

SISTEMA PARA GERENCIAMENTO DE PROJETOS DE SOFTWARE

Carlos Alberto Dutra Mendes¹
Welber Castilho Pereira²
Rafael Rodrigues de Souza³

RESUMO

Este artigo tem como principal objetivo apresentar uma ferramenta web que auxilia no gerenciamento de projetos de software. Problemas relacionados ao prazo de entrega dos projetos, falta de comunicação, objetivos mal definidos e controle ineficiente, ainda hoje, assombram muitas empresas de pequeno e médio porte que trabalham com o desenvolvimento de software. A fim de minimizar estes problemas, aumentar a integração entre as partes envolvidas e disponibilizar informação para a tomada de decisão, foi desenvolvido o Sistema para Gerenciamento de Projetos de Software (SGPS). Em seu desenvolvimento foi utilizado a metodologia de programação em camadas juntamente com stored procedure em SQL Server 2005 e linguagem C# .Net.

Palavras-chave: Gerenciamento de projetos. Programação em camadas.

1 INTRODUÇÃO

Com a evolução da tecnologia os computadores apresentam um papel central e de grande importância no comércio, na indústria, na educação, no governo, na medicina, e na sociedade em geral.

Em um mundo onde os sistemas de software estão em todos os lugares, a demanda por este produto é grande e sua produção, muitas vezes, realizada sem levar em conta a importância do gerenciamento das informações e a escolha de ferramentas adequadas para acompanhar o seu desenvolvimento.

¹ Aluno do Curso de Ciência da Computação do Centro Universitário do Sul de Minas UNIS-MG. E-mail: solracmendes@yahoo.com.br

² Aluno do Curso de Ciência da Computação do Centro Universitário do Sul de Minas UNIS-MG. E-mail: welber.castilho@yahoo.com.br

³ Professor Ms. do Centro Universitário do Sul de Minas UNIS-MG. E-mail: rafaelvga@yahoo.com.br

A necessidade do gerenciamento de software se dá ao fato de que a engenharia deste produto está sempre sujeita a restrições de orçamento e de prazo. Sendo assim, o gerenciamento distingue o desenvolvimento profissional e a programação em nível amador.

Com base nas dificuldades em se gerenciar os projetos de desenvolvimento de software e a crescente utilização de ferramentas que auxiliam as empresas nesse sentido, como mostra a figura 1, foi desenvolvido o sistema SGPS, o qual se trata este artigo.

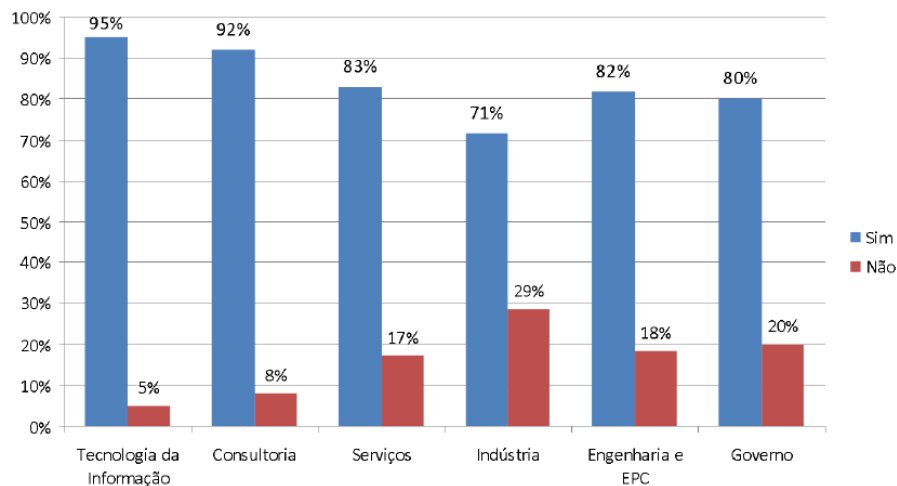


Figura 1: Nível de utilização de Softwares de Gerenciamento de Projetos.

Outra motivação para o desenvolvimento deste trabalho consiste em aprofundar os conhecimentos na modelagem de software utilizando UML (*Unified Modelling Language*), no uso da plataforma Microsoft .Net e SQL Server e na programação em camadas.

