

## **ESTRATÉGIAS PARA UTILIZAÇÃO DAS TECNOLOGIAS DO *GOOGLE CODE* NO GERENCIAMENTO ÁGIL DE PROJETO.**

### **André Montoia Barata**

Bacharel em informática no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo – ICMC/USP

E-mail: [andremontoiab@gmail.com](mailto:andremontoiab@gmail.com) (Brasil)

### **Camila de Araujo**

Doutoranda em Engenharia de Produção na Escola de Engenharia de São Carlos da Universidade de São Paulo – EESC/USP

E-mail: [camiladearaujo@gmail.com](mailto:camiladearaujo@gmail.com) (Brasil)

### **Guilherme Delefrate Martins**

Mestrando em Engenharia de Produção na Escola de Engenharia de São Carlos da Universidade de São Paulo – EESC/USP

E-mail: [guilhermem@gmail.com](mailto:guilhermem@gmail.com) (Brasil)

### **Daniel Capaldo Amaral**

Doutor em Engenharia Mecânica pela Universidade de São Paulo – USP

Professor de Engenharia de Produção na Escola de Engenharia de São Carlos da Universidade de São Paulo – EESC/USP

E-mail: [amaral@sc.usp.br](mailto:amaral@sc.usp.br) (Brasil)

## ESTRATÉGIAS PARA UTILIZAÇÃO DAS TECNOLOGIAS DO *GOOGLE CODE* NO GERENCIAMENTO ÁGIL DE PROJETO.

### RESUMO

O gerenciamento de projetos (GP) de produtos inovadores é caracterizado por incertezas e colaboração entre diferentes equipes. Há críticas sobre a adequação dos softwares atuais de GP e uma solução possível é explorar o conceito de *Cloud Computing*. Um exemplo são os aplicativos *Google Spreadsheet* e *Google Calendar*. O tema é investigado com a avaliação de estratégias de integração de dados de programação de projeto, entre um aplicativo especialmente desenvolvido pelos pesquisadores, programado em JAVA, e os aplicativos do Google, com o uso de APIs do *Google Code*. O artigo descreve os recursos dos aplicativos *Google Gadget*, *Google Spreadsheet*, *Google Apps Script* e *Google Calendar*, identifica possíveis estratégias de integração para uma solução de GP, descreve os resultados de um protótipo realizado e discute potenciais futuros para a aplicação. Ilustra e compara essas tecnologias e descreve o potencial para a criação de soluções de um aplicativo de gerenciamento ágil de projetos. Como resultado final, mostra a melhor estratégia que seria a utilização do *Google Calendar* em interação com uma ferramenta de GP, e indica pesquisas futuras para a criação de uma solução completa baseada no tema.

**Palavras-chave:** *Software* para Gerenciamento de Projetos; Gerenciamento de Projetos; *Google Code*.

## STRATEGIES FOR UTILIZATION GOOGLE CODE TECHNOLOGY IN AGILE PROJECT MANAGEMENT

### ABSTRACT

Project management (PM) of innovation products is characterized by uncertainty and collaboration between different teams. There are discussions about the utilization of current PM software and a possible solution is to explore the concept of *Cloud Computing*. For example are *Google Spreadsheet* and *Google Calendar*. The theme is investigated with strategies evaluation dates of programming project, between an application in JAVA developing by researches and Google application using the *Google Codes API's*. The article describes the applications *Google Gadgets*, *Google Spreadsheet*, *Google Calendar* and *Google Apps Script*, identifying the possible strategies for integration with a PM solution. The results are proven to developing prototype and discussion about the new features for future application. The article illustrates and compares the Google technologies describing the potential for a development a agile project management application. The final results show that the best strategies would be to use *Google Calendar* interaction with a PM tool and indicates future research to create a complete solution based on the theme.

**Keywords:** Project Management Software; Management Project and Google Code.

---

## 1 INTRODUÇÃO

A adoção de softwares para o gerenciamento de projetos (GP) de desenvolvimento de novos produtos é fundamental para o desempenho, eles permitem a consolidação das informações e, conseqüentemente, podem apoiar as decisões de forma a se obter melhores resultados, qualidade e redução de custos de desenvolvimento.

Os produtos inovadores estão inseridos em ambientes caracterizados pela dificuldade em prever o futuro, o alto risco e a necessidade de interação com outras organizações visando o acesso a competências complementares e a projetos colaborativos. Viabilizar as práticas de GP dentro desse contexto exige o uso de ferramentas computacionais flexíveis, isto é, capazes de apoiar as constantes mudanças e interações, e garantir a confiabilidade das informações.

Os dados empíricos existentes demonstram problemas na utilização de softwares para GP (SGP). Há limitações nas soluções disponíveis. Um exemplo é o trabalho de White e Fortune (2002), uma pesquisa tipo *survey* que sondou quase uma centena de empresas no Reino Unido, sobre as dificuldades na área de gerenciamento de projetos em empresas inovadoras. Os SGPs aparecem como uma das principais limitações entre os métodos/ferramentas/técnicas de gerenciamento de projetos. Foram caracterizados como inadequados, principalmente, para projetos complexos, onde várias organizações participam de seu desenvolvimento.

Ren *et al.* (2006) afirmam que uma nova geração de ferramentas de planejamento de projetos colaborativos precisa ser desenvolvida, com base na experiência de proposição de uma solução específica para GPs colaborativos e empresas que tratem de serviços de engenharia. Segundo Woerner e Woern (2005), a maioria das plataformas de colaboração possui arquitetura centralizada e esse é um aspecto inadequado quando se trata de cooperação em pequenas e médias empresas.

Uma necessidade identificada por Araujo (2008), para as ferramentas computacionais de apoio ao GP é o uso de plataformas heterogêneas, que promovam a descentralização da informação e permitam que diferentes equipes possam utilizar sistemas distintos, conforme sua conveniência.

