

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**SISTEMA PARA GERENCIAMENTO DE ATIVIDADES**  
**UTILIZANDO A TÉCNICA KANBAN APLICADO A**  
**DESENVOLVIMENTO DE SOFTWARE**

**GUSTAVO BAEHR**

**BLUMENAU**  
**2016**

**GUSTAVO BAEHR**

**SISTEMA PARA GERENCIAMENTO DE ATIVIDADES  
UTILIZANDO A TÉCNICA KANBAN APLICADO A  
DESENVOLVIMENTO DE SOFTWARE**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Samuel Cristhian Schwebel, Mestre - Orientador

**BLUMENAU  
2016**

**SISTEMA PARA GERENCIAMENTO DE ATIVIDADES  
UTILIZANDO A TÉCNICA KANBAN APLICADO A  
DESENVOLVIMENTO DE SOFTWARE**

Por

**GUSTAVO BAEHR**

Trabalho de Conclusão de Curso aprovado  
para obtenção dos créditos na disciplina de  
Trabalho de Conclusão de Curso II pela banca  
examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Samuel Cristhian Schwebel, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Everaldo Artur Grahl, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Paulo Silva, Doutor – FURB

Blumenau, 04 de Julho de 2016

## RESUMO

O desenvolvimento de software vem evoluindo a cada dia com a criação de novas tecnologias e métodos, visando alcançar produtos com melhor qualidade e em menor tempo. Para isso é preciso uma ferramenta de apoio à gestão, capaz de permitir visualizar o processo como um todo. Este trabalho apresenta um sistema web para acompanhamento de projetos ágeis de desenvolvimento de software, baseado no método *Kanban* e nos princípios defendidos pelo *Manifesto for Agile Software Development*, com o intuito de disponibilizar um ambiente onde todos os envolvidos possam visualizar o trabalho em progresso e evidenciar os problemas antes de finalizar o processo. Para isso, foi realizado um estudo destes conceitos e práticas utilizadas neste método. O sistema foi desenvolvido utilizando a linguagem COS através do *framework* IB Tech Caché, o *framework* jQuery e banco de dados Caché. Como resultado destaca-se a facilidade em visualizar o trabalho em processo e a adequação do sistema aos processos da empresa Adapcon Processamento de Dados Ltda. alinhados aos princípios e técnicas de métodos ágeis em gerenciamento de projetos.

Palavras-chave: *Kanban*. Métodos ágeis. Caché. Gerenciamento de projetos.

## ABSTRACT

The software development is evolving every day with the creation of new technologies and methods in order to achieve products with best quality and shortest time. This requires a support tool management able to allow you to view the process as a whole. This job presents a web system for tracking of agile projects software development, based on the Kanban method and the principles defended by the Manifesto for Agile Software Development, in order to provide an environment where everyone involved can view the work in progress and evidence problems before finalizing the process. For this, a study was conducted of these concepts and practices used in this method. The system was developed using the COS language through framework IB Tech Caché, the jQuery framework and Caché database. As a result there is the facility to view the work in progress and suitability of the system for the company Adapcon Processamento de Dados Ltda. processes, aligned with the principles and techniques of agile project management.

Key-words: *Kanban*. Agile methods. Caché. Project management.

## LISTA DE FIGURAS

Figura 1 - Representação dos conceitos de projeto, atividade, recurso e competências dentro do cenário de gerenciamento de projetos.....	14
Figura 2 - Quadro visual utilizado tradicionalmente em projetos ágeis .....	18
Figura 3 – Quadro físico <i>Kanban</i> utilizado pela empresa .....	20
Figura 4 – Cartão utilizado para representar uma ordem de serviço no quadro <i>Kanban</i> .....	21
Figura 5 – Tela referente ao quadro <i>Kanban</i> da ferramenta.....	22
Figura 6 – Tela principal do quadro <i>Kanban</i> da aplicação.....	23
Figura 7 – Quadro <i>Kanban</i> da ferramenta.....	23
Figura 8– Diagrama de casos de uso referentes às operações que o administrador tem no sistema .....	28
Figura 9 – Diagrama de casos de uso referentes às operações que o colaborador tem no sistema .....	28
Figura 10 – Diagrama de casos de uso referentes às operações que o cliente tem no sistema.....	29
Figura 11– Tela de <i>login</i> do sistema.....	34
Figura 12 – Tela inicial do sistema.....	34
Figura 13 – Tela para alterar a senha do usuário .....	34
Figura 14 – Tela de cadastro de usuário .....	35
Figura 15 – Cadastro de Competências .....	35
Figura 16 – Cadastro de tipos de serviço.....	36
Figura 17 – Cadastro de recurso .....	36
Figura 18 – Cadastro de colaborador (dados gerais) .....	37
Figura 19 – Cadastro de colaborador (competências) .....	38
Figura 20 - Cadastro de clientes (dados gerais).....	39
Figura 21 – Cadastro de clientes (observações) .....	39
Figura 22 – Cadastro de clientes (contatos).....	40
Figura 23 – Cadastro de clientes (consulta).....	40
Figura 24 – Geração de projeto (dados gerais).....	41
Figura 25 – Geração de projeto (tarefas) .....	42
Figura 26 – Geração de projeto (lançamentos por tarefa) .....	42
Figura 27 – Geração de projeto (histórico projeto) .....	43
Figura 28 – Consulta análise de projetos.....	44

Figura 29 – Quadro <i>Kanban</i> .....	45
Figura 30 – Detalhamento da tarefa a partir do quadro <i>Kanban</i> .....	46
Figura 31 – Tela de apontamento de horas realizadas .....	46
Figura 32 – Tela de permissões de acesso de usuário por página .....	47

## LISTA DE QUADROS

Quadro 1 – Comparação entre os trabalhos correlatos .....	24
Quadro 2 - Requisitos funcionais .....	27
Quadro 3 - Requisitos não funcionais .....	27
Quadro 4 – Código fonte da página CSP utilizando o framework ICF.....	31
Quadro 5 – Código fonte de acessos a métodos Caché via componentes Ajax do framework ICF .....	32
Quadro 6 – Código fonte do método <i>drag and drop</i> .....	33
Quadro 7 – Comparação entre os trabalhos correlatos e o sistema desenvolvido.....	48
Quadro 8 – Descrição do caso de uso UC09 .....	53
Quadro 9 – Descrição do caso de uso UC10 .....	54
Quadro 10 – Descrição do caso de uso UC16 .....	55
Quadro 11 – Descrição do caso de uso UC13 .....	55
Quadro 12 – Descrição do caso de uso UC12 .....	56



## LISTA DE ABREVIATURAS E SIGLAS

CHA – Conhecimentos, Habilidades e Atitudes

COS – *Caché Object Script*

CSP – *Caché Server Pages*

DOM – *Document Object Model*

HTML – *Hyper Text Markup Language*

ICF – IB Tech Caché

OS – Ordem de Serviço

UML – *Unified Modeling Language*

WIP – *Work In Process*

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>10</b>
1.1 OBJETIVOS.....	12
1.2 ESTRUTURA.....	12
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>13</b>
2.1 GERENCIAMENTO DE PROJETOS .....	13
2.1.1 Conceitos de competências .....	14
2.2 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE .....	15
2.2.1 Métodos ágeis .....	16
2.2.2 Kanban .....	17
2.3 SISTEMA ATUAL .....	19
2.4 TRABALHOS CORRELATOS .....	21
<b>3 DESENVOLVIMENTO .....</b>	<b>25</b>
3.1 LEVANTAMENTO DE INFORMAÇÕES .....	25
3.2 ESPECIFICAÇÃO .....	26
3.2.1 Requisitos funcionais .....	26
3.2.2 Requisitos não funcionais .....	27
3.2.3 Diagrama de Casos de Uso .....	27
3.3 IMPLEMENTAÇÃO .....	29
3.3.1 Técnicas e ferramentas utilizadas.....	29
3.3.2 Operacionalidade da implementação .....	33
3.4 RESULTADOS E DISCUSSÕES.....	47
<b>4 CONCLUSÕES.....</b>	<b>49</b>
4.1 EXTENSÕES .....	49
<b>APÊNDICE A – DESCRIÇÃO DOS PRINCIPAIS CASOS DE USO.....</b>	<b>53</b>

## 1 INTRODUÇÃO

Segundo Prikladnicki, Willi e Milani (2014, p. 1), a indústria de software está em crescente evolução tornando-se uma das mais importantes indústrias da era moderna. Tal fato pode ser observado na presença de software em inúmeras atividades, desde as mais simples, como processar informações básicas de compra e venda, até atividades mais complexas, como controlar robôs, carros e aviões.

Assim, com o passar dos anos a atual sociedade vem tornando-se cada vez mais dependente de software e tecnologia. Neste cenário, a demanda por software tem crescido exponencialmente, tornando o mercado mais competitivo e exigindo que a engenharia de software e o gerenciamento de projetos desenvolvam capacidades de adaptar-se rapidamente a mudanças (BASSI FILHO, 2008).

**Os gerentes de projeto** que atuam na área de desenvolvimento de software frequentemente se veem diante do desafio de liderar projetos com ciclos de entrega agressivos, constante mudança de requisitos, novas tecnologias e sistemas de informação que cada vez mais suportam a tomada de decisão e os processos-chave de negócio das empresas. Em cenários como esses, uma abordagem adaptativa para o processo de desenvolvimento e gerenciamento de projetos se torna fundamental a fim de maximizar a produtividade e não comprometer a qualidade de produto final. (SANTOS, 2008, p. 38, grifo do autor).

A partir dos anos 90, surge uma nova visão sobre como desenvolver software, denominada como métodos ágeis. Ser ágil está associado a uma mudança cultural, a uma nova forma de pensar. Ao contrário dos métodos tradicionais que focam nos processos, os métodos ágeis se diferenciam por ter o enfoque nas pessoas, no seu conjunto de valores, princípios e práticas que possibilitam a adaptação de novos fatores decorrentes do desenvolvimento do projeto e a rápida resposta as constantes mudanças de mercado (PRIKLADNICKI; WILLI; MILANI, 2014, p. 27).

O termo método ágil difundiu-se em fevereiro de 2001 devido a um movimento iniciado pela comunidade internacional de desenvolvimento de sistemas de informação, onde dezessete representantes de diversos métodos ágeis existentes no mercado procuravam definir valores e princípios comuns a todas elas. Deste movimento originou-se o *Manifesto for Agile Software Development* (AGILE MANIFESTO, 2001).

Com o surgimento dos métodos ágeis, ferramentas e técnicas de apoio começaram a ser utilizadas no gerenciamento de projetos, dentre as quais destaca-se a utilização do *Kanban*. Nessa técnica, as equipes posicionam quadros em áreas visíveis da sala de projeto, preenchendo-os com cartões que sinalizam os itens de trabalho selecionados para uma dada

iteração. Normalmente, os cartões são posicionados conforme seu estado presente: não iniciado, em andamento e finalizado (PRIKLADNICKI; WILLI; MILANI, 2014, p. 142).

*Kanban* é uma das técnicas usadas para implementar o conceito de *Lean Manufacturing* (Produção enxuta), onde a saída de produtos acabados, ao final da linha de montagem, dita o ritmo da introdução de matéria prima no sistema. Desta forma evita-se o acúmulo de produtos inacabados ao longo da linha, diminuindo a quantidade de Trabalho em Processo *Work In Process* (WIP). Com menos produtos intermediários tem-se uma sobrecarga menor no sistema e pode-se adaptar melhor e mais rápido às mudanças na demanda dos clientes (DEBATIN, 2011, p. 12).

O gerenciamento de projetos é um conjunto de técnicas e ferramentas gerenciais que permitem a organização desenvolver um conjunto de habilidades, conhecimentos e competências, dentro de uma sequência clara e lógica de eventos com início, meio e fim, apoiados a parâmetros de custo, recursos e qualidade predeterminados (VARGAS, 2005). Porém, as organizações de desenvolvimento de software preocupam-se na maioria das vezes com problemas técnicos, dedicando poucos recursos e tempo para o lado gerencial, sendo que a correlação de investimentos com projetos tornou-se algo fundamental para que a organização cresça e seja competitiva no mercado (PAULA FILHO, 2003).

Partindo deste cenário, a empresa Adapcon Processamento de Dados Ltda. implantou conceitos de métodos ágeis, em meados de 2014, visando organizar processos e melhorar o gerenciamento de seus projetos. Porém, a empresa não possui um sistema de informação adequado para o gerenciamento ágil de projetos. Os controles dos projetos são feitos a partir da criação de Ordens de Serviço (OS) onde contêm as seguintes informações: número da OS, data de emissão, cliente, orçamento, solicitante, responsável, descrição, data de previsão e estimativa de horas.

Este modelo de gerenciamento de projetos a partir da criação de uma OS atende perfeitamente demandas pequenas (esforço de trabalho de 1 a 16 horas), pois são tarefas pontuais na qual é possível atribuir apenas um responsável para desenvolver a demanda, possibilitando o controle de produtividade do mesmo. Porém, quando se trata de demandas de médio a grande porte (esforço de trabalho de mais de 16 horas) a OS apresenta algumas carências, como:

- a) não permitir dividir a OS em diversas tarefas podendo atrelar responsáveis diferentes com prazos de entrega diferentes;
- b) não possibilitar a visualização de gargalos do projeto;
- c) não possibilitar o controle eficaz de produtividade dos envolvidos.

Para auxiliar no controle da OS a empresa utiliza um quadro físico *Kanban* como ferramenta para controlar o fluxo de desenvolvimento dos projetos, onde o mesmo é dividido

em cinco fases: entrada, em desenvolvimento, cliente validando, necessidade atendida e necessidade atendida com problema. Porém, muitos dos projetos são desenvolvidos diretamente no cliente o que impossibilita a atualização das fases da OS no quadro *Kanban* e principalmente a visualização de gargalos, refletindo diretamente na tomada de decisão da equipe. Também pelo motivo da OS não representar tarefas específicas, existe uma dificuldade de visualizar quem está fazendo qual item, durante um projeto maior.

Diante disto, duas foram às questões essenciais encaminhadas no contexto deste trabalho. A primeira tratou de alinhar a forma como é feito o controle através de OS incorporando as melhores práticas de gerenciamento de projetos, especialmente nas definições de projetos, tarefas e recursos e respectivas competências. A segunda questão foi implementar estes conceitos em um sistema de informação, considerando a técnica *Kanban* como central para o controle dos projetos, tarefas e recursos, estando assim também alinhada aos métodos ágeis, práticas estas que vem sendo disseminadas na cultura da empresa.

## 1.1 OBJETIVOS

O objetivo geral do trabalho é desenvolver um sistema web para acompanhamento de projetos ágeis de desenvolvimento de software baseado na técnica *Kanban*.

Os objetivos específicos do trabalho são:

- a) substituir o painel físico *Kanban* do sistema atual por uma interface no sistema desenvolvido;
- b) limitar o *Work In Process* (WIP) das tarefas por projeto;
- c) permitir visualizar o progresso do projeto;
- d) permitir distribuir as tarefas de modo mais eficiente.

