

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

VINICIUS SOCCOL

SOFTWARE COLABORATIVO E APIS GOOGLE: UM ESTUDO DE CASO

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA

2011

VINICIUS SOCCOL

SOFTWARE COLABORATIVO E APIS GOOGLE: UM ESTUDO DE CASO

Trabalho de Diplomação apresentado ao Curso Superior de Tecnologia em Desenvolvimento de Sistemas de Informação da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título “Tecnólogo em Desenvolvimento de Sistemas de Informação”.

Orientador: Prof *M.Sc.* Romualdo Rubens de Freitas.

MEDIANEIRA

2011



TERMO DE APROVAÇÃO

SOFTWARE COLABORATIVO E APIS GOOGLE: UM ESTUDO DE CASO

Por

Vinicius Soccol

Este Trabalho de Diplomação (TD) foi apresentado às 14:45 h do dia 21 de Junho de 2011 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Desenvolvimento de Sistemas de Informação, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. O candidato foi argüido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. (Romualdo Rubens de Freitas)

UTFPR – *Campus* Medianeira

(Orientador)

Prof. (Neylor Michel)

UTFPR – *Campus* Medianeira

(Convidado)

Prof. (Fernando Schutz)

UTFPR – *Campus* Medianeira

(Convidado)

Prof. Juliano Rodrigo Lamb

UTFPR – *Campus* Medianeira

(Responsável pelas atividades de TCC)

À minha família que muito me apoiou em todos os momentos do desenvolvimento deste trabalho.

Ao meu orientador, que não mediu esforços para que a execução deste se tornasse possível.

RESUMO

SOCCOL, Vinicius. Software Colaborativo e APIs Google: Um Estudo de Caso. 2011. Trabalho de Conclusão de Curso (Tecnologia em Desenvolvimento de Sistemas de Informação). Universidade Tecnológica Federal do Paraná.

Este trabalho contém um estudo sobre o Trabalho Colaborativo através do modelo 3C (Comunicação, Coordenação e Colaboração). Explana também sobre *software* colaborativo e as APIs (*Application Programming Interface*) *javascript* da Google: AJAX de Pesquisa, de Idioma, de Gráficos, *Google Maps*, *Google Earth* e *Google Talk*.

Também apresenta a utilização das APIs Google em um *Software* Colaborativo. Contém uma pequena aplicação desenvolvida na linguagem PHP (Hypertext Preprocessor) através da utilização das APIs Google e finaliza com um estudo de caso sobre o *Software* Colaborativo Synovel Spicebird.

Palavras-chave: 3C, APIs, *javascript*.

ABSTRACT

SOCCOL, Vinicius. Software Colaborativo e APIs Google: Um Estudo de Caso. 2011. Trabalho de Conclusão de Curso (Tecnologia em Desenvolvimento de Sistemas de Informação). Universidade Tecnológica Federal do Paraná.

This work contains a study of collaborative work through the model 3C (Communication, Coordination and Collaboration). Explains also about collaborative software, and APIs (Application Programming Interface) of the Google javascript: AJAX Search, Language, Graphics, Google Maps, Google Earth and Google Talk. It also features the use of Google APIs in a Collaborative Software. Contains a small application developed in PHP (Hypertext Preprocessor) by using the Google APIs and ends with a case study of the Collaborative Software Synovel Spicebird.

Keywords: 3C, APIs, *javascript*.

LISTA DE SIGLAS

3C	Coordenação, Colaboração e Cooperação
AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
PHP	Hypertext Preprocessor
URL	Uniform Resource Locator
XMPP	Extensible Messaging and Presence Protocol

LISTA DE FIGURAS

Figura 1 – Modelo 3C.....	16
Figura 2 - Sistema de Comunicação mediado por computador.....	18
Figura 3- Modelo de Cooperação.....	21
Figura 4 - Fluxo de informações clássico.....	29
Figura 5- Fluxo de informações conforme trabalho colaborativo.....	30
Figura 6- Informação hierarquizada X informação colaborativa.....	31
Figura 7- API Google <i>Earth</i>	33
Figura 8 - Imagem do centro da cidade de Medianeira,PR, gerada pela API Google <i>Static Maps</i>	45
Figura 9 - Página inicial do sistema exemplo.....	56
Figura 10 - API Google <i>Earth</i> em execução.....	56
Figura 11 - Código para geração da API Google <i>Earth</i> no sistema exemplo.....	57
Figura 12 - API de Idiomas.....	58
Figura 13 - Código HTML API de Idiomas.....	58
Figura 14 - Código para tradução API de Idioma.....	59
Figura 15 - API de Pesquisa em funcionamento.....	60
Figura 16 - Código <i>javascript</i> para geração da API de Pesquisa.....	60
Figura 17 - API de Mapas em funcionamento.....	61
Figura 18 - Código para geração dos Gráficos pela API Google <i>Chart</i>	62
Figura 19 - Api <i>Static Maps</i> em execução.....	63
Figura 20 – Tela inicial do sistema colaborativo Synovel Spicebird.....	64
Figura 21 – Ferramentas de <i>e-mail</i> do <i>Software Synovel Spicebird</i>	65
Figura 22 – Ferramentas para trabalho com contatos no <i>software Synovel Spicebird</i>	66
Figura 23 – Cadastro de compromissos no <i>Software Synovel Spicebird</i>	66
Figura 24 – Cadastro de tarefas no <i>Software Synovel Spicebird</i>	67
Figura 25 - Inserção de Applets no Synovel Spicebird.....	68
Figura 26 – Erros apresentados pelo sistema <i>Synovel Spicebird</i>	69

LISTA DE QUADROS

Quadro 1 - Inserção da chave gerada para a API Google <i>Earth</i>	36
Quadro 2 - Carregamento do plug-in do Google Earth.....	36
Quadro 3 - Função de inicialização do plug-in do Google Earth.....	37
Quadro 4 - Estrutura do método <i>initCB</i>	37
Quadro 5 - Estrutura do Método <i>failureCB</i>	38
Quadro 6 - Exemplo completo do funcionamento a API Google <i>Earth</i>	38
Quadro 7 - Tradução de texto através da API Google <i>Translate</i>	41
Quadro 8 - Gráfico gerado pela API Google <i>Chart</i>	42
Quadro 9 - Desenvolvimento de um gráfico através da API Google <i>Chart</i>	42
Quadro 10 - Funcionamento da API de Pesquisa Google.	44
Quadro 11 – Código necessário para exibição da cidade de Medianeira – PR.	45

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 JUSTIFICATIVA.....	13
1.2 OBJETIVOS	13
1.3 METODOLOGIA.....	14
1.4 ESTRUTURA DO TRABALHO	14
2 TRABALHO COLABORATIVO (<i>GROUPWARE</i>)	15
2.1 Modelo de Colaboração 3C	16
2.1.1 Comunicação	17
2.1.2 Coordenação.....	19
2.1.3 Cooperação.....	21
3 SOFTWARE COLABORATIVO	23
3.1 Aplicações para uso individual	25
3.2 Aplicações para trabalho colaborativo.....	26
3.3 Fluxo de dados nas organizações	28
3.3.1 Fluxo de informações clássico	28
3.3.2 Fluxo de informações colaborativo	29
4 APIS.....	32
4.1 API do Google <i>Earth</i>	34
4.1.1 Funcionamento da API.....	34
4.1.2 Utilizando a API do Google <i>Earth</i>	35
4.2 API AJAX de Idioma.....	39
4.2.1 Utilização da API	39
4.2.2 Funcionamento da API.....	40
4.2.3 Invocando a API através de <i>Javascript</i>	40
4.3 API Google <i>Chart</i>	41
4.4 API AJAX de Pesquisa.....	43
4.4.1 Definição da API.....	43
4.5 API Google <i>Static Maps</i>	44

4.6 API Google <i>Talk</i>	46
4.7 Utilização das APIs AJAX de Idioma, de Pesquisa, Google <i>Chart</i> , Google <i>Earth</i> , Google <i>Maps</i> e Google <i>Talk</i> em um Sistema Colaborativo	47
4.8 Pontos Positivos na utilização das APIs	47
4.9 APIs em um Sistema Colaborativo.....	48
4.9.1 API AJAX de Idioma em um Sistema Colaborativo	49
4.9.2 API AJAX de Pesquisa em um Software Colaborativo.....	50
4.9.3 API Google <i>Chart</i> em um Sistema Colaborativo	52
4.9.4 API Google <i>Earth</i> em um Sistema Colaborativo	53
4.9.5 API Google <i>Static Maps</i> em um Sistema Colaborativo.....	53
4.9.6 Utilização da API Google <i>Talk</i> em um Sistema Colaborativo.....	54
4.9.7 Integração entre as APIs Google em um pequeno Sistema na linguagem PHP	55
5 ESTUDO DE CASO: O SISTEMA COLABORATIVO SYNOVEL SPICEBIRD.....	64
5.1 Vantagens na utilização do <i>Software Colaborativo Synovel Spicebird</i>	68
5.2 Conclusões sobre o sistema <i>Synovel Spicebird</i>	69
6 TRABALHOS FUTUROS	71
7 CONCLUSÃO.....	72
8 REFERENCIAS	75

1 INTRODUÇÃO

Nos últimos anos houve uma grande evolução da parte de *hardware* dos computadores. Esta evolução possibilitou o desenvolvimento e a execução de sistemas mais complexos, com diversas funcionalidades. Soma-se à este fato o aumento da largura de banda de Internet e a queda do seu preço.

Estes fatos possibilitaram a integração de diversas ferramentas de comunicação em Sistemas de Informação, tais como: vídeo, *chat* e até mesmo a fala.

Graças à isto surgiram os chamados Sistemas Colaborativos. O objetivo principal deste tipo de sistema é a disponibilização de um ambiente de trabalho virtual, onde seus membros possam interagir e modificar arquivos e projetos como se estivessem no mesmo departamento de uma empresa.

Um dos requisitos para que um Sistema Colaborativo funcione bem é que suas ferramentas devem ser disponibilizadas de forma à tornar a informação acessível ao usuário da maneira mais simplificada possível. Com o conhecimento das informações o usuário pode tomar decisões que o auxiliarão a realizar seus trabalhos diários e até mesmo a solucionar novos problemas.

Com este objetivo está a ideia da integração das APIs (*Application Programming Interface*) Google AJAX de Pesquisa, AJAX de Idioma, API de Gráficos, API do Google *Earth*, Google *Maps* e Google *Talk*. APIs são programas desenvolvidos por terceiros e com funcionalidades específicas. O desenvolvedor somente cria o código necessário para que determinada API seja disponibilizada no seu sistema e que os usuários possam usufruir da mesma. Ele não conhece seu funcionamento profundamente.

A inserção destas APIs em um Sistema Colaborativo tornaria o mesmo mais funcional e mais objetivo quanto ao repasse das informações aos usuários.

Este trabalho contém em sua fundamentação teórica a definição de trabalho colaborativo e o modelo que o mesmo segue, chamado de 3C (Coordenação, Colaboração e Cooperação).

Após estas definições é feita uma explanação sobre como funciona um Sistema Colaborativo, seus requisitos de usuário e requisitos de desenvolvedor também estão no texto.

Na segunda parte do trabalho é feita a definição de API (*Application Programming Interface*) e como algumas destas APIs poderiam ser utilizadas em um sistema colaborativo.

Em sua terceira parte foi feito o desenvolvimento de uma aplicação, que ilustra como as APIs AJAX de Idioma, API do Google *Earth*, API de Pesquisa, API Google *Static Maps* e API de Gráficos podem ser integradas para o desenvolvimento de um sistema.

1.1 JUSTIFICATIVA

Nos últimos anos houve uma alteração no cenário em que as informações são repassadas. Alguns anos atrás, principalmente através da *Internet*, o usuário acessava informações disponibilizadas por terceiros. No momento em que vivemos estamos entrando na era da colaboração. Neste modelo, as informações são repassadas pelos próprios usuários, de acordo com seu grau de informação e experiências anteriores.

Neste contexto entram em uso os *softwares* colaborativos. Seu principal objetivo é facilitar a troca de informações entre usuários posicionados em diferentes locais.

1.2 OBJETIVOS

Este trabalho tem como objetivo: realizar um estudo sobre o Trabalho Colaborativo e *Software* Colaborativo; definição do Modelo 3C; definição do conceito de APIs e funcionamento das APIs AJAX de Idioma, de Pesquisa, Google *Chart*, Google *Earth*, Google *Maps* e Google *Talk*; usos destas APIs em um *Software* Colaborativo; integração de algumas destas APIs em um pequeno sistema PHP (*Hypertext Preprocessor*) e um estudo de caso sobre o *Software* Colaborativo *Synovel Spicebird*.

1.3 METODOLOGIA

A metodologia para a realização deste trabalho foi a pesquisa bibliográfica e a realização de um estudo de caso sobre o *Software Colaborativo Synovel Spicebird* através da instalação e realização de testes no mesmo.

1.4 ESTRUTURA DO TRABALHO

A divisão das estruturas deste Trabalho de Conclusão de Curso é feita da seguinte maneira:

Capítulo 1: Introdução. Apresenta uma introdução ao tema deste trabalho. Contém também a sua justificativa, os objetivos e a metodologia utilizada.

Capítulo 2: Apresenta as definições de Trabalho Colaborativo e o Modelo 3C.

Capítulo 3: É feita uma explanação sobre *Software Colaborativo* e o fluxo de dados nas organizações.

Capítulo 4: Definição do conceito de APIs, detalhamento do funcionamento das APIs Google e utilização das APIs Google em um pequeno sistema PHP.

Capítulo 5: Um estudo de caso sobre o Sistema Colaborativo *Synovel Spicebird*.

Capítulo 6: Trabalhos que poderão ser realizados no futuro.

Capítulo 7: Conclusão.

2 TRABALHO COLABORATIVO (*GROUPWARE*)

Nas últimas décadas houve um grande avanço das tecnologias, tais como na área da informática, das telecomunicações e dos meios de transporte. Esta evolução possibilitou às empresas ampliarem suas fronteiras geográficas e atenderem à clientes em diversas partes do planeta.

Junto à isso também houve um aumento da complexidade dos trabalhos a serem executados pelos funcionários nas organizações, passando estes trabalhos a envolverem profissionais das mais variadas áreas, tais como: tecnologia, administração, psicologia, entre outras.

Surgiu então o conceito de trabalho colaborativo (*Groupware*), que é a distribuição das tarefas para grupos de pessoas (Fuks et al, 2002). Através do mesmo, pode ser realizada a troca de experiência entre profissionais com das mais variadas áreas, evitando possíveis erros que poderiam ocorrer se a visão de um projeto ficasse restrita à somente uma pessoa.

De acordo com Fuks et al (2002), existem também várias outras vantagens que o trabalho colaborativo pode trazer, tais como: redução de custos; excelência nos serviços prestados; resultados visíveis mais rapidamente e projetos menos suscetíveis a erros.

De acordo com Jamil:

O trabalho colaborativo é a forma de trabalho, apoiada num ambiente tecnologicamente adequado, que possibilita que todos os profissionais trabalhem de forma simultânea no projeto ou atividade em que estão envolvidos.(2001, p. 326).

Conforme o ponto de vista do autor Jamil na citação acima, para que o trabalho colaborativo atinja seus objetivos, os envolvidos devem executar as tarefas de forma organizada e seguindo uma ordem lógica.

Nas organizações que desejam executar suas tarefas de modo colaborativo, uma boa prática é adotar o chamado Modelo 3C (Comunicação, Coordenação e Colaboração). Este modelo de trabalho é baseado nos seguintes princípios: comunicação, coordenação e cooperação.

2.1 Modelo de Colaboração 3C

Quando trabalham em grupos os indivíduos podem produzir melhores resultados e através da comunicação surgem entre as pessoas surgem novas ideias. Outro ponto positivo é que erros que provavelmente não seriam percebidos se o trabalho fosse realizado por apenas um indivíduo agora podem ser levantados, pode-se realizar a busca da solução para estes problemas, além da produtividade também ser aumentada. Conforme afirma Fuks, et al:

Para colaborar, os indivíduos têm que trocar informações (se comunicar), organizar-se (se coordenar) e operar em conjunto num espaço compartilhado (cooperar). As trocas ocorridas durante a comunicação geram compromissos que são gerenciados pela coordenação, que por sua vez organiza e dispõe as tarefas que são executadas na cooperação. (2002, p. 2).

O Modelo de Colaboração 3C envolve três etapas: a comunicação - é o momento em que as tarefas são distribuídas; a coordenação - nesta etapa os indivíduos se organizam, ou seja, as tarefas são distribuídas de acordo com a formação de cada um e suas habilidades particulares; a cooperação - neste momento as tarefas são realizadas, conforme já distribuídas anteriormente, visando atingir o objetivo final.

A Figura 1, ilustra as etapas que envolvem o Modelo de Colaboração 3C.

