

# Escalabilidade em Composição Dinâmica de Serviços baseada em QoS: uma Revisão Sistemática

Vivek Nallur<sup>1</sup>, Letícia Duboc<sup>2</sup>, Rami Bahsoon<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação  
Birmingham University, Birmingham, UK

<sup>2</sup>Instituto de Matemática e Estatística (IME)  
Universidade Estadual do Rio de Janeiro (UERJ), Rio de Janeiro, Brasil

{nallurv@cs.bham.ac.uk, leticia@ime.uerj.br, r.bahsoon@cs.bham.ac.uk}

**Abstract.** *The service oriented architecture paradigm has caused a revolution in software development, allowing applications to be constructed from independent services on the Internet. For such, service composition infrastructures select and combine services dynamically, ensuring that the resulting applications offer the desired functionality and quality of service. This selection and the calculation of the combined quality of services are computationally expensive, making scalability an important issue for these infrastructures. This paper performs a systematic literature review to identify how scalability is being evaluated in this domain and to discuss the implications of this scenario.*

**Resumo.** *O paradigma da arquitetura orientada a serviços causou uma revolução no desenvolvimento de software, permitindo que aplicações sejam construídas a partir de serviços independentes disponibilizados na Internet. Para tal, infraestruturas de composição de serviços selecionam e combinam serviços dinamicamente, garantindo que a aplicação resultante ofereça a funcionalidade e a qualidade de serviço desejadas. Esta seleção e o cálculo das qualidades de serviço combinados são computacionalmente caros, fazendo da escalabilidade uma importante preocupação destas infraestruturas. O presente artigo faz uma revisão sistemática de literatura para identificar como escalabilidade está sendo avaliada neste domínio e para discutir as implicações de tal cenário.*

## 1. Introdução

O paradigma da arquitetura orientada a serviços permite que aplicações sejam compostas dinamicamente utilizando serviços independentes. Tal composição é possível graças a infraestruturas, chamadas de aplicações de composição de serviços baseadas em QoS, que permitem a serviços serem publicados, encontrados e combinados em tempo de execução [Ko et al. 2008][Zeng et al. 2004][Gao et al. 2005]. Elas contam com um registro, onde provedores de serviços se inscrevem informando a funcionalidade e a qualidade de serviço (QoS) oferecidas. Este último é normalmente composto por atributos de qualidades de software como confiabilidade, disponibilidade, segurança, desempenho e outros requisitos não funcionais que devem ser providos por uma aplicação de software em tempo real. Incapacidade de satisfazer tais requisitos

pode levar ao abandono do uso do software e até mesmo a perdas contratuais significantes.

Desenvolvedores de aplicações podem utilizar-se de infraestruturas para selecionar serviços para suas partes constituintes. Para tanto, eles devem informar a funcionalidade desejada – normalmente na forma de um workflow de tarefas – assim como a qualidade de serviço requerida. A busca pelos serviços é feita automaticamente, procurando selecionar aqueles que melhor atendam aos critérios de funcionalidade e qualidade de serviço. A qualidade de serviço oferecida pela aplicação, por sua vez, será a combinação dos atributos de QoS de seus serviços constituintes. Por isso, é importante que cada um deles mantenha estes atributos dentro dos valores acordados. Quando isto não ocorre, o serviço pode ser substituído dinamicamente.

A busca por serviços coma combinação ótima de atributos de QoS é um problema conhecido como NP-completo [Yu et al. 2007]; de forma que tais infraestruturas normalmente oferecem uma solução “suficientemente boa”, que represente um equilíbrio entre o tempo de seleção dos serviços e a QoS da aplicação resultante. Este compromisso torna-se ainda mais complicado à medida que o espaço de busca aumenta. Por este motivo, proponentes de soluções neste domínio se preocupam em caracterizar-lhes a escalabilidade.

Em tal cenário, é natural perguntar-se se estes autores estão de comum acordo com relação às métricas que medem escalabilidade e às características do domínio de aplicação e do design dessas infraestruturas cuja variação afeta estes valores assumidos por métricas. Dito acordo não somente permitiria que proponentes de soluções soubessem a priori quais características tratar no projeto de suas infraestruturas, como também tornaria viável a comparação entre as propostas. Com o intuito de investigar essas questões, nós realizamos uma revisão sistemática de literatura (SLR) [Kitchenham and Charters 2007] em escalabilidade em composição dinâmica de serviços.

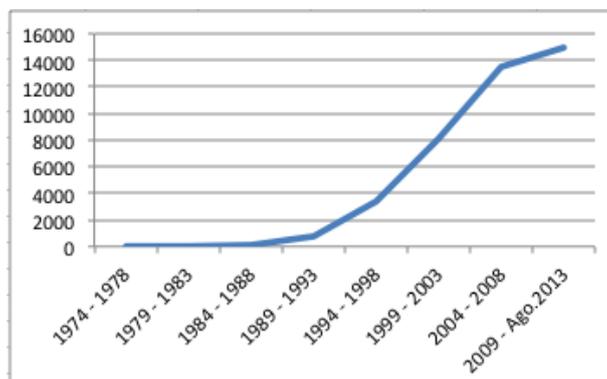
Outras SLR foram feitas no domínio de orientação a serviços. Gu e Lago (2009), por exemplo, discutem os desafios de se construir sistemas orientados a serviços; curiosamente escalabilidade não foi listada como um desses desafios. Leite et al. (2013) fazem uma SLR sobre adaptação em coreografia de serviços, verificando que somente um dos trabalhos revisados [Dar et al. 2011] considera escalabilidade. Outro trabalho relacionado é o de Issarni et al. (2011), que discute o estado da arte em *middleware* orientados a serviços, identificando escalabilidade como um dos principais desafios destes *middlewares*. No entanto, nenhum desses trabalhos tem o propósito de entender como escalabilidade está sendo tratado no domínio de composição dinâmica de serviços.

