



Java<sup>TM</sup>  
ORACLE®



Learn once, code anywhere.

Fica expressamente proibido a reprodução ou utilização deste material sem a devida permissão ou consentimento do autor.

Contato: **fernandofranzini@gmail.com**

**FOR-J © – Copyright 2017 – Todos os direitos reservados.**

# Introdução – Web Services



Java™

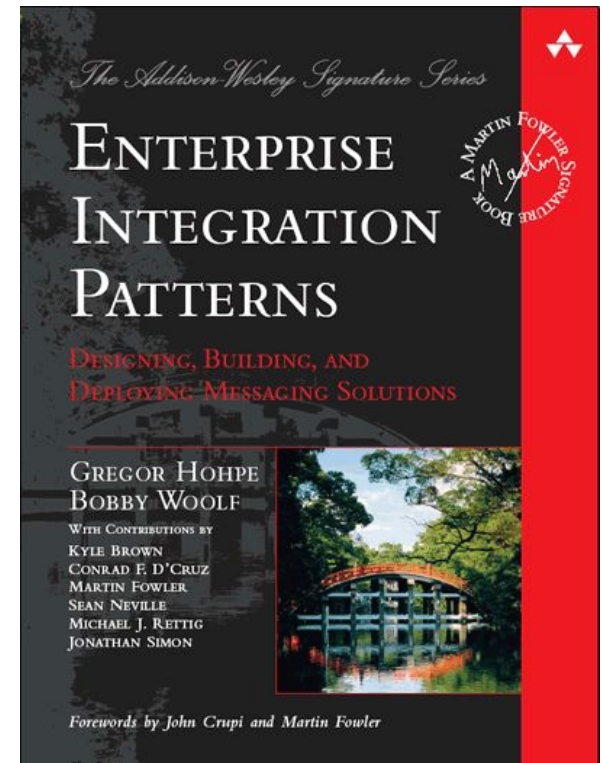
# O que é Web Service?

- ▶ O que é?
- ▶ Para que serve?
- ▶ Resolve que problema?
- ▶ Qual é o contexto?
- ▶ Qual é aplicabilidade?

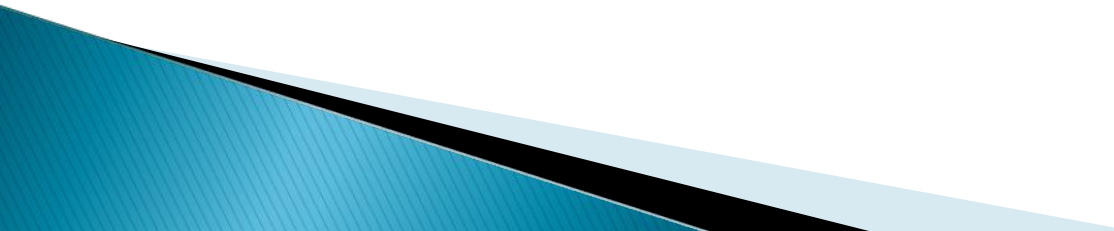


# 1) O que é Web Service?

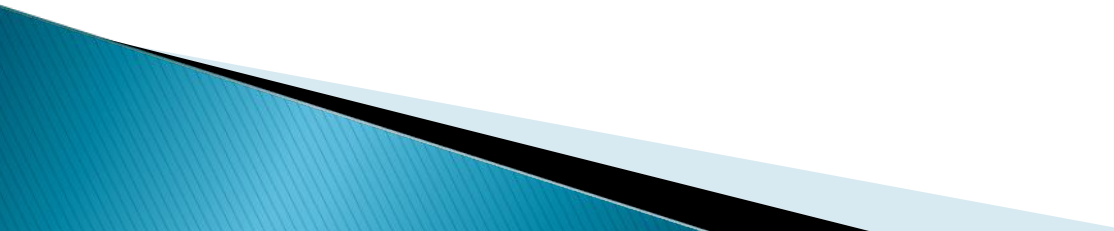
- ▶ O livro “Enterprise Integration Patterns” classifica os web services como um estratégia de integração de soluções, no qual um sistema necessita acessar dados ou executar operações de outro sistema.



# 1) O que é Web Service?

- ▶ **Solução** utilizada na integração de sistemas e na comunicação entre independentes aplicações.
  - ▶ **Tecnologia** que possibilita a integração entre aplicações de forma que elas possam interagir entre si independentemente de plataforma, linguagem, paradigma de programação e fornecedor de tecnologia.
- 

# 1) O que é Web Service?

- ▶ São componentes que permitem às aplicações enviar e receber informações baseados em um formato pré estabelecido.
  - ▶ Um serviço web é uma aplicação disponibilizada via (HTTP) intranet ou internet que através de uma URL pode ser acessado por aplicativos clientes usando protocolos acordados.
- 

## 2) Para que serve Web Service?

- ▶ Único e exclusivo objeto é:

**Interoperabilidade entre sistemas independentemente de plataforma de desenvolvimento, execução, provedor de tecnologia, linguagem e paradigma de desenvolvimento.**



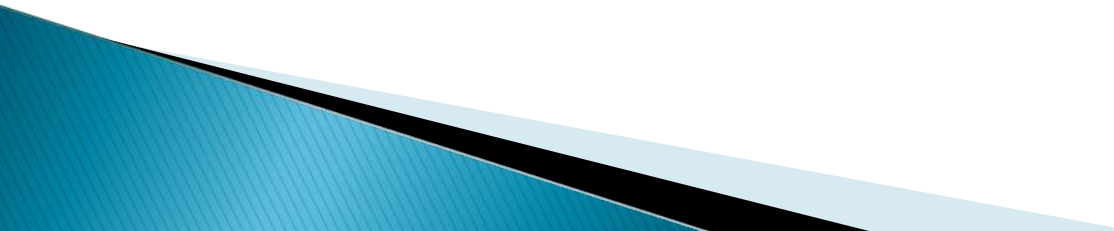
## 2) Para que serve Web Service?

