

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

RODOLFO RIBEIRO CÂNDIDO

Rastro 2.0
Módulo Manutenção

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2014

RODOLFO RIBEIRO CÂNDIDO

Rastro 2.0

Módulo Manutenção

Trabalho de conclusão de curso de graduação, apresentada à disciplina 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 obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Danilo Sipoli Sanches.

CORNÉLIO PROCÓPIO

2014

Dedico a minha família, namorada, amigos e
especialmente à Deus por esta importante
realização em minha vida.

AGRADECIMENTOS

Agradeço primeiramente à Deus que me concedeu saúde e permitiu que eu concluísse este trabalho após todos este tempo de dedicação à faculdade.

Agradeço à minha família que sempre me apoiou nos momentos mais difíceis, que mesmo longe sempre se fizeram presente no meu dia-a-dia.

Aos amigos que por muitas vezes acabaram fazendo eu perder algumas horas de sono para compartilhar de momentos alegres e tristes.

À minha adorável namorada por ser tão compreensiva e me incentivar sempre para a conclusão deste trabalho.

Ao pessoal da ForLogic, lugar onde eu aprendi muitas das coisas que sei.

Agradeço a todos que, direta ou indiretamente, participaram ou torceram pela finalização dessa etapa.

“Se você faz o que sempre fez, obterá o que você sempre obteve.”
(Anthony Robbins)

RESUMO

CANDIDO, Rodolfo R.. **Rastro 2.0: Módulo Manutenção**. 2014. 97p. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Cornélio Procópio 2014.

Este documento descreve o desenvolvimento de um módulo *Web* para realizar o acompanhamento e gerenciamento do ciclo de vida de solicitações para realização de manutenção em sistemas de informação. A importância do desenvolvimento deste sistema é a dificuldade em manter centralizadas as mudanças e correções que devem ser feitas. Este foi desenvolvido baseado em tecnologias Microsoft como: C# e ASP.Net MVC, métodos ágeis baseados no *Scrum*.

Palavra-chave: Manutenção. Gerenciamento. Fluxo. Solicitação.

ABSTRACT

CANDIDO, Rodolfo R.. **Rastro 2.0: Maintenance Module**. 2014. 97p. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Cornélio Procópio 2014.

This document describes the development of a web module to perform monitoring and managing the life cycle of applications to perform maintenance information systems. The importance of the development of this system is the difficulty in maintaining centralized changes and corrections to be made. This was developed based on Microsoft technologies such as C # and ASP.Net MVC, Agile Scrum based.

Keywords: Maintenance. Management. Workflow. Request.

LISTA DE FIGURAS

Figura 1 - O ciclo de vida clássico (Modelo em cascata)	16
Figura 2 - Níveis de Maturidade MPS.Br	20
Figura 3 - Atividades do Scrum	22
Figura 4 - Processo 4L versão 2.9	23
Figura 5 - Concepção	24
Figura 6 - Elaboração	25
Figura 7 - Construção	26
Figura 8 - Transição	27
Figura 9 - Solicitação de Mudança	28
Figura 10 - Padrão MVC	31
Figura 11 - Caso de Uso: Novo Chamado (Cliente)	35
Figura 12 - Caso de Uso: Novo Chamado (ForLogic)	35
Figura 13 - Caso de Uso: Avaliar Trabalho	36
Figura 14 - Caso de Uso: Funcionalidades comuns da ForLogic	36
Figura 15 - Diagrama de classe	37
Figura 16 - Diagrama de Atividades	38
Figura 17 - Gerência de Classificações	39
Figura 18 - Cadastro/Edição de Classificações	39
Figura 19 - Gerência de Níveis	40
Figura 20 - Cadastro/Edição de Níveis	40
Figura 21 - Gerência de Contatos	41
Figura 22 - Cadastro/Edição de Contato	41
Figura 23 - Gerência de Usuários	42
Figura 24 - Cadastro/Edição de Usuários	42
Figura 25 - Gerência de Tipos de Produtos	43
Figura 26 - Cadastro/Edição de Tipos de Produtos	43
Figura 27 - Gerência de Produtos	44
Figura 28 - Cadastro/Edição de Produtos	45
Figura 29 - Gerência de Chamados (Forlogic)	46
Figura 30 - Cadastro de Chamado (Forlogic)	47
Figura 31 - Reestimar chamado	48

Figura 32 - Avaliar reabertura do chamado.....	49
Figura 33 - Classificar e estimar chamado	50
Figura 34 - Cancelar chamado	50
Figura 35 - Finalizar chamado.....	51
Figura 36 - Anexar arquivos ao chamado	51
Figura 37 - Comentar chamado.....	52
Figura 38 - Informar tempo trabalhado.....	53
Figura 39 - Informar progresso do chamado	54
Figura 40 - Visualizar chamado (Aba: Informações do Chamado).....	54
Figura 41 - Visualizar chamado (Aba: Classificação e Estimativa).....	55
Figura 42 - Visualizar chamado (Aba: Histórico)	55
Figura 43 - Visualizar chamado (Aba: Trabalho Realizado)	55
Figura 44 - Visualizar chamado (Aba: Comentários).....	56
Figura 45 - Gerência de Chamados (Cliente).....	56
Figura 46 - Avaliar classificação e estimativa do chamado	57
Figura 47 - Avaliar trabalho do chamado	58
Figura 48 - Cancelar chamado	58
Figura 49 - Anexar arquivos ao chamado	59
Figura 50 - Comentar chamado.....	59
Figura 51 - Visualizar chamado (Aba: Informações do Chamado).....	60
Figura 52 - Visualizar chamado (Aba: Classificação e Estimativa).....	60
Figura 53 - Visualizar chamado (Aba: Histórico)	61
Figura 54 - Visualizar chamado (Aba: Trabalho Realizado)	61
Figura 55 - Visualizar chamado (Aba: Comentários).....	61
Figura 56 - E-mail novo chamado	62
Figura 57 - E-mail classificação e estimativa.....	63
Figura 58 - E-mail avaliação da classificação e estimativa	63
Figura 59 - E-mail comentário	64
Figura 60 - E-mail cancelamento.....	64
Figura 61 - E-mail finalização	65
Figura 62 - E-mail avaliação do trabalho	65
Figura 63 - E-mail avaliação da reabertura	66
Figura 64 - E-mail reestimar	66

LISTA DE TABELAS

Tabela 1 - Concepção	25
Tabela 2 - Elaboração	26
Tabela 3 - Construção	27
Tabela 4 - Transição	28
Tabela 5 - Solicitação de Mudança	29
Tabela 6 - Resultados análise do questionário.....	68
Tabela 7 - Cronograma	68

LISTA DE ABREVIATURA

AJAX	<i>Asynchronous Javascript and XML</i>
CMMI	<i>Capability Maturity Model – Integration</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JS	<i>JavaScript</i>
MPS.BR	Melhoria de Processos de Software Brasileiro
MVC	<i>Model-View-Controller</i>
SOFTEX	Associação para Promoção da Excelência do Software Brasileiro
SCM	<i>Software Configuration Management</i>
SQL	<i>Structured Query Language</i>
UTFPR	Universidade Tecnológica Federal do Paraná
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1. INTRODUÇÃO	13
2. METODOLOGIA	16
2.1 PROCESSO DE DESENVOLVIMENTO ÁGIL	17
2.2 MPS.BR.....	19
2.3 SCRUM	21
2.4 PROCESSO 4L.....	22
3. TECNOLOGIAS UTILIZADAS.....	30
4. MODELAGEM.....	34
5. SISTEMA	39
5.1 CLASSIFICAÇÕES	39
5.2 NÍVEIS.....	40
5.3 CONTATOS	41
5.4 USUÁRIOS.....	42
5.5 TIPOS DE PRODUTOS	43
5.6 PRODUTOS	44
5.7 CHAMADOS (FORLOGIC).....	45
5.8 CHAMADOS (CLIENTE)	56
5.9 E-MAILS	62
6. CONSIDERAÇÕES FINAIS	67
6.1 DIFICULDADES ENCONTRADAS.....	67
6.2 TRABALHOS FUTUROS	67
6.3 RESULTADOS	67
6.4 CRONOGRAMA.....	68
REFERÊNCIAS.....	69
APÊNDICE A	71

1. INTRODUÇÃO

Gerenciamento de configuração de software (*Software Configuration Management – SCM*) é uma atividade aplicada em todo o ciclo de vida de desenvolvimento de um software (PRESSMAN, 1995). Este é o ponto onde este trabalho situa-se na Engenharia de *Software*, pois, na área de desenvolvimento de sistemas é comum acontecer mudanças no projeto ao longo do seu desenvolvimento e ciclo de vida. As mudanças podem ser originadas de diversas maneiras impossíveis de serem previstas, tais como alteração da infraestrutura de onde o sistema está implantado, correção de erros identificados nos fluxos e inclusive de novas funcionalidades solicitadas pelo cliente. Deste modo, tais mudanças gera modificação dos elementos criados no desenvolvimento inicial do projeto, chamados de artefatos. Todo o processo de análise desses artefatos e desenvolvimento das alterações devem ser gerenciado.

O controle do tempo destinado a mudanças em um projeto de desenvolvimento de sistemas é importante, observando que nesta área muitas vezes a forma de cobrança é em horas.

Este trabalho apresenta o processo de desenvolvimento de um sistema visando gerenciar as solicitações feitas por clientes proprietários de sistemas, chamadas de solicitações. Uma solicitação até chegar ao momento de realmente ser desenvolvida pela equipe técnica, a mesma passa por diversos passos estipulados pela empresa que desenvolve o software. Na *ForLogic Software* onde este é utilizado, esse processo é definido pelo setor de qualidade, porém, o fluxo das solicitações serão abordadas detalhadamente nos próximos tópicos.

Existem no mercado algumas ferramentas que possuem o mesmo intuito da que é proposta nesse trabalho, algumas até bem conhecidas no mercado como o caso do TFS (*Team Foundation Server*, da Microsoft), o *Redmine*, que é um software desenvolvido em Ruby on Rails e distribuído sob os termos GPL (*General Public License*), para a *ForLogic*, empresa na qual foi feito este desenvolvimento não foi interessante optar por uma dessas ferramentas, sendo assim a necessidade do estudo do fluxo para gerenciar as manutenções de software e a criação do *software*.

O sistema desenvolvido foi baseado em tecnologias *Web*, utilizando a infraestrutura fornecida pela empresa *Forlogic Software*, foi construído para atender

o fluxo das solicitações que foi desenhado pelo setor de qualidade da empresa. Provavelmente o fluxo não é o mesmo em outras empresas.

A Forlogic fundada por ex-alunos da UTFPR, oriunda de um projeto que se iniciou no Hotel Tecnológico da própria universidade, em 18 de maio de 2004, em Cornélio Procópio – PR. Resultado do primeiro projeto de sucesso na disciplina de empreendedorismo, direcionada ao desenvolvimento de soluções de *software*, que atua em alguns nichos, oferecendo solução de produtos específicos e de fábrica de *software*.

O Produto são ferramentas vendidas em forma de pacote, a Forlogic disponibiliza suporte, conforme o plano contratado e a regra do *software* definida e implementada.

Fábrica de *software* é a criação de um produto, totalmente novo que o cliente não havia encontrado uma solução que atendesse a suas necessidades no *software* pronto. Deste modo, é realizado um levantamento de requisitos e o *software* é desenvolvido sob demanda para atender as necessidades do cliente.

As diversas opções de desenvolvimento, produto ou fábrica, é necessário ter um controle do que está sendo feito, como, por quem, quanto tempo está sendo gasto, ou seja, deve ser possível gerenciar todo o desenvolvimento da empresa.

Atualmente, a empresa possui uma carteira de clientes vasta que utiliza diversos de seus produtos e softwares desenvolvidos sob demanda.

Uma reunião realizada no início do mês de fevereiro de 2013, com o coordenador de projetos e o gerente de desenvolvimento da empresa, levantou-se à necessidade de aplicar conceitos tecnológicos para desenvolver uma ferramenta capaz de gerenciar todo o ciclo de vida dos projetos desenvolvidos pela empresa.

Uma das grandes dificuldades para a Forlogic é o gerenciamento dos projetos, considerando as diversas variáveis que devem ser acompanhadas diariamente para manter sempre os cronogramas.

A manutenção solicitada pelo cliente em um produto de *software* exige acompanhamento, deve ser analisado e aprovado, dando início a atualização de documentação e homologação após o desenvolvimento, verificando os resultados do trabalho.

Desde o início da solicitação é necessário manter em uma única ferramenta todo o controle das solicitações de manutenção. Nesse sentido é possível verificar a importância do desenvolvimento de um módulo para o gerenciamento das

solicitações feitas pelos clientes para a manutenção, como correção, ajuste ou melhoria em algum processo do produto contratado.

Para que haja um controle sobre todo o ciclo de vida da solicitação em questão, é necessário existir um fluxo ou um procedimento para fazer o atendimento.

Primeiramente, a solicitação inicia o ciclo de vida com a equipe de suporte, que analisa a origem do item e estima o tempo necessário para o desenvolvimento requisitado (dando início ao atendimento). O atendimento pode ser pela equipe do suporte ou equipe de desenvolvimento, que utilizará as mesmas informações, contudo, em outra interface fora do escopo deste trabalho.

A ferramenta de gerenciamento das solicitações foi desenvolvida utilizando o processo de desenvolvimento definido pela *Forlogic* atendendo aos requisitos para certificação MPS.Br nível F. Para isso é necessário a geração de diversos artefatos e evidenciados seguindo exigências do processo de desenvolvimento. No segundo capítulo são apresentados alguns com uma breve explicação.

O está organizado em cinco capítulos:

No primeiro capítulo apresentamos uma breve introdução, contendo o objetivo do trabalho, para quem e como foi desenvolvido o sistema.

No segundo capítulo descrevemos os métodos utilizados para o desenvolvimento e alguns dos artefatos gerados.

No terceiro capítulo apresentamos as tecnologias utilizadas e uma breve descrição de cada uma.

No quarto capítulo é apresentado a modelagem que foi realizada para que fosse possível o desenvolvimento do sistema.

No quinto capítulo é apresentado o sistema, as principais páginas (telas) exibindo algumas informações e as descrições das funcionalidades.

No sexto capítulo, são apresentados as considerações finais composta das dificuldades encontradas, trabalhos futuros e conclusão do trabalho.

2. METODOLOGIA

Avaliando o ciclo de vida clássico (Modelo em cascata) da Engenharia de *Software* observamos que é possível aplica-lo na realização deste trabalho.

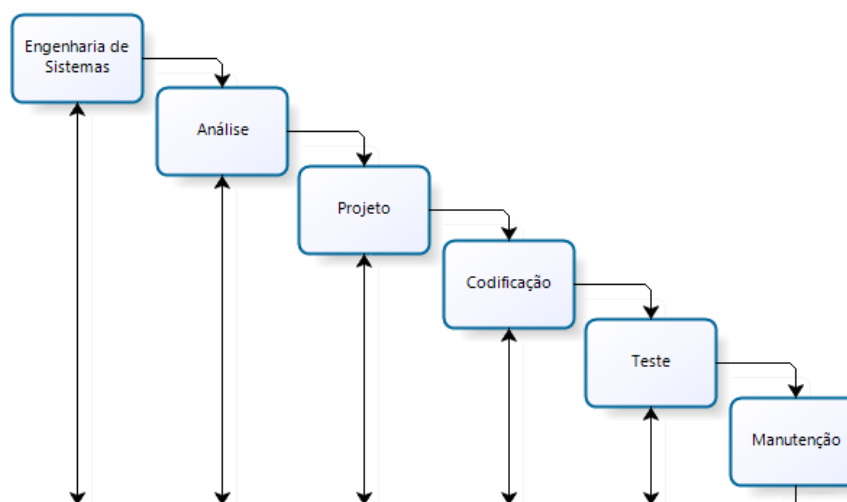


Figura 1 - O ciclo de vida clássico (Modelo em cascata).
Fonte: Sommerville (2007)

O ciclo de vida clássico deixa separada cada fase de um processo em cascata, onde somente uma fase será iniciada após a conclusão de sua antecessora, e por fim temos a fase de manutenção.

Indubitavelmente, o software sofrerá mudanças depois que foi entregue ao cliente (uma possível exceção é o software embutido). Ocorrerão mudanças porque erros foram encontrados, porque o software deve ser adaptado a fim de acomodar mudanças em seu ambiente externo (por exemplo, uma mudança exigida por causa de um novo sistema operacional ou dispositivo periférico) ou porque o cliente exige acréscimos funcionais ou de desempenho. A manutenção de software re replica cada uma das etapas precedentes do ciclo de vida a um programa existente, e não a um novo (PRESSMAN, 1995).

Segundo Sommerville (2007), geralmente (embora não necessariamente) esta é a fase mais longa do ciclo de vida. Pois pode haver a necessidade de corrigir erros não identificados nas outras fases do processo de desenvolvimento. Neste momento, ocorre à implementação de novas funcionalidades e/ou módulos ao sistema, a partir de novos requisitos.

O módulo apresentado nesse trabalho realiza o gerenciamento da fase do ciclo de vida citado acima, fornece o controle das solicitações dos clientes para a empresa. As solicitações devem ser gerenciadas e seguir um fluxo definido para a equipe interna.

O processo de desenvolvimento de *software* utilizado no desenvolvimento deste trabalho, foi desenvolvido pela ForLogic e suas fases e atividades estão apresentadas nesta seção.

Segundo Pressman (2002), o desenvolvimento de *software* um fator determinante para obter os resultados esperados, que foram definidos no início do projeto é a utilização de um ciclo de vida, definindo de forma explícita o que deve ser feito durante todo o processo. Pressman (2002) afirma que no processo de desenvolvimento permite-se ajustes, definindo o processo da melhor maneira, a fim de evoluir em velocidade, qualidade, gerência de riscos e custos, mantendo tudo sobre controle desde o início. A Forlogic fez justamente isso e definiu o seu processo de desenvolvimento, afim de melhorar os mecanismos e tarefas para obter melhores resultados.

2.1 PROCESSO DE DESENVOLVIMENTO ÁGIL

Faz algum tempo em que o desenvolvimento de *software* vem sendo tratado de forma comparável a atividade de produção industrial. Juntamente com essa forma de se trabalhar semelhante vem uma série de atividades e suas respectivas formas de serem executadas conhecido como modelo de processo em cascata (ROYCE, 1970). Este modelo propõem que para cada atividade que seja executada, outra somente será iniciada no momento em que a anterior impreterivelmente tenha sido executada completamente, é nesse momento, para agilizar todo esse processo de produção que entra então o processo de desenvolvimento ágil.

É possível associar o paradigma taylorista, resultado do trabalho do engenheiro Taylor (1995), onde tem por objetivo a produção (desenvolvimento) em série, o que melhora a relação de quantidade com custo e tempo, para isso é necessária a racionalização da organização. Essa racionalização propõem a separação das atividades, dividindo ao máximo as etapas do processo de produção, ou seja, o que antes uma pessoa fazia em determinado período, passa a ser feito por quatro pessoas diferentes por exemplo e levando quatro vezes menos tempo, ou pensando de uma forma diferente, o tempo que era necessário para produzir aquela quantidade com utilização deste conceito acaba por gerar quatro vezes mais determinado produto, aumento o lucro.

Voltando o raciocínio para área industrial, para elevar a produtividade da produção humana, foi criado um método no qual os trabalhadores eram dispostos ao lado das esteiras rolantes que traziam as peças a serem montadas e assim realizam um tarefa muito simples e específica, fazendo com que as atividades fossem realizadas quase que automático pelas pessoas, aumentando a concentração e foco de todos nesse processo de produção (TAYLOR, 1995).

Nesta abordagem percebe-se que para o sucesso, cada membro deve ser responsável e qualificado para realizar determinada tarefa, no desenvolvimento de software essa separação de atividades também ocorre, conseqüentemente, de papéis, por exemplo em uma equipe de desenvolvimento temos o programador que é muito bem qualificado somente para programar e faz somente isso, temos também o líder de projeto, com ótimas habilidades de liderança, organização de projeto, cronograma e faz somente isso, temos os analistas, também especializados nesta função e assim por diante.

Softwares possuem particularidades únicas que os tornam produtos especiais. A imensa quantidade possível de coisas e formas que um software pode assumir o leva a ser considerado um objeto altamente complexo. No momento em que é necessário realizar alguma alteração em um software já existente, o nível de complexidade para se trabalhar com o mesmo cresce não linearmente mas sim exponencialmente devido ao aumento nos elementos envolvidos. Além disso, software está presente nas mais diversas áreas que por si só já são complexas e sofrem mudanças constantes. Diante de toda essa complexidade é essencial que as mudanças no software torna-se uma tarefa fácil sem muitas complicações (BROOKS, 1995).

Desenvolvimento ágil é um nome comum que foi dado para os processos de desenvolvimento de software que se baseiam no Manifesto Ágil (BECK et al., 2001). Este manifesto foi assinado por pelos dos maiores nomes do grupo de desenvolvimento de software, é um breve documento relatando quatro valores e doze princípios que sugerem: estreita colaboração e proximidade entre a equipe de desenvolvimento e os clientes; o reconhecimento de que mudanças nos requisitos representam aprendizado e oportunidades de obtenção de vantagens competitivas; busca por simplicidade e eliminação de documentações e processos desnecessários; ênfase nas pessoas em relação a processos; e o estabelecimento da geração de valor aos clientes como o objetivo por excelência e como principal

métrica de progresso. Todos pontos citados não são novidades, o manifesto ágil apresenta todos os conceitos como um todo de forma aplicável e coesa.

Um fator plausível para explicar o porquê a indústria continua utilizando e defendendo os processos de desenvolvimento tradicionais é a constante presença da relação entre custo de correção do software (BOHEM, 1981).

Prevendo evitar investimentos durante o processo e realizar mudanças no decorrer do desenvolvimento por falta de detalhamento ou até falta de amadurecimento dos requisitos, os processos tradicionais definem fases iniciais extremamente grandes. Nestas fases iniciais o foco principal é realizar todo o planejamento do sistema, é tentar reunir todas as solicitações, tudo o que o sistema deve ter e detalhar ao máximo para que não falte nada e tudo seja bem entendido para que uma possível mudança no futuro não ocorra. De uma forma geral os processos tradicionais defendem que quanto maior for o detalhamento inicial maior que chance do projeto obter sucesso.

Além do levantamento, da geração de documentação abrangente, com grande número de artefatos, é prática padrão de métodos e certificações de processo associadas ao desenvolvimento tradicional como CMMI, MPS.BR e outros.

2.2 MPS.BR

No setor de desenvolvimento de software a garantia da qualidade, não é mais um diferencial no mercado mas um pré-requisito para ser mais competitivo e ter uma boa visibilidade no setor.

