

APLICANDO ATAM NA ARQUITETURA DO SISTEMA DE INFORMAÇÕES DE PREFEITURAS

Paulo Lima, pcdl1970@gmail.com

Adriana Carniello, adrianacarniello@ifsp.edu.br

Andreia Carniello, andreiacarniello@ifsp.edu.br

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP Campus Guarulhos

RESUMO. A importância da arquitetura de software nos sistemas de informações atuais é cada vez mais evidente. Existem diversos métodos propostos para avaliação da arquitetura a fim de verificar e validar a solução proposta com relação ao atendimento das expectativas de negócio e dos requisitos de qualidade. Vários trabalhos avaliaram estes métodos na busca de uma abordagem holística que possa ser aplicada às diversas modalidades de sistemas. Por meio da pesquisa dos principais métodos de avaliação de arquitetura de software e da seleção de um conjunto destes métodos (SAAM, ATAM, CBAM), este trabalho analisa as características, propostas e indicações do conjunto de métodos selecionado para utilização como apoio à tomada de decisão na escolha de arquiteturas de software. Como estudo de caso, foi aplicado o método ATAM no sistema de informações de prefeituras INFOPREFEITURA. Ao final deste estudo de caso, foi apresentada a arquitetura de software proposta para atender aos direcionadores do negócio e aos principais atributos de qualidades de acordo com a pontuação final de importância de cada uma das categorias de requisitos não funcionais, seguindo os cenários indicados pelas partes interessadas. A realização deste trabalho permitiu propor a utilização de métodos baseados em cenários para avaliação de arquiteturas de software e afirmar que o balanceamento dos requisitos de qualidade conflitantes direciona para uma abordagem assertiva na decisão da arquitetura.

Palavras-chave: Arquitetura de Software. Verificação. Validação. Sistemas.

1. INTRODUÇÃO

Hoje em dia, a Arquitetura de um Software é um ativo chave para organizações que constroem sistemas complexos intensivamente baseados em software. A arquitetura é criada no começo do desenvolvimento e fornece aos desenvolvedores meios para criar um projeto de alto nível do sistema de forma a assegurar que seja possível implementar todos os requisitos que precisam ser atendidos. Existem várias definições para Arquitetura de Software, com poucas diferenças de acordo com o domínio e a experiência das pessoas.

A arquitetura descreve quais componentes de alto nível compõem um software, bem como quais responsabilidades esses componentes assumem em relação a outros componentes no sistema. Também descreve a organização dos componentes tanto no nível conceitual quanto em detalhes, visto que cada componente pode ter sua própria estrutura de arquitetura. Por fim, a arquitetura define quais interfaces os componentes expõem aos demais e quais as interfaces e componentes que eles, por sua vez, usam. A arquitetura é criada com base em um conjunto de requisitos obtidos junto aos stakeholders (usuários, desenvolvedores, etc.).

Requisitos funcionais descrevem o que o sistema deve fazer e requisitos de qualidade descrevem um conjunto de características que o sistema deve apresentar (por exemplo, quanto tempo o sistema pode levar para concluir uma operação, grau de facilidade para manutenção do sistema, etc.). Para auxiliar os desenvolvedores de software a garantir que a arquitetura do software atenda aos atributos de qualidade, vários métodos foram propostos na literatura. Este artigo tem como objetivo avaliar os métodos SAAM, ATAM, CBAM e indicar um método para ser utilizado na avaliação da arquitetura de software do sistema de Informação de Prefeituras dos municípios brasileiros, o INFOPREFEITURA, como proposta de validação e verificação dos requisitos que serão implementados no projeto.

2. Métodos de Avaliação de Arquitetura de Software

A arquitetura de um software é a descrição da estrutura do sistema, que compreende os elementos de software, o relacionamento entre estes elementos e as propriedades externamente visíveis destes elementos. A arquitetura é o primeiro artefato que pode ser analisado para determinar o quão bem os seus atributos de qualidade estão sendo alcançados. A arquitetura serve como o plano do projeto e também como um veículo de comunicação, pois é a manifestação das primeiras decisões do projeto e é uma abstração reutilizável que pode ser transferida para sistemas futuros (BASS; CLEMENTS; KAZMAN, 2012).

Atualmente existem métodos de avaliação de arquitetura de software baseados em cenários. Vários desses métodos são baseados no SAAM ou no ATAM, ambos criados a partir da iniciativa e pesquisa do Carnegie Mellon Institute (CMI). Esses métodos normalmente são propostos para algum tipo de sistema e para um conjunto específico de qualidades ou requisitos não-funcionais (RNF) específicos.