A *Cloud Computing* (Computação nas nuvens) é um modelo de computação em que as aplicações e seus dados são acessíveis por meio de uma rede, como por exemplo, a internet (Wikipedia, 2010). São aplicações que propiciam o acesso às informações do projeto por diferentes equipes, independente da sua localização física ou de um SGP com uma plataforma específica.

Um exemplo de tecnologia que permite a operacionalização do conceito é a tecnologia do Google. Ela é formada pelo conjunto de aplicativos Google Docs, Google Gadget e iGoogle. Trata-se de uma plataforma formada por aplicativos com APIs (Application Programming Interface) abertas, serviços gratuitos fornecidos aos usuários e ferramentas de desenvolvimento (*frameworks* e objetos disponíveis na internet). Esses elementos permitem criar aplicativos seguindo o conceito de computação nas nuvens e disponíveis mundialmente. É uma tecnologia recente e que ainda não foi explorada em termos de pesquisa científica na área de GP. O levantamento realizado não identificou qualquer trabalho similar.

Os softwares tradicionais, denominados *standalone*, são os mais difundidos na área de GP. Há uma grande variedade de produtos com funcionalidades sofisticadas, desenvolvidas ao longo dos anos. As primeiras ferramentas para esse fim datam da década de 1980 e são conhecidas dos profissionais de GP. Contudo, conforme apresentado anteriormente, o uso, apresenta deficiências. Além disso, necessitam de instalação no computador em que serão utilizados, o que restringe o acesso aos dados do projeto pelos diferentes membros de uma mesma equipe.

Recriar uma solução completa do tipo *cloud computing* representaria um esforço excessivo. Mas, seria possível compartilhar dados de softwares tradicionais utilizando-se do conceito de computação das nuvens? Uma possibilidade é a integração entre aplicativos computacionais tradicionais com sistemas na web, elaborados a partir da tecnologia Google Code. Esse artigo investiga o assunto.

## 2 OBJETIVO

O objetivo principal é identificar estratégias para integração de dados entre os aplicativos Google e ferramentas tradicionais de GP, de forma a possibilitar uma solução para planejamento e controle de projetos segundo os princípios de *cloud computing*. O resultado indicará meios para se desenvolver uma ferramenta computacional para controle de projetos que permita a descentralização e compartilhamento de informações entre diferentes equipes, para tentar solucionar os problemas identificados na literatura.

### 3 SOFTWARE DE GERENCIAMENTO DE PROJETOS

O GP é uma técnica usada para o controle do projeto, onde seu gerente utiliza de conhecimentos, habilidades de gerência e de coordenação para garantir a qualidade e o sucesso do projeto. Esse modelo é predominante na maioria das empresas que desenvolvem produtos. Para auxiliar no GP, existem alguns conjuntos de técnicas e conceitos que foram criados a partir da década de 1950. Como principais, pode-se citar o gráfico de Gantt, método do caminho crítico, análise de rede (PERT) e a criação das primeiras associações profissionais.

O GP proporciona uma grande demanda de informações, planejamento e tomada de decisões, que precisam ser processadas de forma correta para garantir a qualidade e sucesso do projeto envolvido. Para isso são utilizados os softwares que auxiliam no GP.

Os primeiros softwares, segundo a literatura, que auxiliavam no GP, possuem uma grande defasagem em relação às necessidades dos projetos abordados atualmente. Alguns problemas enfrentados hoje em dia podem ser: falta de informação sobre o que as outras equipes estão desenvolvendo (progresso de tarefas), segundo Hameri e Puittinen (2003); visões diferentes sobre os objetivos, segundo Hameri e Puittinen (2003) e Barnes *et al.* (2000); falta de comunicação efetiva, segundo Barnes *et al.* (2000). Essas são algumas das críticas realizadas à metodologia clássica de GP, como contrapartida a essa metodologia surge o GP ágil, que tem como objetivo suprir essas dificuldades constatadas, segundo o Manifesto Ágil de Projetos (APM) (Beck *et al.*, 2001). O gerenciamento ágil tem o enfoque a colaboração com o cliente e respostas rápidas às mudanças, privilegiando a comunicação e a atualização das informações com um controle regulado de todas as atividades.

Um conjunto de funcionalidades que pode ser constatado nos Softwares para Gestão de Projetos, como citadas por Araujo (2008), são:

- **Gerenciamento de atividades** - registro, visualização e organização das atividades;
- **Gerenciamento de calendário e agenda** - organização e sincronização de calendários e agendas;
- **Gerenciamento de recursos** - gerenciamento de pessoas e materiais para o desenvolvimento;
- **Gerenciamento de custos** - acompanhamento dos gastos;

- **Ferramentas de monitoramento** - funções para acompanhamento, como armazenamento de linhas de base e comparações entre parâmetros de planejamento atual com os parâmetros das linhas de base, bem como análise do valor agregado;
- **Gerenciamento de múltiplos projetos** - análise do portfólio da empresa e compartilhamento dos dados entre diversos projetos.

Nesse estudo usaremos como tema o gerenciamento de atividades, uma funcionalidade de extrema importância para os softwares de gestão de projetos.

Uma tecnologia para auxiliar no gerenciamento ágil de projetos é a do Google *gadgets*. Por ser uma tecnologia baseada em *cloud computing* o acesso às informações do projeto por diferentes equipes e lugares ao mesmo tempo, seria muito mais viável e fácil, proporcionando uma descentralização nas informações do projeto, pois elas se encontrariam todas na web e todas as equipes teriam acesso à informação atualizada independente do software e da plataforma que ela estivesse usando, sem a necessidade de instalação de um software no computador que seria utilizado.