O MPS.Br significa melhoria de processo do software brasileiro, projeto prioritário da sociedade SOFTEX, que visa o aumento da competitividade no mercado de desenvolvimento de software no país, promovido por ações de empreendedorismo, capacitação, apoio à capitalização e ao financiamento, apoia à geração de negócio do Brasil. A SOFTEX tem como missão realizar a transformação no setor e fazer com que o Brasil atinja o ápice da comercialização de software, sendo líder mundial de produção e exportação de software (Modelo de Referência para Melhoria de Processo de Software, 2004).

O MPS.Br não sugere nada novo, apenas aplica os conceitos que foram assinados no manifesto ágil, fornece um guia para que os mesmos sejam aplicados, garantindo a qualidade esperada no desenvolvimento de software.

Os benefícios da certificação MPS para a empresa é o selo que permite a participação em algumas licitações governamentais, gerando diversas melhorias à empresa como: novas oportunidades para a empresa, expansão da área de atuação, aumento da renda por consequência da expansão, aumento da visibilidade no mercado, entre outras.

Destamos nesta metodologia utilização das principais abordagens internacionais voltadas para definição, a avaliação e a melhoria dos processos de software, com as práticas do CMMI.

O CMMI tem níveis de maturidade, o MPS.Br aplica essa hierarquia, de forma diferenciada, os cinco níveis do CMMI, passam a ser sete níveis assegurando o cliente a garantia da qualidade no processo de desenvolvimento de software que a empresa fornece.

A Figura 2, representa os níveis de maturidade existentes no MPS.Br:

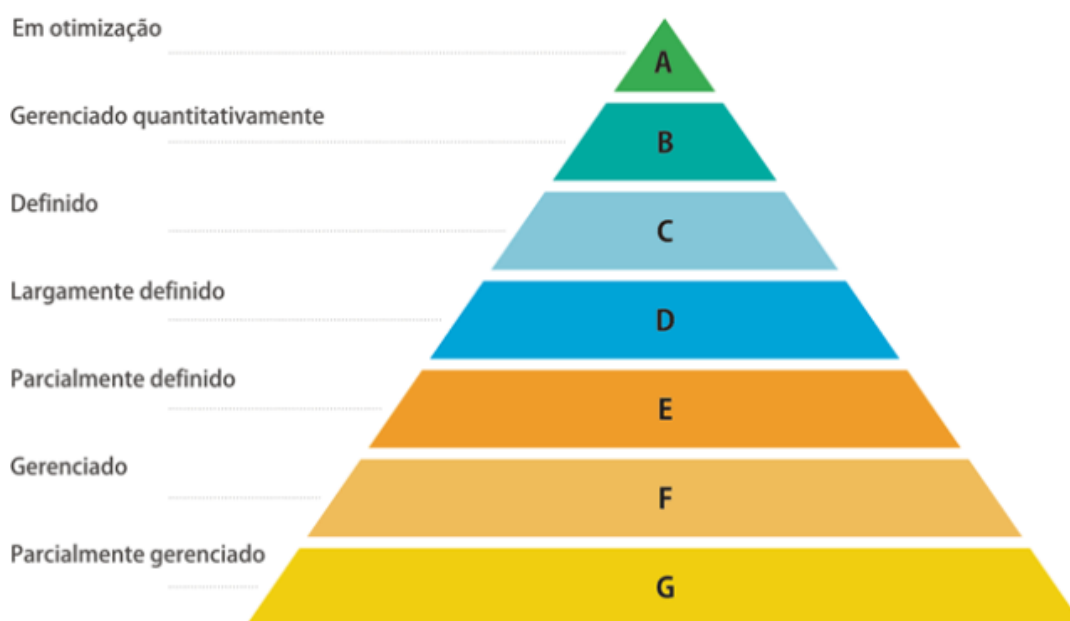


Figura 2 - Níveis de Maturidade MPS.Br
Fonte: FUMSOFT - http://www.fumsoft.org.br/qualidade/modelo_mpsbr

A seguir uma breve descrição do que significa cada nível de maturidade, lembrando que nível “A” é o que agrega e possui maior valor dentro deste modelo.

No contexto geral o MPS.Br não é concorrente do CMMI, mas uma alternativa, que possibilita a micro, pequenas e médias empresas ser qualificadas,

almejando chegar à algum nível da pirâmide do CMMI. Uma grande vantagem do MPS.Br sobre o CMMI é o valor do investimento.

A – Em Otimização: há a preocupação com questões como inovação e análise de causas.

B – Gerenciado Quantitativamente: avalia-se o desempenho dos processos, além da gerência quantitativa dos mesmos.

C – Definido: aqui ocorre o gerenciamento de riscos.

D – Largamente Definido: envolve verificação, validação, além da liberação, instalação e integração de produtos, dentre outras atividades.

E – Parcialmente Definido: considera processos como treinamento, adaptação de processos para gerência de projetos, além da preocupação com a melhoria e o controle do processo organizacional.

F – Gerenciado: introduz controles de medição, gerência de configuração, conceitos sobre aquisição e garantia da qualidade.

G – Parcialmente Gerenciado: neste ponto inicial deve-se iniciar o gerenciamento de requisitos e de projetos.

2.3 SCRUM

Segundo Schwaber e Sutherland (2013), Scrum é um framework estrutural que vem sendo utilizado no auxílio para desenvolver e manter projetos complexos desde 1990. Alguns pontos destacados no Guia Oficial do Scrum são: Leveza; Facilidade de entendimento; Dificuldade no domínio. Este apoia a importância de cada membro do time em desempenhar atividades específicas e essenciais para uso e sucesso do Scrum.

Como toda prática é fundamentada em teoria, o Scrum é fundamentado em teorias empíricas que afirmam que o conhecimento é resultado de experiência e de tomada de decisões baseada em fatos conhecidos. O Scrum emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e a gerência de riscos. Três pilares apoiam a implementação da gerência de processo empírico: transparência (definições comuns e linguagens utilizada para comunicação entre os envolvidos devem ser padronizadas e de fácil acesso), inspeção (deve existir a frequente verificação dos artefatos gerados, porém, nem tão frequente e por outra equipe para não prejudicar a execução das tarefas) e adaptação (quando desvios

forem detectados em projetos, deve-se realizar ajustes brevemente para evitar e/ou minimizar mais desvios).

O time Scrum é composto por três membros, *Product Owner* (dono do produto, responsável pelo *backlog* do produto que compõem os projetos que são e serão executados), Time de Desenvolvimento (responsáveis por transformar o *backlog* definido pelo *Product Owner* em resultados visíveis em cada iteração do framework, destaca-se a habilidade da equipe em se organizar para atingir esse objetivo) e o *Scrum Master* (auxilia o time com relação ao entendimento do que deve ser feito e todos envolvidos à compreender como o time organiza e realiza as entregas esperadas). Destaca-se que o time são auto organizáveis (desnecessário a presença de outros membros externo ao projeto para gerenciar as atividades a serem executadas) e multifuncionais (todos os membros são capacitados para realização das atividades exigidas pelo framework, isentando a necessidade de muitos envolvidos), o modelo de time proposto visa a evolução e melhoria contínua em flexibilidade, criatividade e produtividade.

A Figura 3 representa a forma como é utilizado o framework Scrum.

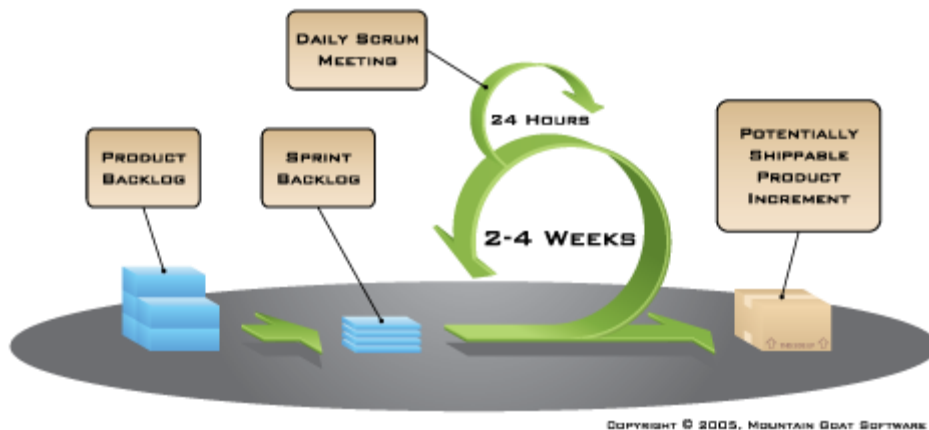


Figura 3 - Atividades do Scrum

Fonte: Scrum Overview - <http://www.mountaingoatsoftware.com/agile/scrum/overview>

2.4 PROCESSO 4L

A Forlogic desde a sua criação mantém uma equipe especializada no estudo de processos de desenvolvimento e de qualidade, a fim de adquirir experiências nas diversas atividades de todo o ciclo de desenvolvimento de *software*. O processo de desenvolvimento de software utilizado atualmente na Forlogic é baseado no ciclo de vida iterativo incremental, baseado no Scrum, utiliza atores essenciais do framework

e as principais atividades propostas, porém, com algumas modificações que foram adaptadas para atender a realidade da empresa. Atualmente esse processo adaptado está na sua versão 2.9 com sua estrutura apresentada na Figura 4.

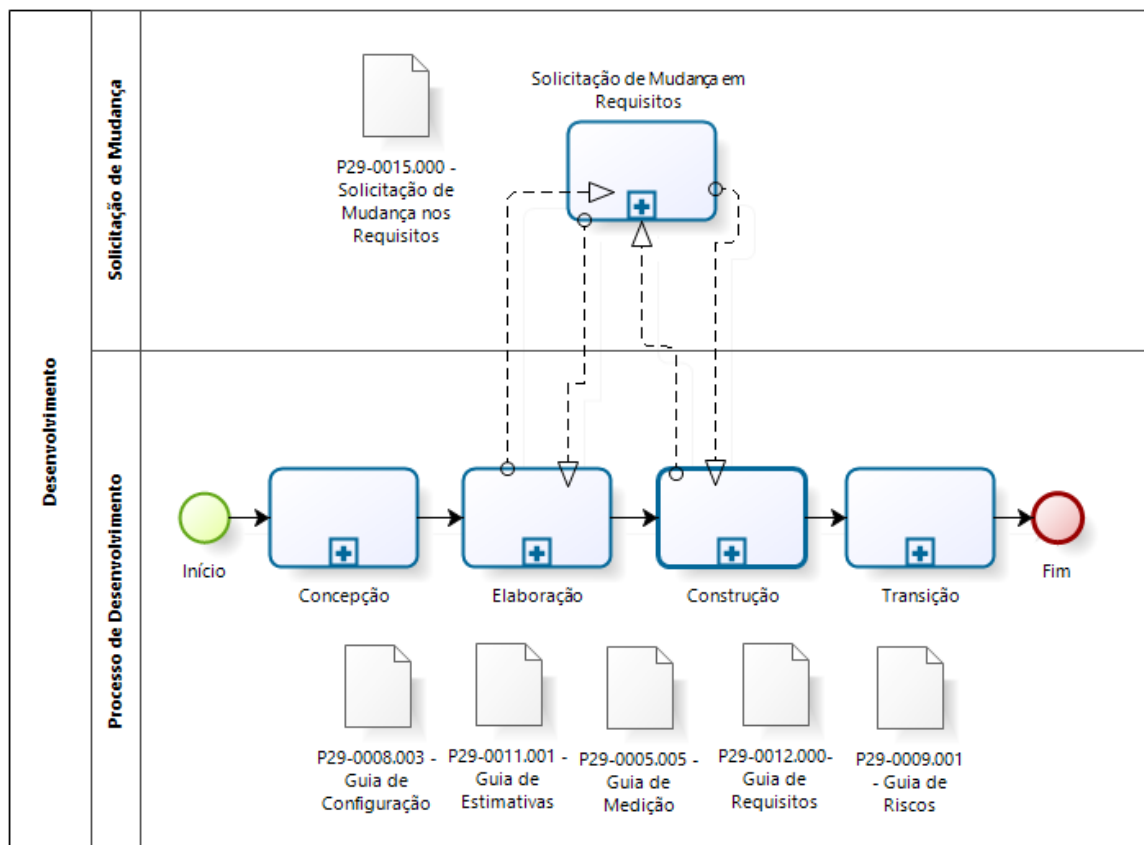


Figura 4 - Processo 4L versão 2.9
Fonte: P29-0008.003 – Guia de Configuração - Forlogic Software

A Figura 4 apresenta o fluxo de execução do projeto: concepção, elaboração, construção e transição; E um processo agregado para solicitações de mudança em requisitos durante o desenvolvimento. Abaixo do diagrama existem alguns guias que estão disponíveis para os colaboradores utilizarem durante todo e qualquer fase do processo de desenvolvimento, nesses guias são detalhados como as atividades devem ser realizadas, por quem e quais são os artefatos que devem ser gerados.

Em cada fase do processo de desenvolvimento existem diversas atividades que devem ser realizadas pela a equipe e pelo líder do projeto, com algumas restrições, serem sequenciais e uma fase só pode ser iniciada após a conclusão da anterior, a não ser a etapa de solicitação de mudança que pode ser iniciada a qualquer momento conforme surgir uma mudança no escopo do projeto durante a

construção. Apresentamos as descrições de cada fase do processo de desenvolvimento, com explicação de cada atividade necessária.

2.3.1 CONCEPÇÃO

A fase de concepção abrange as atividades de comunicação com o fornecedor de requisitos e planejamento do projeto. Contendo atividades como: levantamento de requisitos, definição do escopo negativo, identificação dos riscos, estimativa de tamanho, planejamento de recursos e prazos, análise de viabilidade, entre outros, apresentada na Figura 5.

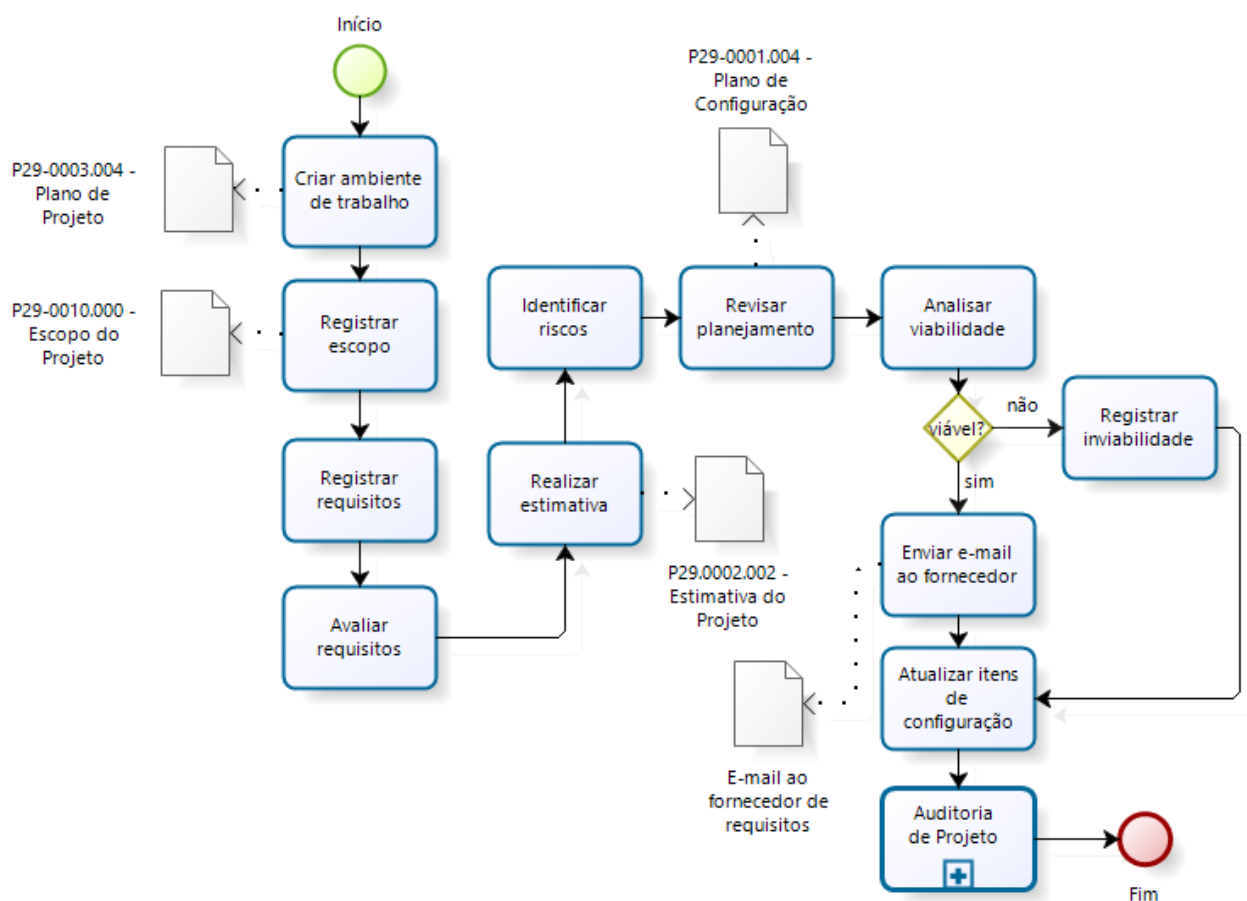


Figura 5 - Concepção
Fonte: P29-0008.003 – Guia de Configuração - Forlogic Software

Nesta etapa há um *checklist* com as atividades que devem ser executadas, durante esta fase do processo de desenvolvimento, conforme a Tabela 1.

Tabela 1 - Concepção

Concepção	01. Criar ambiente de trabalho (cat. Projeto)
	02. Registrar escopo (positivo e negativo) do projeto (cat. Projeto)
	03. Registrar requisitos (cat. Projeto)
	04. Avaliar requisitos (cat. Processo de Desenvolvimento)
	05. Realizar estimativa (cat. Projeto)
	06. Identificar riscos do projeto (cat. Projeto)
	07. Revisar planejamento do projeto (Horas, cronograma, divulgação e recursos) (cat. Projeto)
	08. Analisar viabilidade do projeto (cat. Processo de desenvolvimento)
	09. Enviar e-mail ao fornecedor de requisitos (cat. Processo de desenvolvimento)
	10. Atualizar itens de configuração (cat. Processo de desenvolvimento)
	11. Auditoria do Projeto (cat. Processo de Desenvolvimento)
	Melhoria do ambiente de trabalho (cat. Processo da Empresa)

Fonte: Autoria própria

2.3.2 ELABORAÇÃO

Esta fase está representada na Figura 6 e tem por objetivo criar a estrutura necessária e comunicar a equipe do início do projeto.

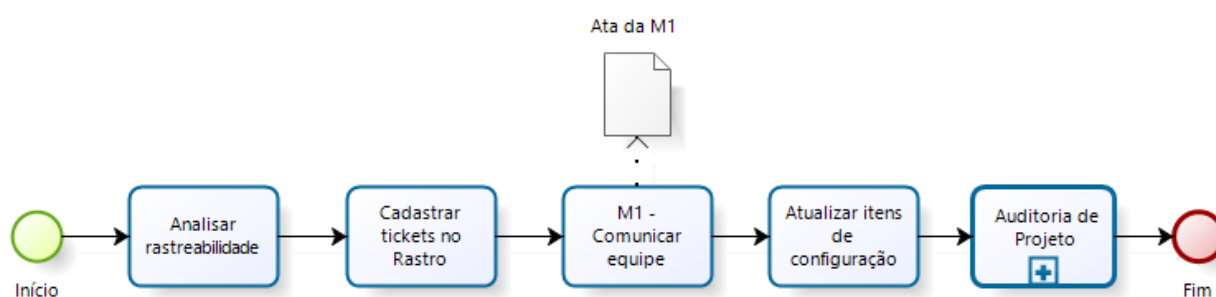


Figura 6 - Elaboração

Fonte: P29-0008.003 – Guia de Configuração - Forlogic Software

As atividades que devem ser executadas durante esta fase do processo de desenvolvimento são apresentadas na **Erro! Fonte de referência não encontrada.**, de elaboração.

Tabela 2 - Elaboração

Elaboração	12. Analisar rastreabilidade (cat. Processo de Desenvolvimento)
	13. Cadastrar <i>tickets</i> no Rastro (cat. Projeto)
	14. M1 – Comunicar equipe de projeto - 1º marco (cat. Projeto).
	15. Atualizar itens de configuração (cat. Processo de Desenvolvimento)
	16. Auditoria de Projeto (cat. Processo de Desenvolvimento)
	Melhoria do ambiente de trabalho (cat. Processo da Empresa)

Fonte: Autoria própria

2.3.3 CONSTRUÇÃO

Esta fase tem por objetivo desenvolver o projeto, transformando os requisitos especificados no escopo em código-fonte, representada na Figura 7.

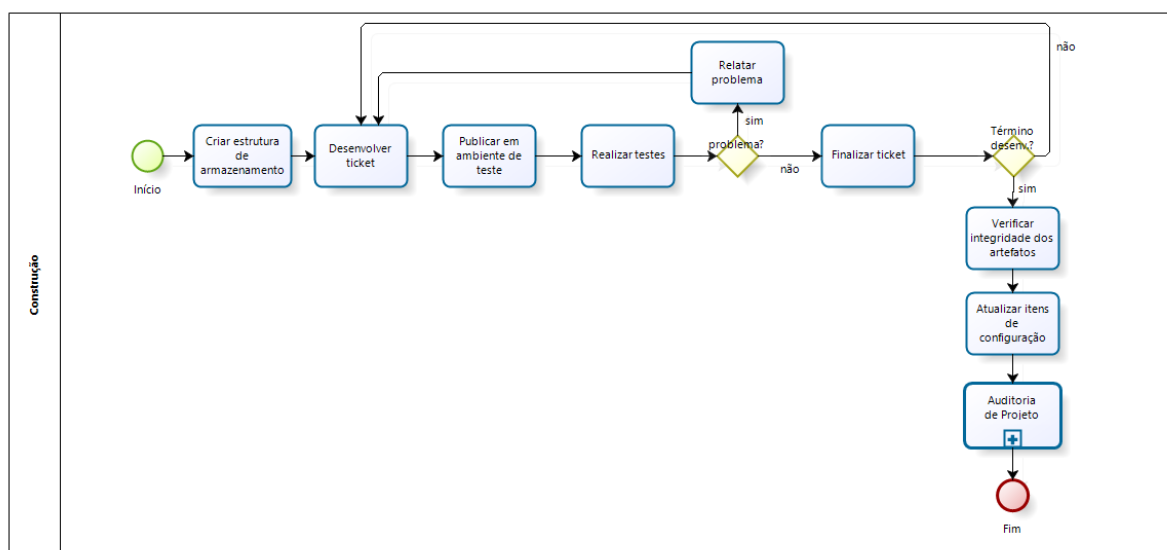


Figura 7 - Construção

Fonte: P29-0008.003 – Guia de Configuração - Forlogic Software

As atividades da **Erro! Fonte de referência não encontrada.** devem ser executadas durante esta fase do processo de desenvolvimento, são elas:

Tabela 3 - Construção

Construção	17. Criar estrutura de armazenamento e diagramas (cat. Projeto)
	18. Reuniões e atividades diárias (cat. Projeto)
	19. Monitoria / Marco Intermediário (cat. Projeto)
	20. Monitoria dos Itens de Configuração (cat. Projeto)
	21. Verificar integridade dos artefatos (cat. Processo de Desenvolvimento)
	22. Planos de Ação (cat. Processo de Desenvolvimento)
	23. Atualizar itens de configuração (cat. Processo de Desenvolvimento)
	24. Auditoria de Projeto (cat. Processo de Desenvolvimento)
Melhoria do ambiente de trabalho (cat. Processo da Empresa)	

Fonte: Autoria própria

2.3.4 TRANSIÇÃO

Seu objetivo é entregar o projeto ao cliente e realizar uma retrospectiva com a equipe, verificando como foi o andamento do projeto e identificar as melhorias para os próximos, apresentada na Figura 8.

