

Edição Nº 6 – 28 de Abril de 2018

ISSN Print: 1646-9976 | ISSN Online: 2184-223X |

DOI: <https://doi.org/10.31112/kriativ-tech-2018-01-15>

<http://www.kriativ-tech.com>

<http://www.kriativ-tech.pt>

## SOA - Principais motivações e benefícios da adopção da arquitectura SOA

Nuno Miguel Carvalho Galego

Professor Adjunto do ISTECS

*ISTEC – Departamento de Estudos e Investigação em Tecnologias de Informação e Sociedade*

[galego.nuno@hotmail.com](mailto:galego.nuno@hotmail.com)

**Resumo** — Nos dias que correm muitas organizações estão a adoptar arquitecturas SOA (*Service Oriented Architecture*) para consumirem serviços em forma de *Web Services*. O objectivo do uso da arquitectura SOA é permitir a interligação de aplicações que estão implementadas em diferentes localizações além das fronteiras da organização em ambientes heterogéneos e distribuídos.

Esta arquitectura pode revolucionar a maneira como as organizações competem, acelerando a tomada de decisões e tirar vantagem das oportunidades de mercado. Baseado na revisão da literatura, o objectivo deste trabalho é discutir as principais motivações e benefícios da adopção da arquitectura SOA.

Este trabalho descreve os resultados de uma pesquisa que teve como objecto central o estudo a arquitectura de *software* orientada a serviços (SOA), onde são identificadas as principais motivações e benefícios em adoptar esta arquitectura.

Também irá ser abordado neste trabalho os tipos de serviços associados ao SOA tal como o modelo de referência para SOA - OASIS.

No final deste trabalho, é apresentada uma ferramenta ao dispor das organizações da IBM que agiliza a implementação de projectos baseados em SOA tal como os benefícios na adopção de uma arquitectura deste tipo.

**Palavras Chave** - *Arquitectura Orientada a Serviços (SOA); Serviços; Web Services; Modelo referencia OASIS; IBM;*

**Abstract** — These days many organizations are adopting SOA architectures (*Service Oriented Architecture*) to consume services form of *Web Services*. The purpose of the use of the SOA architecture is to allow the interconnection of applications that are implemented in different locations beyond the boundaries of the Organization in heterogeneous and distributed environments.

This architecture can revolutionize the way that organizations compete, speeding up decision-making and take advantage of market opportunities. Based on the literature review, of this work discuss the main motivations and benefits of adoption of SOA architecture.

This work describes the results of a survey which had as central study object-oriented software architecture (SOA), which identified the main motivations and benefits in adopting this architecture.

Will also be addressed in this work, the types of services associated with the SOA as the reference model for SOA-OASIS.

At the end of this work, a tool available from IBM organizations that streamlines the implementation of SOA-based projects such as the benefits in adopting an architecture of this type.

**Keywords** - *Service Oriented Architecture (SOA); Web Services; OASIS reference model; IBM;*

## I. INTRODUÇÃO

Arquitectura Orientada a Serviços (SOA), ou, em inglês, *Service-Oriented Architecture* é um termo que descreve duas coisas muito distintas. As duas primeiras palavras, *Service-Oriented*, expressam uma metodologia para desenvolvimento de software. E a terceira palavra, *Architecture* ou Arquitectura, é uma perspectiva dos componentes de um sistema de *software*.

A Arquitectura Orientada a Serviços (SOA) não é uma tecnologia, não é uma metodologia, não é um serviço, mas é um conceito de uma arquitectura que promove a integração entre o negócio e a área de tecnologias de informação por meio de um conjunto de serviços ligados. É um modelo de planeamento estratégico que alinha os objectivos de negócio de uma organização.

As organizações precisam responder de forma efectiva e rápida ao mercado, sendo assim, as aplicações têm que ter flexibilidade em executar mudanças rapidamente. Esta arquitectura tem como objectivo integrar as aplicações, disponibilizar maior flexibilidade para mudanças, suportar serviços independentes de plataforma e protocolos.

Uma arquitectura orientada a serviços é essencialmente um conjunto de serviços interligados que comunicam entre si formando assim um sistema único. A localização dos serviços não é importante, estes podem ser internos à organização ou disponibilizados por outras empresas. A comunicação entre os serviços pode envolver apenas uma simples trocas de dados, ou uma coordenação entre dois ou mais serviços.

A reutilização de serviços é grande vantagem desta arquitectura, aumentando a produtividade, o alinhamento com o negócio e facilidade na gestão das tecnologias de informação. Os serviços são módulos de negócio ou funcionalidades que possuem interfaces expostas que são invocadas via mensagens. Interfaces disponibilizam recursos sem que a implementação do serviço seja conhecida.

No entanto, por se tratar de um modelo de arquitectura orientada a serviços, poderá ser aplicada a qualquer organização que tenha interesse em adoptar esta arquitectura.

## II. ENQUADRAMENTO CONCEPTUAL

### A. O que é SOA

Arquitectura Orientada a Serviços (Barry, 2014) é um estilo de arquitectura que usa serviços como

blocos de construção interligados para facilitar a integração da empresa e o reuso de componentes, é também uma estratégia que congrega a criação de todos os componentes de *software* de uma empresa que comunicam entre si formando assim um único sistema, seguindo a metodologia de programação orientada a serviços.

Este tipo de arquitectura tem um processo evolutivo das técnicas de modularização de *software* que começaram há mais de 50 anos com a introdução da programação estruturada.

A visão SOA tem como componente fundamental os serviços e consiste numa abordagem de arquitectura para a criação de sistemas a partir de serviços autónomos. Com esta filosofia, a integração será uma solução final que será composta por serviços desenvolvidos, em linguagens de programação diferentes, com processos de negócios e modelos de segurança distintos. A localização dos serviços não é importante, estes podem ser internos à empresa ou disponibilizados por outras empresas. A comunicação entre os serviços pode envolver apenas simples trocas de dados, ou uma coordenação entre dois ou mais serviços.

