

# **ARQUITETURA ORIENTADA A SERVIÇOS (SOA): UM ESTUDO DE CASO EM SISTEMAS DE INFORMAÇÃO GEOGRÁFICOS<sup>1</sup>**

*Service Oriented Architecture (SOA): a case study in Geographic Information Systems*

<sup>1</sup>PEDRO FELIPE ALVES DE OLIVEIRA

<sup>2</sup>PEDRO ALVES DE OLIVEIRA

<sup>3</sup>CLODOVEU AUGUSTO DAVIS JUNIOR

Pontifícia Universidade Católica de MG – PUC MG

<sup>1</sup>Programa de Pós-Graduação em Informática

<sup>2</sup>Programa de Pós-Graduação em Geografia – Tratamento da Informação Espacial  
pedrofao.puc@gmail.com; pedroalves@pucminas.br;

<sup>3</sup>Universidade Federal de Minas Gerais – UFMG

Departamento de Ciência da Computação

clodoveu@dcc.ufmg.br

## **RESUMO**

Em Sistemas de Informação Geográficos (SIG) o uso da Arquitetura Orientada a Serviços (SOA) é um terreno ainda pouco explorado, apresentando amplas possibilidades de evolução. SOA tem sido considerada uma evolução no desenvolvimento de aplicações Orientadas a Objetos (OO). As principais razões para isto se devem, em parte, ao reuso e encapsulamento oferecidos pelos serviços Web, mas principalmente à possibilidade de modelar o negócio de acordo com uma nova concepção, integrada aos objetivos organizacionais. Este artigo apresenta um breve estudo teórico na área, complementado através de um experimento sobre endereçamento de imóveis no município de Belo Horizonte (MG). Foram utilizados nos testes dados reais de endereços, tratados através de um processo que compreendeu três etapas: *parsing* (decodificação do endereço), *matching* (casamento com o banco de dados existente), *locating* (estabelecimento da posição).

---

<sup>1</sup> A versão prévia deste artigo foi apresentada no II Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação, em 10 de setembro de 2008, na cidade de Recife.

O grau de precisão permitiu avaliar a acuidade das saídas do algoritmo, em cada caso. Os resultados obtidos foram altamente satisfatórios e podem ser estendidos a outras situações similares, com as devidas adaptações.

**Palavras-chave:** Sistemas de Informação Geográficos; Arquitetura Orientada a Serviços; Serviços Web; Endereçamento.

## ABSTRACT

Service Oriented Architecture (SOA) is a trend for the development of Object-Oriented applications. This is due in part for the possibility of reusing and encapsulating provided by Web services, but is motivated by business modeling concepts which seek to integrate software modeling to corporate objectives. However, SOA can be extended to Geographic Information Systems (GIS), although the use of geographic services according to this architecture still remains a largely unexplored field, showing broad possibilities of evolution. This article presents a short theoretic study on SOA, with an experiment on address geocoding. In this case study, actual addressing data from the city of Belo Horizonte (MG) were employed. Test datasets were validated after being put through a three phase process: parsing, matching, and locating. The degree of precision obtained by the process allowed evaluating the accuracy of the algorithm in each case. Results obtained in the tests made showed a highly satisfactory precision. We believe that this experience could be extended to other similar situations, provided some necessary adaptations are made.

**Keywords:** Geographic Information Systems; Service-Oriented Architecture; Web Services; Addressing; Geocoding.

## 1. INTRODUÇÃO

Embora a automatização de processos dentro das organizações seja atualmente uma prática freqüente, é necessário que haja alinhamento estratégico entre as metas de negócio e os modelos tecnológicos que serão desenvolvidos, visando ao melhor uso dos recursos envolvidos. Desde a publicação, em 1995, do famoso *Chaos Report* do *Standish Group*<sup>2</sup>, quando apenas 31% dos investimentos destinados à área de Tecnologia da Informação (TI) resultavam em retorno econômico, pouca coisa parece ter mudado. De fato, as estatísticas de fracasso da indústria de software têm evidenciado muitos esforços custosos e inúteis.