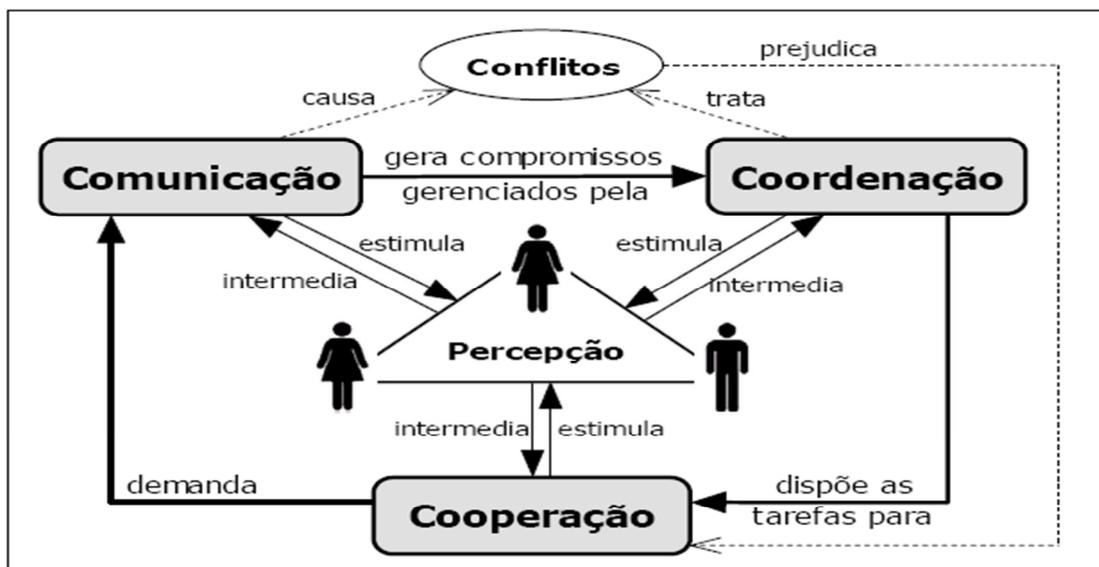


Figura 1 – Modelo 3C.
Fonte: Fuks et al. (2002, p. 5).

Como pode ser observado na Figura 1, o Modelo 3C envolve três etapas principais: comunicação, coordenação e cooperação.

Com o objetivo de tornar mais claro o funcionamento de cada uma destas, nos próximos tópicos que seguem serão feitos os detalhamentos de cada uma delas.

2.1.1 Comunicação

No dia a dia de trabalho a comunicação é de primordial importância. Através da mesma os indivíduos firmam compromissos. Estes compromissos firmados gerarão ações que irão impactar nas organizações. Confirme opinião do autor Bordenave:

Comunicação é a forma como as pessoas se relacionam entre si, dividindo e trocando experiências, ideias, sentimentos, informações, modificando mutuamente a sociedade onde estão inseridas. Sem a comunicação, cada um de nós seria um mundo isolado.(2002, p. 21).

Pode ser percebido que a comunicação é constante, seja através das ideias ou da distribuição de tarefas a outras pessoas. Esta troca de ideias modifica opiniões e o modo de pensar das pessoas, trazendo conhecimento.

É através da comunicação que as atividades são distribuídas. Também é através da mesma que regras e objetivos são definidos.

Com o avanço das tecnologias, surgiram ferramentas para facilitar a comunicação, não somente em organizações, mas também para auxiliar as pessoas no dia a dia na sociedade. É extenso o número de ferramentas que surgiram para auxiliar na troca de informações. Entre estas ferramentas podem ser citadas: *chat*, *e-mail*, videoconferência e chamadas telefônicas pela Internet.

Entre os objetivos destas ferramentas está facilitar a distribuição das informações mais rapidamente; agilizar a tomada de decisões, podendo esta tomada de decisão ser devido à uma informação repassada por um superior; também pode ser citado aqui a redução do tempo para execução das tarefas pois se a informação logo está disponível, ela auxilia no desenvolvimento dos trabalhos; redução de custos com telefonia por exemplo, sendo que neste caso podem ser utilizados sistemas de *chat* para conversa, entre outros inúmeros objetivos.

O desenvolvimento da tecnologia, em especial os componentes de *hardware* dos computadores, possibilitou o surgimento de programas mais complexos, com

mais recursos, e que conseqüentemente exigem mais capacidade de processamento. Esta evolução possibilitou aos desenvolvedores implementarem nos sistemas recursos que até então eram vistos como inviáveis.

É importante destacar aqui um dos pontos principais que possibilitou a evolução dos sistemas de comunicação: o aumento e barateamento da largura de banda de Internet. Este fator torna viável o uso de sistemas de comunicação via Internet, com áudio, vídeo e *chat* operando simultaneamente no mesmo sistema, sem falhas ou interrupções.

Devido à evolução dos recursos citados anteriormente, não é difícil imaginar que entre os sistemas mais utilizados para comunicação atualmente estão os chamados sistemas de comunicação mediados por computador. Uma das grandes vantagens em se utilizar um sistema de comunicação por computador está no fato de que várias formas de comunicação podem ser utilizadas em uma mesma aplicação, tais como: *e-mail*, lista de discussão, *chat*, fóruns e vídeos, entre outros.

Para que se possa melhor compreender como funciona um sistema de comunicação mediado por computador, a Figura 2 ilustra o funcionamento do mesmo.

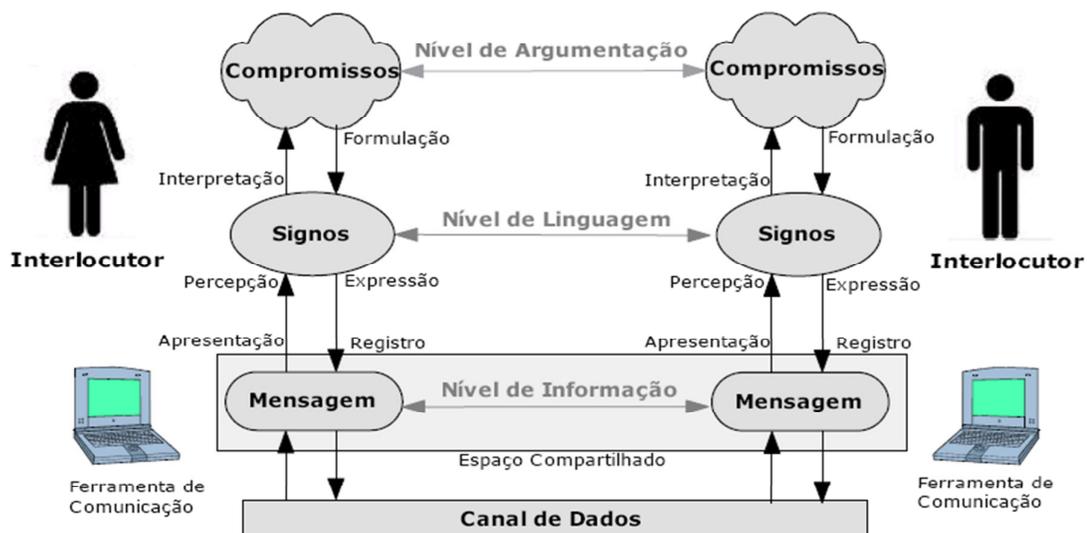


Figura 2 - Sistema de Comunicação mediado por computador.
Fonte: Fuks et al. (2002, p.3).

Após análise da Figura 2, pode-se compreender o funcionamento de um sistema de comunicação mediado por computador. Através dos signos, que é a linguagem, uma mensagem é transmitida de um interlocutor para outro. Esta mensagem é transmitida através de um canal de dados, que pode ser definido como o meio que permite à informação fluir livremente. Esta mensagem que foi transmitida pode gerar um compromisso a ser realizado por quem recebeu a mesma.

Vale destacar que a mensagem somente terá surtido efeito se for compreendida pelo interlocutor que a recebe, ou seja, resultar nos compromissos esperados. Para que a informação possa ser compreendida, o sistema utilizado pelos usuários deve prover recursos para tornar a informação mais clara possível para quem a recebe, ou do contrário os objetivos poderão não ser atingidos ou pode-se também chegar à resultados errôneos.

Outro ponto que faz parte do Modelo 3C é a coordenação. A mesma contém as regras que devem ser seguidas, quais os objetivos individuais de cada participante e como estão envolvidos como um todo no projeto.

Para que o trabalho colaborativo possa transcorrer da melhor maneira possível, a coordenação é de primordial importância. Um trabalho sem coordenação pode não gerar resultados. Neste caso, pode surgir um desinteresse por parte dos colaboradores que faça que não se empenhem em suas tarefas.

Trabalho sem coordenação pode também atingir resultados diferentes do esperado, pois o papel de cada envolvido não ficou claro desde o início dos trabalhos. Será agora descrita a etapa do modelo 3C chamada de coordenação.

2.1.2 Coordenação

Para que o trabalho colaborativo possa gerar resultados satisfatórios, é necessário que haja uma coordenação. A coordenação visa organizar as tarefas de acordo com um fluxo contínuo, desprendendo esforços nos pontos importantes, evitando que as informações sejam contraditórias ou mesmo que não haja um objetivo bem definidos. Nunes diz que:

A coordenação não é mais do que o processo de integração das diferentes atividades desenvolvidas em cada departamento da organização, de forma a atingir os resultados organizacionais pretendidos. (2007, p. 32).

A coordenação é a execução dos trabalhos pelos diferentes setores de uma organização, cada um com suas atividades, executados pelos seus membros.

Para que haja coordenação é necessário realizar um acompanhamento do andamento do processo quando o mesmo estiver em execução e após as tarefas serem realizadas, deve ser desenvolvida uma análise para verificar se realmente chegaram aos resultados desejados. Outro ponto importante é o desenvolvimento de uma documentação dos passos executados, que é chamado de memória do processo. Conforme Fuks et al:

As tarefas da colaboração são originadas dos compromissos assumidos na comunicação. O grupo se coordena através de mecanismos de coordenação de forma a garantir a execução das tarefas, respeitando suas interdependências.(2002, p. 4).

Percebe-se perceber que a colaboração surge após a comunicação. As tarefas são primeiramente definidas e repassadas aos executores, de uma maneira clara, para que os mesmos possam então executá-las. O responsável por repassar as tarefas pode ser quem elaborou as mesmas, ou pode somente estar repassando conforme solicitado por seus superiores. A distribuição destas tarefas pode ser realizada através dos sistemas de comunicação mediados por computador, conforme definido anteriormente.

Em muitos casos entra em cena o chamado protocolo social. O mesmo é caracterizado pela ausência de mecanismos de coordenação explícitos entre as atividades. Neste caso, cada envolvido já sabe as tarefas que deve realizar, sem que se precise informá-lo constantemente de seus deveres.

A terceira etapa envolvida no modelo de colaboração 3C é a cooperação. Na fase de cooperação que são reunidos os esforços individuais em um produto, maior e palpável.

Neste momento entra o trabalho em grupo. A troca de informações se faz realmente necessária e os profissionais de diversas áreas expressam suas opiniões e executam as tarefas que lhes foram atribuídas.

2.1.3 Cooperação

De acordo com Scharage (1996), comunicação e coordenação, apesar de vitais, não são suficientes. É necessário espaço compartilhado para criar entendimento compartilhado. Desta forma entende-se que Cooperação é a reunião das atividades realizadas em conjunto pelos membros de um grupo ou organização, visando atingir os objetivos propostos pela coordenação.

Faz-se necessário citar aqui a opinião do autor Simon, no momento em que diz que:

Cooperação significa trabalhar em conjunto visando somar resultados, aproveitando melhor os recursos disponíveis, em geral escassos. Há interesse especial quando a cooperação envolve grande número de cooperantes. A Cooperação pode resultar em algo mais significativo do que qualquer um dos cooperantes poderia realizar sozinho, ilustrando o ditado: "O todo é maior que a soma das partes". (1999, p. 1).

Assim tem-se que um dos principais objetivos da cooperação é aproveitar as habilidades individuais de cada um, visando gerar um produto final de melhor qualidade e mais bem elaborado do que se o mesmo fosse realizado somente por um indivíduo.

As etapas que representam o conceito de colaboração são apresentadas na Figura 3.

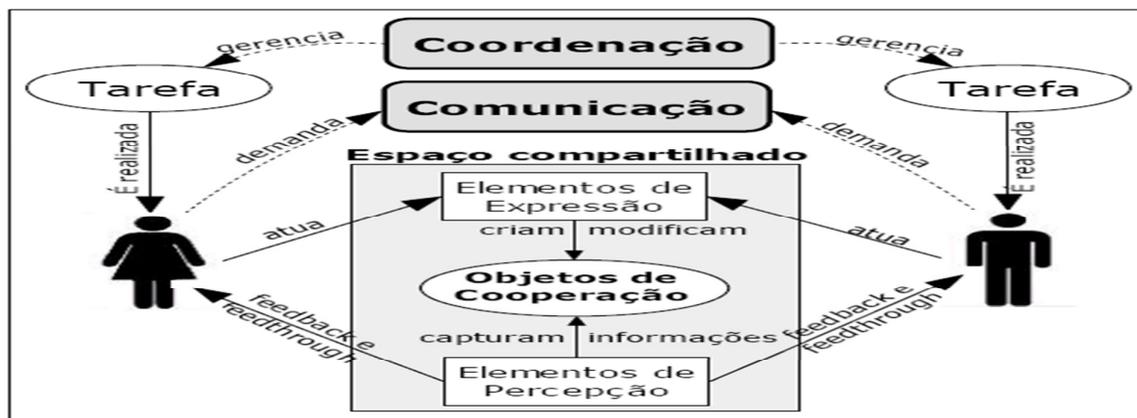


Figura 3- Modelo de Cooperação.
Fonte: Fuks et al. (2002, p.5).

Observando a Figura 3, pode ser percebido como as pessoas realizam tarefas, gerenciam, solicitam a realização, criam, modificam. Os envolvidos fazem todas estas tarefas em um espaço compartilhado, que pode ser, por exemplo, um

departamento de uma organização ou um ambiente de trabalho em um software colaborativo.

Um dos pontos importantes na cooperação é a percepção. Através da mesma os indivíduos inseridos em determinado contexto podem perceber como está a evolução do trabalho de seus companheiros e quais são os seus deveres.

A cooperação entre os membros é um ponto muito importante para as empresas na atualidade. Através da mesma, levando-se em conta uma primordial importância da comunicação, os indivíduos trocam experiências e então se auxiliam para a realização das tarefas.

Após a definição de trabalho colaborativo e do modelo de comunicação 3C será abordado agora o conceito de *Software Colaborativo*.

Sistemas colaborativos são aplicativos que possibilitam a troca de informações e a realização das tarefas por diversos setores e funcionários.

3 SOFTWARE COLABORATIVO

O trabalho colaborativo traz muitos benefícios para as organizações, tais como: redução de custos; menor probabilidade de erros em projetos; maior rapidez na conclusão das tarefas.

O avanço das tecnologias possibilitou o surgimento de uma categoria de *software* que visa facilitar o trabalho colaborativo: os chamados *softwares* colaborativos, ou *groupware*.

Este tipo de sistema tem por objetivo encurtar distâncias entre os colaboradores das empresas, tornando as informações acessíveis à todo o momento para os participantes, possibilitando assim a tomada de decisões imediatas, a troca de informações e a distribuição de tarefas. Conforme as palavras dos autores Gibbs e Rein:

O termo *groupware* representa uma família de aplicações, baseadas em computador, que dá suporte a grupos de pessoas, engajadas em uma tarefa comum e que provê uma interface para compartilhar o ambiente, especialmente ao nível de comunicação, colaboração e suporte à decisão.(1993, p. 10).

Pode-se concluir que os *softwares* colaborativos provêm um ambiente compartilhado. O objetivo deste ambiente compartilhado é proporcionar a reunião dos usuários como se estivessem trabalhando em um mesmo local ou setor de uma empresa.

Existem vários *Softwares* Colaborativos disponíveis no mercado. Dentre os pagos podem ser citados:

- Microsoft Share Point;
- IBM Lotus Notes;
- TalkAndWrite;
- Oracle Beehive Collaboration Software e
- Brazip mysuite.

Já nos casos dos sistemas *opensource*, se enquadram os seguintes:

- OpenGroupware;
- eGroupWare;
- Synovel Spicebird;

- TikiWiki e
- Drupal Framework.

No ambiente dos *Softwares* Colaborativo as informações são trocadas, arquivos disponibilizados e alterados por seus membros. A ideia é tornar este ambiente é tornar as informações, arquivos ou projetos acessíveis à todos, para que possam ao mesmo tempo estar usufruindo das informações neles disponibilizadas ou até mesmo incrementar os modelos existentes.

É importante lembrar que o termo *groupware* refere-se ao conjunto de tecnologias que visa apoiar o trabalho colaborativo. Estas tecnologias podem ser divididas em quatro categorias:

- Interfaces: é o conjunto de hipermídias, interfaces gráficas e elementos de multimídia;
- Orientação a objetos: são as diferentes plataformas, linguagens e banco de dados em que se encontram os sistemas;
- Comunicação: é a infraestrutura de rede e transmissão de dados, gerenciamento de documentos e dados e troca de mensagens;
- Conhecimento: são as técnicas de inteligência artificial, redes neurais e reconhecimento de modelos.

Todas estas categorias possibilitam a existência do trabalho colaborativo. As mesmas interagem entre si em um sistema *groupware*, geralmente não ao mesmo tempo devido à complexidade de cada uma.

Podem ser analisadas aqui as palavras de Araujo et al, quando diz que:

Ambientes computacionais voltados para o trabalho colaborativo propiciam várias vantagens no auxílio ao trabalho individual, em grupo e corporativo. No entanto, os sistemas informatizados precisam prover benefícios tangíveis para cada um desses domínios de colaboração e permitir que o indivíduo possa executar seu trabalho de maneira eficiente e criativa.(2003, p. 3).

Segundo a citação, percebe-se que existem basicamente dois tipos de aplicações nas organizações. São as aplicações de uso individual e as aplicações de trabalho colaborativo.

Nas aplicações de trabalho individual, geralmente o indivíduo inicia a tarefa e a conclui por ele mesmo. Neste tipo de sistema estão contidas somente as ideias de uma única pessoa, expressando nos trabalhos o seu modo de ver as coisas. É

importante ressaltar aqui que muitas vezes esse trabalho foi feito baseado em uma análise muito superficial.

Já em aplicações de trabalho colaborativo várias pessoas desenvolvem o mesmo projeto ou ideia ao mesmo tempo. Percebe-se então que quando são vários os envolvidos, é gerada uma visão mais abrangente do objetivo a ser alcançado.