Os serviços devem ser desenhados para proporcionar disponibilidade e estabilidade. Os serviços são criados para durar, enquanto as configurações e as agregações dos serviços são criadas, para mudar.

### B. O que é um serviço

Para compreender claramente o que é uma arquitectura orientada a serviços é necessário definir o que é um serviço.

Um serviço é um grupo de componentes de *software* que suportam um determinado processo de negócio, na concretização de um objectivo em particular, tal como verificar uma transacção utilizando um cartão de crédito, processar uma encomenda ou gerar uma factura.

No contexto de uma arquitectura orientada a serviços, um serviço representa uma função de um sistema que é disponibilizado para outro sistema em que poderá ser uma entidade de *software*, agrupando granularidade, passível de ser descoberta e auto-contido que interage com aplicações e outros serviços por intermédio de um modelo de comunicação pouco acoplado, usualmente suportado por mensagens assíncronas.

Um serviço é uma função ou funcionalidade que se encontra bem definida e que não depende do

contexto ou estado de outros serviços, excepto nos casos de serviços compostos (*composite services*), e deve possuir uma interface bem definida. Normalmente, a comunicação entre o sistema cliente e aquele que disponibiliza o serviço é realizada através de *web services*.

Existem várias implementações possíveis para serviços, sendo os mais comuns:

- *Web Services* utilizando *Simple Object Access Protocol* (SOAP) e *Web Services Description Language* (WSDL)
- Soluções *middleware* orientadas por mensagens (MOM), tais como IBM Webshere
- Sistemas publicação / subscrição, tais como *Java Messaging Service* (JMS)

### C. *Web Services* e a arquitectura orientada a serviços

Segundo a definição do W3C (W3C, 2002) um *Web Service* é um sistema de *Software* identificado por um URI (*Uniform Resource Identifier*) cujas interfaces e ligações estão definidos e descritos em XML. A sua definição pode ser descoberta por outros sistemas de *Software*, que podem então interagir com o *Web Service* segundo as regras pré-estabelecidas, utilizando mensagens XML enviadas sobre protocolos de *Internet*. Por outras palavras um *Web Service* é uma interface para uma funcionalidade de uma aplicação que está disponível através da rede e que foi construído utilizando protocolos *standard* da *Internet*.

*Web Service* é a disponibilização de um serviço pela *Internet* que pode ser acedido em qualquer lugar. Os utilizadores enviam requisições com informações bem definidas e recebem respostas que podem ser síncronas ou assíncronas. A adopção de *Web services* pelas empresas representa uma mudança infra-estrutural muito grande e importante.

Normalmente, os *Web Services* são construídos com base nos seguintes *standards* (Figura1):

- Protocolo HTTP (W3C, 2014): Transmissão de dados pela *Internet*.
- XML (*Extensible Markup Language*) (W3Schools, 2012) – Um *standard* do formato das mensagens.
- SOAP (*Simple Object Access Protocol*) (W3Schools,, 2012) – Um *standard* do protocolo de comunicações usado na invocação dos serviços usando tecnologias *Web*, tais como, HTTP e XML.

- XSD (*XML Schema Definition*) (Thompson, Mendelsohn, Beech, & M., 2012) – *Standard* para descrever a estrutura e restringir o conteúdo de documentos XML..
- WSDL (*Web Services Description Language*) (Christensen, Curbeta, Meredith, & Weerawarana, 2001) – Um *standard*, em formato XML, usado para descrever as operações e parâmetros suportados por um determinado serviço. Através da análise do ficheiro WSDL de um determinado *Web Service*, este pode ser descoberto e os seus métodos invocados.
- UDDI (*Universal Description, Discovery and Integration*) (OASIS, 2014) – Um *standard* utilizado para registar, descobrir e localizar os *Web Services*.

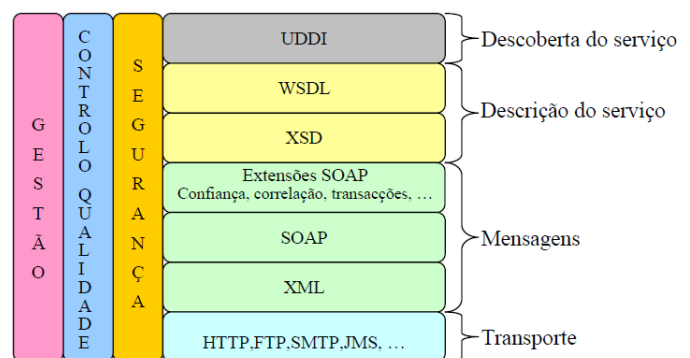


Figura 1: *Standards* dos *Web Services* (Pinho, 2008).

Seguindo estes *standards*, é possível que os *Web Services* residam em qualquer lugar e sejam acedidos por qualquer um. Os *Web Services* são serviços de interacção “servidor-para-servidor”, através de formatos e protocolos já definidos, independentes da plataforma e que utilizam uma linguagem “neutra”. A sua aplicação em conjunto, forma o que é denominada de arquitectura orientada a serviços – SOA (*service oriented architecture*).

Para garantir a fiabilidade da arquitectura onde estão integrados os *Web Services* é necessário tomar medidas que garantam:

- Controlo de qualidade – Contabilizar as falhas ocorridas durante a execução do *Web Service* e verificar se o tempo de resposta do serviço está dentro do preestabelecido.
- Segurança – Controlar os acessos ao serviço e registá-los, garantir a integridade e confidencialidade dos dados, proteger contra possíveis ataques à infra-estrutura, etc.

- **Gestão** – Permite que o fornecedor do *Web Service* monitorize e controlo de estatísticas, históricos e parâmetros do serviço.