A Arquitetura Orientada a Serviços (*Service Oriented Architecture - SOA*) surge como uma tentativa de agregar valor ao negócio, permitindo aprimorar os processos e reduzir o desperdício dos recursos de TI. Muitas razões têm levado o mercado a investir em SOA, mas os desafios na utilização de uma arquitetura tão inovadora e com poucos exemplos de uso corporativo (menos ainda no Brasil) são grandes. Academicamente, justifica-se plenamente o desenvolvimento de novos trabalhos

---

<sup>2</sup> <http://www.standishgroup.com/>

sobre esse tema, que possam auxiliar o mercado a ampliar o conhecimento a respeito dessa arquitetura e auferir suas vantagens, dentre as quais se podem destacar:

- a possibilidade de automatizar o gerenciamento de processos de negócio;
- a facilidade de integração com outros sistemas;
- o reuso de sistemas legados;
- a adaptação de aplicações para suportar mudanças de tecnologia.

O reconhecimento e tratamento de dados geográficos em aplicações convencionais são prejudicados pela ocorrência de abreviaturas e de falhas na grafia dos nomes de lugares, tais como ausência de acentuação ou omissão de partes dos nomes. Endereços postais são geralmente tratados como atributos do tipo texto, não passando por nenhum processo de validação antes de sua carga no banco. Com isso, o uso de dados geográficos para viabilizar a localização de objetos de interesse em um mapa torna-se uma tarefa complicada.

Este problema, dada sua multidisciplinaridade, exige mais de uma abordagem para sua solução. Em primeiro lugar, existe a necessidade de conhecimento específico do contexto, da realidade em que se insere. Faz-se necessário não somente modelar dados e processos, mas também adotar opções tecnológicas adequadas às possibilidades de solução que se apresentam, envolvendo algoritmos, linguagens de programação, bancos de dados e ambiente computacional.

Por outro lado, é notória a escassez de exemplos, na literatura, do uso de SOA em Sistemas de Informação Geográficos (SIG).

Considerando a pouca disponibilidade de estudos em SOA e algumas diferenças conceituais existentes, foi desenvolvido um estudo teórico e aplicado, utilizando os conceitos de Arquitetura Orientada a Serviços e SIG, cujo escopo é um processo de negócio para endereçamento flexível, composto pelos serviços de *parsing* (divisão de um endereço em seus componentes básicos), *matching* (casamento (compatibilização) dos elementos de endereçamento com objetos disponíveis em um banco de dados) e *locating* (estabelecimento da localização geográfica) (DAVIS JR., FONSECA, 2007).

## 2. SERVIÇOS WEB E SOA

Um *serviço Web* é um sistema de *software*, identificado através de um endereço universal (*Uniform Resource Identifier – URI*), no qual interfaces públicas e contratos são definidos e descritos em XML. Estas definições podem ser descobertas por outros sistemas de *software*. Os sistemas podem então interagir com o serviço Web em uma maneira prescrita pela sua definição, usando mensagens baseadas em XML e transportadas por protocolos da Internet (WSA, 2003).

No cotidiano, o conceito de serviço Web é muito confundido com o de SOA. Serviços Web apenas oferecem um recurso para implementar serviços, mas a arquitetura SOA envolve muitas outras características e recursos.

O consórcio da *World Wide Web* (W3C), organismo internacional encarregado de gerenciar todas as normas e padrões referentes à Web, define SOA de três formas (W3C, 2007):

- Uma arquitetura para aproximação de empresas e negócios – um caminho para entender e integrar a empresa no contexto da comunidade e como uma rede de serviços de negócios. SOA, no nível de negócio, é parte da arquitetura da empresa, mostrando como uma rede de serviços confere valor ao negócio.
- Um componente da arquitetura de sistemas de soluções – um caminho para compreender e integrar internamente e externamente processos de trabalho de empresas como elementos de uma rede de serviços de tecnologia. SOA, no nível de sistemas, é a arquitetura de solução mostrando como essa rede trabalha para agregar valor ao negócio.
- Um sistema de integração aproximado – um caminho para expor capacidades existentes para integrar aplicações e criar novas soluções compostas.