## 1.2 ESTRUTURA

Este trabalho está disposto em quatro capítulos.

No primeiro capítulo apresenta-se a introdução, os objetivos e a estrutura do trabalho.

No segundo capítulo tem-se a fundamentação teórica, destacando-se os conceitos de cada elemento envolvido neste processo, com ênfase no método *Kanban*, o sistema atual, bem como os trabalhos correlatos.

No terceiro capítulo é apresentado o desenvolvimento do sistema, incluindo detalhes referentes à especificação, implementação e tecnologia utilizada.

No quarto capítulo apresenta-se a conclusão sobre o trabalho, enfatizando os objetivos alcançados, bem como sugestões para trabalhos futuros como extensão deste.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos a serem apresentados nas seções a seguir, tais como gerenciamento de projetos, processo de desenvolvimento de software, sistema atual e os trabalhos correlatos.

### 2.1 GERENCIAMENTO DE PROJETOS

Segundo Martins (2007), o conceito de gerenciamento de projetos é baseado no conjunto de aplicação de conhecimentos, ferramentas, competências e técnicas na construção de atividades relacionadas para atingir requisitos preestabelecidos, com qualidade, custo e prazo definido, através da mobilização de recursos técnicos e humanos.

O gerenciamento de projetos cria um equilíbrio entre as demandas do escopo, tempo, custo, qualidade e bom relacionamento com o cliente. O sucesso no gerenciamento de projetos está relacionado ao alcance de alguns objetivos tais como a entrega dentro do prazo previsto, dentro do custo orçado, com nível de desempenho adequado, aceitação do cliente, atendimento de forma controlada às mudanças de escopo e respeito à cultura da organização (ZIMERMANN, 2012, p. 18).

O controle do projeto é exercido através da comparação entre o que foi realizado com o que foi planejado. A partir disto, é possível analisar e corrigir os desvios encontrados. É necessário um planejamento para estabelecer metas de desempenho, custos e tempo, mantendo o escopo do projeto no nível correto. Sem isso, não há a possibilidade de exercer o controle do projeto (ALDABÓ, 2001).

As empresas que adotam as práticas de gerenciamento de projetos se beneficiam e se tornam cada vez mais competitivas, se destacam no mercado e principalmente, demonstram para os seus clientes que estão organizadas de acordo com as práticas e metodologias reconhecidas internacionalmente para realizar projetos com qualidade, cumprindo o que foi prometido, dentro do que foi orçado e de acordo com o tempo previsto. (VIEIRA, 2003, p. 18).

Um fator importante a ser considerado no gerenciamento de projetos é a implementação de boas práticas as quais são organizadas em um conjunto de áreas de conhecimento, compostas por conceitos, processos, ferramentas e técnicas, entradas e saídas. Dentre alguns conceitos importantes destas boas práticas, vale destacar os seguintes (PMBOK, 2013):

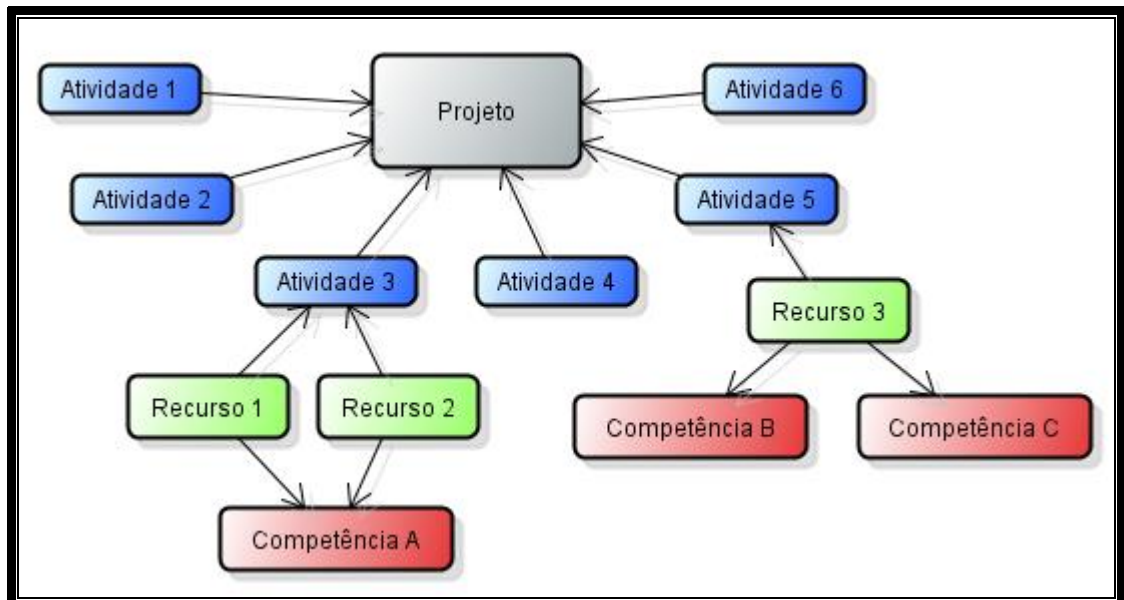
- a) projeto: é um esforço temporário empreendido para criar um produto, serviço ou resultado único. Para tanto, baseia-se essencialmente em um conjunto de tarefas que devem ser realizadas conforme os requisitos estabelecidos;
- b) atividades: por vezes também denominada de tarefas, baseia-se em um trabalho específico de pouca duração, ações específicas a serem realizadas para produzir as

entregas do projeto;

- c) recursos: baseia-se na definição dos recursos e quantidades necessárias para a realização da atividade ou tarefa.

O sucesso do projeto está diretamente relacionado ao planejamento exercido por sua gestão. É a gestão do projeto que vai acompanhar as ações, assim como delegar tarefas e controlar os recursos de acordo com suas competências para que os objetivos sejam atendidos contemplando assim as expectativas da organização. A Figura 1 apresenta os conceitos de projeto, atividade, recurso e competência dentro do cenário de gerenciamento de projetos, sendo que o projeto é um conjunto de atividades realizadas pelos recursos e suas respectivas competências.

Figura 1 - Representação dos conceitos de projeto, atividade, recurso e competências dentro do cenário de gerenciamento de projetos.



### 2.1.1 Conceitos de competências

Segundo Dutra (2012), o conceito de competência foi proposto por David McClelland em 1973, na busca de uma abordagem mais efetiva nos processos de seleção de pessoas para organizações. Competência é uma característica subjacente a uma pessoa que está relacionada ao desempenho superior na execução em determinada situação ou de uma tarefa. Para Chiavenato (1997), competência está relacionada às características intrínsecas da pessoa que contribui para as vantagens competitivas relacionadas ao trabalho, auxiliando no objetivo estratégico do negócio.

Barato (1998) afirma que competências são compreendidas como um conhecimento capaz de produzir determinados desempenhos, marcado geralmente pelas relações de trabalho, cultura da organização, limitações de tempo e recursos etc.

Segundo Rabaglio (2004), possuir competências significa ter conhecimentos, habilidades e atitudes, denominados pela sigla CHA, que sejam compatíveis com o desempenho em determinada situação ou atividade, sendo capaz de colocar em prática esse potencial sempre que necessário. O CHA pode ser dividido em dois grupos:

- a) competências técnicas: referem-se aos conhecimentos e habilidades específicas necessárias para as atribuições ou pré-requisitos que cada função exige, tais como: escolaridade, formação específica, experiências, domínio de idiomas, informática etc;
- b) competências comportamentais: referem-se as atitudes e características pessoais, é o comportamento único de cada indivíduo diante de situações similares, tais como: iniciativa, criatividade, habilidade de relacionamento interpessoal, dentre outros.

O processo de seleção de recurso humano por competências parte das especificações definidas do cargo, função ou tarefa, que consiste em identificar as competências individuais e organizacionais necessárias para cumprir os objetivos preestabelecidos pela organização. Todo este processo tem como consequência um resultado mais eficaz para a organização atendendo as expectativas definidas (ABREU; FREITAS, 2009).

## 2.2 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Um processo de desenvolvimento de software é um modelo que descreve a dinâmica de trabalho a ser seguido pelos desenvolvedores através de uma abordagem organizada, por meio de passos preestabelecidos para atingir um objetivo. Para demonstrar os resultados das diversas fases do processo, utilizam-se representações diagramáticas, como os diagramas de caso de uso da UML (*Unified Modeling Language*) (FURTADO, 2002).

Para Reis (2003, p. 5), um processo de desenvolvimento de software constitui-se em um conjunto de atividades realizadas para construir software, levando em consideração as pessoas envolvidas, as ferramentas utilizadas e os produtos construídos. Desta forma pode-se considerar um processo de desenvolvimento de software como um roteiro dinâmico e iterativo visando produtividade, qualidade e efetividade de projeto.

Embora existam diversos processos de desenvolvimento de software, quatro etapas são fundamentais e comuns a todos os processos (SOMMERVILLE, 2007):



- a) especificação: etapa responsável pela definição dos requisitos e das restrições do software;
- b) desenvolvimento: etapa responsável pelo desenvolvimento do sistema conforme sua especificação;
- c) validação: etapa responsável pela validação das funcionalidades implementadas de acordo com o que foi especificado;
- d) evolução: etapa responsável pela evolução do software conforme a necessidade do cliente.

Como exemplo de técnicas, podem ser utilizadas a Análise Estruturada, a Análise Essencial, a Análise Orientada a Objetos e a UML. Desta forma, o processo é um guia que permite o uso de uma ou várias técnicas por opção dos desenvolvedores do sistema de informação ou software (REZENDE, 2005).

### 2.2.1 Métodos ágeis

Segundo Prikladnicki, Willi e Milani (2014, p. 25), os métodos ágeis foram concretizadas por 17 profissionais da indústria de software, entre eles Ken Schwaber, Mike Beedle, Martin Fowler, Kent Beck e Alistair Cockburn, que estavam desmotivados com os processos de desenvolvimento de software existentes no ano de 2001 e partiram em busca de um novo método. Este método deveria ser capaz de fazer com que as equipes de desenvolvimento respondessem mais rapidamente as mudanças de requisitos, e que os projetos fossem desenvolvidos em menos tempo. Desta reunião originou-se o *Manifesto for Agile Software Development*.

O manifesto ágil consiste em uma declaração com quatro máximas que regem o desenvolvimento ágil, onde passa a valorizar (AGILE MANIFESTO, 2001):

- a) indivíduos e suas interações entre eles mais que processos e ferramentas;
- b) software funcionando mais que documentação abrangente;
- c) colaboração do cliente mais que negociação de contratos;
- d) responder à mudança mais que seguir um plano.

Ao contrário de outras culturas de desenvolvimento, que tem seus requisitos definidos e estáticos, a agilidade não está relacionada à obediência de protocolos preestabelecidos de produção, mas a novos padrões de comportamento e atitude. Portanto, uma equipe não pode se dizer “ágil” se não se comportar assim. Cada método ágil define suas próprias práticas, mas todos compartilham dos valores e princípios postulados pelo Manifesto Ágil. Além das quatro

máximas apresentadas, o manifesto cita estes doze princípios (PRIKLADNICKI; WILLI; MILANI, 2014):

- a) a maior prioridade é satisfazer o cliente, através da entrega contínua e adiantada de software de valor agregado;
- b) aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Os processos ágeis tiram vantagem das mudanças, visando à vantagem competitiva para o cliente;
- c) entregar software funcionando com frequência, na escala de semanas até meses, com preferência à menor escala de tempo;
- d) pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto diariamente, durante todo o projeto;
- e) construir projetos ao redor de indivíduos motivados, dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho;
- f) o método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara;
- g) software funcional é a medida primária de progresso;
- h) processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter um ritmo constante sempre;
- i) contínua atenção à excelência técnica e bom design, aumenta a agilidade;
- j) simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito;
- k) as melhores arquiteturas, requisitos e designs emergem de times auto organizáveis;
- l) em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

O manifesto prega que os projetos deem o suporte adequado às necessidades da equipe, formando profissionais motivados e confiantes, incentivando o trabalho em equipe para tornarem-se mais efetivos, empregando seus esforços em entregar software funcionando. O plano principal são as pessoas, e não o processo em si (PRIKLADNICKI; WILLI; MILANI, 2014, p. 27).

### 2.2.2 Kanban

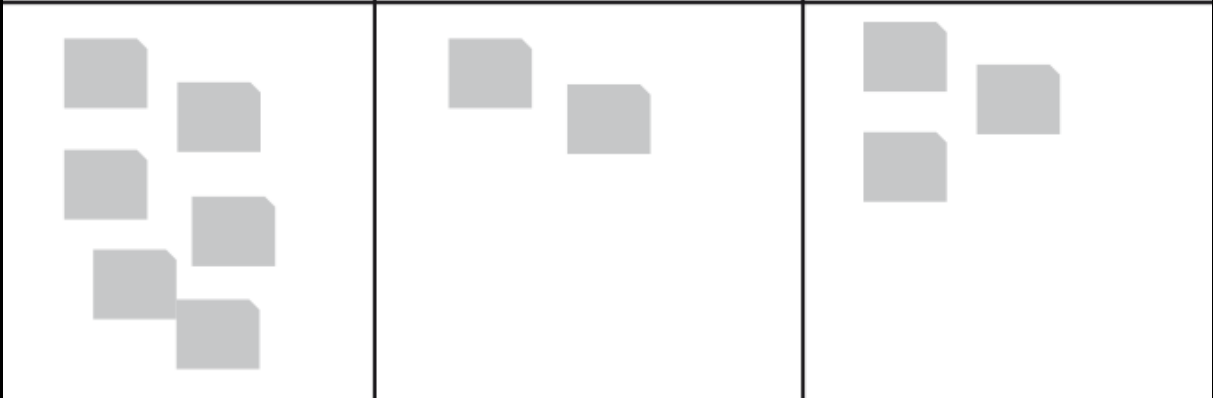
A técnica *Kanban* foi elaborada com o propósito de tornar simples e rápidas as atividades de programação, controle e acompanhamento de sistemas de produção. Foi

projetado para ser usado dentro do contexto da filosofia *Just in Time* (JIT) e busca movimentar e fornecer os itens dentro da produção de acordo com a demanda necessária (TUBINO, 1997).

Na área de desenvolvimento de software, técnicas *Kanban* começaram a ser utilizadas com o surgimento de métodos ágeis, no qual as equipes posicionam quadros em áreas visíveis da sala de projeto, identificam as possíveis fases da iteração como: *backlog* da iteração, em desenvolvimento, em validação e finalizado, e preenchem cada fase com cartões que indicam os itens de trabalho selecionados para uma dada iteração. (PRIKLADNICKI; WILLI; MILANI, 2014, p. 142).

A Figura 1 apresenta o quadro visual *Kanban* utilizado tradicionalmente em projetos ágeis, onde o mesmo está dividido em três fases: *backlog* da iteração, em progresso e pronto.

Figura 2 - Quadro visual utilizado tradicionalmente em projetos ágeis

Backlog da iteração	Em progresso	Pronto
		

Fonte: Prikladnicki, Willi e Milani (2014, p. 143).