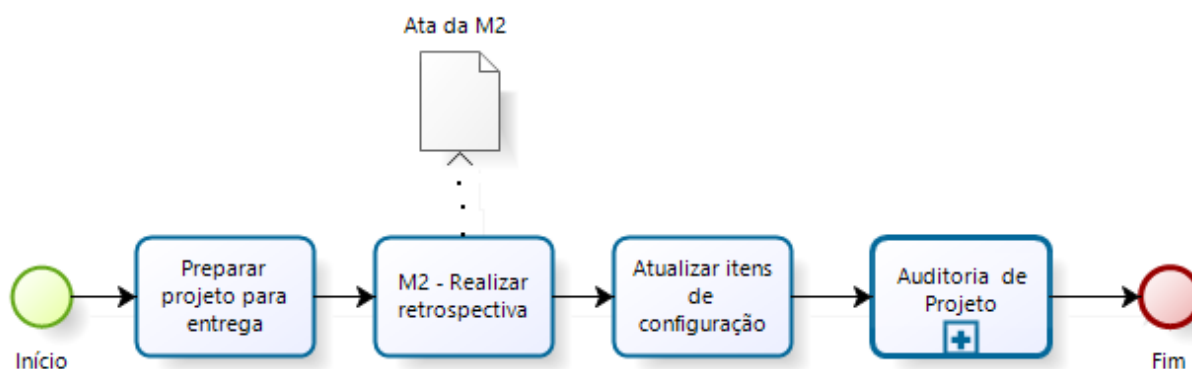


Figura 8 - Transição

Fonte: P29-0008.003 – Guia de Configuração - Forlogic Software

Nesta fase do processo de desenvolvimento as atividades apresentadas na **Erro! Fonte de referência não encontrada.**, devem ser executadas.

Tabela 4 - Transição

Transição	25. Disponibilizar projeto (cat. Projeto).
	26. M2 – Realizar retrospectiva da equipe – 2º marco (cat. Processo de Desenvolvimento).
	27. Atualizar itens de configuração (cat. Processo de Desenvolvimento)
	28. Auditoria de Projeto (cat. Processo de Desenvolvimento)
	Melhoria do ambiente de trabalho (cat. Processo da Empresa)

Fonte: Autoria própria

2.3.5 SOLICITAÇÃO DE MUDANÇA DE REQUISITOS

O processo de solicitação de mudança é utilizado para registrar os pedidos de mudança feitos ao líder de projeto. Sendo necessário, obter uma aprovação formal ou registro de parecer para que a solicitação seja desenvolvida, apresentada na Figura 9. O processo somente será executado nos seguintes casos:

- Inclusão de requisitos;
- Remoção de requisitos;
- Alteração do funcionamento do requisito, como: mudança em seu fluxo interno ou inclusão de nova funcionalidade no protótipo (que não existe classe ou tabela).

Para os demais casos, não há necessidade de executar o processo.

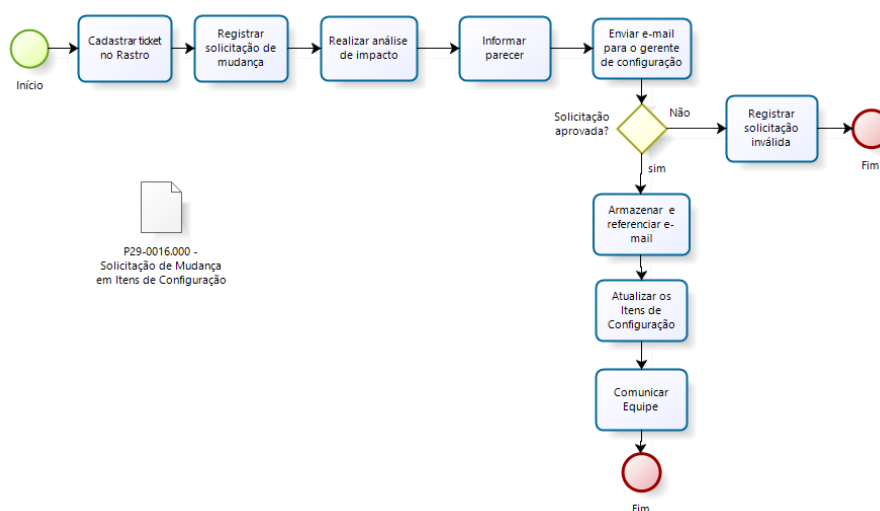


Figura 9 - Solicitação de Mudança

Fonte: P29-0008.003 – Guia de Configuração - Forlogic Software

A Figura 9 não tem relação direta com o objetivo geral do projeto que é controlar as solicitações de manutenção dos projetos.

Tabela 5 - Solicitação de Mudança

Solicitação de Mudança	29. Analisar requisitos.
	30. Preencher documento de solicitação de mudança.
	31. Avaliar o impacto nos requisitos.
	32. Avaliar impacto na execução.
	33. Enviar e-mail ao fornecedor de requisitos.
	34. Atualizar documentação do projeto.
	35. Comunicar equipe sobre mudança.
	36. Verificar integridade dos artefatos
	37. Melhoria do ambiente de trabalho (cat. Processo da Empresa).

Fonte: Autoria própria

A fase de Solicitação de Mudança apresentada por último, não tem a obrigatoriedade de ser executada durante o projeto a não ser que alguma mudança no escopo do projeto aconteça. Essa fase é destinada para o registro e análise de impacto para quando uma mudança de escopo deve ocorrer (porém, o projeto já está em andamento), seja ela por uma decisão do cliente ou por necessidade caso tenha ocorrido alguma falha no levantamento do projeto.

3. TECNOLOGIAS UTILIZADAS

Como a aplicação pertence à Forlogic as tecnologias que devem ser utilizadas já estão definidas, são elas:

- *AJAX – Asynchronous Javascript and XML*

O nome *AJAX* é uma sigla para “*Asynchronous Javascript and XML*”. Trata-se de uma maneira de fazer com que seu navegador, com *Javascript*, carregue conteúdo do servidor sem recarregar a página atual (GARRETT, 2005).

AJAX é carregar e renderizar uma página, utilizando recursos de scripts rodando pelo lado cliente, buscando e carregando dados em *background* sem a necessidade de *reload* da página. *AJAX* não é uma tecnologia, mas sim um conjunto de tecnologias. O conceito de *AJAX* se resume em conhecer bem *JavaScript*, trabalhando com *DOM (Document Object Model)*, *CSS (Cascading Style Sheets)* e *XML*.

- *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* e fornecer uma alternativa ao modelo *WebForm* do *ASP .NET* disponível até então. O *framework ASP.NET MVC* fornece um ambiente robusto e leve que está integrado aos recursos do *ASP .NET* (MACORATTI, 2009).

A utilização deste considera dois motivos, a tecnologia ser nova no mercado, quanto à possibilidade de utilizado com um *framework* Microsoft, e pelas facilidades que são proporcionadas para o desenvolvimento, desde a criação de novas funcionalidades, como a manutenção dos projetos. Esse padrão possui sua estrutura bem separada, seguindo padrões da orientação a objetos.

O modelo consiste em separar os três principais atores que formam o *software* e no *MVC* as camadas são bem definidas, e para alguma eventual manutenção, ser mais fácil a localização do problema e conseqüentemente os locais que devem ser aplicadas as devidas soluções.

Segundo Mendes (2002), o *MVC* divide o sistema em três camadas Modelo (*Model*), Visão (*View*), Controlador (*Controller*).

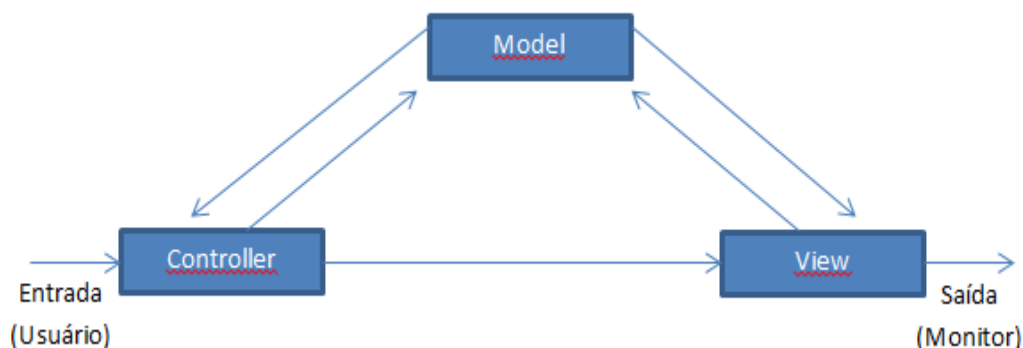


Figura 10 - Padrão MVC
Fonte: Mendes (2002)

Mendes (2002) define *Model*, *View* e *Controller* como um agente e cada agente possuem um estado e conhecimento, podendo iniciar e reagir a eventos. Seguindo este mesmo raciocínio, abaixo apresentamos um resumo da função de cada um dos agentes.

Model: 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*.

View: 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.

Controller: Responsável por controlar todo o fluxo de informação que passa pelo sistema. É este agente 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.

- *CSS – Cascading Style Sheets*

O nome *CSS* é uma sigla para “*Cascading Style Sheets*” que são folhas de estilos para páginas *Web*, adicionadas ou não ao código *HTML*. Fornece aos projetistas e usuários um maior controle sobre como as páginas *Web* são exibidas (AMARAL, 2006). Com *CSS*, pode-se criar folhas de estilo em um arquivo separado do código.

- *Ext.Js 4.1*

Ext é um conjunto de bibliotecas *JavaScript* para criação de aplicações *web* com recurso *RIA (Rich Internet Application)*, foi criado por Jack Slocum em 2006. A versão 4 do *Extjs* foi lançada em 2012 com mais de 350 novos *APIs*, 50 novas classes, e conta com um novo pacote de dados que permite usar a arquitetura *MVC*. O *ExtJs 4* teve seu código totalmente reescrito pensando em melhorar o desempenho. Essa reescrita contribuiu para manter a compatibilidade da biblioteca com a grande gama de navegadores disponíveis atualmente no mercado como Internet Explorer 6+, Firefox 1.5+ (PC/MAC), Safari 3+ e Opera 9+ (PC/MAC).

- *Firebug 1.12*

É um *plug-in open source* que integrada com o *Browser Firefox* fornece ao desenvolvedor uma suíte de ferramentas que seja possível de uma forma simples e rápida que permite a edição, depuração e monitoramento de códigos de páginas *Web nas linguagens HTML, JS e CSS, por exemplo*.

- *JAVASCRIPT*

É uma linguagem de programação interpretada e desenvolvida para atender algumas necessidades da aplicação, tais como: validação de formulários no lado do cliente, cálculos complexos, manipulação dos dados na página entre outros, tornando as páginas interativas e dinâmicas (FLANAGAN, 2006). Sua união com o *CSS* é conhecida como *DHTML*. Usando o *Javascript*, é possível modificar dinamicamente os estilos dos elementos da página em *HTML*.

- *SQL Server 2012*

O *SQL Server* é um servidor de banco de dados da *Microsoft*, que disponibiliza uma ferramenta para facilitar o seu gerenciamento chamada *SQL Server Management Studio*, que fornece uma interface mais facilitada de ser utilizada.

- *TortoiseSVN*

Também conhecido como *Subversion*, é um sistema de controle de versões. Será utilizado para controlar as versões de codificação, possibilitando a rastreabilidade de código-fonte através dos *logs de commits e updates* no repositório.

- *Visual Studio 2012*

O *Visual Studio 2012* é uma suíte de *softwares* disponibilizada pela *Microsoft* em diversas versões, voltada para o desenvolvimento de *software* com foco nas tecnologias pertencentes à própria *Microsoft* (*C#, VB, WCF*), com diversos recursos para o aumento da produtividade do desenvolvedor.

4. MODELAGEM

A utilização de diagramas transmite ao desenvolvedor uma visão macro e até mesmo específica de especificações do sistema que são de suma importância para o desenvolvimento, a boa utilização e estruturação dos mesmos, como o do caso de uso por exemplo permite apresentar os atores e funcionalidades essenciais (BOOCH, et al., 1998).

Os diagramas básicos que foram gerados durante a fase de concepção do projeto para auxiliar no entendimento da regra de negócio e análise do projeto, são representados a seguir.

1. Diagramas de caso de uso

Na utilização do sistema terão diversos usuários, porém, nos diagramas foram representados apenas quatro destes dividindo nas duas frentes do sistema, cliente e ForLogic.

- Cliente (Gerente/Gestor): Contato externo à empresa que terá acesso ao sistema, podendo fazer a abertura de novos chamados e avaliar os trabalhos que forem realizados conforme solicitações enviadas.
- ForLogic (Suporte e Desenvolvedor): Colaborador da ForLogic que realiza o atendimento e acompanhamento básico ao cliente, podendo realizar todas funcionalidades do sistema a não ser a aprovação de reabertura.
- ForLogic (Coordenador de suporte): Colaborador responsável por gerenciar a equipe de suporte, garantindo o cumprimento das atividades no prazo, capaz de realizar todas as funcionalidades do sistema.

A Figura 11 representa o diagrama de caso de uso para cadastro de chamado pelo cliente.

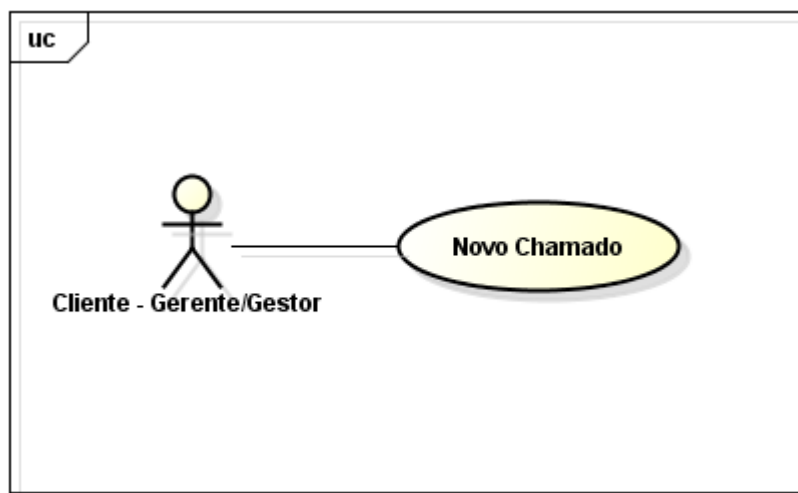


Figura 11 - Caso de Uso: Novo Chamado (Cliente)
Fonte: Autoria própria

A Figura 12 representa o diagrama de caso de uso para cadastro de chamado pelos colaboradores da ForLogic.

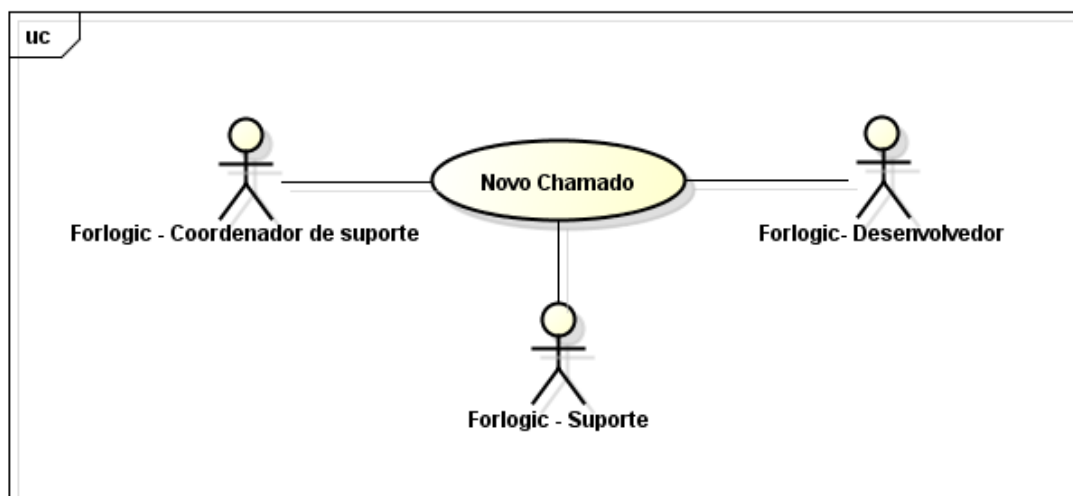


Figura 12 - Caso de Uso: Novo Chamado (ForLogic)
Fonte: Autoria própria

A Figura 13 representa o diagrama de caso de uso para avaliação do trabalho realizado pela ForLogic.

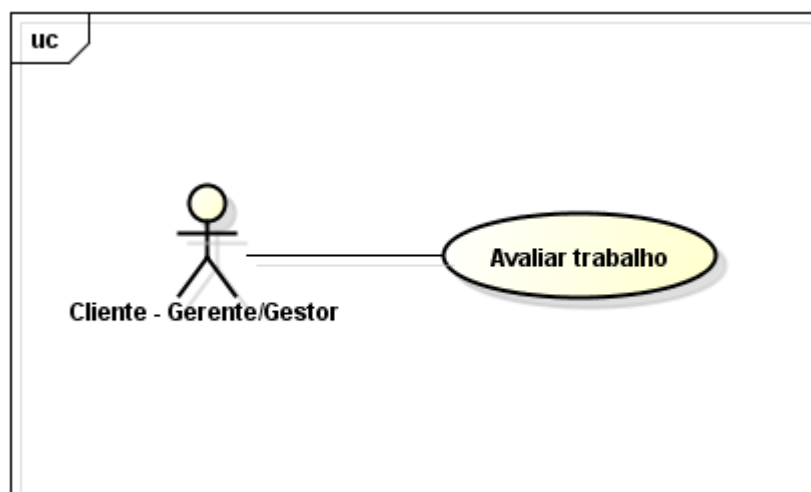


Figura 13 - Caso de Uso: Avaliar Trabalho
Fonte: Autoria própria

A Figura 14 representa os casos de uso de comum acesso entre os colaboradores da ForLogic.

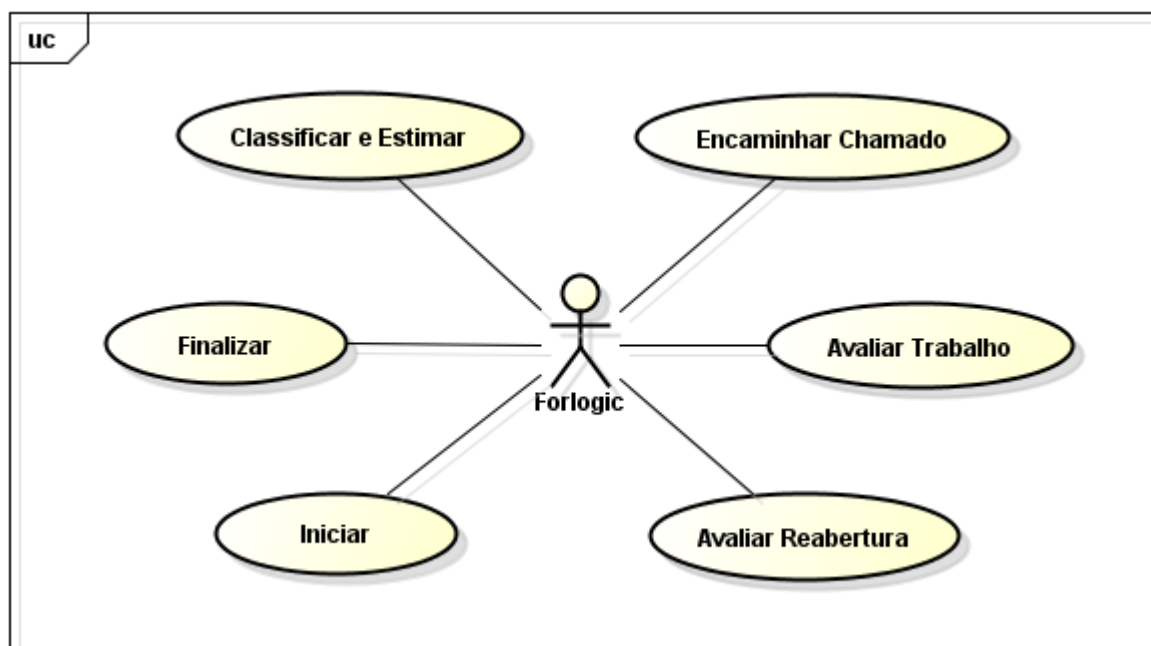


Figura 14 - Caso de Uso: Funcionalidades comuns da ForLogic
Fonte: Autoria própria

2. Diagrama de classe

A Figura 15 representa o diagrama de classe do sistema que foi desenvolvido através deste trabalho.