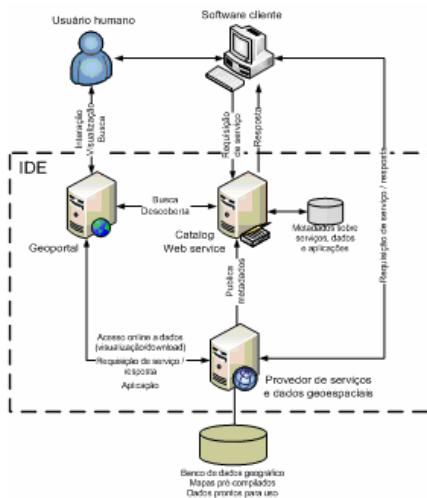
### 3. ARQUITETURAS DE SIG

As arquiteturas de SIG evoluíram a partir de um modelo monolítico (uma camada), até chegar às arquiteturas multicamadas. Esta evolução se deu paralelamente ao *Computer Aided Design - CAD* (desenho assistido por computador), passando pelos produtos *desktop mapping* (aplicações SIG em duas camadas, para uso em estações de trabalho), que prevaleceram no mercado durante um longo tempo, até as arquiteturas mais recentes, baseadas em modelagem Orientada a Objetos (OO) e em tecnologias Web. Dentre as diversas modificações, as últimas vieram acompanhadas com a evolução das redes de computadores e dos sistemas de informação com acesso pela internet. Outra direção nesse processo histórico tem sido a busca do intercâmbio *on-line* de dados, que passa pela interoperabilidade desses dados.

Resolvendo esta questão podem-se solucionar problemas ligados à atualização de dados, presentes nas arquiteturas anteriores. Estudos aprofundados sobre este tema (evolução dos SIG) podem ser obtidos em referências como Oliveira e Davis (2002).

Diversas iniciativas têm contribuído para conceber e implementar uma Infraestrutura de Dados Espaciais (IDE), que fosse abrangente e aproximasse diferentes provedores de dados geográficos. Para esse tipo de aplicação, a tecnologia de serviços Web oferece uma forma de acesso padrão e tecnologicamente neutra, e ainda possibilita a criação e operação de repositórios ou catálogos dos serviços disponíveis. O conceito de IDE é muito amplo, e se refere ao conjunto de tecnologias, dados, pessoas e políticas voltados para o compartilhamento de dados geográficos, com o objetivo de fomentar o desenvolvimento econômico.

A Figura 1 apresenta um modelo de arquitetura de SDI.  
 Figura 1 – Geoportais e IDE.



Fonte: Davis Jr *et al.* (2009) (no prelo).

Um dos componentes apresentados é o *geoportal*, uma aplicação na Web que oferece acesso interativo aos acervos de dados disponíveis nos provedores de serviços. Através do geoportal, usuários humanos podem buscar conjuntos de dados de seu interesse em um catálogo, obter informações sobre os serviços Web que os disponibilizam, e ainda visualizá-los diretamente. Os mesmos serviços podem ser acessados por software, sem a intervenção humana, de modo que os dados disponíveis na Web e acessados por meio de serviços possam ser usados diretamente a partir de sua fonte, dispensando a transferência pela rede e a geração de cópias locais dos dados.

O modelo de Infraestruturas de Dados Espaciais (IDE) é muito abrangente e normalmente tem como objetivo atender necessidades de dados de agências nacionais e com isso há grande volume de dados sobre um mesmo tema e em escalas nacionais ou regionais. Observa-se que o foco das IDEs existentes tende a ser predominantemente técnico, e muitas vezes não gera interesse do setor privado.

Davis Junior e Alves (2006) propõem a criação de uma IDE local (ou seja, de escopo urbano e escala municipal), com um nível de abrangência mais específico, com base na riqueza de aplicações geográficas de escopo urbano e na importância de seus dados. O uso de dados locais em serviços é uma demanda de grupos

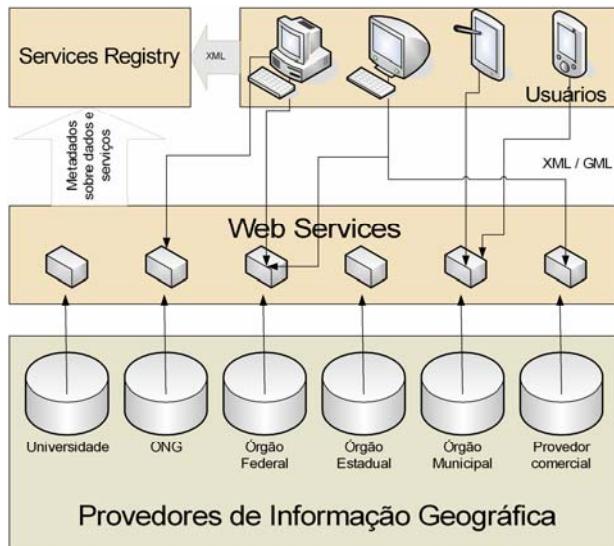
interessados em serviços Web de localização, planejamento de rotas, geocodificação<sup>3</sup> e localização de endereços.