*Kanban* é uma técnica baseada em um modelo de aplicação incremental de mudanças cuja finalidade é incorporar alterações no modelo de trabalho visando à transformação cultural de uma organização (PRIKLADNICKI; WILLI; MILANI, 2014). A utilização do quadro *Kanban*, permite visualizar e otimizar os processos existentes, buscando mudar a cultura organizacional, ao invés de apenas substituir processos (ANDERSON, 2010, p. 51).

Segundo Prikladnicki, Willi e Milani (2014, p. 166), o quadro *Kanban* é uma técnica poderosa para despertar uma cultura de melhoria contínua no ambiente de trabalho, pois sua abordagem sistêmica permite que a forma de trabalho passe a ser discutida tão intensamente e frequentemente quanto o próprio trabalho em si, o que é primordial para a formação dessa cultura. Pode-se destacar as seguintes vantagens na utilização de quadros *Kanban* em ambientes de desenvolvimento de software:

- a) proporciona visualizar o processo como um todo;

- b) possibilita limitar a quantidade de trabalho em progresso (WIP);
- c) permite identificar gargalos;
- d) proporciona visualizar impedimentos do projeto antes que aconteçam;
- e) permite distribuir o trabalho de modo mais eficiente;
- f) possibilita discussões frequentes de estratégias de coordenação;
- g) permite detectar medições fora da faixa de previsibilidade.

Assim, o *Kanban* se apoia em um modelo evolucionário, cujo ponto de partida é sempre a forma corrente de execução do trabalho. Ele dá às pessoas o instrumento para observarem o que fazem e pensarem na próxima solução mais adequada ao seu contexto atual. (PRIKLADNICKI; WILLI; MILANI, 2014, p. 142)

### 2.3 SISTEMA ATUAL

Atualmente a empresa possui um sistema para controle de projetos baseado em ordens de serviço, que tem como finalidade, auxiliar a gestão das atividades operacionais e administrativas relacionadas à execução de serviços. As principais funcionalidades do sistema são: manter ordem de serviço, o apontamento de horas gastas por ordem de serviço, manter clientes, manter funcionários e consultar o andamento dos serviços.

Com a implantação da cultura ágil em meados de 2014, foi implantado um quadro físico *Kanban* para auxiliar o controle dos projetos, sendo que cada cartão representa uma OS.

O quadro é dividido em cinco fases, sendo elas:

- a) entrada: é a fase inicial de toda OS criada, sendo que as mesmas são organizadas de acordo com a data de previsão de entrega;
- b) em desenvolvimento: essa fase é organizada por sub níveis de funcionários representando visualmente quais OS estão sendo desenvolvidas e quem está desenvolvendo;
- c) cliente validando: essa fase também é organizada por sub níveis de funcionários representando visualmente quais OS estão em fase de homologação;
- d) necessidade atendida: representa as OS que foram homologadas no cliente;
- e) necessidade atendida com problema: representa as OS que foram homologadas no cliente com retrabalho.

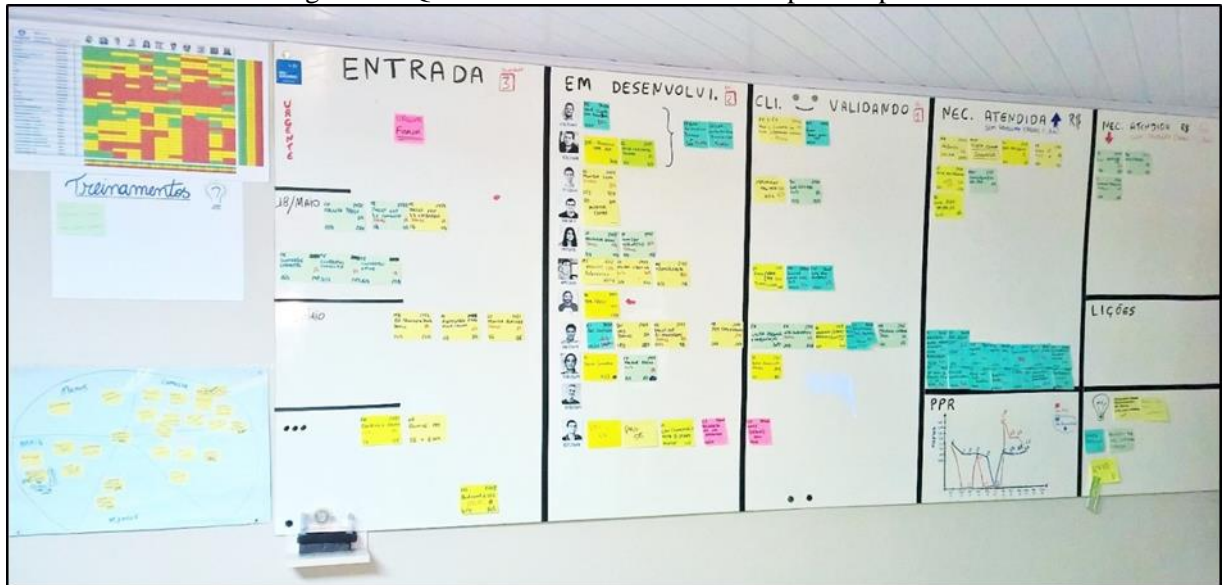
Os cartões são organizados de acordo com as seguintes cores:

- a) verde: representa uma OS com grau de dificuldade baixo;
- b) amarelo: representa uma OS com grau de dificuldade alto;

- c) vermelho: representa uma OS com problema;
- d) azul: representa uma OS em análise para envio de orçamento.

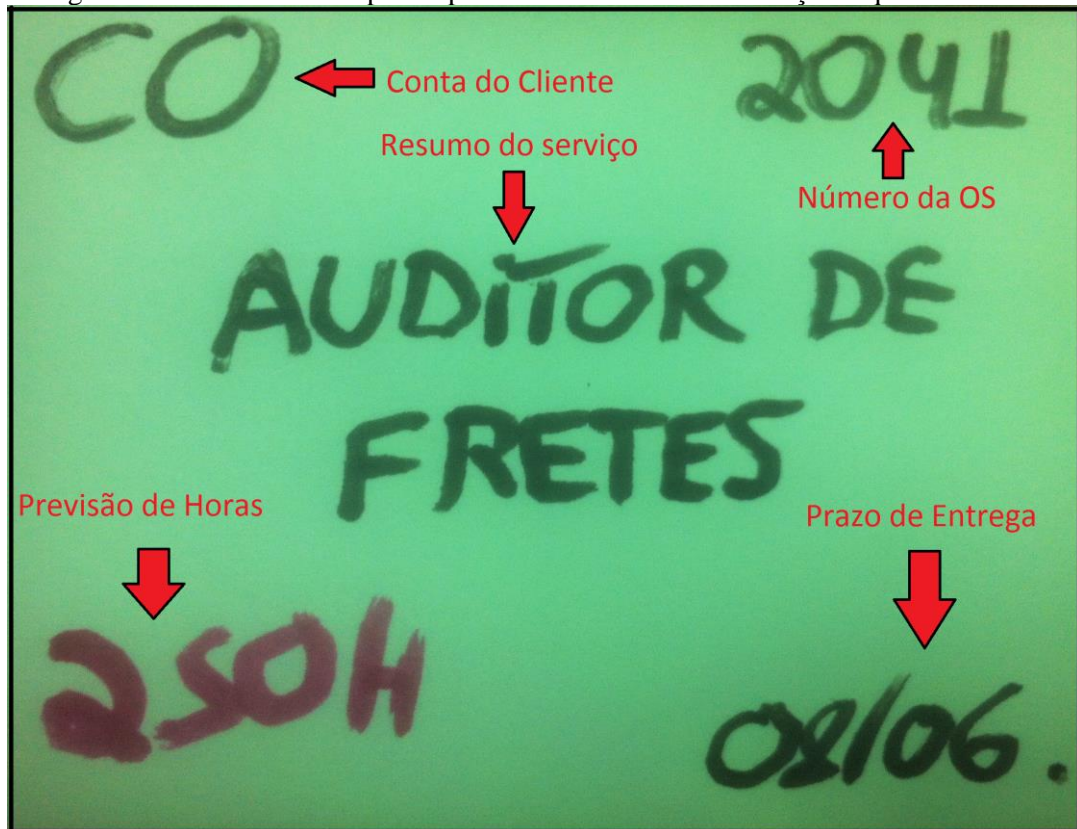
A Figura 2 apresenta o quadro físico *Kanban* utilizado na empresa, sendo que o mesmo está dividido em cinco fases: entrada, em desenvolvimento, cliente validando, necessidade atendida e necessidade atendida com problema.

Figura 3 – Quadro físico *Kanban* utilizado pela empresa



A Figura 3 apresenta o cartão utilizado no quadro físico *Kanban* com as seguintes informações: conta do cliente, prazo de entrega, ordem de serviço, resumo da ordem de serviço e previsão de horas para a realização da tarefa.

Figura 4 – Cartão utilizado para representar uma ordem de serviço no quadro *Kanban*



#### 2.4 TRABALHOS CORRELATOS

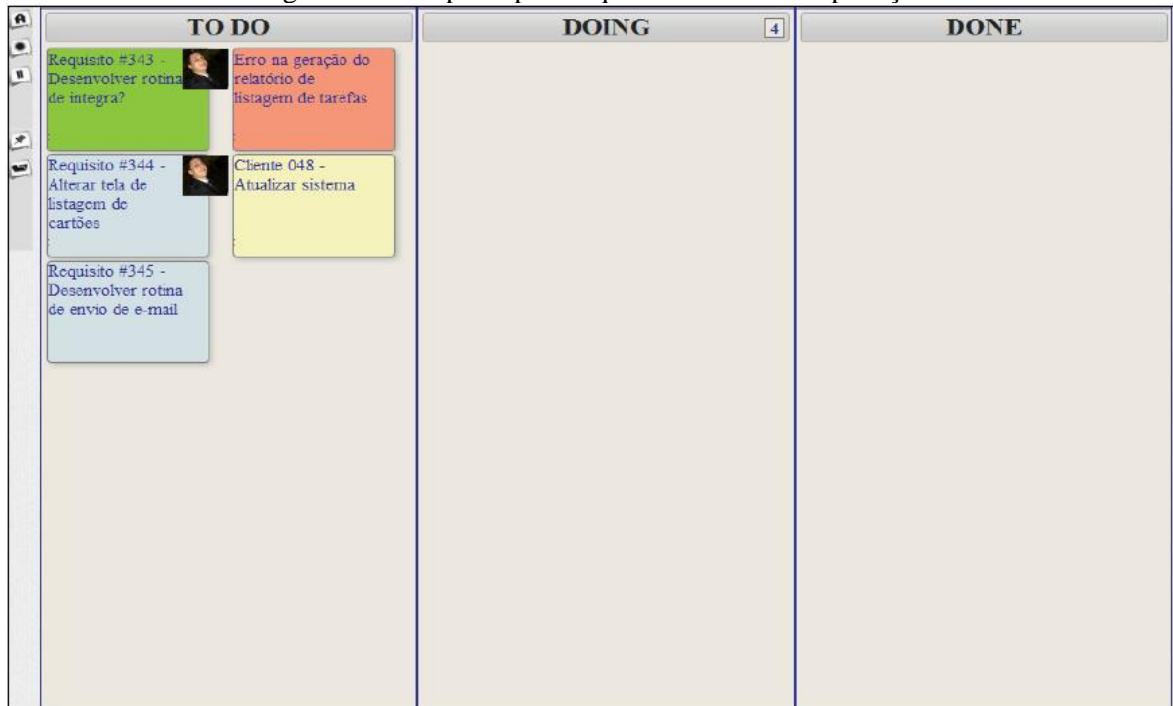
Pode-se citar como trabalho correlato o sistema “Pronto. *agile software development*” desenvolvido por Gomes e Faias Junior (2009) como trabalho de conclusão de curso, sendo que a ferramenta desenvolvida é baseada no método ágil Scrum. O sistema possibilita o gerenciamento e acompanhamento de projetos através do uso de um quadro *Kanban* no qual é possível visualizar as tarefas que devem ser desenvolvidas na forma de cartões em seu estado presente. A Figura 4 apresenta o quadro *Kanban* da ferramenta sendo que é possível visualizar alguns cartões distribuídos pelas quatro colunas: *to do*, *doing*, *testing* e *done*.

Figura 5 – Tela referente ao quadro *Kanban* da ferramenta

Fonte: Gomes e Faias Junior (2009).

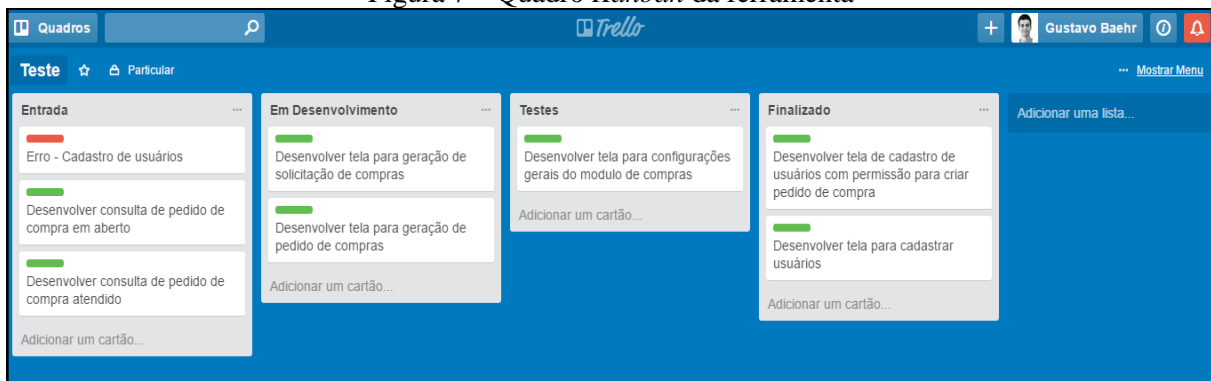
A ferramenta também possibilita a visualização da quantidade de trabalho realizado versus trabalho planejado através da aplicação do gráfico de *burndown*, sendo que o mesmo é um artefato disponível no Scrum que serve como um guia e indicador durante o processo. O gráfico de *burndown* exhibe o conflito entre o que foi planejado e a velocidade real da execução do trabalho.

Outro trabalho correlato que pode-se citar é a aplicação Quick Screen desenvolvida por Debatin (2011) como trabalho de conclusão de curso, sendo que a ferramenta desenvolvida é uma aplicação *web* para auxiliar no desenvolvimento de software, proporcionando um acompanhamento das etapas do processo através da utilização de um quadro *Kanban*. A principal funcionalidade da aplicação é a possibilidade de configurar as colunas do quadro *Kanban* conforme as etapas do processo e de poder definir o WIP das mesmas. Na Figura 5 pode-se visualizar o quadro *Kanban* da aplicação com alguns cartões distribuídos pelas colunas.

Figura 6 – Tela principal do quadro *Kanban* da aplicação

Fonte: Debatin (2011, p. 46).