2 REFERENCIAL TEÓRICO

Esta seção apresenta os principais aspectos teóricos que fundamentam o trabalho, e propiciaram elaborar o sistema desenvolvido.

2.1 Gerenciamento de Projetos

Muitas das atividades desempenhadas em uma empresa precisam ser planejadas e gerenciadas. Segundo a ABNT(2000), um projeto pode ser considerado como uma iniciativa única seja esta para o desenvolvimento de um produto ou serviço específico consistindo de um grupo de atividades com data para início e término com objetivos e requisitos específicos incluindo limitações de tempo, custo e recursos.

De acordo com Prado (2004), gerenciar um projeto significa fazer o planejamento da sua execução antes de iniciá-lo e acompanhar a sua execução. Para garantir que um projeto

tenha sucesso é necessário a aplicação de técnicas e conhecimentos que irão fazer com que seja cumprido tudo o que foi planejado.

2.2 Gerenciamento de Projetos de Software

A famosa crise do software, conhecida pelo fracasso de muitos projetos de software, na década de 60 e início da década de 70, foi à primeira indicação das dificuldades de gerenciamentos de projetos de software. As principais causas desta crise estão ligadas a relativa imaturidade da engenharia de software e a complexidade do processo de software, e se manifestam através dos seguintes problemas:

- a) Software entregue fora do prazo;
- b) Software de baixa qualidade;
- c) Software muitas vezes não atinge os requisitos;
- d) Código difícil de manter;

O custo final do software é varias vezes mais do que previam as estimativas. Segundo Sommerville(2003), o fracasso dos projetos de software não estavam relacionados a incompetência dos programadores ou do gerente de projetos, mas sim as técnicas de gerenciamento provenientes de outras disciplinas de engenharias que eram aplicadas e mostravam-se ineficazes para o desenvolvimento de software.

2.3 UML

UML é o acrônimo de *Unified Modeling Language*, que segundo Medeiros (2008) é uma ferramenta que serve para o auxílio na modelagem de sistemas e que possui a finalidade de proporcionar um padrão para preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais, como processos de negócios, itens concretos, como as classes escritas em qualquer linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis.

2.4 ASP.Net

De acordo com Lotar (2007), o ASP.NET é considerado pela documentação do .NET Framework um modelo de desenvolvimento unificado que inclui diversos serviços necessários para o desenvolvimento de aplicações web com o mínimo de código.

Como principais benefícios do ASP.NET podemos citar:

O código da aplicação ASP.NET é compilado: Quando uma página ASP.NET é requisitada ela é compilada e vai para o cachê do servidor aumentando a performance e sendo assim carregada mais rápida.

Páginas ASP.NET são construídas com controles de interface do lado do servidor: O ASP.NET utiliza web Server controls para construir layouts complexos e adicionar funcionalidades diversas com o mínimo de esforço.

ASP.NET é parte do .NETFramework: O .NET framework possui mais de 3000 classes que podem ser utilizadas para aplicações ASP.NET. (O que é então o ASP.NET, 2010).

ASP.NET é totalmente orientado a objeto.

2.5 Linguagem C#

Segundo Galuppo (2004), C# é uma nova linguagem de programação da plataforma .NET, derivada de C/C++ possui o mesmo poder das duas, só que aliado a alta produtividade do Visual Basic.

C# é simples, moderna, orientada a objetos, fortemente tipada e foi desenhada para facilitar a criação de componentes, cujos conceitos, como propriedades, métodos, eventos e atributos, são fortemente aplicados.

2.6 Desenvolvimento em Camadas

No início da computação era comum encontrar sistemas monolíticos onde um aplicativo era executado em uma única máquina contendo todas suas funcionalidades em uma única grande camada, onde sua atualização e manutenção eram complexas e trabalhosas.

Com o passar do tempo a arquitetura monolítica evoluiu para uma arquitetura cliente/servidor de duas camadas. Com o objetivo de se manter uma única base de dados para diversos aplicativos, a lógica de acesso de dados estava separada do restante da aplicação, porém a interface do sistema (apresentação) e os algoritmos do sistema (lógica de negócios) ainda estavam reunidos em uma única camada. (DESENVOLVIMENTO, 2010).