## **2. Pressupostos Teóricos**

### **2.1. Escalabilidade**

Escalabilidade é um assunto que tem recebido crescente atenção no decorrer dos anos, como mostra a Figura 1. O gráfico ilustra o número de publicações com as palavras *scalability* ou *scalable* em seu título desde os anos 70, utilizando o buscador Google Scholar. Somente no domínio de arquitetura baseada em serviços(SOA), encontramos

908 artigos que tratam diretamente de escalabilidade; isto considerando uma única biblioteca.<sup>1</sup>



**Figura 1. Crescente interesse em escalabilidade.**

Apesar do crescente interesse e atualidade do tema, ainda são comuns alguns equívocos em relação a esta qualidade:

(i) *Escalabilidade não diz respeito única e exclusivamente ao desempenho de um sistema.*

Desempenho é somente um de seus indicadores. Considere-se, por exemplo, o *BART* [Winter et al. 2001], um sistema de controle automatizado de trens que busca transportar um número maior de passageiros através da diminuição do espaçamento entre os mesmos. Neste sistema, a preocupação em relação à escalabilidade é a *distância de separação segura entre trens* (uma medida de segurança) tendo em conta o aumento do número destas locomotivas na rede ferroviária.

(ii) *Escalabilidade não significa necessariamente utilização linear de recursos, à medida que a carga no sistema aumenta.*

Tome-se, por exemplo, o domínio de verificação de modelos, onde o consumo de recursos é normalmente exponencial e um algoritmo com tempo de execução polinomial é considerado escalável.

(iii) *A computação em nuvem não oferece escalabilidade infinita.*

Com o surgimento deste paradigma e sua oferta de recursos computacionais sob demanda, pode-se ter a ilusão de que o problema da escalabilidade já está resolvido. No entanto, os recursos neste modelo de negócio são pagos de acordo com a sua utilização: uma aplicação que não faz uso responsável dos mesmos pode se tornar economicamente inviável. Adicionalmente, a aplicação pode enfrentar outros problemas se não for cuidadosamente projetada para a escalabilidade, como falhas de sincronização e *deadlocks*. É imprescindível, para um bom projeto, que se entendam as características cuja variação pode afetar a escalabilidade de um sistema, assim como os indicadores dessa escalabilidade.

Buscando resolver tais equívocos, Duboc et al. propuseram a seguinte definição de escalabilidade: "*É uma qualidade de software caracterizada pela habilidade de um*

<sup>1</sup>String de busca: ((scalab\*) AND Service-oriented architecture) no banco de dados IEEEExplore.

*sistema em satisfazer suas metas de qualidade quando as características do domínio de aplicação e do design do sistema variam dentro de intervalos operacionais."*

É interessante observar que a definição anterior engloba metas de qualidade distintas, como desempenho, confiabilidade e segurança. Assim, a escalabilidade pode ser medida em função de diferentes métricas e sofrer influência de várias características. Este fato faz com que uma análise de qualquer atributo de qualidade que considere variações em características do domínio de aplicação e do design do sistema, seja, em realidade, uma análise de escalabilidade.

## **2.2. Composição Dinâmica de Serviços**

Composição dinâmica de serviços permite que aplicações sejam construídas a partir de serviços disponibilizados na Internet. Para tal, são utilizadas infraestruturas que busquem automaticamente serviços com as funcionalidades desejadas, combinando-os para criar aplicações e monitorando sua execução. Um subgrupo dessas infraestruturas está preocupado em selecionar um conjunto de serviços que otimize a QoS da aplicação resultante – estas são aplicações de composição de serviços baseadas em QoS. É importante entender alguns conceitos comuns a estas infraestruturas:<sup>2</sup>

- *Serviço Abstrato*: é a especificação funcional de uma certa tarefa.
- *Serviço Candidato*: é a implementação de um serviço abstrato. Cada serviço candidato tem uma qualidade de serviço, que é negociada através de um *Service Level Agreement (SLA)*.
- *Workflow*: define uma estrutura de serviços abstratos para formar uma aplicação.

O cálculo da qualidade de serviço da aplicação varia de acordo com os atributos de QoS. Por exemplo, o tempo de execução de uma aplicação poderá ser a soma dos tempos de execução de suas partes constituintes. Para o cálculo de outros atributos, como confiabilidade, são utilizados mecanismos diferentes que consideram a estrutura da aplicação [Cardoso et al. 2004].

Normalmente, os valores dos atributos de QoS são estabelecidos através de acordos de níveis de serviços (SLA). Algumas infraestruturas renegociam contratos automaticamente quando SLAs expiram, e monitoram a execução dos serviços, substituindo-os em tempo de execução se eles deixam de prover a qualidade prometida.

Como a busca de serviços que otimizem a qualidade de serviço da aplicação é um processo computacionalmente caro, é comum haver preocupação com a escalabilidade dessas infraestruturas. A seguir mostramos dois exemplos clássicos de análises de escalabilidade neste domínio:

- Zeng et al. [Zeng et al. 2004], um dos primeiros artigos de composição de serviços Web, representam o *workflow* da aplicação como um diagrama de estados e comparam o uso de otimização local de QoS versus planejamento global utilizando programação inteira (IP). Em sua análise de escalabilidade, os autores variaram o *tamanho do workflow* e o *número de serviços candidatos*, e medem o *tempo de composição de serviços*.

---

<sup>2</sup>Nomes podem variar de solução em solução.

- Ko et al.(2008) propõem um algoritmo híbrido para encontrar um plano de composição que não viole nenhuma restrição individual de QoS. Para avaliar a escalabilidade da solução, os autores calculam o *tempo de resposta* do algoritmo, também variando o *tamanho do workflow* e o *número de serviços candidatos*.

Contudo, será que a avaliação do tempo de resposta – em função do tamanho do *workflow* e do número de serviços abstratos - é o suficiente para caracterizar a escalabilidade neste domínio? Não haveria outras qualidades de software, além de desempenho, que poderiam ser afetadas? A variação de quais características do domínio da aplicação e do design do sistema impactariam nestas qualidades?

Com o intuito de entender como autores no domínio de composição dinâmica de serviços baseadas em QoS estão avaliando a escalabilidade de suas soluções, fizemos uma revisão sistemática da literatura.