#### **4 CLOUD COMPUTING E GOOGLE CODE**

*Cloud computing*, computação nas nuvens, consiste em um aglomerado de computadores e servidores que estão interligados pela internet. Esses servidores armazenam todos os dados que estão disponíveis na nuvem, onde podem ser acessados de qualquer computador interligado à internet, sem a necessidade de instalação de nenhum software para o acesso a esses dados, que é feito remotamente por intermédio da internet, por essa razão é feita a alusão à nuvem. (Wikipedia, 2010).

Assim, verificou-se a possibilidade de uma aplicação para GP baseada em *cloud computing*, pois as informações dos projetos ficariam nas nuvens, dessa forma, diversas equipes localizadas em meios físicos diferentes teriam acesso à mesma informação, independente da plataforma e do SGP que estivessem utilizando, um simples computador com conexão à internet possibilitará o acesso de todos às informações disponíveis na nuvem.

A tecnologia que desenvolve esse conceito são os aplicativos do Google Code. Entre os que foram estudados nessa pesquisa estão: o Google Docs, iGoogle e Google *Gadget*.

---

## 5 MÉTODO DE PESQUISA

No projeto emprega-se a metodologia da pesquisa-ação, acompanhada de uma extensa revisão bibliográfica sobre o tema. Segundo Thiollent (2002), a pesquisa-ação é recomendada quando o pesquisador precisa explorar novas ideias e criar conhecimento em aspectos práticos.

Para a aplicação computacional baseada em *Cloud Computing* serão utilizadas as ferramentas de desenvolvimento e APIs do Google Spreadsheet (2010d), Google Gadget (2010c), iGoogle (2010) e do Google Calendar (2010b) desenvolvidas e abertas livremente pelo Google. O Google foi escolhido por ser uma plataforma em ascensão, principalmente útil para desenvolvedores distantes da equipe e especialistas que contribuam com a equipe de projeto, por utilizar a tecnologia do Cloud Computing. Já para o SGP do tipo *standalone*, o MS-Project foi escolhido por ser uma das plataformas mais difundidas e ser utilizado em empresas de diferentes tamanhos (Microsoft, 2010). Supõe-se que essas tecnologias tenham sido as mais importantes até as investigações do Grupo de Engenharia Integrada e de Integração (EI<sup>2</sup>) se iniciarem.

Os dados foram coletados a partir de um diário que descreveu o desenvolvimento e os testes da ferramenta que foi desenvolvida. Na sequência, essa ferramenta foi testada com dados de um projeto exemplo, isto é, um projeto real já finalizado de uma empresa escolhida como estudo de caso. Os resultados do cenário foram apresentados a especialistas para verificar sua aplicação. Essa validação teve como intuito fornecer subsídios sobre a aplicabilidade e pontos de melhoria da ferramenta.

A equipe foi formada por pesquisadores, graduandos, pós-graduandos e doutores que atuam na área de GP e desenvolvimentos de produtos. Os dados sobre o gerenciamento de projetos foram coletados por intermédio de vários projetos desenvolvidos pelo grupo e da vasta literatura sobre o assunto; estudou-se a tecnologia do Google - o Google Code e suas possíveis interações para o desenvolvimento de uma aplicação baseada no gerenciamento ágil de projetos.

As atividades realizadas foram:

- a) **Estudo sobre o gerenciamento de projetos** - foi realizado por meio de revisão bibliográfica das funcionalidades e das definições sobre software de GP;
- b) **Estudo dos aplicativos Google** - compreendeu o estudo dos aplicativos Google Gadget, Google Spreadsheet, Google APPS Script e Google Calendar;
- c) **Identificação de estratégias de integração** - com os conhecimentos desenvolvidos nas etapas anteriores, a meta foi elaborar uma especificação das diferentes estratégias

que poderiam ser utilizadas para realizar a integração entre as ferramentas do Google e uma solução tradicional de GP;

- d) **Escolha e implementação de uma das estratégias** - essa atividade compreendeu a escolha de uma das estratégias identificadas na fase anterior.

## 6 DESCRIÇÃO DOS APLICATIVOS GOOGLE

### 6.1 GADGET DO iGOOGLE

O iGoogle consiste em uma página web onde são hospedados seus *gadgets*. Assim, de qualquer máquina que possua uma conexão a internet consegue-se ter acesso aos dados, desde que cada pessoa possua uma conta Google. Um aspecto interessante é que cada usuário, com conta, pode ter acesso aos seus próprios *gadgets*, configurando-os conforme suas próprias conveniências. Isso permite que cada indivíduo crie sua própria área de trabalho, específica para aquela conta Google, e a hospede no iGoogle.

Os *gadgets* são constituídos basicamente por XML e JavaScript, como linguagens web predominantes. O arquivo XML que especifica um *gadget* contém instruções sobre como processá-lo e exibi-lo, ele pode conter todos os dados e códigos e as referências (URLs) para o local onde os outros elementos se encontram. A linguagem JavaScript é usada para adicionar comportamento dinâmico aos *gadgets*.

No estudo aprofundado na API dos *gadgets* para o iGoogle verificou-se que algumas funções possibilitam o acesso a informações de diferentes servidores e páginas Web, de modo a interagir o conteúdo existente nesses servidores com o conteúdo do *gadget*. A primeira possibilidade explorada foi a estratégia de acesso aos dados do projeto em uma planilha do Google *Spreadsheet*, integrando-a ao *gadget*. Se isso fosse possível não seria difícil exportar dados de ferramentas *desktop* para as planilhas, visto que elas conseguem exportar dados no padrão XML.

Ao utilizar as funções de conteúdo remoto, nos *gadgets* do iGoogle, é possível obter informações apenas de um único servidor ou página WEB. Portanto, não é possível enviar informações ao mesmo servidor; por exemplo, com essas funções consegue-se trazer informações de outro site para seu *gadget*, porém não é possível enviar informações do seu *gadget* para esse mesmo site.



É possível obter informações de diferentes tipos, com uso de texto, XML, JSON, FEEDS, que são diferentes formas de obtenção de dados remotamente. Assim, com essa API, há a possibilidade de apenas obter dados de uma planilha, sem conseguir atualizá-la ou enviar-lhe dados.