Perante a evolução do mercado e as constantes mudanças da legislação, o modelo de duas camadas demonstrou-se de difícil manutenção e gerenciamento onde em função da

necessidade de constantes alterações nas regras de negócio era gerado um custo muito elevado de manutenção (CRIANDO, 2010).

Enquanto a arquitetura cliente/servidor ganhava popularidade começava a surgir à orientação a objetos que possuía a resposta para o problema encontrado com o desenvolvimento utilizando duas camadas.

Com o surgimento da internet todos queriam implantar um sistema com acesso por navegador, mas como toda lógica estava em um sistema local seria necessário refazer o sistema para uma interface web (DESENVOLVIMENTO, 2010).

Como uma evolução do modelo de 2 camadas surge o modelo de 3 camadas que envolve a separação das funcionalidades usando camadas, com o objetivo de separar a lógica de apresentação, a lógica de negócio e a conexão ao banco de dados, tornando o sistema mais flexível de modo em que partes podem ser alteradas independentemente, onde uma alteração em uma camada não influi nas demais.

O modelo de 3 camadas é formado por:

- a) Camada de Apresentação: Responsável pela interação entre o usuário e o software. Exibe a informação para o usuário e interpreta os comandos emitidos pelo usuário em ações para as camadas de negócio. Observação: Alterações de interface são menos frequentes do que alterações na regras de negócio.
- b) Camada de Negócio: São as regras de negócio. Determinam de que maneira os dados serão utilizados. Envolve cálculos baseados em dados digitados e em informações armazenadas, validações de informações vindas da camada de apresentação.
- c) Camada de Dados: Responsável por acessar a base de dados recuperando e atualizando os dados necessários.

Mesmo com o modelo de 3 camadas ainda continua o problema referente a atualização da aplicação sempre que for necessário realizar alguma alteração na interface.

Surge então um novo modelo que é visto como uma evolução do modelo de 3 camadas. Este modelo possui como idéia a retirada da camada de apresentação direto do cliente e centralizá-la em um determinado ponto, na maioria das vezes um servidor Web.

7 Banco de Dados

De acordo com Silberschatz (2006), um sistema de gerenciamento de banco de dados consiste em uma coleção de dados inter-relacionados e uma coleção de programas para acessar estes dados. O principal objetivo de um sistema de banco de dados é possibilitar um

ambiente que seja adequado e eficiente para uso na recuperação e armazenamento de informações.

Projetados para gerenciar grandes blocos de informação, os sistemas de banco de dados garantem a segurança das informações armazenadas, apesar de tentativas de acesso não autorizadas ou de falhas de sistema.

2.8 Stored Procedures

Stored Procedures consistem de um conjunto de instruções armazenadas e executadas diretamente no banco de dados. Segundo Lotar (2007), sua grande vantagem está na facilidade em dar manutenção, por estar armazenada no banco de dados uma única alteração pode afetar imediatamente todos os usuários. Outra vantagem está no rápido acesso, já que uma vez executada uma stored procedure seu plano de execução fica na memória.

3 DIAGRAMA DE CASO DE USO

O primeiro passo foi à elaboração do diagrama de caso de uso para a identificação das funcionalidades esperadas pelo sistema. Neste diagrama são demonstradas as funcionalidades que os atores (usuários) podem ativar pela interface do sistema (Figura 2).