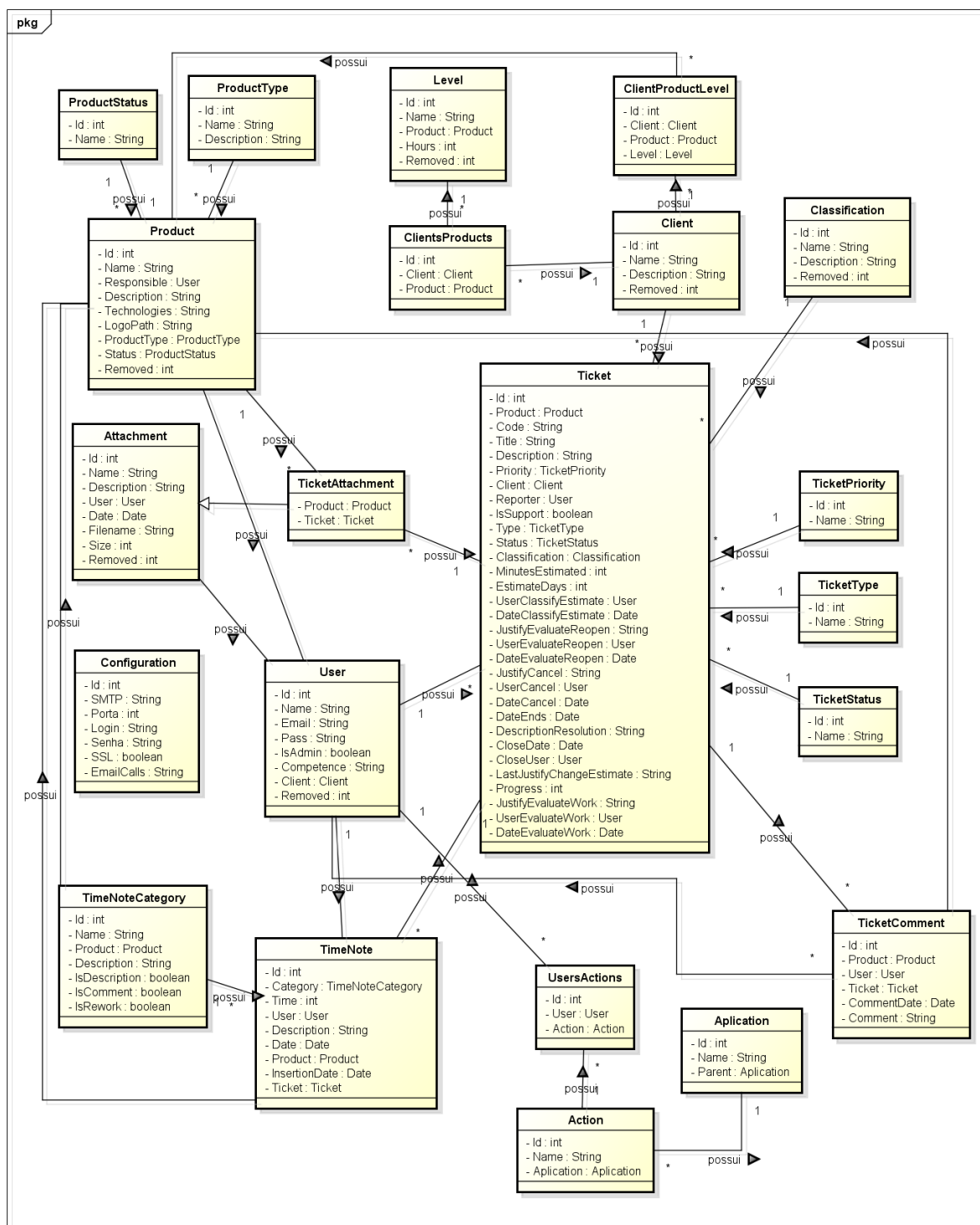


Figura 15 - Diagrama de classe
Fonte: Autoria própria

3. Diagrama de Atividade

A Figura 16 apresenta o fluxo de uma solicitação, em qualquer momento do fluxo é possível realizar o cancelamento da solicitação, realizar o anexo de algum arquivo e fazer um comentário. Todas as operações são salvas no histórico da solicitação e podem ser vistas a qualquer momento, a partir do botão visualizar na barra superior da gerência.

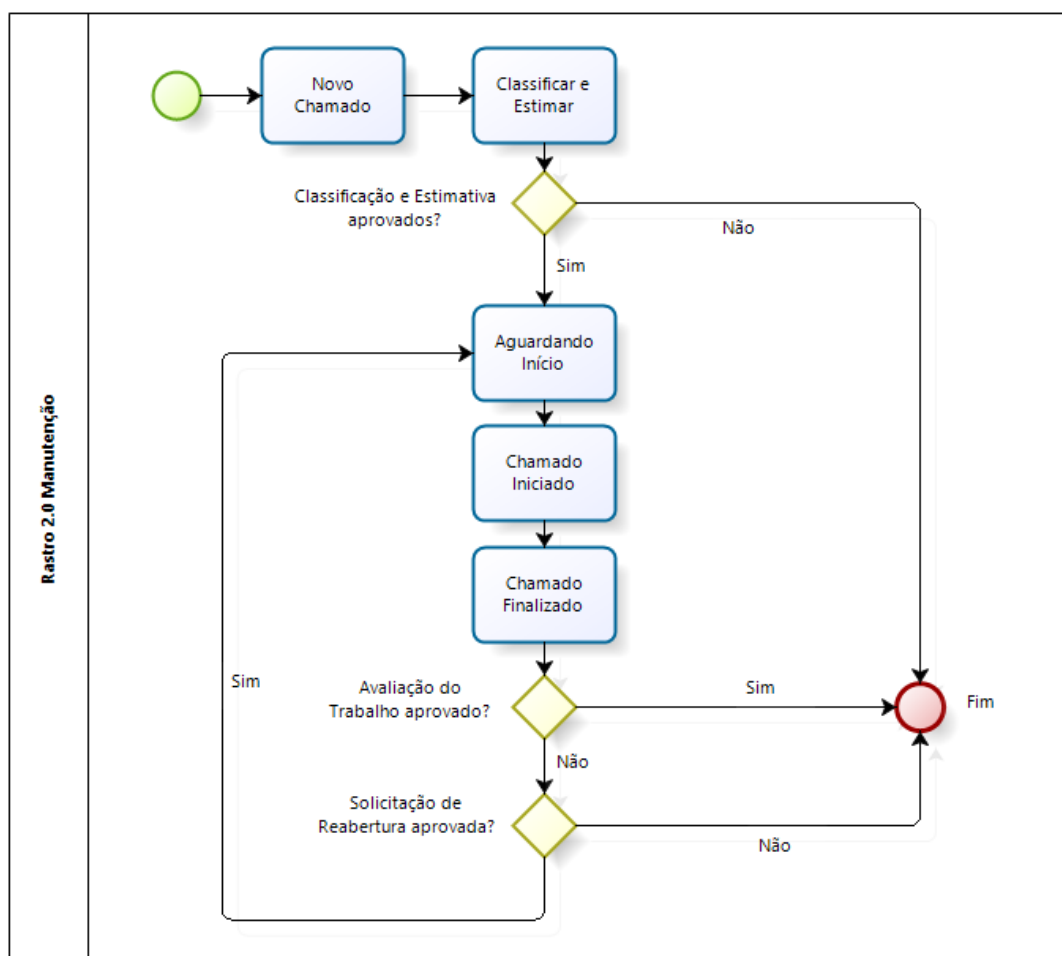


Figura 16 - Diagrama de Atividades
Fonte: Autoria própria

5. SISTEMA

Antes do desenvolvimento do sistema foi realizado o levantamento, visando uma ferramenta amigável de fácil utilização. Para ser utilizada tanto por pessoas que possuem conhecimento técnico na área de desenvolvimento de sistemas e pessoas leigas.

A seguir estão descritos cada um dos requisitos do sistema a partir de uma ilustração da tela da ação dentro do sistema.

5.1 CLASSIFICAÇÕES

A Figura 17 representa a gerência para controlar as classificações dos chamados, onde poderá ter diversos tipos cadastrados, como por exemplo: Erro, Melhoria, etc.

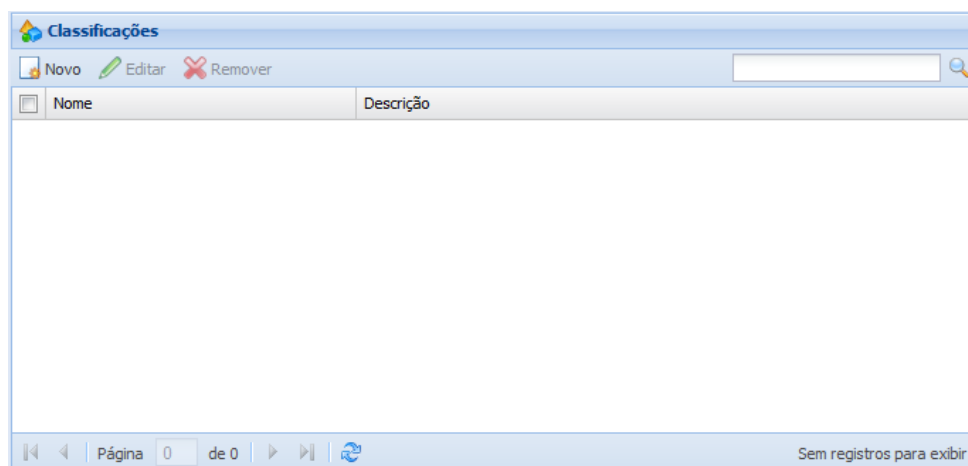


Figura 17 - Gerência de Classificações
Fonte: Autoria própria

Na Figura 18 está a janela utilizada para cadastrar e/ou editar as classificações.

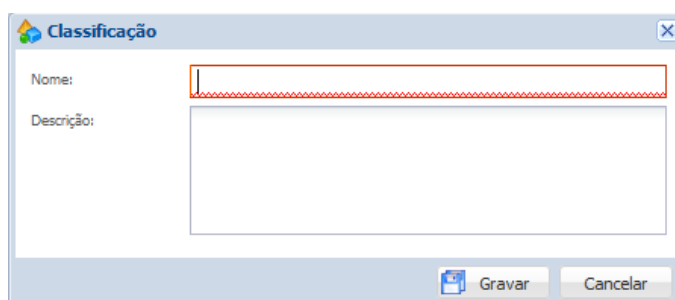
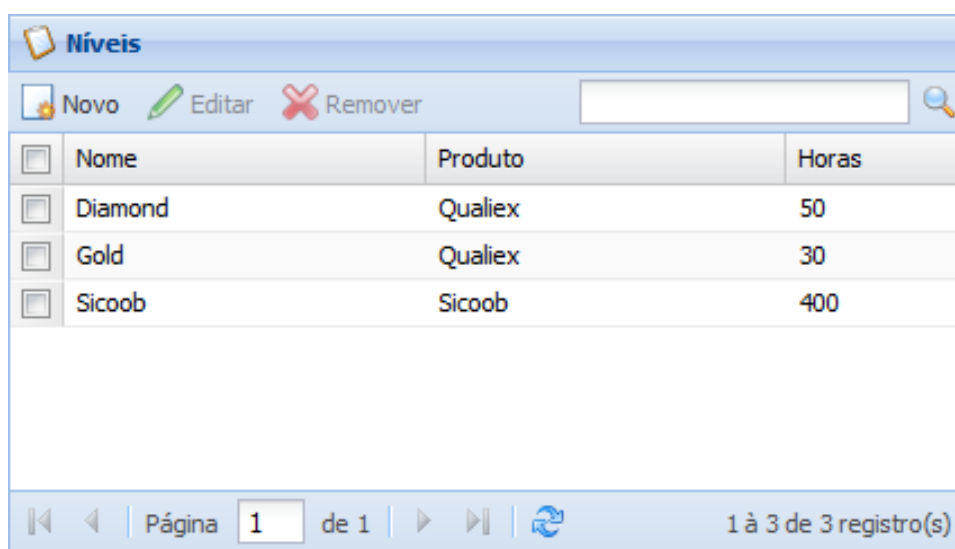


Figura 18 - Cadastro/Edição de Classificações
Fonte: Autoria própria

5.2 NÍVEIS

A Figura 19 representa a gerência, utilizada para controlar os níveis de suporte oferecidos pela Forlogic para cada um dos produtos. Por exemplo, a ForLogic oferece 4 níveis de suporte para o produto Qualiex (Ouro, Prata, Bronze), o nível Ouro para o Qualiex possui 20 horas de suporte, o nível Prata para o Qualiex 10 horas e o nível Bronze para o Qualiex possui 5 horas de suporte ao mês. Desta forma é possível que outro produto também tenha um pacote de suporte com o nome Ouro, Prata e Bronze.



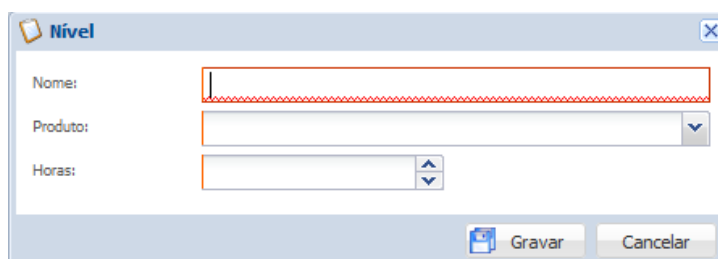
<input type="checkbox"/>	Nome	Produto	Horas
<input type="checkbox"/>	Diamond	Qualiex	50
<input type="checkbox"/>	Gold	Qualiex	30
<input type="checkbox"/>	Sicoob	Sicoob	400

Novo Editar Remover

Página 1 de 1 1 à 3 de 3 registro(s)

Figura 19 - Gerência de Níveis
Fonte: Autoria própria

Na Figura 20 está a janela utilizada para cadastrar e/ou editar os níveis.



Nível

Nome:

Produto:

Horas:

Gravar Cancelar

Figura 20 - Cadastro/Edição de Níveis
Fonte: Autoria própria

5.3 CONTATOS

A Figura 21 representa a gerência para controlar os contatos que recebem e interagem com o suporte oferecido pela Forlogic.

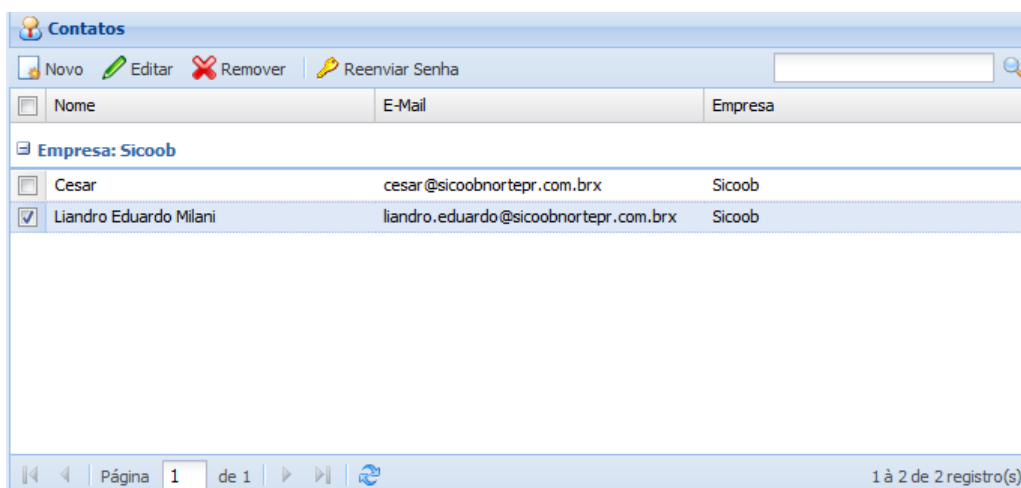


Figura 21 - Gerência de Contatos
Fonte: Autoria própria

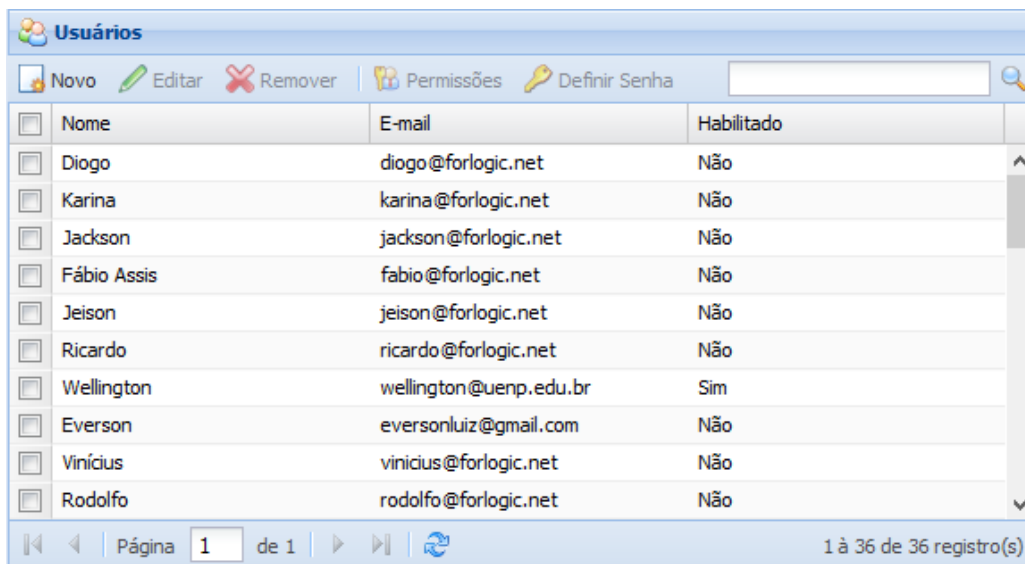
A Figura 22 é a janela que é utilizada para cadastrar e/ou editar os contatos.

A imagem mostra uma janela de diálogo intitulada 'Contato'. Ela contém três campos de entrada: 'Nome', 'E-Mail' e 'Empresa'. O campo 'Nome' está atualmente em foco, indicado por uma borda vermelha. Abaixo dos campos, há dois botões: 'Gravar' e 'Cancelar'.

Figura 22 - Cadastro/Edição de Contato
Fonte: Autoria própria

5.4 USUÁRIOS

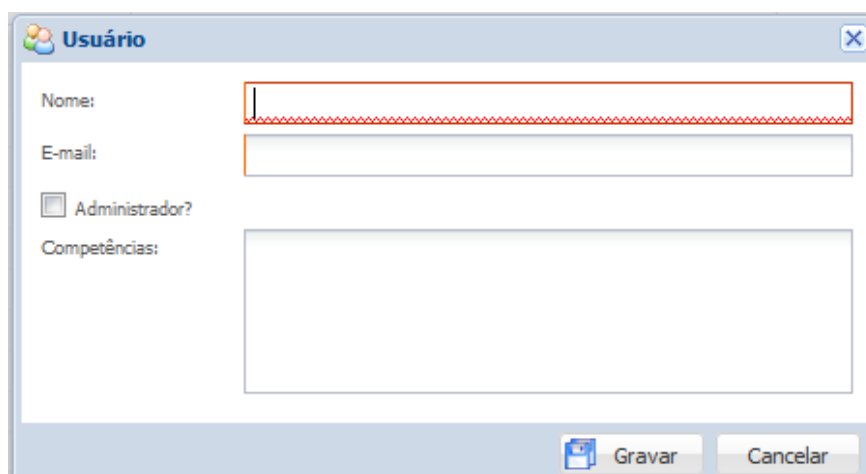
A Figura 23 representa a gerência para controlar os usuários que interagem com o sistema, porém, somente os usuários da ForLogic. A gerência dos usuários externos foi apresentada no item anterior.



<input type="checkbox"/>	Nome	E-mail	Habilitado
<input type="checkbox"/>	Diogo	diogo@forlogic.net	Não
<input type="checkbox"/>	Karina	karina@forlogic.net	Não
<input type="checkbox"/>	Jackson	jackson@forlogic.net	Não
<input type="checkbox"/>	Fábio Assis	fabio@forlogic.net	Não
<input type="checkbox"/>	Jeison	jeison@forlogic.net	Não
<input type="checkbox"/>	Ricardo	ricardo@forlogic.net	Não
<input type="checkbox"/>	Wellington	wellington@uenp.edu.br	Sim
<input type="checkbox"/>	Everson	eversonluiz@gmail.com	Não
<input type="checkbox"/>	Vinicius	vinicius@forlogic.net	Não
<input type="checkbox"/>	Rodolfo	rodolfo@forlogic.net	Não

Figura 23 - Gerência de Usuários
Fonte: Autoria própria

Abaixo na Figura 24 está a janela utilizada para cadastrar e/ou editar os usuários.



Usuário

Nome:

E-mail:

Administrador?

Competências:

Gravar Cancelar

Figura 24 - Cadastro/Edição de Usuários
Fonte: Autoria própria

5.5 TIPOS DE PRODUTOS

A

Figura 25 representa o controle dos tipos de produtos que a ForLogic possui, serve para melhorar a organização dos dados do sistema, permite gerar relatórios separados, por tipos que serão cadastrados.

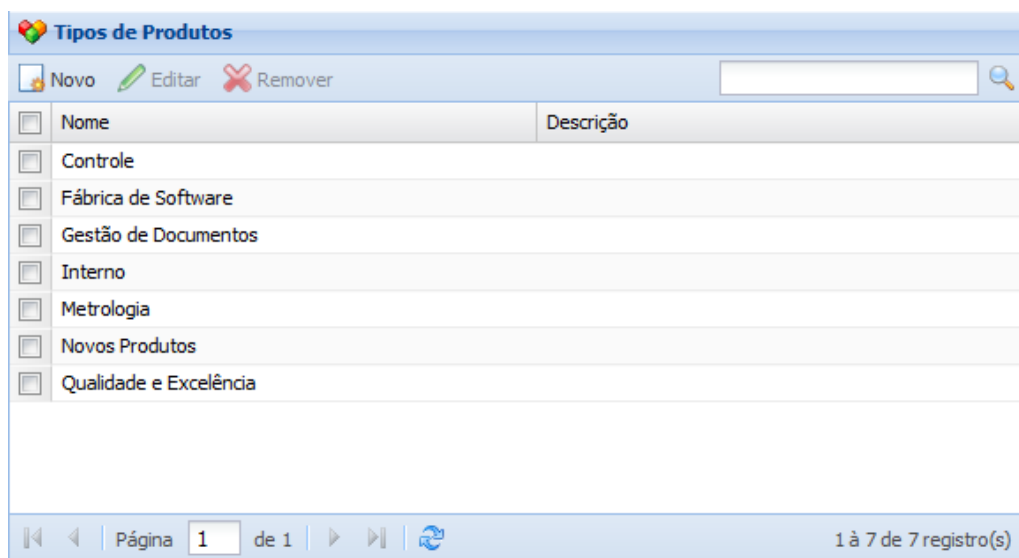


Figura 25 - Gerência de Tipos de Produtos
Fonte: Autoria própria

Na Figura 26 está a janela utilizada para cadastrar e/ou editar os tipos de produtos.