#### D. Elementos estruturantes numa arquitectura de software SOA

Para sustentar a utilização dos serviços um ambiente SOA propõe os seguintes elementos estruturantes: *Enterprise Services Bus* (ESB) (Kress, et al., 2013), *Business Process Modeling* (BPM) (Kress, et al., 2014), *Business Activity Monitoring* (BAM) (Oracle, 2015), *Rules Engine* (Linthicum, 2007) e Catálogo de serviços (Moe, 2015).

##### ○ **ESB – Enterprise Service Bus**

É uma arquitectura de *software* para *middleware* que presta serviços fundamentais para arquitecturas complexas onde inclui recursos necessários para implementar uma arquitectura orientada a serviços (SOA). De modo geral pode ser pensado como um mecanismo de gestão de acesso a aplicações e serviços para apresentar um único, simples e consistente interface ao cliente com base em formulários na *Web* ou aos utilizadores finais *front-ends*.

- Propriedades essenciais:
  - Performance e escalabilidade
  - Segurança, fiabilidade e disponibilidade
  - Distribuição de carga
  - Flexibilidade
  - Visibilidade e controlo

##### ○ **BPM – Business Process Modelling**

É a actividade de representação de processos de uma empresa, de modo a que o processo actual possa ser analisado e melhorado. Define a agregação de *workflows* com o objectivo de criar o conceito de processo de negócio. Um BPM invoca os *workflows* previamente definidos, e cria o conceito de entidade mensurável e mapeável em termos de objectivos de negócio.

O *workflow* define qual a sequência de tarefas que compõem cada processo de negócio, aloca o(s) agente(s) e recurso(s) necessário(s) à execução de cada tarefa em cada instante no tempo. As tarefas podem ser sequenciais ou concorrenciais. Podem também ter o seu início a depender do final de outras tarefas ou não.

##### ○ **BAM – Business Activity Monitoring**

BAM também é conhecido com o termo *Real-Time Business Intelligence* (ou *Real-Time Enterprise*), referindo-se a uma capacidade de analisar e reportar factos e tendências em tempo-real.

Pode ser entendido como a agregação, análise e apresentação em tempo-real das transacções dentro e fora da empresa e os seus impactos sobre os resultados de negócios, a partir dos processos de negócio, identificando situações excepcionais, de forma que estas possam ser investigadas, compreendidas, corrigidas e resolvidas. Permite tomar decisões para os negócios de uma maneira mais assertiva, relatar rapidamente áreas problemáticas envolvidas nos processos e posicionar a empresa de modo a tirar proveito de oportunidades emergentes.

BAM promove monitorização em tempo-real (ou quase tempo-real), entregando o *status* e os resultados das mais diversas operações, processos e transacções do negócio, de modo a prover informação e intervenção no nível operacional.

##### ▪ Objectivos:

- Rápida identificação de situações anómalas
- Obtenção de informação da perspectiva de negócio
- Obtenção de informação agregada em *dashboards* disponíveis a diferentes agentes da organização

##### ○ **Rules Engine**

*Rules Engine* é um sistema de *software* que executa uma ou mais regras de negócio num ambiente de produção em tempo real. Oferece a capacidade de: registar, definir, classificar e administrar todas as regras, verificar a consistência das definições de regras, definir as relações entre diferentes regras, e relacionar algumas dessas regras que são afectadas ou a necessidade de fazer cumprir uma ou mais das regras.

Permite criar uma ligação entre o mundo visual da criação de BPM com o mundo da codificação do ESB.

A criação de regras no desenvolvimento de serviços permite que através da mudança de parametrização os serviços sejam adaptados a novas realidades.

##### ○ **Catálogo de serviços**

O Catálogo de serviços é uma lista de serviços prestados por uma organização ou a um consumidor de serviços (clientes externos ou internos), detalhando o desempenho (níveis de tempo e qualidade de serviço), funcionalidade (a descrição das entradas, saídas e transformação fornecido pelo serviço), custos (e modelo de custo), e entrega do modelo (como o serviço é prestado).

O objectivo do catalogo de serviços é a visão única sobre todos os artefactos da organização, sejam serviços, relatórios, *dashboards*, informação ou mesmo o conhecimento explícito e/ou implícito.

### III. MODELO DE REFERENCIA PARA SOA - OASIS

O Modelo de Referência OASIS (*Organization for the Advancement of Structured Information Standards*) (MacKenzie, Laskey, McCabe, Brown, & Metz, 2006) é um modelo de referência para a arquitectura orientada a serviços (SOA), produzido pela OASIS em 2006.

O objectivo deste modelo de referência é incentivar o crescimento contínuo de implementações específicas, e propor um vocabulário e um entendimento comum de SOA para que os *developers* que trabalham em organizações diferentes tenham o mesmo entendimento do que os termos significam de modo a essas informações serem partilhadas e compreendidas entre aquelas ou futuras implementações.

Este modelo oferece uma referência normativa que permanece relevante para SOA como um modelo abstracto e poderoso, independente das várias e inevitáveis evoluções tecnológicas que venham a influenciar a realização de SOA.

O modelo de referência baseia-se na definição dos principais conceitos da arquitectura e a relação entre eles, tendo o serviço como elemento central.

A Figura 2 ilustra os principais conceitos definidos neste modelo de referência.