Para tornar mais claros os conceitos de aplicações de uso individual e aplicações de trabalho colaborativo as mesmas são definidas a seguir.

3.1 Aplicações para uso individual

Usualmente as aplicações para uso individual nas empresas se enquadram nas categorias que seguem:

- Processamento de texto: são *softwares* para a elaboração de cartas, relatórios, textos e documentos em geral;
- Planilha eletrônica: sistemas para a elaboração de tabelas e quadros, geralmente com cálculos matemáticos envolvidos;
- Banco de dados: para o armazenamento seguro de informações que provavelmente deverão ser recuperados em outro momento;
- Elaboração de apresentações: são sistemas com o objetivo de elaborar apresentações que serão ministradas a outros funcionários, com o objetivo de definir metas, mostrar resultados, buscar soluções para problemas existentes;
- Sistemas de agenda: com o objetivo de armazenar, compromissos, contatos telefônicos ou mesmo tarefas diárias;
- Sistemas de Internet: possuem objetivos variados, como por exemplo cotações com fornecedores, envio de *e-mails*, pesquisa de produtos.

Nas aplicações de uso individual o usuário inicia os trabalhos, termina e grava os dados gerados. Nesta categoria se enquadram os trabalhos em que a sua realização é mais simples e rápida, e a compreensão das atividades realizadas envolve geralmente somente um profissional de uma área específica, ou seja, não é necessário o envolvimento de profissionais de áreas variadas para a solução do problema, é um trabalho específico e focado em uma área determinada. É interessante citar aqui as palavras de Araujo et al:

A principal característica das aplicações individuais é que elas podem organizar, auxiliar, arquivar e, de uma forma geral, melhorar a produtividade das pessoas que utilizam esses sistemas.(2003, p.4).

Entende-se que quando um trabalho é realizado em uma aplicação individual, o mesmo geralmente possui um fluxo de trabalho bem definido, ou seja, organização das ideias, criação e arquivamento.

Este tipo de trabalho possui um prazo definido e os espera-se por resultados palpáveis em um curto período de tempo. A outra categoria de software a ser analisada agora são os colaborativos, ou *groupware*.

Poderá ser percebido na leitura que o foco dos *softwares* colaborativos é o desenvolvimento de trabalho em grupo. As vantagens destes aplicativos vem sido percebidas pelas empresas, vantagens estas já citadas anteriormente.

3.2 Aplicações para trabalho colaborativo

Pode-se começar a falar nas aplicações de trabalho colaborativo analisando a citação dos autores Araújo e Campanário, no momento em que relatam que:

A busca de resultados pelas empresas, aliada à rápida evolução tecnológica observada nas últimas décadas, vem promovendo mudanças organizacionais e comportamentais no ambiente corporativo, onde a informação assume caráter primordial nas atividades empresariais. (2003, p. 1).

Houve uma mudança na forma como os trabalhos são realizados nas organizações. Cada trabalho geralmente era realizado individualmente, com pouca comunicação e troca de informações entre os membros. Mas as empresas começaram a perceber que a troca de informações é cada vez mais fundamental, e graças à evolução das tecnologias, principalmente computacionais, tornou-se mais fácil disseminar as informações para os interessados conforme necessário.

Assim como os sistemas individuais, os sistemas de *groupware* também podem ser classificados em categorias. Dentre as categorias existem as seguintes:

- Sistemas de *e-mail*: são sistemas que possibilitam a troca de arquivos e mensagens e funcionam sob uma rede de computadores;
- *Workflow*: são sistemas criados para realizar um acompanhamento do fluxo dos trabalhos na empresa e de informações da empresa;

- Sistemas de gerenciamento de documentos: fornecem acesso rápido à documentos armazenados de forma digital;
- Reuniões eletrônicas: através dos mesmos é possível realizar reuniões entre pessoas, visando buscar a solução de problemas nas mais diversas situações;
- Sistemas de co-autoria e projeto: possibilitam a entrada de dados em arquivos e projetos por diferentes usuários, localizados em diferentes locais;
- Videoconferência: possibilitam a realização de encontros, com áudio e vídeos sendo transmitidos simultaneamente. Um dos grandes objetivos deste tipo de sistema é evitar custos por parte das empresas no deslocamento de seus funcionários à pontos específicos para a realização de reuniões;
- Telepresença, avatares e realidade virtual: buscam recriar ambientes virtuais para pessoas localizadas em locais distintos. Os avatares representam os membros presentes.

Vale destacar aqui principais objetivos na realização de um trabalho em grupo é o aumento da produtividade que é proporcionado quando este tipo de trabalho é realizado.

Ao ser realizado em grupo, este também melhora a comunicação e a colaboração entre os membros deste grupo. De acordo com Brink (1998) as aplicações *groupware* podem ser divididas em três categorias.

- Mesmo espaço e tempo – Nesta primeira categoria as pessoas estão localizadas no mesmo local de trabalho, ao mesmo tempo. O objetivo destas aplicações é proporcionar encontros virtuais entre os envolvidos. Permitem gerar conteúdo, salvar e gerar relatórios. Temos como exemplo sistemas de projetos em grupo;
- Mesmo tempo e espaços diferentes – Os usuários estão trabalhando em um projeto ao mesmo tempo, mas em locais diferentes. Este tipo de aplicação deve proporcionar ferramentas de telecomunicação para a troca de informações entre os envolvidos. Neste tipo de sistema os principais tipos de aplicação são a videoconferência e os sistemas de troca de arquivos.

- Tempos diferentes em espaços diferentes – Tem por objetivo a distribuição, o encaminhamento e o transporte de informações. Geralmente as informações são armazenadas em banco de dados, o que facilita o seu acesso por outros membros em tempos diferentes.

A maneira como a informação se organiza é chamada de fluxo. Neste momento serão abordados os tipos de fluxo de informação existentes e quais as diferenças entre os mesmos.

3.3 Fluxo de dados nas organizações

Nas organizações tradicionais os meios em que as informações são transmitidas são engessados e as mesmas confiam fielmente nestes meios de compartilhamento de informações. Segundo Araujo et al:

As organizações tradicionais são compartimentadas e confiam excessivamente em seus canais de comando. Nas organizações modernas, a informação deve fluir livremente para onde for útil. As fronteiras e burocracias retardam os processos de negócios e as decisões e isolam as pessoas da realidade. As organizações mais bem-sucedidas são cada vez mais enxutas e ágeis, com uma estrutura fluida capaz de responder rapidamente às mudanças do mercado. (2003, p. 7)

Diante do que relata o autor, podemos avaliar que nas organizações de sucesso a informação deve chegar de maneira ágil e assim que for necessária.

3.3.1 Fluxo de informações clássico

Muitas vezes a burocracia faz com que a informação não chegue ou demore muito a chegar a quem precisa. Isso pode ocasionar a perda de oportunidades de negócios com clientes, atraso no desenvolvimento de projetos, entre outros prejuízos. A Figura 4 abaixo ilustra o fluxo de informações clássico nas organizações.

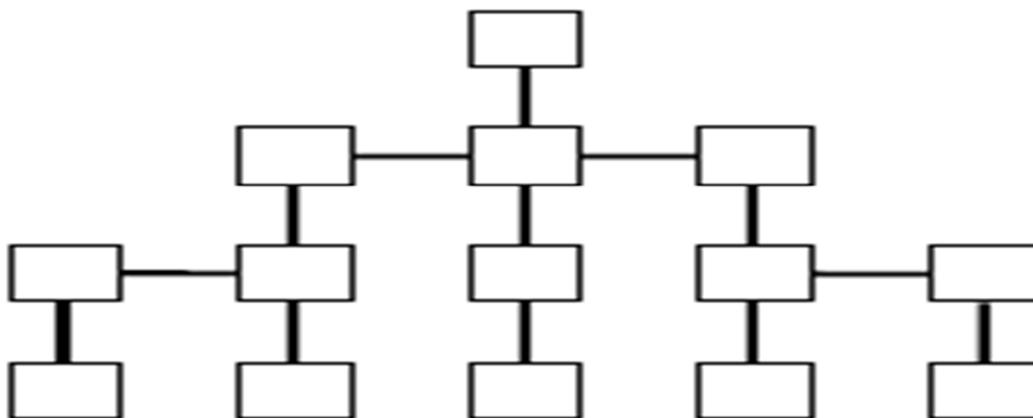


Figura 4 - Fluxo de informações clássico.
 Fonte: Araujo et al. (2003, p. 7).

De acordo com a Figura 4, a informação flui de acordo com as hierarquias da empresa. A mesma passa primeiramente pelos superiores e somente depois chega aos funcionários de nível mais baixo.

Neste modelo muitas vezes a informação fica restrita somente a gerentes e diretores, sendo que a mesma poderia ser bem aproveitada por outros funcionários, para a tomada de decisões.

É claro que existem informações confidenciais e que somente devem ser de conhecimento a quem for interessante. Podem ser citadas por exemplo informações de negociações e desenvolvimento de novos produtos.

3.3.2 Fluxo de informações colaborativo

Outro modelo de fluxo de informações é o que segue o trabalho colaborativo. No mesmo, a informação está disponível em um mesmo nível de hierarquia.

Para melhor esclarecer este modelo abaixo está a Figura 5, que ilustra o mesmo.

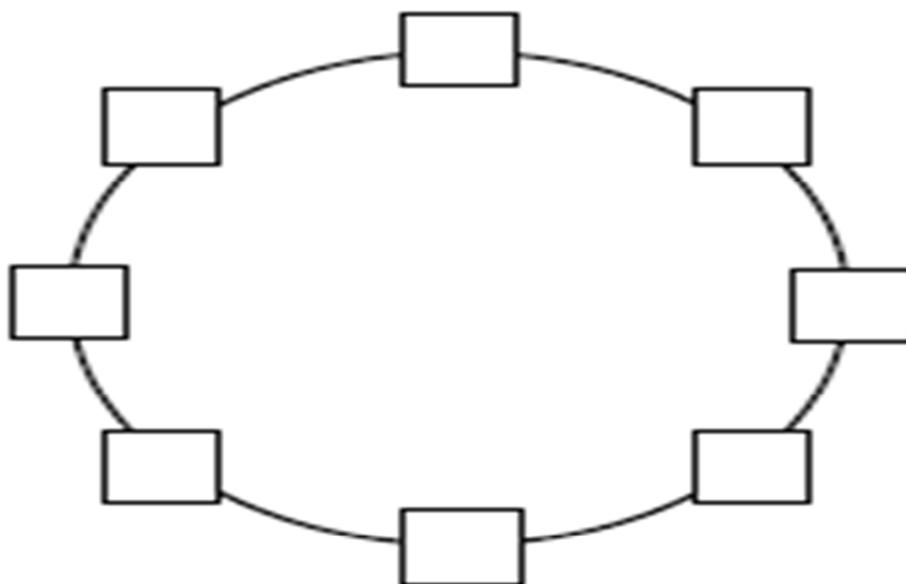


Figura 5- Fluxo de informações conforme trabalho colaborativo.
Fonte: Araujo et al. (2003, p. 7).

A Figura 5 está ilustrando o fluxo de trabalho conforme o trabalho colaborativo. Neste tipo de fluxo de informações todos os envolvidos estão no mesmo nível e a informação pode fluir livremente, tanto para um lado, quanto para o outro. É possível perceber que neste modelo gerentes e diretores podem estar trocando informações diretamente com subordinados ou no sentido inverso.

No mesmo, a troca de informações entre gerentes e subordinados não significa uma quebra da hierarquia funcional, mas sim uma ruptura na hierarquia da informação.

Conforme pode ser percebido na Figura 6, há uma mudança, a qual a informação transforma-se em elos interligados.

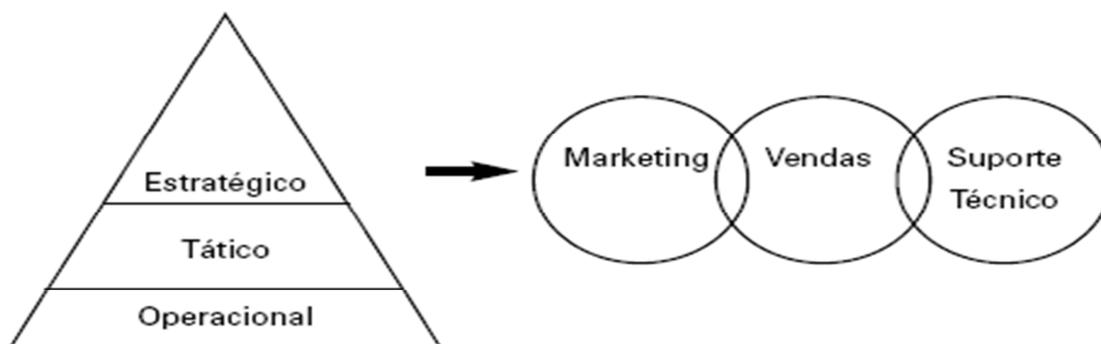


Figura 6- Informação hierarquizada X informação colaborativa.
Fonte: Araujo et al. (2003, p. 7).

Através destes elos, a informação circula entre os departamentos. Isto se faz importante no sentido de que com esta troca de informações os processos são agilizados, esforços desnecessários são poupados, podendo ser gasto tempo de trabalho no desenvolvimento de outros objetivos.

Finalizando, é importante dizer que para que *Software Colaborativo* seja eficaz, o mesmo deve seguir alguns requisitos. Estes são requisitos de usuário e requisitos do desenvolvedor, que serão agora definidos.

4 APIS

Conforme a evolução dos sistemas computacionais, também houve um aumento na complexidade das tarefas a serem executadas pelos mesmos.

Muitas vezes os desenvolvedores de *softwares* necessitam implementar em suas aplicações funções específicas, como por exemplo *chat*, tradução de textos e pesquisa.

Estas funções serão complementos adicionados ao sistema, sendo classificadas como funcionalidades extras, e que tem como objetivo facilitar o dia a dia de trabalho do usuário. Conforme diz Foldoc:

Uma API também pode fornecer uma interface entre uma linguagem de alto nível e utilitários de baixo nível e serviços que foram escritos, sem considerar as convenções de chamada suportadas pelas linguagens compiladas. Neste caso, a principal tarefa da API pode ser a tradução da lista de parâmetros de um formato para outro e à interpretação de chamadas por valor e argumentos de chamada por referência numa ou em ambas as direções.(1995:1).

Uma API (*Application Programming Interface*) também pode ser caracterizada como um interpretador. Pode-se ter, por exemplo, uma API através do qual são realizadas chamadas por um *software* para o mesmo utilizar componentes fornecidos pelo sistema operacional em que está rodando.

Existem funcionalidades principais e de importância primordial para o funcionamento de um sistema que está sendo desenvolvido, que tomam mais tempo dos desenvolvedores. Estas funcionalidades complexas a serem implementadas são chamadas de regras de negócio. Após esta análise por algumas grandes empresas de software, como, por exemplo, a Google, as mesmas focaram no desenvolvimento do conceito de APIs. Uma API é uma forma de duas aplicações de computador trocarem informações uma com a outra, em uma linguagem comum que ambas entendam.

APIs são componentes prontos para serem utilizados pelos desenvolvedores. Uma API também pode ser utilizada quando o desenvolvedor deseja que a realização de alguma tarefa esteja disponibilizada em seu sistema, mas o mesmo não quer conhecer profundamente como esta tarefa será realizada, somente adicioná-la ao sistema.

Podem ser citadas como exemplo as APIs desenvolvidas pela Google e que podem ser incorporadas nas aplicações, tais como: APIs Ajax de Idioma – tem por

objetivo a tradução de textos; API Ajax de Pesquisa – facilita as buscas na Internet; Google *chart* – possibilita a geração de gráficos nas aplicações; Google *Earth* – disponibiliza um globo terrestre com fotos via satélite, utilizado principalmente para localização; Google *Maps* – disponibilização de mapas e Google *Talk* – API para conversas *on-line*.

Atualmente o conceito de APIs é muito empregado no desenvolvimento de sistemas para a *web*. A grande utilização das APIs em sistemas *web* se deve ao fato de que muitas delas fornecem informações em tempo real para os usuários, como por exemplo, APIs que auxiliam na localização, previsão de tempo e *chat*. Estas APIs se conectam a um servidor, buscam as informações, e as fornecem ao usuário do sistema. Um exemplo é a API Google *Earth*, que é utilizada para visualização do globo terrestre por fotos via satélite, ilustrada na Figura 7.



Figura 7- API Google *Earth*.

Fonte: <http://code.google.com/intl/pt-BR/apis/earth/>

A Figura 7 ilustra a API do Google *Earth* em funcionamento. A mesma permite uma visualização do globo terrestre, e através da aproximação as imagens vão se tornando mais nítidas. Esta API pode ser facilmente incorporada a um sistema *web*.

É importante destacar que as APIs da Google permitem que funcionalidades sejam adicionadas a sistemas *web*. A API do Google *Earth*, por exemplo, não representa somente o globo terrestre nos sites *web*, mas sim um conjunto de funções, que permitem adicionar o mesmo aos sites.

As APIs da Google, objetivo deste trabalho são adicionadas a sites *web* através de código *javascript*.

Nos próximos tópicos, para cada API será feita uma definição sobre a mesma, e em seguida será explicado como se deve proceder para fazer uso da mesma na aplicação.

4.1 API do Google *Earth*

A API do Google *Earth* tem como objetivo fornecer funções para a inclusão do globo terrestre em sistemas *web*. Pela visualização do mesmo, o usuário pode localizar diferentes partes do planeta, com fotos via satélite que se tornam cada vez mais nítidas à medida que a versão da API evolui, sendo que atualmente está na versão 1.

