todos os usuários)	Sim	1-1
Requisito	ComboBox (Listar todos os requisitos previamente cadastrados para o projeto)	Sim	1-1
Tag	ComboBox (Listar as tags previamente cadastradas)	Sim	1-1
Estimado(Horas)	Campo Numérico	Sim	1-99999
Descrição	Campo Texto Longo	Sim	1-MAX
Resumo da Tarefa	Campo Texto Longo	Não	1-500

Assim que uma tarefa for cadastrada, ela deve se encontrar no estado de "To Do". Para cadastrar uma tarefa, é necessário que o usuário logado tenha as devidas permissões.

4. Registrar

Esta funcionalidade será criada com o objetivo, de ser apontado o tempo gasto em cada tarefa. Para registrar o tempo gasto é necessário os seguintes campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Tempo Trabalhado	Campo de Horas	Sim	1-24
Data	Campo de Data	Sim	-
Tag	ComboBox (Listar as tags previamente cadastradas)	Sim	1-1
Trabalho Realizado	Campo Texto Longo	Sim	1-500
Finalizar Tarefa	CheckBox	Não	0-1

Protótipo:

O protótipo mostra uma janela intitulada "Registrar Tempo". No topo, há um ícone de relógio e o título "Registrar Tempo" com um botão de fechar. Abaixo, há quatro campos de entrada: "Tempo Trab." (um campo vazio com uma seta para baixo), "Data:" (contendo "21/01/2014" com um ícone de calendário), "Tag:" (contendo "Em desenvolvimento" com uma seta para baixo) e "Trabalho Realizado:" (um campo de texto grande e vazio). Na parte inferior direita, há um campo "Finalizar Tarefa:" com uma caixa de seleção desativada. Na base da janela, há dois botões: "Gravar" e "Cancelar".

Nesta opção se o campo “Finalizar Tarefa”, estiver assinalado, a tarefa deverá ser colocada em estado de “Done”.

5. Aceitar

Essa opção deverá estar na gerencia das tarefas, e só deve ser habilitada se o alguma tarefa estiver selecionada e o estado da tarefa deve estar em “To Do”

A partir do momento que um usuário “Aceitou” a tarefa, a mesma deverá ir para o estado de “Doing”;

6. Reabrir

Essa opção deverá estar na gerencia das tarefas, e só deve ser habilitada se alguma tarefa estiver selecionada e estiver no estado de “Done”.

Assim que o usuário reabriu a tarefa, a tarefa deve voltar pro estado de “To Do”.

7. Finalizar

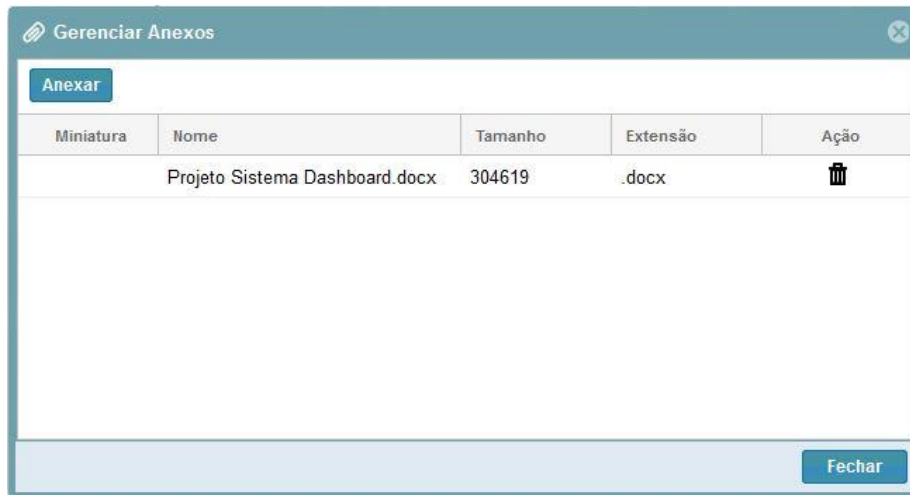
Essa opção deverá estar na gerencia das tarefas, e só deve ser habilitada se alguma tarefa estiver selecionada e estiver no estado de “To Do” ou “Doing”.

Assim que o usuário finalizar uma tarefa, a mesma deve ir para o estado de “Done”.

8. Anexo

Essa opção tem a finalidade de fazer um anexo na tarefa, sendo ela qualquer tipo de arquivo. Para isso, será aberto uma janela, contendo um FileUpload e um Grid, para serem listados os arquivos já anexados.

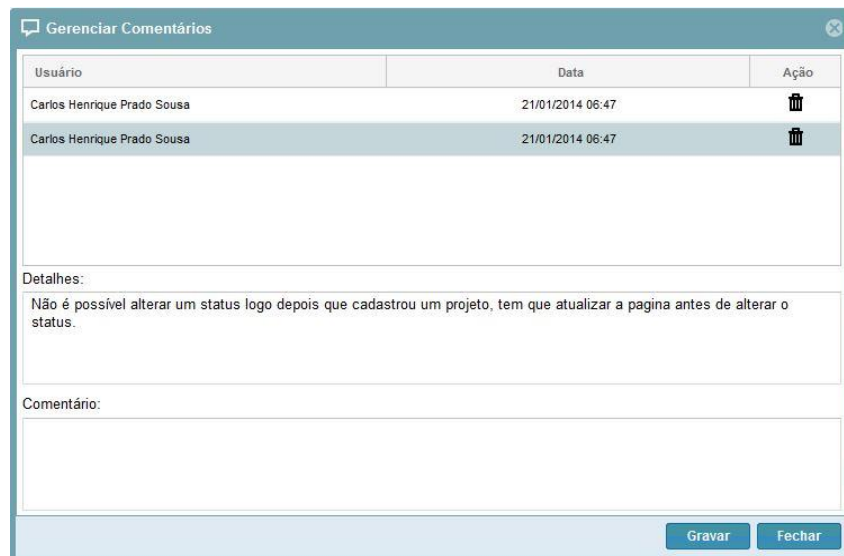
Protótipo:



9. Comentário

Essa opção tem como finalidade, o registro de comentários em uma tarefa. Para isso será necessário um campo texto longo, para escrever o comentário e uma grid com os comentários já realizados.