### 2.3. Revisão Sistemática da Literatura (SLR)

A revisão sistemática da literatura é uma forma de identificar, avaliar e interpretar toda pesquisa disponível concernente a uma questão, área ou fenômeno de interesse em particular. Kitchenham and Charters (2007) adaptou diretrizes da área médica para refletir os problemas específicos da engenharia de software. Este modelo de revisão pode ser usado para: (i) resumir evidência sobre algum tratamento ou tecnologia; (ii) identificar lacunas na pesquisa atual e apontar áreas futuras de pesquisa; (iii) estabelecer uma base para posicionar uma nova pesquisa; ou (vi) mostrar qual evidência empírica contradiz ou dá suporte a uma hipótese.

De acordo com as diretrizes de Kitchenham and Charters (2007), o processo de revisão sistemática consiste de três fases: planejamento, execução e disseminação. A Tabela 1 lista as atividades de cada uma destas fases.

**Tabela 1. Fases de uma SLR**

Planejamento	1. Estabelecer a necessidade da revisão 2. Definir perguntas de pesquisa 3. Desenvolver um protocolo de revisão
Execução	4. Definir a estratégia de busca 5. Selecionar as fontes primárias 6. Avaliar a qualidade das fontes 7. Extrair e monitorar os dados 8. Sintetizar os dados
Disseminação	9. Especificar os mecanismos de disseminação 10. Formatar o relatório

Na fase de planejamento, é feito um protocolo da revisão, que é um plano de ação criado antes da execução da mesma. Esse protocolo dá margem para que o pesquisador identifique falhas no procedimento e avalie se o mesmo é capaz de responder às perguntas de pesquisa. Os principais componentes do protocolo são: (i) as

perguntas de pesquisa a serem respondidas pelo estudo; (ii) a especificação das fontes de dados e da estratégia de busca de literatura dessas fontes; (iii) os critérios de seleção dos resultados relevantes; (iv) a metodologia de extração dos dados; e (v) a estratégia de síntese dos dados. A importância do protocolo de revisão está em definir passos precisos a serem seguidos pelos pesquisadores, de maneira que a revisão possa ser estudada, criticada e/ou replicada.

### **3. SLR em Composição Dinâmica de Serviços**

O primeiro passo é estabelecer a necessidade da revisão: estamos interessados em identificar se, e como, autores no domínio de composição dinâmica de serviços baseados em QoS avaliam a escalabilidade de suas soluções. Que técnicas estão sendo utilizadas? Que características do domínio e do sistema estão sendo consideradas? Que métricas estão sendo adotadas?

Tal análise é bastante relevante para pesquisadores que trabalham neste domínio e para *stakeholders* em busca de uma aplicação de composição de serviços: será que existe uma uniformidade que permita a comparação de soluções alternativas? Com quais características e métricas os proponentes de tais soluções deveriam preocupar-se ao elaborá-las?

É importante enfatizar que se encontra fora do escopo da pesquisa avaliar a escalabilidade de cada solução encontrada.

#### **3.1. Protocolo**

Nesta seção são apresentados os principais elementos do protocolo utilizado.

##### *3.1.1. Perguntas de Pesquisa*

A definição das perguntas de pesquisa é a parte mais importante de uma revisão sistemática. A partir delas define-se o processo de busca, quais dados devem ser extraídos e como deve ser feita a síntese dos dados [Kitchenham 2007]. Nossa SLR procurou responder às seguintes perguntas:

**RQ1.** De que forma autores na área de composição dinâmica de serviços baseados em QoS estão avaliando a escalabilidade de suas soluções?

**RQ2.** Quais características do domínio de aplicação e do design do sistema estão sendo consideradas para caracterizar a escalabilidade de soluções nesta área, e quais intervalos de valores destas características estão sendo assumidos?

**RQ3.** Que métricas de software estão sendo adotadas para medir a escalabilidade destas soluções?

##### *3.1.2. Estratégia de Busca*

A estratégia de busca é normalmente iterativa. Ela pode incluir buscas automáticas ou manuais em bibliotecas científicas, anais de conferências e revistas, consulta a especialista e outras técnicas para expandir a abrangência dos resultados.

Em geral, na engenharia de software, bases de dados eletrônicas são consideradas suficientes [Kitchenham et al. 2004][Gu and Lago 2009]. A nossa SLR considerou os seguintes engenhos de busca:

1. *IEEE Explore*
2. *ACM Digital Library*
3. *Science Direct*
4. *Engineering Village*
5. *ISI Web of Knowledge*
6. *Google Scholar*
7. *CiteSeer*

Adicionalmente, foi empregada a técnica de *snowballing* para expandir a abrangência dos resultados, que busca entre as referências dos artigos selecionados, outros trabalhos de interesse [Budgen 2008].

As fontes de dados foram pesquisadas utilizando-se uma combinação das palavras-chave abaixo. As palavras de busca foram adaptadas para cada engenho.

1. *Quality of Service* – variantes: *QoS*, *QoS-aware*, *QoS-enabled*;
2. *Web Service Composition* – variantes: *WSC*, *Service Composition*, *Service-based*, *Service-Oriented*, *Service-based Architecture*, *Service-oriented Architecture*, *Service-selection*;
3. *Dynamic* – variantes: *Adaptive*, *Adaptation*, *Self-adaptive*, *Self-optimizing*, *Self-healing*, *Self-managing*.

### 3.1.3. Seleção dos Estudos

A seleção procura identificar quais estudos primários proveem evidência para responder às perguntas de pesquisa, e ao mesmo tempo tenta eliminar os estudos não relacionados. Para tal, foram adotados os seguintes critérios:

#### **Critério de Inclusão:**

Artigos de conferências, de revistas, de *workshops* e relatórios técnicos.

#### **Critério de Exclusão:**

**E1:** Artigos que não lidam com a seleção dinâmica de serviços;

**E2:** Artigos que pertencem a um determinado domínio, pois estamos interessados em entender os atributos e métricas que são relevantes para o problema geral de composição dinâmica de serviços;

**E3:** Artigos que tinham extensões em revistas;

**E4:** Artigos que datam de antes do ano 2000, quando o padrão de Serviços Web foi publicado;

**E5:** Artigos que não foram publicados em inglês;

**E6:** Referências duplicadas;

**E7:** Artigos que não puderam ser obtidos de bancos de dados eletrônicos ou dos autores;

**E8:** Artigos lidando exclusivamente com melhorias técnicas à infraestrutura base, como *SOAP*, *WSDL*, *BPEL*, *DAML-S*, *OWL-S*.