Pode-se citar o Trello como trabalho correlato por ser uma ferramenta *web* de apoio ao gerenciamento de projetos, no qual permite criar quadros totalmente customizáveis de acordo com as necessidades do usuário. A ferramenta permite criar e movimentar cartões entre as colunas do quadro tornando visíveis as etapas do processo. Os cartões podem receber diversas informações como: etiquetas, responsáveis, listas e imagens, auxiliando na gestão das informações. Na Figura 6 pode-se visualizar um quadro *Kanban* com alguns cartões divididos entre etapas previamente cadastradas.

Figura 7 – Quadro *Kanban* da ferramenta

Fonte: Trello (2016).

O Quadro 1 apresenta uma comparação entre os trabalhos correlatos citados baseados nas principais funcionalidades de cada ferramenta.



Quadro 1 – Comparação entre os trabalhos correlatos

<b>Funcionalidades / Características</b>	<b>Pronto</b>	<b>Quick Screen</b>	<b>Trello</b>
Plataforma web	Sim	Sim	Sim
Relatórios estatísticos	Sim	Sim	Não
Flexibilidade na escolha do processo	Não	Sim	Sim
Histórico de movimentação de cartões	Sim	Sim	Sim
Gráfico <i>burndown</i>	Sim	Não	Não
Língua portuguesa	Sim	Sim	Sim

### 3 DESENVOLVIMENTO

Neste capítulo estão descritas as especificações e detalhamento do sistema desenvolvido, apresentando as suas características, requisitos funcionais, requisitos não funcionais, diagramas de caso de uso, tecnologias usadas, a operacionalidade do sistema e ao final os resultados e discussão.

#### 3.1 LEVANTAMENTO DE INFORMAÇÕES

O sistema desenvolvido contempla aspectos que visam maximizar os resultados da empresa, bem como melhorar o sistema atual alinhando-o com os objetivos da abordagem ágil auxiliando a empresa na tomada de decisão, sendo que através da análise dos trabalhos correlatos e a vivencia no setor de desenvolvimento de software foi possível realizar o levantamento da funcionalidade do sistema.

O sistema desenvolvido é executado na web e auxilia o gerenciamento de desenvolvimento ágil de software, proporcionando um ambiente onde todos os envolvidos possam visualizar o acompanhamento das etapas do processo possibilitando evidenciar os problemas antes de finalizar o processo. Para isso baseia-se na técnica *Kanban*, por ser uma ferramenta sistêmica de melhoria contínua que permite além de visualizar o processo com um todo, limitar a quantidade de trabalho em progresso (WIP), identificar gargalos e impedimentos, fomentando discussões frequentes de estratégias de coordenação entre os envolvidos no processo.

A principal funcionalidade do sistema é o quadro *Kanban* moldado conforme as etapas existentes no processo de desenvolvimento de projetos da empresa, sendo que pode-se definir um WIP por projeto restringindo o número de cartões que cada etapa pode receber simultaneamente. Assim, o quadro *Kanban* dá ao gestor do projeto um instrumento capaz de avaliar quais etapas do processo estão sobrecarregadas ou ociosas, auxiliando-o a realizar mudanças e ajustes antes de finalizar o processo, buscando manter o fluxo contínuo de cartões sobre o quadro.

Os cartões representam as tarefas que devem ser executadas pela equipe para cumprir com o objetivo final do projeto, sendo que todos os usuários com acesso ao quadro podem assumir a responsabilidade do cartão, movimenta-lo sobre o quadro e registrar o apontamento de horas realizadas sobre os trabalhos desenvolvidos. Além destas funcionalidades, o sistema grava um histórico com todas as ações realizadas na evolução do projeto que podem ser consultadas posteriormente.

O sistema conta com três níveis de acesso ao sistema, sendo eles: Administrador, Colaborador e Cliente. O Administrador tem acesso a todas as funcionalidades do sistema, geralmente é o usuário responsável pela gestão do processo. O Colaborador é o usuário responsável pela execução dos trabalhos descritos nos cartões e seu acesso está restrito as funcionalidades disponíveis no quadro. O Cliente é o usuário final do processo, quem irá receber o produto desenvolvido, sendo que será responsável pelo acompanhamento de seus projetos e seu acesso também está restrito as funcionalidades disponíveis no quadro.

## 3.2 ESPECIFICAÇÃO

Esta seção descreve os Requisitos Funcionais (RF), que representam as funcionalidades e o comportamento que o sistema deve possuir, os Requisitos Não Funcionais (RNF), que representam as restrições que o sistema terá sobre alguns serviços ou funções oferecidas como usabilidade, navegabilidade, portabilidade, segurança e hardware, bem como os diagramas de casos de uso desenvolvidos para o sistema. A especificação foi realizada a partir da ferramenta Astah Community, utilizando a linguagem de modelagem UML na elaboração dos diagramas de casos de uso.

### 3.2.1 Requisitos funcionais

O Quadro 2 apresenta os requisitos funcionais previstos para o sistema e sua rastreabilidade, ou seja, vinculação com os casos de uso associados.

Quadro 2 - Requisitos funcionais

Requisitos Funcionais	Caso de Uso
RF01. O sistema deverá permitir a manutenção de usuários.	UC01
RF02. O sistema deverá permitir o usuário efetuar o <i>login</i> (autenticação) no sistema.	UC02
RF03. O sistema deverá permitir o usuário alterar senha do <i>login</i> (autenticação).	UC03
RF04. O sistema deverá permitir o administrador controlar as permissões dos usuários no sistema.	UC04
RF05. O sistema deverá permitir a manutenção de colaboradores.	UC05
RF06. O sistema deverá permitir a manutenção de clientes.	UC06
RF07. O sistema deverá permitir a manutenção de tipos de serviço.	UC07
RF08. O sistema deverá permitir a manutenção de competências.	UC08
RF09. O sistema deverá permitir a manutenção de projetos	UC09
RF10. O sistema deverá permitir a manutenção de tarefas.	UC10
RF11. O sistema deverá permitir a manutenção de recursos.	UC11
RF12. O sistema deverá permitir um ou mais colaboradores efetuarem lançamentos de horas realizadas nas tarefas.	UC12
RF13. O sistema deve permitir a manutenção e movimentação do cartão referente à tarefa no quadro <i>Kanban</i> .	UC13
RF14. O sistema deverá permitir configurar por projeto o WIP das colunas do quadro <i>Kanban</i> .	UC09
RF15. O sistema deverá eleger os melhores recursos para assumirem a responsabilidade da tarefa, considerando as competências do colaborador frente às necessidades da tarefa.	UC13
RF16. O sistema deverá permitir ao colaborador assumir a responsabilidade do cartão.	UC13
RF17. O sistema deverá permitir consultar o histórico de ações realizadas no projeto.	UC14
RF18. O sistema deverá permitir consultar o apontamento de horas realizadas por tarefa do projeto.	UC15
RF19. O sistema deverá permitir consultar análise de projeto.	UC16

### 3.2.2 Requisitos não funcionais

O Quadro 3 apresenta os requisitos não funcionais previstos para o sistema.

Quadro 3 - Requisitos não funcionais

Requisitos Não Funcionais
RNF01. O sistema deverá utilizar o banco de dados Caché.
RNF02. O sistema deverá utilizar a linguagem de programação Caché Object Script (COS).
RNF03. O sistema deverá ser desenvolvido utilizando o <i>framework</i> IB Tech Caché (ICF).
RNF04. O sistema deverá utilizar a linguagem de programação web Caché Server Pages (CSP).
RNF05. O sistema deverá ser compatível com o navegador Chrome.
RNF06. O sistema deverá utilizar recursos jQuery.

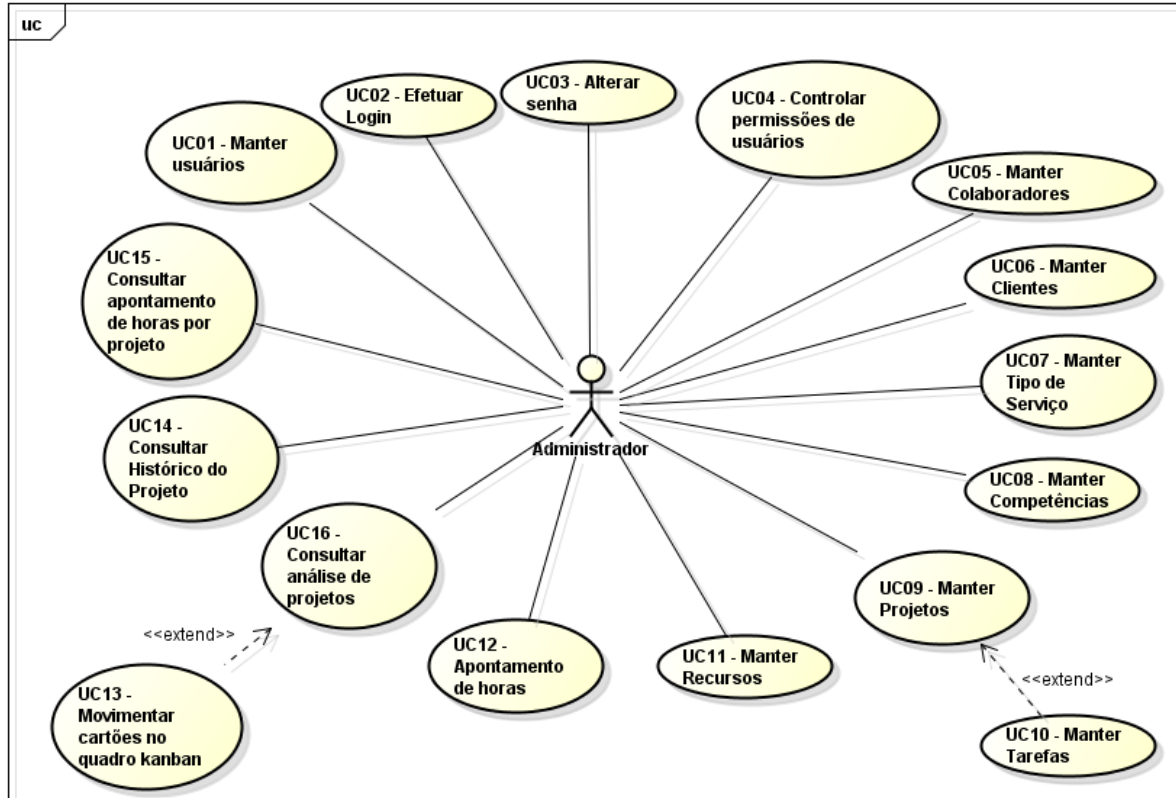
### 3.2.3 Diagramas de Casos de Uso

Os diagramas de casos de uso indicam as funcionalidades do sistema de acordo com as permissões de cada usuário cadastrado. Os casos de uso estão distribuídos em três áreas e representam as funcionalidades que cada usuário tem acesso, de acordo com seu tipo de perfil,

os quais são: colaborador, cliente e administrador. O detalhamento dos principais casos de uso encontra-se no Apêndice A.

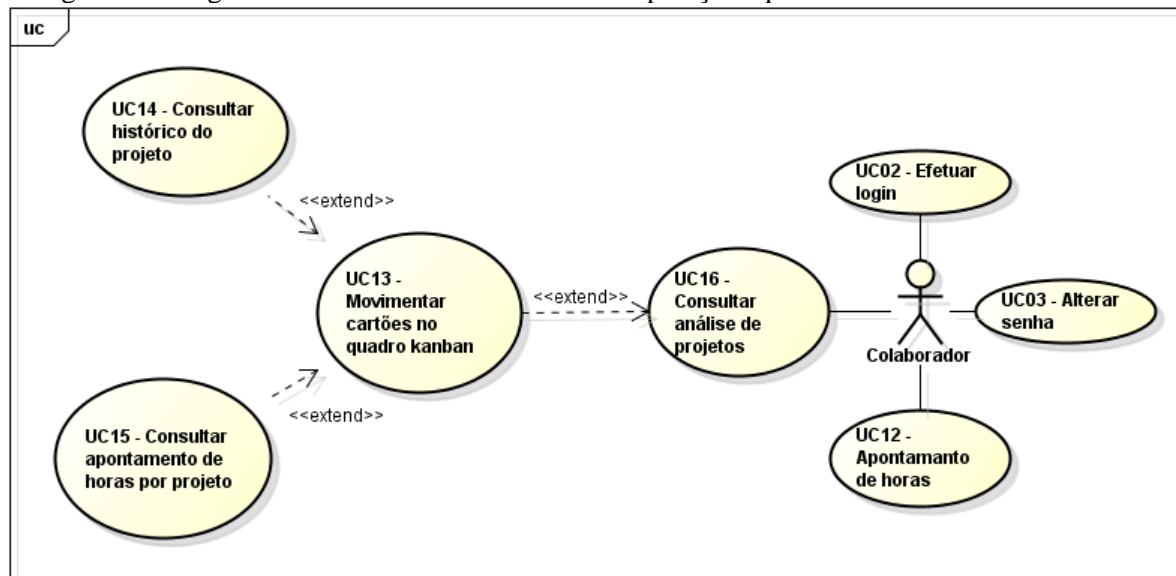
Na figura 7 pode-se ver a representação dos casos de uso referentes às operações que o ator administrador tem no sistema.

Figura 8– Diagrama de casos de uso referentes às operações que o administrador tem no sistema



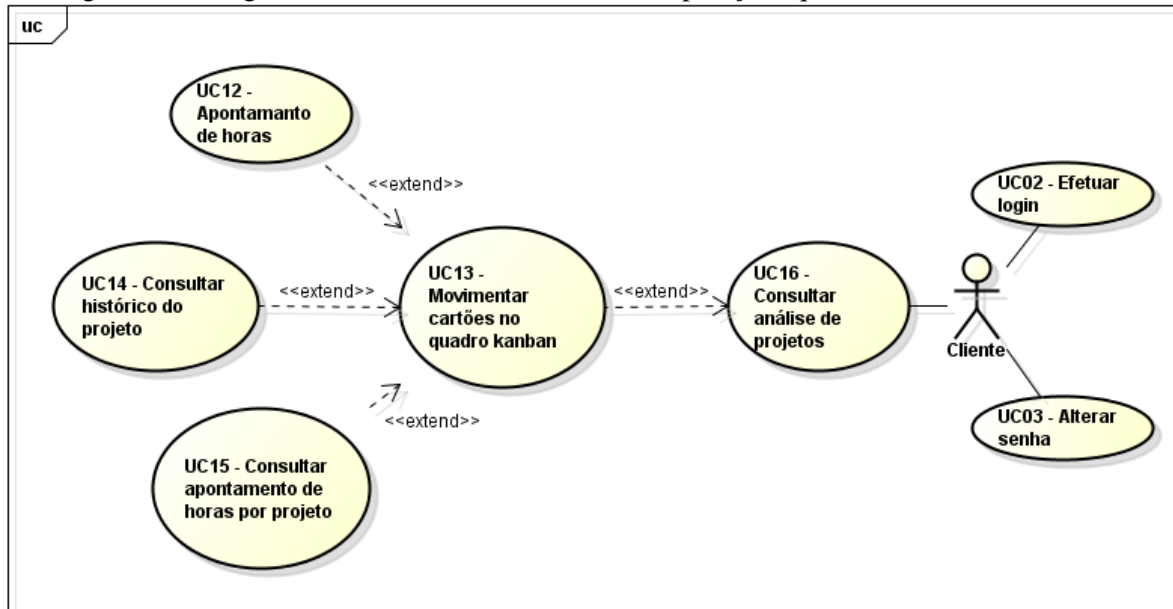
Na figura 8 pode-se ver a representação dos casos de uso referentes às operações que o ator colaborador tem no sistema.