A Figura 2 apresenta alguns dos possíveis provedores de informações geográficas para uma IDE local, que podem ser universidades, órgãos de governo, empresas comerciais, dentre outros. Os serviços podem ser localizados através de um catálogo, onde estão registrados metadados de dados e serviços.

A utilização de XML permite que até mesmo equipamentos com pouca capacidade de processamento e armazenamento tenham acesso aos serviços providos por essa SDI local orientada para serviços.

Um requisito para o bom funcionamento deste modelo é a confiabilidade dos dados gerados pelos provedores. É importante que estes gerem um conjunto amplo de dados que permita as mais diversas utilizações e que a atualização seja constante.

Figura 2 – Uma IDE local.



Fonte: Davis Jr., Alves (2006).

Parte deste trabalho consistiu na implementação de um serviço Web para prover geocodificação, dentro da arquitetura apresentada, e que pode ser utilizado em uma IDE local. Esse assunto é abordado na próxima seção.

<sup>3</sup> O termo geocodificação designa o estabelecimento de uma localização geográfica a partir de uma descrição textual, como um endereço postal

#### 4. ESTUDO DE CASO: UM SERVIÇO DE GEOCODIFICAÇÃO

Um serviço de geocodificação foi implementado e testado utilizando dados do Cadastro Técnico Municipal (CTM) da Prefeitura Municipal de Belo Horizonte (PBH), gerenciados e atualizados pela Empresa de Informática e Informação do Município de Belo Horizonte (PRODABEL). Foi adotada a metodologia apresentada em (DAVIS JR., FONSECA 2007).

A implementação do sistema para geocodificação de endereços foi feita em Java SE, e o banco de dados estruturado usando o gerenciador PostGIS. O processo de geocodificação foi implementado em três fases, cada qual transformada em um serviço Web. O primeiro (*Parsing*) analisa os dados de entrada, e separa os componentes do endereço, tais como tipo de logradouro, nome do logradouro, número do imóvel, etc. Na segunda fase (*Matching*) os dados são comparados com endereços presentes no banco de dados. Na última fase, chamada de *Locating*, o endereço é localizado de acordo com o resultado do *Matching*. O resultado do processo é um par de coordenadas, correspondente à melhor aproximação que foi possível obter a partir dos dados fornecidos, e um indicador numérico da qualidade do resultado. Esse indicador é denominado GCI (*Geocoding Certainty Indicator*). O GCI é obtido pelo produto de três indicadores, referentes respectivamente à qualidade estimada do resultado obtido em cada uma das três fases: *Parsing Certainty Indicator* (PCI), *Matching Certainty Indicator* (MCI) e *Locating Certainty Indicator* (LCI). Quanto mais próximo de 1 for o GCI, mais certeza se tem sobre o posicionamento do endereço. Usando o PCI, o MCI e o LCI, pode-se verificar em que fase houve mais problemas para a geocodificação.

Os serviços correspondentes a cada fase foram tornados visíveis para a rede com base em protocolos padronizados pelo W3C. Foi utilizada a abordagem *bottom-up* de desenvolvimento, em que primeiro são implementadas as classes Java dos serviços, e depois essas classes são descritas usando a linguagem WSDL (*Web Services Description Language*), conforme padronização do W3C. Criou-se uma aplicação composta e, através do código BPEL (*Business Process Execution Language*), os serviços passaram a poder ser invocados.

Cada um dos serviços pode estar localizado em máquinas diferentes, o que ajuda muito no desempenho global do sistema, tendo em vista que estes serviços consomem muitos recursos computacionais e utilizar máquinas diferentes é uma alternativa interessante. Além disso, pode-se também utilizar uma máquina como servidor dedicado de banco de dados, e essa máquina pode ser acessada pelos serviços de geocodificação.

