

**CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

VINICIUS LIMA BETONI

**SEGAN – SERVIÇO DE GEOPROCESSAMENTO DE APOIO AO
NEGÓCIO**

**MARÍLIA
2012**

**CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

VINICIUS LIMA BETONI

Trabalho de Conclusão de Curso apresentado ao Centro Universitário Eurípides de Marília como parte dos requisitos necessários para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof^a. Elvis Fusco

**MARÍLIA
2012**

BETONI, Vinicius Lima

SEGAN – SERVIÇO DE GEOPROCESSAMENTO DE APOIO AO NEGÓCIO / Vinicius Lima Betoni; orientador: Prof^ª. MSc. Elvis Fusco. Marília, SP: [s.n.], 2012.

58f.

Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Centro Universitário Eurípides de Marília.

1.Ferramentas de Desenvolvimento 2.Interoperabilidade 3.SIG
4.SOA

CDD: 005.2



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM SISTEMA DE INFORMAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

VINICIUS LIMA BETONI

**SEGAN – SERVIÇO DE GEOPROCESSAMENTO DE APOIO AO
NEGÓCIO**

Banca examinadora do trabalho de conclusão de curso apresentado ao Centro Universitário Eurípides de Marília como parte dos requisitos necessários para a obtenção do grau de Bacharel em Sistemas de Informação.

Resultado: _____

ORIENTADOR: Prof^a. Elvis Fusco

1º EXAMINADOR: _____

2º EXAMINADOR: _____

Marília, ____ de _____ de 2012.

A minha família, que sempre com muito amor e incentivos, me deram força e estímulo para estudar com objetivo de crescer profissionalmente.

AGRADECIMENTOS

Agradeço a todos que nunca pouparam esforços para me apoiar nesta fase, sempre me auxiliando na conclusão desse projeto.

“Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível.”

Charles Chaplin

BETONI, Vinicius Lima. **SEGAN – Serviço de Geoprocessamento de Apoio ao Negócio**. 2012. 58f. Trabalho de Conclusão de Curso de Bacharelado em Sistemas de Informação – Centro Universitário Eurípides de Marília, Fundação de Ensino “Eurípides Soares da Rocha”, Marília, 2012.

RESUMO

Este projeto visa colaborar com novos empreendedores na tomada de decisão, utilizando-se de conceituadas tecnologias e métodos de desenvolvimentos. Ele foi desenvolvido para disponibilizar informações de empresas de mapa, podendo auxiliar novas empresas na escolha de seu local de instalação, bem como contribuir na identificação de possíveis e próximos concorrentes, com o objetivo de transformar esses novos empreendedores mais concorrentes, reduzindo as chances de fracasso. Para exibição das empresas nos mapas, foi utilizado *Sistema de Informação Geográfica (SIG)*, resultado da integração entre hardware, software e dados para capturar, analisar e disponibilizar todas as formas de informação geograficamente referenciadas, que permite visualizar as devidas informações das empresas nos seus respectivos pontos.

Palavras-chave: Ferramentas de Desenvolvimento; Interoperabilidade; SIG; SOA.

BETONI, Vinicius Lima. **SEGAN – Serviço de Geoprocessamento de Apoio ao Negócio**. 2012. 58f. Trabalho de Conclusão de Curso de Bacharelado em Sistemas de Informação – Centro Universitário Eurípides de Marília, Fundação de Ensino “Eurípides Soares da Rocha”, Marília, 2012.

ABSTRACT

This project aims to work with new entrepreneurs in decision making, using technologies and methods of prestigious developments. It was developed to provide map information for companies and can assist new businesses in choosing their place of installation as well as help identify potential competitors and next, with the goal of transforming these new entrepreneurs more competitive, reducing the chances of failure. To view the companies in the maps, was used Geographic Information System (GIS), the result of integration between hardware, software and data for capturing, analyzing and providing all forms of geographically referenced information, which allows to view the appropriate information of the companies in their respective spots.

Keywords: Development Tools; Interoperability; GIS; SOA.

LISTA DE ILUSTRAÇÕES

Figura 1 - Correspondência entre padrões de metadados (CROMWELL-KESSLER, 1998).....	20
Figura 2 - Esquema.....	21
Figura 3 - Modelo Lógico do Banco de Dados	34
Figura 4 - Plataforma Microsoft .NET.....	35
Figura 5 - Exemplo de Web services utilizado no projeto SEGAN, utilizado para carregar os estados durante a Alteração dos dados cadastrais da empresa.....	39
Figura 6 - Página disponibilizada pelo Framework ASP.NET onde é possível realizar os devidos testes com os métodos desenvolvidos.....	40
Figura 7 - Parâmetros solicitados pelos métodos do Web Service.	41
Figura 8 - XML de retorno do Web Service	42
Figura 9 - Abertura do sistema, tela principal.....	44
Figura 10- Busca Específica.....	45
Figura 11 - Retorno do Método utilizando Serviço de Geocodificação.....	46
Figura 12 - Exibição da empresa no mapa junto aos dados cadastrados.....	47
Figura 13 - Solicitação de acesso à área restrita	48
Figura 14 - Dados Cadastrais Empresas.	49
Figura 15 - Tela inicial SEGAN ADM.....	50
Figura 16 - Tela de Solicitações, permitindo consulta e alteração.....	51
Figura 17 - Cadastro de Empresas e exemplo de AJAX CascadingDropDown	52
Figura 18 - Gravação da senha criptografada no banco de dados.....	53

LISTA DE ABREVIATURAS E SIGLAS

AJAX	Asynchronous Javascript and XML
API	Interface de Programação de Aplicativos
ASP	Active Server Pages
BD	Banco de Dados
CGIS	Canadá Geographic Information System
CLR	Common Language Runtime
COM	Component Object Model
CRUD	Create, Read, Update e Delete
CSS	Cascading Style Sheets
C#	Linguagem de Programação C Sharp
C++	Linguagem de Programação C++
DC	Dublin Core
DER	Diagrama Entidade Relacionamento
DoDAF	Department of Defense Architecture Framework
DOM	Document Object Model
DP	Data Provides
DSPACE	Repositories institutional
ER	Entidade Relacionamento
GEM	Global Entrepreneurship Monitor 2011
HTML	Hypertext Markup Language
HTTP	Protocolo de Transferência de Hipertexto
IAAS	Infraestrutura como serviço
IBQP	Instituto Brasileiro da Qualidade e Produtividade

IDE	Integrated Development Environment
ISSO	International Standard Organization
MARC	Machine-Readable Cataloging
MFC	Meta Content Framework
NATO	Organização do Tratado do Atlântico Norte
NISO	National Information Standard Organization
OAI	Open Archives Initiative
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvestin
OCS	Open Conference Systems
OJS	Open Journal Systems
OO	Orientação a objetos
PKP	Public Knowledge Project
RAD	Desenvolvimento Rápido de Aplicação
RDF	Resource Description Framework
SEBRAE	Serviço Brasileiro de Apoio às Micro e Pequenas Empresas
SEGAN	Serviço de Geoprocessamento de Apoio ao Negócio
SDLIP	Protocolo Simple Digital Library Interoperability Protocol
SGBD	Sistemas de Gerência de Banco de Dados
SIG	Sistema de Informação Geográfica
SMTP	Simple Mail Transfer Protocol
SOA	Arquitetura Orientada a Serviços
SOAP	Protocolo Simples de Acesso a Objetos
SODA	Service Oriented Architecture Device
SQL	Linguagem de Consulta Estruturada
UDDI	Universal Description, Discovery and Integration
URL	Uniform Resource Locator

UTM	Universal Transverse Mercator
VB	Linguagem de Programação Visual Basic
WSDL	Web Services Description Language
WWW	World Wide Web
XLST	Linguagem Extensível para folhas de Estilo de Transformações
XML	Extensible Markup Language

SUMÁRIO

INTRODUÇÃO	15
1.INTEROPERABILIDADE.....	16
1.1 Níveis de Interoperabilidade	17
1.2 Abordagens de Interoperabilidade	18
1.3 O Uso de Metadados na Interoperabilidade.....	19
2.SERVICE-ORIENTED ARCHITECTURE (SOA)	21
2.1 Modelando Informação como Serviços	23
2.2 O uso de padrões de projeto existentes em SOA	24
2.3 Análise de Design Arquitetura SOA	24
3 WEB SERVICES	25
4 SISTEMA DE INFORMAÇÃO GEOGRÁFICA (SIG).....	26
4.1 Georreferenciamento	28
5. Serviço de Geoprocessamento de apoio ao Negócio (SEGAN).....	29
5.1 Arquitetura	29
5.1.1 Orientação a Objeto.....	31
5.1.2 Arquitetura em três camadas.....	32
5.2 Banco de Dados.....	33
5.3 Implementação.....	35
6. Segan Web.....	43
6.1Segan ADM.....	49
CONCLUSÃO.....	54
REFERÊNCIAS.....	55

INTRODUÇÃO

O Brasil tem se tornado um país de empreendedores, de acordo com os dados divulgados pelo *Global Entrepreneurship Monitor 2011* (GEM) realizado em conjunto com o *Serviço Brasileiro de Apoio às Micro e Pequenas Empresas* (SEBRAE) e *Instituto Brasileiro da Qualidade e Produtividade* (IBQP), cerca de 27 milhões de pessoas estão envolvidas em seu próprio negócio ou na criação de um novo negócio, transformando assim o Brasil no terceiro país entre 54 países no número de empreendedores, ficando atrás apenas da China e dos Estados Unidos¹. Muitos desses novos negócios não possuem estudo ou análise sobre alguns pontos que deveriam ser considerados essenciais, sendo eles, a escolha do local adequado para instalação, a análise sobre os possíveis e próximos concorrentes e a devida divulgação sobre o novo negócio. Baseando-se nestes pontos, surgiu o Serviço de Geoprocessamento de Apoio ao Negócio (SEGAN), com o intuito de auxiliar as empresas na tomada de decisão sobre os pontos mencionados anteriormente, disponibilizando a fácil visualização e a disposição das empresas já ativas no mapa geográfico.

Valendo-se de tecnologias e métodos conceituados, como ferramentas de desenvolvimentos, *Integrated Development Environment* (IDE) e banco de dados relacional, no qual proporcionaram o desenvolvimento de ambas as aplicações e armazenamento das informações. Utilizando-se de diversas fontes de dados, alimentado de diferentes sistemas fez-se necessário um estudo sobre interoperabilidade, no qual permite a troca e/ou reutilização de informação sem a necessidade de adaptação mantendo assim sua integridade.

Atualmente, a mobilidade tem se apresentado como uma das grandes inovações tecnológicas, baseando-se na infinita gama de informações, serviços e recursos disponíveis eletronicamente. Considera-se que são poucos os mercados e pessoas que deixariam de utilizar essa nova tecnologia, o grande crescimento dessa nova tecnologia fez com que o SEGAN fosse arquitetado de forma que pudesse atender a esse mercado, dessa maneira foi implementado baseando-se em tecnologias orientadas a serviços (SOA), no qual consiste um conjunto de princípios e metodologias, para a concepção e desenvolvimento de software voltado para interoperabilidade de serviços. Estes serviços são construídos como componentes de software que permitem a reutilização para fins diferentes, deste modo foi implementado *Web Service*, no qual a reutilização dos dados utilizados para a exibição das empresas em

¹ Disponível em: <http://www.pr.agenciasebrae.com.br/noticia.kmf?canal=722&cod=13712121>.
<http://www.infomoney.com.br/negocios/noticia/2485034/brasil-terceiro-pais-com-maior-numero-mpreendedores>

outros sistemas ou formas de exibição são possíveis.

1. INTEROPERABILIDADE

Pode-se definir interoperabilidade como o requisito principal para integração de ambientes colaborativos informacionais. Segundo a International Standard Organization (ISO) interoperabilidade é a habilidade de dois ou mais sistemas interagirem e trocarem dados a partir de um processo definido, que tem por objetivo alcançar resultados esperados.

Em simples definição, interoperabilidade é a habilidade de realizar tarefas em conjunto.

A literatura sobre engenharia de software considera a interoperabilidade como o esforço necessário para se vincular um sistema a outro, um fator de garantia de qualidade de software, conjuntamente com manutenibilidade, portabilidade, integridade, confiabilidade, etc. (PRESSMAN, 1995).

Para que exista o compartilhamento e/ou troca de dados entre organizações, é necessário que haja por ambas as partes o entendimento necessário. Portanto, é necessária a implementação de padrões e normas que possibilitem o entrosamento de ambas as partes “para que haja interoperabilidade semântica, é necessário que haja acordos quanto à utilização destes descritores, isto é, que um termo tenha o mesmo significado que o utilizado em outra base” (OLIVEIRA, 2005).