## 6.2 GOOGLE SPREADSHEET API

Com o estudo da API do Google *Spreadsheet* confirmou-se a possibilidade de uma interação do *gadget* do iGoogle com a planilha, porém, esse *gadget* só consegue recuperar os dados que ela possui. A planilha funciona como uma fonte de dados para o *gadget*, não consegue modificar seu conteúdo. Esse *gadget* pode ser visualizado tanto na própria planilha, como em outra página web, como o iGoogle.

Com a criação de um *Script* em um *gadget* consegue-se acessar a planilha, usando-a como fonte de dados e recuperando seus dados para o *gadget*, o qual pode ser hospedado no iGoogle. Porém o inverso não é possível, ao atualizar o *gadget* a planilha não é atualizada automaticamente, nesse *Script* apenas é possível utilizar a planilha como fonte de dados, não como editor de seu conteúdo pelo *gadget*.

Uma atualização na API do Google *Spreadsheet*, que consiste na integração da API do Google *Spreadsheet* com a API de dados do Google, permite a integração de um aplicativo cliente com o conteúdo do *Spreadsheet* na forma de *feeds* de API de dados. Por meio de funções e bibliotecas específicas da API de dados, consegue-se exibir o conteúdo de uma planilha e uma lista de planilhas de um determinado usuário, editar, atualizar e excluir o conteúdo de uma planilha existente.

Essa API consiste em uma extensão de bibliotecas criadas pelo Google para a API do Google *Spreadsheet*, proporcionando uma maior interação com o conteúdo de planilhas do Google. As bibliotecas utilizadas para interagir com as planilhas estão disponíveis em várias linguagens, como .NET, PHP, Python e Java. Optou-se pelas bibliotecas em Java pelo conhecimento prévio do grupo de pesquisa e facilidade de obtenção de informações. Com essas bibliotecas Java o aplicativo pode ter acesso a planilhas de um determinado usuário que ele deseja, editando seu conteúdo.

Como por exemplo, por uma conta Google de um usuário e a sua senha obtém-se todos os documentos do Google Docs de sua conta. A partir das listas de documentos desse usuário consegue-se escolher um documento, editá-lo, excluir um documento, atualizá-lo e, até, há a possibilidade de criar um novo documento. Para isso é desenvolvido um código através da API de dados do Google *Spreadsheet*, usando a biblioteca cliente em Java para o acesso à informação do

Google Docs. O grande potencial dessa API é que existe a possibilidade de escolher o usuário que se deseja, possuindo a conta Google do usuário e sua senha, consegue-se realizar manipulações nos seus documentos do Google Docs, pelos métodos da biblioteca cliente Java, que irão interagir com a API.

### 6.3 GOOGLE APPS SCRIPT

O Google APS *script* consiste em uma ferramenta disponibilizada pelo Google para realizar uma integração entre as ferramentas do próprio Google. Google APPS consiste em um serviço do Google para uso de domínio próprio em diversos produtos do Google que possibilita uma interação entre várias pessoas que utilizam esses serviços, ferramenta, atualmente, muito empregada em grandes empresas (Google APPs, 2010a).

Nesse serviço existe a possibilidade de se usar o Google Apps *Script*. A possibilidade de criação de *script* para suas próprias planilhas, personalizando-as da melhor maneira de acordo com suas necessidades, por exemplo: “linkar” vários produtos do Google, agendar eventos no Google Calendar por uma planilha de emails, automatizar tarefas, personalizar produtos, entre outros.

Essa funcionalidade de criação de *Scripts* só funciona com uma conta no Google Apps, com um domínio próprio, portanto os *Scripts* para as planilhas serão feitos para um domínio específico, diferentemente da API de dados do Google *Spreadsheet* que tem a possibilidade de escolher o usuário de onde deseja obter os dados, com isso aumenta enormemente a possibilidade de interação entre diversos usuários.

### 6.4 WEB SERVICE dotPROJECT

Uma tentativa também é o acesso ao Web *Service* do dotProject por meio do *gadget* tarefa do iGoogle, para isso é necessário um *gadget* do iGoogle que utiliza uma solicitação SOAP para o acesso, pelas funções da API do Google *gadget* com acesso remoto a um web *service*.

Isso seria uma possibilidade de acesso direto ao banco de dados do dotProject, sem a necessidade do Google *Spreadsheet*, assim, o *gadget* do iGoogle estaria ligado diretamente ao banco de dados. Porém o *gadget* ligado ao Google *Spreadsheet* é muito mais interessante, pois qualquer pessoa com uma conta Google pode ter acesso à planilha online e pode ser compartilhada

com várias pessoas ao mesmo tempo, diferentemente do *gadget* no iGoogle o que somente aquela pagina do iGoogle onde o *gadget* está hospedado terá acesso às informações.