4.1.1 Funcionamento da API

De acordo com a documentação da API, para o correto funcionamento da mesma os seguintes passos devem ser executados (2011):

1. Carregar a API do Google *Earth*;
2. Criar um elemento Div para armazenar o plug-in;
3. Criar funções para inicializar o plug-in;
4. Chamar as funções de inicialização quando a página estiver carregada.

De acordo com o que consta na documentação, até o momento a mesma está disponível para as seguintes plataformas:

Microsoft Windows (2000, XP e Vista)

1. Google Chrome 1.0 ou superior
2. Internet Explorer 6.0 ou superior
3. Firefox 2.0 ou superior
4. Flock 1.0 ou superior

Apple Mac OS X 10.4 ou superior (Intel e PowerPC)

1. Safari 3.1 ou superior
2. Firefox 3.0 ou superior

Para realizar a instalação do plug-in, basta acessar à alguma página em que o mesmo está sendo executado, como, por exemplo, <http://code.google.com/apis/earth>. Desta maneira o navegador solicitará permissão para instalar o plug-in, para que o conteúdo inserido no site possa ser visualizado. Lembrando que este é o procedimento para que o globo possa ser visualizado.

4.1.2 Utilizando a API do Google *Earth*

A utilização da API do Google *Earth* é simples. O processo detalhado está contido na documentação da API (disponível em: <http://code.google.com/intl/pt-BR/apis/earth/documentation/>).

Para poder utilizar os arquivos *javascript* que compõem a API, é necessário gerar um código. Este código pode ser obtido facilmente acessando o seguinte endereço: <http://code.google.com/intl/pt-BR/apis/maps/signup.html>. O desenvolvedor deve somente concordar com os termos e inserir a URL do seu site. Após isto a chave será gerada.

Após este procedimento ser realizado, a chave recebida deverá ser colocada dentro da tag `<head>` na página html desejada, conforme Quadro 1.

```
<script src="http://www.google.com/jsapi?key=ABCDEF"></script>
```

Quadro 1 - Inserção da chave gerada para a API Google *Earth*.
Fonte: - <http://code.google.com/intl/pt-BR/apis/earth/documentation/>

De acordo com o Quadro 1, a chave gerada, conforme instruções repassadas anteriormente, deve ser inserida no código no lugar das letras “ABCDEF”. O *source* deste *script* (src), apontará para um arquivo contendo somente o método *google.load*, que é utilizado para carregar as APIs da Google.

Dando continuidade aos procedimentos, agora deverá ser aberta uma nova tag `<script>`. Esta tag conterá o código *google.load*("earth", "1"). Isso diz ao Google para carregar o módulo *earth* no *namespace* *google.earth* e especifica a versão 1 (a última versão estável da API).

O plug-in do Google *Earth* é carregado em um elemento `<div>` com um ID exclusivo, conforme Quadro 2.

```
<div id="map3d" style="height: 400px; width: 600px;"></div>
```

Quadro 2 - Carregamento do plug-in do Google Earth.
Fonte: <http://code.google.com/intl/pt-BR/apis/earth/documentation/>

Analisando o Quadro 2, percebe-se que o Plug-in do Google *Earth* será carregado dentro do div com id igual a *map3d*.

Para a representação visual do Plug-in na página *web* é necessária a inicialização do mesmo. Esta inicialização é composta de três funções. Os objetivos destas funções são:

- tentativa de criar uma nova instância do plug-in;
- ser chamado quando a instância do plug-in for criada com êxito;
- ser chamado se a instância não puder ser criada.

Para tornar mais fácil a compreensão de cada uma destas funções, agora serão definidas as mesmas. A primeira função deve ser semelhante à apresentada no Quadro 3.

```
function init() {
    google.earth.createInstance('map3d', initCB, failureCB);
}
```

Quadro 3 - Função de inicialização do plug-in do Google Earth.
 Fonte: <http://code.google.com/intl/pt-BR/apis/earth/documentation/>

É perceptível após análise do Quadro 3, que no mesmo ocorre a criação de uma instância da API do Google *Earth*, dentro da `<div>` de id `map3d` criada anteriormente.

Também podem ser percebidos no Quadro 3 outros dois parâmetros repassados. Os mesmos são "initCB" e "failureCB". Estes parâmetros representam duas funções: a função `initCB` será chamada se a instância da API for criada com êxito, e a função `failureCB` se a criação da API falhar. Para melhor compreensão serão definidas estas duas funções. Sendo a função `initCB` ilustrada no Quadro 4 e a função `failureCB` no Quadro 5.

```
function initCB(instance) {
    ge = instance;
    ge.getWindow().setVisibility(true);
}
```

Quadro 4 - Estrutura do método `initCB`.
 Fonte: <http://code.google.com/intl/pt-BR/apis/earth/documentation/>

De acordo com a documentação da API:

A função de retorno de chamada bem-sucedida terá todos os códigos necessários para configurar a sua experiência de "primeira execução" - todos os objetos e visualizações que aparecerão primeiro quando a sua instância do plug-in for carregada no navegador. Essa função deve conter o método `getWindow.setVisibility`, configurando a visibilidade da janela para `true`, de forma que o plug-in esteja visível dentro do DIV.(2010).

Nota-se que o método `initCB` contém as configurações que permitirão a primeira execução da API do Google *Earth* no navegador do usuário, tornando possível a exibição da API, através do método `getWindow.setVisibility`.

Será agora analisado o método `failureCB`, que é o outro método que pode ser chamado no momento da instanciação da API. Este método está ilustrado no Quadro 5.

```
function failureCB(errorCode) {  
}
```

Quadro 5 - Estrutura do Método `failureCB`.

Fonte: <http://code.google.com/intl/pt-BR/apis/earth/documentation/>

Nota-se após leitura do Quadro 5, que dentro do método `failureCB` pode-se tratar o erro da maneira desejada. Pode por exemplo ser apresentada uma mensagem de erro para o usuário através de um `alert`. Este método será chamado quando ocorrer uma falha no momento de instanciar a API do Google *Earth*.

Para finalizar a definição da API do Google *Earth*, está detalhado um exemplo completo de funcionamento da mesma no Quadro 6.

```
<html>  
<head>  
  <title>Sample</title>  
  <script src="http://www.google.com/jsapi?key=ABCDEFGH" >/script>  
  <script type="text/javascript">  
    var ge;  
    google.load("earth", "1");  
  
    function init() {  
      google.earth.createInstance('map3d', initCB, failureCB);  
    }  
  
    function initCB(instance) {  
      ge = instance;  
      ge.getWindow().setVisibility(true);  
    }  
  
    function failureCB(errorCode) {  
    }  
  
    google.setOnLoadCallback(init);  
  </script>  
</head>  
<body>  
  <div id="map3d" style="height: 400px; width: 600px;"></div>  
</body>  
</html>
```

Quadro 6 - Exemplo completo do funcionamento a API Google *Earth*.

Fonte: <http://code.google.com/intl/pt-BR/apis/earth/documentation/>

O Quadro 6 demonstra um exemplo completo do funcionamento da API do Google *Earth*. Conforme percebe-se, estão contidas nele todas as definições detalhadas até agora. Desde o momento de inserção da chave gerada, criação da instância, os métodos necessários, até o momento onde a API é inserida dentro do `div` de `id` igual a “map3d”.

4.2 API AJAX de Idioma

Outra API criada pela Google, para ser incorporada pelos desenvolvedores nas suas aplicações é a API AJAX de Idioma.

Conforme a documentação da API (disponível em <http://code.google.com/intl/pt-BR/apis/language/>), ela é composta por ferramentas poderosas, tornando mais fácil a comunicação entre usuários do mundo todo. Entre estas ferramentas estão:

- Tradução entre centenas de idiomas, aos pares;
- Provê um teclado virtual para o usuário de acordo com a língua escolhida;
- Auxilia aos usuários não nativos de determinada língua a pronunciar as palavras.

4.2.1 Utilização da API

Para poder utilizar esta API em suas aplicações, o desenvolvedor deve possuir uma conta Google. Caso ainda não possua o usuário deverá criá-la, através do endereço: <https://www.google.com/accounts/NewAccount?hl=pt-BR>

Após criar a sua conta Google, o desenvolvedor deve acessar o seguinte endereço: <https://code.google.com/apis/console/?api=translate&pli=1#welcome>. Deve então acessar o link “Add project”. Logo depois o usuário deve concordar com os termos.

4.2.2 Funcionamento da API

A API AJAX de Idioma é uma ferramenta que automaticamente traduz um texto de uma língua para a outra. O texto de origem é o qual será traduzido, a língua de origem é a língua em que o texto de origem está escrito e a língua de destino é a qual o texto será traduzido. Existem os seguintes métodos que podem ser invocados para a utilização da API:

- `translate`: traduz o texto de origem, na língua de origem, para a língua de destino e é mapeado através do método `get` do padrão `http`;
- `languages`: lista as línguas de origem e destino suportadas pelos métodos de tradução e é mapeado através do método `get` do padrão `http`;
- `detect`: este método detecta a língua do texto de origem.

Existem dois métodos de invocar a API: utilizando REST (*Representational State Transfer*) diretamente e utilizando REST diretamente com *javascript*.

4.2.3 Invocando a API através de *Javascript*

A API pode ser chamada através de *javascript*, usando o parâmetro de consulta de retorno e uma função de callback. Isto permite escrever aplicativos ricos que exibem dados e traduzem textos sem escrever qualquer código do lado do servidor. Segue exemplo no Quadro 7 , onde o texto traduzido é colocado abaixo do texto original.

```

<html>
<head>
  <title>Translate API Example</title>
</head>
<body>
  <div id="sourceText">Hello world</div>
  <div id="translation"></div>
  <script>
    function translateText(response) {
      document.getElementById("translation").innerHTML += "<br>" + response.data.translations[0].translatedText;
    }
  </script>
  <script>
    var newScript = document.createElement('script');
    newScript.type = 'text/javascript';
    var sourceText = escape(document.getElementById("sourceText").innerHTML);
    var source = 'https://www.googleapis.com/language/translate/v2?key=INSERT-YOUR-KEY&source=en&target=de&callback=translateText&q=' + sourceText;
    newScript.src = source;

    // When we add this script to the head, the request is sent off.
    document.getElementsByTagName('head')[0].appendChild(newScript);
  </script>
</body>
</html>

```

Quadro 7 - Tradução de texto através da API Google *Translate*.

Fonte: http://code.google.com/intl/pt-BR/apis/language/translate/v2/getting_started.html

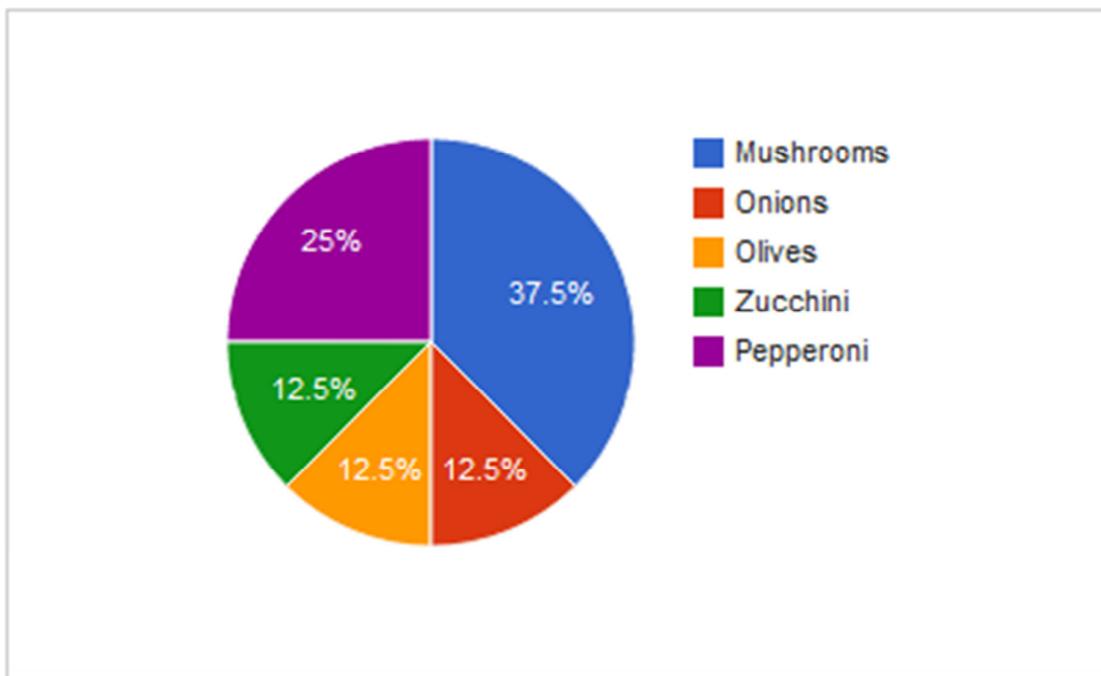
Nota-se após análise do Quadro 7, que a primeira `div` conterá o texto de origem, e a segunda o texto traduzido.

Conforme pode ser percebido também no Quadro 7, o texto de destino é repassado em uma conexão segura através de um URL (*Uniform Resource Locator*) e o mesmo retorna para o código através de uma função de call-back.

4.3 API Google *Chart*

Esta API tem por objetivo a geração de gráficos nas aplicações. A documentação da API se encontra disponível no endereço: http://code.google.com/intl/pt-BR/apis/chart/interactive/docs/quick_start.html.

Com o objetivo de tornar mais claro o funcionamento da API está ilustrado um exemplo de funcionamento nos Quadros 8 e 9.



Quadro 8 - Gráfico gerado pela API Google *Chart*.

Fonte: http://code.google.com/intl/pt-BR/apis/chart/interactive/docs/quick_start.html

Através da leitura do Quadro 8, pode ser observado um gráfico gerado pela API Google *Chart*. Nele constam as categorias, e o gráfico propriamente dito, com os percentuais de cada categoria.

```

<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">

  // Load the Visualization API and the piechart package.
  google.load('visualization', '1', {'packages':['corechart']});

  // Set a callback to run when the Google Visualization API is loaded.
  google.setOnLoadCallback(drawChart);

  // Callback that creates and populates a data table,
  // instantiates the pie chart, passes in the data and
  // draws it.
  function drawChart() {

    // Create our data table.
    var data = new google.visualization.DataTable();
    data.addColumn('string', 'Topping');
    data.addColumn('number', 'Slices');
    data.addRows([
      ['Mushrooms', 3],
      ['Onions', 1],
      ['Olives', 1],
      ['Zucchini', 1],
      ['Pepperoni', 2]
    ]);

    // Instantiate and draw our chart, passing in some options.
    var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
    chart.draw(data, {width: 400, height: 240});
  }

```

Quadro 9 - Desenvolvimento de um gráfico através da API Google *Chart*.

Fonte: http://code.google.com/intl/pt-BR/apis/chart/interactive/docs/quick_start.html

O Quadro 9 contém o código utilizado para gerar o gráfico constante no Quadro 8. Assim como as outras APIs citadas até este momento, o mesmo também é gerado através de código *javascript*.

Logo no início a API é carregada, através do método `google.load`. Após isto é criada a função `drawChart`, que é responsável por desenhar o gráfico. Dentro desta função são adicionadas as colunas e as linhas do gráfico, com seus respectivos valores.

Finalmente, é chamado o método `PieChart`, responsável por apresentar o gráfico na tela.

4.4 API AJAX de Pesquisa

A API AJAX de Pesquisa é utilizada para incluir uma caixa de pesquisa nos sistemas, através da qual são feitas pesquisas no Google. A documentação pode ser encontrada no endereço: <http://code.google.com/intl/pt-BR/apis/websearch/docs/>.

4.4.1 Definição da API

Para utilizar a API de Pesquisa em seu sistema, o desenvolvedor deverá incluir a seguinte tag na sua página web: (<https://www.google.com/JSAPI>). Esta inclusão permite carregar várias bibliotecas através do método `google.load("API", "versão")`.

Com o objetivo de tornar mais clara a compreensão do funcionamento desta API está ilustrado um exemplo no Quadro 10.

```

<script src="https://www.google.com/jsapi"
  type="text/javascript"></script>
<script language="Javascript" type="text/javascript">
  /*
google.load('search', '1');

function OnLoad() {
  // Create a search control
  var searchControl = new google.search.SearchControl();

  // Add in a full set of searchers
  var localSearch = new google.search.LocalSearch();
  searchControl.addSearcher(localSearch);
  searchControl.addSearcher(new google.search.WebSearch());
  searchControl.addSearcher(new google.search.VideoSearch());
  searchControl.addSearcher(new google.search.BlogSearch());
  searchControl.addSearcher(new google.search.NewsSearch());
  searchControl.addSearcher(new google.search.ImageSearch());
  searchControl.addSearcher(new google.search.BookSearch());
  searchControl.addSearcher(new google.search.PatentSearch());

  // Set the Local Search center point
  localSearch.setCenterPoint("New York, NY");

  // tell the searcher to draw itself and tell it where to attach
  searchControl.draw(document.getElementById("searchcontrol"));

  // execute an initial search
  searchControl.execute("VW GTI");
}
google.setOnLoadCallback(OnLoad);
</pre>
</div>
<div data-bbox="168 439 597 469" data-label="Text">
<p>Quadro 10 - Funcionamento da API de Pesquisa Google.<br/>
  Fonte: <a href="http://code.google.com/intl/pt-BR/apis/websearch/docs/">http://code.google.com/intl/pt-BR/apis/websearch/docs/</a></p>
</div>
<div data-bbox="168 491 876 534" data-label="Text">
<p>Pode-se notar através da visualização do Quadro 10, que as funcionalidades da API de Pesquisa Google também são executadas via código <i>javascript</i>.</p>
</div>
<div data-bbox="168 541 876 583" data-label="Text">
<p>Logo no início do código a biblioteca da API é referenciada, no comando <i>src</i> na definição do script. Esta inclusão permite a utilização dos métodos da API.</p>
</div>
<div data-bbox="168 590 876 658" data-label="Text">
<p>Mais abaixo está presente a função <i>onLoad</i>. Neste método são carregados os tipos de pesquisa que serão utilizados, como por exemplo: pesquisa na web; pesquisa de vídeos e blogs, entre outros.</p>
</div>
<div data-bbox="168 664 876 757" data-label="Text">
<p>Mais abaixo, através da chamada ao método <i>LocalSearch.setCenterPoint</i> são passados os parâmetros a serem pesquisados, e abaixo no método <i>searchControl.execute</i> a pesquisa é realizada.</p>
</div>
<div data-bbox="168 806 417 825" data-label="Section-Header">
<h4>4.5 API Google <i>Static Maps</i></h4>
</div>
<div data-bbox="168 850 876 893" data-label="Text">
<p>A API Google <i>Static Maps</i> permite a incorporação de uma imagem do Google <i>Maps</i> à sites <i>web</i> sem a necessidade de programação <i>javascript</i> ou de</p>
</div>
```


Na Figura 8 está apresentada imagem gerada pela API Google *Static Maps* através dos parâmetros repassados a URL informada no Quadro 11.