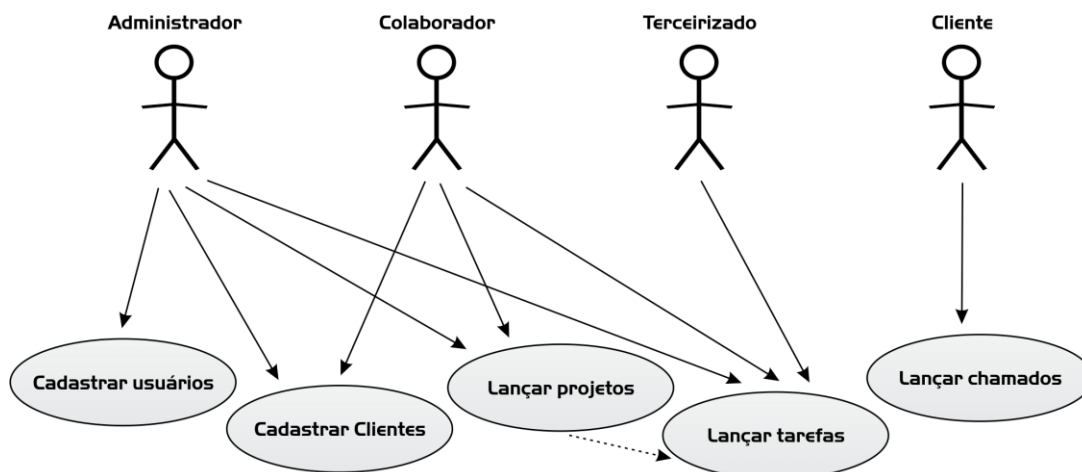


Figura 2: Diagrama de caso de uso do sistema

Foram identificados quatro atores principais:

Administrador: Responsável por gerenciar a equipe e seus usuários.

Colaborador: Desenvolve projetos e tarefas.

Terceirizado: Desenvolve projetos e tarefas específicas.

Cliente: Interage a fim de agilizar o feedback.

Tabela 1 - Descrição do caso de uso	
Caso de Uso	Descrição
Cadastrar Usuários	Possibilita efetuar o cadastra de usuários.
Cadastrar Clientes	Possibilita efetuar o cadastro de clientes.
Lancar Proieto	Possibilita efetuar o cadastro das informações mais importantes referentes ao projeto.
Lançar Tarefa	Possibilita cadastrar as informações referentes à tarefa.
Lançar Chamados	Possibilita que o cliente possa solicitar a alteração de um projeto relacionado a ele ou a criação de um novo.

4 APLICAÇÃO EM CAMADAS

O sistema é composto por três camadas, a camada Web (camada de interface com o usuário), a camada de regras de negócio denominada de BLL e a camada de acesso aos dados conhecida como DAL. A comunicação entre as camadas é realizada pelas classes do projeto denominado Modelo.

Foram criadas as classes libraries referentes a cada camada citada anteriormente com exceção da camada Web, pois nesta ficam as páginas do sistema. Dentro de cada classe *library* foi criada uma classe para cada entidade do banco de dados.

As classes da camada DAL se comunicam com as stored procedures armazenadas no banco de dados onde são realizadas as inserções, seleções, alterações e exclusões.

A Figura 3 mostra a interação das camadas ao realizar o cadastro de um usuário.