O *Concise Oxford Dictionary* define interoperabilidade como a habilidade que um sistema tem em trabalhar com outros sistemas sem precisar de qualquer empenho específico por parte do cliente. Ainda sobre esta definição, Arms (2002) afirma que o escopo da interoperabilidade é implementar serviços ligados aos usuários, utilizando-se de meios informacionais onde são tecnicamente diferentes e gerenciados diferentemente por diversas organizações. Isto requer acordos de cooperação em três níveis:

- Nível Técnico, oferece a interoperabilidade tecnológica;
- Nível Conteúdo, refere à interoperabilidade semântica, na qual a organização das informações são consideradas peças-chaves a serem estudadas;
- Nível Organizacional, remete à interoperabilidade política, quando organizações se empenham com o intuito de atingir a interoperabilidade, desenvolvendo padrões e tecnologias que colaborem com a meta.

1.1 NÍVEIS DE INTEROPERABILIDADE

Partindo do princípio da interoperabilidade de informações é indispensável o entendimento de três níveis característicos na interação das informações: semântico, estrutural e sintático.

a) **Nível Semântico:** Possibilita entender o sentido de cada elemento descritor do recurso e suas associações. A utilização de vocabulários específicos, ontologias e/ou padrões de metadados são fundamentais para garantir esse tipo de interoperabilidade (MOURA, 2002).

De acordo com Marino (2001), existem dois sub níveis no nível semântico:

- **Epistemológico:** trata-se do significado dos elementos descritores do formato e das relações nele existentes;
- **Ontológico:** trata-se do uso de ontologias, vocabulários controlados e padrões de metadados para firmar os significados dos dados representados.

b) **Nível Estrutural:** Determina cada composição de um padrão de metadados, expõem os seus tipos, escala os possíveis valores para os determinados componentes e o mecanismo utilizado para agrupar e/ou relacionar esses elementos de modo que o processo possa ser realizado automaticamente. Segundo Barreto (1999), quanto mais complexa for a estrutura do padrão de metadados, mais complexo deve ser o modelo de dados empregado para descrevê-los, como exemplos nesse nível, observa-se padrões de metadados *Dublin Core (DC)* e *MARXML*.

c) **Nível Sintático:** Segundo Barreto (1999), a sintaxe disponibiliza uma linguagem comum para representar a estrutura dos metadados. Para Moura (2002), esse nível estabelece como os metadados devem ser codificados para a transferência de informações. Tem-se como exemplo: A linguagem XML (*eXtensible Markup Language*), no comando da troca de informações.

1.2 ABORDAGENS DE INTEROPERABILIDADE

Sobre integração de base de dados heterogêneos, inúmeros são os estudos a fim de identificar possíveis problemas existentes e também possíveis soluções. Diversas intervenções devem ser feitas para buscar atingir uma integração que permita a interoperabilidade ideal, disponibilizando o acesso as instituições, de forma rápida e simples onde toda a complexidade por trás da interoperabilidade se torne totalmente transparente ao usuário. Almeida (2002) cita algumas abordagens adotadas:

- Protocolos, modelos e visualizações: são níveis a serem considerados ao projetar uma solução que objetiva interoperabilidade. Protocolos compõem a base da infraestrutura para ambientes informacionais de rede. Em modelos conceituais, a pesquisa engloba áreas como recuperação da informação, bibliotecas, banco de dados e trabalhos cooperativos em computadores. Já as técnicas de visualização são imprescindíveis para apresentar aos usuários os componentes do ambiente;

- Nível sintático versus nível semântico: um dos problemas de interoperabilidade, que permeiam os sistemas informatizados das instituições, está relacionado ao grau de heterogeneidade das fontes e dos componentes em ambos os níveis. Prover interoperabilidade semântica, a qual proporciona maior possibilidade para a interpretação de uso dos dados, se torna uma tarefa mais complexa;

- Middleware: serve como elemento que aglutina e dá coerência a um conjunto de aplicações e ambientes;

- Ontologias: em relação à classificação dos tipos de dados para a extração, baseiam-se na concepção de uma hierarquia fundamentada em uma ontologia para um determinado domínio. As ontologias são implantadas para prover elementos semânticos passíveis de leitura para computadores, esses elementos semânticos possibilitam a comunicação entre diversos agentes (humanos ou softwares);

- Interfaces: está fundamentada em uma interface que possibilita a intermediação entre inúmeros tipos de recursos de trabalho em grupo utilizando a Internet como meio de comunicação;

- Metadados: tem como objetivo solucionar problemas de interoperabilidade em ambientes heterogêneos utilizando-se de uma arquitetura de metadados que integram informações sobre as diversas coleções e serviços disponíveis;

- Linguagens de Consulta: abordam problemas de busca em ambientes heterogêneos baseando-se em dificuldades em diferentes estilos e linguagens de consultas booleanas. Propõe uma arquitetura e um mecanismo para tradução de consultas em uma linguagem comum.

Essas citações mostram a diversidade e complexidade do assunto entre outros estudos e inúmeras possibilidades devem ser pesquisadas na implementação do intercâmbio de sistemas de informações.

1.30 USO DE METADADOS NA INTEROPERABILIDADE

A interoperabilidade é a capacidade de compartilhamento de informações em diversos ambientes informacionais e que, por meio de tecnologias como linguagens de marcação associada ao uso de padrões de metadados, permite que informações persistidas e em plataforma heterogêneas possam ser compartilhadas possibilitando assim a contribuindo com a integração entre diferentes sistemas de informações.

Os metadados cumprem um papel importante na descoberta de recursos, ou seja, possibilitam a busca de elementos de informação por critérios relevantes, identificação de recursos, junção de recursos similares, diferenciação de recursos não similares e a obtenção da informação de localização. A definição de recursos por meio de metadados permite que sejam semanticamente compreendidos por diversos sistemas de informações possibilitando assim a interoperabilidade entre aplicações e o compartilhamento de informações entre sistemas.

Os diferentes significados que uma mesma informação pode ter, também conhecido como heterogeneidade semântica, é causadora de um obstáculo entre diversas fontes de dados gerando conflitos como sinonímia, ou seja, dados com o mesmo conteúdo semântico, mas com nomes diferentes, e homonímia, isto é, dados com o mesmo nome, mas com conceitos diferentes. Os metadados são mecanismos que auxiliam a resolução deste problema da interoperabilidade, os dados podem estar descritos por um único padrão de metadados, para tanto, há a necessidade de correspondência entre padrões de metadados, mesmo dentro de uma mesma área.

A Figura 1 exibe parte da tabela de “Correspondências entre padrões de metadados” (CROMWELL-KESSLER, 1998), explicado a construção de relações entre diferentes padrões de metadados e os requisitos para interoperar sistemas baseados nesses tipos de bases

heterogêneos.

CDWA	Object ID	FDA	VRA Core	REACH	USMARC	DC
Titles or names	Title	Group/Item Identification-Repository Title Group/Item Identification-Descriptive Title Group/Item Identification-Inscribed Title	W2. Title	Field #4: Object Name/Title	24Xa Title and Title-Related Information	Title
State					562c Copy and Version Identification Note-Version Identification	Description
Classification						Subject
Copyright/Restrictions	Copyright restriction	Internal Documentation Restriction			540a Terms Governing Use and Reproduction	Rights

Figura 1 - Correspondência entre padrões de metadados (CROMWELL-KESSLER, 1998).

É possível verificar nessa correspondência que nem todos os elementos têm correspondências nos diversos padrões. Segundo Moura (2002), as arquiteturas de metadados foram desenvolvidas para garantir a interoperabilidade entre diversos padrões de metadados, com o propósito de representar e dar suporte ao transporte de uma variedade de esquemas de metadados em ambientes distribuídos, promovendo interoperabilidade nos níveis sintático, estrutural e semântico.

Destaca-se entre os tipos de arquitetura de metadados na Web a arquitetura Warwick, MFC – Meta Content Framework, a RDF – Resource Description Framework e a arquitetura de modelagem de quatro níveis.

Baseando-se em Alves (2005), a RDF vem sendo indicada e recomendada pelo W3C para interoperabilidade na rede, sendo que, unida aos metadados, ao uso da linguagem XML e às ontologias, possibilita de modo flexível, a interoperabilidade nos três níveis.

A Figura 2 esboça o sentido de um esquema para interoperar diversos sistemas utilizando padrões de metadados distintos, onde é definido uma arquitetura de metadados padrão, em que deve haver uma correspondência entre os diversos metadados que participam

da integração das bases de dados.

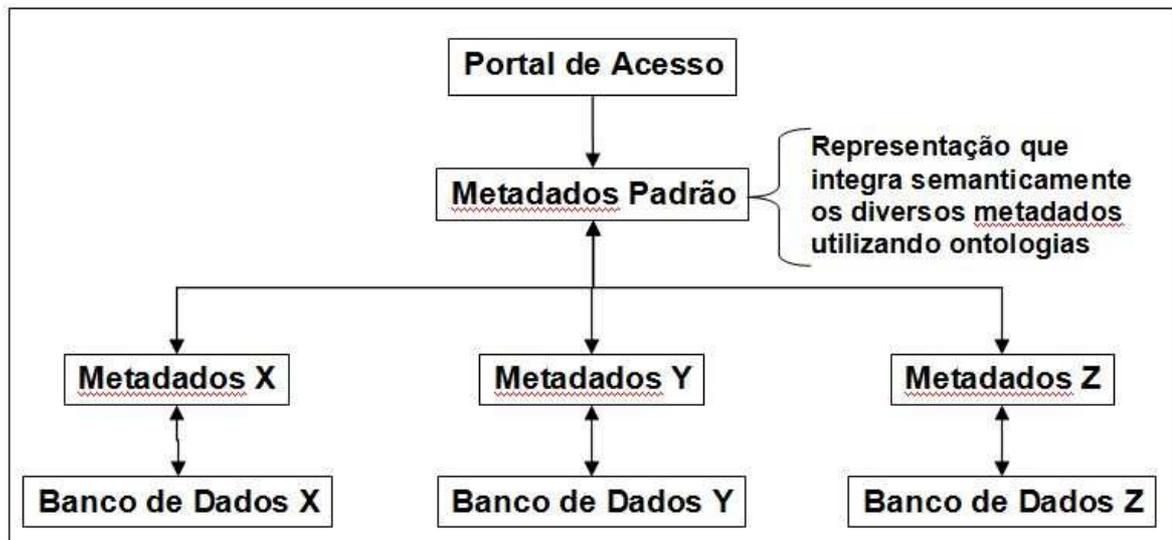


Figura 2 – Esquema

2. SERVICE-ORIENTED ARCHITECTURE (SOA) – ARQUITETURA ORIENTADA A SERVIÇO

A Arquitetura Orientada a Serviços (SOA) pode ser considerada como um estilo arquitetônico que se destacou desde o estouro da Web 2.0, XML e Web Services. Como o próprio nome indica, um completo modelo de arquitetura capaz de fornecer suporte a um sistema por inteiro, Sanders, et AL (2008) nos relembra que “SOA é mais adequado para ser unido com uma estrutura de arquitetura como o DoDAF para descrever um sistema completo” (DoDAF referindo-se ao *Department of Defense Architecture Framework*, no qual aborda uma visão múltipla para a descrição de um sistema que engloba lógica, estrutura e descritivo para um sistema por completo). Tal como, podemos imaginar SOA como um modelo de alto nível, que pode deixar detalhes do projeto considerados um nível mais abaixo como i.e, UML *Static Structure*, Diagramas de Sequência, estados de transição entre outros isolados.

Na verdade, o melhor entendimento que se pode ter para SOA é a noção de estilos arquitetônicos, como citados por Zhu (2005). Como um estilo arquitetônico, o termo para SOA não se refere a uma arquitetura concreta que descreve um sistema. Em vez disso, é um conjunto de abstrato de diretrizes que rege um estilo de sistema que pode resultar em

ilimitadas e diferentes arquiteturas SOA. O conceito de estilos de “componentes genéricos universal” (Zhu, 2005)

SOA também apresenta componentes que se comunicam através de mensagens, onde cada um desses componentes desempenha uma determinada atividade. No entanto, essas atividades geralmente são associadas aos processos de negócio. Mensagens e componentes de interface, tipicamente utilizam-se de uma linguagem de definição comum.

Todos os componentes utilizados em SOA são chamados de serviços e encapsulam um determinado processo de negócios ou alguma funcionabilidade reutilizável. Serviços envolvendo SOA habitualmente possuem definidas API's, geralmente separados em arquivos no qual auxiliam no suporte, que executam atividades e/ou métodos específicos para outros componentes ou serviços. São partes de um sistema distribuído no qual utiliza-se desses fatores para tirar vantagens em relação a alta portabilidade, aproveitando-se da natureza de plataformas independentes.