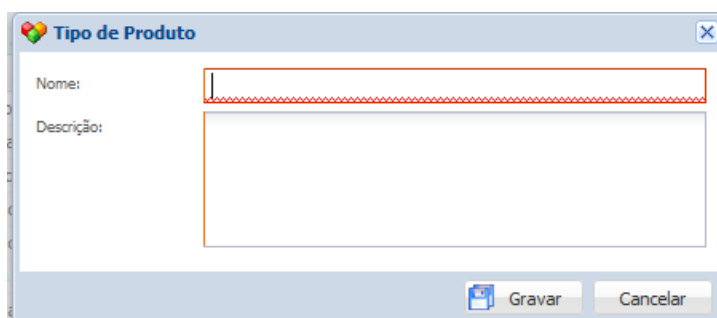
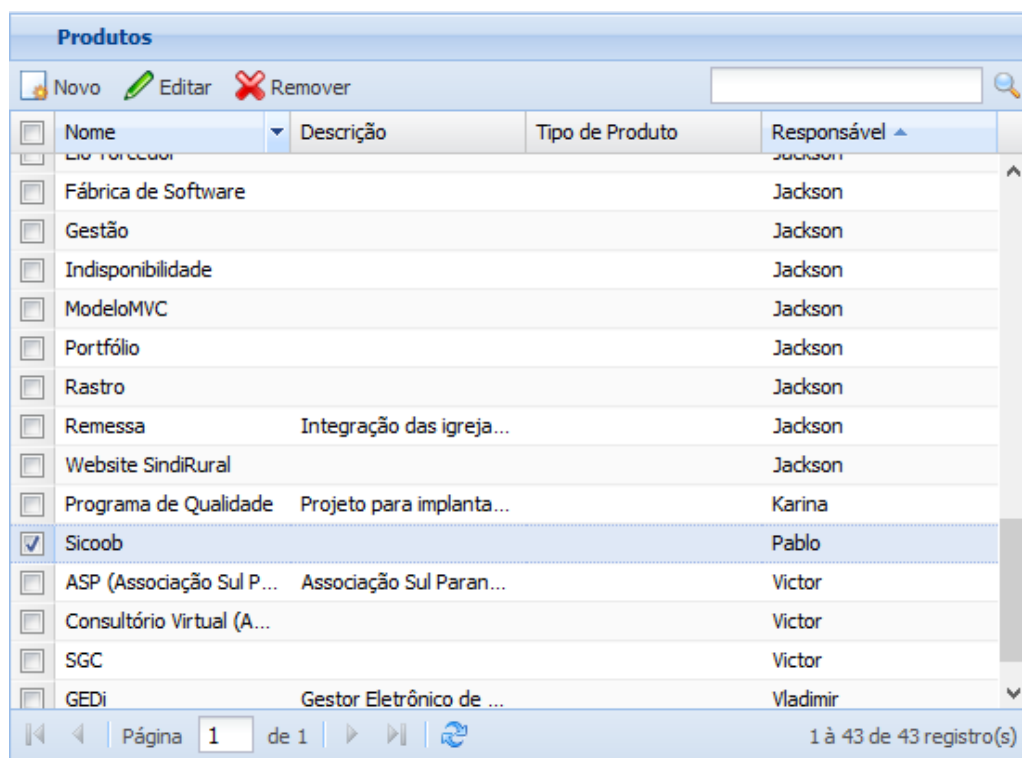


Figura 26 - Cadastro/Edição de Tipos de Produtos
Fonte: Autoria própria

5.6 PRODUTOS

A Figura 27 representa a gerência para controlar os produtos que a ForLogic possui, incluindo dados adicionais referentes a esses produtos.

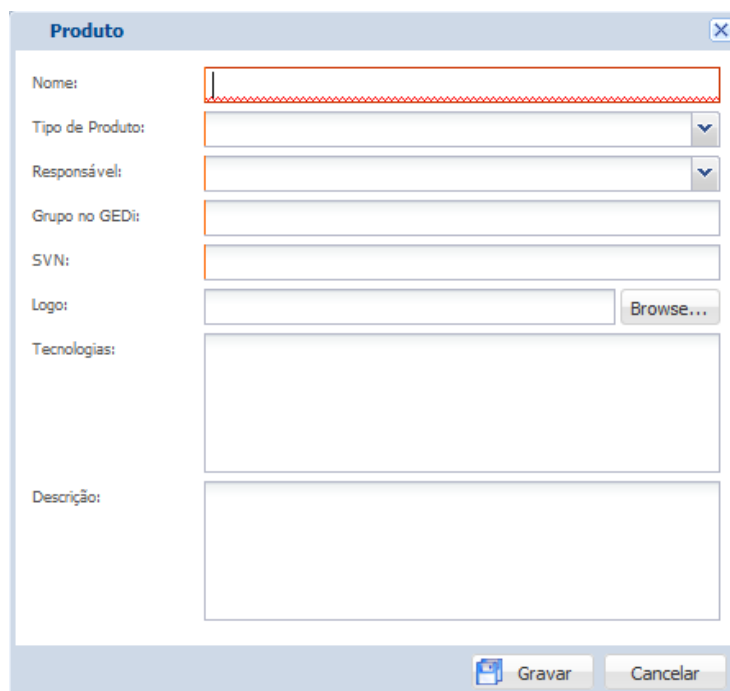


<input type="checkbox"/>	Nome	Descrição	Tipo de Produto	Responsável
<input type="checkbox"/>	...			Jackson
<input type="checkbox"/>	Fábrica de Software			Jackson
<input type="checkbox"/>	Gestão			Jackson
<input type="checkbox"/>	Indisponibilidade			Jackson
<input type="checkbox"/>	ModeloMVC			Jackson
<input type="checkbox"/>	Portfólio			Jackson
<input type="checkbox"/>	Rastro			Jackson
<input type="checkbox"/>	Remessa	Integração das igreja...		Jackson
<input type="checkbox"/>	Website SindiRural			Jackson
<input type="checkbox"/>	Programa de Qualidade	Projeto para implanta...		Karina
<input checked="" type="checkbox"/>	Sicoob			Pablo
<input type="checkbox"/>	ASP (Associação Sul P...	Associação Sul Paran...		Victor
<input type="checkbox"/>	Consultório Virtual (A...			Victor
<input type="checkbox"/>	SGC			Victor
<input type="checkbox"/>	GEDI	Gestor Eletrônico de ...		Vladimir

1 à 43 de 43 registro(s)

Figura 27 - Gerência de Produtos
Fonte: Autoria própria

Na Figura 28 está a tela utilizada para cadastrar e/ou editar os produtos.



The image shows a software window titled "Produto" with a close button in the top right corner. The form contains the following fields and controls:

- Nome:** A text input field with a red dashed border.
- Tipo de Produto:** A dropdown menu.
- Responsável:** A dropdown menu.
- Grupo no GEDi:** A text input field.
- SVN:** A text input field.
- Logo:** A text input field followed by a "Browse..." button.
- Tecnologias:** A large text area.
- Descrição:** A large text area.

At the bottom of the window, there are two buttons: "Gravar" (Save) and "Cancelar" (Cancel).

Figura 28 - Cadastro/Edição de Produtos
Fonte: Autoria própria

5.7 CHAMADOS (FORLOGIC)

Os dois próximos requisitos são escopo do trabalho, todos os itens descritos até então, servem para auxiliar no desenvolvimento do trabalho proposto.

A Figura 29 representa a gerência para controlar os chamados.

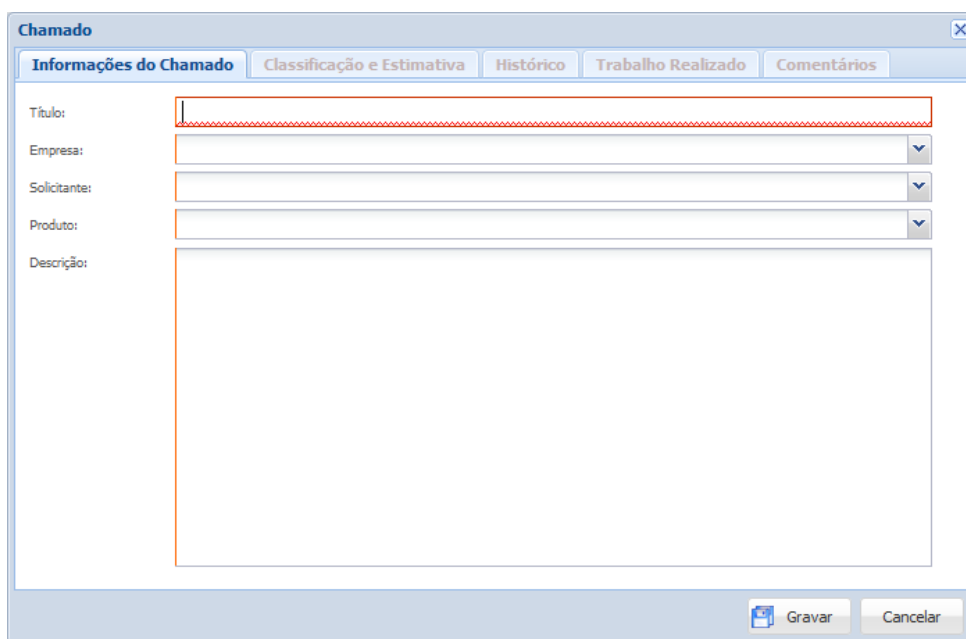
Chamados									
Novo Reestimar Alterar Status Mais opções Tempo Progresso Visualizar									
Ícones	Código	Cliente	Título	Contato	Data de Abertura	Previsão de Entrega	Iniciado por	Classificação	Prioridade
Status: Aguardando Avaliação de Reabertura (2 Itens)									
	T1509	Sicoob	Teste 3	Jackson	15/12/13 20:12	19/12/13	Jackson	Defeito	Normal
	T1510	Sicoob	Teste 4	Jackson	15/12/13 22:12			Melhoria	Normal
Status: Aguardando Classificação e Estimativa (1 Item)									
	T3765	Sicoob	Teste 5	Jackson	26/12/13 23:12				Normal
Status: Aguardando Início (1 Item)									
	T3763	Sicoob	Teste 1	Jackson	15/12/13 20:12	19/12/13	Jackson	Defeito	Normal
Status: Finalizado (1 Item)									
	T3764	Sicoob	Teste 2	Jackson	15/12/13 20:12	18/12/13	Jackson	Melhoria	Normal

Figura 29 - Gerência de Chamados (Forlogic)
Fonte: Autoria própria

Abaixo serão apresentados as funcionalidades que são realizadas à partir desta gerência.

5.7.1 NOVO

Na Figura 30 está à janela utilizada para cadastrar novos chamados. Uma interface para quando a solicitação é identificada pela própria equipe da ForLogic ou então quando o cliente informa a equipe de alguma maneira, pedindo o cadastro da solicitação, é necessário informar qual é o cliente, quem solicitou e qual produto que a solicitação pertence.



The image shows a software window titled "Chamado" with a close button in the top right corner. Below the title bar, there are five tabs: "Informações do Chamado", "Classificação e Estimativa", "Histórico", "Trabalho Realizado", and "Comentários". The "Informações do Chamado" tab is active and contains the following fields:

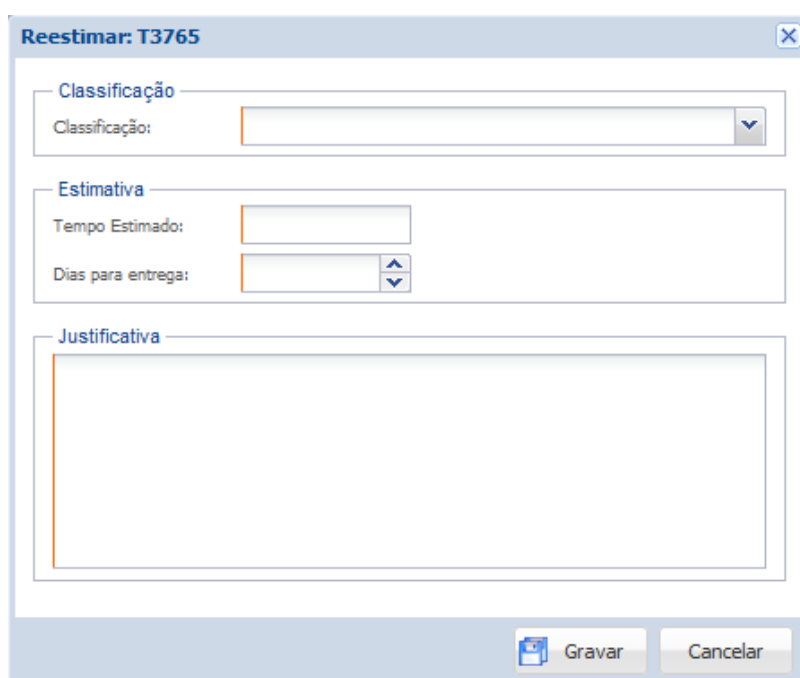
- Título: A text input field.
- Empresa: A dropdown menu.
- Solicitante: A dropdown menu.
- Produto: A dropdown menu.
- Descrição: A large text area.

At the bottom right of the window, there are two buttons: "Gravar" (Save) and "Cancelar" (Cancel).

Figura 30 - Cadastro de Chamado (Forlogic)
Fonte: Autoria própria

5.7.2 REESTIMAR

Esta funcionalidade serve para alterar a estimativa e classificação que foi informado no momento de realizar esta tarefa diante do fluxo das solicitações. Importa destacar que nesta etapa é obrigatória a informação de uma justificativa que será enviado para o cliente para que ele fique ciente da alteração. Para realizar esta ação é necessário que o usuário que esteja utilizando o sistema e tenha a devida permissão para isso. A Figura 31 representa como é a tela para realizar essa alteração.



A imagem mostra uma janela de software intitulada "Reestimar: T3765". A janela contém três seções principais:

- Classificação:** Um campo de texto rotulado "Classificação:" com uma seta para baixo no final, indicando uma lista suspensa.
- Estimativa:** Duas linhas de entrada. A primeira é rotulada "Tempo Estimado:" e a segunda "Dias para entrega:", esta última com setas para cima e para baixo, indicando um campo de spin.
- Justificativa:** Um grande campo de texto retangular para a inserção de texto.

Na base da janela, há dois botões: "Gravar" (acompanhado de um ícone de disco) e "Cancelar".

Figura 31 - Reestimar chamado
Fonte: Autoria própria

5.7.3 AVALIAR REABERTURA

Esta funcionalidade serve para avaliar a solicitação para reabrir o chamado por parte do cliente, isso ocorre quando o cliente avalia a realização do trabalho como incorreto e então a Forlogic tem a possibilidade de avaliar se realmente o que o cliente alegou procede ou não. Caso a informação que o cliente informou não seja verdadeira, a Forlogic reprova a reabertura e o chamado é finalizado, caso contrário o chamado volta para aguardando início. A Figura 32 ilustra a tela de avaliação de reabertura.



A imagem mostra uma janela de diálogo com o título "Avaliar reabertura: R000653". A janela contém dois campos de texto para a inserção de justificativas. O primeiro campo, rotulado "Justificativa para reabertura:", está atualmente vazio. O segundo campo, rotulado "Justificativa:", também está vazio e possui uma borda vermelha pontilhada, indicando que é o campo focado. Na base da janela, há três botões de ação: "Aprovar", "Reprovar" e "Cancelar".

Figura 32 - Avaliar reabertura do chamado
Fonte: Autoria própria

5.7.4 CLASSIFICAR E ESTIMAR

Esta funcionalidade apresentada na Figura 33 é destinada para que a equipe de suporte da ForLogic identifique a solicitação que o cliente encaminhou e após uma breve análise classifique-a se é alguma falha, melhoria, etc, e informe quanto tempo em horas e em dias serão necessários para que a tarefa seja atendida e disponibilizada para o cliente.

A janela de interface gráfica 'Classificar e Estimar: R000653' contém os seguintes campos e controles:

- Chamado:**
 - Título: Campo de texto para o título do chamado.
 - Descrição: Área de texto grande para a descrição do chamado.
- Classificação:**
 - Classificação: Menu suspenso para selecionar a categoria do chamado.
- Estimativa:**
 - Tempo Estimado: Campo de texto com uma seta para baixo, para inserir o tempo em horas.
 - Dias para entrega: Campo de texto com setas para cima e para baixo, para inserir o número de dias.

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

Figura 33 - Classificar e estimar chamado
Fonte: Autoria própria

5.7.5 CANCELAR

Esta funcionalidade apresentada na Figura 34 é destinada para que o chamado seja cancelado, qualquer uma das partes que interagem com o sistema pode realizar esta tarefa, e ambas são notificadas por e-mail quando essa ação ocorre, contendo a justificativa para o cancelamento.

A janela de interface gráfica 'Cancelar: R000653' contém o seguinte campo e controle:

- Justificativa:** Área de texto grande para fornecer a justificativa para o cancelamento do chamado.

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

Figura 34 - Cancelar chamado
Fonte: Autoria própria

5.7.6 INICIAR

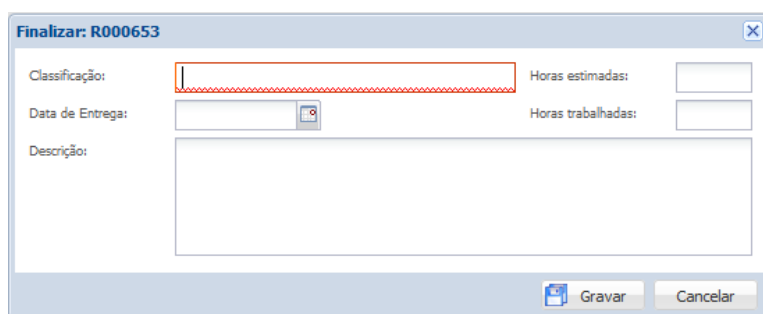
Utilizada para controle interno da Forlogic para saber quem é o responsável pela realização de cada um dos chamados.

5.7.7 ENCAMINHAR

Esta funcionalidade altera o responsável pela realização do chamado.

5.7.8 FINALIZAR

Realiza a finalização do trabalho com o chamado, onde indica para o cliente que o trabalho foi concluído e está pronto para que ele avalie se sua solicitação foi atendida conforme solicitado, apresentada na Figura 35.



A janela de interface gráfica intitulada "Finalizar: R000653" contém os seguintes elementos:

- Um campo de texto rotulado "Classificação:" com uma borda vermelha.
- Um campo de texto rotulado "Data de Entrega:" com um ícone de calendário.
- Um campo de texto rotulado "Descrição:".
- Dois campos de texto rotulados "Horas estimadas:" e "Horas trabalhadas:".
- Dois botões na base da janela: "Gravar" e "Cancelar".

Figura 35 - Finalizar chamado
Fonte: Autoria própria

5.7.9 ANEXOS

A funcionalidade apresentada na Figura 36 permite anexar qualquer tipo de arquivo que não exceda 10Mb para facilitar a equipe da Forlogic a identificar o problema, ou entender melhor a solicitação, qualquer uma das partes pode anexar um arquivo a qualquer momento, mesmo que o chamado já tenha sido finalizado.

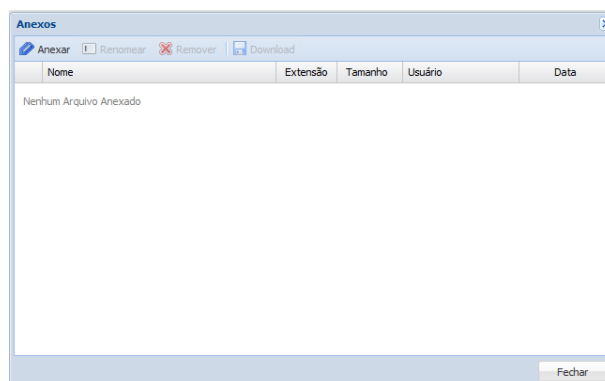


Figura 36 - Anexar arquivos ao chamado
Fonte: Autoria própria

5.7.10 COMENTAR

Esta funcionalidade apresentada na Figura 37 permite adicionar comentários ao chamado, fazendo com que fique registrado alguns acontecimentos importantes caso algum ator do sistema julgue importante.

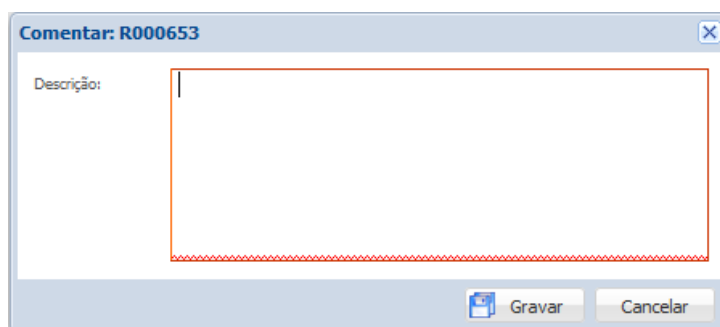
A imagem mostra uma janela de diálogo de software com o título "Comentar: R000653". O conteúdo principal da janela é um campo de texto grande e vazio, precedido pelo rótulo "Descrição:". Na base da janela, há dois botões: "Gravar" e "Cancelar".

Figura 37 - Comentar chamado
Fonte: Autoria própria

5.7.11 REENVIAR ESTIMATIVA PARA O CLIENTE

Esta funcionalidade permite que seja reenviado o e-mail de notificação de que o chamado foi classificado e estimado e que está aguardando a avaliação para que o fluxo seja continuado.

5.7.12 TEMPO

Esta funcionalidade apresentada na Figura 38 permite à equipe da Forlogic registrar o tempo em que trabalho nessa atividade, através dessa funcionalidade que é feito o controle das horas de suporte que já foram prestadas ao cliente para determinado produto que ele possui contratado.

Tempo: R000653

Dados

Categoria:

Tempo trabalhado:

Data:

Descrição

Comentário

Histórico

Categoria	Data	Tempo
-----------	------	-------

Gravar Cancelar

Figura 38 - Informar tempo trabalhado
Fonte: Autoria própria

5.7.13 PROGRESSO

A funcionalidade representada na Figura 39 permite à equipe da Forlogic, atualizar e informar o cliente o progresso que a atividade está tendo, decidiu-se informar o progresso dessa forma pois a realização automática pelo tempo que é informado na atividade pode não corresponder ao progresso real, já que uma atividade pode ter tido complicações e o tempo necessário para concluir seja maior que o previsto, então para casos assim o tempo gasto pode estar em 100 por cento e o progresso estagnado em 50 por cento.

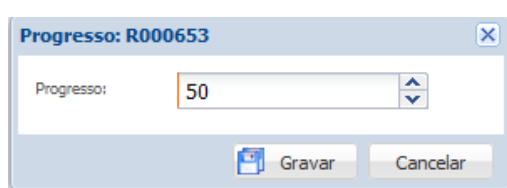


Figura 39 - Informar progresso do chamado
Fonte: Autoria própria

5.7.14 VISUALIZAR

A Figura 40 até a Figura 44, representam a funcionalidade que permite qualquer um dos atores do sistema acompanhar e visualizar todas as informações referentes ao chamado até o dado momento, contando com a aba de histórico que mostra o momento exato em que o usuário realizou determinada ação.