**E9:** Artigos que não mencionam um mecanismo de avaliação de atributos de QoS;

**E10:** Artigos que não mencionam um *workflow* ou serviços abstratos em um *workflow*;

**E11:** Artigos que não mencionam serviços candidatos para serviços abstratos;

### **Justificativa do Critério de Exclusão:**

Os critérios de exclusão E1 a E7 são auto explicativos, os restantes merecem uma rápida justificativa:

E8: Artigos que discutem melhorias em tecnologias como *SOAP*(2007) ou *WSDL* (2007) estão normalmente relacionados com a problemática da eficiência da descrição de serviços e com a banda utilizada para transferir ontologias de serviços. Enquanto avanços nessas tecnologias podem afetar a eficiência de transferência dessas descrições, eles não resolvem o problema de selecionar serviços baseados em seus atributos de QoS. Melhorias em linguagens de descrição, como *DAML-S*[Paolucci and Sycara 2003] e *OWL-S* (2004), podem ajudar a encontrar serviços. Destas, só foram excluídas as que não mencionavam nada em relação à correspondência com atributos de QoS.

E9: Para determinar se um conjunto de serviços candidatos atende aos requisitos de uma aplicação, e entre estes qual é a melhor alternativa, é necessário algum mecanismo para avaliar os atributos de QoS. Utilizar somente custo não é o suficiente para o problema em que estamos interessados. Portanto, artigos que não mencionavam algum mecanismo QoS qualidade de serviço anunciada foram excluídos.

E10: Serviços baseados em aplicações são compostos em um *workflow*. Um mesmo conjunto de serviços combinados de uma forma distinta pode apresentar uma qualidade de serviço total diferente. Estamos preocupados com a qualidade de serviço apresentada por toda a aplicação, não pelos serviços individuais. Um mecanismo que não leva em consideração a estrutura do *workflow* não resolve o problema que nos interessa. Portanto, artigos que não mencionam nenhuma noção de *workflow*, ou estrutura de tarefas em um *workflow*, deixaram de ser considerados.

E11: O problema da composição dinâmica envolve escolher um serviço dentre um conjunto de serviços candidatos. A dificuldade está no fato de existirem muitos serviços candidatos, e que uma escolha má poderia ultrapassar o orçamento da aplicação ou resultar em uma qualidade de serviço inadequada. Portanto, artigos que não lidam com a problemática de serviços candidatos múltiplos para um serviço abstrato foram excluídos do estudo.

### **3.1.4. Extração dos Dados**

Em nosso estudo, os seguintes dados foram extraídos, juntamente com o nome dos artigos, de seus autores, e do local e ano de publicação:

- O artigo alega escalabilidade?
- A alegação é baseada em uma avaliação?
- Qual mecanismo foi utilizado para avaliar a escalabilidade da solução?

- Que métricas foram consideradas para avaliar a escalabilidade da solução?
- Que características do domínio de aplicação e do design do sistema foram consideradas para avaliar a escalabilidade, e que intervalo de valores foi utilizado?
- Que técnica foi aplicada na composição de serviços?
- Que atributos de QoS foram considerados para a composição de serviços e quais eram suas restrições, quando especificadas?
- Que função de otimização/utilidade foi empregada para encontrar um conjunto de serviços adequado?

### 3.2. Processo

O processo de seleção, extração e síntese dos dados seguiu o fluxograma da Figura 2.

A primeira busca eletrônica retornou 13.532 artigos. Somente os 50 primeiros de cada engenho de busca foram considerados, a fim de obter um número gerenciável de artigos. Depois de aplicados os critérios de inclusão, de exclusão e a técnica de *snowballing*, os 32 artigos na Tabela 2 foram selecionados para análise. Os critérios E8-E11 só foram aplicados após uma leitura rápida de todos os artigos que não haviam sido excluídos pelos critérios E1-E7.

Tanto a seleção como a extração dos dados foram realizadas por dois pesquisadores. Dividiram-se os artigos em dois grupos, sendo que um subconjunto foi analisado por ambos os pesquisadores, para verificar consistência. Em seguida, os dados foram sintetizados por um único pesquisador.

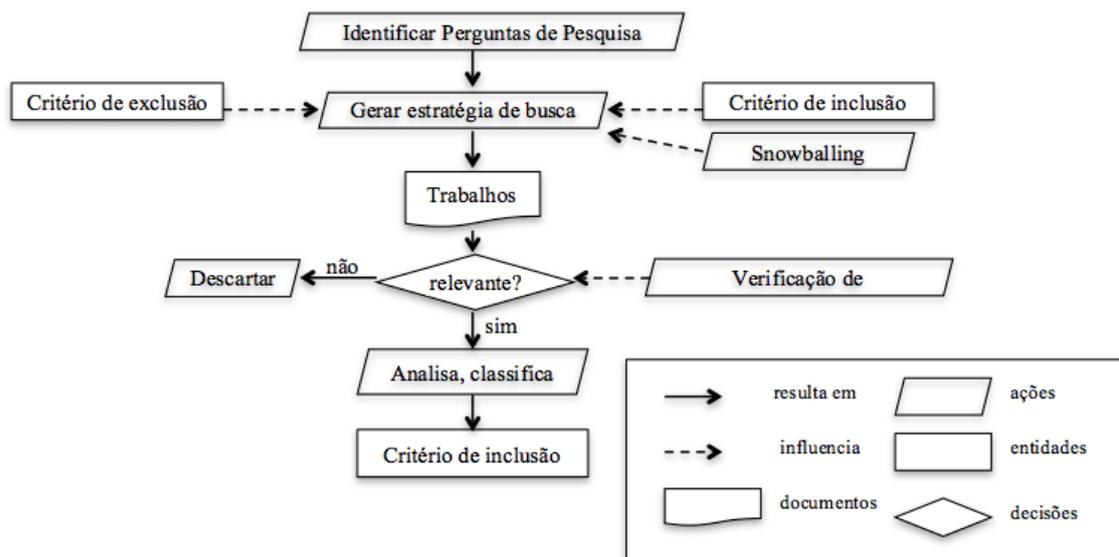


Figura 2. Processo da SLR

### 3.3. Resultado

Os dados relacionados às perguntas de pesquisa são apresentados abaixo. Eles serão discutidos na seção seguinte.