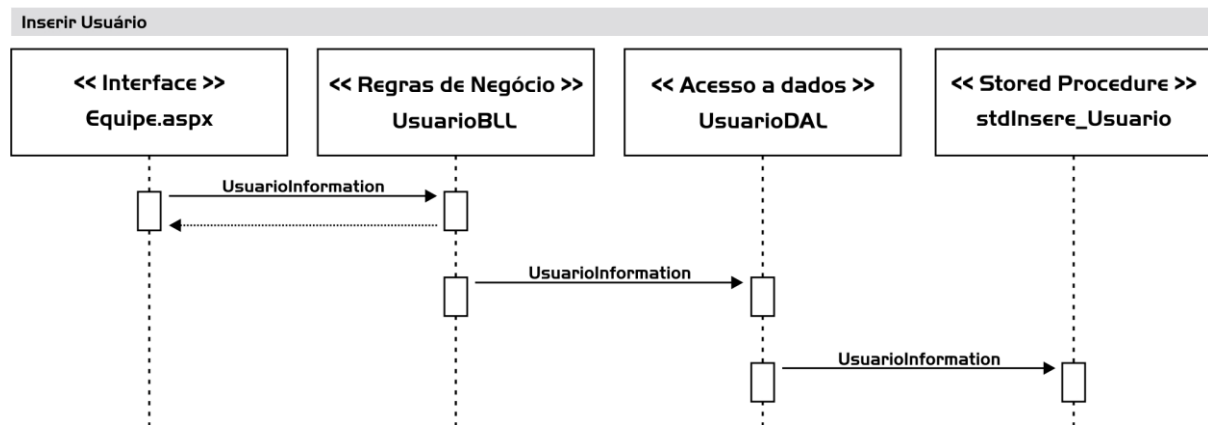


Figura 3: Diagrama de sequência para inserção de usuário

5 O SISTEMA

O sistema foi desenvolvido em linguagem C# .Net devido a grande aceitação no mercado e sua fácil utilização. Definida a linguagem de programação, a metodologia em camadas foi adotada para dar mais flexibilidade na produção e manutenção do código fonte. Ao utilizar este tipo de metodologia o tempo de desenvolvimento prático é maior se comparado a programação convencional, mas os benefícios de uma programação estruturada são altamente relevantes em grandes projetos. Portanto, o tempo demandado na programação é compensado pela facilidade de futuras alterações e baixo custo de manutenção.

Para deixar o processo de comunicação entre cliente e servidor mais rápido e seguro foi feito uso de stored procedures. Elas também facilitam a manutenção sendo armazenadas no banco de dados, assim, uma alteração afeta todos os usuários de uma vez.

Com todas essas técnicas e métodos de programação o sistema possui um grande potencial para atender todos os seus objetivos de forma rápida e eficiente.

Ainda complementando o sistema, os conceitos da Interface Homem Máquina foram bastante utilizados a fim de disponibilizar todos os recursos de forma amigável facilitando ao máximo a usabilidade. Tendo em vista que o sistema foi projetado para uso contínuo, a relação entre disposição dos objetos, símbolos e cores nas páginas foram trabalhadas para fornecer uma melhor e intuitiva navegabilidade, diminuindo o desgaste do usuário.

Fatores como cores utilizadas influenciam psicologicamente no rendimento do usuário. Segundo Farina (2002), as cores transmitem estados e sensações, induzem uma pessoa. Tons claros de azul e cinza, por exemplo, atraem a atenção de profissionais adultos entre 20 e 40 anos.

5.1 Principais funcionalidades do sistema

O sistema auxilia em todo gerenciamento de projetos de software, sendo estes desenvolvidos em plataforma web ou desktop. O SGPS é acessado via web e possibilita tanto o cadastro de equipes corporativas quanto o de grupos de trabalho autônomo. Ao cadastrar uma equipe, conseqüentemente o usuário também se cadastra adquirindo privilégios de administrador.

O SGPS permite o cadastro de várias equipes sem qualquer relação entre as mesmas, onde o usuário só tem acesso à equipe em que é cadastrado.

Não há um limite para cadastros de membros na equipe. No cadastro de um novo usuário o administrador define seu nível e lhe fornece um login e senha para acesso.

Quando o usuário lança um novo projeto ele se torna responsável em gerenciar todo seu desenvolvimento, cadastrando informações bem definidas como principais datas, escopo e associando usuários a fim de formar uma sub-equipe responsável pela execução das tarefas.

Em um projeto podem ser lançadas várias tarefas, de acordo com as suas necessidades. Cada tarefa possui, assim como o projeto, as informações necessárias para o seu desenvolvimento, além de possibilitar a interação dos usuários associados (sub-equipe) através de mensagens e logs informando a sua situação.

O sistema possui páginas específicas onde podem ser vistas várias informações detalhadas referente à equipe, tais como: todos os membros cadastrados; projetos; tarefas; clientes; chamados; histórico de tudo que foi gerenciado.

Na sequência deste artigo as funcionalidades do SGPS serão descritas detalhadamente por partes, a fim de explanar melhor todo potencial que esta ferramenta possui para o auxílio na gerência de projetos de software.

5.1.1 Usuários

O sistema trabalha com quatro níveis de usuário, sendo eles: Administrador, Colaborador, Terceirizado e Cliente. Cada nível possui permissões de acesso diferentes a fim de facilitar a gerência através da hierarquia.

Sendo assim o Administrador é responsável pelos principais cadastros, liberação acompanhamento e finalização de projetos, exclusões e alterações que se fazem necessárias. Os administradores sempre estarão informados dos principais fatos através de logs gerados dinamicamente pelo sistema, sempre que aconteça alguma interação relevante.

O Colaborador integra a equipe com a finalidade de desenvolver projetos, tarefas e atender a chamados. Com restrições de acesso mais rígidas, o Terceirizado tem como objetivo contribuir somente para o desenvolvimento do(s) projeto(s) ao qual foi associado.