Baseando-se em propriedades derivadas da Arquitetura, SOA suporta inúmeras propriedades úteis, como parte de sua abordagem para o projeto do sistema em si. Como mencionado anteriormente, a portabilidade está presente devido à natureza da plataforma independente do formato da mensagem e/ou protocolos utilizados e também a reutilização está muito presente na SOA. Além disso, a flexibilidade surge como um serviço que muitas vezes é imperceptível aos usuários: "Qualquer componente que demonstra a funcionalidade reutilizáveis em diferentes aplicações pode ser um candidato para um serviço" (Sanders, et al 2008).

Geralmente as vantagens mais comuns mencionadas para SOA são: a manutenção, a escalabilidade e a modificabilidade devido à natureza do sistema de distribuição, o acoplamento flexível e a natureza dos próprios serviços. Com isso a real funcionalidade fica invisível para o usuário, onde o mesmo percebe a funcionalidade, porém não visualiza os detalhes da implementação.

2.1 MODELANDO INFORMAÇÃO COMO SERVIÇO

Information as a Service (IAAS), baseia-se nas informações de negócio e tende a envolvê-las como um serviço, possibilitando aproveitar os benefícios de SOA. De fato, a IAAS poderia ser qualificada como um subtipo de SOA, caso não houvesse uma divergência significativa em sua arquitetura. Nesse sentido, o correto desenvolvimento de métodos para captura de informação como serviço pode ser considerado um ótimo modelo de SOA.

Segundo Dan, et al (2007), existem três principais áreas que a metodologia atual de desenvolvimento precisa ser modificada a fim de apoiar IAAS: modelagem e identificação, realização e nível empresarial como serviço.

Modelagem e identificação associam-se com a noção em que os serviços não podem ser mapeados um para um em processos específicos de negócio. Assim, a fim de obter uma representação correta em relação aos serviços, devem em primeiro lugar ser definidos os processos do negócio, desta forma torna-se possível decompor os processos em partes menores, proporcionando maior reutilização dos serviços.

De acordo com Dan, et al (2007), realização menciona os reais acessos que os serviços se utilizarão para recuperar as informações. Utilizar esse procedimento sem definições e um estudo sobre o impacto que podem ser causado, podem comprometer seriamente um projeto. Para evitar esse tipo de situação, deve se levar em consideração os sistemas existentes, os requisitos de idioma, e tempo de acesso/possíveis erros que podem ocorrer entre múltiplos serviços na mesma base de dados, por esta razão, Diagramas de Domínio devem ser utilizados, além de se atentar para o tempo entre os Diagramas de Sequência a fim de garantir o comportamento adequado do sistema em geral.

Última área mencionada por Dan, et al (2007) está no nível empresarial como serviços, tipicamente os serviços são definidos para atender as necessidades do negócio e, geralmente, destinados à prestação de serviço tanto dentro de uma organização, intranet, quanto a *World Wide Web* (WWW). Com isso, podem surgir inúmeras divergências em informações na qual podem comprometer seriamente o projeto.

2.2 O USO DE PADRÕES DE PROJETOS EXISTENTES EM SOA

Existem inúmeros padrões de projetos para arquitetura SOA, onde esses padrões são livres e disponibilizados na internet. Thomas Erl disponibiliza uma imensa área sobre esses padrões que podem ser acessados na seguinte URL: <http://www.soapatterns.org/>.

Mauro & Krcmar (2010), utilizaram estes padrões de projetos para identificar possíveis soluções existentes, identificando problemas dentro de seu próprio trabalho na *Service Oriented Architecture Device* (SODA). Utilizaram, ainda, uma lista de nove problemas conhecidos com a sua solução de alto nível arquitetônico e, em seguida, foram capazes de cruzar contra 108 diferentes padrões de design. Eventualmente foram capazes de encontrar diversos padrões que solucionaram os problemas conhecidos, com isso surgiu mais direcionamento nas áreas para pesquisa a fim de solucionar as inconsistências remanescentes em sua arquitetura.

2.3 ANÁLISE DE DESIGN ARQUITETURA SOA

Zhou & Zhang (2009) apresentam duas formas para realmente analisar design Arquitetura SOA: arquitetura e qualidade. Em termos de arquitetura, basicamente consiste na atribuição de pesos aos componentes em diferentes camadas do modelo de serviço, adicionando o peso até obter uma sensação geral para a importância de um determinado nível para o design total. Por outro lado, a análise de qualidade envolve mais intuição e é matematicamente menos formulada.

A análise de qualidade apresentada por Zhou & Zhang (2009) visa identificar dois indicadores de qualidade de design: grau de reutilização e grau de acoplamento. Estes indicadores foram escolhidos para auxiliar no apoio as vantagens globais de um SOA, em termos de agilidade nos negócios, reutilização, modificação, flexibilidade, e integração.

O grau de capacidade de reutilização por ser determinado baseando-se em uma soma ponderada numérica de propriedades atribuídas aos componentes de serviços da camada que oferecem em termos de reutilização ou reaproveitamento e/ou no número de maneiras diferentes que os mesmos são utilizados. Como auxílio na avaliação, vale lembrar que esses serviços em termos de publicação ou relacionamento produtor/consumidor, respondem quantos fluxos consomem os serviços (internos) e quantos serviços são consumidos.

O grau de acoplamento pode ser determinado observando a média numérica dos

componentes que dependam diretamente dos serviços dentro de um projeto SOA, o processo é semelhante ao determinar a reutilização externa de um serviço, também é necessário considerar tarefas definidas por Zhou & Zhang (2009) como “chamada de uma ação de serviços ou outro tipo de ação dentro do processo de negócio”. Basicamente é necessário descobrir como muitos outros componentes dentro do sistema seriam afetados por alterações em um determinado serviço do sistema. Com isso deve ser criado um valor médio de todos os serviços dentro do design para que seja calculado a média.

3. WEB SERVICES

Web Services são um conjunto de aplicações autodescritivas que podem ser publicadas, localizadas e invocadas por meio da web. Estas aplicações podem ser desde processos simplificados, como por exemplo, troca de mensagens, até transações comerciais ou industriais como um processo de compra de mercadorias. Uma vez que um *Web Service* é publicado, inúmeras aplicações, até mesmos outros *Web Services*, podem consumi-los, tanto para obtenção de dados, como para interação com serviços que uma organização oferece (TAMAE, 2004).

Um *Web Service* é acessado por meio de protocolos e formato de dados independentes de plataformas, ou seja, de uma maneira mais simples, é possível ser consumido por várias aplicações, como por exemplo uma aplicação voltada para plataforma *Windows* e outra para plataforma *Linux*, independentes da linguagem de programação utilizada para desenvolvimento da ferramenta (caso a mesma disponibilize suporte a esse serviço.) como o http, XML e SOAP. A interface de um *Web Service* é acessível através de mensagens XML padronizadas. São descritos baseando-se em um padrão formal chamado, descrição de serviço que envolve os detalhes necessários para a interação com o serviço, incluindo o formato das mensagens, tipos de dados e localização. A descoberta e invocação dinâmica de serviços e uma colaboração baseada em mensagens permite o desenvolvimento de aplicações distribuídas com enorme grau de interoperabilidade.

O principal atrativo de um *Web Service* está em suas características de implementação: os protocolos que compõem a arquitetura baseiam-se principalmente em padrões aceitos e amplamente adotados pelo mercado, como o *Hiper Text Transfer Protocol* (HTTP) e a linguagem padrão de representação de dados XML. Resumindo, um *Web Service* toma partido

de todos os recursos já existentes e muito utilizados que compõem a Internet e *World Wide Web*.

Partindo do princípio que os *Web Services* se utilizam do protocolo HTTP para compartilhamento de mensagens, possibilitam a interação entre diversos outros *Web Services* e o uso de XML para a descrição das informações também é uma das vantagens da arquitetura, pois a linguagem vem se tornando padrão universal, além de tornar a representação de dados independentes de linguagem ou de plataformas, de tal forma que um tipo de dado descrito em XML pode ser convertido para representações específicas dentro de cada plataforma ou linguagem por se basear em formato de texto. Todo o foco em procurar estabelecer um padrão aberto e amplamente aceito, tem como principal objetivo garantir a interoperabilidade e a manutenção de código legado, evitando elevados custos de um processo de reengenharia tecnológica.

Interoperabilidade surge como um dos principais pontos na tecnologia dos *Web Services*, pois parte do princípio de aproveitar todo o código legado, independente da linguagem na qual ele foi escrito e transformá-lo em uma funcionalidade pronta para qualquer outro tipo de linguagem. A implementação dos protocolos da arquitetura dos *Web Services* nas diversas plataformas de programação se encarrega de promover essa conversão transparente dos dados.

4. SISTEMA DE INFORMAÇÃO GEOGRÁFICA (SIG)

Um sistema de informação Geográfica (SIG) é projetado para capturar, armazenar, visualizar, comunicar, transformar, analisar e arquivar informações georreferenciada, isto é, a informação ligada a locais específicos na superfície da Terra. SIG's são capazes em certas medidas de substituir o tradicional papel desempenhado pelos mapas, além de tratar informação sob a forma de imagens de satélites da superfície da Terra, bem como informações de pesquisas e registros administrativos que foram georreferenciado. Estão se tornando cada vez mais utilizados nas ciências sociais de apoio a investigação com base em cortes transversais de dados ou estudos para qual localização geográfica do contexto são importantes e úteis.

A origem dos SIG's pode ser atribuída a meados dos anos 1960, os primeiros computadores foram projetados principalmente para processamento numérico, seguindo o

exemplo de Babbage e outros, mas por volta de 1965 outras aplicações começaram a surgir, apoiado em partes pelo desenvolvimento de especializados periféricos, nomeadamente *pen plotters e map digitizers*.² Com isso passou a ser gerado enorme problema com processamento de mapas, como remover inúmeros mapas criados para documentar recursos e terras subutilizadas e produzir tabelas das quantidades de terras disponíveis para outros tipos de desenvolvimento e utilização. Medição de áreas a partir de mapas sempre foram processos demorados, tediosos e imprecisos, porém se o mapa pode ser convertido para uma forma digital, um simples algoritmo permitiria que áreas fossem medidas e tabuladas eletronicamente. O *Canadá Geographic Information System (CGIS)* foi uma resposta para essa situação.

Na década de 1980, SIG's comerciais começaram a surgir, oferecendo uma ampla gama de funções na qual inúmeras maneiras eram muito complexas, tediosas e imprecisas e com um alto valor financeiro para ser realizadas manualmente, isto inclui simples medição de áreas e comprimento, transformações necessárias para alterar os formatos de dados, simples análises estatísticas, tais como o cálculo de médias e desvio padrão e uma série de mais métodos complexos geralmente denominados como análise espacial. Além disso, foram os SIG's os responsáveis pelo fornecimento de recursos avançados de visualização de dados, incluindo mapeamento e várias formas de dados visualizados. A comunidade científica logo percebeu o potencial do SIG, e por volta de 1980 e 1990 ele surgiu como uma ferramenta indispensável para a pesquisa em qualquer superfície da Terra. Nas ciências sociais, algumas das primeiras aplicações foram voltadas para área de arqueologia, política cientista, criminologistas, demógrafos e epidemiologista.

SIG passou por uma transformação significativa, no que se refere às aplicações começaram a surgir muito além da noção inicial de um assistente digital. O advento da *World Wide Web (WWW)* em 1995 induziu uma acentuada mudança de perspectiva, onde SIG's eram vistos como um meio para o compartilhamento de informações entre as pessoas, além do seu papel mais tradicional. Muitos sites foram criados, oferecendo-se para fornecer aos visitantes um conjunto de dados geográficos, ou para executar simplificados serviços SIG utilizando informação dos usuários ou do próprio site.

Avanços da tecnologia trouxeram a promessa SIG de que não está mais confinado ao escritório, mas levada ao campo no formulário portátil e dispositivos portáteis. A

² Disponível em: <http://www.grossmont.net/judd.curran/thx2article.pdf>

comunicação sem fio transforma disponível o *download* e *upload* de dados para serviços da internet onde possuem poder suficiente sobre os dispositivos portáteis para apoiar qualquer forma de operação SIG. O advento do SIG fornece ao campo revolucionar a natureza e a prática em trabalhar pesquisas sociais e outros campos sociais baseados em ciência.

O coração de um SIG é um sistema de representação, pelo que as características do mundo real são codificados no alfabeto binário do computador digital. Representações SIG incluem tipicamente três aspectos das características do mundo real: localização da superfície da Terra, utilização de um sistema de coordenadas convenientes tal como latitude e longitude, seus atributos, ou o que são conhecidos sobre eles, e todas as relações de importância entre eles. Exemplos de relações incluem adjacência, tais como o relacionamento que pode existir entre dois bairros, e a conectividade entre partes de uma rede de ruas.