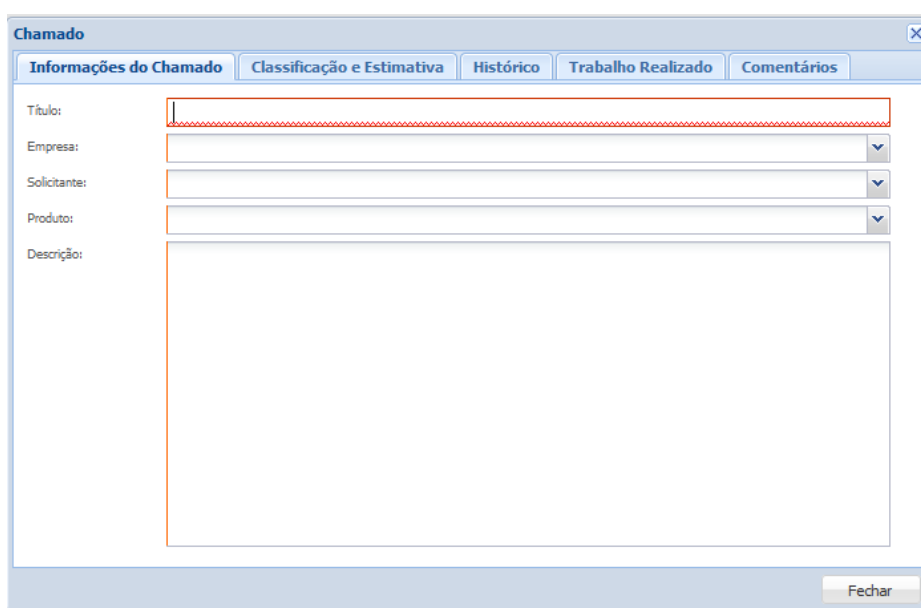


Figura 40 - Visualizar chamado (Aba: Informações do Chamado)
Fonte: Autoria própria

The screenshot shows the 'Chamado' application window with the 'Classificação e Estimativa' tab selected. The window has a title bar with a close button. Below the title bar are five tabs: 'Informações do Chamado', 'Classificação e Estimativa', 'Histórico', 'Trabalho Realizado', and 'Comentários'. The main content area is divided into two sections. The first section, 'Classificação', contains a label 'Classificação:' followed by a dropdown menu. The second section, 'Estimativa', contains two labels: 'Tempo Estimado:' followed by a dropdown menu, and 'Dias para entrega:' followed by a spinner control. A 'Fechar' button is located at the bottom right of the window.

Figura 41 - Visualizar chamado (Aba: Classificação e Estimativa)
Fonte: Autoria própria

The screenshot shows the 'Chamado' application window with the 'Histórico' tab selected. The window has a title bar with a close button. Below the title bar are five tabs: 'Informações do Chamado', 'Classificação e Estimativa', 'Histórico', 'Trabalho Realizado', and 'Comentários'. The main content area displays a table with three columns: 'Responsável', 'Ação', and 'Data'. The table is currently empty. A 'Fechar' button is located at the bottom right of the window.

Figura 42 - Visualizar chamado (Aba: Histórico)
Fonte: Autoria própria

The screenshot shows the 'Chamado' application window with the 'Trabalho Realizado' tab selected. The window has a title bar with a close button. Below the title bar are five tabs: 'Informações do Chamado', 'Classificação e Estimativa', 'Histórico', 'Trabalho Realizado', and 'Comentários'. The main content area contains several fields: 'Classificação:' with a dropdown menu, 'Data de Entrega:' with a date picker, 'Descrição:' with a text area, 'Horas estimadas:' with a text input field, and 'Horas trabalhadas:' with a text input field. A 'Fechar' button is located at the bottom right of the window.

Figura 43 - Visualizar chamado (Aba: Trabalho Realizado)
Fonte: Autoria própria

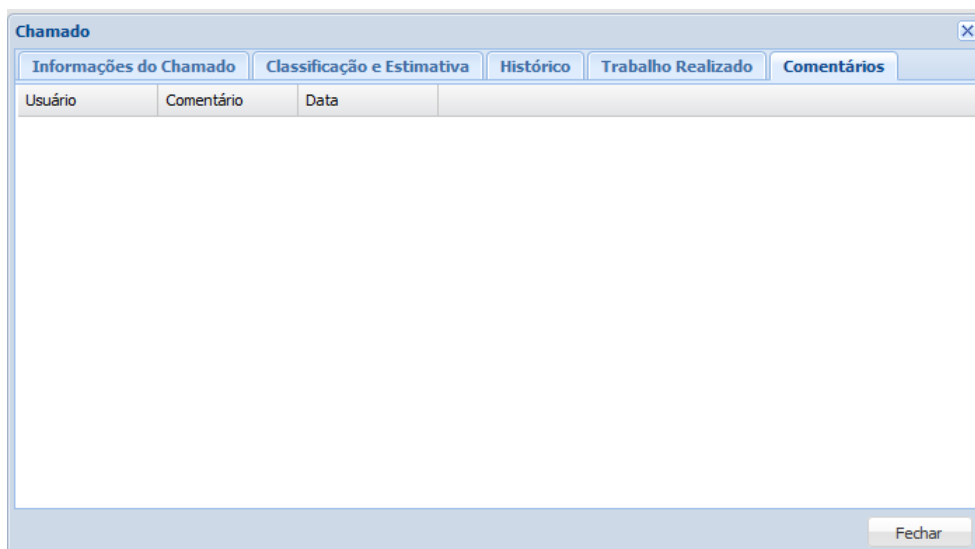


Figura 44 - Visualizar chamado (Aba: Comentários)
Fonte: Autoria própria

5.8 CHAMADOS (CLIENTE)

Os próximos itens são referentes à interface de interação que o cliente possui com o sistema, para que com as ações que ele realizar possa dar continuidade no fluxo do sistema, fechando então todo o ciclo de vida de uma solicitação.

A Figura 45 apresenta a gerência que é disponível para o cliente no momento em que ele acessa o sistema.

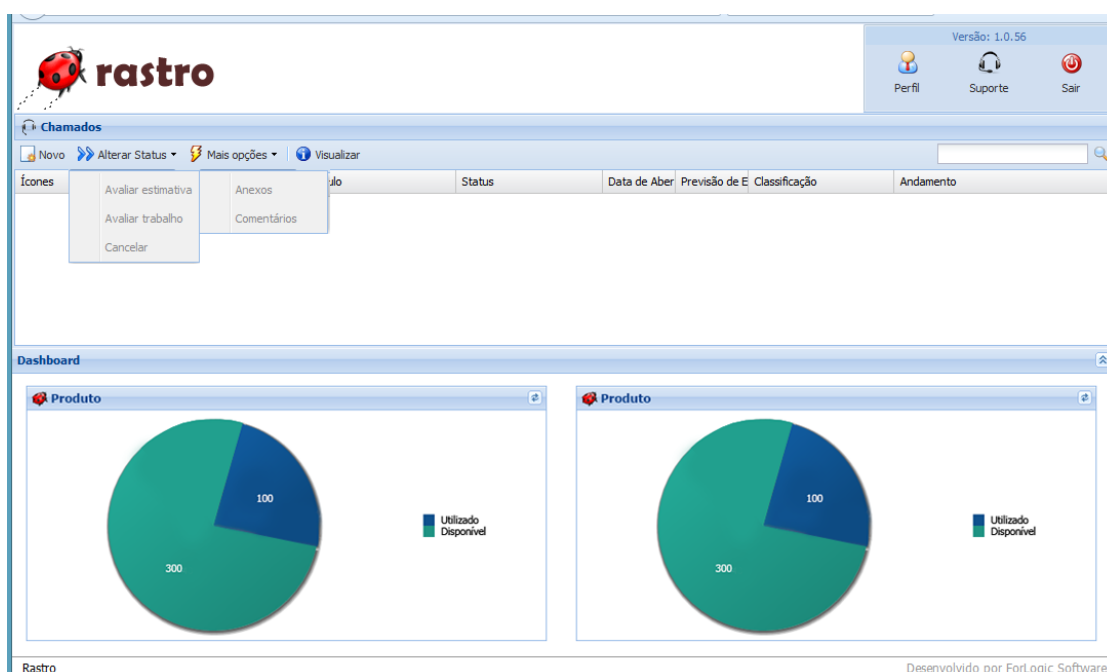


Figura 45 - Gerência de Chamados (Cliente)
Fonte: Autoria própria

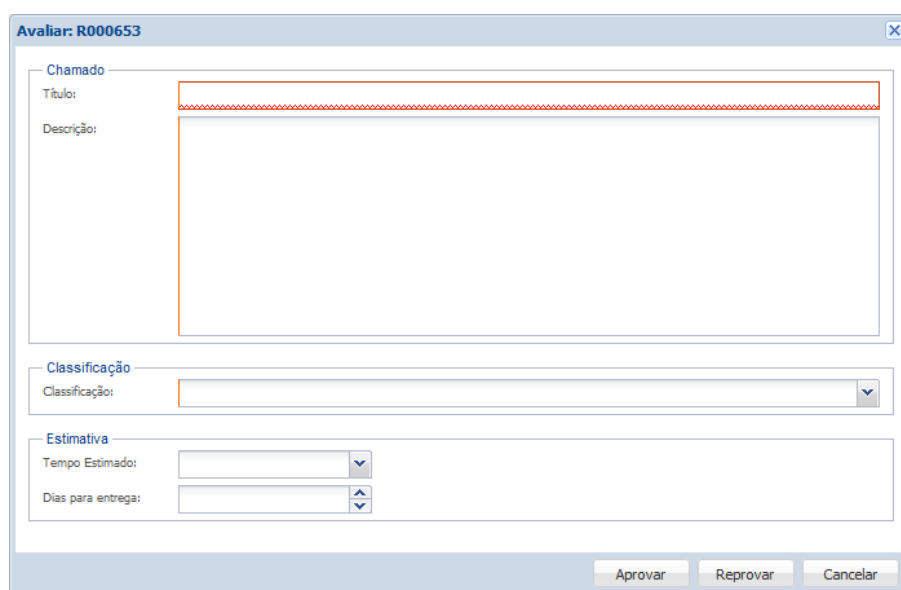
Algumas das funcionalidades que seguem já foram explicadas e exemplificadas nos itens anteriores, sendo assim, serão apenas citadas não seguindo de explicações e ilustrações.

5.8.1 NOVO

Esta funcionalidade tem o mesmo objetivo descrito na frente disponível aos usuários colaboradores da ForLogic, apenas diferenciada por já identificar o cliente e solicitante automaticamente.

5.8.2 AVALIAR CLASSIFICAÇÃO E ESTIMATIVA

Esta funcionalidade é para que o cliente avalie a classificação e estimativa que foi informado pela equipe da Forlogic e então tome sua decisão se aceita ou não o que foi proposta para resolução do problema. A Figura 46 ilustra a tela para realizar esta ação.



A imagem mostra uma janela de software intitulada "Avaliar: R000653". O formulário contém as seguintes seções:

- Chamado:** Campos para "Título:" e "Descrição:".
- Classificação:** Um campo rotulado "Classificação:" com uma seta para baixo.
- Estimativa:** Campos para "Tempo Estimado:" e "Dias para entrega:", ambos com setas para cima e para baixo.

Na base da janela, há três botões: "Aprovar", "Reprovar" e "Cancelar".

Figura 46 - Avaliar classificação e estimativa do chamado
Fonte: Autoria própria

5.8.2 AVALIAR TRABALHO

Esta funcionalidade é para que o cliente avalie se o trabalho que foi solicitado para a ForLogic apresentada na Figura 47 foi realmente desenvolvido conforme suas expectativas, caso tenha sido resolvido, o cliente aprova o trabalho e o fluxo do chamado se encerra, caso ainda tenha ficado algum item pendente, esse trabalho

deve ser reprovado e então o cliente informa o que faltou para que a Forlogic avalie se realmente isso estava explicado no início da abertura do chamado.



Figura 47 - Avaliar trabalho do chamado
Fonte: Autoria própria

5.8.3 CANCELAR

Esta funcionalidade é destinada para que o chamado seja cancelado, qualquer uma das partes que interagem com o sistema pode realizar esta tarefa, e ambas são notificadas por e-mail quando essa ação ocorre, contendo a justificativa para o cancelamento, apresentada na figura Figura 48.

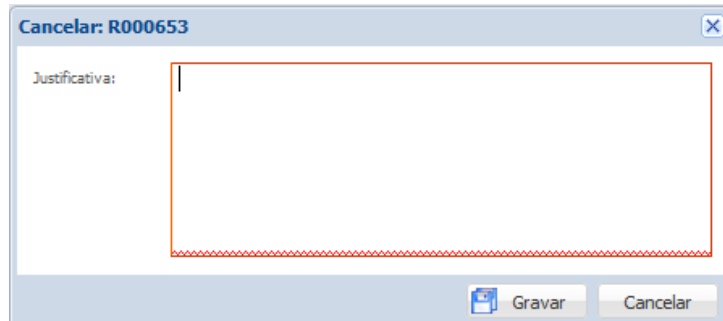


Figura 48 - Cancelar chamado
Fonte: Autoria própria

5.8.4 ANEXOS

Esta funcionalidade representada na Figura 49 permite anexar qualquer tipo de arquivo que não exceda 10Mb para facilitar a equipe da Forlogic a identificar o problema, ou entender melhor a solicitação, qualquer uma das partes pode anexar um arquivo a qualquer momento, mesmo que o chamado já tenha sido finalizado.

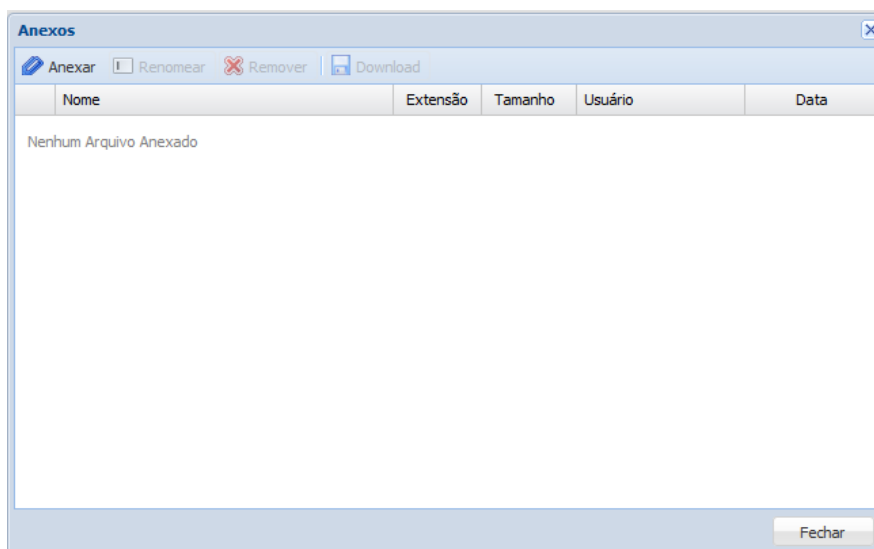


Figura 49 - Anexar arquivos ao chamado
Fonte: Autoria própria

5.8.5 COMENTAR

Esta funcionalidade representada na Figura 50 permite adicionar comentários ao chamado, fazendo com que fique registrado alguns acontecimentos importantes caso algum ator do sistema julgue importante.

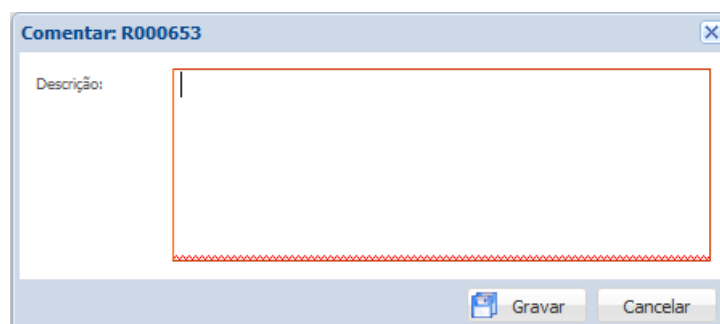
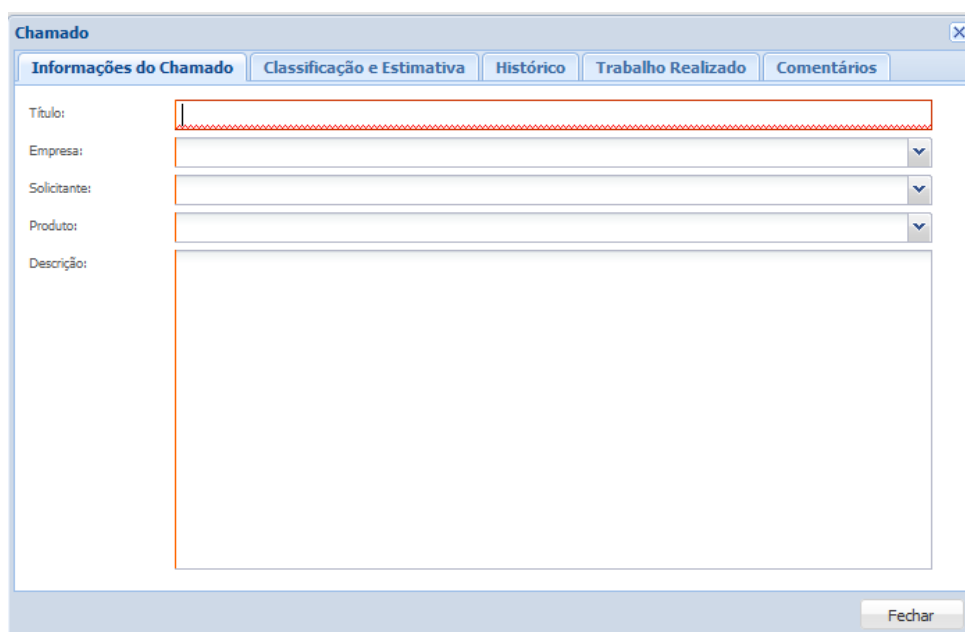


Figura 50 - Comentar chamado
Fonte: Autoria própria

5.8.6 VISUALIZAR

A Figura 51 até a Figura 55 representam a funcionalidade que permite qualquer um dos atores do sistema acompanhar e visualizar todas as informações referentes ao chamado até o dado momento, contando com a aba de histórico que mostra o momento exato em que o usuário realizou determinada ação.

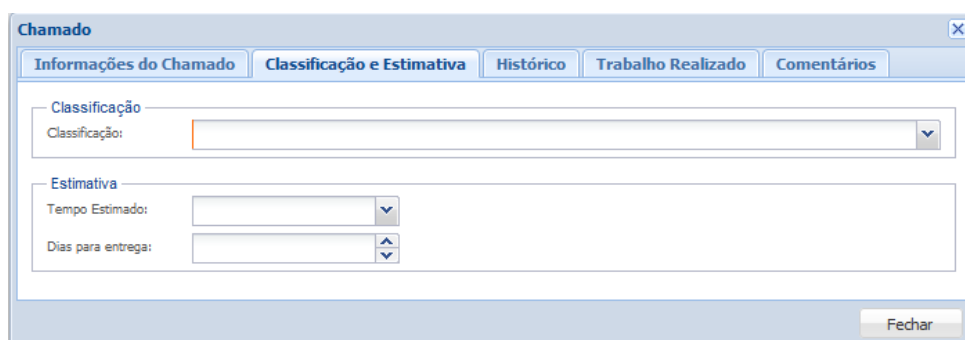


A imagem mostra a interface de usuário para visualizar um chamado, especificamente a aba "Informações do Chamado". O formulário contém os seguintes campos:

- Título: Campo de texto.
- Empresa: Campo de seleção com uma seta para baixo.
- Solicitante: Campo de seleção com uma seta para baixo.
- Produto: Campo de seleção com uma seta para baixo.
- Descrição: Área de texto grande para a descrição do chamado.

Na parte inferior direita do formulário, há um botão "Fechar".

Figura 51 - Visualizar chamado (Aba: Informações do Chamado)
Fonte: Autoria própria



A imagem mostra a interface de usuário para visualizar um chamado, especificamente a aba "Classificação e Estimativa". O formulário contém os seguintes campos:

- Classificação: Campo de seleção com uma seta para baixo.
- Estimativa: Seção contendo:
 - Tempo Estimado: Campo de seleção com uma seta para baixo.
 - Dias para entrega: Campo de seleção com setas para cima e para baixo.

Na parte inferior direita do formulário, há um botão "Fechar".

Figura 52 - Visualizar chamado (Aba: Classificação e Estimativa)
Fonte: Autoria própria

The screenshot shows a window titled 'Chamado' with a close button in the top right corner. Below the title bar are five tabs: 'Informações do Chamado', 'Classificação e Estimativa', 'Histórico', 'Trabalho Realizado', and 'Comentários'. The 'Histórico' tab is selected. Below the tabs is a table with three columns: 'Responsável', 'Ação', and 'Data'. The table is currently empty. At the bottom right of the window is a 'Fechar' button.

Figura 53 - Visualizar chamado (Aba: Histórico)
Fonte: Autoria própria

The screenshot shows the same 'Chamado' window with the 'Trabalho Realizado' tab selected. The main area contains several form fields: 'Classificação:' with a text input field, 'Data de Entrega:' with a date picker, 'Descrição:' with a large text area, 'Horas estimadas:' with a numeric input field, and 'Horas trabalhadas:' with a numeric input field. A 'Fechar' button is located at the bottom right.

Figura 54 - Visualizar chamado (Aba: Trabalho Realizado)
Fonte: Autoria própria

The screenshot shows the 'Chamado' window with the 'Comentários' tab selected. Below the tabs is a table with three columns: 'Usuário', 'Comentário', and 'Data'. The table is empty. A 'Fechar' button is at the bottom right.

Figura 55 - Visualizar chamado (Aba: Comentários)
Fonte: Autoria própria

5.9 E-MAILS

Este requisito contempla a especificação do corpo do e-mail que será enviado pelo sistema conforme a ação realizada pelos atores do sistema que pode ser no caso o usuário sendo um colaborador da Forlogic ou então um cliente.

5.9.1 NOVO CHAMADO

A Figura 56 ilustra o e-mail que é enviado no momento em que um novo chamado é cadastrado no sistema.