Figura 9 – Diagrama de casos de uso referentes às operações que o colaborador tem no sistema



Na figura 9 pode-se ver a representação dos casos de uso referentes às operações que o ator cliente tem no sistema.

Figura 10 – Diagrama de casos de uso referentes às operações que o cliente tem no sistema



### 3.3 IMPLEMENTAÇÃO

Nesta seção estão apresentadas informações sobre as ferramentas e técnicas utilizadas no desenvolvimento do sistema, juntamente com a operacionalidade da implementação.

#### 3.3.1 Técnicas e ferramentas utilizadas

O sistema foi desenvolvido utilizando a linguagem COS através do *framework* IB Tech Caché, *framework* jQuery e banco de dados Caché.

##### 3.3.1.1 Caché

O Caché é um sistema gerenciado de banco de dados pós-relacional de alta desempenho. Características (INTERSYSTEMS CORPORATION, 2007):

- a) um ambiente de desenvolvimento orientado a objeto;
- b) acesso a objeto e SQL integrado (arquitetura de dados unificados);
- c) ferramentas de desenvolvimento web;
- d) conectividade a partir de diversas linguagens;
- e) uma base de dados multidimensional eficiente e escalável.

Extremamente econômico e eficaz como ambiente de produção, o Caché é um banco de dados que utiliza recursos do modelo de dados multidimensional para romper as limitações

de desempenho e modelagem do modelo relacional. Produto da InterSystems Corporation, os aplicativos desenvolvidos em Caché oferecem (IB TECH IT SOLUTIONS, 2009):

- a) alta disponibilidade e segurança de dados;
- b) eliminação de qualquer armazenamento desnecessário de informações;
- c) alto desempenho e escalabilidade na disponibilização de Interfaces Gráficas para Usuários (aplicações *Graphical User Interfaces – GUI*) ou *World Wide Web (WWW)*;
- d) modificação dos modelos de dados de maneira eficiente;
- e) superioridade de recursos em relação ao modelo relacional;
- f) um fácil e rápido modelo de aplicação utilizando objetos;
- g) uma nova linguagem de programação orientada a objetos: o COS;
- h) alta performance, através dos objetos;
- i) rápido desenvolvimento de aplicações;
- j) exportação nativa de objetos Caché como objetos Java;
- k) exportação nativa de objetos Caché como objetos ActiveX, principalmente para Visual Basic;
- l) dualização flexível de servidores;
- m) desempenho elevado para sistemas Unix, via servidores de processos.

### 3.3.1.2 Caché Server Pages (CSP)

O CSP é uma tecnologia utilizada para desenvolvimento de aplicações para web, na forma de páginas dinâmicas, integrado ao banco de dados Caché e baseado na linguagem COS. Pode-se dizer que ele é uma tecnologia similar ao PHP, Active Server Pages (ASP), Java Server Pages (JSP) etc. (INTERSYSTEMS CORPORATION, 2007).

### 3.3.1.3 IBTech Caché Framework (ICF)

O IBTech Caché Framework é um *framework* proprietário, desenvolvido pela empresa IBTech It Solutions, que tem por objetivo otimizar o processo de desenvolvimento de páginas web dinâmicas utilizando a tecnologia CSP. O IBTech Caché *Framework* consiste em um conjunto de classes Caché, TAGs customizadas na forma de *Rules* CSP e bibliotecas Java Script (IB TECH IT SOLUTIONS, 2009).

Dentre as vantagens do IBTech Caché *Framework* podemos citar:

- a) diminuição significativa da mistura de código HTML (*Hyper Text Markup Language*) e Java Script com o código CSP;

- b) padronização do desenvolvimento através de TAGs customizadas;
- c) diminuição da curva de aprendizagem da tecnologia CSP;
- d) eliminação da necessidade de desenvolvimento de métodos (lógicas de negocio) dentro das paginas CSP. Tais métodos ficam concentrados em classes Caché, acessíveis via componentes Ajax do *framework*;
- e) separação da camada de interface da camada de negócios da aplicação.

Para ilustrar o desenvolvimento, no Quadro 4 é apresentado um trecho do código fonte da página CSP responsável pelo cadastro de projetos, sendo que podemos observar a utilização das TAGs customizadas do *framework* ICF e no Quadro 5 podemos visualizar o acesso a métodos concentrados em classes Caché via componentes de Ajax do *framework* ICF.

Quadro 4 – Código fonte da página CSP utilizando o framework ICF

```

286
287 <body>
288
289 <form id='myForm' autocomplete='off' method='post' action='' onsubmit="return false;">
290
291 <icf:body caption="Geração de Projeto" cols="120" rows="23" version="Tcc.Utils"
292   focus=#{($s(%request.Get("isCalled"))=""?"false",1:"true")}# showresize="true">
293
294 <icf:datasource class="Tcc.Adm.Projeto" keys="id" name="dsProjeto" type="class">
295 <icf:datasource class="Tcc.Adm.ProjetoTarefa" keys="id" mastersource="dsProjeto" name="dsProjetoTarefa" type="class">
296
297 <icf:hidden id="cpIsCalled" value="#"(%request.Get("isCalled"))#>
298 <icf:hidden id="cpTypeCalled" value="#"(%request.Get("typeCalled"))#>
299 <icf:hidden id="cpIdProjetoCalled" value="#"(%request.Get("idProjeto"))#>
300 <icf:hidden id="cpIdTarefaCalled" value="#"(%request.Get("idTarefa"))#>
301 <icf:hidden id="cpIdProjeto" value="" field="dsProjetoTarefa.Projeto">
302
303 <!-- Capa Projeto-->
304 <icf:panel id="pnTopo" rows=2 border=0>
305 <icf:label colspan=2 newline="true">
306
307 <icf:label caption="Projeto" bold="true">
308 <icf:editdisplay id="cpProjeto" canfilter="false" maxlength="6" method="ajaxGetDisplay" par1="cpProjeto" par2="fix:1"
309 showdisplay="false" showf8="false" showf7="true" size="5" type="codeinteger">
310
311 <icf:updatebar datasource="dsProjeto" id="uptBarProjeto" colspan="2" newline="false" shownew="true">
312 </icf:panel>
313

```



Quadro 5 – Código fonte de acessos a métodos Caché via componentes Ajax do framework ICF

```

184 function salvarRecursosTarefa(recursosSelecao) {
185     var method = new IcfFormMethod("Tcc.Adm.ProjetoTarefa","ajaxSalvarRecursosTarefa",
186                                     'cpProjeto','cpTarefa','fix:'+recursosSelecao);
187     var sc = method.execute();
188     if (sc==null) return;
189 }
190
191 function salvarCompetenciasTarefa(competenciaSelecao) {
192     var method = new IcfFormMethod("Tcc.Adm.ProjetoTarefa","ajaxSalvarCompetenciasTarefa",
193                                     'cpProjeto','cpTarefa','fix:'+competenciaSelecao);
194     var sc = method.execute();
195     if (sc==null) return;
196 }

```

#### 3.3.1.4 Biblioteca jQuery

A biblioteca jQuery utiliza um seletor simples como forma de obter e manipular elementos denominados Document Object Model (DOM), sendo que pode-se utilizar as próprias TAGs do HTML como seletores ou através de atributos como ID e NAME. Quando um elemento DOM é obtido, ele se torna um objeto jQuery e qualquer método da biblioteca padrão ou de extensões do jQuery podem ser invocados desses objetos. Além destas funcionalidades o jQuery possibilita o uso de seletores complexos, como: elementos baseados em relacionamento pai-filho, atributos, classes e filtros (MANOR,2010).

Segundo Manor (2010), a biblioteca jQuery busca ser simples e intuitiva incluindo os seguintes itens:

- a) navegação da árvore DOM;
- b) manipulação de elementos DOM;
- c) manipulação de CSS;
- d) gerenciamento de Eventos (onclick, onsubmit);
- e) interação com Ajax;
- f) funções utilitárias;
- g) biblioteca UI;
- h) biblioteca de Efeitos.

A principal finalidade do uso de jQuery é controlar o comportamento de uma página web. O principio de funcionamento baseia-se na sua capacidade de encontrar elementos HTML e anexar a estes seus métodos, tornando a aplicação muito mais dinâmica (SILVA, 2010).

A utilização da biblioteca jQuery foi de extrema importância no desenvolvimento deste trabalho, com o auxílio de plug-ins foi possível implementar o efeito de *drag and drop* (arrastar e soltar) na principal funcionalidade do sistema, sendo ela o quadro *Kanban*. No Quadro 6 pode-se observar um trecho do código onde é construído o método de *drag and drop* dos cartões nas colunas do quadro *Kanban*.

Quadro 6 – Código fonte do método *drag and drop*

```

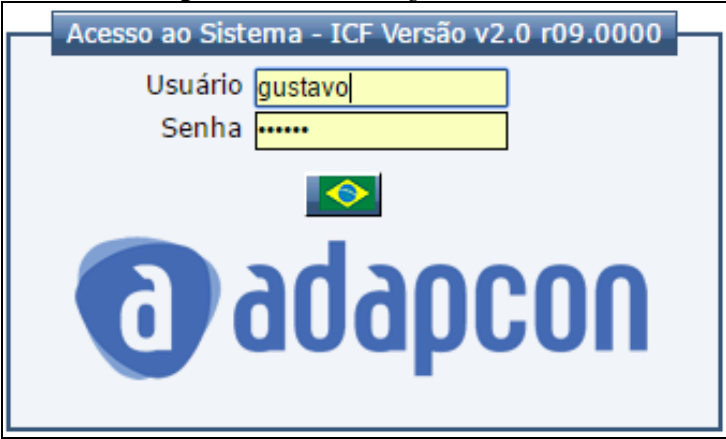
63  function onLoadJS() {
64
65      // Habilita todos os cartões com draggable igual a verdadeiro
66      var cards = document.querySelectorAll('.card');
67      for (var i = 0, n = cards.length; i < n; i++) {
68          var card = cards[i];
69          card.draggable = true;
70      };
71      var board = $('#board');
72      var hideMe;
73      var target_transferred;
74      jQuery.event.props.push('dataTransfer');
75
76      // Trata seleção inicial do cartão no quadro
77      $("#board").on('selectstart',function(e) {
78          e.preventDefault();
79      });
80
81      // Trata inicio da movimentação do cartão no quadro
82      $("#board").on('dragstart', function(e){
83          e.dataTransfer.setData("card",e.target.id);
84          e.dataTransfer.effectAllowed = "copy";
85          hideMe = e.target;
86          target_transferred = e.target;
87      });
88
89      // Trata final da movimentação do cartão no quadro
90      $("#board").on('dragend', function(e){
91          $(e.target).show();
92      });

```

### 3.3.2 Operacionalidade da implementação

A seguir serão apresentadas as telas do sistema, com explicações referentes às suas funcionalidades.


Para acessar o sistema é necessário informar um *login* e senha, sendo que conforme o perfil do usuário o sistema disponibiliza o acesso às funcionalidades. Na Figura 10 pode-se visualizar a tela de *login* do sistema.


Figura 11– Tela de *login* do sistema

Acesso ao Sistema - ICF Versão v2.0 r09.0000

Usuário

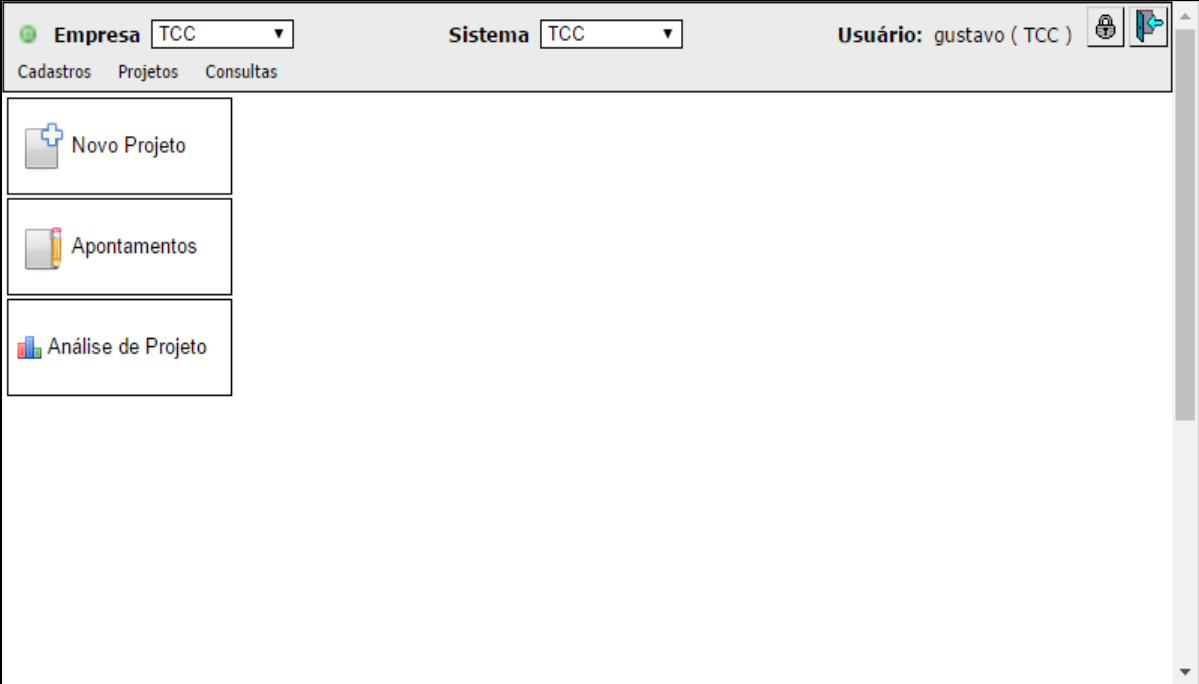
Senha






Após efetuar o *login* o usuário acessa a tela principal do sistema contendo o menu personalizado de acordo com o perfil do usuário (Administrador, Colaborador e Cliente). Na Figura 11 é mostrada a tela inicial do sistema.