A Figura 3 apresenta o formulário para entrada dos dados, considerando o modelo brasileiro de endereçamento. Como trabalho futuro a busca de endereços pode ser ampliada, podendo incluir até mesmo países que não seguem o padrão brasileiro. Para isso, é necessário adaptar o serviço de *Parsing* e expandir o conteúdo do banco de dados.

Figura 3 – Formulário da página de entrada de endereços.

## Buscar Localidade

Logradouro:

Tipo Logradouro:  RUA  N°:

Bairro:

Cidade:

Estado:

CEP:

### 4.1 Serviço de Parsing

O serviço de *Parsing* procura reconhecer os componentes do endereço, identificados na Tabela 1 como TPLOG (tipo de logradouro), NOLOG (nome do logradouro), NUIMV (número de porta do imóvel), NOBAI (nome do bairro), NOREF (referências ou descrições complementares), CID (cidade), EST (estado) e CEP (código de endereçamento postal). Os valores nas colunas indicam se o componente do endereço foi reconhecido (valor 1) ou não (zero), e as linhas representam todas as combinações possíveis de componentes. De acordo com essa combinação de elementos reconhecidos pelo processo, obtém-se um valor de PCI, que consta na última coluna da Tabela 1.

Por exemplo, a terceira linha da tabela corresponde à situação em que só foi possível reconhecer a referência ao estado. O valor de PCI para essa situação é de 0,10, pois o endereço buscado pode estar em qualquer ponto dentro do estado indicado.

Em contrapartida, a segunda linha trata do caso de que apenas o Código de Endereçamento Postal (CEP) foi reconhecido. Nesse caso, o valor do PCI é melhor (0,5), pois, dependendo do tamanho da cidade, o CEP pode indicar a localização do endereço com boa aproximação. Embora em pequenas cidades seja ainda utilizado um mesmo CEP para toda cidade, em grandes cidades cada rua, ou até mesmo grandes edifícios, tem um CEP exclusivo.

Os algoritmos usados para reconhecer cada componente do endereço não serão discutidos aqui, por limitação de espaço. Observe-se que nesta etapa não há verificação se as entradas estão corretas ou não. Isto só ocorre na fase seguinte.

Tabela 1 – Fragmento da tabela de PCI que contem os valores do *Parsing*.

	TPLOG	NOLOG	NUIMV	NOBAI	NOREF	CID	EST	CEP	PCI
0	0	0	0	0	0	0	0	0	0,00
1	0	0	0	0	0	0	0	1	0,50
2	0	0	0	0	0	0	1	0	0,10
3	0	0	0	0	0	0	1	1	0,50
4	0	0	0	0	0	1	0	0	0,50
5	0	0	0	0	0	1	0	1	0,50
6	0	0	0	0	0	1	1	0	0,50
7	0	0	0	0	0	1	1	1	0,55
8	0	0	0	0	1	0	0	0	0,10
...	...	...	...	...	...	...	...	...	...
248	1	1	1	1	1	0	0	0	0,50
249	1	1	1	1	1	0	0	1	1,00
250	1	1	1	1	1	0	1	0	0,50
251	1	1	1	1	1	0	1	1	1,00
252	1	1	1	1	1	1	0	0	0,95
253	1	1	1	1	1	1	0	1	1,00
254	1	1	1	1	1	1	1	0	1,00
255	1	1	1	1	1	1	1	1	1,00

#### 4.2 Serviço de *Matching*

Nesta etapa é feita a correlação do dado fornecido, já fracionado nos componentes descritos na seção anterior, com o conteúdo do banco de dados, usando algoritmos de reconhecimento de padrões e de casamento de cadeias de caracteres. É recebida uma tupla, contendo os seguintes valores: <CEP, Cidade, Estado, Bairro, Ponto\_ref, Logradouro, NumPorta>. A cada etapa de compatibilização, um desses elementos é comparado com dados existentes em busca de seu reconhecimento. Caso se tenha sucesso, a localização do dado fornecido torna-se mais precisa.