4.1 GEORREFERENCIAMENTO

Um sistema para identificar com precisão a localização da superfície da Terra é um componente essencial de qualquer representação SIG. A conferência de Meridian 1884 definiu latitude e longitude como padrões universais para georreferenciamento, com base em medições a partir do meridiano de Greenwich e do Equador. Infelizmente a Terra não é uma esfera perfeita e tem sido aproximada por uma variedade de funções matemáticas ao longo do tempo e em diferentes partes do mundo, cada um dos quais potencialmente leva a uma latitude ligeiramente diferente da longitude. O sistema ou dados de escolha da América do Norte é o *North American Datums* de 1983, mais outros dados podem ser encontrados, tais como o NAD27 utilizados por outros países. Tudo isso significa que é impossível determinar a localização exata e as variações de como tanto quanto 200m no solo podem existir entre determinações de latitude e longitude utilizando dados diferentes. Softwares SIG modernos tornam possível converter facilmente a partir de um ponto de referência para cientistas, no entanto, ocasionalmente são encontrados diferenças de referenciais.

Além de latitude e longitude, georreferenciamento muitas vezes faz uso de métodos para projetar a superfície curva da Terra sobre um plano associado aos sistemas de coordenadas planas. Estes incluem o sistema *Universal Transverse Mercator* (UTM), o NATO amplamente utilizado nas agências nacionais. UTM consiste de 60 projeções distintas e sistemas de coordenadas, cada um projetado para fornecer uma precisão dentro de uma zona

de longitude.

5. SERVIÇO DE GEOPROCESSAMENTO DE APOIO AO NEGÓCIO (SEGAN)

Devido ao crescente surgimento de novas empresas e o fato de o Brasil ser um país de empreendedores, onde todos os dias diversas empresas surgem no mercado em todos os ramos, porém, muitas dessas novas empresas sem um estudo adequado sobre alguns pontos considerados essenciais para criação, sendo eles, escolha do local adequado, conhecimento sobre os concorrentes próximos e a devida divulgação sobre o novo negócio. Baseando-se nesses pontos, surgiu o Serviço de Geoprocessamento de Apoio ao Negócio (SEGAN), com o intuito de auxiliar a criação de novos negócios, disponibilizando aos novos empreendedores a fácil visualização e disposição das empresas no mapa geográfico, possibilitando assim a escolha para o devido local de instalação de novos negócios, estudo sobre os pontos de carência em determinadas áreas das cidades, visualização de possíveis concorrentes próximos, tentando assim reduzir a taxa de descontinuidades dessas novas empresas.

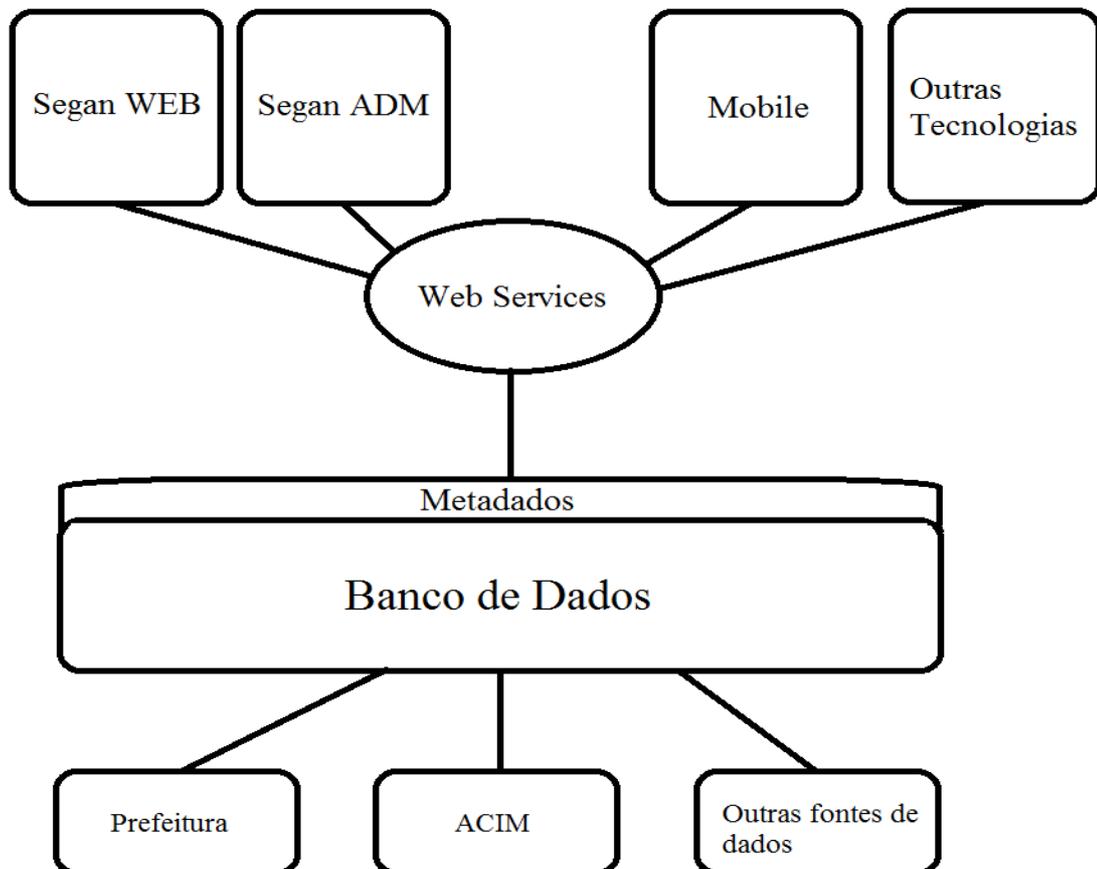
5.1 ARQUITETURA

Arquitetura de sistemas pode estar relacionada aos componentes de software e suas propriedades externas, seus relacionamentos com outros softwares já existentes, relacionamento e propriedades externas. Simplificando a definição, arquitetar é basear-se em informações, padrões, restrições, premissas, e montar uma comunicação entre os elementos existentes ou que serão criados para que, com a união de todos esses itens seja gerado o sistema esperado.

A arquitetura deve ser considerada um ponto fundamental para qualquer sistema de media ou alta complexidade, acredita-se que em sistemas de baixa complexidade não seja extremamente necessário, pois o custo para uma análise arquitetural não justificaria a análise, visto que muitos projetos de baixa complexidade demandam utilização de várias tecnologias e elementos externos maduros e coesos para seu devido funcionamento. Além do mais, arquitetura de sistema serve para prover documentos, especificações, informações, ditar premissas, restrições do projeto em âmbito de requisitos não funcionais ou funcionais do

sistema. É uma forma muito útil, e um forte exercício de arquitetura pode ajudar a todos os envolvidos a enxergar o poder de crescimento da aplicação até atingir as suas características lógicas e físicas.

Um projeto desenvolvido sem análise arquitetônica pode ocasionar problemas futuros em relação à manutenção, fazendo com que a manutenção seja mais árdua e dolorosa.



Na figura consta a arquitetura utilizada no projeto Segan, onde na parte inferior consistem as fontes de dados, que podem ser de diferentes organizações e/ou empresas. No nível acima consta banco de dados no qual é consumido pelo *web service*, disponibilizando assim as informações para as demais aplicações.

5.1.1 ORIENTAÇÃO A OBJETO

A abordagem de orientação a objetos favorece a aplicação de diversos conceitos considerados fundamentais para o desenvolvimento de ótimos sistemas, tais como abstração e encapsulamento, no qual esses conceitos não são exclusivamente da orientação a objetos, mas são melhores suportados no desenvolvimento orientado a objetos do que em outras metodologias.

Abstração consiste em manter o foco nos aspectos essenciais inerentes a uma entidade e desviar o foco sobre as propriedades consideradas “acidentais”. Considerando, em termos de desenvolvimento de sistemas, significa concentrar-se na função em que um objeto deve realizar antes de implementá-lo. O uso de abstração mantém a liberdade para tomar decisões de desenvolvimentos ou implementação apenas quando há um melhor entendimento da inconsistência a ser resolvida. Inúmeras linguagens de programação modernas suportam o conceito de abstração, porém, utilizando-se de abstração juntamente com polimorfismo e herança se torna um mecanismo muito poderoso suportado em orientação a objetos. O uso apropriado de abstração permite que seja utilizado para todas as fases de desenvolvimento de um sistema.

Encapsulamento, também conhecido como esconder informação, consiste em separar os aspectos considerados externos de um objeto, no qual pode ser acessível a outros objetos, dos detalhes considerados internos da implementação do objeto, os quais não são acessíveis a outros objetos. A devida utilização de encapsulamento faz com que o sistema se torne muito interdependente, ou seja, evita que uma pequena mudança tenha imensos efeitos colaterais, permitindo assim uma extrema facilidade para realizar alterações no sistema, tais como, melhoria de desempenho, correções de erros e mudanças de plataformas. Como abstração, o conceito de encapsulamento não é considerado exclusivo da abordagem de orientação a objetos, entretanto, a enorme habilidade de combinar estruturas de dados e comportamentos em uma única entidade, faz com que encapsulamento se torne mais elegante e poderoso em relação a linguagens convencionais que separam estruturas de dados e comportamento.

5.1.2 ARQUITETURA EM TRÊS CAMADAS

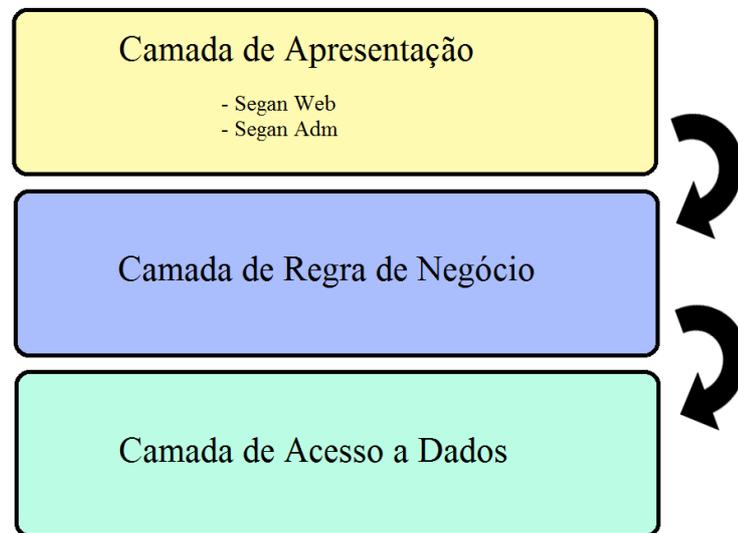
Durante o processo de estudos e de análise sobre qual arquitetura seria desenvolvido o projeto SEGAN, chegamos a conclusão que a melhor arquitetura para utilizarmos seria arquitetura em três camadas, pois possibilita a reutilização do código e facilita a manutenção e o aperfeiçoamento.

A arquitetura em três camadas, como o próprio nome já diz, é dividida em três camadas, Camada de Apresentação, Regra de Negócio e Acesso a Dados. Camada de apresentação oferece o conteúdo estático e conteúdo dinâmico, que pode ser apresentado nos mais variados formatos disponíveis, no projeto foram utilizados os formatos *Web Forms* e *User Control*. O objetivo dessa camada é permitir ao desenvolvedor adquirir produtividade e facilidade no desenvolvimento das interfaces. Essa classe utiliza os recursos disponibilizados pela camada de negócio.

A Camada de Regra de Negócio é responsável por implementar a lógica de negócio da aplicação, no qual ficam disponíveis todas as classes inerentes ao domínio da aplicação, e por fim a Camada de Acesso a Dados é responsável pela persistência e acesso aos dados da aplicação, mantendo o resto da aplicação isolado do meio de armazenamento utilizado, isso possibilita a simples e rápida manutenção caso o meio de armazenamento seja trocado, onde apenas nessa classe deverá ser alterado.

Arquitetura em três camadas tem como objetivo a modularidade, ou seja, dividir a aplicação em módulos o mais independente possível. Tem ainda como objetivo a manutenibilidade, redução dos custos de manutenção da aplicação, extensibilidade, disponibilidade para que novas funcionalidades sejam adicionadas sem grande impacto nas já existentes e reusabilidade, permitindo que classes e componentes sejam reutilizados em outros módulos da aplicação ou em outras aplicações.

Esta arquitetura possui várias vantagens consideráveis, como o desenvolvimento realizado de forma rápida, simples e com custo muito baixo; o acesso à fonte de dados em seu próprio componente de forma que o código da aplicação de frente não possua de maneira alguma código SQL embutido, além da informação da conexão ser mantida apenas em serviços XML, facilitando a manutenção do cliente.



A figura acima descreve a distribuição do projeto em três camadas, na Camada de Apresentação contém as aplicações *Segan Web* e *Segan Adm* (interface visual do projeto), na Camada de Regra de Negócio, possui todas as regras que envolvem o projeto, como validações, tratamentos, etc. Já a camada de Acesso a Dados consiste na comunicação com o *web service* resgatando as informações e exibindo-as na interface gráfica.