Figura 12 – Tela inicial do sistema




Empresa  Sistema  Usuário: gustavo ( TCC )

Cadastros Projetos Consultas

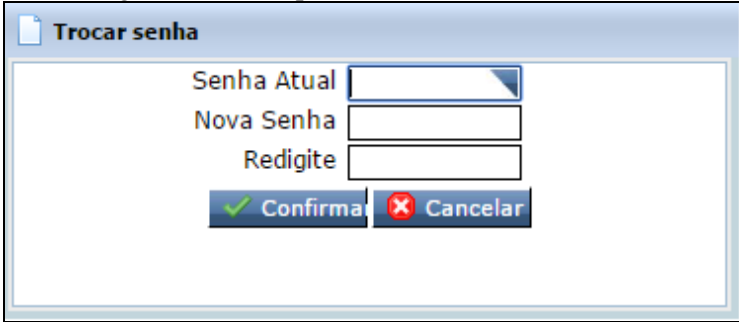
 Novo Projeto

 Apontamentos

 Análise de Projeto

Na Figura 12 pode-se visualizar a tela para alterar a senha do usuário.

Figura 13 – Tela para alterar a senha do usuário



Trocar senha

Senha Atual

Nova Senha

Redigite

Na Figura 13 pode-se visualizar a tela de cadastro de usuário no sistema, sendo que é possível adicionar, editar ou excluir um usuário.

Figura 14 – Tela de cadastro de usuário

The screenshot shows a window titled 'Usuários' with the following fields and options:

- Login: gustavo
- Nome: gustavo
- Senha: [masked]
- E-mail: gustavo@adapcon.com.br
- Depurar: Sim
- Tipo: Administrador
- Situação: Ativo
- Formato de Exportação para Excel: XML
- Revalidar senha ao fazer login: Não
- Válido até: 31/12/2100
- Arquivos de Estilos: Genérico (aqua), Tabelas (aqua)
- Acessa todas as empresas: Sim
- Acessa todos os sistemas: Sim
- Conta padrão: TCC (TRABALHO CONCLUSIVO DE CURSO)
- Sistema padrão: TCC
- Empresa padrão: TCC
- Sistemas list: TCC (checked), Utilitários (unchecked) - 2 registros [Páginas]
- Empresas list: TCC (checked) - 1 registro [Páginas]
- Buttons: Salvar, Excluir, Cancelar, Sair

Para criar um projeto, primeiramente é necessário alimentar algumas informações presentes na opção de Cadastros, sendo eles: clientes, colaboradores, recursos, tipos de serviços e competências. Na Figura 14 pode-se visualizar a tela de cadastro de competências, na qual é possível adicionar, editar e excluir uma competência.

Figura 15 – Cadastro de Competências

The screenshot shows a window titled 'Cadastro de Competências' with the following fields and options:

- Sigla: [input field]
- Nome: [input field]
- Tipo: -- Selecione --
- Situação: -- Selecione --
- Buttons: Salvar, Excluir, Cancelar, Sair
- Table of existing competencies:

Sigla	Nome	Tipo	Situação
COS	Cache COS	Programação	Ativo
OO	Cache Orientado a Objetos	Programação	Ativo
IDE	Caché IDE	Programação	Ativo
WS	Web Service	Programação	Ativo
SVN	Versionamento (SVN)	Utilitarios	Ativo
LNX	Linux Basico	Utilitarios	Ativo
ORC	Análise e Desenvolvimento de Orçamentos	Negócio	Ativo
MOD	Modelagem de Sistemas	Negócio	Ativo
AGIL	Cultura Agil	Negócio	Ativo
CUS	Modulo de Custos	Negócio	Ativo
NEG	Negociação	Comportamental	Ativo
GCF	Gestão de Conflitos	Comportamental	Ativo
RES	Resolução de Problemas	Comportamental	Ativo

Na Figura 15 pode-se visualizar a tela de cadastro de tipos de serviço, na qual é possível adicionar, editar e excluir um tipo de serviço.

Figura 16 – Cadastro de tipos de serviço

Sigla	Nome	Situação
DEV	Desenvolvimento	Ativo
TEST	Testes	Ativo
IMP	Implantação	Ativo
ANA	Analise	Ativo
RET	Retrabalho	Ativo

5 registros [Páginas]

Na Figura 16 pode-se visualizar a tela de cadastro de recurso, na qual é possível adicionar, editar e excluir um recurso.

Figura 17 – Cadastro de recurso

Sigla	Nome	Situação
ANA	Analista	Ativo
DES	Desenvolvedor	Ativo
TES	Testador	Ativo

3 registros [Páginas]

Nas Figuras 17 e 18 podem-se visualizar a tela de cadastro de colaborador, na qual é possível adicionar, editar e excluir um colaborador. A tela de cadastro de colaborador é dividida em duas abas, sendo elas:

- a) **dados gerais:** contemplam todas as informações pessoais do colaborador como nome, CPF, RG, data nascimento, data admissão, telefone celular, telefone residencial, e-mail, ramal interno, usuário do sistema, situação, data saída e função;
- b) **competências:** contemplam todas as competências do colaborador de acordo com os níveis iniciante, praticante e experiente. Sendo que, o nível iniciante significa que o colaborador não conhece o assunto, o nível praticante significa que o colaborador conhece o assunto e o nível experiente significa que o colaborador pode ensinar o assunto.

Figura 18 – Cadastro de colaborador (dados gerais)

A imagem mostra a interface de usuário de um sistema de cadastro de colaboradores. O título da janela é "Cadastro de Colaboradores". No topo, há um campo "Sigla" com o valor "GB" e uma lupa. Abaixo dele, há quatro botões: "Salvar", "Excluir", "Cancelar" e "Sair".

Existem duas abas: "Dados Gerais" (ativa) e "Competências".

Os campos de "Dados Gerais" são os seguintes:

- Nome Completo: Gustavo Baehr
- CPF: 08796011912
- RG: 5240053
- Data Nascimento: 29/04/1992
- Data Admissão: 10/01/2010
- Tel. Celular: 99912410
- Tel. Residencial: 33708069
- Email: gustavo@adapcon.com.br
- Ramal Interno: 30
- Usuário Sistema: gustavo
- Situação: Ativo
- Data Saída: (campo vazio)
- Função: Analista

Figura 19 – Cadastro de colaborador (competências)

Tipo	Competência	Iniciante	Praticante	Experiente
Programação	Cache COS	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Programação	Cache Orientado a Objetos	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Programação	Caché IDE	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Programação	Web Service	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Utilitários	Versionamento (SVN)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Utilitários	Linux Básico	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Negócio	Análise e Desenvolvimento de Orçamentos	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Negócio	Modelagem de Sistemas	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Negócio	Cultura Ágil	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Negócio	Módulo de Custos	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Comportamental	Negociação	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Comportamental	Gestão de Conflitos	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Comportamental	Resolução de Problemas	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Nas Figuras 19, 20, 21 e 22 podem-se visualizar a tela de cadastro de cliente, na qual é possível adicionar, editar, excluir e consultar um cliente. A tela de cadastro de cliente é dividida em quatro abas, sendo elas:

- dados gerais: contemplam todas as informações essenciais do cliente como nome, telefone, conta, endereço, bairro, cidade, estado, CEP, CNPJ, inscrição estadual e ramo de atividade;
- observações: contempla o armazenamento de informações adicionais;
- contatos: contempla todas as informações referentes aos contatos desse cliente como nome, setor, telefone, e-mail, data de nascimento e usuário sistema.
- consulta: contempla em uma ferramenta de consulta de clientes cadastrados.

Figura 20 - Cadastro de clientes (dados gerais)

**Cadastro de Clientes**

Código: 2

Novo Salvar Excluir Cancelar Sair

Dados Gerais Observações Contatos Consulta

Nome: Confecção ABC

Telefone:

Conta: ABC

Endereço:

Bairro:

Cidade: Jaragua do Sul

Estado: SC

CEP: 99999999

CNPJ: 75.364.570/0007-55

IE: 1239199299391

Ramo de Atividade: Confecção

Figura 21 – Cadastro de clientes (observações)

**Cadastro de Clientes**

Código: 2

Novo Salvar Excluir Cancelar Sair

Dados Gerais Observações Contatos Consulta

IP Servidor de testes: 192.168.1.1  
login: teste  
senha: teste123



Figura 22 – Cadastro de clientes (contatos)

**Cadastro de Clientes**

Código: 2

Novo Salvar Excluir Cancelar Sair

Dados Gerais Observações **Contatos** Consulta

Nome: João da Silva  
 Setor: TI  
 Telefone: 9999-1111  
 Email:  
 Data Nasc.:  
 Usuário Sistema:

Novo Salvar Excluir Cancelar

Nome	Setor	Telefone	Email	Data Nasc.
João da Silva	TI	9999-1111		

1 registro [Páginas]

Figura 23 – Cadastro de clientes (consulta)

**Cadastro de Clientes**

Código: 1

Novo Salvar Excluir Cancelar Sair

Dados Gerais Observações Contatos **Consulta**

Nome:  Consultar

Código	Nome
2	Confecção ABC
1	Metalurgica C.O.

2 registros [Páginas]

Após realizar todos os cadastros gerais, pode-se criar um projeto. Nas Figuras 23, 24, 25 e 26 pode-se visualizar a tela de geração de projeto, na qual é possível adicionar, editar, excluir e consultar um projeto. A tela de geração de projeto é dividida em quatro abas, sendo elas:

- a) dados gerais: contemplam as informações da capa do projeto, sendo elas a data de abertura, o cliente, o solicitante, o prazo de entrega e o objetivo. Também pode ser configurado o WIP das colunas do quadro *Kanban*;
- b) tarefas: pode-se incluir, editar e excluir uma tarefa do projeto, sendo que baseia-se em um trabalho específico de pouca duração podendo definir um resumo, uma complexidade, um prazo de entrega específico, uma quantidade de horas previstas para desenvolver a tarefa, uma observação, atrelar recursos e definir as competências necessárias para o desenvolvimento da mesma;
- c) lançamentos por tarefa: pode-se consultar os apontamentos de horas realizadas para cada tarefa do projeto;
- d) histórico projeto: pode-se consultar o histórico de ações realizadas durante o processo de desenvolvimento do projeto.

Figura 24 – Geração de projeto (dados gerais)

A imagem mostra a interface de usuário do sistema 'Geração de Projeto'. No topo, há uma barra de ferramentas com os botões 'Novo', 'Salvar', 'Excluir', 'Cancelar' e 'Sair'. Abaixo, há uma barra de navegação com as abas 'Dados Gerais', 'Tarefas', 'Lançamentos por Tarefa' e 'Histórico Projeto'. A aba 'Dados Gerais' está selecionada e contém os seguintes campos:

- Data de Abertura: 04/06/2016
- Cliente: Metalurgica C.O. (CO) (1)
- Solicitante: Paulo
- Prazo Entrega: 08/08/2016
- Objetivo: Desenvolver modulo de Vendas Varejo

Na parte inferior da aba, há uma seção intitulada 'Configuração do WIP das colunas do quadro Kanban' com os seguintes valores:

Coluna Desenvolvendo	2
Coluna Para Testes	4
Coluna Cliente Validando	4

Um contador no canto inferior direito da área de texto indica '35 / 999'.

Figura 25 – Geração de projeto (tarefas)

**Geração de Projeto**

Projeto 3

Novo Salvar Excluir Cancelar Sair

Dados Gerais Tarefas Lançamentos por Tarefa Histórico Projeto

Tarefa 2

Resumo Configurações por Operador

Complexidade Baixa

Prazo Entrega Especifico 20/06/2016

Horas Previstas 08:00

Observação

Configurações do Operador

- % de desconto maximo
- Permissão Cancelamento nota
- Permissão Cancelar Venda
- Permissão de excluir itens lançados.
- Permissão para alterar venda
- Conta Padrão

240 / 9999

Recursos

Recurso	
Analista	<input type="checkbox"/>
Desenvolvedor	<input checked="" type="checkbox"/>
Testador	<input type="checkbox"/>

3 registros [Páginas]

Competências Necessárias

Competência	
Cache COS	<input checked="" type="checkbox"/>
Cache Orientado a Objetos	<input checked="" type="checkbox"/>
Cache IDE	<input type="checkbox"/>
Web Service	<input type="checkbox"/>

Novo Salvar Excluir Cancelar Sair

Tarefa	Resumo	Complexidade	Total de Horas
1	Configurações gerais	Baixa	8,00
2	Configurações por Operador	Baixa	8,00
3	Interface de Venda	Alta	80,00
4	Nova tarefa	Baixa	4,00
5	Tarefa 3	Baixa	3,00
			<b>103,00</b>

6 registros [Páginas]

Figura 26 – Geração de projeto (lançamentos por tarefa)

**Geração de Projeto**

Projeto 3

Novo Salvar Excluir Cancelar Sair

Dados Gerais Tarefas Lançamentos por Tarefa Histórico Projeto

Tarefa	Resumo	Data	Colaborador	Realizado	Tipo Serviço	Anotação
1	Configurações gerais	04/06/2016	Gustavo Baehr	02:00	Desenvolvimento	desenvolvimento de tela de configurações gerais

2 registros [Páginas]

Figura 27 – Geração de projeto (histórico projeto)

Data	Hora	Descrição Histórico
04/06/2016	17:27	O usuário Gustavo Baehr criou o projeto 3
04/06/2016	17:27	O usuário Gustavo Baehr adicionou/alterou a tarefa 1
04/06/2016	17:27	O usuário Gustavo Baehr adicionou/alterou a tarefa 2
04/06/2016	17:27	O usuário Gustavo Baehr adicionou/alterou a tarefa 3

Na Figura 27 é apresentada a tela de consulta de análise de projetos, sendo que todos os projetos que atendem aos filtros iniciais da tela são carregados automaticamente. A consulta apresenta as seguintes informações por coluna: projeto, data de abertura, prazo de entrega, prazo em dias faltantes, situação, cliente, resumo, horas previstas, horas realizadas, saldo de horas (previsto menos realizado) e o percentual realizado do projeto. Também apresenta as informações dividindo-as em três cores:

- verde: corresponde a todos os projetos que estão dentro do prazo estipulado e com o saldo de horas positivo;
- amarelo: corresponde a todos os projetos que não estão dentro do prazo estipulado, porém estão com o saldo de horas positivo;
- vermelho: corresponde a todos os projetos que estão com o saldo de horas negativo.

A consulta de análise de projetos também é utilizada pelo cliente, sendo que o mesmo consegue visualizar apenas seus projetos.