Protótipo:



10. Alterar Tag

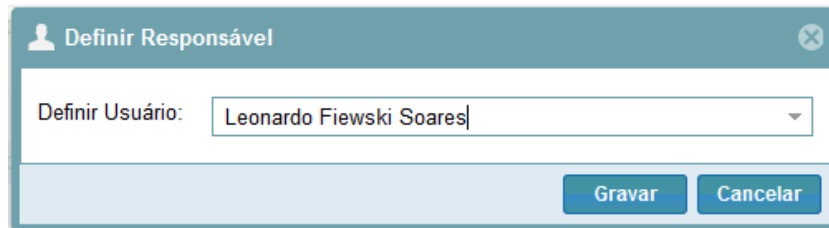
Essa funcionalidade, servirá para realizar a troca da tag da tarefa, para que haja a troca será necessário as seguintes campos:

Propriedade	Tipo	Obrigatório	Mínimo - Máximo
Tag	ComboBox (Listando as tags já cadastradas)	Sim	1-1
Observações	Campo Texto Longo	Não	500

11. Definir Responsável

Essa funcionalidade, tem a ação de atribuir um responsável a tarefa, para que isso ocorra, será necessário informar um usuário, que deve ser listado em um combobox, para que possa ser feita a seleção do usuário.

Protótipo:



O protótipo mostra uma janela de diálogo com o título "Definir Responsável". No topo, há um ícone de usuário e um botão de fechar. O campo principal é um combobox rotulado "Definir Usuário:" com o nome "Leonardo Fiewski Soares" selecionado. Na base da janela, há dois botões: "Gravar" e "Cancelar".

APÊNDICE E - Proposta Trabalho de Diplomação

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

LEONARDO FIEWSKI SOARES

SISTEMA RASTRO 2.0 – MÓDULO ADMINISTRATIVO

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2013

LEONARDO FIEWSKI SOARES

SISTEMA RASTRO 2.0 – MÓDULO ADMINISTRATIVO

Proposta de trabalho de conclusão de curso apresentada como requisito parcial a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná – Campus Cornélio Procópio.

Orientador: Fábio Fernandes da Rocha
Vicente

CORNÉLIO PROCÓPIO

2013

LISTA DE ILUSTRAÇÕES

Figura 2 – Padrão MVC	32
Figura 3 - Processo Forlogic Versão 2.2.....	76

LISTA DE TABELAS

Tabela 1 - Tempo estimado x Tempo trabalhado	70
Tabela 2 - Tabela de contagem de Pontos por Função	79
Tabela 3 - Cronograma Inicial.....	80

LISTA DE ABREVIATURAS E SIGLAS

CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
JS	<i>JavaScript</i>
MVC	<i>Model-View-Controller</i>
SQL	<i>Structured Query Language</i>
UTFPR	Universidade Tecnológica Federal do Paraná
XML	<i>eXtensible Markup Language</i>
SGBD	Sistema Gerenciador de Banco de Dados
MS	Microsoft
ALI	Arquivos Lógicos Internos
IE	<i>Input Externo</i>
OE	<i>Output Externo</i>

SUMÁRIO

1. INTRODUÇÃO	68
2. JUSTIFICATIVA	69
3. OBJETIVOS	71
4. ARQUITETURA DO SOFTWARE	74
5. METODOLOGIA.....	76
6. TECNOLOGIAS UTILIZADAS.....	78
7. ESTIMATIVA DE PONTOS POR FUNÇÃO	79
8. CRONOGRAMA.....	80

1. INTRODUÇÃO

A presente proposta de trabalho de conclusão de curso à Universidade Tecnológica Federal do Paraná – UTFPR, campus Cornélio Procópio – tem como finalidade apresentar um projeto de sistema *web* para controle de processo e desenvolvimento de *software* da empresa Forlogic *Software*.

A Forlogic *Software* é uma empresa que está situada na cidade de Cornélio Procópio, desde 18 de Maio de 2004. A empresa atua em 4 áreas distintas, sendo elas, Metrologia, Gestão de documentos, Qualidade e Excelência e Fábrica de *software*.

Com o intuito de aprimorar o processo de desenvolvimento de *software*, a Forlogic *Software*, possui o nível G do MPS Br., e atualmente está em fase de auditoria para a o nível F.

O MPS Br. ou Melhoria de Processo de Software brasileiro é um movimento de melhoria no processo de desenvolvimento apoiado pelo Ministério da Ciência e Tecnologia, voltado para pequenas e médias empresas de desenvolvimento de *softwares*, possuindo sete níveis de maturidade de A a G, sendo G o primeiro nível.

Para o controle de processo de desenvolvimento a empresa em questão conta com a ferramenta Rastro, que é um sistema desenvolvido pela Forlogic *Software*, e fornece suporte tanto para a equipe de desenvolvimento quanto para as pessoas responsáveis pela gerencia de empresa.

A equipe de desenvolvedores faz uso da ferramenta, para designar quais tarefas devem ser executadas dentro da construção de um produto, em contrapartida a gerencia, tem um maior controle das atividades e dos produtos, que estão em desenvolvimento na empresa.

Para Sommerville processo de desenvolvimento é “*um conjunto de atividades e resultados associados que produzem um produto de software*”, sendo assim o processo de desenvolvimento, é um conjunto de atividades e regras a serem seguidas para que no final de um projeto o sistema saia de acordo com o especificado.

Existem no mercado atual algumas ferramentas, que compartilham algumas funcionalidades parecidas com a do Rastro, mas nenhuma que agrupe todas as funcionalidades em uma única ferramenta tem o exemplo do sistema Trac *Project Manegement*, que é um sistema *Open Source* para controle de versionamento de um

projeto, e Visual Studio Team System que é um sistema desenvolvido pela Microsoft, que faz o acompanhamento do ciclo de vida de um produto.

Nenhuma das ferramentas se adapta as necessidades da empresa, pois as mesmas só trabalham com tarefas a serem executadas, e não atendem aos requisitos mínimos de controle de processo de desenvolvimento de *software* estipulados pelo MPS Br.

Segundo Pressman (2006), o processo de desenvolvimento é importante porque fornece estabilidade, controle de organização para uma atividade que pode, se deixada sem controle, tornar-se bastante caótica.

Tendo em vista que o controle de processo é preciso, foi de interesse comum dentro da empresa a construção de um sistema que fizesse o controle, para que não ocorram riscos durante um projeto e para atender a normas estipuladas pelo MPS br.

O principal objetivo deste trabalho é a construção de um módulo administrativo para o sistema Rastro, que ofereça uma estrutura base para a adaptação dos módulos principais do processo, e Manutenção.