5.2 BANCO DE DADOS

Com o surgimento de Sistemas de Gerenciamento de Banco de Dados (SGBD) no início da década de 70, com o objetivo de facilitar a programação de aplicações de Banco de Dados (BD), iniciou-se um investimento considerável em pesquisas na área de BD no qual resultou em um tipo de SGBD, o SGBD Relacional, a partir da década de 80 passou a conquistar o mercado, tornando-se um padrão universal. O projeto de BD, que anteriormente era feito com técnicas empíricas por poucos especialistas no SGBD específico, é executado hoje com auxílio de técnicas padronizadas e suportados por diversas ferramentas. Geralmente, um projeto de BD é feito em três partes Modelagem Conceitual, Projeto Lógico e Projeto Físico.

Modelo conceitual é o informativo do banco de dados de forma independente da implementação de um SGBD, esse modelo registra a forma em que os dados podem aparecer no banco de dados, mas não como estes dados estarão armazenados em nível de SGBD, uma

das técnicas mais utilizadas é a abordagem Entidade-Relacionamento (ER), onde nessa técnica, usualmente é representado por um diagrama, conhecido como diagrama entidade-relacionamento (DER).

Modelo lógico é uma descrição de uma BD no nível de abstração visto pelo SGBD, desta forma o modelo lógico é dependente do SGBD que está sendo utilizado, no caso SQL Server 2008. Na figura 3, podemos visualizar o modelo lógico definido no projeto SEGAN. Finalmente, modelo físico é gerado a partir do modelo lógico e descreve as estruturas físicas do banco de dados, como tipo e tamanho dos campos, nome das tabelas e seus respectivos relacionamentos. Diagrama Entidade Relacionamento (DER) é considerado o modelo diagramático que descreve todo o modelo de dados de um sistema com alto nível de abstração. É a principal representação do Modelo de Entidade e Relacionamento, é utilizado para representar o modelo conceitual do negócio.

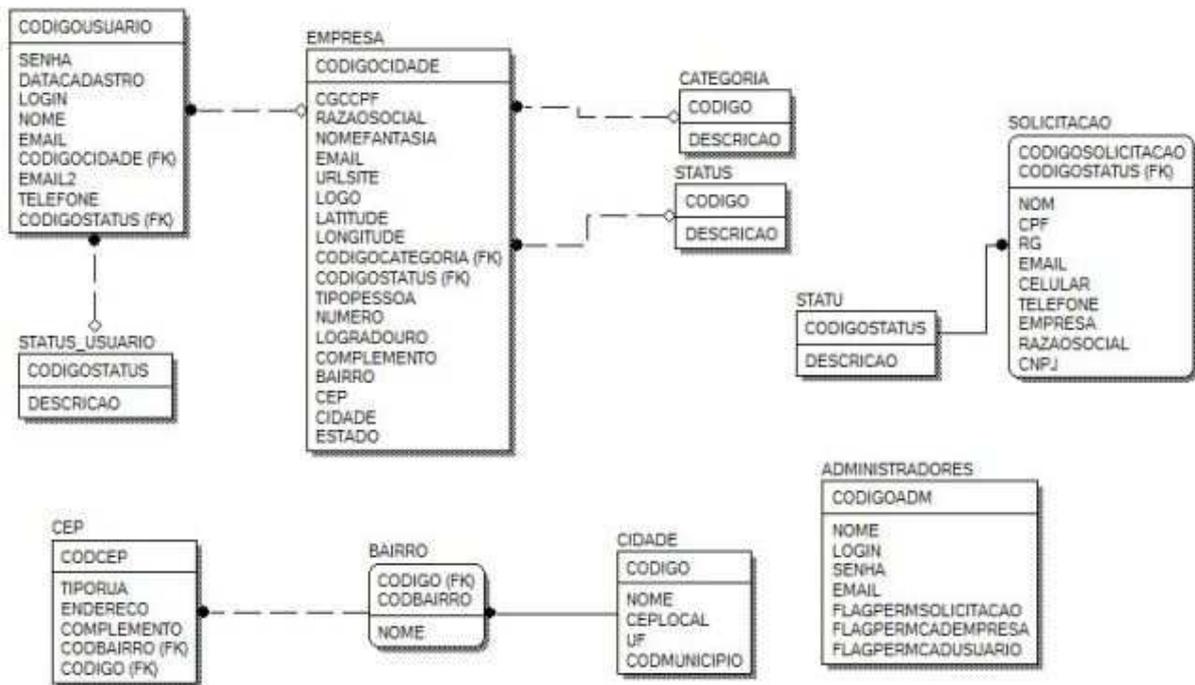


Figura 3 - Modelo Lógico do Banco de Dados

5.3 IMPLEMENTAÇÃO

Neste capítulo descreveremos as tecnologias utilizadas no desenvolvimento do sistema e funcionamento do sistema em si. Como banco de dados, foi utilizado o *MS SQL Server* é um SGBD, ou seja, um sistema gerenciador de Banco de dados Relacional criado pela *Microsoft* em parceria com a *Sybase* em 1988 e inserido como produto complementar do *Windows NT*. Em 1994, a *Microsoft* finalizou a parceria e manteve o aperfeiçoamento do produto. Com essa versão o *Microsoft SQL Server 2008* fornece uma plataforma de dados confiável, produtiva e inteligente que nos permite executar qualquer tipo de aplicação independente de sua exigência.

Utilizando-se do *Visual Studio 2010* (onde consiste em um) *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado (IDE), junto com as seguintes tecnologias, *ASP.NET*, *Microsoft Enterprise Library*, *Ajax*, *CSS*, *JavaScript*, *ASP.NET Web Services* e *Web Service*.

Em Julho de 2002, a *Microsoft* anunciou o início da plataforma *.NET*, que consiste um novo quadro de desenvolvimento, com uma nova interface de programação de serviços *Windows* e APIs, integrando uma série de tecnologias que surgiram durante a década de 90, onde algumas tecnologias foram incorporadas, sendo elas, *COM+ Component Services*, *ASP Web Development Framework*, *XML*, Orientação a Objetos (OO), suporte para novos *Web Services* como SOAP, WSDL e UDDI com foco para a Internet.

A plataforma *.NET Framework* consiste em cinco componentes, conforme exibidos na Figura 4, onde na menor camada se encontra o Sistema Operacional (OS), no qual pode ser uma grande variedade de plataformas de *Windows*.

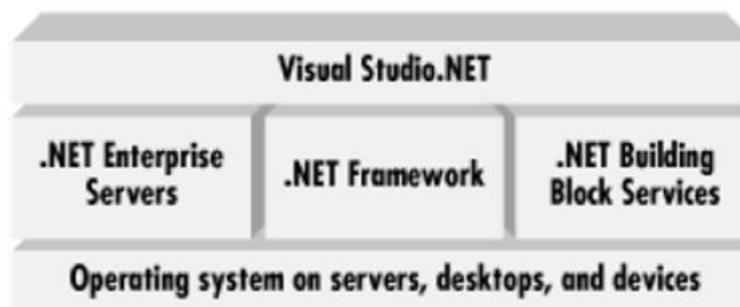


Figura 4 - Plataforma Microsoft .NET.

Na camada superior encontra-se a ferramenta de desenvolvimento Integrado (IDE), *Visual Studio*, que suporta várias linguagens tornando mais rápido o desenvolvimento das aplicações. No centro de tudo, *.NET Framework* é um novo método de desenvolvimento e infra-estrutura em tempo de execução, que mudou todo o desenvolvimento de aplicações de negócio na plataforma *Windows*, incluindo a *Common Language Runtime (CLR)* e um quadro comum de classes que pode ser usados por todo o *.NET Framework*.

Junto com esta tecnologia, surgiu o ASP.NET, baseado no seu descendente *Microsoft Active Server Pages (ASP)*, ele é um servidor *web* que permite tecnologias de *scripts* habilitando dinâmicas páginas da *web*. Uma página ASP contém HTML e os *Server-Side Scripts* que geram dinamicamente o conteúdo HTML. Os *Server-Side Scripts* executam quando a página ASP é acessada, chegando ao servidor *Web*. As solicitações em páginas ASP são enviadas por meio de HTTP POST e Métodos GET. ASP fornece um modelo de objetos para simplificar as obrigações dos desenvolvedores, além de utilizar-se de objetos do ASP *Object Model* como aplicativos, servidores, *Request*, *Response*, *Session* e qualquer componente COM disponível no servidor.

Como evolução desta tecnologia o ASP.NET simplifica o desenvolvimento de páginas *web* com base em formulário de programação, estes formulários são chamados de *Web Forms* e são analogias de *Windows Form*, substituindo as páginas ASP. Os *Web Forms* são baseados em eventos, no qual não é necessário programar *scripts* em linha ASP e conta com *top-down* interpretação de análise como em programação ASP e também permite a separação da lógica da aplicação em camadas.

ASP.NET evolui da programação ASP com os seguintes benefícios:

- Separação clara entre a lógica da aplicação, *Code Behind*, com a camada de apresentação, ou seja, *Code Behind* refere-se ao código da página ASP.NET, que está contido dentro de um arquivo de classe separado, permitindo assim a separação limpa do Código HTML da lógica da apresentação;
- Um rico conjunto de controle de servidores que automaticamente tornam o HTML adequado a todos os clientes e gerenciam seus estados;
- Melhoria no gerenciamento de *Session*;
- Programação baseada em eventos, no qual torna mais simples e intuitivo;
- Lógica da aplicação pode ser escrito em qualquer idioma *Microsoft .NET* (VB, C#, C++ entre outros). Onde o código é compilado para um melhor desempenho;

- *Visual Studio* com uma ferramenta RAD, (*Rapid Application Development* – Desenvolvimento Rápido de Aplicação) o que simplifica o processo de desenvolvimento, visto que é um modelo de processo de desenvolvimento de software interativo e incremental.

O *Microsoft Enterprise Library*, um dos esforços da *Microsoft* ao longo dos últimos anos, tem fornecido aos desenvolvedores, bibliotecas de código úteis que ilustram as melhores práticas, com isso *Patterns and Practices Group*, foi encarregada de desenvolver inúmeros blocos de aplicações, que são bibliotecas *open source*, e apresenta o intuito de resolver as tarefas mais comuns, como a redução dos custos de desenvolvimento e aumentar a confiança.

Como vantagens na utilização do *Microsoft Enterprise Library*, nota-se maior produtividade, cada um dos blocos de aplicação fornece várias interfaces destinadas a satisfazer as preocupações de aplicativos comuns, como *Configuration-Driven Design*, onde é possível lidar adequadamente com uma exceção, não comprometendo as funcionalidades do sistema, além de uma melhor testabilidade do sistema e redução significativa de trabalho do desenvolvedor.

Cascading Style Sheets (CSS) é uma tecnologia que permite criar páginas da *web* de uma maneira mais exata, essa tecnologia permite ainda fazer muitas coisas que não eram possíveis utilizando somente HTML, como por exemplo, incluir margens, tipos de letras, cores, etc. CSS pode ser escrito dentro do código HTML, ou em um arquivo, *Style Sheet*, onde é associado a página HTML. Anteriormente, a estrutura dos sites era feita em incontáveis tabelas, hoje, com a criação de CSS, toda a arte de estruturação de um site contendo inúmeras páginas pode ser feita através de um único documento, diminuindo assim as linhas de código escritas no HTML, definindo todos os tópicos relacionados ao *layout* das páginas. Outro fator importante é o tamanho em *bytes*, como já mencionado, pelos sites não serem mais construídos baseando-se em tabela, a estrutura do site se torna mais leve e dinâmica.

JavaScript, é uma linguagem de programação *Web* que é executado do lado cliente, desenvolvida pela *Netscape*. Essa linguagem possibilita adicionar recursos dinâmicos as páginas HTML. Em sua essência, a linguagem *JavaScript* atua em meios ao código HTML das páginas Web, essa inserção pode ser feita de diversas formas, desde a inserção de código em uma determinada área da página, ou em vários pontos. Uma das vantagens da linguagem *JavaScript* é que não é necessária a instalação de softwares especiais para execução, depende apenas da capacidade dos navegadores em entender as instruções, hoje praticamente todos os navegadores possuem essa funcionalidade. Embora seja uma tecnologia antiga, com o

surgimento de novas técnicas, continua sendo amplamente utilizada para a apresentação dinâmica de algum conteúdo, ou mesmo nas tradicionais verificações de formulários.

O *Asynchronous JavaScript and XML* (AJAX) não é considerado uma tecnologia e sim uma técnica de programação que possui a capacidade de retornar uma informação de forma dinâmica sem a necessidade de recarregar a página e seu conteúdo. Essa técnica utiliza diversas tecnologias citadas anteriormente, principalmente *JavaScript*, gerenciando a comunicação entre servidor e cliente e XML encapsulando a informação, cada uma a evoluir a sua maneira e a convergir de uma maneira vantajosa, podemos citar também:

- Apresentação baseada em padrões, que utilizam XHTML e CSS;
- Exibição e interação dinâmica por meio de *Document Object Model* (DOM);
- Troca e manipulação de dados por meio de uso de XML e XSLT;
- Recuperação assíncrona de dados com *XMLHttpRequest*;
- E *JavaScript* que une todos os pontos mencionados.