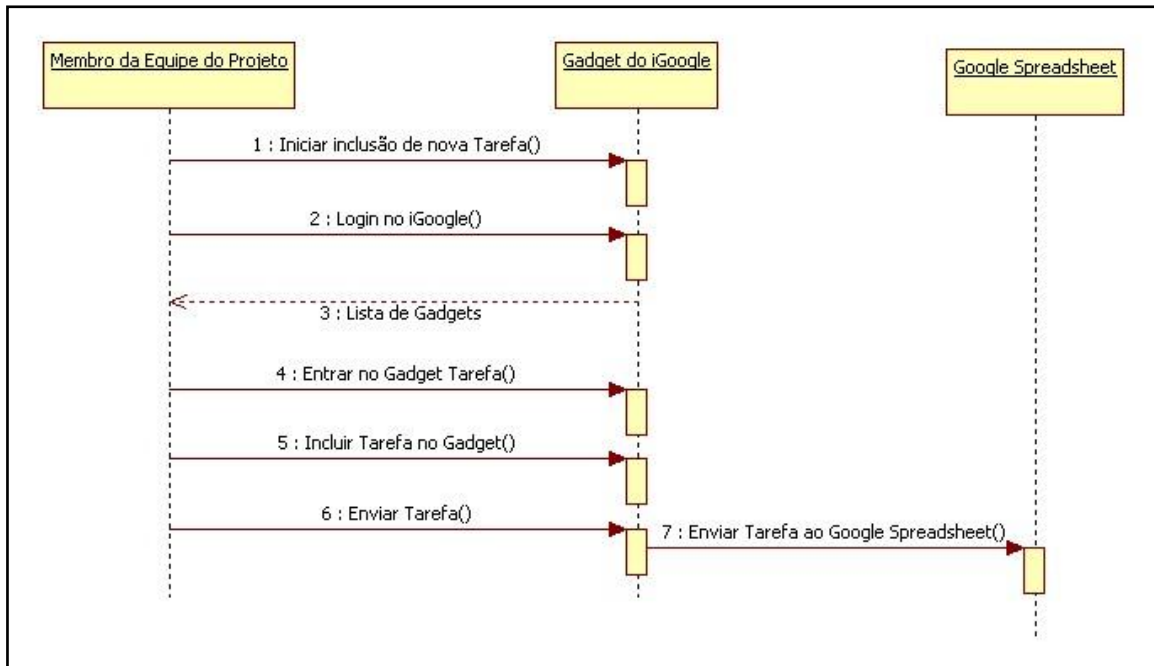
Uma interação interessante para pesquisas futuras seria o *gadget* interagindo tanto com o Google *Spreadsheet*, quanto com o banco de dados do dotProject. O *gadget* obteria informação do dotProject e atualizaria essa informação simultaneamente na planilha e também se algum usuário atualizar o *gadget* ele automaticamente atualizaria o dotProject e o Google *Spreadsheet*.

## 7 IDENTIFICAÇÃO DAS POSSÍVEIS ESTRATÉGIAS DE INTEGRAÇÃO

A partir do estudo das potenciais ferramentas do Google *Code* e suas possíveis interações com as ferramentas tradicionais de GP, a equipe elaborou as seguintes estratégias para interação com: o *Gadgets* do *iGoogle* (Figura 1), o *Google Spreadsheet API* (Figura 2), o *Google APPS Script*, o *Web Service dotProject* (Figura 3), e o *Google Calendar* (Figura 4).

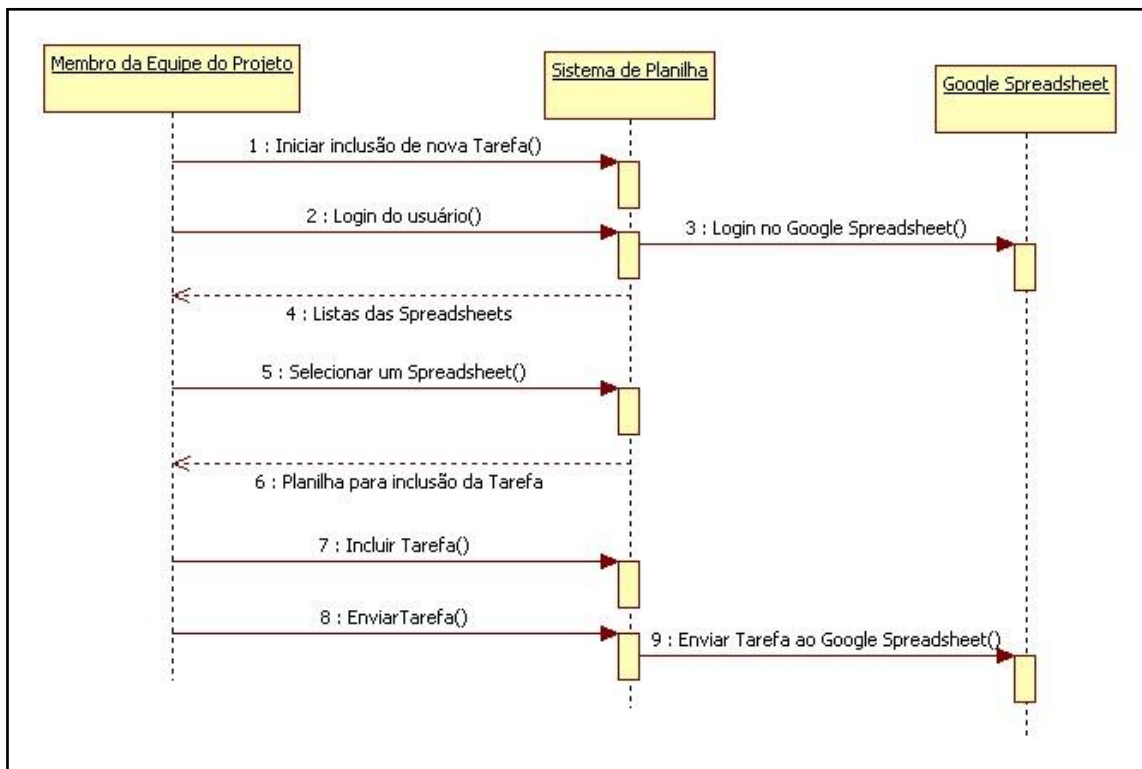
Para ilustração das estratégias citadas foi desenvolvido um diagrama de sequência com um ator que consiste em um membro da equipe de projeto, que realiza a ação de inclusão de uma nova tarefa no projeto. Cada diagrama de sequência representa uma estratégia de integração diferente para a realização do mesmo caso de uso, verificando assim os diferentes modos de interação com o usuário. Na integração com o Google *Spreadsheet*, o sistema de planilha consiste em uma aplicação cliente em Java que consegue realizar manipulações nas planilhas do Google *Spreadsheet*, assim como o sistema de agenda manipula as agendas do Google *Calendar*.