No capítulo 2 justifica-se a necessidade de desenvolvimento do *software*, o capítulo 3 descreve os objetivos gerais e específicos, no capítulo 4 discute-se a arquitetura de *software* escolhida para construção, no capítulo 5 a metodologia abordada para o processo de desenvolvimento, o capítulo 6 descreve as tecnologias utilizadas em todo o processo, no capítulo 7 faz-se estimativa de prontos por função e finalmente, no capítulo 8 apresenta-se o cronograma de todo o trabalho.

2. JUSTIFICATIVA

Neste capítulo será abordado um pouco sobre a justificativa, do porquê desenvolver o sistema. O sistema Rastro encontra-se implantado na empresa Forlogic Software, mas não está se adequando 100% ao que a empresa necessita.

Com o passar do tempo à empresa foi estruturando melhor o seu processo de desenvolvimento, com o isso a ferramenta foi ficando desatualizada. Para fazer com que a ferramenta se adequasse ao processo da empresa, foram feitas algumas atualizações. Mas essas atualizações acabaram dificultando futuras modificações.

Tendo como base o relatório da última versão que foi gerado, podemos perceber que a dificuldade na manutenção do sistema foi alta.

Resumo:

	Tempo Estimado	Tempo Trabalhado
Tickets:	190h 13min	307h 27min
Outros Apontamentos:		02h 13min
Total:	190h 13min	309h 40min

Tabela 4 - Tempo estimado x Tempo trabalhado

Como podemos constatar ao analisar a tabela houve uma estimativa de desenvolvimento de 190 horas, para a construção da nova versão, mas o tempo de construção acarretou em 309 horas de trabalho, uma diferença de 60% a mais do que o previsto.

Durante o desenvolvimento foram encontrados alguns erros no sistema que dificultaram o desenvolvimento do mesmo, aumentando assim o tempo da construção.

Com as dificuldades encontradas durante o desenvolvimento, foi constatado que se novas versões do sistema fossem criados, mais dificuldades seriam encontrados, com isso, foi de comum acordo entre a diretoria da empresa, para que o sistema fosse reconstruído.

Como a diretoria decidiu pela reconstrução, foi pedido pelos mesmo que fosse colocado juntamente ao Rastro o módulo de suporte. A empresa possui um setor de suporte ao cliente. Os colaboradores do suporte possuem uma ferramenta denominada Chamados, e foi desenvolvida pela ForLogic, para controlar os pedidos de manutenção ou melhorias no sistema. Com o passar do tempo foi verificado que os funcionários que utilizavam o sistema Chamados, tinham que apontar o tempo trabalhado no próprio sistema e no Rastro, tendo assim um trabalho dobrado, com essa análise foi verificado a necessidade de unificar o sistema.

Outro motivo para a construção desse sistema, é a atualização da tecnologia, atualmente o sistema Rastro é Asp.net WebForm, com essa nova proposta de desenvolvimento iremos passar o sistema para o framework Asp.net MVC 4 da Microsoft.

3. OBJETIVOS

Nesse capítulo será abordado os objetivos gerais e específicos, e o que se espera com a construção do sistema.

3.1. OBJETIVOS GERAIS

Como objetivo principal deste trabalho é a reestruturação do sistema, de modo que o mesmo se adapte as novas mudanças do processo, e assim se adapte ao modelo do MPS Br.

O sistema atual é dividido em dois módulos, o administrativo, que é a parte gerencial do sistema, como por exemplo, cadastro de novos usuários, controle de permissão, etc..., e o módulo de produtos, onde se controla o processo, e as tarefas a serem feitas, em um produto.

Com a nova proposta de construção do software, o sistema irá se dividir em três módulos, Administração, Produtos e Manutenção. Neste trabalho será planejado e desenvolvido somente o módulo de administração.

3.2. OBJETIVOS ESPECIFICOS

- Estruturar o Sistema:

Refere-se à padronização do sistema, constituir a base e como irá funcionar o sistema, tal como o *login*, controle de sessão, *cookies*, padrão de nomenclatura. Desenvolver a base do projeto, para que todo o sistema seja construído a partir dele.

- Gerenciar Usuários:

Gerencia para o cadastro de usuários. Onde serão cadastrados os usuários que terão acesso ao sistema. Atualmente nesta gerência existem informações que não são utilizados, nos quais eles serão retirados neste projeto e será acrescentado um tipo de usuário, que poderá ser desenvolvimento, suporte ou cliente que servirá como filtro.

- Gerenciar Clientes:

Gerência para cadastro de clientes da empresa, para que possa ser referenciada no Projeto, para identificar, quais são os projetos de cada cliente. A única alteração que esta gerência será removida algumas informações que não são usados, e será feito um cadastro mais detalhados, como a inserção de cnpj, razão social, para fins de relatório.

- Gerenciar Produto:

Gerência para cadastrar os produtos que são desenvolvidos dentro da empresa. Na versão antiga do rastro esta gerencia tinha o nome de Projeto, mas depois da atualização e estruturação do processo, foi definido que a terminologia correta seria Produto.

- Tipo de Produto:

Gerência para cadastro de tipos de projetos, por exemplo, se é produto que será desenvolvido pra fábrica de software ou se é produto interno, a gerência servirá para associar um tipo de projeto a um produto, para que posteriormente possa facilitar a criação de relatórios.

- Gerenciar Processos:

Gerência para cadastro de tipo de processo será para ditar qual processo a versão de um projeto irá utilizar, sendo assim será necessário informar quantos e quais fases o processo terá, e quais são os requisitos e tickets padrões. Atualmente não existe, ela será feita, pois na versão atual do sistema, o processo é ligado diretamente ao projeto. Serão baseadas em scripts de sql, para fazer inserções de tickets, fases, e requisitos padrões de uma versão.

- Palavras – Chave:

Gerência para cadastro de palavras chaves, que será usada em todos os projetos, elas servirão para identificação dos artefatos, de uma versão, para que assim possa ser criado o relatório de Matriz Vertical, na qual identifica qual requisito interferiu em qual artefato. Para que possa identificar quais os artefatos modificados ou criados, será usado o log de *commit*, gerado pela ferramenta de SVN.

- Gerenciar Artefatos:

Gerência de artefatos para edição ou remoção de artefatos cadastrados no sistema.

Os artefatos serão cadastrados através da gerencia de tickets, que não estará presente neste escopo, mas a finalidade um artefato, é verificar qual tarefa executada, modificou algum conteúdo do projeto.