Esta API retorna uma imagem em resposta à solicitação HTTP (*Hypertext Transfer Protocol*) gerada pelo endereço. Para cada solicitação, você pode especificar a localização do mapa, o tamanho da imagem, o nível de zoom, o tipo do mapa e o posicionamento de marcadores opcionais em locais do mapa. Além disso, você pode nomear os seus marcadores usando caracteres alfanuméricos, para que possa utilizá-los em uma "legenda".

Outro ponto importante a ser destacado é que o uso da API está sujeito à um limite diário de consulta de 1.000 solicitações diferentes de imagem por visualizador.

4.6 API Google *Talk*

De acordo com a documentação da API Google *Talk*, disponível em <http://code.google.com/intl/pt-BR/apis/talk/>:

O Google *Talk* é um serviço de mensagens instantâneas construído sobre protocolos abertos. O Google abriu seu serviço de IM para que você possa conectar seus próprios aplicativos cliente ao serviço do Google *Talk*. Você também pode conectar (associar) o seu serviço ao nosso. (2010).

O Google *Talk* foi desenvolvido pela Google com o objetivo de facilitar a troca de mensagens instantâneas entre seus usuários, facilitando a comunicação.

Após leitura da citação, nota-se então que a API Google *Talk* foi desenvolvida para que o programador possa disponibilizar na sua aplicação uma maneira com que os usuários se conectem ao Google *Talk*.

Um dos pontos positivos em relação à utilização desta API, é que o usuário não necessita fazer o download e instalação do Google *Talk* no seu computador. O mesmo somente acessa o serviço através da API.

A implementação dos mecanismos de comunicação nesta APIs é feita através do protocolo de comunicação XMPP (Extensible Messaging and Presence

Protocol), estando sua documentação disponível no endereço <http://xmpp.org/xmpp-protocols/protocol-namespaces/>.

4.7 Utilização das APIs AJAX de Idioma, de Pesquisa, Google *Chart*, Google *Earth*, Google *Maps* e Google *Talk* em um Sistema Colaborativo

Para que um trabalho possa ser executado de maneira colaborativa, é importante que haja uma constante comunicação entre os envolvidos. Através da comunicação são definidos os objetivos e os passos que serão seguidos para a conquista destes objetivos. Na comunicação também são definidos os papéis dos envolvidos e as tarefas são distribuídas de acordo com a formação e as habilidades individuais.

No dia a dia a comunicação é feita através da utilização de sinais e símbolos, assim como também com a utilização do corpo para expressar a mensagem que deve ser repassada ao receptor.

Assim também deve funcionar com um *Software* Colaborativo, ou seja, o mesmo deve prover recursos para facilitar a comunicação e fazer com que a compreensão da mensagem a ser passada seja a melhor possível.

Vale destacar aqui que um *Software* Colaborativo deve ser um ambiente completo para o trabalho do usuário. O mesmo deve prover recursos para que ele possa realizar a maioria das suas tarefas dentro do próprio ambiente. O *Software* deve prover recursos para manipular diferentes tipos de arquivos, como por exemplo projetos de arquitetura, planilhas e textos. O sistema deve ser o principal ambiente de trabalho do usuário.

Neste ponto entra a utilização das APIs AJAX de Idioma, AJAX de Pesquisa, Google *Chart*, Google *Earth*, Google *Maps* e Google *Talk* que visam aumentar a quantidade de funcionalidades disponíveis.

4.8 Pontos Positivos na utilização das APIs

A utilização de APIs traz uma série de vantagens no desenvolvimento de um sistema, entre elas:

- Fácil utilização – através de poucas linhas de código o desenvolvedor pode adaptar uma API ao seu sistema;
- Não necessidade de conhecimento do funcionamento da API profundamente – o desenvolvedor não necessita conhecer a fundo o funcionamento da API e como ela foi desenvolvida, o mesmo somente a utiliza para atingir um objetivo;
- Fácil adaptação – como as mesmas podem ser instanciadas e utilizadas através de poucas linhas de código *javascript*, torna-se claro que podem ser adaptadas facilmente à outras linguagens de programação para a *web*;
- Outro ponto positivo em relação a utilização de APIs é que o desenvolvedor não precisa se preocupar com questões de atualização da mesma, pois isso fica à cargo de quem a criou.

Nota-se então após a descrição dos pontos positivos em relação à utilização de APIs que o desenvolvedor pode fazer melhores sistemas em menos tempo, visto que ele pode se focar em pontos importantes do funcionamento, como por exemplo as regras de negócio e particularidades específicas, e no momento de incrementar o sistema com ferramentas que interagem com o usuário o mesmo não precisa desenvolvê-las, mas sim somente utilizá-las.

Podem ser destacadas aqui como desvantagens na utilização de APIs o não conhecimento da lógica da mesma pelo programador em alguns casos, como em algum momento em que ele precise de uma certa personalização da API em seu sistema e também a solução de erros na lógica da API que ela possa vir a apresentar.

4.9 APIs em um Sistema Colaborativo

Quando se pensa no desenvolvimento de um Sistema Colaborativo, existem alguns pontos importantes que devem ser considerados:

- A comunicação deve ter papel primordial, e com isto, ferramentas devem ser disponibilizadas para que a troca de informações se torne clara e constante;

- O sistema deve fornecer ferramentas que auxiliem os usuários nos seus trabalhos, como, por exemplo, para a pesquisa de informações na Internet;
- O seu *layout* deve ser elaborado de forma que torne o trabalho do usuário o mais produtivo possível. As ferramentas mais importantes devem estar de fácil acesso e sua manipulação deve ser simples e objetiva.

Outro ponto interessante a ser pensando em um Sistema Colaborativo diz respeito às grandes distâncias que podem separar os colaboradores de uma organização. Pensando nisso o sistema pode prover ferramentas para localização tais como mapas e fotos via satélite.

Um dos grandes objetivos de um Sistema Colaborativo é criar um ambiente de trabalho virtual. Mesmo os usuários estando separados por grandes distâncias, devem se sentir como se estivessem trabalhando todos ao mesmo tempo em um mesmo departamento de uma empresa.

Neste ponto entra a utilização das APIs. O Objetivo da utilização das mesmas em um Software Colaborativo é prover funcionalidades extras no sistema, que serão utilizadas para os usuários e que auxiliarão na troca de informações e no aumento da produtividade.

A partir deste momento será definido como cada uma das APIs da Google que fazem parte deste trabalho podem ser utilizadas no desenvolvimento de um Software Colaborativo e podem auxiliar o usuário final.

4.9.1 API AJAX de Idioma em um Sistema Colaborativo

Em grandes corporações a comunicação entre os seus funcionários e membros de outras organizações em países diferentes e com línguas diferentes é constante.

Isto se faz necessário devido ao comércio de produtos, através da exportação ou até mesmo da importação de bens e matéria-prima.

Com o objetivo de tornar esta comunicação mais simples aos usuários não fluentes em determinados idiomas entra neste contexto a API AJAX de Idioma do Google.

Esta API possui funcionalidades muito bem elaboradas. Entre estas funcionalidades está a que auxilia o usuário na pronúncia das palavras da língua com a qual o mesmo não é familiarizado. Outro ponto importante é que no momento em que uma palavra é traduzida a API informa os diversos significados que a mesma possui.

Esta API se baseia na ferramenta da Google chamada de Google Tradutor, disponível no endereço: <http://translate.google.com>. Após o uso constante desta ferramenta, percebe-se que o texto gerado após a tradução é de boa qualidade.

O objetivo desta API não é somente facilitar a comunicação entre os seus usuários, mas também outro ponto muito importante diz respeito à tradução de textos, artigos, revistas e outros.

Com esta API incorporada ao *Software* Colaborativo o usuário pode utilizá-la para traduzir material bibliográfico que pode auxiliar no desenvolvimento de seus trabalhos e na aquisição de conhecimentos, traduzir também mensagens que serão repassadas a pessoas que falam idiomas diferentes e possibilitar à conversa fluir livremente.

O uso desta API em um *Software* Colaborativo seria muito benéfico. Para a tradução de textos o usuário poderia realizá-la dentro do próprio sistema, com a importação de documentos e a realização da tradução logo em seguida, não necessitando assim da utilização de outras ferramentas, tornando o trabalho mais prático e mais rápido.

Assim que traduzidos os documentos já podem ser salvos para serem utilizados no mesmo momento ou em outras oportunidades.

4.9.2 API AJAX de Pesquisa em um Software Colaborativo

Sabe-se da importância pela busca de novas informações a cada dia, seja para resolver problemas que surgem ou mesmo para adquirir conhecimentos extras. A pesquisa por novas informações é uma das atividades mais realizadas pelos usuários na Internet diariamente.

O mesmo ocorre no dia a dia de trabalho, onde em alguns momentos necessitamos utilizar uma ferramenta de buscas que nos auxilie a adquirir informações para dar continuidade a realização dos trabalhos.

Com este objetivo, a implantação da API AJAX de Pesquisa em um Software Colaborativo seria muito benéfica. No momento que surgisse a dúvida, o usuário já poderia através do próprio ambiente do Software Colaborativo buscar por soluções para suas respostas.

Esta API também possui métodos que fazem buscas específicas. Com a utilização destes métodos, poderiam então ser implantados diferentes módulos de pesquisa em um Software Colaborativo, como, por exemplo, módulos para pesquisa de imagens, vídeos e notícias, conforme a necessidade.

A implantação dos módulos de pesquisa em um Sistema Colaborativo tornaria muito mais ágil o trabalho dos usuários. Pode-se pensar, por exemplo, em um Sistema Colaborativo sendo utilizado por desenvolvedores para criar um sistema em uma determinada linguagem de programação. Assim que surgisse uma dúvida específica sobre um método da linguagem, por exemplo, o programador já poderia estar utilizando da API de Pesquisa neste mesmo momento para sanar a sua dúvida e continuar com suas atividades, evitando perda de tempo.

Outra funcionalidade interessante a ser implementada em um Sistema Colaborativo seria uma base de conhecimentos, onde nela podem ser salvos os materiais mais interessantes coletados através de pesquisas pela API. Assim cada material poderia estar relacionados com alguns problemas surgidos e que o mesmo auxiliou na resolução. Desta maneira, no momento que surgisse uma dúvida o usuário poderia primeiramente consultar esta base de conhecimento para verificar se outro usuário já não enfrentou este mesmo problema e como o resolveu. Se a sua dúvida não estivesse relacionada à nenhum material salvo anteriormente, o mesmo seria então automaticamente redirecionado para a API para buscar na Internet relacionadas à sua dúvida.

Assim com as outras APIs da Google, a API AJAX de Pesquisa é de fácil implantação, pois possui poucos métodos e com objetivos bem definidos e é utilizada através da linguagem de programação *javascript*, tornando seu uso adaptável facilmente à outras linguagens para a *web*.

4.9.3 API Google *Chart* em um Sistema Colaborativo

Constantemente nas organizações os superiores devem repassar informações aos seus subordinados. Dentre estas informações repassadas estão as metas e os resultados alcançados.

Uma das formas mais comuns de se repassar resultados e metas é através da geração e exibição de gráficos. Gráficos tem a vantagem de repassar uma grande quantidade de informação de uma maneira simples. Neste ponto entra a utilização da API Google *Chart*.

Assim como as outras APIs da Google, seu uso e sua implantação são simples, é bastante adaptável à diversas linguagens de programação para a *web*, devido ao fato de ser utilizada com *javascript* e possui poucos métodos e com objetivos bem definidos.

Pensando na utilização desta API em um Sistema Colaborativo, poderia ser desenvolvido um módulo somente de gráficos no sistema. Neste módulo estariam registrados os gráficos com as metas de venda para o ano por exemplo. Em outra categoria neste módulo poderia ser desenvolvido de tal maneira que o sistema colaborativo fosse integrado com o sistema de vendas da empresa. Assim conforme as vendas fossem aumentando os gráficos também teriam seus valores alterados, possibilitando aos vendedores e supervisores acompanharem em tempo real as vendas.

Como um dos objetivos de um Sistema Colaborativo é integrar uma grande quantidade de ferramentas, evitando que o usuário tenha que sair do mesmo para desenvolver outros trabalhos, poderia ser desenvolvido um módulo especial para a geração de gráficos. Assim como existem softwares disponíveis que geram gráficos, poderia ser implantado o próprio módulo de geração de gráficos do Sistema Colaborativo. Assim os usuários, tais como gerentes e supervisores, poderiam gerar gráficos com as informações que julgassem adequadas, salvá-los e utilizá-los em documentos ou apresentações.

Isto tornaria mais ágil os trabalhos, pois o usuário não precisaria buscar por outra ferramenta extra para a geração destes gráficos.

4.9.4 API Google *Earth* em um Sistema Colaborativo

Esta API é baseada no Google *Earth*, desenvolvido pela Google. O objetivo da mesma é disponibilizar aos usuários o mapeamento do globo terrestre inteiro, com fotos tiradas via satélite.

Atualmente tem sido inúmeros os seus usos, que vão desde marcações que os usuários utilizam para disponibilizar as imagens de suas empresas, até marcações para mostrar onde residem.

Pensando no caso de um Sistema Colaborativo, a mesma auxiliaria os usuários na localização de endereços de clientes por exemplo. Outro ponto positivo é que através do traçado de linhas nos territórios é possível realizar o cálculo de distâncias.

Seu uso seria muito útil por supervisores e gerentes para distribuir a área de abrangência dos vendedores por exemplo. Com estas informações disponibilizadas no sistema, um novo vendedor poderia conhecer melhor as suas regiões de atuação através de consultas aos clientes na API.

Outra vantagem da utilização desta API, é que a mesma permite a criação de aplicativos complexos, que utilizam mapas em 3D. Olhando por este lado, os desenvolvedores poderiam disponibilizar sistemas aos usuários para os mais variados objetivos, utilizando-se dos mapas já existentes no Google *Earth*.

Uma das grandes vantagens desta API, conforme atualizações são disponibilizadas, a mesma as apresenta e as fotos territoriais tornam-se mais nítidas para o usuário.

Assim como as outras APIs, esta também é acessada através de *javascript*, e possui poucos e bem definidos métodos.

4.9.5 API Google *Static Maps* em um Sistema Colaborativo

Constantemente utilizamos mapas no nosso dia a dia, seja para localizar destinos ou mesmo efetuar divisão de regiões.

Com o objetivo de disponibilizar mapas aos usuários em um Sistema Colaborativo está a utilização da API Google *Map*.

Esta API pode ser integrada facilmente à aplicações *web* através de código *javascript*.

Uma aplicação interessante para a API em um sistema colaborativo seria um mapa associado à cada cliente da empresa. Isso facilitaria a localização por gerentes e vendedores.

Um módulo de mapas poderia estar contido no sistema. Através deste módulo o usuário do sistema entraria com o endereço do cliente ou lugar de interesse e logo em seguida o sistema geraria um mapa com a localização do estabelecimento na tela.

Um dos grandes pontos positivos desta API é que a mesma disponibiliza mapas em diversos formatos, tais como: mapa padrão, imagem via satélite, mapa de relevo físico, mostrando terreno e vegetação e imagem híbrida contendo mapa e imagem via satélite. Pensando neste ponto estas diversas opções poderiam ser disponibilizadas ao usuário do sistema.

Como vários tipos de mapas estão disponíveis, é grande a quantidade de setores que esta API poderia atender, desde setores de vendas até setores que precisam realizar a medição de áreas, por exemplo, como da construção civil para a divisão de lotes.

4.9.6 Utilização da API Google *Talk* em um Sistema Colaborativo

Um dos pontos principais que deve existir em um Sistema Colaborativo é a facilidade na comunicação entre seus usuários.

Com este objetivo está a inserção da API do Google *Talk*. O Google *Talk* é um mensageiro instantâneo desenvolvido pela Google.

Esta API poderia ser introduzida na tela principal do Sistema Colaborativo, entre as principais funcionalidades.

Através da utilização da mesma a comunicação entre os usuários seria facilitada, permitindo a troca de informações entre os colaboradores da empresa. Outro ponto positivo na utilização deste sistema é que o mesmo pouparia gastos da empresa com outras formas de comunicação, como a telefonia por exemplo.

Uma das grandes facilidades proporcionadas por esta API é que ao invés do usuário realizar o *download* e instalação do Google *Talk* na sua máquina, o mesmo poderia acessá-la somente através do Sistema Colaborativo.