**[RQ1]** Como autores na área de composição dinâmica de serviços baseados em QoS estão avaliando a escalabilidade de suas soluções?

Como discutido na Seção 2.1, a análise ordinária de um atributo de qualidade considerando-se a variação de valores em características no domínio de aplicação e no design do sistema, é na realidade, uma análise de escalabilidade. Portanto, nesta revisão sistemática, foi observado que, dos 32 artigos avaliados, 27 apresentavam algum tipo de alegação ou análise de escalabilidade.

Alguns alegavam escalabilidade explicitamente, como no artigo [P2], que tem uma seção intitulada *QoS Scalability*. Nessa seção, os autores analisam o tempo de execução, dada a variação nas quantidades de serviços concretos e de serviços abstratos. Outros trabalhos apresentam análises sem mencionar de maneira explícita o termo “escalabilidade”. Este é o caso de [P10], que testa a percentagem de soluções ótimas, dado um crescente número de serviços a serem combinados. Dos cinco artigos que não avaliam escalabilidade, quatro ([P22][P24][P25][P28]) apresentam exemplos numéricos sem nenhum tipo de variação. Por exemplo, o artigo [P24] mostra os valores da utilidade de oito serviços abstratos e 40 serviços candidatos. Somente um dos artigos revisados, [P28], não apresenta nenhum exemplo numérico.

Nossa SLR também mostra que todos os artigos que apresentam uma análise de escalabilidade (com variação) ou simplesmente uma alegação em relação a algum atributo de qualidade (sem variação), baseiam suas alegações em alguma forma de avaliação. Na maioria das vezes, em 26 artigos, são utilizadas simulações. Os outros sete utilizam testes ou exemplos numéricos.

**[RQ2]** Que características do domínio de aplicação e do design do sistema estão sendo consideradas para caracterizar a escalabilidade de soluções nesta área, e quais intervalos de valores destas características estão sendo assumidos?

Nossa SLR mostrou que os artigos consideram a variação de uma ampla gama de características, ambos do domínio de aplicação e do design do sistema. Quase todos os artigos consideram o *tamanho do workflow* e o *número de serviços candidatos* em suas análises, respectivamente 29 e 28 artigos. Outras 19 características do domínio de aplicação foram mencionadas, como o número de *atributos de QoS* considerados, o número de *requisições por dia*, a quantidade de *dados transmitida entre serviços* e o número de *ataques a serviços*, entre outras. Com exceção do número de atributos de QoS, que foi mencionado por três artigos, todas as outras características foram listadas por, no máximo, um artigo.

Os intervalos de valores considerados variam ainda mais. Pode-se observar, por exemplo, os intervalos de valores do tamanho do *workflow*; este varia de 5 a 10.000, considerando todos os artigos. No entanto, cada artigo avaliado adota seu próprio intervalo. Por exemplo, o artigo [P2] considera o intervalo [5 – 16], enquanto que o artigo [P15] testa um *workflow* de tamanho [10-100]. Somente um trabalho, o [P4], avalia um *workflow* com mais de 100 serviços. Analogamente, foram considerados de 1 a 2.000 serviços candidatos por serviço abstrato no *workflow*. Cada artigo adotou o seu próprio intervalo.

Em relação ao design do sistema, 16 características foram mencionadas. Algumas variam em uma escala numérica. O artigo [P1], por exemplo, varia o número de *ants* em um algoritmo de *ant colony optimization (ACO)* [P13], entre 3, 6 e 7. Já o artigo [P3] varia o valor do *max non-improving generations (MNIG)*, variável de design que determina uma condição de término, considerando os valores 1.000 a 5.000. Outras características de design representam limiares, como o limite para um modelo estatístico

de volatilidade, em [P16]. Finalmente, outras características representam escolhas de design, como a seleção de algoritmos de construção de árvores em [P12] e o modo de configuração (dinâmico local, dinâmico global e estático) em [P6].