- Gerencia de relatórios:

Como os relatórios que o sistema utiliza são construídos a partir da ferramenta Stimulsoft, esta gerencia será utilizada, para fazer o upload dos arquivos que a ferramenta cria. Está gerencia não existe na versão atual do sistema.

- Consultas:

Esta funcionalidade servirá para construir qualquer tipo de consulta referente ao sistema, onde será preciso informar somente a sql, para que a informação seja mostrada em uma listagem, e não será modificado do jeito que é tratada.

- Acompanhamento:

Será uma gerencia, com listagem de todos os apontamentos de tempos em um ticket, de acordo com cara usuário do sistema, esta gerencia não existe na versão atual do sistema, e ela tem o intuito, de fazer o acompanhamento de todos os tickets que o usuário apontou tempo, para saber o desenvolvimento do seu trabalho dentro da empresa.

- Treinamento:

Está funcionalidade tem a finalidade de fazer o cadastro, e avaliação, de treinamentos disponibilizados pela empresa para os seus funcionários.

Sempre que houver um treinamento interno, será cadastrado no sistema, e todos os funcionários, que são usuários do sistema, irão fazer a avaliação do treinamento.

O intuito dessa funcionalidade é medir a capacidade e a competência de cada funcionário.

- Gerenciar Papéis

Cadastros para associar ao usuário e a uma versão, os papéis, servem para definir as ações que cada usuário poderá executar dentro da execução de uma versão.

- Gerenciar Perfis

Esta gerencia atualmente não existe no sistema, ela servirá para associar permissões ao usuário, para que assim cada usuário tenha acesso a determinadas funcionalidades do sistema, no módulo de administração.

- Gerenciar Treinamentos

Esta gerencia não existe no sistema atual, ela será desenvolvida, para controlar os treinamento e workshops que a empresa disponibiliza, para que assim possa ser feito a matriz de competência do colaborador. Para a funcionalidade também será preciso à criação de dois relatórios, relatório geral com todas as respostas de um treinamento, e um relatório com a resposta do usuário específico.

4. ARQUITETURA DO SOFTWARE

Nesse capítulo será apresentado o padrão de arquitetura, que será utilizado na construção do sistema.

Nas palavras de Metsker “Um padrão é uma maneira de fazer algo, ou de buscar um objetivo. Tal ideia se aplica a cozinhar, fazer fogos de artifício, desenvolver software e qualquer outro ofício.” (METSKER, 2004, p.17).

Para este projeto em questão será utilizado o padrão de codificação MVC – Model View Controller – para que as camadas do sistema fiquem bem definidas, e para que em alguma eventualidade se for necessário uma manutenção, fica mais fácil de localizar os problemas e suas respectivas soluções.

Para GAMMA o “MVC é composto por três tipos de objetos. O modelo é o objeto de aplicação, a vista é a apresentação na tela e o controlador define a maneira como a interface do usuário reage às entradas do mesmo. Antes do MVC, os projetos de interface para o usuário tendiam em agrupar esses objetos. MVC para aumentar a flexibilidade e a reutilização.” (GAMMA et al. 2000, p. 20).

O MVC divide o sistema em 3 camadas Modelo (Model), Visão(View), Controlador (Controller).

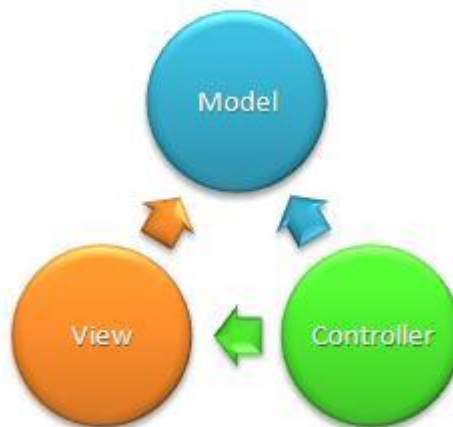


Figura 15 – Padrão MVC

Modelo

O modelo (model) é utilizado para manipular informações de forma mais detalhada, sendo recomendado que, sempre que possível, se utilize dos modelos para realizar

consultas. É o modelo que tem acesso a toda e qualquer informação sendo essa vinda de um banco de dados, arquivo XML.

Visão

A visão (view) é responsável por tudo que o usuário final visualiza, toda a interface, informação, não importando sua fonte de origem, é exibida graças a camada de visão.

Controlador

O Controlador (controller) é responsável por controlar todo o fluxo de informação que passa pelo site/sistema. É na controladora que decide “se”, “o que”, “quando” e “onde” deve funcionar. Define quais informações devem ser geradas, quais regras devem ser acionadas e para onde as informações devem ir.

5. METODOLOGIA

Neste capítulo será abordada a metodologia de desenvolvimento do Projeto Rastro.

5.1. CICLO DE VIDA

Segunda PRESSMAN “O modelo de ciclo de vida é um dos elementos essenciais utilizados no desenvolvimento de software, pois ajuda a garantir que os objetivos finais de cada atividade realizada durante o desenvolvimento sejam alcançados. A escolha de um modelo adequado ao projeto tem grande influência sobre o sucesso de produção. Alocando os processos que serão utilizados no projeto em um ciclo de vida, é possível aumentar a velocidade de desenvolvimento, melhorar a qualidade do produto e dos processos, localizar e controlar pontos de melhoria, minimizar riscos, melhorar a comunicação com os clientes, manter custos e prazos estipulados no cronograma.” (PRESSMAN, 2002).

O ciclo de vida utilizado será o Processo ForLogic Versão 2.5, que é baseado no ciclo de vida Iterativo e Incremental.

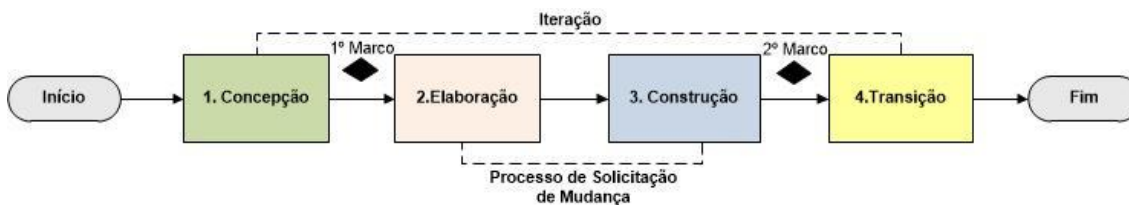


Figura 16 - Processo Forlogic Versão 2.5

Fonte: Forlogic Software 2013.