O processo tenta reconhecer o nome do município e estado, bairro, ponto de referência, logradouro e eixo de via, nessa ordem. A cada comparação de um atributo informado pela etapa de *parsing* com o conteúdo do banco de dados, faz-se a comparação permitindo uma similaridade não inferior a 80%. Caso existam múltiplos casamentos, ou seja, mais de um objeto do banco de dados corresponde a um nome dado, são usados métodos de desambiguação baseados nos demais

elementos de dados disponíveis. Em qualquer caso, a similaridade inferior a 100% e a ocorrência de ambigüidade causam reduções no valor do MCI.

Como exemplo, considere-se um endereço constituído de CEP, bairro, logradouro e número de porta, de uma determinada cidade.

O casamento entre o dado fornecido pelo usuário com o dado presente no banco de dados é iniciado pelo Código de Endereçamento Postal. Se o CEP tiver sido digitado e encontrado na base de dados, será possível ter uma aproximação da localização do endereço.

O próximo passo é encontrar um bairro cujo nome encontra-se da mesma forma como foi digitado pelo usuário. Se não for possível encontrar, utiliza-se um algoritmo de casamento aproximado de padrões, como o de Levenshtein (LEVENSHTEIN, 1966; DAVIS JR., SALLÉS, 2007). Esse algoritmo retorna a similaridade entre dois nomes com base no número de alterações (inserção de caracteres, exclusão, troca de posição) que têm que ser feitas em um para se obter o outro. O resultado é expresso como um percentual do número de caracteres do maior nome. No serviço implementado, foi exigida similaridade mínima de 80%. Quando é necessário usar esse recurso, existe uma possibilidade de ocorrência de falso positivo, ou seja, a associação do nome fornecido com outro diferente do pretendido. Quando existe esse risco, o indicador de qualidade no casamento de padrões (MCI) sofre uma penalidade.

A partir de então, inicia-se a busca pelo nome do logradouro. Primeiramente tenta-se encontrar o nome fornecido, da forma como foi digitado, no banco de dados. Se não for possível encontrar, procura-se um nome de logradouro dentro da similaridade de 20% e são aplicadas penalidades. Se, mesmo assim, seu nome não tenha sido encontrado, o logradouro é buscado usando-se o nome alternativo dele. Neste trabalho foi considerado como nome alternativo o nome anterior (que às vezes pode não existir anterior).

Ainda assim, se o logradouro não foi encontrado, ou se foi encontrado mais de um, o que é algo muito comum, é feita, se possível, uma comparação entre o nome de logradouro obtido através do CEP com o que foi digitado, usando o algoritmo de Levenshtein.

Caso o logradouro seja reconhecido em um dos passos anteriores, o algoritmo tenta encontrar no banco de dados um endereço individual naquele logradouro com o número de porta fornecido. Devido à grande probabilidade de erros no número de porta achou-se melhor desconsiderar os outros tipos de análise. Existem três possibilidades: A primeira delas é o número de porta estar correto, com isso o endereço é geocodificado no ponto correto. Outra possibilidade é a ausência do número de porta, com isso considera-se que o endereço individual pode ser qualquer ponto no logradouro. E a terceira hipótese, se esse número foi digitado e não foi encontrado, o algoritmo busca o número mais próximo na seqüência numérica do logradouro.

### **4.3 Serviço de *Locating***

Neste módulo ocorre o georreferenciamento do endereço, usando as indicações obtidas na etapa anterior. Supondo que foi encontrada uma edificação, o *locating* retornará sua coordenada geográfica e o indicador LCI terá valor máximo (1). Se aquela edificação não estivesse na tabela de pontos de referência, mas seu logradouro fosse encontrado, o cálculo do LCI seria feito considerando o quarteirão do qual o número de porta está mais próximo.

O valor do *locating* para um endereço individual é ((comprimento do trecho \* largura do logradouro \* fator de correção) / área município). Se for considerado o logradouro inteiro, basta substituir o comprimento do trecho pelo do logradouro.

Caso não tenha sido possível obter um logradouro ou um endereço, o cálculo do LCI equivale a ( $\text{área do bairro} / \text{área da cidade}$ ). A coordenada a ser utilizada é um ponto no centro do bairro. Se nem o bairro for localizado, a coordenada retornada será um ponto aleatório no interior dos limites do município, e o LCI será zero.