Figura 28 – Consulta análise de projetos

Projeto	Data Abertura	Prazo Entrega	Prazo (Dias)	Situação	Cliente	Resumo	Resultados (Físico)			
							Proporcionado	Realizado	Saldo	%
2	04/06/2016	30/05/2016	-39	Em Aberto	Confecção ABC	Incluir coluna com a Data de entrada da nota na consulta de pedidos de compra em aberto	3,00	0,00	3,00	0,00
1	04/06/2016	10/06/2016	-28	Em Aberto	Metalurgica C.O.	Desenvolver ferramenta para auditar custos da empresa teste	26,00	0,00	26,00	0,00
4	04/06/2016	28/06/2016	-10	Em Aberto	Metalurgica C.O.	Projeto referente a melhorias no cadastro de pedido de compra.	1,00	3,00	-2,00	300,00
3	04/06/2016	08/08/2016	31	Em Aberto	Metalurgica C.O.	Desenvolver modulo de Vendas Varejo	103,00	5,00	98,00	4,85
							133,00	8,00	125,00	6,02

Na Figura 28 é apresentada a tela do quadro *Kanban*, principal funcionalidade do sistema. Para acessar esta tela, o usuário deve entrar na tela de análise de projetos e selecionar na coluna projeto o código do projeto desejado, sendo que o sistema irá direcionar para o quadro *Kanban* representando todas as tarefas do projeto como cartões no quadro. Nesta tela o usuário tem acesso às informações do cartão e acesso a tela de apontamento de horas realizadas, podendo também movimentar os cartões sobre as colunas. Todos os cartões iniciam na coluna “Entrada” e o sistema ordena-os de acordo com seu prazo de entrega e sugere o melhor recurso considerando as competências do colaborador frente às necessidades da tarefa. A escolha do melhor recurso para cada tarefa é realizada através de um algoritmo desenvolvido que compara as competências necessárias da tarefa, considerando os recursos selecionados, com as competências dos colaboradores, sendo que cada nível de competência representa uma pontuação interna para o sistema, sendo que para a categoria iniciante é atribuído um ponto, para a categoria praticante são atribuídos três pontos e para a categoria experiente são atribuídos cinco pontos. A escolha do melhor recurso para atender a necessidade da tarefa, é baseada no somatório de suas competências frente às necessidades da tarefa, sendo que o recurso com maior pontuação é sugerido. Caso haja um empate de pontuação, o sistema sempre sugere o primeiro recurso considerando a ordem alfabética da sigla do colaborador.

Além da sugestão do recurso, cada cartão apresenta as seguintes informações: resumo da tarefa, prazo de entrega, horas realizadas (em azul), horas previstas (em verde) e responsável, sendo que as informações de prazo de entrega e horas previstas são apresentadas da seguinte forma:

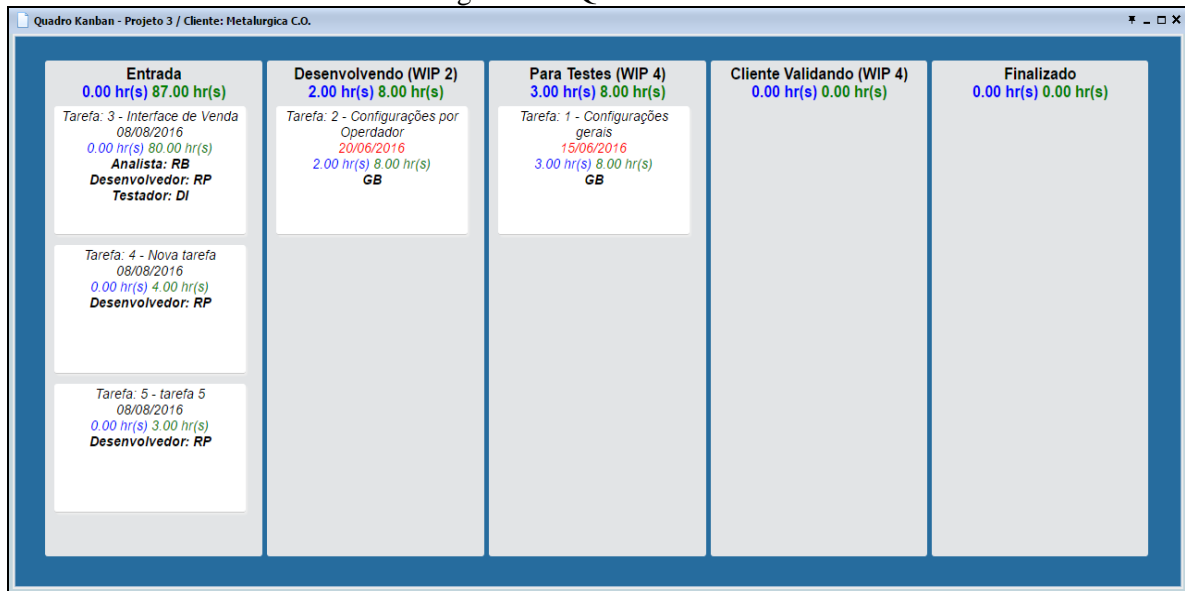
- a) prazo de entrega: esta informação pode ser apresentada em preto, laranja e vermelho. Sendo que o preto representa que o prazo para entrega da tarefa está dentro do previsto comparado com a data atual, o laranja representa que o prazo

para entrega da tarefa é a data atual e o vermelho representa que o prazo para entrega da tarefa já passou, ou seja, esta tarefa está atrasada;

- b) horas previstas: esta informação pode ser apresentada em verde e vermelho. Sendo que o verde representa o saldo positivo frente às horas realizadas e o vermelho representa o saldo negativo frente às horas realizadas.

Quando um colaborador movimenta o cartão entre as colunas, o sistema automaticamente torna-o responsável pela tarefa, sinalizando com a sigla do colaborador sobre o cartão. O quadro apresenta um totalizador de horas realizadas e previstas por coluna auxiliando na identificação de gargalos no projeto.

Figura 29 – Quadro *Kanban*



Na Figura 29 pode-se visualizar o detalhamento da tarefa a partir do quadro *Kanban*, sendo que apenas o campo observação está habilitado para edição. Para abrir a tela de detalhamento da tarefa, o usuário precisa clicar com o botão esquerdo do mouse sobre o cartão.

Figura 30 – Detalhamento da tarefa a partir do quadro *Kanban*

The screenshot shows the 'Geração de Projeto' application window. The 'Tarefas' tab is active, displaying details for task '3' titled 'Interface de Venda'. The task has a complexity of 'Alta' and a specific delivery date of '08/08/2016'. The estimated hours are '80:00'. The observation field contains a list of tasks: 'Rotina 1 - Lançamento de Itens', 'Numero da Caixa (Venda) - Automatico', 'Conta - Carregado o padrão do operadores', 'Vendedor - Seleção sobre os registros habilitados (Se houver somente um habilitado já traz carrega)', 'Cliente - Consulta f7 + cadadsro f8 - possibilitar o preenchimento do CPF \ CPJ no CSLE para carregar o cliente.', and 'Saldo Limite (Calcula o saldo limite do cliente a partir do limite de crédito - os titulos em aberto que não)'. Below the observation field, there are two tables: 'Recursos' and 'Competências Necessárias'. The 'Recursos' table lists 'Analista', 'Desenvolvedor', and 'Testador' with checkmarks. The 'Competências Necessárias' table lists 'Cache COS', 'Cache Orientado a Objetos', 'Cache IDE', 'Web Service', and 'Versionamento (SVN)' with checkmarks. At the bottom, a summary table shows the task details and total hours.

Tarefa	Resumo	Complexidade	Total de Horas
1	Configurações gerais	Baixa	8,00
2	Configurações por Operador	Baixa	8,00
3	Interface de Venda	Alta	80,00
			<b>96,00</b>

Na Figura 30 pode-se visualizar a tela de apontamento de horas realizadas a partir do quadro *Kanban*. Para abrir a tela de apontamento de horas realizadas o usuário precisa clicar com o botão do centro do mouse sobre o cartão.

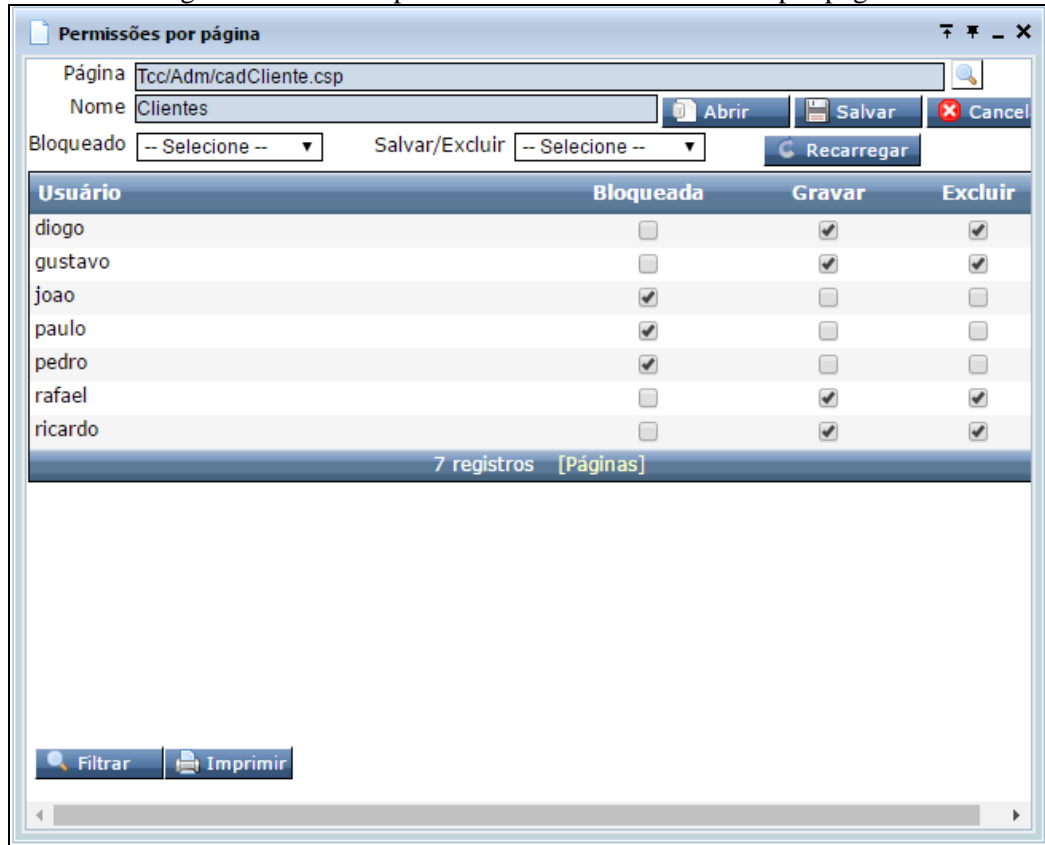
Figura 31 – Tela de apontamento de horas realizadas

The screenshot shows the 'Apontamento de Horas' application window. The 'Projeto' is '3' and the 'Tarefa' is '1'. The date is '05/06/2016' and the hours worked are '03:00'. The service type is 'Testes'. The annotation is 'Testes na aplicação desenvolvida'. Below the annotation field, there is a table showing the log of hours worked.

Data	Hora Realizada	Tipo Serviço	Anotação	Colaborador
04/06/2016	02:00	Desenvolvimento	desenvolvimento de tela de configurações gerais	Gustavo Baehr

Na Figura 31 pode-se visualizar a tela de configuração de permissões de acesso de usuários por página do sistema.

Figura 32 – Tela de permissões de acesso de usuário por página



### 3.4 RESULTADOS E DISCUSSÕES

O estudo e documentação realizados sobre método ágil, especialmente para a técnica de gerenciamento do quadro *Kanban* para visualizar o processo, foram fundamentais para a criação do sistema proposto. O sistema desenvolvido proporciona uma visão geral do trabalho em progresso do projeto, permitindo atribuir recursos conforme as competências do colaborador frente às necessidades da tarefa, tornando o processo de desenvolvimento mais ágil e menos burocrático.

Após o estudo dos trabalhos correlatos, comparados com o sistema desenvolvido, foi identificado que o sistema proposto contempla os principais objetivos do *Kanban* que são: identificar gargalos do processo, acompanhar o fluxo do trabalho e definir o WIP das colunas. O Quadro 7 apresenta uma comparação do sistema desenvolvido com os trabalhos correlatos, baseado na fundamentação teórica e nas principais características e funcionalidades apresentadas.



Quadro 7 – Comparação entre os trabalhos correlatos e o sistema desenvolvido

<b>Funcionalidades / Características</b>	<b>Sistema desenvolvido</b>	<b>Pronto</b>	<b>Quick Screen</b>	<b>Trello</b>
Plataforma web	Sim	Sim	Sim	Sim
Relatórios estatísticos	Sim	Sim	Sim	Não
Flexibilidade na escolha do processo	Não	Não	Sim	Sim
Histórico de movimentação de cartões	Sim	Sim	Sim	Sim
Gráfico <i>burndown</i>	Não	Sim	Não	Não
Língua portuguesa	Sim	Sim	Sim	Sim
Sugestão de recursos por tarefa (Competência)	Sim	Não	Não	Não
Limitar o WIP por projeto	Sim	Não	Não	Não

## 4 CONCLUSÕES

O mercado de metodologia ágil em gerenciamento de projetos de software, vem crescendo nos últimos anos, apresentando diversas soluções para problemas distintos, visando aumentar a produtividade e a qualidade dos projetos. O sistema web desenvolvido auxilia o gerenciamento de desenvolvimento ágil de software, proporcionando um ambiente onde todos os envolvidos possam visualizar o acompanhamento das etapas do processo e evidenciar os problemas antes de finalizar o processo. Para isso, baseia-se na técnica *Kanban*, por ser uma ferramenta sistêmica de melhoria contínua que permite além de visualizar o processo com um todo, limitar a quantidade de trabalho em progresso (WIP), identificar gargalos e impedimentos, fomentando discussões frequentes de estratégias de coordenação entre os envolvidos no processo.

A principal funcionalidade do sistema é o quadro *Kanban* moldado conforme as etapas existentes no processo de execução de projetos da empresa, sendo que pode-se definir um WIP por projeto restringindo o número de cartões que cada etapa pode receber simultaneamente. Assim, o quadro *Kanban* dá ao gestor do projeto um instrumento capaz de avaliar quais etapas do processo estão sobrecarregadas ou ociosas, auxiliando-o a realizar mudanças e ajustes antes de finalizar o processo, buscando manter o fluxo contínuo de cartões sobre o quadro.

Além de proporcionar uma visão do processo de desenvolvimento do projeto, a ferramenta desenvolvida, permite distribuir as tarefas de modo mais eficiente, sendo que sugere o melhor recurso considerando as competências do colaborador frente às necessidades da tarefa, aproveitando de maneira eficaz a mão de obra disponível.

Diante das funcionalidades implementadas, o sistema desenvolvido conseguiu atender seu objetivo principal e seus objetivos específicos sendo que a empresa Adapcon Processamento de Dados Ltda. poderá utilizar uma ferramenta adequada que comporta os procedimentos, técnicas e princípios de métodos ágeis para gerenciar seus projetos.

### 4.1 EXTENSÕES

Como sugestão de implementação de funcionalidade para trabalhos futuros pode-se destacar:

- a) desenvolver um módulo para acesso através de dispositivos portáteis, para permitir mobilidade no uso do sistema;
- b) desenvolver um cadastro de quadros com colunas dinâmicas, para permitir modelar o quadro de acordo com o processo do projeto;

- c) desenvolver o gráfico de *Burndown Chart*, para permitir a visualização da quantidade de trabalho realizado versus trabalho planejado;
- d) desenvolver o gráfico de *Gantt*, para permitir um acompanhamento visual do progresso do projeto como um todo.