Como podemos ver na figura anterior o ciclo de vida é dividido em quatro fases, Concepção, Elaboração, Construção e Transição, havendo dois grandes marcos, um entre a concepção e a elaboração, que serve para repassar a equipe de desenvolvimento o que será desenvolvida neste projeto, e outra na transição, que tem como objetivo, recapitular tudo o que foi feito, para ver se tudo ocorreu de acordo, com o projetado.

Durante a Concepção é feito o levantamento de requisitos, análise dos possíveis riscos, e prototipação de como irá funcionar o sistema, depois de escrito o documento de escopo, é repassado ao analista de requisito, que não pode ser o líder do projeto, para ser feito a avaliação dos requisitos.

Na Elaboração será criada a estrutura de armazenamento, onde o código fonte, documentação e diagramas ficarão armazenados, depois de criado a estrutura, será necessário cadastrar os requisitos e tarefas no sistema Rastro, para que os

desenvolvedores possam seguir o escopo. Nesta fase ocorre o primeiro grande marco, onde é feita a reunião com a equipe de desenvolvimento, para expor o escopo do projeto e como será o padrão e linguagem de desenvolvimento adotado.

A fase de Construção é onde ocorre a codificação e testes do sistema proposto, durante essa fase o líder do projeto foca na ocorrência dos riscos e foco da equipe no projeto, para que o projeto não atrase ou fique inviável, se algum risco ocorre, cabe ao líder procurar a melhor solução. Para fazer essas verificações de foco no projeto e riscos, uma vez por semana durante a fase de construção são feitos marcos intermediários.

A fase de transição só poderá ocorrer, se toda a codificação e testes forem finalizados. Nesta fase ocorre o segundo grande marco, que também se caracteriza em uma reunião com todo grupo de desenvolvimento, para ser feita uma retrospectiva dos fatos que ocorrerão durante o desenvolvimento, também será apresentado o resultado final do produto. Depois de feita a reunião o líder do projeto, apresenta ao cliente ou fornecedor de requisitos, como o sistema ficou depois de pronto, para ter sua aprovação.

6. TECNOLOGIAS UTILIZADAS

Neste capítulo serão listadas as tecnologias e ferramentas.

6.1. TECNOLOGIAS

- **ASP.net MVC**

O ASP.NET MVC é uma implementação da arquitetura MVC para o ASP .NET em um framework com o objetivo de criar aplicações WEB no padrão MVC.

- **SQL Server 2012**

O Microsoft SQL Server é um SGBD - sistema gerenciador de Banco de dados relacional criado pela Microsoft em parceria com a Sybase em 1988 e inserido como produto complementar do Windows NT.

- **Ext JS 4.1**

É um framework, que é uma biblioteca JavaScript cross-browser, que provê ao desenvolvedor uma ferramenta para criar interfaces de sistemas *web-based* com alta performance. Atualmente o ExtJS é suportado oficialmente pelos navegadores Internet Explorer 6+, FireFox 1.5+ (PC, Mac), Safari 3+ e Opera 9+ (PC, Mac).

6.2. FERRAMENTAS

- **MICROSOFT VISUAL STUDIO 2010**

A Microsoft Visual Studio é um pacote de programas da Microsoft para desenvolvimento de software especialmente dedicado ao NET Framework e às linguagens, Visual Basic (VB), C, C++, C# (C Sharp) e J#. Também é um grande produto de desenvolvimento na área web, usando a plataforma do ASP. NET. As linguagens com maior frequência nessa plataforma são: VB. NET e o C#.

- **MICROSOFT MANAGEMENT STUDIO 2008**

O Microsoft SQL Server 2008 Management Studio Express (SSMSE) é um ambiente de desenvolvimento integrado para acessar, configurar, gerenciar e desenvolver todos os componentes do SQL Server 2008 R2, ou versões anteriores.

- **FIREBUG 1.10**

Firebug é uma ferramenta integrada com o com ferramentas de desenvolvimento na que funciona enquanto está navegando. Sendo possível editar, depurar e monitorar CSS, HTML e JavaScript em qualquer página da web.

7. ESTIMATIVA DE PONTOS POR FUNÇÃO

A estimativa de pontos de função tem como objetivo mensurar o tamanho e a dificuldade em um projeto, para que assim a equipe de desenvolvimento, possa saber estimar corretamente o projeto.

Para efetuar o cálculo dos pontos, foi utilizado o Diagrama de Entidade Relacionamento, que se encontra no Anexo A.

Neste trabalho o cálculo de pontos por função tem como objetivo, preencher um requisito do trabalho de conclusão, pois dentro do meu ciclo de vida não há necessidade dos pontos por função.

O cálculo de estimativa e dificuldade do sistema no ciclo de vida que irei utilizar é através de bases históricas de desenvolvimento.

Segue a tabela com a contagem de pontos por função:

ALI		
Baixa	29	203
Média	0	0
Alta	0	0
IE		
Baixa	8	24
Média	3	12
Alta	0	0
OE		
Baixa	1	3
Média	1	5
Alta	0	0
	Total	248

Tabela 5 - Tabela de contagem de Pontos por Função

8. CRONOGRAMA

Neste capítulo será apresentado o cronograma inicial o projeto, destacando quais serão os principais marcos do projeto.

	Fevereiro	Março	Abril	Maió	Junho	Julho	Agosto	Setembro
Proposta do Trabalho	█	█	█					
Concepção			█					
Elaboração				█				
Construção				█	█	█		
Transição							█	
Escrita da Proposta			█	█	█	█	█	█
Apresentação da Proposta								█

Tabela 6 - Cronograma Inicial

Como mostrado no cronograma inicial, após a aprovação da proposta de trabalho de conclusão de curso será dado início ao processo de desenvolvimento.

A concepção do processo se dará início no dia na terceira semana de abril, tendo previsão de duração de uma semana. Terminado a concepção será iniciada a elaboração, que terá que o prazo de três dias para ser finalizado.

A fase de construção do sistema ocorrerá em nos meses de maio, junho e finalizando no final de julho. No término da construção será dado início a transição que poderá durar uma semana, que será na primeira semana de agosto.

A construção do trabalho de conclusão de curso será feito desde o início do processo de desenvolvimento, mas terá um foco maior no final da fase de construção.

Para finalizar a apresentação do trabalho e do sistema construído se dará na semana do dia 09 de setembro de 2013.

REFERÊNCIAS

Forlogic Software, Guia do Ciclo de Vida: Versão 2.5. Cornélio Procópio, 2013.