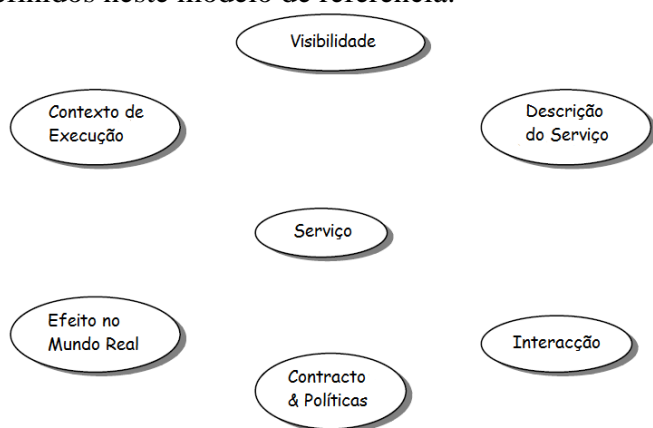


Figura 2 – Principais conceitos do Modelo de Referência OASIS (Bueno, Correa, Onoe, Borsoi, & Kiramoto, 2016).

<b>Serviço</b>	O conceito central do modelo de referência é o do serviço, que o
----------------	--

	Modelo de Referência define da seguinte forma: Um mecanismo que permite o acesso a um ou mais recursos, onde o acesso é fornecido através de uma <i>interface</i> .
<b>Visibilidade</b>	É o que habilita a interacção entre o serviço oferecido e o seu consumidor. O modelo prevê três condições para a visibilidade: <u>Consciência</u> - A consciência do serviço requer que a descrição do serviço e política estejam disponíveis de tal maneira que, directa ou indirectamente, um consumidor potencial esteja ciente da existência e das competências do serviço. <u>Concordância (Boa vontade, cooperação)</u> - Refere-se à concordância entre o fornecedor e consumidor para interagir, com base em políticas de fornecimento e utilização do serviço acordadas. <u>Acessibilidade</u> – Os participantes do serviço precisam estar aptos a interagir. Se um serviço não é acessível ele não estará efectivamente visível.
<b>Interacção</b>	Envolve a execução de acções na direcção do serviço. Geralmente é realizada na forma de mensagens, mas pode também ocorrer a partir de mudanças de estado de um recurso partilhado. Existem dois conceitos chave estão envolvidos na interacção são: <u>Modelo de informação</u> - Informação que pode ser trocada com o serviço. <u>Modelo de comportamento</u> – Refere-se a acções envolvidas no serviço e no processo ou aspectos temporais de uma interacção com o serviço. Exemplo: autorização para acesso a dados em um serviço de alteração em uma base de dados.

<b>Efeito no Mundo Real</b>	<p>Refere-se ao propósito particular relacionado com a interação com um serviço. Exemplo: Uma solicitação de uma reserva de um voo muda o estado da reserva e é um dos requisitos para habilitar uma pessoa a embarcar no avião. O utilizador não conhece os detalhes da implementação e o serviço não sabe se a reserva foi realizada pelo próprio interessado ou alguém em seu nome, mas o serviço tem efeito sobre o estado da reserva e na habilitação do passageiro.</p> <p>O resultado concreto da utilização deste serviço pode ser o retorno da informação ou da alteração no estado das entidades que estão envolvidos na interação.</p>
<b>Descrição do Serviço</b>	<p>As informações necessárias para utilizar, ou considerar o uso de um serviço. O propósito da descrição é facilitar a interação e a visibilidade do serviço.</p> <p>Em geral, deve possuir as seguintes informações:</p> <ol style="list-style-type: none"> <li>1. Que o serviço existe e está acessível (acessibilidade);</li> <li>2. Que o serviço executa certa função ou conjunto de funções (funcionalidades);</li> <li>3. Que o serviço opera sob um conjunto específico de restrições e políticas;</li> <li>4. Que o serviço irá (para alguma extensão implícita ou explícita) concordar com as políticas como prescritas pelo consumidor do serviço (políticas);</li> <li>5. Como interagir com o serviço de forma a alcançar os objectivos desejados, incluindo o formato e conteúdo da informação trocada entre o serviço e o consumidor e as sequências de</li> </ol>

	<p>informações trocadas que podem ser esperadas (<i>interface</i>).</p>
<b>Contracto &amp; Políticas</b>	<p>Uma política representa alguma restrição ou condição sobre o uso, distribuição ou descrição de uma entidade.</p> <p>Um contrato, por outro lado, representa um acordo entre duas ou mais partes.</p> <p>Políticas aplicam-se a muitos aspectos do SOA: Segurança, Privacidade, Qualidade de Serviço e assim por diante.</p> <p>É natural que a descrição do serviço contenha referências às políticas associadas com o serviço. Assim como as políticas, os contratos podem cobrir uma ampla faixa de aspectos de serviços: acordos de qualidade de serviço e acordos comerciais.</p>
<b>Contexto de Execução</b>	<p>É o conjunto de elementos de técnicos e de negócios que são identificados como parte de uma interação de serviço instanciado, que formam um caminho entre aqueles com necessidades e aqueles com competências.</p> <p>Todas as interações são baseadas num contexto de execução particular, que permite que os prestadores de serviços e os consumidores interajam e forneçam um ponto de decisão para que todas as políticas e contratos possam estar em vigor.</p> <p>O consumidor e o prestador de serviços podem estar em locais distintos no mapa e, para um serviço ser realmente invocado, um caminho precisa ser estabelecido entre estes dois locais. Este caminho é o contexto de execução.</p>

Tabela 1 - Quadro resumo dos principais conceitos do Modelo de Referência OASIS.

O modelo de referência OASIS propõe ainda um

guia para estabelecer a conformidade com a arquitectura SOA, indicando que as seguintes características devem estar presentes:

1. Ter entidades que podem ser identificadas como serviços e definidas no Modelo de Referência;
2. Estar apta a identificar como a visibilidade é estabelecida entre os prestadores e consumidores de serviço;
3. Estar apta a identificar como a interacção será mediada;
4. Estar apta a identificar como os efeitos do uso de serviços são compreendidos;
5. Ter descrições associadas com serviços;
6. Estar apta a identificar o contexto de execução requerido para suportar interacções;
7. Ser possível identificar como as políticas são tratadas e como os contractos podem ser modelados e formatados.