## REFERÊNCIAS

- ABREU, Cristina V. FREITAS, Maria N. C. Seleção por competências: a percepção dos profissionais de RH sobre o método de seleção por competências. **Revista Pesquisas e Práticas Psicossociais** v. 3, p. 2, São João del-Rei, Mar. 2009.
- AGILE MANIFESTO. *Manifesto for agile software development*. [S.l.], 2016. Disponível em: <<http://agilemanifesto.org/iso/ptbr/>>. Acesso em: 10 mar. 2016.
- ALDABÓ, Ricardo. **Gerenciamento de projetos: procedimento básico e etapas essenciais**. 2. ed. São Paulo: Artliber, 2001.
- ANDERSON, David J. *Kanban: successful evolutionary change for your technology business*. Seattle: Blue Hole, 2010.
- BASSI FILHO, Dairton L. **Experiências com desenvolvimento ágil**. 2008. 154f. Dissertação (Mestrado em Ciências da Computação) - Instituto de Matemática e Estatística da Universidade de São Paulo, São Paulo.
- BARATO, Jarbas N. **Competências essenciais e avaliação do ensino universitário**. Brasília: UNB, 1998.
- CHIAVENATO, Idalberto. **Administração de recursos humanos**. São Paulo: Atlas, 1997.
- DEBATIN, Luiz. F. **Aplicação web para acompanhamento de projetos ágeis utilizando o método Kanban**. 2011. 65f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau, Blumenau.
- DUTRA, Joel S.; FLEURY, Maria T. L.; RUAS, Roberto. **Competências: conceitos, métodos e Experiência**. São Paulo: Atlas S.A. 2012.
- FURTADO, Vasco. **Tecnologia e gestão da informação na segurança pública**. Rio de Janeiro: Garamond, 2002.
- GOMES, André F.; FAIAS JUNIOR, Luiz dos S. Pronto! - **Software para gestão de projetos ágeis**. 2009. 66f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Faculdade de Informática e Administração Paulista, São Paulo.
- IBTECH IT SOLUTION. **IBtech Caché Framework**. Santa Catarina: 2009. (Não publicado).
- INTERSYSTEMS CORPORATION. **Caché Object Script**. São Paulo: 2007. (Não publicado).
- MARTINS, Jose C. C. **Técnicas para gerenciamento de projetos de software**. Rio de Janeiro: Brasport, 2007.
- MANOR, Gilad. **JQuery, um Framework Java Script**. [S.l.], 2016. Disponível em: [http://www.infoq.com/br/news/2010/05/jquery\\_framework](http://www.infoq.com/br/news/2010/05/jquery_framework). Acesso em: 10 de maio 2016.
- PAULA FILHO, Wilson de P. **Engenharia de Software: Fundamentos, Métodos e Padrões**. Rio de Janeiro: LTC, 2003.
- PMBOK. **Um Guia do Conhecimento em Gerenciamento de Projetos**, 5. ed. Pennsylvania: PMI, 2013.
- PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos ágeis para desenvolvimento de software**. Porto Alegre: Bookman, 2014.
- RABAGLIO, Maria O. **Seleção por competências**. 4. ed. São Paulo: Educator, 2004.

- REIS, Christian. **Caracterização de um processo de software para projetos de software livre**. 2003. 247 f. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos.
- REZENDE, Denis A. **Engenharia de software e sistemas de informação**. 3. ed. Rio de Janeiro: Brasport, 2005.
- SANTOS, Otávio. A. R. dos. Gerenciamento ágil de projetos: uma abordagem adaptativa. **Mundo Project Management**, v. 22, p. 38-42, ago./set. 2008.
- SILVA, Maurício S. **jQuery - A Biblioteca do Programador Java Script**. 2. ed. São Paulo: Novatec, 2010.
- SOMMERVILLE, Ian. **Engenharia de software**. 8.ed. São Paulo: Pearson Addison Wesley, 2007.
- TRELLO. **Software para gestão de projetos**. [S.l.], 2016. Disponível em: <<https://trello.com/home/>>. Acesso em: 08 mai. 2016.
- TUBINO, Dalvio. F. **Manual de planejamento e controle da produção**. São Paulo: Atlas, 1997.
- VARGAS, Ricardo. **Gerenciamento de projetos**. 6. ed. Rio de Janeiro: Brasport, 2005
- VIEIRA, Marconi F. **Gerenciamento de projetos de tecnologia da informação**. Rio de Janeiro: Campus, 2003.
- ZIMERMANN, Francielle T. **Aplicação da técnica *analytic hierarchy process* (AHP) na priorização e seleção de projetos**. 2012. 87f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau, Blumenau.

## APÊNDICE A – Descrição dos principais Casos de Uso

Os quadros 8, 9, 10, 11 e 12 apresentam a descrição dos principais casos de uso do sistema.

Quadro 8 – Descrição do caso de uso UC09

<b>Caso de uso</b>	UC09 – Manter Projetos
<b>Descrição</b>	O sistema deve permitir ao ator administrador cadastrar, editar, excluir e consultar projetos.
<b>Ator</b>	Administrador.
<b>Pré-condição</b>	Administrador deve fazer <i>login</i> no sistema.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O sistema apresenta somente o campo “Projeto” habilitado.</li> <li>2. O ator seleciona o botão “Novo”.</li> <li>3. O sistema habilita edição dos dados da capa do projeto.</li> <li>4. O ator preenche dados da capa do projeto e seleciona o botão “Salvar”.</li> <li>5. O sistema grava o projeto.</li> </ol>
<b>Fluxo Alternativo 1</b>	<p>No passo 2, do fluxo principal, caso o ator deseje alterar o projeto, deve selecionar um projeto já existente.</p> <ol style="list-style-type: none"> <li>2.1. O ator consulta os projetos cadastrados e seleciona um projeto.</li> <li>2.2. O sistema carrega os dados da capa do projeto.</li> <li>2.3. O sistema habilita edição dos dados da capa do projeto.</li> <li>2.4. O ator edita os dados da capa do projeto e seleciona o botão “Salvar”.</li> <li>2.5. O Sistema grava o projeto.</li> </ol>
<b>Fluxo Alternativo 2</b>	<p>No passo 2, do fluxo principal, caso o ator deseje excluir um projeto, deve selecionar um projeto já existente.</p> <ol style="list-style-type: none"> <li>2.1. O ator consulta os projetos cadastrados e seleciona um projeto.</li> <li>2.2. O sistema carrega os dados da capa do projeto.</li> <li>2.3. O ator seleciona o botão “Excluir”.</li> <li>2.5. O sistema exclui o projeto.</li> </ol>
<b>Fluxo Alternativo 3</b>	<p>No passo 2, do fluxo principal, caso o ator deseje consultar o histórico do projeto, deve selecionar um projeto já existente.</p> <ol style="list-style-type: none"> <li>2.1. O ator consulta os projetos cadastrados e seleciona um projeto.</li> <li>2.2. O sistema carrega os dados da capa do projeto.</li> <li>2.3. O ator seleciona a aba “Histórico Projeto”.</li> <li>2.4. O sistema carrega dados da consulta.</li> <li>2.5. O ator consulta o histórico do projeto.</li> </ol>
<b>Fluxo de Exceção 1</b>	No passo 2.3, do fluxo alternativo 2, o sistema verifica se existem apontamentos realizados para o projeto e apresenta mensagem de alerta.
<b>Pós-condição</b>	Ator administrador cadastrou, editou, excluiu e consultou projetos.

Quadro 9 – Descrição do caso de uso UC10

<b>Caso de uso</b>	UC10 – Manter Tarefas
<b>Descrição</b>	O sistema deve permitir ao ator administrador cadastrar, editar, excluir e consultar tarefas do projeto.
<b>Ator</b>	Administrador.
<b>Pré-condição</b>	1. Administrador deve fazer <i>login</i> no sistema. 2. Capa do projeto deve estar cadastrada.
<b>Fluxo Principal</b>	1. O ator entra na tela de cadastro de projeto. 2. O sistema apresenta somente o campo “Projeto” habilitado. 3. O ator consulta os projetos cadastrados e seleciona um projeto. 4. O sistema habilita edição dos dados da capa do projeto. 5. O ator seleciona a aba “Tarefas”. 6. O sistema apresenta somente o campo “Tarefa” habilitado. 7. O ator seleciona o botão “Novo”. 8. O sistema habilita edição dos dados da tarefa. 9. O ator preenche dados da tarefa e seleciona o botão “Salvar”. 10. O sistema grava a tarefa do projeto.
<b>Fluxo Alternativo 1</b>	No passo 7, do fluxo principal, caso o ator deseje alterar a tarefa, deve selecionar uma tarefa já existente. 7.1. O ator consulta as tarefas cadastradas e seleciona uma tarefa. 7.2. O sistema carrega os dados da tarefa. 7.3. O sistema habilita edição dos dados da tarefa. 7.4. O ator edita os dados da tarefa e seleciona o botão “Salvar”. 7.5. O sistema grava a tarefa do projeto.
<b>Fluxo Alternativo 2</b>	No passo 7, do fluxo principal, caso o ator deseje excluir uma tarefa, deve selecionar uma tarefa já existente. 7.1. O ator consulta as tarefas cadastradas e seleciona uma tarefa. 7.2. O sistema carrega os dados da tarefa. 7.3. O ator seleciona o botão “Excluir”. 7.5. O sistema exclui a tarefa do projeto.
<b>Fluxo Alternativo 3</b>	No passo 7, do fluxo principal, caso o ator deseje consultar os apontamentos realizados para as tarefa, deve selecionar a aba “Lançamentos por Tarefa”. 7.1. O sistema carrega dados da consulta. 7.2. O ator consulta os lançamentos realizados por tarefa.
<b>Fluxo de Exceção 1</b>	No passo 7.3, do fluxo alternativo 2, o sistema verifica se existem apontamentos realizados para a tarefa e apresenta mensagem de alerta.
<b>Pós-condição</b>	Ator administrador cadastrou, editou, excluiu e consultou tarefas no projeto.

Quadro 10 – Descrição do caso de uso UC16

<b>Caso de uso</b>	UC16 – Consultar Análise de Projetos
<b>Descrição</b>	O sistema deve permitir aos atores administrador, colaborador e cliente consultarem os projetos.
<b>Atores</b>	Administrador, colaborador e cliente.
<b>Pré-condição</b>	Administrador, colaborador e cliente devem fazer <i>login</i> no sistema.
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. O ator acessa a tela do sistema.</li> <li>2. O sistema apresenta todos os projetos que atendem aos filtros iniciais da tela.</li> <li>3. O ator seleciona um projeto.</li> <li>4. O sistema abre a tela do quadro <i>Kanban</i> do projeto do caso de uso UC13</li> </ol>
<b>Fluxo de Exceção 1</b>	No passo 2 do fluxo principal, o sistema apresenta mensagem de alerta caso não encontre projetos que atendam aos filtros informados.
<b>Pós-condição</b>	Administrador, colaborador ou cliente consultou um projeto.

Quadro 11 – Descrição do caso de uso UC13

<b>Caso de uso</b>	UC13 – Movimentar Cartões no Quadro <i>Kanban</i>
<b>Descrição</b>	O sistema deve permitir aos atores administrador, colaborador e cliente movimentarem os cartões no quadro <i>Kanban</i> .
<b>Atores</b>	Administrador, colaborador e cliente.
<b>Pré-condição</b>	1. Ator deve fazer <i>login</i> no sistema.
<b>Fluxo Principal</b>	<ol style="list-style-type: none"> <li>1. O ator entra na tela do quadro <i>Kanban</i> do projeto selecionado no caso de uso UC16.</li> <li>2. O sistema apresenta as tarefas em forma cartão no quadro.</li> <li>3. O ator seleciona o cartão e movimenta entre as colunas do quadro.</li> <li>4. O sistema grava informações da movimentação.</li> </ol>
<b>Fluxo Alternativo 1</b>	<p>No passo 3, do fluxo principal, caso o ator deseje alterar a tarefa, deve clicar com o botão esquerdo do mouse sobre o cartão.</p> <ol style="list-style-type: none"> <li>3.1. O sistema abre a tela de cadastro de projeto na aba tarefas.</li> <li>3.2. O sistema carrega os dados da tarefa.</li> <li>3.3. O sistema habilita apenas o campo de observação dos dados da tarefa.</li> <li>3.4. O ator edita o campo de observação da tarefa e seleciona o botão “Salvar”.</li> <li>3.5. Sistema grava os dados de edição da tarefa do projeto.</li> </ol>
<b>Fluxo Alternativo 2</b>	<p>No passo 3, do fluxo principal, caso o ator deseje realizar o apontamento de horas realizadas para uma tarefa, deve clicar com o botão central do mouse sobre o cartão.</p> <ol style="list-style-type: none"> <li>3.1. O sistema abre a tela de apontamento de horas do caso de uso UC12.</li> </ol>
<b>Fluxo de Exceção 1</b>	No passo 3, do fluxo principal, o sistema verifica a quantidade de cartões da coluna mais o cartão que está sendo movimentado e apresenta mensagem de alerta caso o numero total de cartões da coluna ultrapasse o WIP estabelecido na configuração da capa do projeto.
<b>Pós-condição</b>	Atores administrador, colaborador e cliente movimentaram os cartões no quadro <i>Kanban</i> .



Quadro 12 – Descrição do caso de uso UC12

<b>Caso de uso</b>	UC12 – Apontamento de Horas
<b>Descrição</b>	O sistema deve permitir aos atores administrador, colaborador e cliente realizar o apontamento de horas realizadas por tarefa do projeto.
<b>Atores</b>	Administrador, colaborador e cliente.
<b>Pré-condição</b>	1. Ator deve fazer <i>login</i> no sistema.
<b>Fluxo Principal</b>	1. O sistema apresenta os campos habilitados 2. O ator preenche dados do apontamento e seleciona o botão salvar. 3. O sistema grava informações do apontamento.
<b>Fluxo Alternativo 1</b>	No passo 1, do fluxo principal, caso o ator esteja fazendo o apontamento pelo quadro <i>Kanban</i> no caso de uso UC13, o sistema deve apresentar os campos “projeto” e “tarefa” desabilitados. 1.1. O sistema habilita os campos “data”, “tipo serviço”, “horas realizadas” e “anotação”. 1.2. O ator preenche os campos e seleciona o botão “Salvar”. 1.3. O sistema grava informações do apontamento.
<b>Fluxo Alternativo 2</b>	No passo 1, do fluxo principal, caso o ator deseje alterar o apontamento, deve selecionar um apontamento já existente. 1.1. O ator consulta os apontamentos cadastrados e seleciona um apontamento. 1.2. O sistema carrega os dados do apontamento. 1.3. O sistema habilita edição dos dados do apontamento. 1.4. O ator edita os dados do apontamento e seleciona o botão “Salvar”. 1.5. O sistema grava informações do apontamento
<b>Fluxo Alternativo 3</b>	No passo 1, do fluxo principal, caso o ator deseje realizar a exclusão do apontamento de horas realizadas para uma tarefa, deve selecionar um apontamento já existente. 1.1. O ator consulta os apontamentos cadastrados e seleciona um apontamento. 1.2. O sistema apresenta os dados do apontamento realizado 1.3. O ator seleciona o botão “Excluir”. 1.4. O sistema exclui apontamento.
<b>Pós-condição</b>	Ator cadastrou, editou, excluiu e consultou o apontamento de horas.