De posse dos três indicadores é possível calcular o GCI e obter as coordenadas do endereço. A Figura 4 apresenta a visualização do resultado, realizada superpondo a coordenada obtida ao mapa da cidade, disponível no Google Maps.

Figura 4 – Resultado da busca.



O GCI corresponde a PCI × MCI × LCI. Note que, mesmo se o usuário tiver preenchido todos os campos do formulário com dados incorretos e tiver obtido PCI igual a 1, isso não garante que o GCI terá um valor próximo de 1 (100%), pois o valor do MCI e do LCI poderão ser muito baixos e o resultado final, por conseguinte, será baixo também.

## 5. CONSIDERAÇÕES FINAIS

Através da utilização do protótipo do sistema pode-se perceber que o custo computacional dos serviços de *matching* e *locating* é muito grande, representando um desafio para a utilização em larga escala do sistema.

Devido ao grande número de acessos ao banco de dados, a realização de diversas consultas geográficas, a utilização de inúmeras estruturas de dados e várias estruturas de repetição, a busca de até mesmo um endereço individual pode ser demorada. Principalmente se for buscado um logradouro cujo nome se repete bastante (como por exemplo, rua A, em Belo Horizonte).

Com relação aos testes da precisão dos endereços, o indicador global GCI foi altamente preciso e refletiu a realidade. Os testes foram realizados a partir de uma lista de pontos de coleta de doadores de materiais recicláveis para a associação dos catadores de papel, papelão e material reciclável (ASMARE) de Belo Horizonte. Um caminhão da empresa era responsável por percorrer um trajeto que incluía vários pontos de coleta e a distância entre esses pontos era analisada, pois o gasto com combustível pelo caminhão às vezes era superior ao ganho com os materiais coletados. Freqüentemente a lista era preenchida com erros de digitação, ou erros causados por desconhecimento do endereço correto por parte dos doadores, ou ainda erros intencionais, como citar um logradouro como pertencente a um bairro vizinho mais valorizado.

Um dos exemplos testados foi o endereço “Rua Leci Gomes Barbosa, nº 227, Bairro Vale do Jatobá, Belo Horizonte, MG”. Este endereço teve o índice PCI penalizado em 0.05, por não constar o CEP. Na etapa de *matching* este endereço foi penalizado novamente. Pois há um erro no nome do logradouro (o correto era Leci Gomes Barbosa) e um erro no nome do bairro. O nome do bairro em que o logradouro está contido é “Jatobá”. Em Belo Horizonte só o bairro Jatobá contém este logradouro e por isso foi possível fazer o casamento correto do endereço, apesar dos erros. Como foi possível encontrar um endereço compatível com os dados de entrada e como os erros presentes no dado de entrada não foram tão significativos o indicador MCI teve valor alto (0.95). O número de porta fornecido não foi encontrado, com isso o serviço de *locating* posicionou o endereço em um ponto próximo ao número 227, como pode ser visto na Figura 4. O LCI teve valor aproximado de 0.99 e o Indicador de Certeza Global (GCI) por volta de 0.90.

Um sistema de geocodificação convencional dificilmente conseguiria localizar um endereço conforme o citado anteriormente. Outra vantagem do sistema proposto

foi a mensuração do grau de confiança de um endereço informado. Isto pode ser muito útil principalmente para empresas que oferecem serviços de entrega.

O uso de SOA tem se revelado promissor. Apesar de existirem obstáculos como a falta de treinamento e de orçamento nas empresas e barreiras culturais, sistemas criados neste modelo de arquitetura são feitos para durar. O trabalho mostrou que SOA promove o reuso de ativos, além de facilitar a implementação de novos serviços e promover a interoperabilidade. A cada novo serviço lançado seria preciso apenas adaptá-lo às interfaces oferecidas pela IDE e inclusive novos serviços poderiam ser construídos utilizando outros serviços já existentes. Como por exemplo, poderia ser criado no futuro um serviço de planejamento de rotas que utilize o serviço de geocodificação desenvolvido neste trabalho.

As conclusões obtidas neste trabalho mostram que é possível e justificável sua extensão a outras situações, dada a aplicabilidade das necessidades de endereçamento.