**Figura1** – Integração com os *Gadgets* do Google (iGoogle).



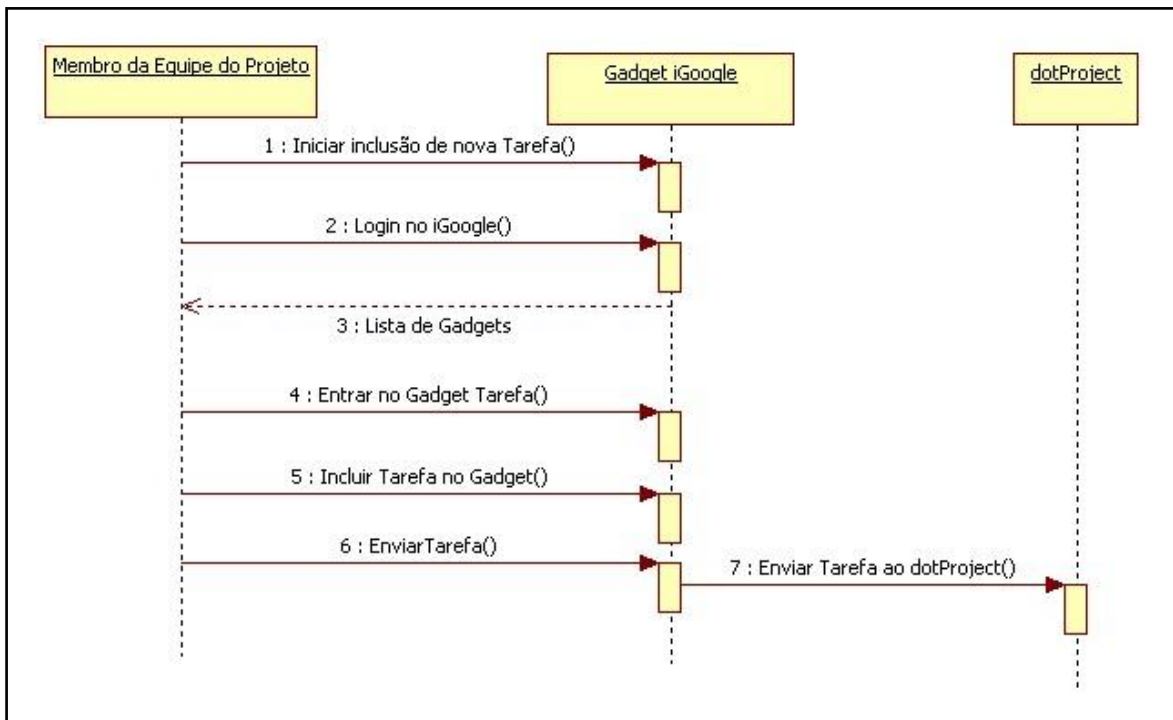
Fonte: Elaborado pelos autores.

**Figura 2** – Integração com o *Google Spreadsheet API*.



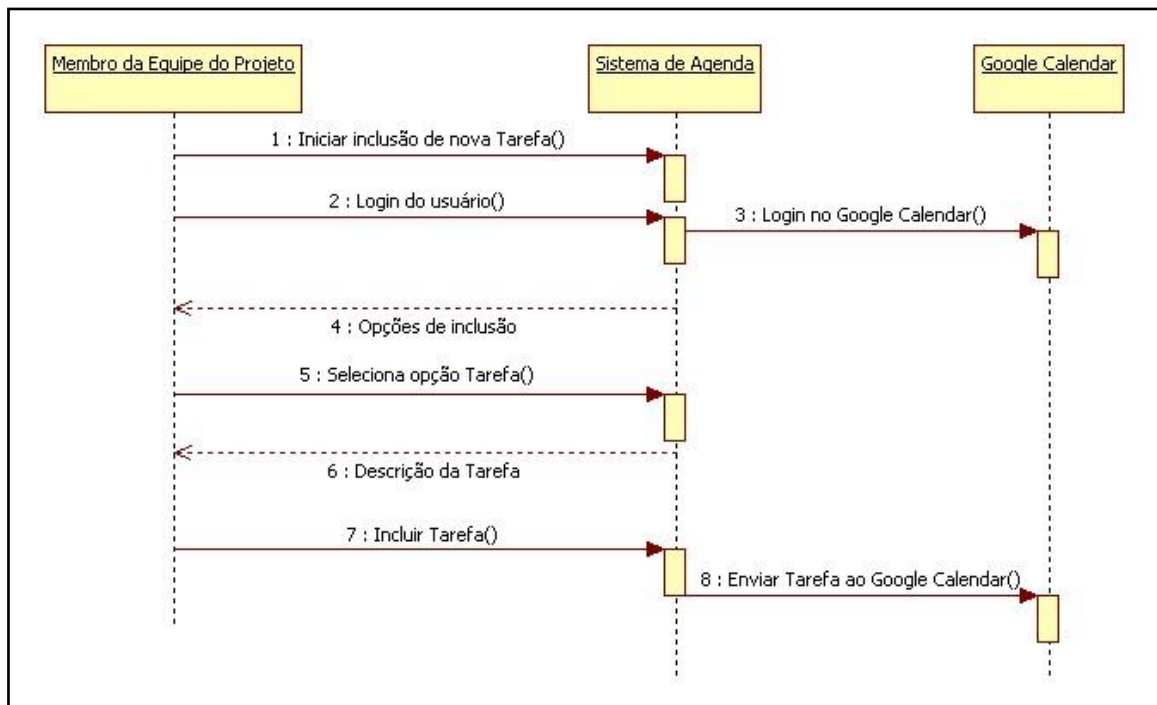
Fonte: Elaborado pelos autores.

**Figura 3** – Integração com o dotProject.



Fonte: Elaborado pelos autores.