Forlogic Software, Procedimentos do Processo 4L: Versão 2.5. Cornélio Procópio, 2013.

PRESSMAN, R.S., **Engenharia de Software**. 5ªed. Rio de Janeiro: McGraw-Hill, 2002.

SOFTEX, MPS – **Melhoria de Processo de Software e Serviços**. Agosto de 2012.

GAMMA, Erich et al. **Padrões de Projeto: Soluções reutilizáveis de software Orientado a Objetos**. Porto Alegre: Bookman, 2000.

GAMMA, Erich, **Padrões de Projeto**. Porto Alegre: Bookman, 2009.

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Addison Wesley, 2003.

METSKER, Steven John. **Padrões de projeto em Java**. Porto Alegre: Bookman, 2004.

LADAS, C. **Scrumban: Essays on Kanban Systems for Lean Software Development**. USA: Modus Cooperandi Inc, 2008. p.177.

APÊNDICE F – Diagrama Entidade Relacionamento

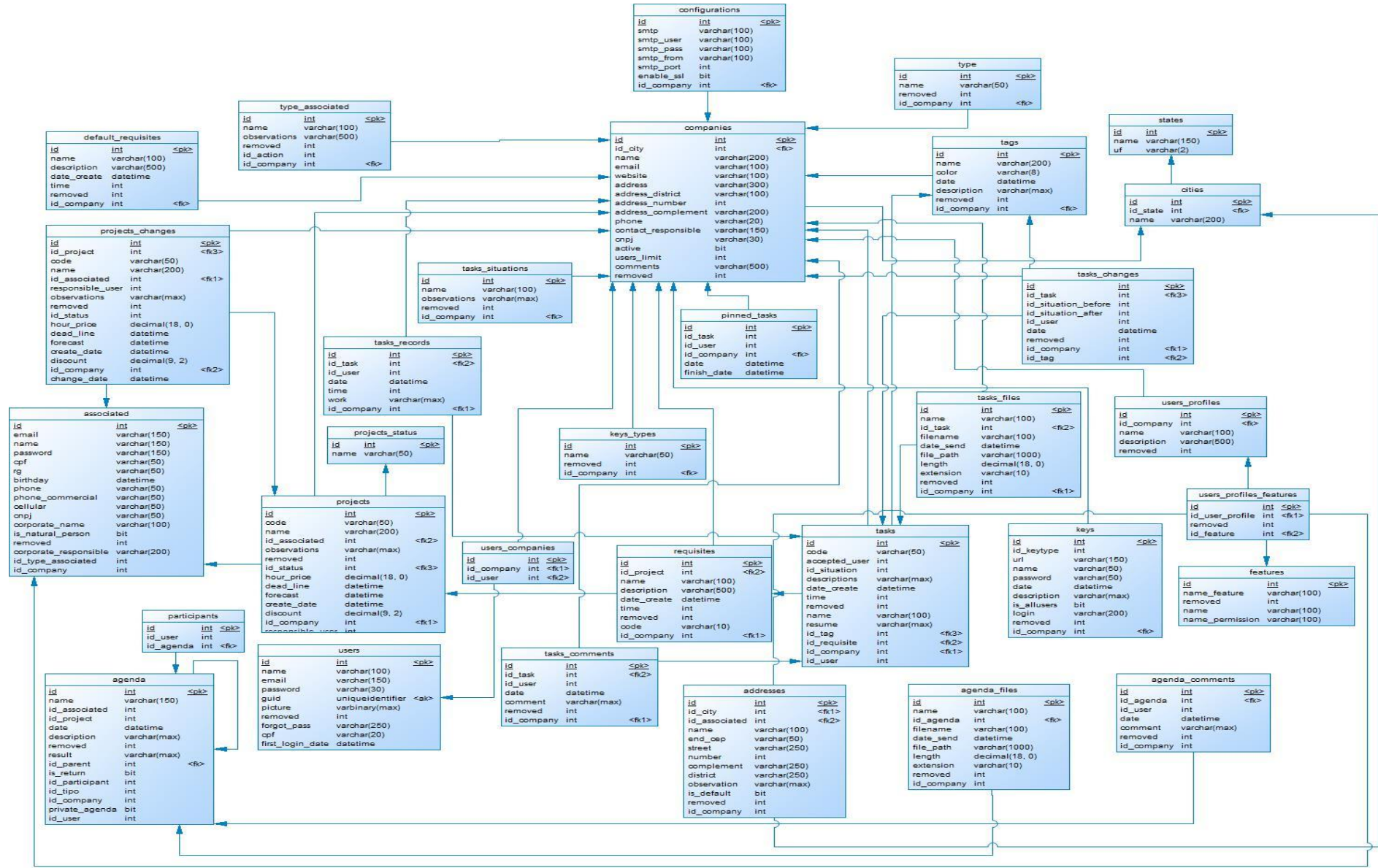
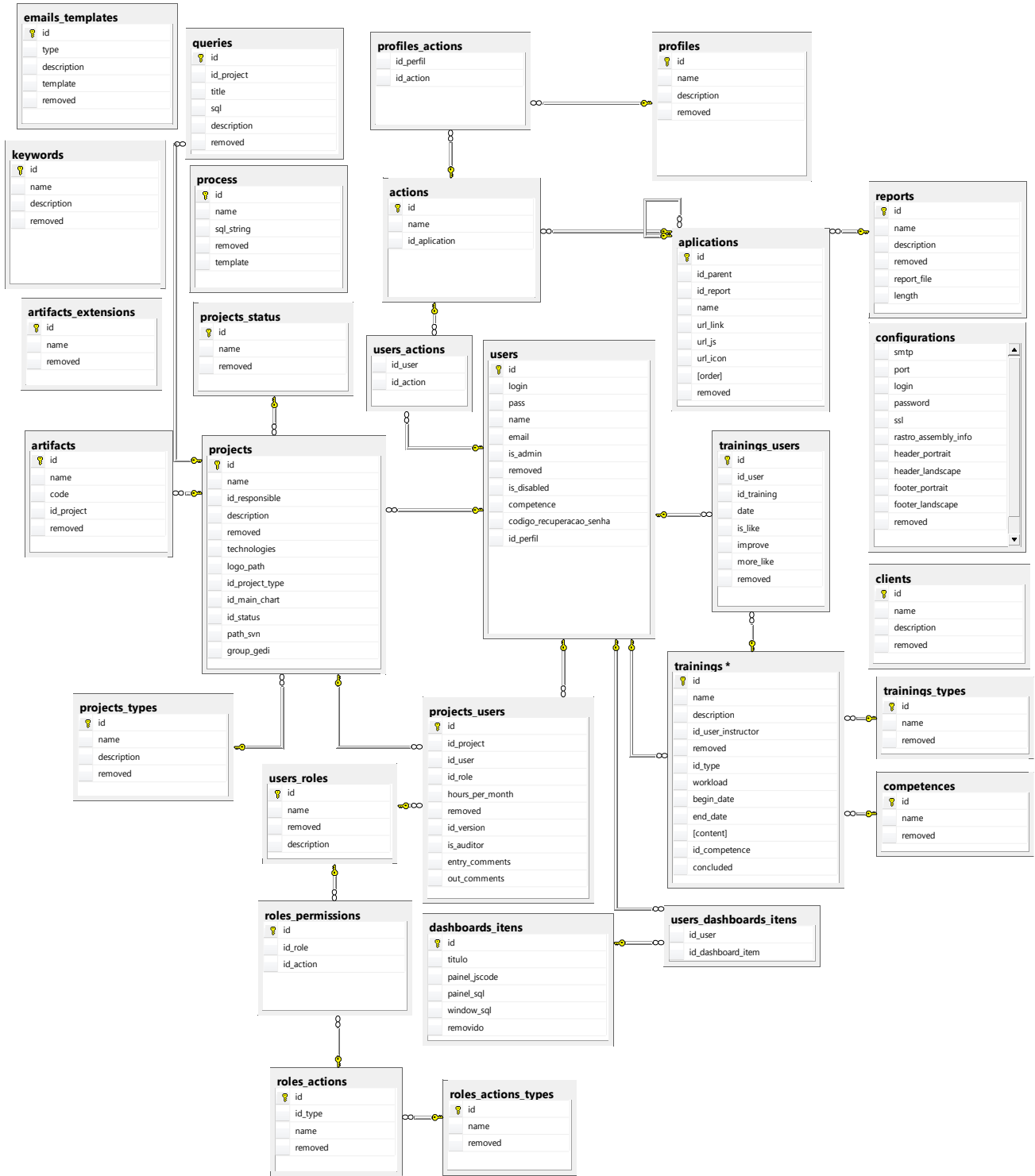