A implementação do AJAX é baseada na classe *XMLHttpRequest*, que interage entre o servidor e o *browser* (cliente). As grandes vantagens de utilizar AJAX, é que permite proceder as alterações na página com muito menos transferência de informação, pois podem ser transferidos apenas os dados novos em cada alteração, essa atualização normalmente é muito mais rápida do que se toda a página fosse atualizada, isso faz com que o usuário tenha a sensação de que tudo acontece em tempo real. Não há necessidade de instalação ou configurações especiais para a execução do AJAX, ou ficar preso a apenas uma tecnologia, todas as tecnologias utilizadas no desenvolvimento são multi-plataformas e são executados na maioria dos *browsers*. Como o AJAX é composto por várias tecnologias, cada qual com sua função, fica fácil a sua manutenção.

A estrutura ASP.NET simplifica o desenvolvimento de ASP.NET *Web Services*, todo o trabalho de baixo nível, como *packaging and unpacking* dados em formato XML e utilizando o protocolo HTTP para transportar as mensagens entre os componentes distribuídos, são feitos pela estrutura, isso permite que os desenvolvedores se concentrem apenas na lógica da aplicação. O *NET Framework* usa *asmx* como extensão de arquivo padrão pra *web services*, ao contrário de *aspx* para *Web Forms* e *ascx* para controle *web*.

Todos os *asmx* começam com a diretiva *@WebService* que instrui o ASP.NET em como compilar o código, bem como o nome da classe principal. Um *web service* possui os seguintes atributos, Linguagem, Classe e *Code Behind*.

Linguagem especifica o código que foi escrito, desse modo instrui o *Framework* ASP.NET para usar o compilador apropriado para construir o *web service*; Classe especifica o conjunto principal que compõem os métodos da web. *Framework* ASP.NET instancia essa classe, a fim de servir os métodos da web para os clientes e *Code Behind*, especifica o arquivo de origem para o seu código, o que permite a separação entre o código e a página ASP.

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
[System.Web.Script.Services.ScriptService()]
public class CascadingDropdown : System.Web.Services.WebService
{
    [WebMethod]
    public CascadingDropDownNameValue[] carregaEstado(string knownCategoryValues, string category)
    {
        DataTable dtEstados = new DAL.CidadesDAL().getEstados(string.Empty);
        //
        List<CascadingDropDownNameValue> estados = new List<CascadingDropDownNameValue>();
        foreach (DataRow dtrow in dtEstados.Rows)
        {
            string codigo = dtrow["CODIGO"].ToString();
            string descricao = dtrow["DESCRICAO"].ToString();
            estados.Add(new CascadingDropDownNameValue(codigo, descricao));
        }
        return estados.ToArray();
    }
}
```

Figura 5 - Exemplo de Web services utilizado no projeto SEGAN, utilizado para carregar os estados durante a Alteração dos dados cadastrais da empresa.

Web Services, conforme já dito, permitem o acesso a componentes de *software* por meio de padrões da *web*, como o HTTP e SMTP. Usando a internet e XML, possibilitando a criação de componentes de software que se comunicam com os outros, independente da linguagem, cultura ou plataforma. *Web services* combinam o melhor de ambos componentes distribuídos e *World Wide Web* (WWW), ele estende à computação distribuída as faixas mais amplas de aplicativos clientes.

O *Framework* ASP.NET disponibiliza uma interface *web*, que disponibiliza os métodos desenvolvidos possibilitando visualizar o retorno dos métodos.

Service1

Não há suporte para as operações a seguir. Para obter uma definição formal, examine a [Descrição de Serviço](#).

- [getCategorias](#)
- [getEmpresas](#)

Métodos disponíveis no Web Service

Este serviço da Web está usando <http://tempuri.org/> como espaço para nome padrão.

Recomendação: altere o espaço para nome padrão antes que o Serviço da Web em XML seja publicado.

Cada Serviço da Web em XML precisa de um espaço para nome exclusivo para que os aplicativos cliente o diferenciem de outros serviços.

O Serviço da Web em XML deve ser identificado por um espaço para nome controlado por você. Por exemplo, você pode usar o nome de recursos reais na Web (espaços para nome dos Serviços da Web em XML são URIs).

Na criação dos Serviços da Web em XML com o ASP.NET, o espaço para nome padrão pode ser alterado usando a propriedade `Namespace` "http://microsoft.com/webservices/":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementação
}
```

Visual Basic

```
<WebService(Namespace="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementação
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementação
};
```

Para obter mais detalhes sobre espaços para nome XML, consulte a recomendação W3C em [Namespaces in XML](#) (site em inglês).

Para obter mais detalhes sobre WSDL, consulte [WSDL Specification](#) (site em inglês).

Para obter mais detalhes sobre URIs, consulte [RFC 2396](#) (site em inglês).

Figura 6 - Página disponibilizada pelo Framework ASP.NET onde é possível realizar os devidos testes com os métodos desenvolvidos.

Nota-se na Figura 6, que todos os métodos desenvolvidos no *web service* foram disponibilizados na página permitindo assim realizar a consulta.

Service1

Clique [aqui](#) para obter uma lista completa das operações.

getEmpresas Nome do Método

Testar

Para testar a operação usando o protocolo HTTP.POST, clique no botão 'Chamar'.

Parâmetro	Valor
strCategoria:	<input type="text"/>
strEmpresa:	<input type="text"/>
strLogradouro:	<input type="text"/>
strBairro:	<input type="text"/>

Parâmetros solicitados pelo método, no exemplo será utilizado como Filtro para realizar a busca das empresas.

SOAP 1.1

O exemplo a seguir mostra uma solicitação e uma resposta SOAP 1.1. Os espaços reservados mostrados devem ser substituídos por valores reais.

```
POST /WebServices.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/getEmpresas"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getEmpresas xmlns="http://tempuri.org/">
      <strCategoria>string</strCategoria>
      <strEmpresa>string</strEmpresa>
      <strLogradouro>string</strLogradouro>
      <strBairro>string</strBairro>
    </getEmpresas>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getEmpresasResponse xmlns="http://tempuri.org/">
      <getEmpresasResult>
        <xsd:schema>schema</xsd:schema></getEmpresasResult>
      </getEmpresasResponse>
    </soap:Body>
  </soap:Envelope>
```

SOAP 1.2

O exemplo a seguir mostra uma solicitação e uma resposta SOAP 1.2. Os espaços reservados mostrados devem ser substituídos por valores reais.

```
POST /WebServices.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

Figura 7 - Parâmetros solicitados pelos métodos do Web Service.

Na Figura 7 podemos visualizar todos os parâmetros solicitados pelos métodos, no caso esses parâmetros são solicitados para realizar o filtro na busca das empresas. Caso não seja informado nada como parâmetro será retornado para todas as empresas cadastradas.

Na Figura 8 é exibido o XML de retorno do *Web Service*, neste caso é disponibilizado os dados cadastrais das empresas de acordo com o filtro realizado anteriormente.

```

▼<DataSet xmlns="http://tempuri.org/"
▼<xs:schema xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" id="NewDataSet">
▼<xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="">
▼<xs:complexType>
▼<xs:choice minOccurs="0" maxOccurs="unbounded">
▼<xs:element name="Table">
▼<xs:complexType>
▼<xs:sequence>
<xs:element name="NOMEFANTASIA" type="xs:string" minOccurs="0"/>
<xs:element name="ENDERECO" type="xs:string" minOccurs="0"/>
<xs:element name="COMPLEMENTO" type="xs:string" minOccurs="0"/>
<xs:element name="BAIRRO" type="xs:string" minOccurs="0"/>
<xs:element name="LATITUDE" type="xs:double" minOccurs="0"/>
<xs:element name="LONGITUDE" type="xs:double" minOccurs="0"/>
<xs:element name="URLSITE" type="xs:string" minOccurs="0"/>
<xs:element name="LOGO" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
▼<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
▼<NewDataSet xmlns="">
▼<Table diffgr:id="Table1" msdata:rowOrder="0">
<NOMEFANTASIA> </NOMEFANTASIA>
<ENDERECO> </ENDERECO>
<COMPLEMENTO/>
<BAIRRO>I </BAIRRO>
<LATITUDE>-22.2071958</LATITUDE>
<LONGITUDE>-49.9515616</LONGITUDE>
<URLSITE/>
<LOGO> - </LOGO>
</Table>
▼<Table diffgr:id="Table2" msdata:rowOrder="1">
<NOMEFANTASIA> </NOMEFANTASIA>
<ENDERECO> </ENDERECO>
<COMPLEMENTO/>
<BAIRRO> </BAIRRO>
<LATITUDE>-22.2163454</LATITUDE>
<LONGITUDE>-49.9519395</LONGITUDE>
<URLSITE/>
<LOGO> </LOGO>
</Table>
▼<Table diffgr:id="Table3" msdata:rowOrder="2">
<NOMEFANTASIA> </NOMEFANTASIA>
<ENDERECO> </ENDERECO>
<COMPLEMENTO/>
<BAIRRO> </BAIRRO>
<LATITUDE>-22.2168371</LATITUDE>
<LONGITUDE>-49.9385965</LONGITUDE>
<URLSITE/>
<LOGO> - </LOGO>

```

Figura 8 - XML de retorno do Web Service

Um ponto crucial para o funcionamento do sistema como um todo, é a utilização das APIs *Google Maps*, em Fevereiro de 2005 foi lançado o *Google Maps*, ainda em sua versão beta, tornando-se rapidamente uma eminente referência. Contando com uma interface rica e interativa, a aplicação permitia o acesso a uma gigantesca base de dados contendo uma infinidade de cidades, bairros, ruas e avenidas nos Estados Unidos. Com o tempo, a esperada evolução foi ocorrendo e novas funcionalidades foram adicionadas ao sistema. Em meados de

maio de 2007, o *Google Maps* finalmente chegou ao Brasil, disponibilizando as consultas de endereços. Em outubro de 2007, uma versão estendida e totalmente traduzida para o português, foi oferecida ao público brasileiro com a possibilidade de localizar restaurantes, hotéis, traçar rotas, entre outras utilidades. Com isso, cada vez mais os sites passaram a utilizar esse serviço com as mais diversas finalidades.

A primeira versão da *Application Programming Interface* (API) foi disponibilizada em Junho de 2005, desde então, inúmeras funcionalidades foram apropriadas, culminando na versão (3.9) atual, liberada em 14 de agosto de 2012. A API consiste basicamente em um conjunto de classes *JavaScript* que fornece as ferramentas necessárias para que o desenvolvedor possa construir aplicações para exibir os mapas, realizar consultas de endereços, disponibilizar recursos como zoom, adição de pontos de referências e descrições no mapa dentre outras possibilidades. Tudo partindo do princípio do AJAX, como já mencionado.

6. SEGAN WEB

Com o intuito de auxiliar na tomada de decisão, conforme já mencionado, o *SEGAN WEB* conta com a poderosa ferramenta *Google Maps*, que disponibiliza todas as empresas cadastradas na base de dados (mencionados anteriormente) no mapa em seus respectivos locais.

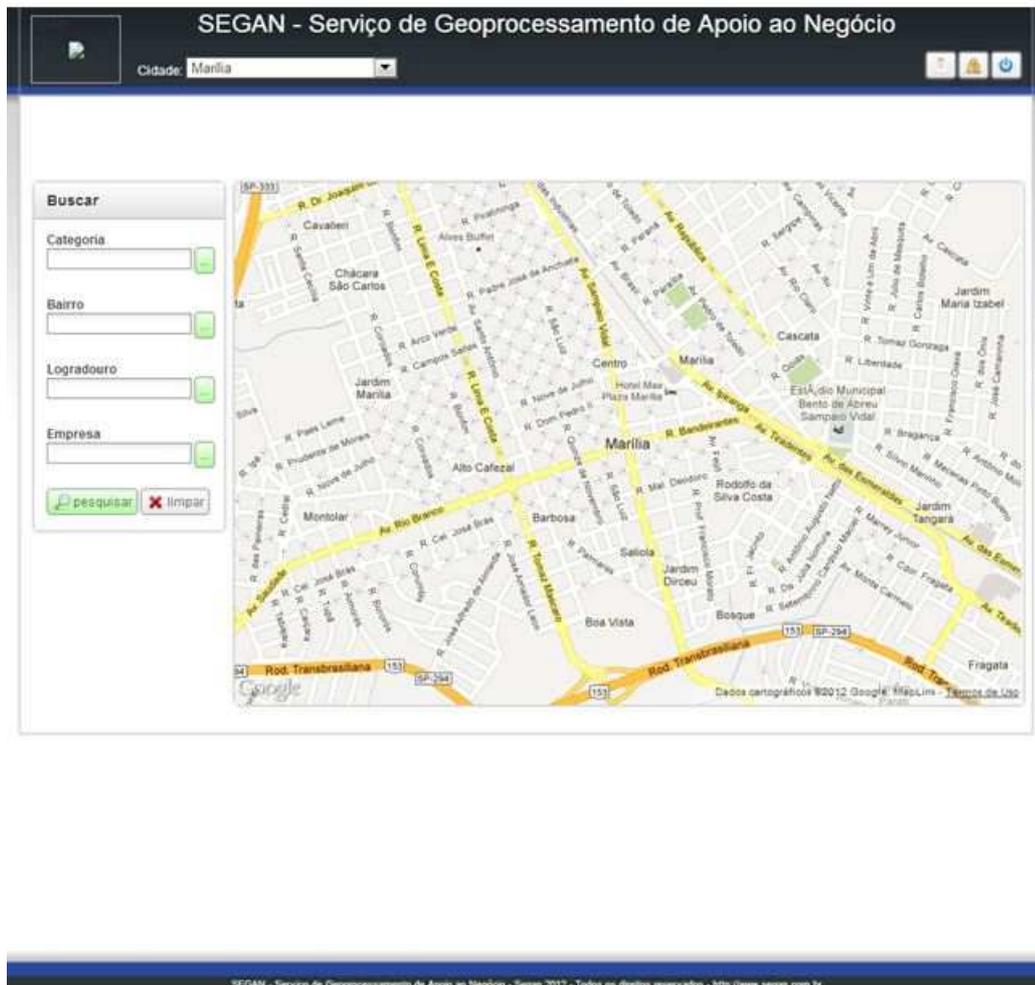


Figura 9 - Abertura do sistema, tela principal.

Contando com quatro opções para buscar as empresas, sendo elas: categoria, bairro, logradouro e empresa. Junto com uma opção para realizar busca específica, onde o sistema auxilia o usuário disponibilizando as informações cadastradas, de acordo com o filtro selecionado.

Os filtros são realizados de maneira cascadeada, ou seja, caso esteja sendo utilizado mais de um filtro o sistema irá considerar todos os filtros para retornar a informação ao mapa. Por exemplo, ao selecionar uma categoria e um bairro serão exibidas todas as empresas que se enquadrem nessas seleções.

Busca Específica - Logradouros

Pesquisar:

DESCRICAÇÃO	
✓	AL FERNAO DIAS
✓	ALAMEDA SANTA CAROLINA
✓	AV CELESTE CASAGRANDE
✓	AV DOUTOR HERCULES GALLETTI
✓	AV ERNESTO DAUN
✓	AV EXPEDICIONARIO DE POMPEIA
✓	AV PERIMETRAL
✓	AVENIDA AVENIDA VICENTE FERREIRA,
✓	AVENIDA MANOEL THOMAZ DA SILVA
✓	AVENIDA ALCEBIADES SPADOTTO

Página 1 de 40 Ir para a página

Figura 10 - Busca Específica.

Para realizar a exibição da(s) empresa(s) no mapa, o sistema baseia-se nas informações básicas cadastradas como, logradouro, número, cidade, estado e País, utilizando um dos serviços da API *Google Maps*, Serviço de Geocodificação, que é o processo de conversão de endereços em coordenadas geográficas, latitude e longitude, possibilitando a exibição no mapa. O acesso a esse serviço é feito de forma assíncrona, pois a API *Google Maps* precisa fazer uma chamada a um servidor externo, por esta razão, durante a inclusão dos dados na base do sistema, todas as empresas já foram mapeadas, ou seja, para inclusão de cada empresa foi necessário a utilização desse serviço, a fim de evitar possíveis lentidões durante o processo de exibição no mapa.

Para utilização desse serviço é necessário enviar um método de retorno de chamada para execução e após a conclusão do pedido, é retornado um objeto que contém as informações enviadas ao método e suas respectivas coordenadas geográficas. Podemos visualizar na Figura 11 o retorno desse método.