A maioria dos métodos foram desenvolvidos a partir de outro existente, propondo uma evolução ou extensão do método tomado como base. Com relação à avaliação da eficácia destes métodos, observa-se que não tem havido muito

esforço nesta direção. No entanto, existe uma iniciativa do grupo de trabalho internacional *Software Architecture Review and Assessment* (SARA) na publicação da revisão dos métodos de avaliação existentes (IONITA; HAMMER; OBBINK, 2002).

A qualidade do software é definida como o grau em que o software possui uma desejada combinação de atributos. Os requisitos de qualidade que um software deve satisfazer são comumente divididos em dois grupos: Qualidades de Desenvolvimento (manutenibilidade, entedibilidade e flexibilidade) e Qualidades Operacionais (desempenho e usabilidade). Durante o processo de desenvolvimento da arquitetura é importante validar que a arquitetura tem os atributos de qualidade necessários. Isso é feito usualmente utilizando uma ou mais avaliações da arquitetura (MATTSON; GRAHN; MÅRTENSSON, 2006). Os atributos de qualidade em foco são:

Manutenibilidade – Grau de Facilidade para modificar, corrigir ou melhorar um componente ou um sistema de software;

Desempenho – O grau com o qual um sistema pode completar suas funções dentro de determinadas restrições (tais como velocidade, precisão, uso de memória, etc.);

Testabilidade – Grau de facilidade com o qual pode-se estabelecer critérios de teste e testes de desempenho para saber se os requisitos foram atendidos;

Portabilidade – Facilidade com a qual um sistema pode ser transferido entre ambientes de hardware ou software.

Neste trabalho, avaliam-se os seguintes métodos de análise dos atributos de qualidade de arquiteturas de software:

1. SAAM, *Software Architecture Analysis Method*;
2. ATAM, *Architecture Trade-off Analysis Method*;
3. CBAM, *Cost Benefit Analysis Method*.

2.1. SAAM

Foi o primeiro método de análise de arquitetura de software baseado em cenários amplamente divulgado. Foi criado para avaliar a modificabilidade da arquitetura em vários aspectos. Foi proposto como método para expressar diversas reivindicações de arquitetura através de cenários e a partir da avaliação confrontando cenários atuais. Na prática SAAM tem provado ser útil para avaliação rápida de muitos atributos de qualidade como modificabilidade, portabilidade, extensibilidade, integralidade, assim como requisitos funcionais. Na análise de uma única arquitetura, SAAM indica os pontos fracos ou fortes, juntamente com os pontos onde a arquitetura falha no atendimento aos requisitos de modificabilidade. Se duas ou mais arquiteturas candidatas, que forneçam as mesmas funcionalidades, são comparadas com relação à modificabilidade, SAAM pode produzir um ranking relativo entre essas.

O método consiste em cinco passos, começando com a documentação da arquitetura de uma forma que todos os participantes da avaliação consigam entender. Depois, cenários que descrevam o uso pretendido do sistema são criados. Os cenários devem representar os interesses de todos os *stakeholders*. Os cenários são avaliados e um conjunto que representa os aspectos que devem ser avaliados é selecionado. Cenários que interagem entre si são identificados como uma maneira de medir a modularidade da arquitetura. Os cenários são ordenados de acordo com a prioridade e impacto previsto na arquitetura.

2.2. ATAM

Segundo o trabalho de (IONITA; HAMMER; OBBINK, 2002), o *Architecture Trade-off Analysis Method* (ATAM), é um método de arquitetura baseado em cenários para avaliação de atributos de qualidade como modificabilidade, portabilidade, extensibilidade, e integralidade. Além disso, explora a interação entre os atributos de qualidade e suas interdependências destacando o mecanismo de balanceamento e oportunidades entre as diferentes qualidades.

Para a condução com sucesso de uma sessão de avaliação do ATAM, os participantes deste método devem satisfazer ao seguinte conjunto inicial de pré-requisitos:

- Os avaliadores devem entender a arquitetura do sistema, reconhecer os parâmetros arquiteturais, definir suas implicações com relação aos atributos de qualidade do sistema, e confrontar estas implicações com os requisitos;
- As áreas problemas são chamadas “pontos de sensibilidade”, “contrapontos” e “riscos”;
- Os atributos de qualidade devem ser entendidos através de cenários descritivos para avaliação dos atributos de qualidade;
- Uma sessão de avaliação do ATAM utiliza como entradas os requisitos iniciais do sistema e a descrição da arquitetura de software do sistema.