#### **IV. FERRAMENTAS AO DISPOR DAS ORGANIZAÇÕES PARA IMPLEMENTAÇÃO DE PROJECTOS COM A ARQUITECTURA SOA**

No processo de desenvolvimento de *software*, a escolha das ferramentas a utilizar tem uma grande importância. As ferramentas poderão auxiliar o trabalho em todas as fases do desenvolvimento, análise desenvolvimento, testes, instalação e manutenção.

Na área do SOA, conhecer e escolher as ferramentas que irão auxiliar o desenvolvimento também irá ter um grande peso no sucesso dos projectos. Quais as ferramentas que existem? Quais as funcionalidades que oferecem? Qual o grau de dificuldade na aprendizagem do funcionamento de uma ferramenta? De que maneiras irão auxiliar o desenvolvimento? Estas perguntas deverão ser respondidas para que se possa escolher as ferramentas correctas.

Por vezes a adopção de uma ferramenta irá implicar o seguimento de determinada metodologia de desenvolvimento. Quando se escolhe as ferramentas também deveremos verificar se esta se adapta às metodologias de desenvolvimento seguidas ou se implica uma nova metodologia. No caso de implicar uma nova metodologia, é necessário decidir se essa metodologia aplica-se ao problema em questão. Para o desenvolvimento de projectos que adoptam uma arquitectura SOA, existem várias

ferramentas que poderão ser adoptadas. Qual a melhor irá depender muito do tipo de projectos e do tipo da equipa de desenvolvimento.

A título de exemplo, neste trabalho iremos apresentar um conjunto de ferramentas da IBM que auxiliam o desenvolvimento de projectos que utilizam arquitectura SOA e que baseiam-se na modelação de processos de negócio. Outros fornecedores tais como a Oracle, TIBCO ou Microsoft possuem ferramentas que auxiliam o desenvolvimento de projectos utilizando a arquitectura SOA.

##### **o IBM WebSphere Business Integration**

IBM WebSphere Business Integration (Jessani, Chilanti, & Iyengar, 2007) é um produto que integra vários subprodutos que permitem a modelação, o desenvolvimento, a instalação e a execução de processos de negócio para *web services*. A utilização deste produto irá ser muito útil para o desenvolvimento de soluções SOA. Também é uma boa escolha quando se pretende seguir uma metodologia de desenvolvimento “*On Demand*”. De seguida apresentamos cada um dos subprodutos que constituem este produto.

As ferramentas de integração de negócios são construídas para suportar uma arquitectura orientada a serviços, o que diminui a complexidade no desenvolvimento porque representa todos os objectos, componentes, aplicações e processos, como serviços (baseados em padrões abertos). Estes serviços são aplicações *self-contained*, e, modulares que podem ser publicadas, localizadas e invocadas através da Internet, automatizando uma ampla gama de funções de negócio, incluindo activos existentes. Usando este tipo de arquitectura, é possível criar e interagir com todos os componentes de software, sem considerar sua implementação.

##### **o IBM WebSphere Business Integration Modeler**

IBM WebSphere Business Integration Modeler (Jessani, Chilanti, & Iyengar, 2007) é uma ferramenta de modelação que permite a análise do negócio no formato BPMN desenvolvida pela IBM. Com esta ferramenta é possível fazer a modelação, simulação e análise de modo a que os analistas de negócio possam conceber, documentar e implementar processos de negócio. Pode ser utilizada para:

- Documentar, analisar e otimizar os processos de negócio;
- Proporcionar um início de desenvolvimento rápido dos processos de negócio otimizados;
- Ajudar às organizações clientes a atingir um processo de negócio o mais competitivo possível;
- Permite às organizações avaliar o retorno sobre o investimento (ROI);

Esta ferramenta é baseada e utiliza na plataforma de desenvolvimento Eclipse. Consegue capturar modelos de negócio e consegue gerar representações dos modelos na linguagem BPEL4WS (*Business Process Execution Language for Web Services*) e FDL (*Flow Definition Language*). O WebSphere Business Integration Modeler oferece um ambiente de trabalho de fácil compreensão, de fácil utilização (“*user-friendly*”) para a modelação de processos de negócio. Oferece também capacidades de colaboração para desenhar os processos de negócios. Suporta múltiplas metodologias de modelação e vários *standards*. Permite redesenhar rapidamente o processo quando existem alterações no negócio. Tem funcionalidades que permitem importar modelos e que permitem publicar os modelos na *Web*.

Dentro de uma empresa esta ferramenta oferece um ambiente de colaboração para partilha e manutenção das versões dos modelos. Os analistas de negócio poderão utilizar as capacidades de otimizar os processos de negócio através de simulações de várias hipóteses para os modelos. Para cada uma destas hipóteses poderá criar relatórios para futuras análises.

- **IBM WebSphere Studio Application Developer Integration Edition**

O WebSphere Studio Application Developer Integration Edition (Sadler & Kovari, 2004) fornece o suporte necessário para o desenvolvimento e teste dos processos de negócio com recurso a um editor gráfico para montar os processos de negócio definidos na linguagem BPEL4WS, principalmente através de suas ferramentas de integração de negócios.

Esta ferramenta incorpora o que um programador necessita para construir processos de negócio nomeadamente um editor visual de processos que pode ser utilizado para criar ou modificar processos de negócio.

Este editor suporta as linguagens de modelação de negócio BPEL4WS e FDL. Inclui a hipótese de migração entre estas duas linguagens. Consegue gerar todo o código necessário para a execução dos processos. Um processo poderá ser criado para ser executado como um “EJB bean”, como “*message driven bean*” ou como um “*web service*”. Os “*web services*” poderão ser criados para correrem sobre SOAP/HTTP ou SOAP/JMS.