Já o Cliente, ao ser cadastrado no sistema, é relacionado a um projeto para que possa acompanhar o seu desenvolvimento e lançar chamados. A área do cliente permiti acesso apenas as suas informações pessoais e as dos projetos que está envolvido.

Todo usuário cadastrado no SGPS recebe, por e-mail, o login e senha de acesso.

5.1.2 Equipe

Na página Equipe se têm toda relação de usuários cadastrados, suas informações pessoais e níveis de acesso. Também é possível enviar mensagens para qualquer usuário da equipe através dessa página.

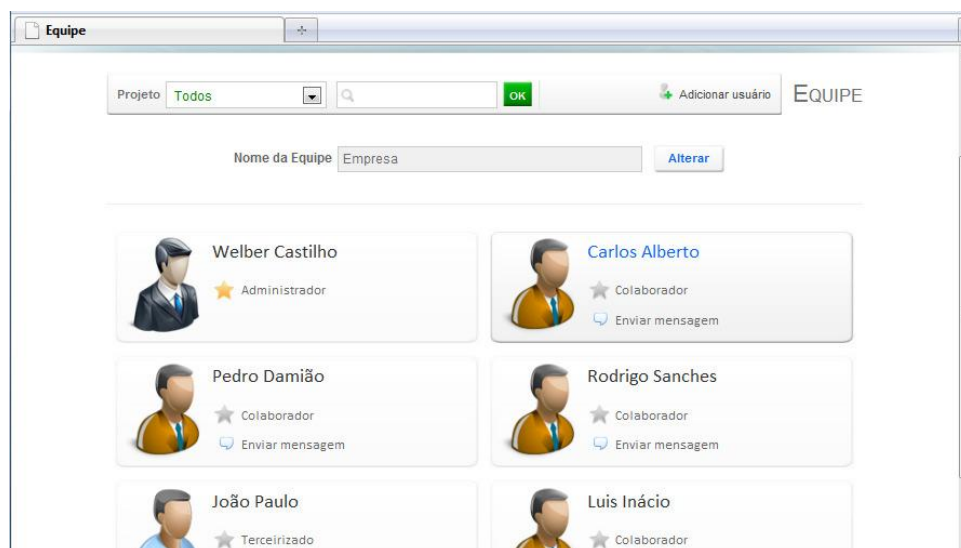


Figura 4: Página Equipe

5.1.3 Projetos

O lançamento de projetos é restrito apenas a usuários Administradores ou Colaboradores. No cadastro das informações é fundamental que o usuário responsável já tenha tudo definido para não comprometer o desenvolvimento do projeto. Informações como data prevista para término, escopo e ferramentas a serem utilizadas influenciam diretamente no progresso das tarefas.

Ainda no cadastro, é definido quais usuários da equipe devem ser associados ao projeto. Esses formam uma sub-equipe. Assim, somente usuários dessa sub-equipe podem colaborar no desenvolvimento do projeto.

É importante ressaltar que após o cadastro de um projeto é necessário a aprovação de um Administrador para que o mesmo tenha continuidade. E quando todas as tarefas do projeto forem finalizadas o seu responsável deve fechá-lo, para que posteriormente um Administrador possa finalizar o seu desenvolvimento.

Na página Projetos é possível visualizar várias informações como: cadastros ativos, finalizados e inativos; cadastros referentes ao usuário que está logado; porcentagem e barra de progresso em relação ao desenvolvimento, entre outros detalhes.

5.1.4 Tarefas

Para que o usuário possa lançar tarefa em um projeto é necessário que ele esteja associado ao mesmo. Toda tarefa deve ter uma descrição objetiva, data prevista para término, um usuário criador (usuário responsável pela criação da tarefa), um usuário executor (usuário responsável pela execução da tarefa), uma prioridade e arquivos em anexo se necessário.

A tarefa só pode ser fechada pelo usuário executor e finalizada pelo usuário criador.

Os demais usuários que pertencem à equipe, mas não estão associados ao projeto (sub-equipe) ao qual a tarefa está associada, não podem alterar a situação da mesma.