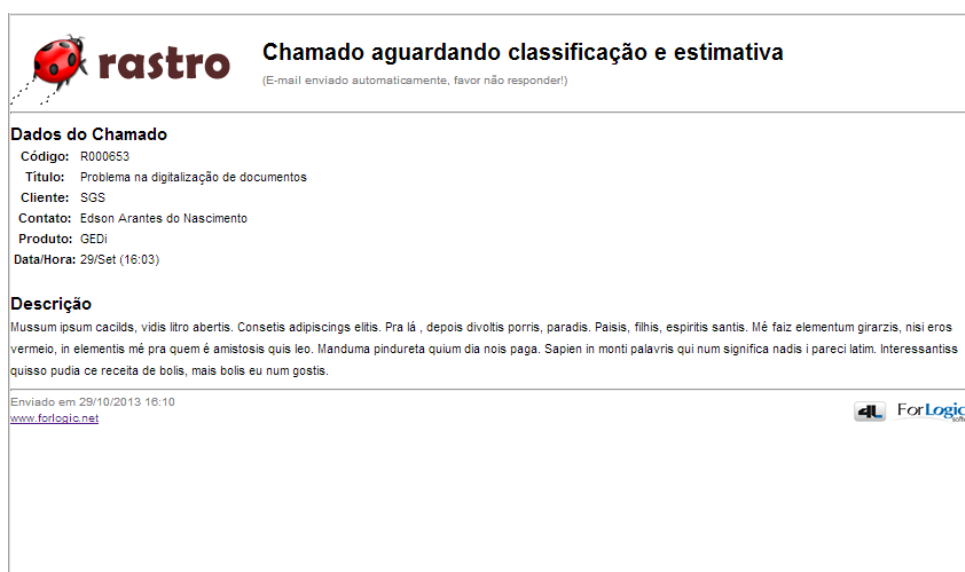


Figura 56 - E-mail novo chamado
Fonte: Autoria própria

5.9.2 CLASSIFICAÇÃO E ESTIMATIVA

A Figura 57 ilustra o e-mail que é enviado no momento em que um chamado acaba de ser classificado e estimado pela equipe da Forlogic.



Figura 57 - E-mail classificação e estimativa
Fonte: Autoria própria

5.9.3 AVALIAÇÃO DA CLASSIFICAÇÃO E ESTIMATIVA

A Figura 58 ilustra o e-mail que é enviado no momento em que a classificação e estimativa de um chamado acabou de ser avaliada pelo cliente.

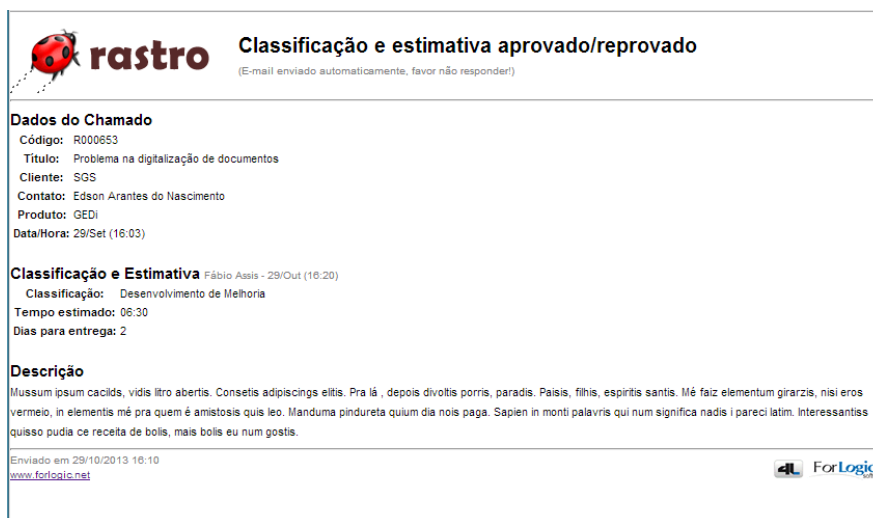


Figura 58 - E-mail avaliação da classificação e estimativa
Fonte: Autoria própria

5.9.4 COMENTÁRIO

A Figura 59 ilustra o e-mail que é enviado no momento em que um comentário é feito em um chamado, por qualquer um dos usuários.

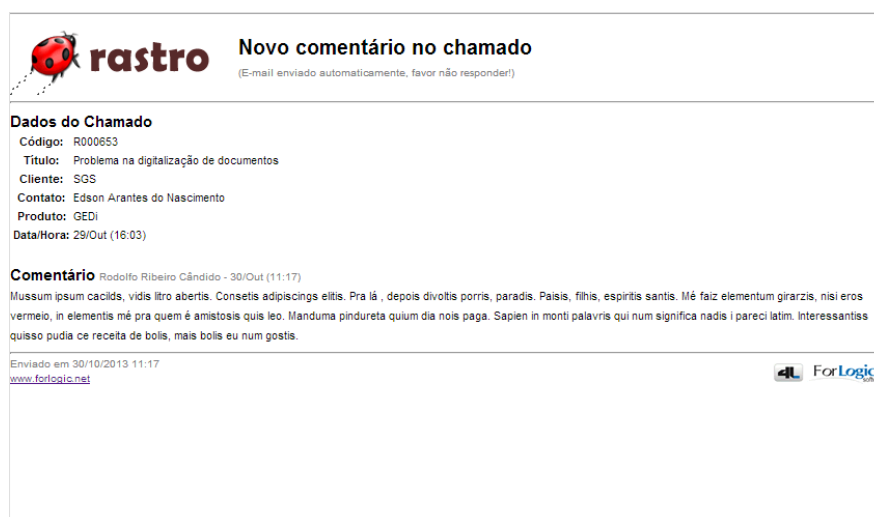


Figura 59 - E-mail comentário
 Fonte: Autoria própria

5.9.5 CANCELAMENTO

A Figura 60 ilustra o e-mail que é enviado no momento em que um chamado é cancelado, por qualquer um dos usuários.

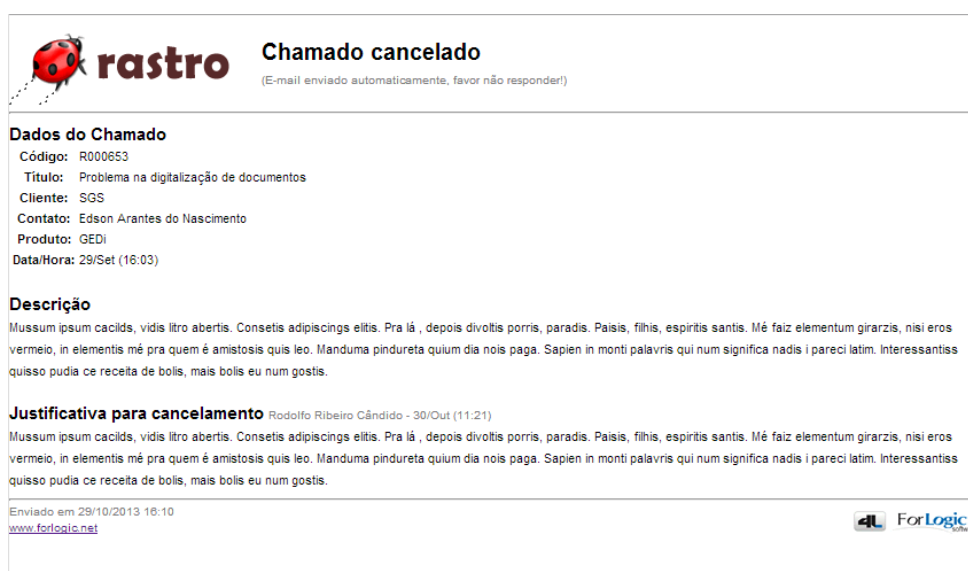


Figura 60 - E-mail cancelamento
 Fonte: Autoria própria

5.9.6 FINALIZAÇÃO

A Figura 61 ilustra o e-mail que é enviado no momento em que um chamado é finalizado pela equipe da Forlogic e passa a aguardar a avaliação do trabalho pelo Cliente.

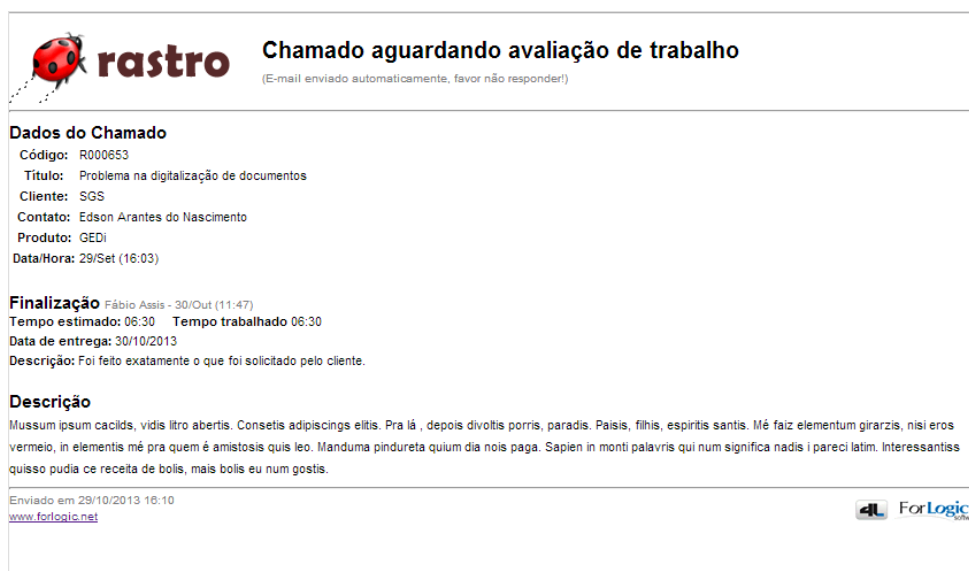


Figura 61 - E-mail finalização
 Fonte: Autoria própria

5.9.7 AVALIAÇÃO DO TRABALHO

A Figura 62 ilustra o e-mail que é enviado no momento em que a avaliação do trabalho é realizada.

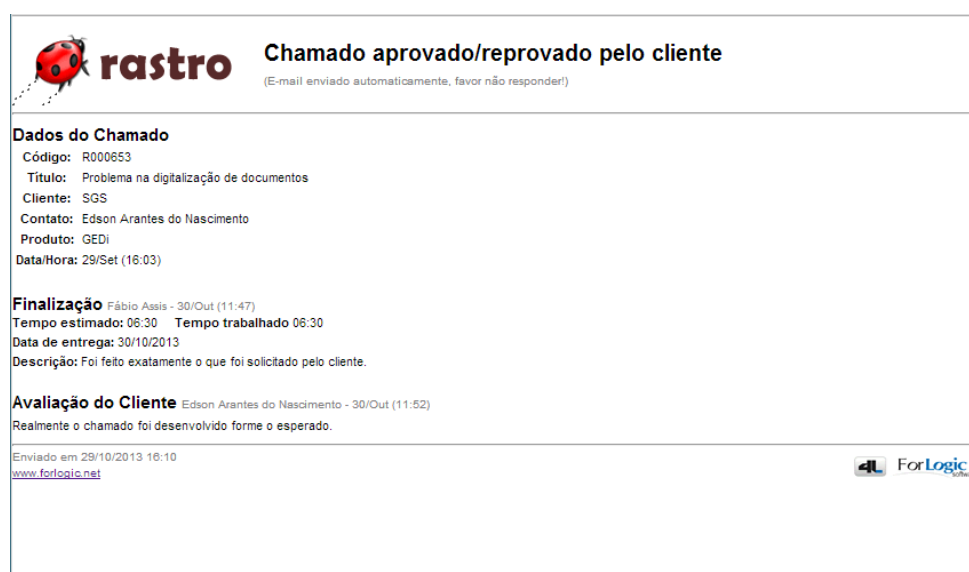


Figura 62 - E-mail avaliação do trabalho
 Fonte: Autoria própria

5.9.8 AVALIAÇÃO DA REABERTURA

A Figura 63 ilustra o e-mail que é enviado para quando a avaliação do trabalho realizada pelo cliente foi negativa e ele solicita que o mesmo seja reaberto para a conclusão correta do desenvolvimento.

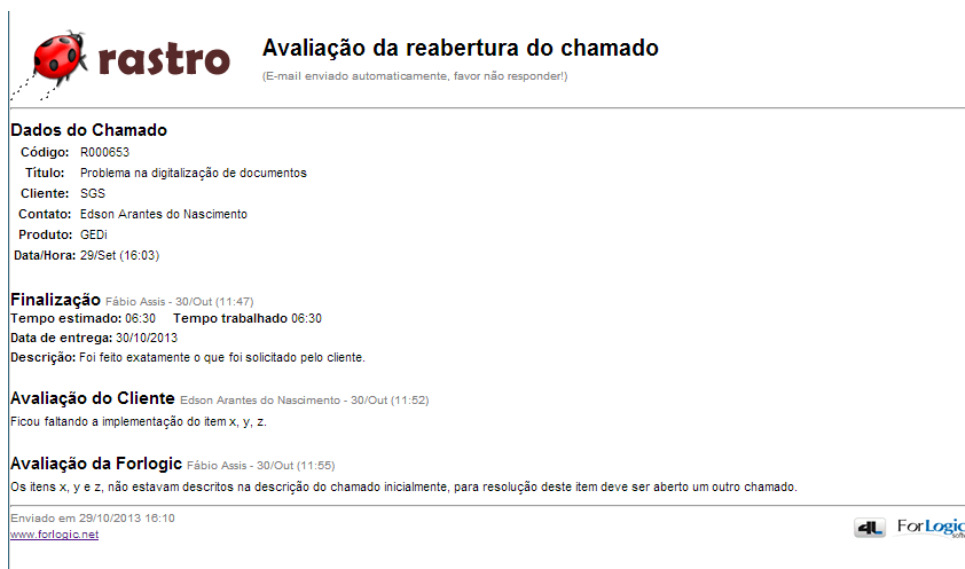


Figura 63 - E-mail avaliação da reabertura
 Fonte: Autoria própria

5.9.9 REESTIMAR

A Figura 64 ilustra o e-mail que é enviado no momento em que um chamado é feita qualquer alteração na classificação e estimativa pela Forlogic, através da funcionalidade reestimar.



Figura 64 - E-mail reestimar
 Fonte: Autoria própria

6. CONSIDERAÇÕES FINAIS

6.1 DIFICULDADES ENCONTRADAS

Foram encontradas diversas dificuldades durante o desenvolvimento, dentre elas a mais grave ocasionou a retirada de um requisito de projeto. Nesta primeira versão foi a falta de conhecimento e tempo para estudo da biblioteca ExtJS da Sencha, necessários para a implementação de gráficos, que ficariam dispostos abaixo da grid de chamados da área de cliente para ilustrar e fornecer de maneira prática os dados referente à horas de suporte, que a empresa possui contratualmente e a quantidade de horas de suporte que já foram prestadas através dos chamados que já foram abertos.

6.2 TRABALHOS FUTUROS

A página inicial com gráficos para acompanhamento que foi proposto no início do projeto não pode ser desenvolvida por dificuldades técnicas que foram encontradas e que demandariam muito tempo de estudo para ser implementada. Assim, essa funcionalidade ficará como trabalho futuro para o sistema construído.

Diante dos resultados apresentados no Capítulo 5, a ferramenta atende a necessidade da ForLogic, contudo, algumas solicitações de melhorias foram identificadas durante o período de teste realizado enquanto o sistema estava sendo homologado.

Essas solicitações de melhorias estão sendo avaliadas internamente pela equipe responsável na empresa e possivelmente estarão sendo definidas também como funcionalidades a serem desenvolvidas futuramente.

6.3 RESULTADOS

Utilizando o questionário de avaliação que foi elaborado e aplicado aos usuários que participaram da homologação e período de testes do sistema, foi realizada a análise dos resultados, permitindo atingir um resultado um pouco mais concreto, sobre a aceitação e usabilidade da ferramenta. Com relação ao campo aberto na avaliação onde os usuários poderiam descrever livremente sugestões, críticas e melhorias, foi encaminhado ao responsável da equipe de suporte para que avalie qual o impacto que os itens geram e priorize-os para que sejam atendidos.

Em geral, após a análise dos questionários o resultado da utilização da ferramenta foi positivo. A aceitação do sistema pelos avaliados gerou uma nota média igual a nove e vinte e cinco. Além disso uma pergunta pede para o usuário sugerir para a ferramenta ser aprimorada, permitindo identificar novas funcionalidades interessantes, ficou claro a necessidade dos gráficos na *dashboard*, não desenvolvida. O questionário comprovou que o sistema possui uma boa navegação, permitindo que os usuários encontrem as informações que precisam de forma simples e rápida. A Tabela 6 ilustra os dados que foram coletados através das respostas do questionário.

Tabela 6 - Resultados análise do questionário

Avaliado	Nota
Indivíduo 1	10
Indivíduo 2	10
Indivíduo 3	9
Indivíduo 4	8
Média	9,25

Fonte: Autoria própria

6.4 CRONOGRAMA

A Tabela 7 representa o cronograma que foi utilizado desde o início do projeto quando foi dado início à proposta do desenvolvimento do trabalho até a conclusão do projeto.

Tabela 7 - Cronograma

Etapa	2013												2014	
	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	
Proposta	■	■	■											
Monografia				■	■	■	■	■	■	■	■	■	■	■

Fonte: Autoria própria

REFERÊNCIAS

AMARAL, L. G. **CSS Cascading stylesheets: guia de consulta rápida**. São Paulo: Novatec, 2006.

BECK, K. et al. Manifesto for Agile Software Development. 2001. Disponível em: <<http://agilemanifesto.org>>. Acesso em: 26/06/2010.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. Unified Modeling Language User Guide. Não Disponível: Addison Wesley, 1998. 512 p.

BROOKS, F. The Mythical Man-Month. New York: Addison-Wesley, 1995.

FLANAGAN, David. **JavaScript: O guia definitivo**. 4. ed. Porto Alegre: Artmed, 2002.

Forlogic Software. **Guia do Ciclo de Vida**. Versão 2.9. Cornélio Procópio, 2013.

Forlogic Software. **Procedimentos do Processo 4L**. Versão 2.9. Cornélio Procópio, 2013.

FORLOGIC SOFTWARE. **P29-0008.003: Guia de Configurações**. Cornélio Procópio, 2013. 16 p.

FUMSOFT. **Modelo MPS.BR. Informativo FUMSOFT News**. Disponível em: <http://www.fumsoft.org.br/qualidade/modelo_mpsbr>. Acesso em: 20 jan. 2014.

GARRETT, J. **Ajax: Uma nova Abordagem para Aplicação Web**. 2005. Disponível em: <<http://www.adaptivepath.com/publications/essays/archives/000385.php>>. Acesso em: 11 fev. 2013

Kival C. Weber, Ana Regina Rocha, Ângela Alves, Arnaldo M. Ayala, Austregésilo Gonçalves, Benito Paret, Clênio Salviano, Cristina F. Machado, Danilo Scalet, Djalma Petit¹, Eratóstenes Araújo, Márcio Girão Barroso, Kathia Oliveira, Luiz Carlos A. Oliveira, Márcio P. Amaral, Renata Endriss C. Campelo, Teresa Maciel. **Modelo de Referência para Melhoria de Processo de Software: uma abordagem brasileira**. 2004. Disponível em: <http://www.softex.br/wp-content/uploads/2013/09/Artigo-CLEI.pdf>. Acesso em: 20/12/2013

MENDES, Antonio. **Arquitetura de Software: Desenvolvimento orientado para arquitetura**. Rio de Janeiro: Campus, 2002.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995.

ROYCE, W. **Managing the Development of Large Software Systems**. IEEE WESCON, IEEE, v. 26, p. 1–9, 1970.

SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum, Um guia definitivo para o Scrum: As regras do jogo**. 2013. Disponível em: <<https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 26 fev. 2014.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

TAYLOR, F. **Princípios da Administração Científica**. 8. ed. São Paulo: Atlas, 1990.

APÊNDICE A
QUESTIONÁRIO DE AVALIAÇÃO DO SISTEMA

Questionário de Avaliação do Sistema *Rastro 2.0 Manutenção*

Nome:

Função:

Atividades:

1. O sistema possui uma fácil navegação?

2. As informações são fáceis de serem localizadas?

3. Foi possível obter o status/progresso de um chamado de forma rápida e prática?

4. Foi possível saber todos os passos e respectivos responsáveis que um chamado passou?

5. O que você sugere que seja feito para evolução e aprimoramento da ferramenta?

6. De um nota de 0 (zero) a 10 (dez) ao sistema.

Nota: _____

ANEXO A
PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

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

RODOLFO RIBEIRO CÂNDIDO

Rastro 2.0
Módulo Manutenção

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2013

RODOLFO RIBEIRO CÂNDIDO

Rastro 2.0
Módulo Manutenção

Proposta de Trabalho de Conclusão de Curso, apresentada 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 obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Danilo Sapoli.

CORNÉLIO PROCÓPIO

2013

RESUMO

CANDIDO, Rodolfo R.. **Rastro 2.0: Módulo Manutenção**. 2013. 24p. Proposta de Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Cornélio Procópio 2013. Este documento descreve o desenvolvimento de um módulo *Web* para realizar o acompanhamento e gerenciamento do ciclo de vida de solicitações para realização de manutenção em sistemas de informação. A importância do desenvolvimento deste sistema é a dificuldade em manter centralizadas as mudanças e correções que devem ser feitas.

Palavra-chave: Manutenção. Gerenciamento. Fluxo. Solicitação.

ABSTRACT

CANDIDO, Rodolfo R.. **Rastro 2.0: Maintenance Module**. 2013. 24p. Proposta de Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Cornélio Procópio 2013. This document describes the development of a web module to perform monitoring and managing the life cycle of applications to perform maintenance information systems. The importance of the development of this system is the difficulty in maintaining centralized changes and corrections to be made.

Keywords: Maintenance. Management. Workflow. Request.

LISTA DE FIGURAS

Figura 1 - O ciclo de vida clássico (Modelo em cascata). **Erro! Indicador não definido.**

Figura 2 - Processo 4L **Erro! Indicador não definido.**

Figura 3 - Padrão MVC **Erro! Indicador não definido.**

Figura 4 - Diagrama Entidade Relacionamento..... **Erro! Indicador não definido.**

LISTA DE TABELAS

Tabela 1 – Concepção	Erro! Indicador não definido.
Tabela 2 – Elaboração	Erro! Indicador não definido.
Tabela 3 – Construção	Erro! Indicador não definido.
Tabela 4 – Transição.....	Erro! Indicador não definido.
Tabela 5 - Solicitação de Mudança	Erro! Indicador não definido.
Tabela 6 – Cronograma Geral.....	Erro! Indicador não definido.
Tabela 7- Pontos por Função	Erro! Indicador não definido.

LISTA DE ABREVIATURA

AJAX	<i>Asynchronous Javascript and XML</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</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>

SUMÁRIO

1. INTRODUÇÃO	Erro! Indicador não definido.
2. JUSTIFICATIVA.....	11
3. OBJETIVOS.....	12
3.1. GERAL	Erro! Indicador não definido.
3.2. ESPECÍFICOS	12
4. MÉTODOS.....	13
6. ARQUITETURA DO SOFTWARE	19
7. CRONOGRAMA GERAL E PONTOS POR FUNÇÃO	21
REFERÊNCIAS.....	22
ANEXO A.....	23

1. INTRODUÇÃO

Este documento tem como objetivo apresentar a proposta de trabalho de conclusão de curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – UTFPR, campus Cornélio Procópio, cujo tema refere-se ao desenvolvimento de um módulo *Web* para realizar o controle de projeto, horas e solicitações de suporte para os produtos da empresa *Forlogic Software*.