Assim que acessasse o Sistema Colaborativo, logo em seguida o usuário poderia acessar a API do Google *Talk*. Tendo os outros colaboradores da empresa adicionados, a API serviria também para a entrega de recados e distribuição de tarefas, pois mesmo o usuário não estando no seu local de trabalho em determinado momento, outros colaboradores poderiam deixar recados e tarefas para o mesmo, para que pudesse realizar assim que voltasse.

4.9.7 Integração entre as APIs Google em um pequeno Sistema na linguagem PHP

Com o objetivo de tornar mais claro o funcionamento das APIs Google em um Sistema Colaborativo, foi desenvolvida uma pequena aplicação na linguagem PHP (*Hypertext Preprocessor*) na qual as seguintes APIs estão integradas: API Google *Earth*; API de Idioma, API de Pesquisa; API de Gráficos e API de Mapas.

O objetivo do desenvolvimento desta aplicação visa demonstrar como as APIs podem ser facilmente integradas em um Sistema Colaborativo desenvolvido para a plataforma web, visto que as mesmas são referenciadas à partir da linguagem *javascript*.

A Figura 9 ilustra a página inicial do sistema, que permite que o usuário acesse às APIs, conforme os links.



Figura 9 - Página inicial do sistema exemplo.

A página demonstrada na Figura 9 é um arquivo na linguagem PHP que envolve somente código HTML para a definição do seu *layout* e criação dos links para acesso às APIs

A próxima imagem, Figura 10, é a visualização da API Google *Earth* em funcionamento.

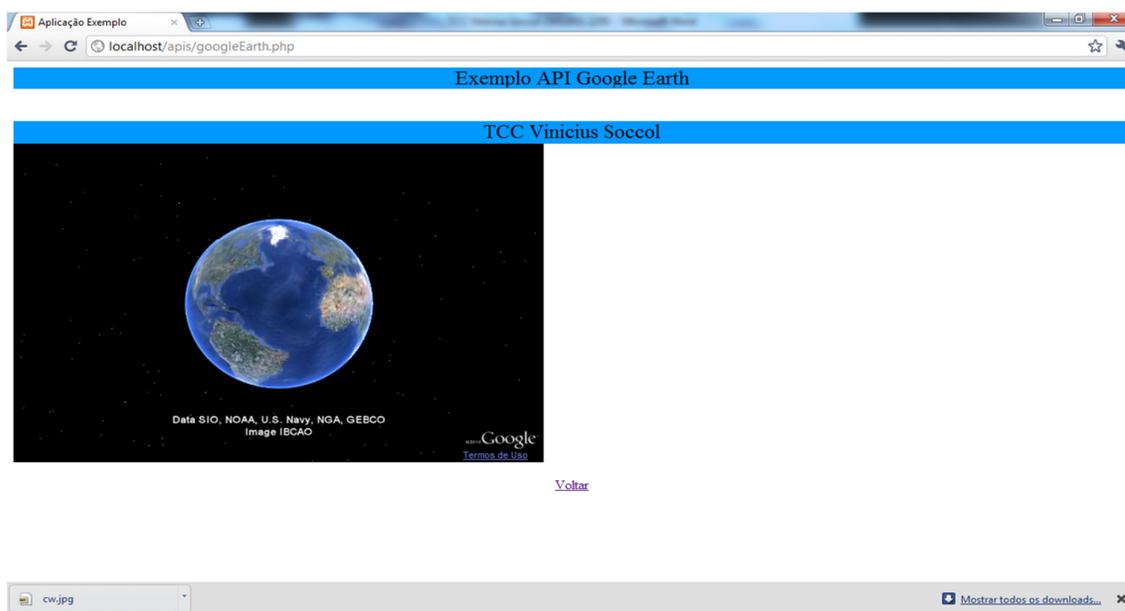


Figura 10 - API Google *Earth* em execução.

Assim como a Figura 9, a Figura 10 também é um arquivo na linguagem PHP. Para a geração do mapa da API do Google *Earth* na página é utilizada a linguagem *javascript* com funções próprias da API, conforme ilustrado na Figura 11.

```
<script src="http://www.google.com/jsapi?key=ABQIAAAAAa7HdpHkwTrbdg7ACR0F2RSBh9EtCxc7HC1u5u0ifqspwETnyxQbYu-Sr1Yd5xQdCZvLNVPTt
</script>
<script type="text/javascript">
  var ge;
  google.load("earth", "1");

  function init() {
    google.earth.createInstance('map3d', initCB, failureCB);
  }

  function initCB(instance) {
    ge = instance;
    ge.getWindow().setVisibility(true);
  }

  function failureCB(errorCode) {
  }

  google.setOnLoadCallback(init);
</script>
```

Figura 11 - Código para geração da API Google *Earth* no sistema exemplo.

Após leitura da Figura 11, percebe-se que na primeira linha de código foi inserida a chave gerada para utilização da API. A geração desta chave já foi detalhada anteriormente neste trabalho.

Nas linhas de código que se seguem estão declarados e implementados os métodos necessários para a geração e visualização da API.

A Figura 12 está mostrando a API de Idiomas em funcionamento. Nesta página é inserido o texto e escolhido o par de línguas para a tradução. Após isto, clica-se no botão “Traduzir” e então o texto traduzido é gerado na caixa abaixo.



Figura 12 - API de Idiomas.

Assim como nas outras APIs utilizadas no desenvolvimento deste sistema exemplo, na tela ilustrada na Figura 12 também foi utilizada a linguagem *javascript* para referenciar a API de Idiomas e possibilitar a tradução dos textos. Para demonstrar a lógica de funcionamento desta API estão as Figura 12 e 13.

```
<form onsubmit="return false">
  <p>Informe o texto a ser traduzido</p>
  <textarea name="translate" cols="50" rows="5"></textarea>
  <br />
  <select name="langpair">
    <option value="en|it">Inglês para Italiano</option>
    <option value="en|pt">Inglês para Português</option>
    <option value="en|es">Inglês para Espanhol</option>
    <option value="pt|en">Português para Inglês</option>
  </select>
  <input type="button" value="Traduzir" onclick="gtrans(this.form)" />
  <p id="translation" style="border:1px solid #00f;padding:5px;width:400px"></p>
</form>
```

Figura 13 - Código HTML API de Idiomas.

Na Figura 13 está ilustrado o código HTML que foi utilizado para a geração do formulário que receberá os dados de entrada pelo usuário. Este formulário contém o texto a ser traduzido, o par de idiomas para tradução selecionado pelo

usuário, o botão que chama a função *javascript* para a tradução e a caixa em baixo que apresentará o texto traduzido.

A Figura 14 representa o código *javascript* necessário para realizar a tradução do texto informado pelo usuário.

```
<script language="javascript" src="http://www.google.com/jsapi"></script>
<script language="javascript">
google.load ("language", "1");
function gtrans (f) {
  document.getElementById("translation").innerHTML = "-";
  var lang = f.langpair.value.split("|");
  google.language.translate (f.translate.value, lang[0], lang[1], function (result) {
    if (!result.error) {
      var container = document.getElementById("translation").innerHTML = result.translation;
    }
  });
  return false;
}
</script>
```

Figura 14 - Código para tradução API de Idioma.

Na primeira linha de código na Figura 14 a API de Idioma é referenciada. A tradução do texto é feita através do método `google.language.translate`, onde as línguas de origem e de destino são passadas para o método. Se o método não retornar erros então a tradução é inserida na caixa em baixo, conforme Figura 12.

A Figura 14 ilustra o funcionamento da API de Pesquisa. Assim como as outras APIs anteriores, nesta página também foi feita na linguagem PHP. Seu *layout* foi feito na linguagem HTML e foi utilizada a linguagem *javascript* para permitir a utilização da API.

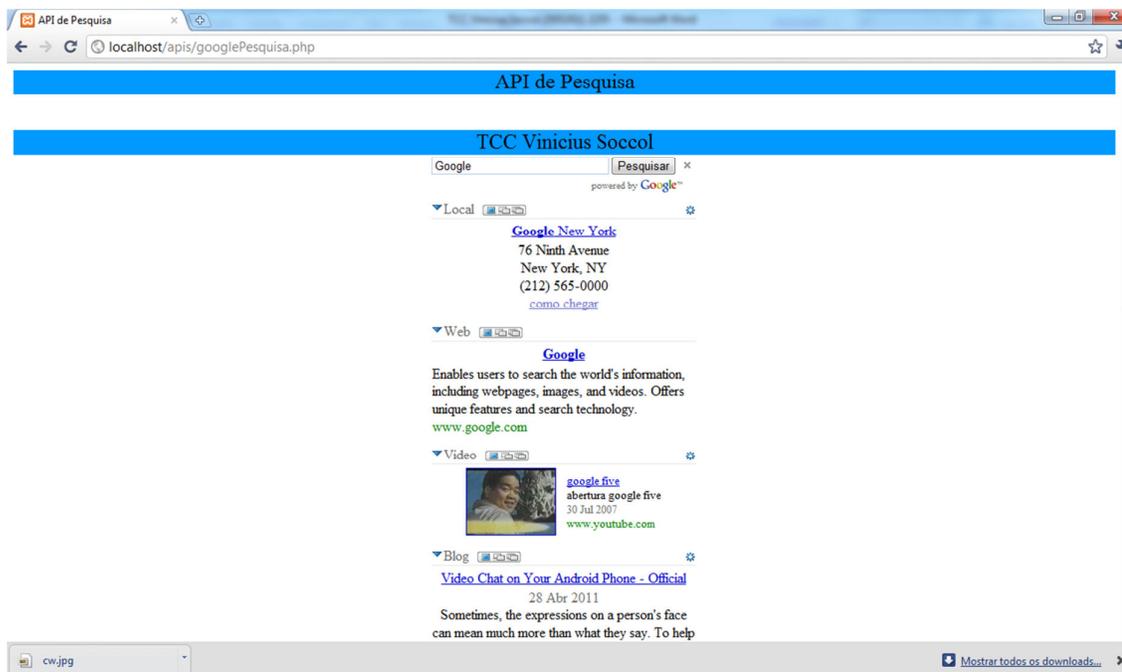


Figura 15 - API de Pesquisa em funcionamento.

Para demonstrar o funcionamento da API é necessário mostrar o código *javascript* utilizado. Este código está contido na Figura 16.

```
<script src="https://www.google.com/jsapi"
  type="text/javascript"></script>
<script language="JavaScript" type="text/javascript">
google.load('search', '1');

function OnLoad() {
  var searchControl = new google.search.SearchControl();

  var localSearch = new google.search.LocalSearch();
  searchControl.addSearcher(localSearch);
  searchControl.addSearcher(new google.search.WebSearch());
  searchControl.addSearcher(new google.search.VideoSearch());
  searchControl.addSearcher(new google.search.BlogSearch());
  searchControl.addSearcher(new google.search.NewsSearch());
  searchControl.addSearcher(new google.search.ImageSearch());
  searchControl.addSearcher(new google.search.BookSearch());
  searchControl.addSearcher(new google.search.PatentSearch());

  localSearch.setCenterPoint("New York, NY");

  searchControl.draw(document.getElementById("searchcontrol"));

  searchControl.execute("");
}
google.setOnLoadCallback(OnLoad);

//]]>
</script>
```

Figura 16 - Código *javascript* para geração da API de Pesquisa.

Na primeira linha de código na Figura 16 está a referência para a API. Logo abaixo, na chamada ao método `google.load('search', '1')` é feito o carregamento da API. Abaixo estão as chamadas aos métodos que fazem os diferentes tipos de busca como, na *web*, por vídeos, blogs, notícias e assim por diante. No método `searchControl.execute("")` a execução da pesquisa é realizada.

A próxima API utilizada para o desenvolvimento foi a API de Gráficos. Para melhor ilustrar o seu funcionamento, está imagem da mesma na Figura 17.

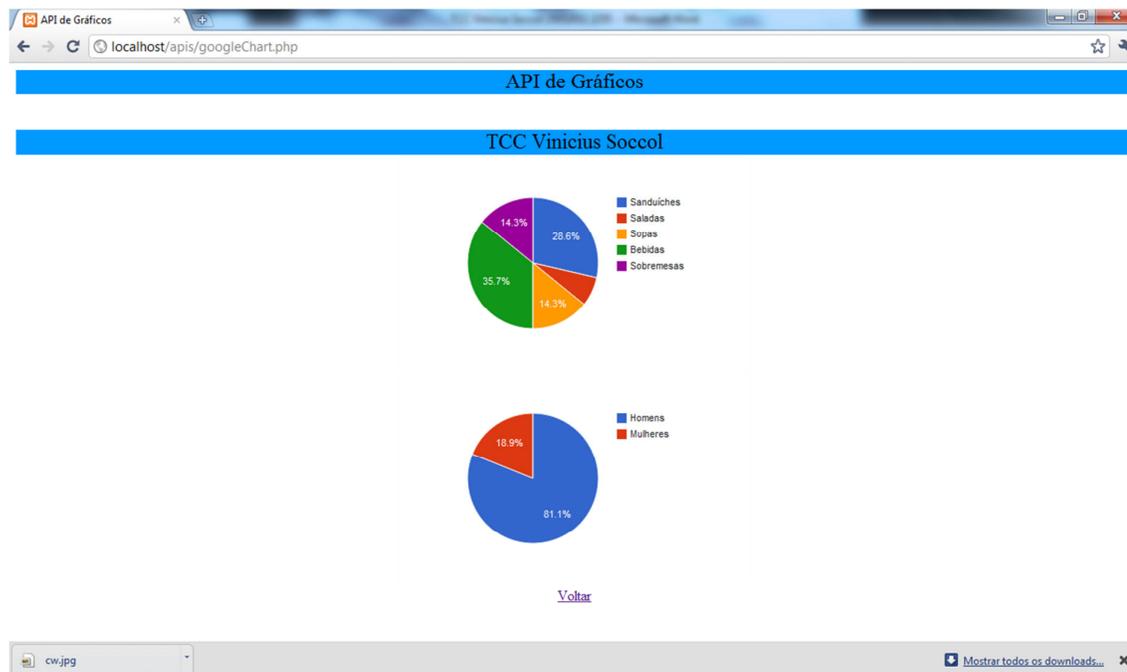


Figura 17 - API de Mapas em funcionamento.

A Figura 17 ilustra os gráficos gerados pela API de Gráficos. Para tornar mais fácil a compreensão do funcionamento desta API o código *javascript* necessário para a geração dos mesmos está contido na Figura 18.

```

<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">

google.load('visualization', '1', {'packages':['corechart']});

google.setOnLoadCallback(drawChart);

function drawChart() {

var data = new google.visualization.DataTable();
data.addColumn('string', 'Topping');
data.addColumn('number', 'Slices');
data.addRows([
  ['Sanduiches', 4],
  ['Saladas', 1],
  ['Sopas', 2],
  ['Bebidas', 5],
  ['Sobremesas', 2]
]);

var chart = new google.visualization.PieChart(document.getElementById('grafico1'));
chart.draw(data, {width: 400, height: 240});
}
</script>

```

```

<script type="text/javascript">

google.load('visualization', '1', {'packages':['corechart']});

google.setOnLoadCallback(drawChart);

function drawChart() {

var data = new google.visualization.DataTable();
data.addColumn('string', 'Topping');
data.addColumn('number', 'Slices');
data.addRows([
  ['Homens', 150],
  ['Mulheres', 35],
]);

var chart = new google.visualization.PieChart(document.getElementById('grafico2'));
chart.draw(data, {width: 400, height: 240});
}
</script>

```

Figura 18 - Código para geração dos Gráficos pela API Google *Chart*.

No lado esquerdo da Figura 18 está o código *javascript* necessário para a geração do primeiro gráfico da página, conforme Figura 17. No lado direito está o código para a geração do segundo gráfico.

Conforme pode ser percebido em ambos os casos, na primeira linha está o código que tem por objetivo referenciar a API. Abaixo são definidos os títulos e o valor de cada um. Mais abaixo é chamado o método `PieChart`, que desenha o gráfico na tela do usuário do sistema.

Por fim, a última API utilizada no desenvolvimento deste sistema é a API *Static Maps*. Para tornar mais claro o seu funcionamento a Figura 19 demonstra a API em execução.

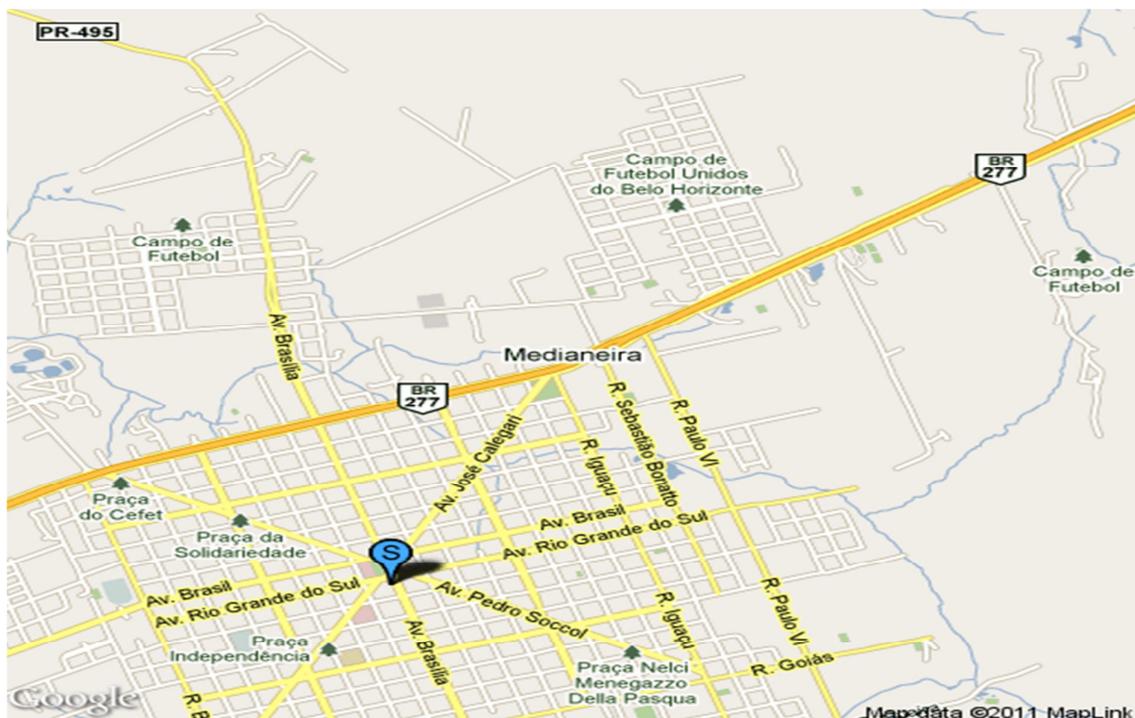


Figura 19 - Api Static Maps em execução.