Figura 15 - Diagrama Entidade Relacionamento

ANEXO A – Diagrama Entidade Relacionamento.



ANEXO A – Tarefas Agrupadas pelos Estados

Kontroller2 gestor de projetos

Kreea Sistemas Web

Leonardo Fiewski Soares
leonardo@kreea.net

Meus Registros Alterar Senha Configurações Sair

Dashboard Administração Agenda Financeiro Projetos

Tarefas

Procurar: Filtar Projeto: Dashboard de Projetos

Requisito: [Todos] Tag: [Todos]

Nova Editar Remover Editar Projeto Requisitos Ações

Código	Nome	Requisito	Resumo	Data	Andamento	Usuário	Tempo Est.	Extras
Situação: Done								
T0005	Gráfico de Projetos/Status	Projetos	Gráfico de Projetos/Status	23/01/2014 11:36	188.00%		05h 00min	
T0002	Gerencia de Status	Status	Criação da gerencia de status, novo, editar e remover. O cadastro deve conter Id, ...	23/01/2014 11:12	112.00%		05h 00min	
T0001	Criação da base de dados	Estrutura	Criação da estrutura da base de dados	22/01/2014 12:56	285.00%		02h 00min	
Total							12h 00min	
Situação: To Do								
T0006	Gerencia de Perfis de Usuário	Usuário	Criação da gerencia de perfis do sistema juntamente com a estrutura de permiss...	22/01/2014 19:17	105.00%		06h 00min	
T0004	Gerencia de Usuarios	Usuário		14/01/2014 00:53	50.00%		05h 00min	
T0003	Gerencia de Projetos	Projetos	Desenvolvimento da gerencia de projetos, será feito as funcionalidades novo, edit...	13/01/2014 10:22	0.00%		06h 00min	
Total							17h 00min	

Projeto: Dashboard de Projetos **Cliente:** SICOOB **Descrição:** Projeto da fábrica, com a ideia de criar um sistem... **Status:** Em Desenvolvimento

Sua empresa possui 6 de 100 usuários ativos. [Gerenciar](#)

O Kontroller é um produto ©2014 Kreea [Termos e Privacidade](#)

Dúvidas? Entre em contato conosco: contato@kreea.net - [Manual do Kontroller](#)

Figura 16 - Exemplo tarefas agrupadas

ANEXO B – Avaliações Realizadas do Questionário

AVALIAÇÃO DE USABILIDADE E
FUNCIONALIDADE DE SISTEMA KONTROLLER

NOME: Diogo Cezar Teixeira Batista

IDADE:27

CARGO: Diretor de Criação

ROTEIRO DE TAREFAS:

- 1 - ACESSAR O SISTEMA;
- 2 - ACESSAR O MENU DE PROJETOS;
- 3 - CADASTRAR UM NOVO PROJETO;
- 4 - CADASTRAR UM NOVO REQUISITO;
- 5 - CADASTRAR 3 NOVAS TAREFAS NO PROJETO RECÉM CRIADO;
- 6 - ACEITAR UMA NOVA TAREFA;
- 7 - REGISTRAR TEMPO GASTO EM UMA TAREFA;

QUESTIONÁRIO

1 – Você teve alguma dificuldade em acessar o sistema?

R: Não.

2 – O menu de projetos estava em fácil localização?

R: Sim.

3 – Você teve alguma dificuldade em cadastrar um novo projeto? Se sim, quais foram as dificuldades?

R: Não houve dificuldade, o sistema apresenta uma interface e intuitiva.

4 – Você teve alguma dificuldade em cadastrar um novo requisito? Se sim, quais foram as dificuldades?

R: Não houveram dificuldades em cadastrar um novo requisito, principalmente por dois motivos: A replicação do menu dentro de cada projeto e a possibilidade de se importar requisitos.

5 - Teve alguma dificuldade em cadastrar uma nova tarefa? Se sim, quais foram as dificuldades e quais sugestões você poderia dar para melhorar?

R: Não, todas as tarefas foram cadastradas com sucesso. Uma sugestão seria uma possível importação de tarefas que ocasionalmente são anotadas em txt's ou arquivos do Excel.

6 - Alguma dificuldade em encontrar o botão para aceitar uma tarefa?