## AGRADECIMENTOS

Os autores gostariam de agradecer às diversas pessoas e entidades que contribuíram para o desenvolvimento deste estudo, dentre os quais se destacam a Prodabel pela cessão da base de dados, o CNPq e a FAPEMIG pelo financiamento da pesquisa. Também agradecem às universidades PUC Minas e UFMG pelo apoio institucional à pesquisa.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ACQUAVIVA, C. *SOA Service-Oriented Architecture*. Sun Microsystems, 2007.
- ALVES, A. *Arquitecturas Orientadas a Serviços – SOA*. Sun Microsystems – Portugal. JavaPT06, 2006.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software Architecture in Practice*. 2. ed. EUA: Addison-Wesley, 2003.
- BENEDETE Jr, A. C. *Roteiro para a definição de uma arquitetura SOA utilizando BPM*. 2007. 68f. Monografia (MBA) - Escola Politécnica da Universidade de São Paulo, MBA em Tecnologia da Informação.
- CASANAVE, C. *The Architecture of Services*. Workshop on eGovernment and the Web, 2007. W3C.
- CASANOVA, M. A. et al. *Bancos de Dados Geográficos*. 1. ed. Curitiba: Mundogeo, 2005.
- CHAMPION, M. et al. *Web Services Architecture*. W3C Working Draft, 2002. W3C.
- D'AURIA, P. *Desconstruindo BPM - Pontos de entrada para SOA*. 2o Fórum SOA – Tecnologia e Soluções. São Paulo. 2007.
- DAVIS Jr., C. A.; ALVES, L. L. *Infra-Estruturas de Dados Espaciais: Potencial para Uso Local*. Revista IP (P.65 - 80). 2006. Disponível em:<<http://www>.

- ip.pbh.gov.br/ANO8\_N1\_PDF/ANO8N1\_Clodoveu.pdf>. Acesso em 01 jun 2008.
- DAVIS Jr., C. A.; FONSECA, F. T. *Assessing the Certainty of Locations Produced by an Address Geocoding System*. *Geoinformatica* 11(1):103–129. 2007.
- DAVIS Jr., C.A.; FONSECA, F.; Câmara, G. *Understanding Global Change: The Role of Geographic Information Science in the Integration of People and Nature*. SAGE Handbook of GIS and Society. Nyerges, T., Couclelis, H. and McMaster, R.B., SAGE. 2009 (no prelo).
- DAVIS Jr., C. A.; OLIVEIRA, P. A. *SIG Interoperável e Distribuído para Administrações Municipais de Grande Porte*. Revista IP (P.121 – 141). 2002.
- DAVIS Jr., C.A.; SALLES, E. *Approximate string matching for geographic names and personal names*. Proceedings of the IX Brazilian Symposium on GeoInformatics (GeoInfo 2007), Campos do Jordão (SP), Brazil. P.49 - 60. 2007.
- DIGITALASSETS. *SOA e Reúso na Prática*. Disponível em: <<http://www.digitalassets.com.br/soa-modelo-denegocios>>. Acesso em 29 mai 2009.
- GARTNER GROUP. *The Gartner Programming Language Survey*. Gartner Advisory. 1997.
- HAAS, H.; BROWN, A. *Web Services Glossary*. W3C Working Group, 2004. Disponível em: <<http://www.w3c.org/TR/2004/NOTE-ws-gloss-20040211/>>. Acesso em 12 abr 2008.
- LEVENSHTEIN, V.I. *Binary Codes Capable of Correcting Deletions, Insertions and Reversals*. Sovietic Physics Doklady. Vol 10(8) (P.707 – 710). 1966.
- MENDES, M. A. *Arquiteturas Orientadas por Serviço*. Archware. 2007.
- OLIVEIRA, P. A. *Prospecção de Ferramentas SIG – Relatório Final*. Prodabel: 2001.
- SABBAH, D. *The Future of Software Delivery*. White Paper. 2007. IBM
- SAMPAIO, C. *SOA e Web Services em Java*. 1. ed. Rio de Janeiro: Brasport, 2006.
- ZHAN, B. *The NCGIA Core Curriculum in GIScience*. Unit 064 – Representing Networks, 1998. Disponível em: <<http://www.ncgia.ucsb.edu/giscu/units/u064/>>. Acesso em 02 mai 2008.

(Recebido em abril de 2009. Aceito em dezembro de 2010)