**Figura 4** – Integração com o Google Calendar.



Fonte: Elaborado pelos autores.

## 8 IMPLEMENTAÇÃO DA ESTRATÉGIA *GOOGLE CALENDAR*

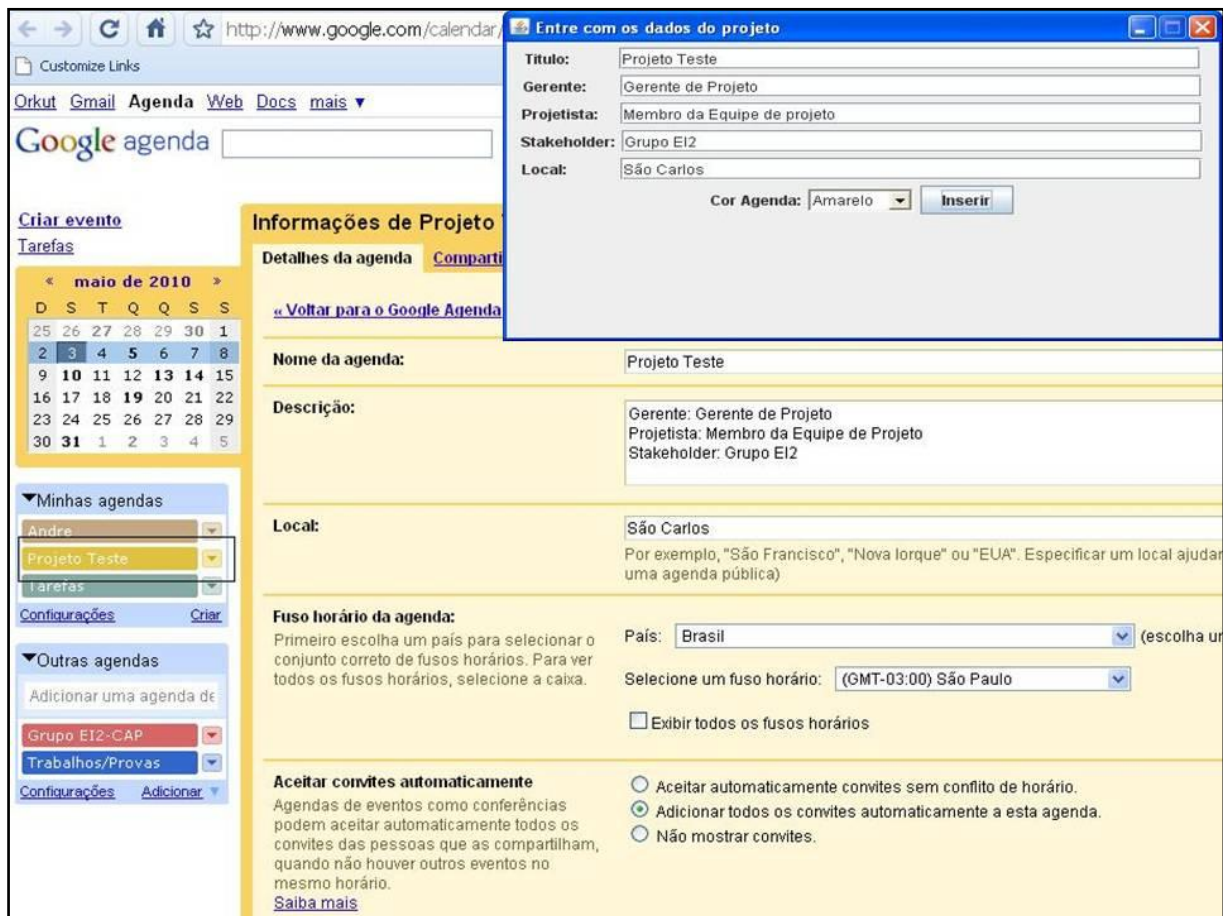
Dentre os métodos estudados, esse consiste no mais adequado para o desenvolvimento do projeto. Como a API de dados do *Google Spreadsheet*, também existe a API de dados do *Google Calendar* que permite a criação de aplicativos que realizam o acesso a todas as funcionalidades do *Google Agenda*. Com esses aplicativos podem-se criar novos eventos, editar e excluir eventos, consultar eventos específicos, listar todos os eventos, criar agenda, atualizar agenda, listar agendas, entre outras. Esses aplicativos, assim como na API de dados do *Google Spreadsheet*, são desenvolvidos em Java, por similaridade com a linguagem, e por meio desses aplicativos o usuário que possui uma conta *Google* pode modificar suas agendas, sem acessar a agenda.

Observou-se que esses aplicativos de acesso ao *Google Calendar* são mais interessantes para a integração com ferramentas de GP, que os aplicativos do *Google Spreadsheet*. Supondo que no *Google Calendar* cada agenda seja um projeto diferente e os eventos da agenda consistam em entregas de cada projeto, pode-se obter um maior detalhamento entre os projetos e as tarefas, como data de início e fim, pode-se fazer uma conexão com os vários projetos que estão em desenvolvimento e também um compartilhamento de agendas com vários usuários do grupo, assim todos que possuem aquela agenda, estarão atualizados online de todos os projetos e entregas do grupo.

Além disso, nas agendas e nos eventos pode-se fazer uma descrição do projeto como, gerente de projeto, participantes, *stakeholder*, assim como no evento pode-se descrever nome da tarefa, *status*, responsável, entre outras. Com isso o *Google Calendar* irá proporcionar uma maior interação com um aplicativo de GP, possibilitando um maior nível de detalhamento e interação das tarefas e dos projetos.