A Forlogic foi criada por ex-alunos da UTFPR, oriunda de um projeto que se iniciou no Hotel Tecnológico da própria universidade, foi oficialmente fundada em 18 de maio de 2004, em Cornélio Procópio – PR. Sendo o primeiro projeto a realmente ter continuidade, formado pelos alunos na disciplina de empreendedorismo, está direcionada para o desenvolvimento de soluções de *software* que atualmente possui alguns nichos de trabalho, oferecendo solução através de produtos específicos e fábrica de *software*.

Produto seria uma das ferramentas em que são vendidas em forma de pacote, a Forlogic disponibiliza suporte conforme o plano contratado e a regra do *software* já está definida e implementada.

Fábrica de *software* seria a criação de um produto totalmente novo onde o cliente não encontrou uma solução que atendesse suas necessidades em um *software* pronto, sendo assim, é feito um levantamento de requisitos e o *software* é feito sob demanda para atender as necessidades que o cliente precisa.

Com todas essas opções de desenvolvimento, seja produto ou fábrica, é preciso sempre ter um controle do que está sendo feito, como, por quem, quanto tempo está sendo gasto, etc., ou seja, deve ser possível gerenciar todo o desenvolvimento da empresa. A empresa hoje possui diversos clientes utilizando os produtos.

De acordo com o ciclo de vida clássico (Modelo em cascata) da engenharia de *software* é possível concluir alguns fatores para a realização deste trabalho.

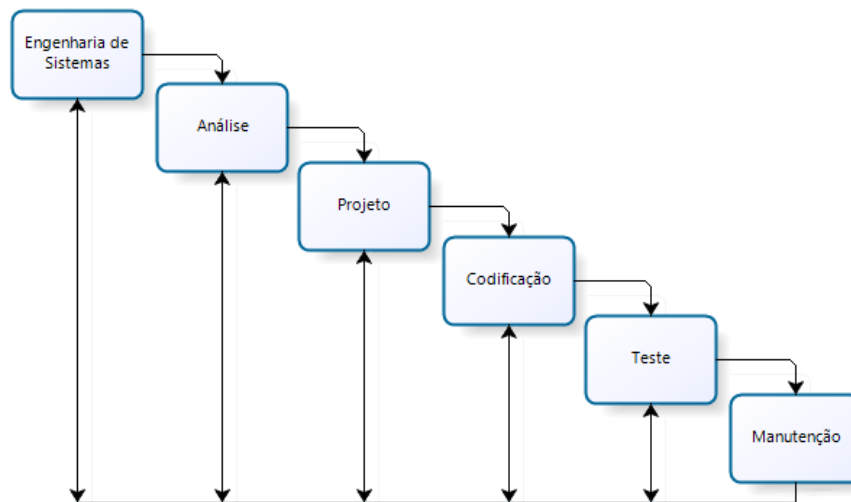


Figura 1 - O ciclo de vida clássico (Modelo em cascata).
Fonte: Sommerville (2007)

O ciclo de vida clássico deixa separada cada fase de um processo em cascata, onde somente uma fase será iniciada após a conclusão de sua antecessora, e por fim temos a fase de manutenção.

Indubitavelmente, o software sofrerá mudanças depois que foi entregue ao cliente (uma possível exceção é o software embutido). Ocorrerão mudanças porque erros foram encontrados, porque o software deve ser adaptado a fim de acomodar mudanças em seu ambiente externo (por exemplo, uma mudança exigida por causa de um novo sistema operacional ou dispositivo periférico) ou porque o cliente exige acréscimos funcionais ou de desempenho. A manutenção de software re replica cada uma das etapas precedentes do ciclo de vida a um programa existente, e não a um novo (PRESSMAN, 1995).

Segundo Sommerville (2007), geralmente (embora não necessariamente) esta é a fase mais longa do ciclo de vida. Podendo haver a necessidade de corrigir erros que não foram identificados nas outras fases do processo de desenvolvimento. Neste momento, também, ocorre à implementação de novas unidades ao sistema conforme novos requisitos são identificados.

O módulo que apresentado nesse trabalho visa realizar o gerenciamento desta fase do ciclo de vida citado acima, o que irá fornecer o controle das solicitações que os clientes fazem para a empresa. Tais solicitações devem ser gerenciadas e devem possuir um fluxo definido o qual a equipe interna deve seguir.

Este trabalho tem como foco principal a construção desta ferramenta utilizando uma tecnologia mais atual no contexto de tecnologias *Microsoft*, que é o *Asp.Net MVC* com *C#* especificamente.

2. JUSTIFICATIVA

Em uma reunião realizada no início do mês de fevereiro de 2013 com o coordenador de projetos e o gerente de desenvolvimento da empresa, foi levantado à necessidade de aplicar conceitos tecnológicos atualizados e desenvolver uma ferramenta que fosse capaz de gerenciar todo o ciclo de vida dos projetos desenvolvidos pela empresa.

Uma das grandes dificuldades para a Forlogic é esse gerenciamento, onde existe diversas variáveis que devem ser acompanhadas diariamente para manter sempre os cronogramas.

A manutenção solicitada pelo cliente em um produto de *software* exige um acompanhamento, o qual deve ser analisado, aprovado para então iniciar um procedimento para atualização de documentação e homologação após o desenvolvimento para verificação dos resultados do trabalho.

Desde o início da solicitação é necessário manter em uma única ferramenta todo o controle das solicitações de manutenção. Nesse sentido é possível verificar a importância do desenvolvimento de um módulo para o gerenciamento das solicitações feitas pelos clientes para a manutenção, seja ela alguma correção, ajuste ou melhoria em algum processo no produto contratado.

Para que haja um controle sobre todo o ciclo de vida da solicitação em questão, é necessário existir um fluxo ou um procedimento para fazer o atendimento.

Primeiramente, a solicitação inicia o ciclo de vida com a equipe de suporte, a qual analisa a origem do item e estima o tempo que será gasto para o desenvolvimento requisitado, dando início ao atendimento. Esse atendimento pode ser pela própria equipe do suporte ou então passado para a equipe de desenvolvimento, que fará utilização das mesmas informações, porém, em outra interface que não está contida no escopo deste trabalho.

3. OBJETIVOS

3.1 GERAL

O objetivo geral deste trabalho é desenvolver uma ferramenta capaz de promover o acesso por dois tipos de usuários, o cliente da empresa e o colaborador. O primeiro terá acesso a algumas opções para realizar a abertura de solicitações de suporte e ainda acompanhar o desenvolvimento destas livremente e, o colaborador que fará o acesso conforme as permissões que serão atribuídas ao seu cargo.

3.2 ESPECÍFICOS

- Possuir tela para acesso para usuário do tipo colaborador suporte.
- Possuir tela de acesso para usuário do tipo colaborador desenvolvedor.
- Possuir tela de acesso para usuário do tipo cliente.
- Estruturar a forma como é feito o atendimento e gerenciamento das solicitações.
- Permitir que o cliente visualize todos os seus chamados e acompanhe o andamento a qualquer momento.
- Permitir que seja apontado o tempo trabalhado pela equipe do suporte diretamente na tela de acesso para o usuário tipo suporte.
- Permitir a passagem de uma solicitação que está em suporte seja enviada para o desenvolvimento.

4. MÉTODOS

O processo de desenvolvimento de *software* que será utilizado para o desenvolvimento deste trabalho, é de autoria da empresa Forlogic e nesta seção serão apresentadas suas fases e atividades.

Segundo Pressman (2002), no desenvolvimento de *software* um fator determinante para obter os resultados esperados, que foram definidos no início do projeto é a utilização de um ciclo de vida, pois o mesmo colabora deixando de forma explícita o que deve ser feito durante todo o processo. No texto, Pressman afirma também que no ciclo de vida pode ser feito ajustes para definir o processo da melhor maneira possível a fim de evoluir em velocidade, qualidade, gerência de riscos e custos mantendo sempre tudo sobre controle desde o início.

Dessa forma, a Forlogic desde a sua criação mantém uma equipe especializada no estudo de processos de desenvolvimento e de qualidade, a fim de adquirir experiências nas diversas atividades de todo o ciclo que engloba o desenvolvimento de *software*. O processo de desenvolvimento de *software* utilizado atualmente na Forlogic é baseado no ciclo de vida iterativo incremental e está na sua versão 2.5 com a seguinte estrutura:

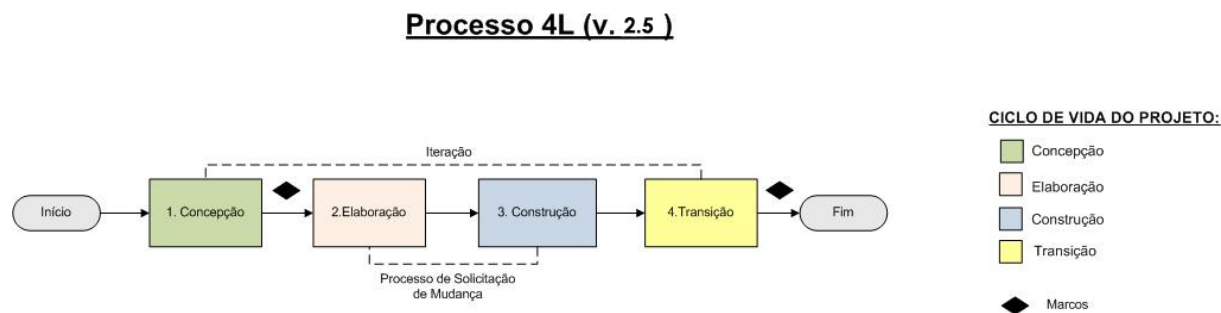


Figura 2 - Processo 4L
 Fonte: Guia de Ciclo de Vida (2012)

A figura 2 apresenta o fluxo de execução do projeto: Concepção, Elaboração, Construção e Transição. Durante a elaboração e transição há dois marcos no projeto, Marco 1 (Elaboração) e Marco 2 (Transição). Com a possibilidade de existir outros marcos durante a construção, chamados marcos intermediários.

Para o Marco 2 (Transição) é obrigatória a presença de toda a equipe que se envolveu durante todo o ciclo de vida do projeto, é nessa reunião onde são

levantadas as falhas e possíveis melhorias que ocorreram no decorrer do projeto, afim de serem feitas melhorias no processo de desenvolvimento.

Em cada fase do processo de desenvolvimento existem diversas atividades que devem ser feitas pela a equipe e pelo líder do projeto, com algumas restrições, devem ser feitas sequencialmente e uma fase só pode ser iniciada após a conclusão da anterior, a não ser a etapa de solicitação de mudança que pode ser iniciada a qualquer momento conforme houver uma mudança no escopo do projeto durante a construção. As atividades são:

Tabela 1 – Concepção

Concepção	38. Registrar escopo (positivo e negativo) do projeto.
	39. Registrar requisitos (cat. Projeto).
	40. Avaliar requisitos (cat. Processo de Desenvolvimento).
	41. Realizar estimativa (cat. Projeto).
	42. Identificar os riscos do projeto (cat. Projeto).
	43. Realizar planejamento do projeto (Horas, cronograma, divulgação e recursos).
	44. Analisar viabilidade do projeto (cat. Processo de desenvolvimento).
	45. Enviar e-mail ao fornecedor de requisitos (cat. Processo de desenvolvimento).
46. Melhoria do ambiente de trabalho (cat. Processo da Empresa).	

Tabela 2 – Elaboração

Elaboração	47. Criar estrutura de armazenamento e diagramas (cat. Projeto).
	48. Criar matriz de rastreabilidade horizontal (cat. Processo de Desenvolvimento).
	49. Cadastrar tickets no Rastro (cat. Projeto).
	50. M1 – Comunicar equipe de projeto - 1º marco (cat. Projeto).
	51. Melhoria do ambiente de trabalho (cat. Processo da Empresa).

Tabela 3 – Construção

Construção	52. Reunião e atividades diárias (cat. Projeto).
	53. Monitoria / Marco Intermediário (cat. Projeto).
	54. Verificar matriz de rastreabilidade vertical (cat. Processo de Desenvolvimento).
	55. Verificar integridade dos artefatos (cat. Processo de Desenvolvimento).
	56. Planos de ação para riscos (cat. Processo de Desenvolvimento).
	57. Melhoria do ambiente de trabalho (cat. Processo da Empresa).

Tabela 4 – Transição

Transição	58. Preparar projeto para entrega ao cliente (cat. Projeto).
	59. M2 – Realizar retrospectiva da equipe – 2º marco (cat. Processo de Desenvolvimento).
	60. Finalizar projeto (cat. Processo de Desenvolvimento).
	61. Melhoria do ambiente de trabalho (cat. Processo da Empresa).

A tabela abaixo não tem relação direta com o objetivo geral do projeto que é controlar as solicitações de manutenção dos projetos.

Tabela 5 - Solicitação de Mudança

Solicitação de Mudança	62. Analisar requisitos.
	63. Preencher documento de solicitação de mudança.
	64. Avaliar o impacto nos requisitos.
	65. Avaliar impacto na execução.
	66. Enviar e-mail ao fornecedor de requisitos.
	67. Atualizar documentação do projeto.
	68. Comunicar equipe sobre mudança.
	69. Verificar integridade dos artefatos
	70. Melhoria do ambiente de trabalho (cat. Processo da Empresa).

Todas as tarefas da tabela 5 são da categoria Processo de Solicitação de Mudança.

A fase de Solicitação de Mudança que foi apresentada por último, não tem a obrigatoriedade de ser executada durante o projeto a não ser que alguma mudança no escopo do projeto aconteça. Essa fase é destinada para o registro e análise de impacto para quando uma mudança de escopo deve ocorrer (porém, o projeto já está em andamento), seja ela por uma decisão do cliente ou por necessidade caso tenha ocorrido alguma falha no levantamento do projeto.

5. TECNOLOGIAS UTILIZADAS

Como a aplicação pertence à Forlogic e a empresa já uma definição das tecnologias que devem ser utilizadas, neste capítulo, tais tecnologias serão apresentadas:

- *AJAX – Asynchronous Javascript and XML*

O nome *AJAX* é uma sigla para “*Asynchronous Javascript and XML*”. Trata-se de uma maneira de fazer com que seu navegador, com *Javascript*, carregue conteúdo do servidor sem recarregar a página atual (GARRETT, 2005). *AJAX* é carregar e renderizar uma página, utilizando recursos de scripts rodando pelo lado cliente, buscando e carregando dados em *background* sem a necessidade de *reload* da página. *AJAX* não é uma tecnologia, mas sim um conjunto de tecnologias. O conceito de *AJAX* se resume em conhecer bem *JavaScript*, trabalhando com *DOM (Document Object Model)*, *CSS (Cascading Style Sheets)* e *XML*.

- *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* e fornecer uma alternativa ao modelo *WebForm* do *ASP .NET* disponível até então. O *framework ASP.NET MVC* fornece um ambiente robusto e leve que esta integrado aos recursos do *ASP .NET* (MACORATTI, 2009).

- *CSS – Cascading Style Sheets*

O nome *CSS* é uma sigla para “*Cascading Style Sheets*” que são folhas de estilos para páginas *Web*, adicionadas ou não ao código *HTML*. Fornece aos projetistas e usuários um maior controle sobre como as páginas *Web* são exibidas (AMARAL, 2006). Com *CSS*, podem-se criar folhas de estilo em um arquivo separado do código.

- *Ext.Js 4.1*

Ext é um conjunto de bibliotecas *JavaScript* para criação de aplicações *web* com recurso *RIA (Rich Internet Application)*, foi criado por Jack Slocum em 2006. A

versão 4 do *Extjs* foi lançada em 2012 com mais de 350 novos *APIs*, 50 novas classes, e conta com um novo pacote de dados que permite usar a arquitetura *MVC*. O *ExtJs* 4 teve seu código totalmente reescrito pensando em melhorar o desempenho. Essa reescrita contribuiu para manter a compatibilidade da biblioteca com a grande gama de navegadores disponíveis atualmente no mercado como Internet Explorer 6+, Firefox 1.5+ (PC/MAC), Safari 3+ e Opera 9+ (PC/MAC).

- *Firebug 1.12*

É um *plug-in open source* que integrada com o *Browser Firefox* fornece ao desenvolvedor uma suíte de ferramentas que seja possível de uma forma simples e rápida que permite a edição, depuração e monitoramento de códigos de páginas *Web nas linguagens HTML, JS e CSS, por exemplo*.

- *JAVASCRIPT*

É uma linguagem de programação interpretada e desenvolvida para atender algumas necessidades da aplicação, tais como: validação de formulários no lado do cliente, cálculos complexos, manipulação dos dados na página entre outros, tornando as páginas interativas e dinâmicas (FLANAGAN, 2006). Sua união com o *CSS* é conhecida como *DHTML*. Usando o *Javascript*, é possível modificar dinamicamente os estilos dos elementos da página em *HTML*.

- *SQL Server 2012*

O *SQL Server* é um servidor de banco de dados da *Microsoft*, que disponibiliza uma ferramenta para facilitar o seu gerenciamento chamada *SQL Server Management Studio*, que fornece uma interface mais facilitada de ser utilizada.

- *TortoiseSVN*

Também conhecido como *Subversion*, é um sistema de controle de versões. Será utilizado para controlar as versões de codificação, possibilitando a rastreabilidade de código-fonte através dos *logs de commits e updates* no repositório.

- *Visual Studio 2012*

O *Visual Studio 2012* é uma suíte de *softwares* disponibilizada pela *Microsoft* em diversas versões, voltada para o desenvolvimento de *software* com foco nas tecnologias pertencentes à própria *Microsoft* (*C#*, *VB*, *WCF*), com diversos recursos para o aumento da produtividade do desenvolvedor.

6. ARQUITETURA DO SOFTWARE

Nesse capítulo será apresentado o modelo de arquitetura MVC (*Model View Controller*), será utilizado este padrão por dois motivos, um relacionado diretamente pela tecnologia ser nova no mercado quando se diz respeito à possibilidade de utilizado juntamente com um *framework* Microsoft, e outro pelas facilidades que são proporcionadas para o desenvolvimento, desde a criação de novas funcionalidades como a manutenção dos projetos, esse padrão possui sua estrutura bem separada, seguindo alguns dos padrões da orientação a objetos.

A utilização correta deste modelo consiste em separar os três principais atores que formam o *software* e o MVC trabalha de forma que as camadas do sistema fiquem bem definidas, e para que em alguma eventualidade manutenção, fique mais fácil de localizar os problemas e conseqüentemente os locais onde devem ser aplicadas as devidas soluções.

Segundo Mendes (2002), o MVC divide o sistema em três camadas Modelo (*Model*), Visão (*View*), Controlador (*Controller*).

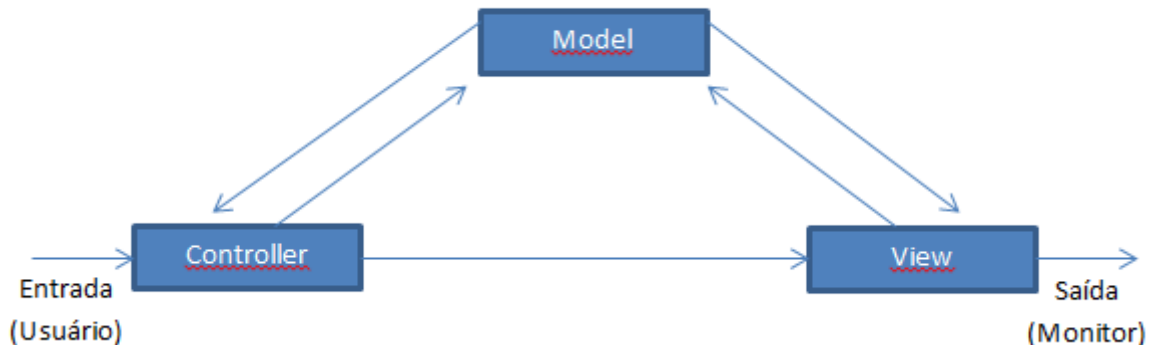


Figura 3 - Padrão MVC

Fonte: Mendes (2002)

Mendes define *Model*, *View* e *Controller* como um agente e cada agente possuem um estado e conhecimento, podendo iniciar e reagir a eventos. Seguindo este mesmo raciocínio abaixo resumo da função de cada um dos agentes.

- *Model*

- 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*.
- *View*
 - Responsável por tudo que o usuário final visualiza, toda a interface, informação, não importando sua fonte de origem, é exibida graças á camada de visão.
- *Controller*
 - Responsável por controlar todo o fluxo de informação que passa pelo sistema. É este agente 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.

7. CRONOGRAMA GERAL E PONTOS POR FUNÇÃO

Tabela 6 – Cronograma Geral

CRONOGRAMA (2013)																																
	Fevereiro				Março				Abril				Maio				Junho				Julho				Agosto				Setembro			
Proposta do Trabalho	■	■	■	■	■	■	■	■	■	■	■	■																				
Concepção									■	■																						
Elaboração											■	■																				
Construção													■	■	■	■	■	■	■	■												
Transição																			■	■												
Redação/Apresentação																	■	■	■	■	■	■	■	■	■	■	■	■				

Tabela 7- Pontos por Função

ALI		
Baixa (ALlb)	14	98
Média (ALIm)	0	0
Alta (ALla)	0	0
IE		
Baixa (IEb)	10	30
Média (IEm)	6	24
Alta (IEa)	4	24
Total		176

NPF

REFERÊNCIAS

AMARAL, L. G. **CSS Cascading stylesheets: guia de consulta rápida**. São Paulo: Novatec, 2006.

GARRETT, J. **Ajax: Uma nova Abordagem para Aplicação Web**. 2005. Disponível em: <<http://www.adaptivepath.com/publications/essays/archives/000385.php>>. Acesso em: 11 fev. 2013

FLANAGAN, David. **JavaScript: O guia definitivo**. 4. ed. Porto Alegre: Artmed, 2002.

Forlogic Software. **Procedimentos do Processo 4L**. Versão 2.2. Cornélio Procópio, 2011.

Forlogic Software. **Guia do Ciclo de Vida**. Versão 2.2. Cornélio Procópio, 2011.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995.

MENDES, Antonio. **Arquitetura de Software: Desenvolvimento orientado para arquitetura**. Rio de Janeiro: Campus, 2002.

COHN, MIKE. **Agile Succeeds Three Times More Often Than Waterfall**. 2013. Disponível em <http://www.mountangoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall>. Acesso em: 06 março 2013.