O método ATAM consiste de quatro fases: apresentação, investigação e análise, teste e relatório. Cada fase consiste em uma coleção de passos. A fase de apresentação envolve a troca de informações entre as apresentações. A fase de investigação e análise se preocupa com a avaliação dos requisitos de atributos chaves de qualidade versus as abordagens arquiteturais. A fase de teste compara os resultados das fases anteriores para as necessidades das partes interessadas mais relevantes. Finalmente, a fase de relatório sumariza os resultados do ATAM.

2.3. CBAM

O CBAM quantifica benefícios usando valores de utilidade que representam preferência do projeto, determinados por votos ou consenso entre as partes interessadas. O benefício que será adicionado ao sistema através da implementação de uma Estratégia de Software (ES) é referido como de utilidade. Cada ES oferece um nível específico de utilidade para os interessados, e cada ES tem implicações de custo e cronograma. Mas, desde que o valor de utilidade de um projeto é determinado de forma independente, não podemos garantir que o valor seja razoável. Além disso, a tomada de decisão por consenso da importância dos RNF no início do processo, pode não refletir o conhecimento de cada parte interessada, pode gerar muita incerteza na decisão de projeto. Além disso, uma vez que as partes interessadas devem considerar vários tipos de informação de decisão simultaneamente e de forma independente, a possibilidade de erro de julgamento é alta.

3. Descrição do sistema INFOPREFEITURA

O escopo do INFOPREFEITURA é criar uma “Homepage” para Prefeituras a fim de atender a demanda da Lei Federal nº 9755/98, adicionando ainda informações sobre o município e outros serviços para a população.

A “Homepage” da Prefeitura visa informar os cidadãos, turistas e prestadores sobre os mais diversos tipos de serviços e demandas da cidade.

Utiliza informações em tempo real, conectando diretamente com o Banco de Dados da Prefeitura, podendo assim executar tarefas agilmente, como solicitações dos municípios acerca dos mais diversos pedidos, como limpeza pública, transporte, energia, água, etc.

Apresenta, também, dados demográficos, estatísticos e pontos turísticos, também acessando banco de dados da Prefeitura e do Censo, apresentando inclusive imagens de arquivos digitalizadas.

Disponibiliza informações on-line para licitações e cadastramento de prestadores de serviços, visando, assim, agilizar os processos de aquisição de bens e serviços, sendo as partes interessadas (stakeholders):

- Contribuintes: São usuários que buscam informações e obtenção de documentos referentes aos impostos, relatórios de gestão, certidões, etc.
- Fornecedores: São aqueles que possuam contratos de prestação de serviços e/ou materiais com o Município, com participações em licitações;
- Órgãos públicos: São órgãos ligados à Prefeitura, autarquias que se submetem a ela;
- Visitantes: São usuário comuns que acessam o site apenas para obter informações da cidade, como, por exemplo, turistas;
- Atendentes: São funcionários que acessam a Homepage para prestar a orientação ao cidadão.

Como Requisitos Não-Funcionais constam:

- Confiabilidade: RNF001 – O sistema deve estar disponível e operacional; RNF002 – Deve possuir um sistema de backup, apto a entrar em operação caso aconteça falha no servidor principal;
- Segurança e privacidade: RNF003 – O Sistema deve garantir ao usuário que seus dados estejam seguros na base de dados, bem como suas informações só possam ser acessadas por ele ou por quem possuir autorização prévia;
- Desempenho: RNF004 – O Sistema deve garantir um fluxo de informação estável e contínuo, ainda que dentro de um limite pré-estabelecido; RNF005 – O tempo de resposta há uma transação não deve exceder o tempo mínimo pré-estabelecido; RNF006 – Caso o limite de alguma função estiver excedido, o usuário deverá receber informação instantaneamente; RNF007 – O Sistema deve trabalhar com uso máximo de 75% da capacidade de processamento do servidor, 75% da memória RAM e 75% da conexão de rede, a fim de evitar sobrecarga no sistema computacional;
- Qualidade: RNF008 – O Sistema deve seguir os mesmos padrões e linguagem em todos os seus recursos, a fim de facilitar a interação do usuário;
- RNF009 – O código-fonte do sistema deve ser padronizado e documentado, a fim de facilitar possíveis ajustes por outros funcionários;
- Usabilidade: RNF010 – O Sistema deve possuir recurso de ajuda ao usuário e e-mail para contato; RNF011 – Deve possuir auxílio para pessoas portadoras de necessidades especiais;
- Portabilidade: RNF012 – Deve ser possível acessar o site em qualquer navegador de internet; RNF013 – O acesso ao sistema não poderá depender do Sistema Operacional, ou seja, deve ser desenvolvido com linguagem multiplataforma; RNF014 – O acesso deve estar disponível para quaisquer dispositivos, seja PC, Tablet ou Smartphone.