**Tabela 2. Artigos selecionados para análise**

<p>[P1] Wang X.; Jing Z.; Yang H. 2011. Service Selection Constraint Model and Optimization Algorithm for Web Service Composition. Information Technology Journal</p>	<p>[P2] Calinescu R.; Grunske L.; Kwiatkowska M.; Mirandola R.; Tamburrelli T.; 2011. Dynamic QoS Management and Optimization in Service-Based Systems. IEEE Trans. Softw. Eng.</p>	<p>[P3] Li S.; Chen M. 2010. An adaptive-GA based QoS driven service selection for Web services composition, In Intl. Conf. on Computer Application and System Modeling (ICCSM)</p>
<p>[P4] Ardagna D.; Mirandola R. 2010. Per-flow optimal service selection for Web services based processes. J. Syst. Softw.</p>	<p>[P5] Liu Z.; Xiao R. 2010. Towards Service Selective Optimizing Based on Key Business Process Performance. In 2nd Intl. Conf. on e-Business and Information System Security (EBISS)</p>	<p>[P6] Yang H.; Li Z. 2010. Improving QoS of Web Service Composition by Dynamic Configuration. Information Technology Journal</p>
<p>[P7] Li J.; Zhao Y.; Liu M.; Sun Hd; MaD. 2010. An adaptive heuristic approach for distributed QoS-based service composition, In IEEE Simp. on Computers and Communications (ISCC)</p>	<p>[P8] Sheu R.; Lo W.; Lin C.; and Yuan S. 2010. Design and Implementation of a Relaxable Web Service Composition System. In Intl. Conf. on Cyber-Enabled Distributed Computing and Knowledge Discovery (CYBERC '10)</p>	<p>[P9] Alrifai M.; Risse T. 2009. Combining global optimization with local selection for efficient QoS-aware service composition. In 18th Intl. Conf. on World wide web (WWW '09)</p>
<p>[P10] Chen Z.; Wang H.; Pan P. 2009. An Approach to Optimal Web Service Composition Based on QoS and User Preferences. In Intl. Joint Conf. on Artificial Intelligence (JCAI '09)</p>	<p>[P11] Guoping Z.; Huijuan Z.; Zhibin W. 2009. A QoS-Based Web Services Selection Method for Dynamic Web Service Composition. In First Intl. Workshop on Education Technology and Computer Science (ETCS '09)</p>	<p>[P12] Hristoskova, A.; Volckaert, B.; De Turck, F. 2009. Dynamic Composition of Semantically Annotated Web Services through QoS-Aware HTN Planning Algorithms, In 4th Intl. Conf. on Internet and Web Applications and Services, 2009. ICIW '09</p>
<p>[P13] Yang, L.; Dai, Y.; Zang B.. 2009. Performance Prediction Based EX-QoS Driven Approach for Adaptive Service Composition, J. Inf. Sci. Eng.</p>	<p>[P14] Chen Z; Yao Q. 2008. A Framework for QoS-aware Web Service Composition in Pervasive Computing Environments, In 3rd Intl. Conf. on Pervasive Computing and Applications (ICPCA'08)</p>	<p>[P15] Ko, J. M.; Kim, C. O. &amp; Kwon, I.-H. 2008. Quality-of-service oriented web service composition algorithm and planning architecture. In Journal of Systems and Software</p>
<p>[P16] Chafle, G.; Doshi, P.; Harney, J.; Mittal, S.; Srivastava, B. 2007. Improved Adaptation of Web Service Compositions Using Value of Changed Information, In IEEE Intl. Conf. on Web Services (ICWS'07)</p>	<p>[P17] Eckert, J.; Repp, N.; 0002, S. S.; Berbner, R. &amp; Steinmetz, R. 2007. An Approach for Capacity Planning of Web Service Workflows. in John A. Hoxmeier &amp; Stephen Hayne, ed., 'AMCIS', Association for Information Systems</p>	<p>[P18] Wancheng N.; Lingjuan H.; Lianchen L.; Cheng W. 2007. Commodity-Market Based Services Selection in Dynamic Web Service Composition. In 2nd IEEE Asia-Pacific Service Computing Conference</p>
<p>[P19] Mohabey, M.; Narahari, Y.;</p>	<p>[P20] Yang Y.; Tang S; Xu Y.,</p>	<p>[P21] Yu T.; Zhang Y.; Lin</p>

Mallick, S.; Suresh, P.; Subrahmanya, S. V. 2007. A Combinatorial Procurement Auction for QoS-Aware Web Services Composition. In IEEE Intl. Conf. on Automation Science and Engineering (CASE'07).	Zhang W., Fang L. 2007. An Approach to QoS-aware Service Selection in Dynamic Web Service Composition. In 3rd Intl. Conf. on Networking and Services (ICNS '07)	K. 2007. Efficient algorithms for Web services selection with end-to-end QoS constraints. ACM Trans.\
[P22] Berbner, R.; Spahn, M.; Repp, N.; Heckmann, O. & Steinmetz, R. 2006, An Approach for Replanning of Web Service Workflows., in Guillermo Rodríguez-Abitia & Ignacio Ania B., ed., 'AMCIS', Association for Information Systems	[P23] Gao A.; Yang D.; Tang S.; Zhang M. 2005. Web Service Composition Using Integer Programming-based Models. In IEEE Intl. Conf. on e-Business Engineering (ICEBE '05)	[P24] Yu T.; Lin K. 2005. A broker-based framework for QoS-aware Web service composition. In IEEE Intl. Conf. on e-Technology, e-Commerce and e-Service (EEE '05)
[P25] Canfora G.; Penta M.; Esposito R.; Villani M. 2005. QoS-Aware Replanning of Composite Web Services. In IEEE Intl. Conf. on Web Services (ICWS '05)	[P26] Zeng L.; Benatallah B.; Ngu A., Dumas M.; Kalagnanam J.; Chang H. 2004. QoS-Aware Middleware for Web Services Composition. IEEE Trans. Softw. Eng.	[P27] Menasce D. A. 2004. Composing Web Services: A QoS View. In IEEE Internet Computing
[P28] Aggarwal R.; Verma K.; Miller J.; Milnor W. 2004. Constraint Driven Web Service Composition in METEOR-S. In IEEE Intl. Conf. on Services Computing (SCC '04)	[P29] Badidi E.; Esmahi L.; Serhani M. A. 2005. A Queuing Model for Service Selection of Multi-classes QoS-aware Web Services. In 3rd European Conf. on Web Services (ECOWS '05)	[P30] Boone B.; Hoecke S. V.; Seghbroeck G. V.; Joncheere N.; Jonckers V.; Turck F. D.; Develder C.; Dhoedt B. 2010. SALSA: QoS-aware load balancing for autonomous service brokering. J. Syst. Softw.
[P31] Guo H.; Huai J.; Li H.; Deng T.; Li Y.; Du Z.. 2007. ANGEL: Optimal Configuration for High Available Service Composition. In IEEE Intl. Conf. on Web Services, (ICWS'07)	[P32] Serhani M. A.; Dssouli R.; Hafid A.; Sahraoui H. 2005. A QoS Broker Based Architecture for Efficient Web Services Selection. In IEEE Intl. Conf. on Web Services (ICWS '05)	

**[RQ3]** *Que métricas de software estão sendo adotadas para medir a escalabilidade destas soluções?*

Os trabalhos adotam uma ampla gama de métricas para avaliar a escalabilidade de suas soluções. A maior parte dos trabalhos (18 dos 32) avalia o impacto da variação das características em *tempo de execução*, como [P1], [P15] e [P26]. Quatro artigos utilizam, como sua métrica de escolha, taxas de sucesso e falha na seleção de serviços ([P6][P8][P11][P26]). Os trabalhos [P9], [P14] e [P19] observam o valor da utilidade alcançada, considerando o resultado ótimo, enquanto que os artigos [P12] e [P17] medem os custos. Outras métricas são mencionadas por, no máximo, um artigo, como o *valor de adequação* (fitness) do algoritmo, a *média dos QoS violados*, *speedup*, *complexidade de tempo*, *percentagem de soluções ótimas*, *tempo de interrupção* causado pelo processo de re-seleção, *QoS agregada*, *uso de memória*, e *disponibilidade*. Aproximadamente metade dos artigos, 15 dos 32, utiliza mais de uma métrica para avaliar suas soluções.