Para o desenvolvimento da tarefa, existe um ambiente para interações contendo suas principais informações e que fornece a possibilidade de enviar mensagens, mudar a situação do desenvolvimento, adicionar um usuário da equipe para acompanhar o seu progresso ou até mesmo encaminhar a tarefa para outro usuário da sub-equipe.

Para facilitar a gerência, como na figura 5, as tarefas possuem símbolos intuitivos que informam sua situação de desenvolvimento, nível de prioridade e interações recentes.

The screenshot displays the 'Tarefas' (Tasks) page. At the top, there is a filter bar with the following status counts: Novas (13), Abertas (12), Em Produção (0), Fechadas (0), and Finalizadas (0). Below this, the user profile for Pedro Damião is shown, including a profile picture, name, role (Colaborador), and performance metrics: Taxa de execução: 0%, Tarefas executadas: 0 / 13, and Tarefas atrasadas: 8. A table of tasks is listed below, with columns for Projeto, Tarefa, Criado Por, Previsão, and Última interação. The tasks are as follows:

Projeto	Tarefa	Criado Por	Previsão	Última interação
Projeto 2	Tarefa 1	Welber	12/01/2011	05/12/2010
Projeto 2	Tarefa 3	Welber	12/01/2011	05/12/2010
Projeto 2	Tarefa teste anexo	Welber	12/11/2010	05/12/2010
Projeto 2	Teste anexo 2	Welber	12/11/2010	05/12/2010
Projeto 2	Anexo 3	Welber	12/11/2010	05/12/2010
Projeto 2	Anexo Teste 1	Welber	12/11/2010	05/12/2010

Figura 5: Página Tarefas

5.1.5 Histórico

O histórico possui todos os registros de projetos, tarefas e chamados que já foram finalizados. Ao finalizar qualquer um desses itens, após um dia eles só podem ser visualizados através da página Histórico. Na página, a busca pelo que já foi realizado é facilitada por meio de vários filtros. Todos os detalhes podem ser visualizados, mas não podem ser alterados.

5.2 Regras de Negócio do SGPS

O sistema é composto por várias regras de negócio a fim de organizar todo o conteúdo cadastrado e facilitar a gerência. Essas regras foram elaboradas com base nos conceitos de órgãos como o PMBOK e PMI, sobre gerência de projetos de software. A seguir é listado todas as regras presentes no SGPS.

5.2.1 Permissões restritas ao usuário Administrador

- Cadastrar Administradores, Colaboradores ou Terceirizados;
- Alterar o nome da equipe;
- Excluir ou alterar a situação (ativo ou inativo) de usuários;
- Permitir o início, alterar a situação, finalizar, ou excluir um projeto;
- Cadastrar ou excluir opções de ferramentas para desenvolvimento;
- Interagir em qualquer tarefa.

5.2.2 Permissões em comum entre usuários Administradores e Colaboradores

- Lançar projetos;
- Lançar tarefas;
- Cadastrar clientes;
- Atender chamados;
- Enviar mensagens para outros usuários da equipe;
- Alterar a situação da tarefa;
- Visualizar ou alterar seus dados pessoais.

5.2.3 Permissões únicas do usuário Terceirizado

- Visualizar ou seus dados pessoais;
- Lançar tarefas no projeto que está associado;
- Interagir em qualquer tarefa do projeto que está associado;
- Visualizar informações dos membros da sub-equipe em que está associado;
- Enviar mensagens para usuários da sub-equipe em que está associado;
- Visualizar informações do projeto que está associado.

5.2.4 Permissões únicas do usuário Cliente

- Visualizar ou alterar seus dados pessoais;
- Lançar chamados nos projetos que está associado;
- Lançar chamados para criação de um novo projeto;
- Visualizar informações dos projetos que está associado.