- **WebSphere MQ Workflow**

O WebSphere MQ Workflow (Lind & Schwenkreis, 2017) é um produto destinado a ajudar as organizações a automatizar os seus processos de negócio e a fornecer processos de integração com suporte avançado.

O componente *Buildtime* é a ferramenta que define o processo gráfico e faz parte do produto WebSphere MQ Workflow. Com a ferramenta *Buildtime* é possível definir graficamente os processos de negócio e as suas actividades ao nível de detalhe necessário para a sua automatização, onde os processos de negócio definidos podem ser exportados pela ferramenta *Buildtime* em arquivos de FDL (*Flow Definition Language*).

Como parte do WebSphere Business Integration, o WebSphere MQ Workflow também permite o uso com o WebSphere Business Integration Modeler para projectos de análise, simulação e acompanhamento dos processos de negócio.

- **WebSphere Business Integration Server Foundation**

Esta ferramenta (IBM, 2004) fornece a base para a construção e implementação orientada a serviços e que permite a criação de uma plataforma para criar e instalar aplicações compostas por “*web services*” numa arquitectura SOA.

- Combina serviços de processos com serviços de aplicações;
- Permite a criação, a execução e gestão de processos de negócio;
- Com esta ferramenta é possível criar novos serviços e serviços com base em aplicações e sistemas, que podem ser combinadas com aplicações existentes no processo de negócio;

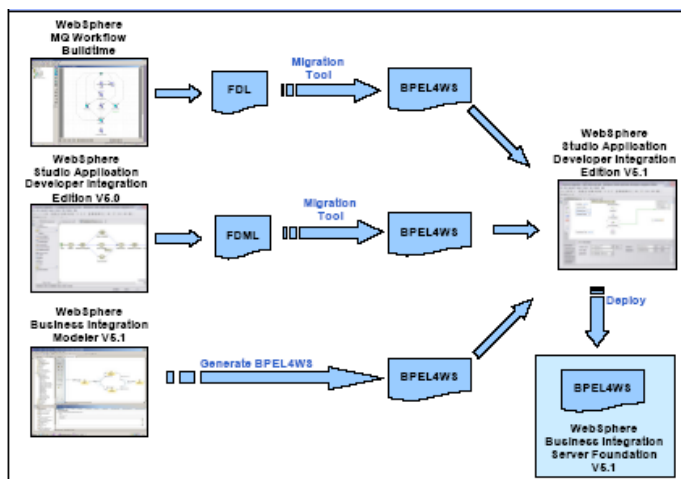


Figura 3. WebSphere Business Integration, Desenvolvimento, instalação e monitorização (Keen, et al., 2004)

## V. BENEFÍCIOS E DESAFIOS DE UMA IMPLEMENTAÇÃO SOA

O benefício de uma implementação SOA inclui a redução de custos de integração onde é proporcionada melhor agilidade e flexibilidade negócios, reutilização de activos e maior retorno sobre o investimento.

A mudança para uma arquitectura SOA tem sido facilitada pelo nível sem precedentes de padronização em torno de normas fundamentais de serviços Web: XML, SOAP, WSDL e UDDI e vários Serviços Web (*Web Services*) que incorporam serviços de segurança, confiabilidade, transacções, coordenação e gestão.

### A. Benefícios na adopção de SOA

Uma arquitectura orientada a serviços pode ajudar as organizações a atingir um nível de flexibilidade que lhes permita acompanhar os rápidos desenvolvimentos dos mercados e rentabilizar os seus recursos de forma a reduzir os custos.

A adopção de uma arquitectura SOA permite a obtenção de um conjunto de benefícios:

- **Facilidade de integração**

É mais fácil integrar aplicações existentes através de protocolos tal como *Web Services*. Ao adoptar este método em vez de adquirir novas soluções diminui não só os custos no investimento, mas também mitiga os riscos da mudança.

- **Redução de custos e aumento da reutilização - Melhoria no ROI (Retorno do Investimento) dos projectos**

A redução dos custos é feita principalmente através de reutilização de código e funcionalidades onde estes podem ser facilmente reutilizados e combinados de forma a satisfazer as necessidades de negócio. Isto significa menor duplicação de recursos, maior potencial de reutilização e menores custos de desenvolvimento e manutenção.

As arquitecturas SOA também potencializam a automatização de processos através de programas de *Workflow*, reduzindo assim os custos com o pessoal.

Desta forma o ROI melhora, como resultado de uma melhor integração e agilidade dos sistemas em adaptarem-se ao negócio.

- **Integração e comunicação de sistemas com parceiros e clientes**

Com a arquitectura SOA é possível fazer a interligação dos sistemas com a utilização de *de simple object access protocol* (SOAP) e *Web services description language* (WSDL) através de HTTP.

- **Reaproveitamento de recursos**

As arquitecturas SOA fornecem uma camada de abstracção que permite às empresas aproveitar o investimento feito anteriormente em tecnologias de informação, envolvendo os recursos existentes como serviços que disponibilizam funcionalidades de negócio.

Desta forma as empresas podem continuar a retirar valor desses recursos sem que seja necessário recomeçar do início.

- **Maior capacidade de reacção**

Capacidade de criar novos serviços a partir de serviços já existentes é uma vantagem para as empresas que necessitam de ter agilidade e de responder rapidamente as exigências do mercado. A reutilização de componentes faz com que o tempo de desenvolvimento do *Software* necessário seja reduzido, o que leva a uma rápida criação de novos serviços que implementam novas funcionalidades de negócio.

Desta forma o tempo que os produtos levam a atingir o mercado é reduzido e a empresa adapta-se mais rapidamente as mudanças. Por outro lado, melhora também a capacidade de tomada de decisão pois os dados são partilhados entre todos os serviços,

o que por sua vez melhora a capacidade de reacção da empresa as alterações do mercado ou as oportunidades de negócio.