### [Outras observações]

Outra fonte de variação entre os artigos está nas técnicas de composição dinâmica de serviços utilizadas. No conjunto de artigos avaliados, são adotadas 26 técnicas diferentes e sete funções de otimização/utilidade para encontrar um bom conjunto de serviços concretos. Das funções de otimização, a mais popular é a soma ponderada, adotada por 20 artigos.

Outra fonte de variação são as QoS consideradas para a composição de serviços – Mencionam-se 21 qualidades. Observe-se que estas são diferentes das qualidades utilizadas na avaliação de escalabilidade. A maior parte das soluções adotou *tempo de execução* (27 artigos), seguido por *custo* (21 artigos), *disponibilidade* (19 artigos) e *confiabilidade* (18 artigos). Outras QoS mencionadas foram *throughput*, *popularidade / reputação*, *taxa de sucesso*, *facilidade de composição* e *qualidade da documentação*. O número de QoS consideradas também variou. A maior parte dos artigos, 27 dos 32, usam entre 2 e 6 QoS. Somente dois, [P9] e [P7], usam um número maior de QoS, 9 e 10, respectivamente. Três artigos não especificaram quais são as QoS utilizadas.

### 3.4. Discussão e Trabalhos Futuros

A presente revisão sistemática buscou identificar se, e como, autores no domínio de composição dinâmica de serviços estão avaliando a escalabilidade de suas soluções. Tal análise permite-nos avaliar se existe (1) uniformidade na avaliação de escalabilidade; e, ao mesmo tempo (2) um conjunto comum de variáveis com as quais os autores, neste domínio, devem preocupar-se ao desenvolver suas soluções. Com este objetivo, foi construído e seguido um protocolo, como definido por Kitchenham and Charters (2007).

A seguir serão discutidos os resultados desta revisão, assim como as ameaças à sua validade, e uma proposta de trabalhos futuros.

#### 3.4.1. Discussão dos Resultados

A resposta à pergunta RQ1 confirma que escalabilidade é realmente importante no domínio de composição dinâmica de serviços baseados em QoS. A maior parte dos autores tenta justificar as suas alegações de escalabilidade (explícitas ou não) com avaliações das qualidades de software de interesse, dadas as variações de características no domínio de aplicação e no design do sistema. No entanto, as respostas para as perguntas RQ2 e RQ3 demonstram que estas avaliações variam muito.

Com relação às características cujos valores variam (RQ2), os autores estão preocupados tanto com o domínio da aplicação quanto com o design do sistema. Existe um acordo geral de que o tamanho do *workflow* e o número de serviços candidatos devem ser considerados. Alguns trabalhos também concordam que o número de atributos de QoS é importante. No entanto, para todas as outras quantidades do domínio de aplicação, cada artigo tem suas próprias preocupações. Em relação às características do design do sistema, não existe nenhum consenso sobre quais delas são de interesse. Isto era esperado, dado que cada estudo tem a sua própria abordagem para a composição de serviços. Os valores considerados para as características do domínio de aplicação e do design do sistema também diferem amplamente. Mesmo em relação às variáveis mais utilizadas, como tamanho do *workflow* e o número dos serviços candidatos, os intervalos variam em ordens de magnitude.

Em relação às métricas usadas para medir a escalabilidade das soluções (RQ3), dos trabalhos concordam que o tempo de execução deve ser uma preocupação importante, seguido das taxas de sucesso/falhas, da utilidade e dos custos. O estudo também revelou muitas outras métricas menos populares. Adicionalmente, cada trabalho avaliou a sua própria combinação de métricas.

Todas estas diferenças na análise de escalabilidade das soluções não surpreendem, uma vez que também foi observada uma ampla gama de técnicas utilizadas para a composição dinâmica de serviços, incluindo diferentes atributos de QoS e funções de otimização. O fato de que não existe um único conjunto de métricas e de características que seja adequado para todas as soluções deste domínio é consistente com a visão de escalabilidade em Duboc et. al (2007).

O problema com esta falta de uniformidade é que autores trabalhando na área de composição dinâmica de serviços podem encontrar dificuldades em planejar a avaliação de escalabilidade de suas soluções, ou até mesmo em comparar a escalabilidade de suas abordagens com as outras existentes na literatura. O que esses autores necessitam é de um mecanismo sistemático para a seleção de métricas e características a serem utilizadas em suas análises de escalabilidade. Tal mecanismo, permitiria identificar, para cada solução, quais aspectos do domínio de aplicação e do design do sistema podem afetar a escalabilidade do mesmo. A utilização de abordagens *ad-hoc* pode resultar no esquecimento de variáveis relevantes para a análise.

#### **3.4.2. Ameaças à Validade da SLR**

A fim de reconhecer as limitações desta revisão sistemática, são discutidas suas ameaças à validade:

*Validade de Constructo* está preocupada em estabelecer as medidas operacionais corretas para os conceitos que estão sendo estudados. Nosso estudo busca entender como autores em composição dinâmica de serviços baseada em QoS estão avaliando a escalabilidade de suas soluções. As medidas nele coletadas foram listadas na Seção 3.1.4. O maior risco envolvendo estas medidas diz respeito à sua adequação para responder às perguntas de pesquisa. Acreditamos que as medidas coletadas são suficientes para prover uma caracterização justa do estado-da-arte na análise de escalabilidade, de acordo com a definição de Duboc et. al (2007).