4. A escolha do método de Arquitetura para o Sistema INFOPREFEITURA

O método ATAM analisa quão bem a arquitetura de software satisfaz objetivos particulares de qualidade. Ele também fornece direções sobre interdependências dos atributos de qualidade - o que significa como trade-off contra o outro. O ATAM é baseado no Método de Análise de Arquitetura de Software (SAAM) e é considerado uma evolução desse método. O ATAM é um método de arquitetura baseado em cenários para avaliação de atributos de qualidade como modificabilidade, portabilidade, extensibilidade e integralidade. Além disso, explora a interação entre os atributos de qualidade e suas interdependências destacando o mecanismo de balanceamento e oportunidades entre as diferentes qualidades.

O CBAM prioriza aspectos não abordados no ATAM, pois trata-se de um método de arquitetura centrado em analisar os custos, benefícios e implicações no cronograma de decisões de arquitetura. O CBAM também avalia o nível de incerteza associado à essas decisões. O CBAM é um método centrado na arquitetura para a análise das implicações dos custos, benefícios e prazo das decisões de arquitetura. Ele também permite a avaliação da incerteza em torno das decisões de custos e benefícios, proporcionando assim uma base para a tomada de decisão informada sobre projeto arquitetural e sua evolução.

Devido às características do sistema INFOPREFEITURA que prioriza os aspectos técnicos e qualidades sistêmicas, e considerando que o ATAM é mais abrangente que o SAAM, além disso, pelo fato do foco do sistema não está relacionado com os aspectos de custo e prazos que é o ponto forte do CBAM, será adotado o método ATAM no estudo de caso proposto.

Seguem as fases e os passos do ATAM aplicados no estudo de caso INFOPREFEITURA:

Fase Apresentação:

ATAM Passo 1 – apresentando ATAM

ATAM foi apresentado para os participantes:

- Chefes de seções envolvidos no projeto;
- Líderes técnicos envolvidos no desenvolvimento;
- Gerência de projeto.

ATAM Passo 2 – Apresentação dos direcionadores de negócio

A gerência de projeto apresenta os objetivos do sistema INFOPREFEITURA e enfatiza os indicadores dos requisitos de qualidade de arquitetura, conforme descritivos dos requisitos não-funcionais (RNF) listados: Confiabilidade; Segurança; Desempenho; Qualidade; Usabilidade; Portabilidade.

ATAM Passo 3 – Apresentação da arquitetura

O arquiteto apresenta a arquitetura de software do sistema focando em como endereça os direcionadores de negócio apresentados no passo anterior.

Fase Investigação e Análise:

ATAM Passo 4 – Identificar as abordagens Arquiteturais:

As abordagens arquiteturais consideradas para o INFOPREFEITURA são:

- .NET Framework MVC
- PHP MVC
- Java EE

ATAM Passo 5 – Gerar Árvore Utilidade de Atributo de Qualidade:

As interessadas listaram e quantificaram (*Quantidade RNF*) os fatores de qualidade do sistema de acordo com as categorias de requisitos abaixo:

- Confiabilidade,
- Segurança,
- Desempenho,
- Qualidade,
- Usabilidade,
- Portabilidade.

Para cada categoria de requisito de qualidade do sistema (RNF) foi atribuído um grau de importância através de uma escala numérica, na qual o menos relevante foi atribuindo o peso um (menos importante) e o mais relevante foi atribuído o peso cinco (mais importante) de acordo com as expectativas e considerações dos envolvidos na definição dos requisitos do sistema.

Foram analisados os cenários de caso de uso e a ocorrência dos requisitos de qualidade e a importância de cada um no contexto do comportamento e resultado esperados no sistema. A partir desta análise obteve-se a quantidade de cada

RNF por categoria cuja importância está representada por um o indicador final que Indicador Final corresponde ao produto da quantidade de RNF multiplicada pelo peso atribuída à categoria do requisito. Deste modo o Indicador Final é obtido pela expressão: $Indicador\ Final = Quantidade\ RNF \times Peso$.