#### ▪ **Preparação para o futuro**

As arquitecturas SOA permitem às empresas estarem preparadas para o futuro. Processos que envolvam uma série de serviços relativos ao negócio podem ser criados, modificados e mantidos facilmente ao longo do tempo, correspondendo assim às necessidades da empresa. Uma SOA fornece flexibilidade e adaptabilidade, características que são críticas para quase todos os negócios.

Em resumo, os benefícios da arquitectura SOA recaem em três áreas:

- Eficácia do negócio
  - Agilidade, capacidade de resposta ao mercado e dinâmica competitiva;
  - Maior eficiência do processo;
  - Implementação de recursos com base nas necessidades de negócios;
- Eficiência de custo
  - Custos de manutenção reduzido;
  - Esforços reduzidos para mudança de negócios;
  - Optimização preço/desempenho com base na liberdade para seleccionar plataforma, tecnologia e localização de forma independente;
- Redução do risco
  - Maior nível de qualidade e produtividade das equipas de desenvolvimento;
  - Implementação incremental;
  - Retorno rápido do investimento;

#### B. *Quais são os desafios?*

Deve-se ressaltar as normas técnicas XML, SOAP, WSDL, etc.. sozinhas não garantem que os benefícios de negócios da SOA serão plenamente realizados. Igualmente, uma implementação de padrões de serviços Web não constitui uma arquitectura orientada a serviço. A adopção de padrões de tecnologia é uma condição prévia necessária para alcançar o baixo custo integração entre plataformas tecnológicas, mas outras questões que incluem análise de serviço, *design* e integração entre as estruturas de dados em toda a organização (ou toda a indústria) é igualmente importante.

A transição para uma estrutura de IT tradicional para SOA apresenta uma série de desafios importantes. Esses desafios que vão desde questões puramente técnicas tais como desempenho do protocolo SOAP e os padrões de *Web Services* que incluem considerações de disponibilidade e custos de infra-estrutura SOA.

Enquanto o núcleo dos *Web services* (ou seja, XML, SOAP, WSDL e UDDI) são relativamente maduros e estáveis, muitos dos padrões adicionais têm importantes questões como segurança e confiabilidade (por exemplo, WS-Coordination, WS-Atomic Transaction, WSDM, WS-Reliability, etc.).

Para obter os benefícios da reutilização e capacidade de resposta às novas necessidades de negócio, os serviços devem ser especificados ao nível da abstracção e granularidade. Os serviços não devem ser apenas refinados e com interfaces bem-definidas, sendo necessária uma metodologia de projecto abrangente que orienta o processo de *design* de serviço e produz serviços reutilizáveis que possam ser utilizados como blocos de construção. Os desafios e limitações podem ser resumidos como se segue:

- Padrões de *Web Services*: estes padrões são abertos e muitos ainda estão rascunhos. Segurança e serviços de nível mais alto ainda não foram testados, isso pode resultar numa ameaça em conformidade com as normas novas e em evolução.
- Protocolos: eles não são projectados para a confiabilidade, entrega garantida ou ordem de entrega para o serviço sendo necessário garantir que as mensagens foram entregues ou recebidas em tempo útil.
- Interoperabilidade: As implementações de SOA usam serviços web para trocar mensagens pela Internet. Estas encapsulam os dados em XML. Problemas podem ser encontrados ao integrar estes serviços em ambientes heterogéneos.
- Ligações de rede: Se a arquitectura e os serviços são altamente interligados, uma falha de rede parcial pode criar um problema enorme para as organizações. Actualmente, embora a tecnologia necessária esteja

disponível, não há nenhuma garantia de que as infra-estruturas serão robustas o suficiente com redundância nas suas ligações para lidar com os tempos de inactividade do sistema.

- **Segurança:** Quando estão a ser utilizados padrões abertos, um serviço é muito mais aberto a outros serviços e aplicativos do que uma aplicação típica e, assim, a segurança torna-se um problema. Como mencionado antes, protocolos de Internet também faltam. Embora a WS-Security aborde essas questões, há uma considerável quantidade de trabalho que ainda precisa ser feito.
- **Formação:** Leva tempo adquirir conhecimento de novas tecnologias. Para uma organização adoptar esta arquitectura, sem o conhecimento aprofundado das tecnologias subjacentes pode ser altamente crítico (Hohpe, 2002).

Mudar para uma arquitectura SOA não é uma transição transversal. Requer uma mudança de como podemos ter aplicações baseadas em serviços maximizando o investimento em TI existente (Mahmoud, 2005). SOA requer a construção de sistemas a nível empresarial, já a prestação de serviços deve centrar-se sobre os requisitos de negócio. Uma vez que os processos de negócio e as estruturas sejam definidas, se pode pensar numa tecnologia necessária para construir uma SOA totalmente operacional. Deve ser um desenvolvimento incremental.

## VI. CONCLUSÕES

No mercado empresarial é cada vez mais importante fornecer produtos de qualidade, o que leva à especialização das empresas. Por sua vez a especialização das empresas obriga à cooperação entre elas para que sejam criados novos produtos. É aqui que as arquitecturas orientadas a serviços podem fazer a diferença pois fornecem a flexibilidade e a agilidade necessárias para que exista interoperabilidade efectiva entre as empresas.

As arquitecturas orientadas a serviços têm como base principal a interoperabilidade entre componentes de *Software* e os *Web Services* são vistos como o meio preferencial para os obter. Os *Web Services* levam à interoperabilidade porque

escondem a implementação dos serviços por trás de interfaces, deixando então de ter qualquer importância em que linguagem está implementada o serviço ou em que plataforma está a correr.