*Validade Interna* estabelece um relacionamento causal, no qual se demonstra que certas condições levam a outras condições. No contexto, as principais ameaças são a seleção incompleta ou errada de estudos primários e o viés dos pesquisadores individuais na avaliação dos estudos. Para atenuar tais ameaças, seguiu-se um protocolo pré-definido. A seleção dos estudos e a extração dos dados foi feita por dois pesquisadores, dado que parte dos artigos foi analisada por ambos para garantir consistência. A decisão de computar os primeiros 50 artigos em cada engenho de busca é um tanto arbitrária. Qualquer artigo classificado abaixo deste limiar foi perdido. Também os valores inferidos dos gráficos publicados nos artigos são aproximados, em razão da baixa resolução de alguns deles. Portanto, esses valores podem ser internamente consistentes, mas não são necessariamente exatos.

*Validade Externa* está relacionada com a generalização dos achados dos estudos. O escopo do nosso estudo é a área de análise de escalabilidade da composição dinâmica de serviços baseados em QoS. Houve, da nossa parte, a decisão consciente de utilizar

somente bancos de dados acadêmicos. O que exclui automaticamente relatórios comerciais e apresentações não indexadas naqueles engenhos. É possível que alguma entidade comercial que não publica em meios acadêmicos tenha uma forma diferente de medir escalabilidade.

### 3.4.3. Trabalhos Futuros

A falta de uniformidade no tratamento da escalabilidade, no domínio de composição dinâmica de serviços baseadas em QoS, pode representar certa dificuldade para autores e *stakeholders* na área. Esta é, entretanto, uma situação inevitável, dada a ampla gama de técnicas existentes. Para resolver o problema, pretendemos investigar o uso do Framework KAOS [Lamsweerde 2008][Duboc 2013] no estabelecimento das características que afetam as metas de qualidade de interesse do sistema, assim como as métricas que as quantificam. Espera-se com tal abordagem obter um modo sistemático de ajudar os autores a planejar, implementar e avaliar a escalabilidade de suas infraestruturas de composição dinâmica de serviços, assim como permitir que *stakeholders* comparem a escalabilidade de soluções alternativas.

## 4. Referências

- Budgen, D., Turner, M. Brereton, P. and Kitchenham, B. (2008) “Using Mapping Studies in Software Engineering”. In Proceedings of PPIG 2008, Lancaster University, pp.195-204.
- Dar, K., Taherkordi, A., Rouvoy, R., Eliassen, F.: (2011) “Adaptable service composition for very-large-scale internet of things systems”. In: Proceedings of the 8th Middleware Doctoral Symposium, MDS '11, pp. 2:1 {2:6. ACM
- Duboc L., Letier E., and Letier D. (2013) “Systematic elaboration of scalability requirements through goal-obstacle analysis,” *Software Engineering, IEEE Transactions on*, vol. 39, no. 1, pp. 119 –140, jan. 2013.
- Duboc L., Rosenblum D. and Wicks T. (2007) “A framework for characterization and analysis of software system scalability”. Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering - ESEC-FSE '07, page 375, 2007. 41, 103.
- Gao A., Yang D., Tang S. and Zhang M.(2005). “Webservice composition using integer programming-based models”. *IEEE International Conference on e-Business Engineering (ICEBE'05)*, pages 603–606, 2005. 42, 43
- Gu Q. and Lago P. (2009) “Exploring service-oriented system engineering challenges: a systematic literature review,” *Service Oriented Computing and Applications*, vol. 3, no. 3, pp. 171–188, Jul.
- Issarny, V., Georgantas, N., Hachem, S., Zarras, A., Vassiliadis, P., Autili, M., Gerosa, M. and Hamida, A. (2011) 'Service-oriented Middleware for the Future Internet: State of the Art and Research Directions', *Journal of Internet Services and Applications* 2 (1) , 23-45 .

- Kitchenham B. A. and Charters S. (2007) “Guidelines for performing systematic literature reviews in software engineering”. *Evidence-Based Software Engineering*, (EBSE 2007- 001), 2007. 27, 36
- Kitchenham B., Dyba T. and Jorgensen M. (2004) “Evidence-based software engineering,” *Proceedings. 26th International Conference on Software Engineering*, pp. 273–281.
- Ko J., Kim C., and Kwon I. (2008)“Quality-of-service oriented web service composition algorithm and planning architecture”. *Journal of Systems and Software*, 81(11):2079–2090. xi, 42, 45
- Lamsweerde A. (2008)“Systematic Requirements Engineering: From System Goals to UML Models to Software Specifications”. John Wiley & Sons.
- Leite, L. A. F., Oliva, G. A., Nogueira, G. M., Gerosa, M. A., Kon, F. & Milojicic, D. S. (2013). A systematic literature review of service choreography adaptation..*Service Oriented Computing and Applications*, 7, 199-216.
- OWL-S: Semantic Markup for Web Services. (2004) Disponível em: <http://www.w3.org/Submission/OWL-S/> Acessado em: Out. 2013.
- Paolucci, M.; K. Sycara (2003). "Autonomous Semantic Web services". *Internet Computing (IEEE)* 7 (5): 34–41
- Sheth J., A., Miller J., Arnold J. and Kochut K.(2004) “Quality of service for workflows and web service processes,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, no. 3, pp. 281–308.
- SOAP 1.2. - Simple Object Access Protocol Version 1.2 Part 1: Messaging Framework (2007). Disponível em: <http://www.w3.org/TR/soap12/> Acessado em: Out. 2013.
- Winter V. L., Berg R. S. and Ringland J. T.(2001) “High integrity software.” Norwell, MA, USA: Kluwer Academic Publishers, pp. 115–135. Disponível em: <http://dl.acm.org/citation.cfm?id=503188.503196>
- WSDL 2.0 - Web Services Description Language Version 2.0 Part 1: Core Language. (2007) Disponível em: <http://www.w3.org/TR/wsdl20/> Acessado em: Out. 2013.
- Yu T., Zhang Y., and Lin K. (2007) Efficient algorithms for Web services selection with end-to-end QoS constraints. *ACM Transactions on the Web*, 1(1):6–32, May 2007. 27, 42, 44, 45
- Zeng L., Benatallah B., Ngu A.H.H., Dumas M., Kalagnanam J. and Chang H.. (2004)“QoS-aware middleware for Web services composition”. *IEEE Transactions on Software Engineering*, 30(5):311–327