Tabela1 dos atributos de qualidade que compreendem o sistema INFOPREFEITURA:

Categoria Requisito	Quantidade RNF	Peso (1 a 5)	Indicador Final
Confiabilidade	2	3	6
Segurança	1	3	3
Desempenho	4	5	20
Qualidade	2	5	10
Usabilidade	2	3	6
Portabilidade	3	5	15

ATAM Passo 6 – Analisar Abordagens Arquiteturais

Baseados nos cenários mais prioritários identificados no passo anterior, as abordagens arquiteturais que endereçam esses cenários foram elicitadas e analisadas. Durante este passo os riscos potenciais, possíveis não-riscos, os pontos de sensibilidade e contrapontos foram identificados. As abordagens arquiteturais avaliadas como possíveis opções são:

- .NET Framework MVC
- Java EE

Fase de Teste:

ATAM Passo 7 – Avaliar e priorizar cenários

Os cenários foram priorizados por votação com relação ao peso de cada um para atendimento aos requisitos de qualidade do INFOPREFEITURA. Os resultados foram pontuados na coluna Indicador Final.

ATAM Passo 8 – Reanalisar Abordagens Arquiteturais

Os cenários priorizados foram utilizados para reiteraões do Passo 6 com objetivo de selecionar a abordagem arquitetural mais indicada para INFOPREFEITURA.

Fase Relatório:

ATAM Passo 9 – Apresentação Resultados

A abordagem arquitetural escolhida para atendimento aos requisitos de qualidade do sistema INFOPREFEITURA foi o padrão de arquitetura Java EE.

Segue na Figura 1 a arquitetura de software proposta para solução do INFOPREFEITURA:

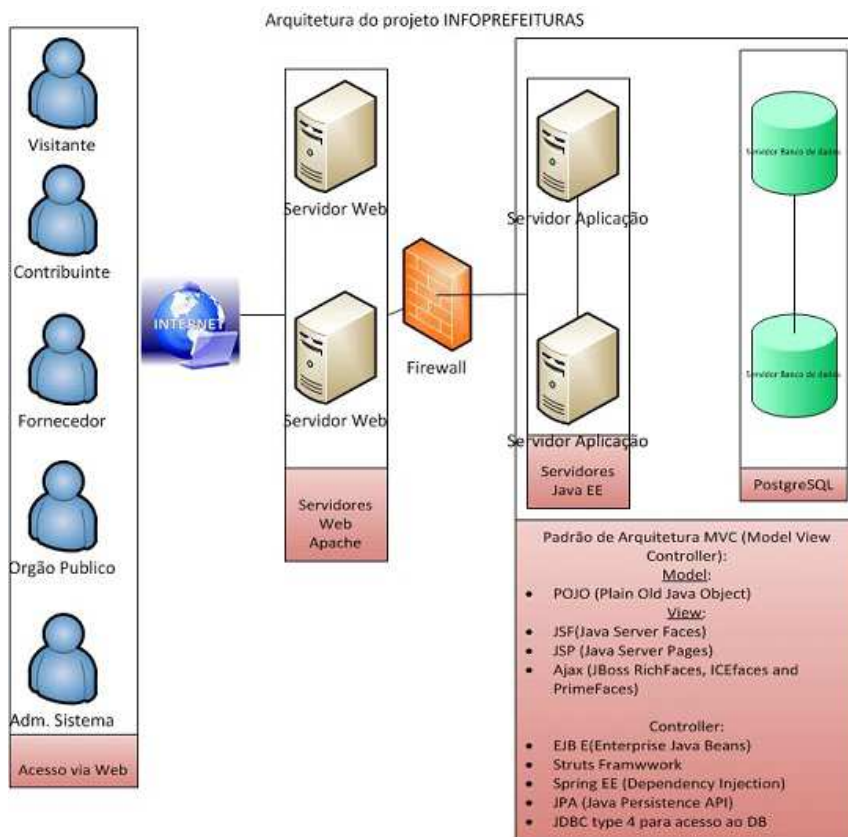


Figura 1. Diagrama de arquitetura em camadas do sistema INFOPREFEITURA

Segundo JUNEAU (2013), Java é uma linguagem de programação madura, que tem sido aperfeiçoada ao longo dos anos em uma linguagem produtiva amplamente utilizada no desenvolvimento de aplicações corporativas. A plataforma Java EE 7 abrange as soluções utilizando as mais atuais tecnologias Java Enterprise e permite desenvolver muito mais rápido do que com as tecnologias mais antigas.