Nota-se na Figura 19 que a API fez o carregamento de um mapa do centro da cidade de Medianeira na página. E

Todas estas definições foram feitas neste capítulo para demonstrar ao leitor através de um sistema exemplo desenvolvido como é simples a implantação das APIs em um Sistema Colaborativo baseado na plataforma *web*.

5 ESTUDO DE CASO: O SISTEMA COLABORATIVO SYNOVEL SPICEBIRD

Com o objetivo de tornar mais claro o funcionamento de um *Software* Colaborativo, será feito neste capítulo um estudo de caso sobre o sistema *Synovel Spicebird*.

Este estudo de caso tem por objetivo realizar uma avaliação das funcionalidades deste software e tomar conhecimento das suas vantagens.

O *download* do instalador do *software* pode ser feito no seguinte endereço: <http://www.spicebird.com/>.

Após baixar o instalador pode-se somente ir avançando nos passos que se seguem que então a instalação é concluída. A Figura 20 ilustra a tela inicial do sistema.

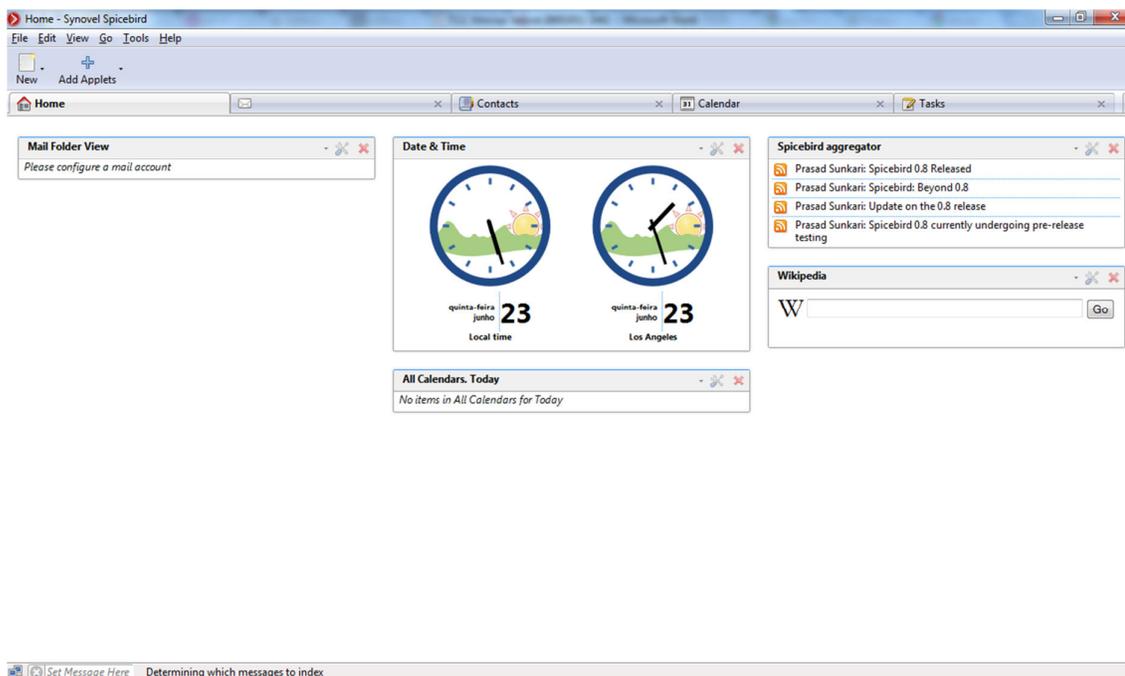


Figura 20 – Tela inicial do sistema colaborativo Synovel Spicebird

Percebe-se, logo de início, que um dos pontos importantes em um sistema colaborativo é atendido pelo *Synovel Spicebird*, no quesito onde as funcionalidades devem estar de maneira acessível ao usuário. Percebe-se isto na divisão das ferramentas do sistema por abas.

A primeira aba tem o título de “*Home* .” Nela estão localizadas as funções de *e-mail*, calendário, leitor de *feeds*, e uma caixa de textos que faz buscas diretamente

no site da *Wikipedia*. Assim como no caso da API AJAX de Idioma, as buscas geradas na *Wikipedia* são apresentadas no próprio sistema, em uma nova aba que abre automaticamente.

A próxima aba (Figura 21), refere-se a outro conjunto de aplicações colaborativas, que são os softwares de *e-mail*.

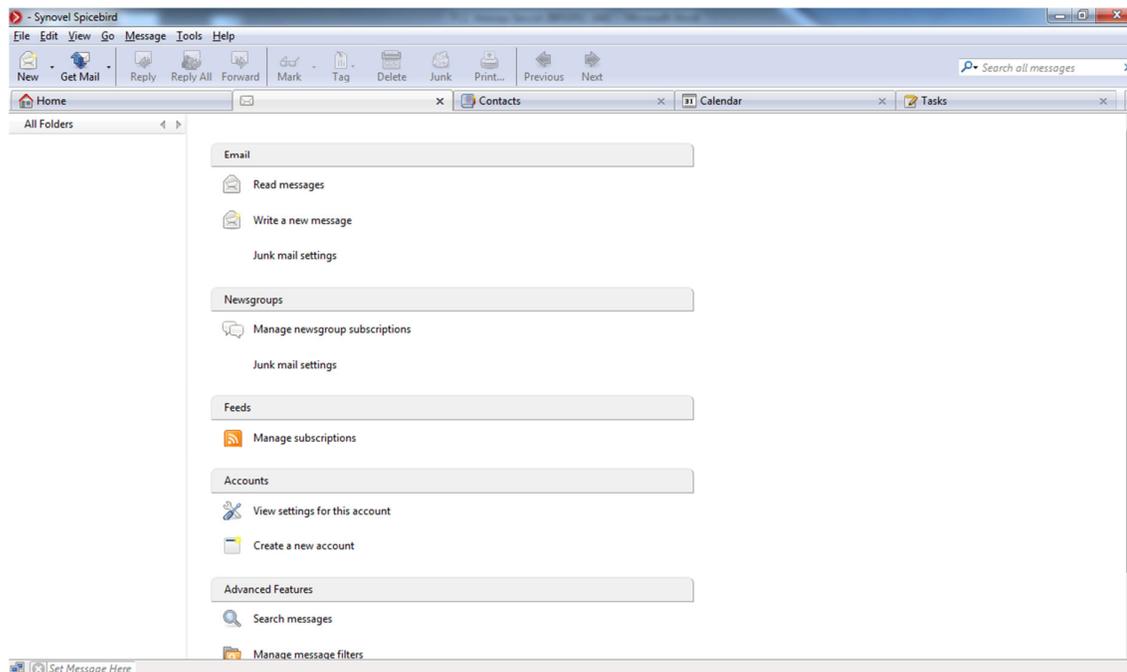


Figura 21 – Ferramentas de *e-mail* do Software *Synovel Spicebird*

Nota-se na Figura 21, que nesta aba também a distribuição das ferramentas é feita de uma forma organizada, facilitando o trabalho do usuário.

As funcionalidades são distribuídas de acordo com categorias, de maneira a agilizar as tarefas, como por exemplo: ler mensagens; escrever novas mensagens; mudar configurações de e-mail. Na mesma tela também estão opções para criar uma nova conta de e-mail e buscar por mensagens. As principais funções que geralmente estariam distribuídas em várias telas em um sistema de e-mails estão todas acessíveis na mesma tela para o usuário.

A próxima aba (Figura 22) contém funções para se trabalhar com o cadastro de contatos.

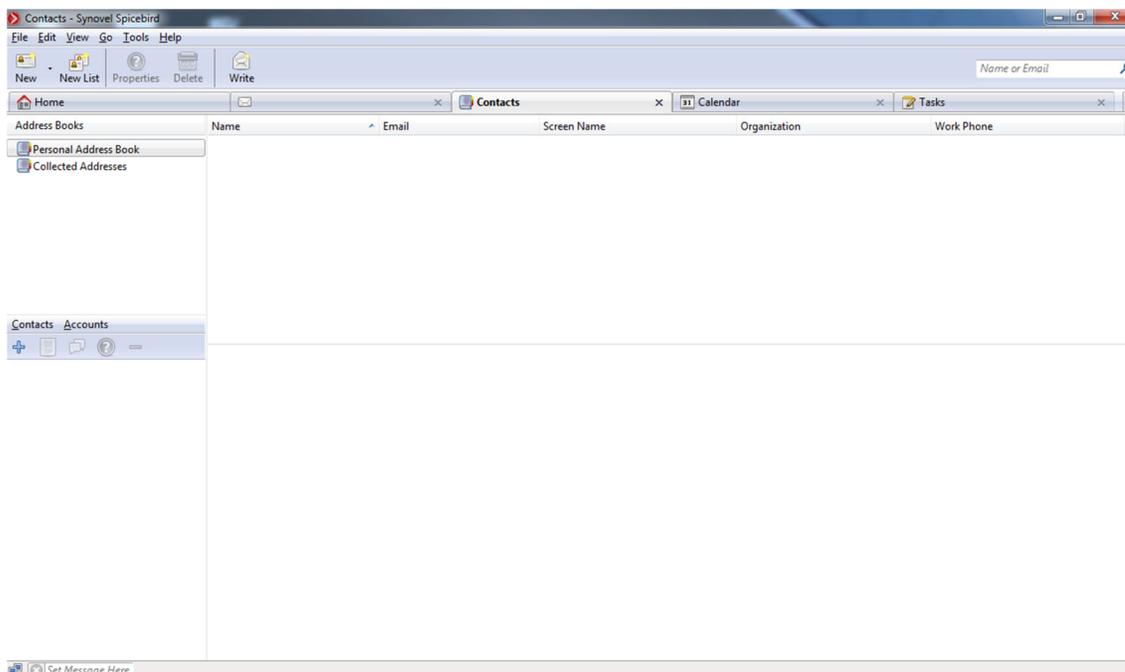


Figura 22 – Ferramentas para trabalho com contatos no *software Synovel Spicebird*

Através desta aba (Figura 22) podem ser armazenados contatos. Possui vários campos para cadastro referente aos contatos, como por exemplo: nome, e-mail, organização, fone comercial, departamento, número celular entre outros.

A próxima aba (Figura 23) tem as funções de calendário e agendamento de compromissos.

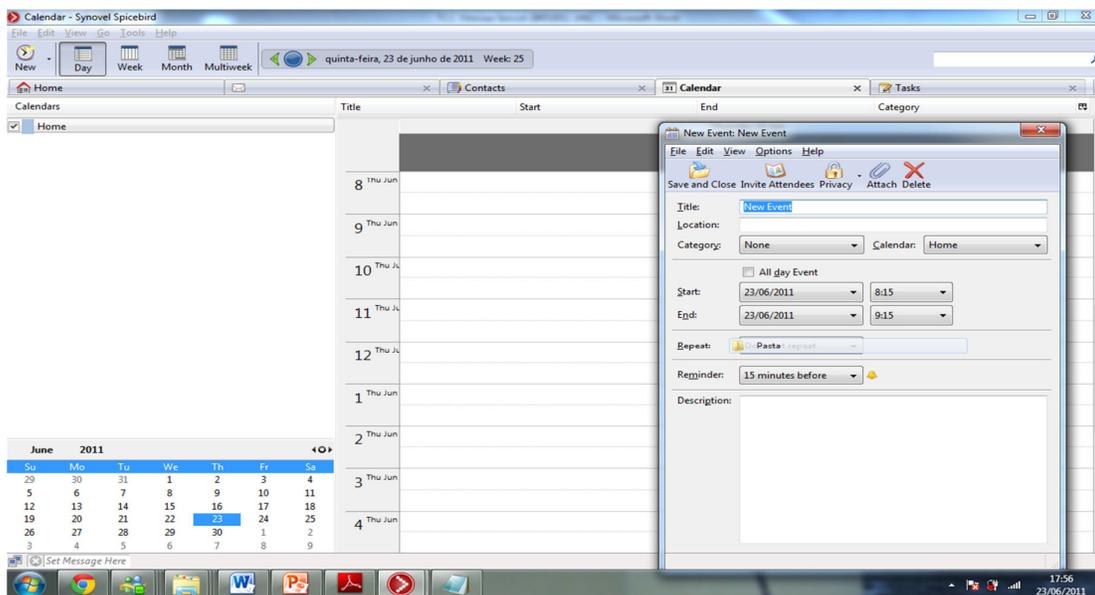


Figura 23 – Cadastro de compromissos no *Software Synovel Spicebird*

A Figura 23 ilustra a aba de compromissos com o calendário aberto. Após o usuário clicar duas vezes sobre uma data no calendário, a janela à frente, conforme demonstrado na Figura 23 permite o cadastro de um compromisso. Dentro dos dados disponíveis para o cadastro do compromisso estão o seu título, localização, horário de início e de fim e quanto tempo antes o sistema deve alertar o usuário sobre o seu compromisso.

A próxima aba (Figura 24), permite o cadastro de tarefas a serem executadas pelo usuário.

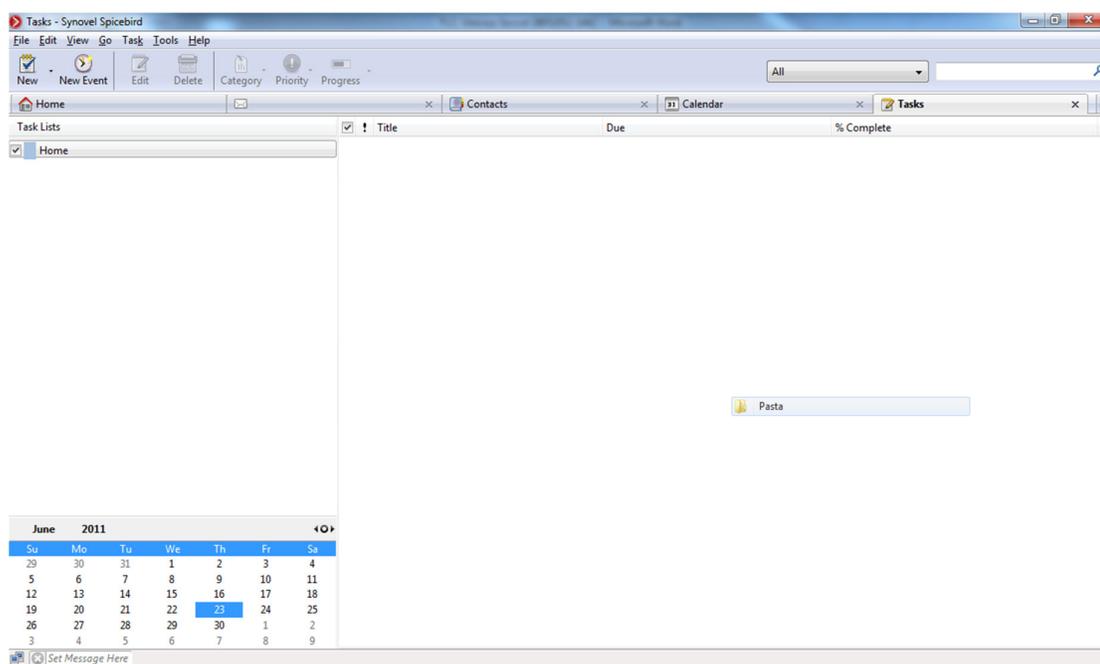


Figura 24 – Cadastro de tarefas no *Software Synovel Spicebird*

As funções contidas nesta aba são muito úteis no caso de necessidade do desenvolvimento de grandes tarefas pelo usuário. Conforme a tarefa vai sendo desenvolvida o mesmo pode ir modificando por exemplo o percentual já realizado. Assim o usuário pode acompanhar o desenvolvimento de suas tarefas todos os dias.

5.1 Vantagens na utilização do *Software Colaborativo Synovel Spicebird*

Foram percebidas grandes vantagens na utilização do sistema colaborativo *Synovel Spicebird*. Entre estas vantagens está o fato de todas as suas ferramentas estarem organizadas de forma a facilitar o trabalho do usuário. O acesso à todas é muito rápido e as suas funções são bem específicas, fazendo com que o usuário chegue aos resultados esperados rapidamente.

Um outro grande ponto positivo está ilustrado na Figura 25.