```

{
  "name": "rua nove de julho , marília - SP, BRASIL",
  "Status": {
    "code": 200,
    "request": "geocode"
  },
  "Placemark": [ {
    "id": "pl",
    "address": "R. Nove de Julho - Marília - São Paulo, República Federativa do Brasil",
    "AddressDetails": {
      "Accuracy" : 6,
      "Country" : {
        "AdministrativeArea" : {
          "AdministrativeAreaName" : "São Paulo",
          "Locality" : {
            "LocalityName" : "Marília",
            "Thoroughfare" : {
              "ThoroughfareName" : "R. Nove de Julho"
            }
          }
        }
      },
      "CountryName" : "República Federativa do Brasil",
      "CountryNameCode" : "BR"
    }
  },
  "ExtendedData": {
    "LatLonBox": {
      "north": -22.2117432,
      "south": -22.2225092,
      "east": -49.9392325,
      "west": -49.9615926
    }
  },
  "Point": {
    "coordinates": [ -49.9502066, -22.2181002, 0 ]
  }
} ]
}

```

Figura 11 - Retorno do Método utilizando Serviço de Geocodificação.

Após a localização das coordenadas geográficas, latitude e longitude, é exibida no mapa a localização de cada empresa. Possui os recursos básicos do *Google Maps*, como adicionar ou remover zoom, seleção do tipo de mapa, podendo alterar para Satélite. Além do mais, para uma fácil assimilação do usuário, junto com a localização da empresa são exibidos os dados principais cadastrados de cada empresa, dados como, nome fantasia, endereço, número, caso possua cadastrado, URL Site e logo.



Figura 12 - Exibição da empresa no mapa junto aos dados cadastrados.

SEGAN WEB possui uma área destinada às empresas, onde as mesmas podem enviar solicitações aos administradores do sistema, solicitações para acesso a área restrita disponível. Estas solicitações servem para que as empresas já cadastradas ou novas empresas solicitem o acesso ao sistema, podem então ter suas informações disponibilizadas no mapa. Para realizar a solicitação é necessário informar dados básicos como, nome, CPF, RG, e-mail, telefone, celular e a empresa, caso não possua empresa cadastrada ainda é possível realizar a solicitação como uma nova empresa, onde são solicitados CNPJ e razão social da empresa. Para garantir a integridade da solicitação, são válidas informações básicas como CPF, RG, e-mail e CNPJ, além do preenchimento obrigatório dos outros campos. Após o preenchimento dos dados e confirmado o envio da solicitação, o sistema automaticamente envia um e-mail ao usuário, que realizou a solicitação e para o administrador do sistema informando sobre essa nova solicitação.



SEGAN - Serviço de Geoprocessamento de Apoio ao Negócio

Cidade: Manlia

Solicitação de Acesso

Nome CPF RG Email

Telefone Celular Empresa

Figura 13 - Solicitação de acesso à área restrita

O *SEGAN WEB* conta também com uma área restrita destinada às empresas, que possibilita as empresas alterarem as informações cadastradas, inclusive inserir uma logo a fim de exibi-la no mapa. Para o controle de acesso ao sistema, tanto para *SEGAN Web* quanto *SEGAN ADM* é utilizada funcionalidade robusta de autenticação nativa do ASP.NET, onde é possível restringir o acesso as páginas ASP.NET, colocando-os em uma pasta protegida liberando o acesso apenas para usuários autenticados, bloqueando o acesso a usuários não autenticados e forçando o redirecionamento à autenticação, permitindo funcionalidades básicas, como manter usuário autenticado por um determinado período e alteração de senhas. A gravação da senha no banco de dados é feita de maneira criptografada, a fim de manter a credibilidade do sistema. Após acesso bem sucedido é disponibilizado uma nova opção, que permite alteração dos dados cadastrais das empresas.

SEGAN - Serviço de Geoprocessamento de Apoio ao Negócio

Cidade: Marília

Dados Cadastrais

CNPJ:

Razão Social:

Nome Fantasia:

Email:

Logradouro:

Número:

Compl.:

Bairro:

CEP:

Estado:

Cidade:

Categoria:

Status:

Tipo Pessoa:

URL Site:

Logo:

gravar

Figura 14 - Dados Cadastrais Empresas.

6.1 SEGAN ADM

Baseando-se na ideia inicial utilizado pelo *SEGAN Web*, foi necessário o surgimento dessa ferramenta que permite aos administradores do *SEGAN Web*, gerenciarem os dados e solicitações enviadas através do *SEGAN WEB*. *SEGAN ADM* é um sistema que se baseia em estrutura *Create, Read, Update e Delete* (CRUD) onde contempla as quatro operações básicas utilizadas em banco de dados relacional, ou seja, permite criar, ler, atualizar e apagar informações referentes às empresas e seus respectivos usuários.

Conforme mencionado anteriormente, *SEGAN ADM* utiliza o mesmo método para autenticação de usuários como o *SEGAN WEB*, utilizando-se dos recursos nativos do *Framework ASP.NET*.



Figura 15 - Tela inicial SEGAN ADM.

Após ser efetuado o acesso ao sistema, este permite ao administrador efetuar as devidas manutenções e consultas às informações cadastradas. Na página inicial, encontram-se disponíveis para utilização as seguintes opções: Solicitações, Cadastro de Empresa e Cadastro de Usuários.

Solicitações permitem aos administradores responderem às solicitações feitas através do *SEGAN Web*, no qual os administradores consultam e realizam o procedimento necessário, podem alterar o status da solicitação de acordo com o atendimento realizado.

The screenshot shows the SEGAN web application interface. At the top, there is a header with the title "SEGAN - Serviço de Geoprocessamento de Apoio ao Negócio" and a welcome message: "Seja bem vindo, ADM. Hoje é sexta-feira, 16 de novembro de 2012." Below the header, there is a navigation bar with icons for mail, documents, and user profile. The main content area is titled "Solicitações Pendentes" and contains a table with the following columns: NOME, CPF, RG, EMAIL, TELEFONE, and CELULAR. The first row of the table has a green checkmark in the first column. Below the table, there is a form with several input fields: "Nome Solicitante", "CPF", "RG", "Email", "Telefone", "Celular", "Empresa", and "CNPJ Empresa". There is also a "Status Solicitação" dropdown menu currently set to "AGUARDANDO" and a "gravar" button.

Figura 16 - Tela de Solicitações, permitindo consulta e alteração.

Outra ferramenta disponível no sistema é a de cadastro de empresas, que permite ao administrador do sistema realizar consulta nas empresas já cadastradas, incluindo sistema de busca, realização de alterações caso necessário, além da opção de inclusão de uma nova empresa. Esta funcionalidade contempla um sistema CRUD completo. São feitas validações nos campos considerados essenciais com objetivo de garantir a integridade da aplicação. Durante a alteração ou inclusão de empresas é realizado o procedimento citado acima, para resgatar as coordenadas geográficas, latitude e longitude, possibilitando assim a exibição no mapa. Um ponto interessante nessa tela, é a utilização de *CascadingDropDown*, uma extensor AJAX para componentes ASP.NET, que tem como objetivo popular automaticamente um conjunto de controles *DropDownList*. A cada seleção é disparado uma chamada para um serviço *web* específico para recuperar a lista de valores para o próximo componente, este serviço foi utilizado no relacionamento entre estado e cidade, conforme exibido na Figura 18.

SEGAN - Serviço de Geoprocessamento de Apoio ao Negócio
Seja bem vindo, ADM. Hoje é sexta-feira, 16 de novembro de 2012.

Cadastro de Empresas

Pesquisar:

RAZAO/SOCIAL	DESCRICAO
✓	
✓	
✓	
✓	

Página 1 de 164 Ir para a página

CNPJ Razão Social Nome Fantasia Email

Logradouro Número Compl. Bairro

CEP Estado Cidade Categoria

Status Tipo Pessoa URL Site

Logo

Figura 17 - Cadastro de Empresas e exemplo de AJAX CascadingDropDown

Última opção disponível no sistema SEGAN ADM, é o Cadastro e Manutenção de usuários, essa funcionalidade do sistema permite ao administrador cadastrar, atualizar e excluir usuários que são utilizados para acesso ao sistema *SEGAN Web*. Os campos considerados essenciais são obrigatórios e o seu não preenchimento possibilita a ruptura da manutenção. Em relação à senha, o sistema criptografa a mesma para evitar possíveis fraudes a fim de manter a devida credibilidade ao sistema. Caso seja realizada uma consulta simples no banco de dados, não é possível verificar ou até mesmo recuperar a senha cadastrada para o usuário, conforme demonstrado na figura 10.

CONCLUSÃO

Tendo em vista as premissas iniciais para o desenvolvimento deste projeto obtém-se o resultado de uma aplicação *web*, de fácil assimilação, no qual disponibiliza os dados das empresas pré-cadastradas no mapa geográfico utilizando-se de métodos de busca, contando com outras opções como solicitações de acesso e alteração dos dados cadastrais.

Além de um projeto voltado para área administrativa da aplicação, contando com um completo *create, read, update, delete* (CRUD), que permita aos administradores do sistema atender as solicitações feitas por meio de uma segunda aplicação, *SEGAN WEB*, inclusão de novas empresas, alteração e até mesmo exclusão das mesmas, adição e manutenção dos usuários do sistema, possibilitando assim o acesso a área restrita na aplicação *SEGAN WEB*, utilizando-se de um *Web Service*, no qual possibilite a expansão do projeto para novas tecnologias.

A utilização das funções descritas nesse projeto fornece aos usuários um ambiente propício à identificação da melhor localização para instalação de novos negócios, possibilitando a identificação de possíveis e próximos concorrentes, além da devida divulgação sobre o próprio negócio.