A comprovação da possibilidade da integração foi feita por meio do desenvolvimento de um aplicativo em Java com as bibliotecas cliente do *Google*, que interage com o *Google Calendar*, o qual já está em funcionamento, como protótipo, isto é, sem os detalhes de acabamento e de funcionalidades administrativas. A Figura 5 mostra uma tela do protótipo, no canto superior direito, realiza-se a inserção de um novo projeto e suas características e a tela principal do *Google Calendar* mostra como ficaram os dados dessa inserção que foi realizada.

Figura 5 – Telas do protótipo desenvolvido.



Fonte: Elaborada pelos autores.

Por meio desse aplicativo em Java consegue-se realizar manipulações na agenda, sem precisar acessá-la. O aplicativo foi desenvolvido para *desktop*, mas pode ser transportado também para a web, pelo Google App Engine, um aplicativo do Google que permite rodar programas na web com a infraestrutura do Google, sem necessidade de servidores, tudo fica hospedado na base do Google.

## 9 CONSIDERAÇÕES FINAIS E CONCLUSÕES

Com a análise das diversas APIs do *Google Code*, constatou-se que ocorreu um maior êxito na utilização do *Google Calendar* para o gerenciamento ágil de projetos, pois esse possibilita uma maior interação entre os diversos projetos do grupo e suas respectivas tarefas, possibilitando uma maior interação do grupo com todos os projetos que estão em andamento e suas respectivas tarefas. Da mesma maneira que o sistema manipula os dados do *Google Calendar* a partir de entradas realizadas pelo usuário, seria possível criar macros em sistemas de GP do tipo *standalone* que manipulassem os dados desses sistemas e os sincronizasse com os dados do *Google Calendar*.

O próximo passo dessa pesquisa será a utilização dos códigos gerados nesse estudo para a proposição de um sistema computacional completo e distribuído, esse sistema possibilitará um controle visual de projetos, voltado para equipes ágeis. O aplicativo será capaz de coletar os dados de diferentes agendas do *Google Calendar*, devidamente customizadas com dados necessários para o controle de projetos, e armazená-los em uma ferramenta de GP, que também será customizada para gerar um painel de controle para equipes ágeis.

A meta é projetar os dados em monitores *LCD* no local de trabalho, onde esteja parte da equipe de projeto, formando um painel de controle eletrônico, ao mesmo tempo em que parte da equipe distribuída possa acompanhar o andamento via o *Google Calendar*.

## REFERÊNCIAS

- Araujo, C. (2008). *Softwares de apoio ao gerenciamento ágil de projetos colaborativos de novos produtos: análise teórica e identificação de requisitos*. Dissertação de mestrado, Engenharia de Produção, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP, Brasil.
- Beck, K; Beedle, M.; Bennekum, A van; Cockburn, A; Cunningham, W.; Fowler, M. *et al.* (2001). *Manifesto for agile software development*. Recuperado em 10 janeiro, 2010, de <<http://www.agilemanifesto.org/>>.
- Barnes, T. A.; Pashby, I. R.; and Gibbons, A. M. (2000, November). Collaborative R&D projects: a framework for effective management (Vol. 1, pp. 210-216). *Proceedings of the IEEE International Conference on Management of Innovation and Technology – ICMIT 2000*, Singapore, 12-15.



- Google APPs. (2010a). *APIs do Google Apps*. Recuperado em 18 janeiro, 2010, de <http://code.google.com/intl/pt-BR/apis/apps>
- Google Calendar. (2010b). *APIs e ferramentas do Google Agenda*. Recuperado em 17 março, 2010, de <http://code.google.com/intl/pt-BR/apis/calendar>.
- Google Gadgets API. (2010c). *API de gadgets*. Recuperado em 18 janeiro, 2010, de < <http://code.google.com/intl/pt-BR/apis/gadgets>>
- Google Spreadsheets API. (2010d). *APIs e ferramentas do Google Spreadsheets (Google Labs)*. Recuperado em 18 janeiro, 2010, de <http://code.google.com/intl/pt-BR/apis/spreadsheets>.
- Hameri, A. and Puitinen, R. (2003). WWW-enabled knowledge management for distributed engineering projects. *Computers in Industry*, 50, 165-177.  
[http://dx.doi.org/10.1016/S0166-3615\(02\)00118-5](http://dx.doi.org/10.1016/S0166-3615(02)00118-5)
- iGoogle. (2010). *Página inicial de desenvolvedor do iGoogle (Google Labs)*. Recuperado em 10 janeiro, 2010, de < <http://code.google.com/intl/pt-BR/apis/igoogle>>.
- Microsoft. (2010). *Microsoft® Project 2010*. Recuperado em 18 janeiro, 2010, de <http://www.microsoft.com/project/en/us/default.aspx>.
- Ren, Z.; Anumba, C. J.; Hassan, T. M.; Augenbroe, G.; and Mangini, M. (2006, April). Collaborative project planning: a case study of seismic risk analysis using an e-engineering hub. *Computers in Industry*, 57(3), 218-230.  
<http://dx.doi.org/10.1016/j.compind.2005.12.002>
- Thiollent, M. (2002). *Metodologia da pesquisa-ação* (11a ed.). São Paulo: Cortez.
- White, D. and Fortune, J. (2002, January). Current practice in project management: an empirical study. *International Journal of Project Management*, 20(1), 1-11.  
[http://dx.doi.org/10.1016/S0263-7863\(00\)00029-6](http://dx.doi.org/10.1016/S0263-7863(00)00029-6)
- Wikipedia. (2010). *Computação em nuvem*. Recuperado em 10 janeiro, 2010, de [http://pt.wikipedia.org/wiki/Cloud\\_computing](http://pt.wikipedia.org/wiki/Cloud_computing).
- Woerner, J. and Woern, H. (2005, May). A security architecture integrated co-operative engineering platform for organised model exchange in a Digital Factory environment. *Computers in Industry*, 56(4), 347-360.  
<http://dx.doi.org/10.1016/j.compind.2005.01.011>

---

Data do recebimento do artigo: 29/03/2011

Data do aceite de publicação: 15/05/2011