5.2.5 Regras de negócio aplicadas aos Projetos

Projetos possuem quatro situações de desenvolvimento: Aberto, Produção, Fechado e Finalizado. Essas situações são alteradas da seguinte forma:

- Aberto: ao ser criado;
- Aberto → Produção: pelo sistema, quando uma ou mais tarefas são lançadas;
- Produção → Fechado: pelo sistema, quando todas as tarefas estão finalizadas;
- Fechado → Finalizado: por um usuário administrador;
- Fechado → Produção: por um usuário administrador;

5.2.6 Regras de negócio aplicadas as Tarefas

Tarefas possuem cinco situações de desenvolvimento: Nova, Aberta, Produção, Fechada e Finalizada. Essas situações são alteradas da seguinte forma:

- Nova → Aberta: pelo usuário executor da tarefa, ao acessá-la a primeira vez;
- Aberta → Produção: pelo usuário executor;
- Produção → Fechada: pelo usuário executor;
- Fechada → Finalizada: pelo usuário criador da tarefa;
- Fechada → Aberta: pelo usuário criador.

As tarefas podem ser encaminhadas ou ter usuários adicionados para acompanhá-las:

A tarefa só pode ser encaminhada para usuários da sub-equipe do projeto;

Qualquer usuário da equipe pode ser adicionado para acompanhar a tarefa;

Usuários que estão acompanhando só podem interagir com mensagens.

6 CONCLUSÃO

Este trabalho permitiu além da criação de uma ferramenta que auxilia equipes de TI, vivenciar as dificuldades na gerência de um projeto de software. A oportunidade de desenvolver o sistema possibilitou colocar em prática a utilização de ferramentas, metodologias e conhecimentos que foram passados ao longo do curso.

Experiências adquiridas com as tecnologias utilizadas contribuíram para o crescimento acadêmico e profissional.

Como trabalhos futuros pretende-se implementar uma área para relatórios, fazer o gerenciamento financeiro dos projetos e enriquecer o sistema com gráficos melhorando a visualização de estatísticas e metas.

SYSTEM FOR PROJECT MANAGEMENT SOFTWARE

ABSTRACT

This article has as main objective to present a Web tool that assists in managing software projects. Problems related to the delivery of projects, poor communication, poorly defined goals and inefficient control, still haunts many small and medium-sized businesses that work with software development. To minimize these problems by increasing integration between the parties involved and provide input into the decision-making system was developed for Project Management Software (SGPS). In its development methodology was used to programming in layers along with stored procedure in SQL Server 2005 and C#. Net

Keywords: *Project managemen. Programming in layers.*

REFERÊNCIAS

ABNT. Gestão de qualidade – Diretrizes para a Qualidade no Gerenciamento de Projetos, 2000.

GALUPPO, Fabio. Desenvolvimento com C#. Editora Bookman, 2004.

LIMA, Edwin. C# e .NET - Guia do Desenvolvedor. Rio de Janeiro: Campus, 2002.

LOTAR, Alfredo. Como Programar com ASP.NET e C#. Editora Novatec, 2007.

MEDEIROS, Ernani. Desenvolvendo Software com UML 2.0, 2008.

PRADO, Darci. Planejamento e Controle de Projetos. INDG Tecnologia e Serviços Ltda., 2004.

SILBERSCHATZ, Abraham. Sistema de Banco de Dados. Editora Campus, 2006.

SOMMERVILLE, Ian. Engenharia de Software. Editora Addison Wesley, 2003.

O que é então o ASP.NET, 2010, Disponível em:

<[HTTP://www.macoratti.net/aspnet_1.htm](http://www.macoratti.net/aspnet_1.htm)>. Acessado em 01 de Agosto de 2010.

Desenvolvimento de sistema web utilizando arquitetura em três camadas e applets, 2004, Disponível em: <[HTTP:// http://inf.unisul.br/~ines/workcomp/cd/pdfs/2905.pdf](http://inf.unisul.br/~ines/workcomp/cd/pdfs/2905.pdf)>.

Acessado em 15 de Agosto de 2010.

Criando aplicação em 3, 4 ou N camadas, Disponível em:

<[HTTP://www.juliobattisti.com.br/artigos/ti/ncamadas.asp](http://www.juliobattisti.com.br/artigos/ti/ncamadas.asp)>.

Acessado em 20 de julho de 2010.

Desenvolvimento em camadas, 2010, Disponível em:

<[HTTP://www.microsoft.com/brasil/msdn/tecnologias/arquitetura/Layers_Developing.msp](http://www.microsoft.com/brasil/msdn/tecnologias/arquitetura/Layers_Developing.msp)>

. Acessado em 15 de Agosto de 2010.