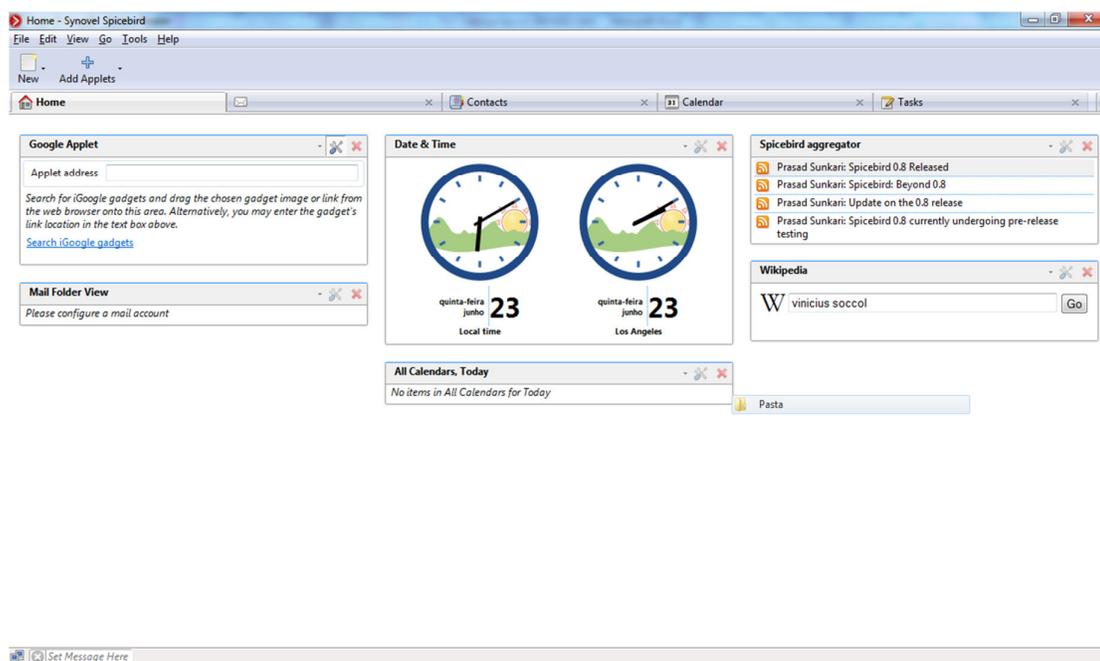


Figura 25 - Inserção de *Applets* no *Synovel Spicebird*

Esta vantagem está no fato do sistema permitir a adição de *applets* de terceiros, como pode ser percebido na caixa que diz "Google Applet". Para adicionar um aplicativo da Google basta somente inserir o endereço eletrônico deste aplicativo que o mesmo será carregado nesta mesma tela do sistema.

Isto permite ao usuário incorporar ao sistema funcionalidades que não são nativas do mesmo, e personalizá-lo de acordo com a sua preferência.

Se o usuário não souber o endereço da *applet* desejada o mesmo pode buscar pela mesma clicando no link "Search iGoogle Gadgets".

Sendo assim, a página principal pode conter os mais diversos tipos de aplicativos, e pode-se posicioná-los de acordo a ordem preferida.

Outro ponto positivo no sistema é a sua configuração. Sua instalação é muito fácil, e após instalá-lo já está pronto para ser utilizado pelo usuário.

Um outro ponto a ser considerado no sistema é que o mesmo possui um módulo voltado para o desenvolvedor. Através da barra de ferramentas, o desenvolvedor pode acessar: *Tools*, em seguida *Developer Tools* e *Error Console*. Através da tela que se abre o programador pode verificar os erros que foram gerados pelo sistema, conforme ilustra a Figura 26.

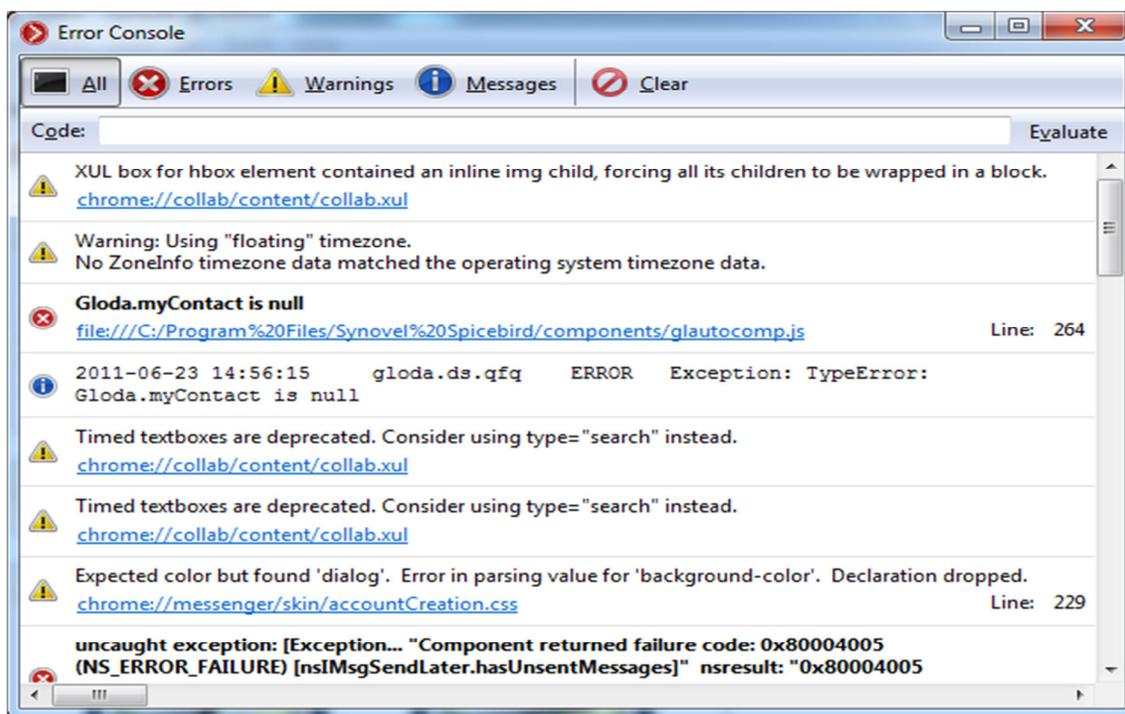


Figura 26 – Erros apresentados pelo sistema *Synovel Spicebird*

5.2 Conclusões sobre o sistema *Synovel Spicebird*

Após realizado este estudo de caso sobre o *Software Colaborativo Synovel Spicebird* pode-se dizer que realmente ele atende aos requisitos de um bom sistema colaborativo.

Todos os seus recursos são organizados de uma forma bem objetiva, através de abas. Isto facilita para a maioria dos usuários, pois já estão familiarizados com a navegação em abas que os navegadores Internet trabalham.

Outro ponto positivo é a sua fácil instalação e configuração. Poucos segundos após começar a instalar o *software* o usuário já está utilizando o mesmo. Suas ferramentas são intuitivas, e com poucos passos o resultado esperado é gerado, até mesmo para quem nunca utilizou o sistema.

Um dos maiores pontos positivos no sistema é o fato do mesmo permitir adicionar *applets* Google. Com isto o usuário pode personalizar o sistema com diversos tipos de aplicativos que preferir, desde mapas até aplicativos financeiros. Com isto o sistema pode ser utilizado por usuários dos mais diversos setores nas organizações.

Um ponto negativo na utilização do sistema é que o mesmo poderia prover mais recursos para o usuário, divididos em mais abas. Poderiam ser providos também no mesmo por exemplo sistemas para *chat* e vídeo conferência.

O sistema foca-se bastante nas ferramentas para organização de tarefas e compromissos do usuário e deixa a desejar na função de prover ferramentas que facilitem a comunicação entre os usuários, que está dentre as principais funcionalidades de um *software* colaborativo, ficando assim ela engessada, pois depende somente do *e-mail*.

6 TRABALHOS FUTUROS

O desenvolvimento deste trabalho trouxe muitos conhecimentos, tornando possível perceber quão importante é como é promissora a área dos softwares colaborativos.

Olhando-se por este lado surgiu a vontade do estudo mais profundo dos softwares colaborativos *opensource*. Um dos objetivos é começar a participar das comunidades que mantêm este tipo de sistema e mais futuramente, quando possuir conhecimentos mais complexos sobre o funcionamento dos mesmos, a viabilidade de implementar a integração de algumas APIs Google em um sistema destes, o que certamente irá trazer muitos benefícios.

Também tem-se como objetivo começar a participar de palestras e buscar conhecimentos com os estudiosos para inteirar-se mais ainda sobre *software* e trabalho colaborativo, buscando trazer novidades para auxiliar no dia a dia no local de trabalho.

7 CONCLUSÃO

Um trabalho colaborativo torna-se menos suscetível à erros, pois a visão de diversos profissionais esteve presente ao longo do seu desenvolvimento. Outro ponto importante é a redução de custos, pois esforços desnecessários são evitados.

A existência de um ambiente colaborativo em uma empresa proporciona um melhor fluxo das informações. Através do trabalho em um ambiente colaborativo também é maior a integração entre os departamentos das organizações, pois as informações ficam centralizadas em um sistema, evitando assim dados dispersos.

Pode-se concluir também que a introdução de APIs em um *Software Colaborativo* traria muitos benefícios, tanto para o desenvolvedor quanto para o usuário.

Olhando-se pelo lado do desenvolvedor o mesmo pode simplesmente em poucas linhas de código disponibilizar diversas funcionalidades extras aos usuários. Como o mesmo não precisa necessariamente conhecer o funcionamento da API profundamente, pode utilizar seu tempo para a execução de outras atividades importantes, como por exemplo lógicas de negócio importantes para o bom funcionamento do sistema, e particularidades específicas, como regras que o sistema deve seguir para gerar informações de acordo com leis e normas.

Olhando-se pelo lado do usuário o mesmo seria o maior beneficiado. Através da existência de APIs no sistema ele pode obter informações para a realização de seus trabalhos através do próprio ambiente, sem a necessidade de dispendir tempo buscando por sites ou até mesmo outros sistemas para realizar tarefas específicas.

Uma grande vantagem destas APIs é o fato das mesmas serem gratuitas e atualizadas pelo próprio desenvolvedor.

A implementação das APIs é bem simplificada. São acessadas pela linguagem *javascript* com poucas linhas de código e seus métodos são claros e com funcionalidades bem específicas.

Sua utilização é bastante simples também. Como são acessadas através da linguagem *javascript* podem ser facilmente integradas com várias linguagens de programação para a *web*.

Existem algumas APIs para desenvolvimento da parte gráfica de aplicações como por exemplo *primefaces* e *richfaces*. Ao contrário das APIs Google, estas necessitam a disponibilização de arquivos jar junto com a aplicação. Já no caso das APIs Google as mesmas somente referenciam uma URL. Esta URL contém a lógica para o funcionamento da API e a implementação de seus métodos.

O conceito de trabalho colaborativo está cada vez mais presente na atualidade, principalmente na Internet. Alguns exemplos de colaboração na Internet atualmente podem ser os fóruns e as redes sociais. Estamos na era em que a informação na Internet é disponibilizada principalmente pelos próprios usuários, e não mais por terceiros como anos atrás. Alguns autores dizem que estamos na era da colaboração.

No momento atual os sistemas existentes proporcionam meios para tornar possível a troca de informações constantemente e rapidamente.

Nas empresas para a realização deste papel entram os Sistemas Colaborativos, tais como: *e-mail*, videoconferência e os sistemas de *chat*, agindo como um ambiente de trabalho integrado, onde cada um possa colaborar com os outros membros, através do repasse de informações, que são geradas pelos seus conhecimentos, formação, habilidades particulares e experiências vividas.

Analisando pelo lado das APIs da Google, as mesmas trariam inúmeros benefícios em um *Software* Colaborativo, tais como: um melhor repasse de informações, diminuição de custos com a compra de softwares pagos, trabalhos finais mais bem elaborados devido à informação se fazer presente em todas as fases do processo.

O processo de desenvolvimento de um *software* colaborativo também torna-se mais simplificado, pois o desenvolvedor não se preocupa com a lógica das APIs, somente as implanta.

Em relação ao estudo de caso sobre o sistema *Synovel Spicebird* pode-se dizer que ele realmente cumpre com o papel que deve ser seguido por um software colaborativo, que é o de prover diversas ferramentas aos usuários, que possam ser acessadas de forma rápida e que os resultados sejam visíveis rapidamente. O sistema deixa a desejar no ponto da comunicação, pois somente ferramenta de e-mail é nativa no mesmo.

Em contrapartida possibilita a adição de *applets* da Google, tornando o mesmo bastante personalizável.

Concluindo, faz-se necessário dizer aqui que cada vez mais vem se utilizando o termo trabalho colaborativo.

Sistemas colaborativos estão cada vez mais presentes nas organizações. Um exemplo disso na área de Tecnologia da Informação são as empresas de desenvolvimento de sistemas que utilizam *software* de videoconferência para realizar apresentações de seus sistemas para possíveis clientes.

Realmente os autores e estudiosos estão com grande esperança de que a forma de trabalho colaborativo venha a substituir a individual nos próximos anos. É só assistir aos noticiários e fazer leituras na Internet que é possível perceber como grandes empresas, como por exemplo a IBM vem investindo grandes quantias no desenvolvimento de *Software Colaborativos*.

A empresa IBM está investindo no projeto *Jazz* que tem por objetivo desenvolver uma grande plataforma de *software* colaborativo.

Voltando ainda à área de Tecnologia da Informação temos o conceito de *Pair Programming*. Neste conceito dois desenvolvedores se revezam em um mesmo computador. Ora um está programando e o outro ajudando com as ideias, ora os dois se substituem.

8 REFERENCIAS

ARAUJO, José Braz de, CAMPANÁRIO, Milton de Abreu. **Trabalho Colaborativo e Groupware**. UNINOVE. 2003.

BORDENAVE, Juan Díaz. **O que é comunicação**. S. Paulo: Brasiliense, 2002 (27a. ed.).

BORGHOFF, U.M., Schlichter, J.H. **Computer-Supported Cooperative Work: Introduction to Distributed Applications**. Springer, USA, 2002.

BRINCK, T. **Groupware**. 1998. Disponível: <www.usabilityfirst.com/groupware>. Acesso: 10 Abril de 2001.

ELLIS, C. A.; GIBBS, S. J.; REIN, G. L. Groupware. **Some Issues and Experiences. Communications of the ACM, v. 34, n. 1, jan. 1991**. In: BAECKER, R. M. Readings in Groupware and Computer-Supported Cooperative Work, c. 1, p. 9-28. San Mateo (Estados Unidos): Morgan and Kaufmann, 1993.

FUKS, H., RAPOSO, A.B., GEROSA, M.A. **Engenharia de Groupware: Desenvolvimento de Aplicações Colaborativas**. XXI Jornada de Atualização em Informática, Anais do XXII Congresso da Sociedade Brasileira de Computação, V2, Cap. 3, ISBN 85-88442-24-8, 2002.

FOLDOC – Free On-line Dictionary of Computing. **Application Program Interface**. 2010. Disponível em: <<http://foldoc.org/Application+Program+Interface>> – Acesso em 12 dez. 2010, 20:25.

FUKS, H., RAPOSO, A.B., GEROSA, M.A. **Do Modelo de Colaboração 3C à Engenharia de Groupware**. Depto. de Informática, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), 2002.

GOOGLE. **API Google Earth**. 2010. Disponível em: <<http://code.google.com/intl/pt-BR/apis/earth/documentation/>> - Acesso em 12 maio 2010, 21:00.

GOOGLE. **API Google Talk**. 2010. Disponível em: <http://code.google.com/intl/pt-BR/apis/talk/talk_developers_home.html> - Acesso em 12 maio 2010, 23:00.

GOOGLE. **API Google Static Maps**. 2010. Disponível em: <<http://code.google.com/intl/pt-BR/apis/maps/documentation/staticmaps/>> - Acesso em 15 maio 2010, 22:30.

GOOGLE. **API AJAX de Idioma**. 2010. Disponível em: <<http://code.google.com/intl/pt-BR/apis/language/>> - Acesso em 11 maio 2010, 21:40.

GOOGLE. **API AJAX de Pesquisa.** 2010. Disponível em: <http://code.google.com/intl/pt-BR/apis/websearch/docs/> - Acesso em 18 maio 2010, 18:40.

GOOGLE. **API Google Chart.** 2010. Disponível em: http://code.google.com/intl/pt-BR/apis/chart/interactive/docs/quick_start.html - Acesso em 12 maio 2010, 21:00.

IBM. **Desenvolvimento de Software Colaborativo.** 2011. Disponível em: <http://www-01.ibm.com/software/br/rational/info/features/collaboration/>. Acesso em 20 de Abril de 2011.

JAMIL, G. L. **Repensando a TI na Empresa Moderna.** Rio de Janeiro: Axcel Books, 2001.

MIKNOD. **Facilite o trabalho em grupo, tenha seu software colaborativo (Groupware).** Disponível em: <http://www.mknod.com.br/?q=node/155>. Acesso em 23 de abril de 2011.

NUNES, **Conceito de Coordenação.** 2007. Disponível em: <http://www.knoow.net/cienceconempr/gestao/coordenacao.htm>. Acesso em 10 de maio de 2011.

ORLIKOWSKI, Wanda J. **Learning from Notes: Organizational Issues in Groupware Implementation.** Sloan School of Management. Cambridge, MA.

Schrage, M. **Cultures of Prototyping,** In: Bringing Design to Software, Edited by Terry Winograd, ACM Press, USA, 1996. ISBN 0-201-85491-0.

SIMON, Imre. **Cooperação.** Disponível em <http://www.ime.usp.br/~is/ddt/mac333/aulas/tema-1-04mar99.html>. Acesso em 10 de maio de 2011.

SOS Estudante. **Groupware – Software Colaborativo.** Disponível em: <http://www.sosestudante.com/informatica/groupware-software-colaborativo.html> Acesso em 28 de abril de 2011.

WEBNUCLEO. **Groupware.** Disponível em: <http://www.webnucleo.com/solucoes/groupware>. Acesso em 23 de Abril de 2011.