Tem-se em vista melhorias na aplicação administrativa por meio da disponibilização de acesso a relatórios, novas consultas e manutenções, caminhando lado a lado com a aplicação *SEGAN WEB* sempre se atualizando com as novidades lançadas no mercado. Além do desenvolvimento de novas funcionalidades ao projeto, como consultas de informações temporais e demográficas, ou seja, exibir a quantidade de novas empresas de determinado ramo, em determinadas regiões da cidade, determinar ainda indicadores e relatórios gráficos visando movimentação econômica, gráficos de acesso ao sistema. Bem como, possíveis pesquisas para identificar a satisfação do usuário ao utilizar o sistema, tendo em vista o auxílio aos novos empreendedores, focando sempre na constante atualização, tentando assim evitar que o sistema se torne obsoleto e inutilizável e também possibilitar a expansão para novas tecnologias e, finalmente possibilitar a criação do serviço para aplicações móbile.

REFERÊNCIAS

ALMEIDA, M. B. A necessidade de integração de fontes heterogêneas de dados em projetos de gestão de conhecimento. *Perspectivas em Ciência da Informação*. v. 7, n. 1, 2002. Disponível em: <<http://www.eci.ufmg.br/pcionline/viewarticle.php?id=373>>. Acesso em: 1 nov. 2012.

ALMEIDA, P. et al. *SInBAD – Sistema integrado de biblioteca e arquivo digital*. 2007. Disponível em: <<http://vecpar.fe.up.pt/xata2006/papers/6.pdf>>. Acesso em: 2 nov. 2012.

ALVES, M. das D. R.; SOUZA, M. I. F. Estudo de correspondência de elementos metadados: Dublin Core e MARC 21. *Revista Digital de Biblioteconomia e Ciência da Informação*. v. 4, n.2, p. 20-38, 2007. Disponível em: <<http://server01.bc.unicamp.br/seer/ojs/viewarticle.php?id=94>>. Acesso em: 2 nov. 2012.

ALVES, R. C. V. *Web semântica: uma análise focada no uso de metadados*. 2005. 180 f. Dissertação (Mestrado em Ciência da Informação) - Faculdade de Filosofia e Ciências - Universidade Estadual Paulista, Marília, 2005.

ALECRIM, Emerson. Linguagem JavaScript.<<https://fit.faccat.br/~jonis/javascript.pdf>>. Acesso em Novembro de 2012.

ALLEN, Kms. GREEN, SW, and Zubrow EBW (eds.) (1990) *Interpreting Space: GIS and Archaeology*. New York: Taylor and Francis.

ANGEL, Miguel Alvarez. *O que é CSS*. <<http://www.criarweb.com/artigos/173.php>>. Acesso em Novembro de 2012.

_____. *O que é XML*.<<http://www.criarweb.com/artigos/203.php>>. Acesso em Novembro de 2012.

ARMS, W. A spectrum of interoperability: the site for science prototype for the NSDL. *Dlib Magazine*, v. 8, n. 1, jan. 2002.

BAILEY, TC and GATRELL, AC (1995) *Interactive Spatial Data Analysis*. Harlow, UK: Longman.

BARRETO, C. M. *Modelo de metadados para a descrição de documentos eletrônicos na web*. 1999. 189 f. Dissertação (Mestrado em Ciências em Sistemas de Computação)–Instituto Militar de Engenharia, Rio de Janeiro, 1999.

BJERKESTRAND, A. A., ZEID, A., SKAAR, L. A. A., & Zimmermann, O. (2007). *Companion to the 22nd ACM SIGPLAN conference on object oriented programming systems and applications companion - OOPSLA '07; fifth international workshop on SOA & web services best practices*. Paper presented at the 746. doi:10.1145/1297846.1297866

BURKE. LM. (1993). *Environmental Equity in Los Angeles*. Unpublished M.A. thesis, Santa Barbara, University of California.

CROMWELL-KESSLER W. Correspondências entre metadatos e interoperabilidade: que significa todo esto?. In: BACA, M. (Ed.). *Introducción a los metadatos vias a la información digital*. Traducido al español por Marisol Jacas-Santoll. Los Angeles, CA: J. Paul Getty Trust, 1998.

DCMI. *DCMI Frequently Asked Questions (FAQ)*. Dublin Core Metadata Initiative. 2008. Disponível em: <<http://dublincore.org/resources/faq/index.shtml>>. Acesso em: 3 nov. 2012.

DEITEL, H.M. *Internet & World Wide Web – Como Programar*. 2. Porto Alegre, RS, Brasil: Ed. Bookman, 2003.

EIS, Diego. *Browsers e seus diferenciais*. <<http://www.tableless.com.br/>>. Acesso em Novembro de 2012.

ELMASRI, R. & NAVATHE, S.B. *Fundamentals of Database Systems*. Second Edition. Benjamin/Cummings, Redwood City, California, 1994.

ERL, T. *SOA patterns*. Disponível em: <<http://www.soapatterns.org/>> Acesso em 11 nov. 2012.

FREIRE, H. *Web Services: A Nova Arquitetura da Internet*. Developer's CIO Magazine, São Paulo, N.73, p.24-25, Setembro, 2002.

GARCIA, P. de A. B. e SUNYE M. S. *O protocolo OAI-PMH para Interoperabilidade em Bibliotecas Digitais*. 2007. Disponível em: <http://conged.deinfo.uepg.br/~iconged/Artigos/artigo_09.pdf>. Acesso em: 3 nov. 2012.

GERIC, S., & VRCEK, N. (2009). *Prerequisites for successful implementation of service-oriented architecture*. Paper presented at the Information Technology Interfaces, 2009. ITI '09. Proceedings of the ITI 2009 31st International Conference on, 175-180.

GOODCHILD, MF (2000) *Communicating geographic information in a digital age*. Ann. Assoc. Am. Geogr. 90: 344–355. King G (1997) *A Solution to the Ecological Inference Problem: Reconstructing Individual Behavior from Aggregate Data*. Princeton, NJ: Princeton University Press.

GOOGLEMapsJavascriptAPIV3Reference.<<https://developers.google.com/maps/documentation/javascript/3.exp/reference?hl=pt-PT>>Acesso em Julho de 2012.

GRACIO, J. C. A. *Metadados para a descrição de recursos da Internet: o padrão Dublin Core, aplicações e a questão da interoperabilidade*. 2002. 127f. Dissertação (Mestrado em Ciência da Informação) - Faculdade de Filosofia e ciências, Universidade Estadual Paulista, Marília, 2002.

GUIMARÃES, Renato. *ASP.NET - Uma mudança radical no desenvolvimento web*. <<http://www.imasters.com.br/artigo/1624>>. Acesso em novembro de 2012.

KORTH, H. & SILBERSCHATZ, A. *Sistemas de Bancos de Dados*. 2ª edição, Makron Books, 1994.

LONGLEY, PA; GOODCHILD, MF; MAGUIRE, DJ and RHIND, DW (2001). *Geographic Information Systems and Science*. New York: Wiley.

MARC21. *MARC 21 Specifications for Record Structure, Character Sets, and Exchange Media*. Library of Congress. Network Development and MARC Standards Office. 2000. Disponível em: <<http://www.loc.gov/marc/specifications/>>. Acesso em: 4 nov. 2012.

MARCHEZAN, M. L. e BERMEJO, P. H. S. *Web Services: Conceitos e aplicação ao Portal ScienTI*. INSTITUTO STELA. 2007. Disponível em: <<http://espacio.bvsalud.org/files/0/5/061207012005/WSPortalScienTI-BermejoMarchezan.ppt>>. Acesso em: 6 nov. 2012.

MARCONDES, C. H. e SAYÃO, L. F. *Integração e interoperabilidade no acesso a recursos informacionais eletrônicos em C&T: a proposta da Biblioteca Digital Brasileira*. Ciência da Informação. V. 30, N° 3 (2001). Disponível em: <<http://revista.ibict.br/ciinf/index.php/ciinf/article/view/190/167>>. Acesso em: 5 nov. 2012.

_____. *Documentos digitais e novas formas de cooperação entre sistemas de informação em C&T*. *Ci Inf*, Brasília, 2002; 31 (3): 42-54, set./dez.

MARINO, M. T. *Integração de informações em ambientes científicos na web: uma abordagem baseada na arquitetura RDF*. 2001. 122 f. Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2001.

MARTINI, R. *Governo Eletrônico e Interoperabilidade*. Portal do SERPRO. Serviço Federal de Processamento de Dados (SERPRO). Tema 173 - N° 73 – 2004. Disponível em: <<http://www.serpro.gov.br/publicacao/tematec/tematec/2004/ttec73>>. Acesso em 6 nov. 2012.

MAURO, C., LEIMEISTER, J. M., & KRCCMAR, H. (2010). *Service oriented device integration - an analysis of SOA design patterns*. Paper presented at the System Sciences (HICSS), 2010 43rd Hawaii International Conference on, 1-10.

MELGAR, Robinson. *Ajax Asynchronous Javascript and XML*. Revista *www.com.br*, n. 69, p.26-27, abril. 2006. [LARMAN 2003] Larman, Craig. : *A Manager's Guide*. Addison-Wesley Professional, 2003.

MIRANDA, A. *A integração de serviços bibliotecários e de informação e o acesso ao documento primário: evolução dos conceitos e situação atual no Brasil*. Anais do 2º Seminário Nacional de Comutação Bibliográfica, 1994; Campinas; Brasil, 1994.

MOURA, A. M. A. de C. *A Web Semântica: fundamentos e tecnologias*. [S. l.: S. n.], 2002.

NIANJUN, Zhou & LIANG-JIE Zhang. (2009). *Analytic architecture assessment in SOA solution design and its engineering application*. Paper presented at the Web Services, 2009. ICWS 2009. IEEE International Conference on, 807-814.

OLIVEIRA, V S. *Buscando Interoperabilidade entre diferentes bases de dados: O caso da Biblioteca do Instituto Fernandes Figueira*. 2005. 109f. Dissertação (Mestrado em Gestão da

Informação e Comunicação em Saúde) - Escola Nacional de Saúde Pública Sérgio Arouca - FIOCRUZ, Rio de Janeiro, 2005.

PAEPCKE, A.; BRANDRIFF, R.; JANEE, G.; LARSON, R.; LUDAESCHER, B.; MELNIK, S. e RAGHAVAN, S. *Search Middleware and the Simple Digital Library Interoperability Protocol*. 2000. Disponível em: <<http://citeseer.ist.psu.edu/paepcke00search.html>>. Acesso em: 7 nov. 2012.

PATTERNS and PRATICES,<<http://msdn.microsoft.com/pt-br/practices>>. Acesso em Junho de 2012.

PKP. *What Is the Public Knowledge Project?* 2008. Disponível em: <<http://pkp.sfu.ca/about>>. Acesso em: 8 nov. 2012.

PRESSMAN, R. *Engenharia de Software*. São Paulo, Makron Books, 1995.

ROSETTO, M. *Uso do Protocolo Z39.50 para recuperação de informação em redes eletrônicas*. Ciência da Informação. Print ISSN 0100-1965. Ci. Inf. vol.26 n.2 Brasília May/Aug. 1997. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19651997000200004>. Acesso em: 9 nov. 2012.

TAMAE, R.Y. *Sisprodimes – Sistema de Processamento Distribuído de Imagens Médicas com XML – Qualificação de Mestrado*. PPGCC-UNIVEM – Fundação Eurípides de Marília, São Paulo. Março de 2004.

TSAI, W. T., GAO, J., XIAO Wei, & YINONG Chen. (2006). *Testability of software in service-oriented architecture*. Paper presented at the Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International, 2 163-170.

YAN, Zhao. (2006). *Enterprise service oriented architecture (ESOA) adoption reference*. Paper presented at the Services Computing, 2006. SCC '06. IEEE International Conference on, 512-512.

ZHU, H. (2005). *Software design methodology*. Oxford; Boston: Elsevier Butterworth-Heinemann.