5. Proof-of-Concept (PoC)

Segundo EELES e CRIPPS (2009), depois de tomar as decisões sobre as tecnologias específicas a serem utilizadas na arquitetura, muitas vezes faz sentido provar a arquitetura através da criação de uma Proof-of-Concept (PoC) de arquitetura. A Prova de Conceito arquitetural é (opcionalmente) desenvolvida na fase inicial para ajudar a determinar a viabilidade do projeto, avaliar os riscos técnicos inerentes ao seu desenvolvimento, e formular e refinar os requisitos arquitetonicamente significativos.

A Prova de Conceito de arquitetura (PoC) pode assumir várias formas, por exemplo:

- Uma lista de tecnologias conhecidas (*frameworks*, padrões, arquiteturas executáveis) que parecem apropriadas para a solução;
- um esboço de um modelo conceitual de uma solução usando uma notação, como UML;
- uma simulação de uma solução;
- um protótipo executável.

Neste projeto de pesquisa, propõe-se a realização da Prova de Conceito (PoC) implementando uma arquitetura de referência através da implementação em camadas desta arquitetura a fim de validar a arquitetura proposta.

Considerando a Prova de Conceito (PoC) na sua forma de realização através de tecnologias, neste projeto de pesquisa, pretende-se aplicar a plataforma Java EE 7 na validação da arquitetura do sistema proposto, demonstrando, assim, como realizar PoC através de implementação de parte do sistema para corroborar na verificação e validação da arquitetura proposta para solução do sistema que será desenvolvido.

6. CONCLUSÕES

A arquitetura de um sistema de software tem sido um aspecto importante do desenvolvimento de software. Uma boa arquitetura deve melhorar a probabilidade de que o sistema atenda aos seus requisitos. Muitos dos métodos somente abordam um atributo de qualidade e poucos avaliam vários atributos simultaneamente no mesmo *framework* ou método. Especificamente, somente o método ATAM inclui análise de *trade-off* (“*perde-e-ganha*”).

A partir da pesquisa dos principais métodos de avaliação de arquitetura de software e da seleção do método ATAM, este trabalho analisou suas características, propostas e indicações para utilização como apoio à tomada de decisão na escolha de arquiteturas de software. O método ATAM foi aplicado como apoio a tomada de decisão da arquitetura proposta no sistema de informações de prefeituras INFOPREFEITURA. Foi justificada a decisão da arquitetura de software proposta pelo atendimento aos direcionadores do negócio e aos principais atributos de qualidades de acordo com a pontuação final de importância de cada uma das categorias de requisitos não-funcionais, seguindo os cenários indicados pelas partes interessadas. A realização deste trabalho permitiu propor a utilização do método ATAM baseado em cenários para avaliação da arquitetura de software e afirmar que o balanceamento dos requisitos de qualidade conflitantes direciona para uma abordagem assertiva na decisão da arquitetura.

Para projetos futuros seria interessante avaliar uma combinação dos métodos introduzidos neste artigo e a elaboração de uma PoC, a fim de ajudar a avaliar a viabilidade de implementação da arquitetura proposta na plataforma indicada e da produtividade das equipes envolvidas nessa arquitetura para atendimento dos prazos, custos e eficácia esperados no projeto.

7. REFERÊNCIAS

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software architecture in practice (3rd Edition)** (SEI Series in Software Engineering). Addison-Wesley Professional, 2012.

EELES, P.; CRIPPS, P. **The process of software architecting**. [s.l.] Pearson Education, 2009.

IONITA, M. T.; HAMMER, D. K.; OBBINK, H. **Scenario-based software architecture evaluation methods: An overview**. ICSE/SARA, Eindhoven (HOL), 2002.

JUNEAU, J. **Java EE 7 Recipes: A problem-solution approach**. Apress, 2013.

MATTSON, M.; GRAHN H.; MÅRTENSSON. **Software Architecture Evaluation Methods for Performance, Maintainability, Testability, and Portability**. Blekinge Institute of Technology, Ronneby(SUE), 2006.

LEE, J.; KANG, S.; KIM, C. **Software architecture evaluation methods based on cost benefit analysis and quantitative decision making**. Springer Science and Business Media, 2008.

8. NOTA DE RESPONSABILIDADE

Os autores são os únicos responsáveis pelo conteúdo deste artigo.