Outra perspectiva de futuro trazida pelas SOAs é que com o desenvolvimento de *standards* todas as aplicações de um determinado tipo respeitarão uma determinada interface, tornando-se assim possível trocar de fornecedor de aplicações sem que seja necessário fazer alterações no sistema. Por outro lado com o aumento do número de serviços disponíveis, tanto dentro como fora das empresas, a construção de aplicações e a modelação de processos de negócio tornar-se-á em pouco mais do que a agregação de serviços existentes.

Hoje em dia já muitas empresas optaram por reformular os seus sistemas implementado arquitecturas orientadas a serviços e estão a tirar grandes benefícios da sua utilização. Alguns benefícios, como a capacidade de interoperar com outros sistemas, eram esperados mas outros, como por exemplo o aumento da fiabilidade, são uma surpresa agradável.

Tendo em conta que relação entre custos e benefícios é tão boa, é bastante provável que as arquitecturas orientadas a serviços estejam aqui para ficar.

## VII. REFERÊNCIAS BIBLIOGRÁFICAS

- Barry, D. (2014). *Service-Oriented Architecture (SOA) Definition*. Obtido de [http://www.service-architecture.com/articles/web-services/service-oriented\\_architecture\\_soa\\_definition.html](http://www.service-architecture.com/articles/web-services/service-oriented_architecture_soa_definition.html)
- Bueno, J., Correa, P., Onoe, A., Borsoi, B., & Kiramoto, T. (19 de Junho de 2016). OASIS. *Modelo de Referência para Arquitetura*.
- Christensen, E., Curbeta, F., Meredith, G., & Weerawarana. (Março de 2001). *Web Services Description Language (WSDL)*. Obtido de <http://www.w3.org/TR/wsdl>
- Hohpe, G. (Maio de 2002). *Stairway to Heaven, Software Development*.
- IBM. (Maio de 2004). *IBM WebSphere Business Integration Server*. Obtido de [https://www-01.ibm.com/common/ssi/rep\\_ca/4/897/ENUS204-114/ENUS204-114.PDF](https://www-01.ibm.com/common/ssi/rep_ca/4/897/ENUS204-114/ENUS204-114.PDF)
- Jessani, V., Chilanti, M., & Iyengar, A. (2007). *WebSphere Business Integration Primer: Process Server, BPEL, SCA, and SOA*. IBM.
- Keen, M., Cavell, J., Hill, S., Kee, C. K., Neave, W., Rumph, B., & Tran, H. (2004). *BPEL4WS Business Processes with WebSphere Business Integration: Understanding, Modeling, Migrating*. IBM.

- Kress, J., Maier, B., Normann, H., Schmeidel, D., Schmutz, G., Trops, B., & Winterberg, T. (Julho de 2013). *Enterprise Service Bus*. Obtido de <http://www.oracle.com/technetwork/articles/soa/ind-soa-esb-1967705.html>
- Kress, J., Maier, B., Normann, H., Schmeidel, D., Schmutz, G., Trops, B., & Winterberg, T. (Fevereiro de 2014). *SOA and Business Processes: You are the Process!* Obtido de <http://www.oracle.com/technetwork/articles/soa/ind-soa-bpm-2133403.html>
- Lind, K., & Schwenkreis, F. (Janeiro de 2017). *Understanding how WebSphere MQ Workflow and WebSphere Application Server Enterprise Process Choreographer can work together*. Obtido de [https://www.ibm.com/developerworks/websphere/library/techarticles/wasid/WPC\\_Interop/WPC\\_Interop.html](https://www.ibm.com/developerworks/websphere/library/techarticles/wasid/WPC_Interop/WPC_Interop.html)
- Linthicum, D. (Fevereiro de 2007). *Rules Engines and SOA*. Obtido de <http://www.infoworld.com/article/2641011/service-oriented-architecture/rules-engines-and-soa.html>
- MacKenzie, M., Laskey, K., McCabe, F., Brown, P., & Metz, R. (Outubro de 2006). *Reference Model for Service Oriented Architecture 1.0*. Obtido de <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>
- Mahmoud, H. (2005). *Service-oriented architecture and web services: the road to enterprise application integration*. Sun Microsystems Inc.
- Moe, J. (2015). *SOA and Service Catalogues*. Obtido de <http://www.soainstitute.org/resources/articles/soa-and-service-catalogues>
- OASIS. (2014). *OASIS UDDI Specification*. Obtido de <https://www.oasis-open.org/committees/uddi-spec/faq.php>
- Oracle. (2015). *Oracle Business Activity Monitoring*. Obtido de <http://www.oracle.com/technetwork/middleware/bam/overview/index.html>
- Pinho, S. (2008). *Arquitetura de Segurança num ambiente SOA (Service Oriented Architecture)*. Porto.
- Sadtler, C., & Kovari, P. (2004). *WebSphere Business Integration Server Foundation Architecture and Overview*. IBM.
- Sadtler, C., & Kovari, P. (2004). *WebSphere Business Integration Server Foundation Architecture and Overview*. IBM.
- Sadtler, C., Cuomo, G., Ganci, J., Haberkorn, M., Jones, C., Kovari, P., . . . Will, R. (2005). *WebSphere Product Family Overview and Architecture*. IBM.
- Thompson, H., Mendelsohn, N., Beech, D., & M., M. (Abril de 2012). *W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures*. Obtido de <http://www.w3.org/TR/xmlschema11-1/>
- W3C. (2002). *Web Services Activity*. Obtido de <http://www.w3.org/2002/ws/>
- W3C. (Junho de 2014). *HTTP - Hypertext Transfer Protocol*. Obtido de <http://www.w3.org/Protocols/>
- W3Schools. (2012). *ML Soap*. Obtido de [http://www.w3schools.com/xml/xml\\_soap.asp](http://www.w3schools.com/xml/xml_soap.asp)
- W3Schools. (2012). *XML Web Services*. Obtido de [http://www.w3schools.com/xml/xml\\_services.asp](http://www.w3schools.com/xml/xml_services.asp)