R: Não.

7 - Você teve alguma dificuldade em registrar o tempo gasto em uma tarefas? Se sim, quais foram elas e quais melhorias você sugere?

R: Não houve dificuldade, a interface é intuitiva, e apresenta recursos que facilitam a contagem do tempo.

8 - Depois de ter utilizado, você teria alguma sugestão ou crítica sobre o sistema?

R: Uma sugestão no apontamento de tempo é recuperar quanto tempo a usuário ficou sem apontar, ou seja a diferença entre agora e o último apontamento, facilitando a contagem de tempo.

AValiação de Usabilidade e
Funcionalidade de Sistema Kontroller

NOME: Edilson Nabhen Filho

IDADE:24

CARGO: Desenvolvedor

ROTEIRO DE TAREFAS:

- 1 - ACESSAR O SISTEMA;
- 2 - ACESSAR O MENU DE PROJETOS;
- 3 - CADASTRAR UM NOVO PROJETO;
- 4 - CADASTRAR UM NOVO REQUISITO;
- 5 - CADASTRAR 3 NOVAS TAREFAS NO PROJETO RECÉM CRIADO;
- 6 - ACEITAR UMA NOVA TAREFA;
- 7 - REGISTRAR TEMPO GASTO EM UMA TAREFA;

QUESTIONÁRIO

1 – Você teve alguma dificuldade em acessar o sistema?

R: Não.

2 – O menu de projetos estava em fácil localização?

R: Sim.

3 – Você teve alguma dificuldade em cadastrar um novo projeto? Se sim, quais foram as dificuldades?

R: Não, o sistema é bem intuitivo e de fácil entendimento.

4 – Você teve alguma dificuldade em cadastrar um novo requisito? Se sim, quais foram as dificuldades?

R: Não, fácil entendimento com botões e submenus.

5 - Teve alguma dificuldade em cadastrar uma nova tarefa? Se sim, quais foram as dificuldades e quais sugestões você poderia dar para melhorar?

R: Não.

6 - Alguma dificuldade em encontrar o botão para aceitar uma tarefa?

R: Não.

7 - Você teve alguma dificuldade em registrar o tempo gasto em uma tarefas? Se sim, quais foram elas e quais melhorias você sugere?

R: Não, uma combo específica e de fácil entendimento para esse quesito.

8 - Depois de ter utilizado, você teria alguma sugestão ou crítica sobre o sistema?

R: Sistema muito útil e de fácil compreensão.

AValiação de Usabilidade e
Funcionalidade de Sistema Kontroller

NOME: Carlos Henrique Prado Sousa

IDADE:22

CARGO: Diretor de Projetos

ROTEIRO DE TAREFAS:

- 1 - ACESSAR O SISTEMA;
- 2 - ACESSAR O MENU DE PROJETOS;
- 3 - CADASTRAR UM NOVO PROJETO;
- 4 - CADASTRAR UM NOVO REQUISITO;
- 5 - CADASTRAR 3 NOVAS TAREFAS NO PROJETO RECÉM CRIADO;
- 6 - ACEITAR UMA NOVA TAREFA;
- 7 - REGISTRAR TEMPO GASTO EM UMA TAREFA;

QUESTIONÁRIO

1 – Você teve alguma dificuldade em acessar o sistema?

R: Não.

2 – O menu de projetos estava em fácil localização?

R: Sim.

3 – Você teve alguma dificuldade em cadastrar um novo projeto? Se sim, quais foram as dificuldades?

R: Tive dificuldade apenas para entender para que servia alguns campos. Mas no geral foi fácil.

4 – Você teve alguma dificuldade em cadastrar um novo requisito? Se sim, quais foram as dificuldades?

R: Não.

5 - Teve alguma dificuldade em cadastrar uma nova tarefa? Se sim, quais foram as dificuldades e quais sugestões você poderia dar para melhorar?

R: Não, a tela de cadastro é bem fácil e intuitiva.

6 - Alguma dificuldade em encontrar o botão para aceitar uma tarefa?

R: Não, mas ele poderia estar fora do menu de ações, facilitando o acesso.

7 - Você teve alguma dificuldade em registrar o tempo gasto em uma tarefas? Se sim, quais foram elas e quais melhorias você sugere?

R: Não. Registre o tempo sem nenhum problema.

8 - Depois de ter utilizado, você teria alguma sugestão ou crítica sobre o sistema?

R: O sistema é muito bom e útil para o dia a dia da empresa. A gestão de projetos será mais fácil com a utilização dele.

AVALIAÇÃO DE USABILIDADE E
FUNCIONALIDADE DE SISTEMA KONTROLLER

NOME: Paola Caroline Kishi Fredegoto

IDADE:24

CARGO: Comercial

ROTEIRO DE TAREFAS:

- 1 - ACESSAR O SISTEMA;
- 2 - ACESSAR O MENU DE PROJETOS;
- 3 - CADASTRAR UM NOVO PROJETO;
- 4 - CADASTRAR UM NOVO REQUISITO;
- 5 - CADASTRAR 3 NOVAS TAREFAS NO PROJETO RECÉM CRIADO;
- 6 - ACEITAR UMA NOVA TAREFA;
- 7 - REGISTRAR TEMPO GASTO EM UMA TAREFA;

QUESTIONÁRIO

1 – Você teve alguma dificuldade em acessar o sistema?

R: Não.

2 – O menu de projetos estava em fácil localização?

R: Estava.

3 – Você teve alguma dificuldade em cadastrar um novo projeto? Se sim, quais foram as dificuldades?

R: Nenhum problema, sistema é bem intuitivo.

4 – Você teve alguma dificuldade em cadastrar um novo requisito? Se sim, quais foram as dificuldades?

R: Consegui cadastrar requisitos facilmente.

5 - Teve alguma dificuldade em cadastrar uma nova tarefa? Se sim, quais foram as dificuldades e quais sugestões você poderia dar para melhorar?

R: É bem tranquilo cadastrar tarefas no Kontroller.

6 - Alguma dificuldade em encontrar o botão para aceitar uma tarefa?

R: Nenhuma.

7 - Você teve alguma dificuldade em registrar o tempo gasto em uma tarefas? Se sim, quais foram elas e quais melhorias você sugere?

R: Me perco na hora de saber quanto tempo eu gastei na tarefa, mas para registrar é super fácil.

8 - Depois de ter utilizado, você teria alguma sugestão ou crítica sobre o sistema?

R: Sistema, fácil de usar e bem intuitivo.