- ▶ Interoperabilidade é a capacidade de um sistema informatizado ou não de se comunicar de forma transparente ou o mais próximo disso com outro sistema semelhante ou não.

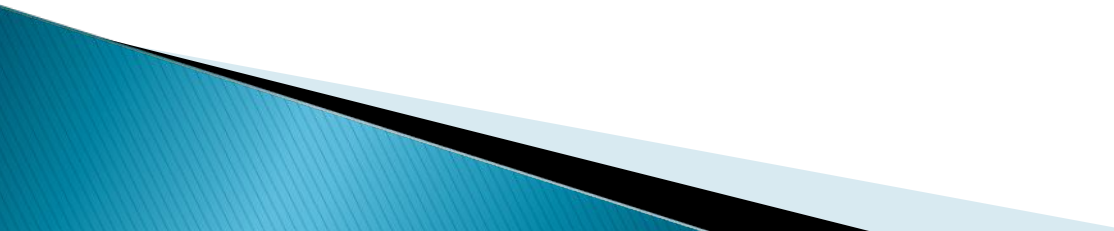
### 3) Quando usar Web Services?

- ▶ Aplicações corporativas abrangem uma série de cenários simples até os mais complexos que apresentam requisitos de interoperabilidade entre soluções corporativas dentro de um ambiente empresarial heterogêneo.

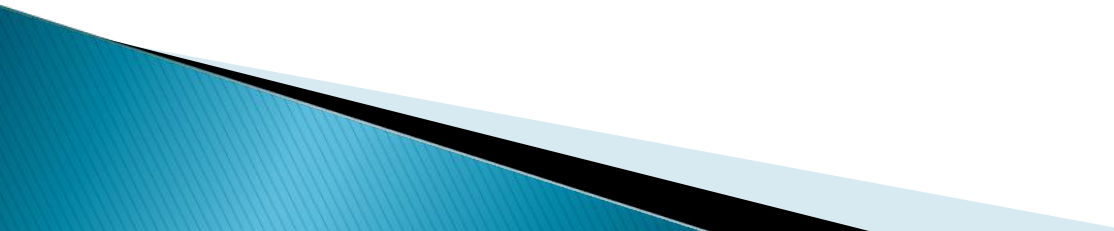
# Cenário de Integração 1

- ▶ Grandes, médias e pequenas empresas possuem investimento significativos e crescentes em sistemas pertencentes a diferentes plataformas ao longo de sua história.
  - ▶ Web services é a solução para fazer a integração entre o legado e as novas aplicações.
- 

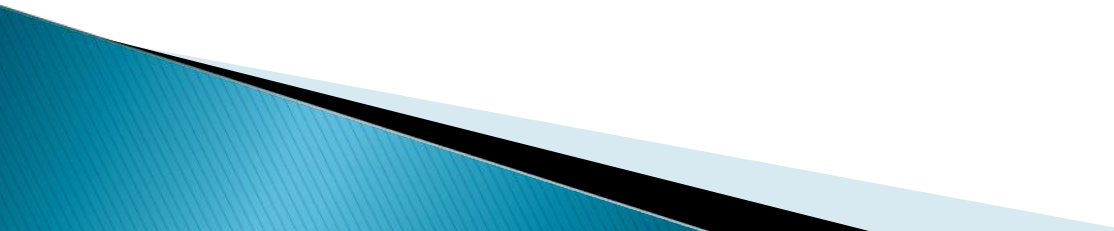
# Cenário de Integração 2

- ▶ Na atualidade temos empresas adquirindo outras empresas e cada uma delas apresentando soluções próprias em diferentes plataformas.
  - ▶ Web services é a solução para fazer a integração entre estas aplicações até então desconhecidas.
- 

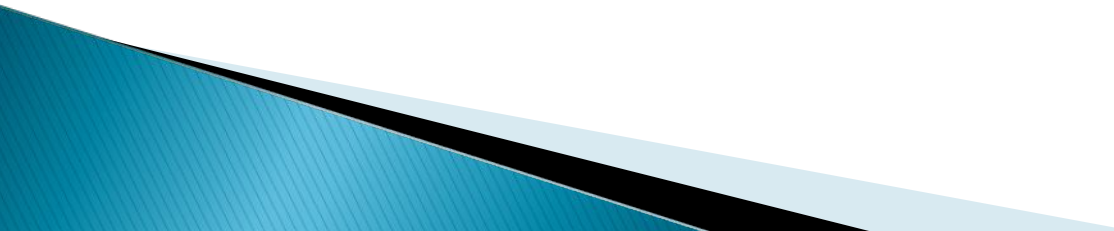
# Cenário de Integração 3

- ▶ Muitas empresas têm expandido suas redes de negócios no estabelecimento de “parcerias” com outras empresas “sócios” de outros setores que unidas formam uma nova vertente de negócio.
  - ▶ Web services é a solução para fazer a integração entre estas aplicações de cada empresa, garantindo eficiência no negócio e melhor experiência para o usuário final.
- 

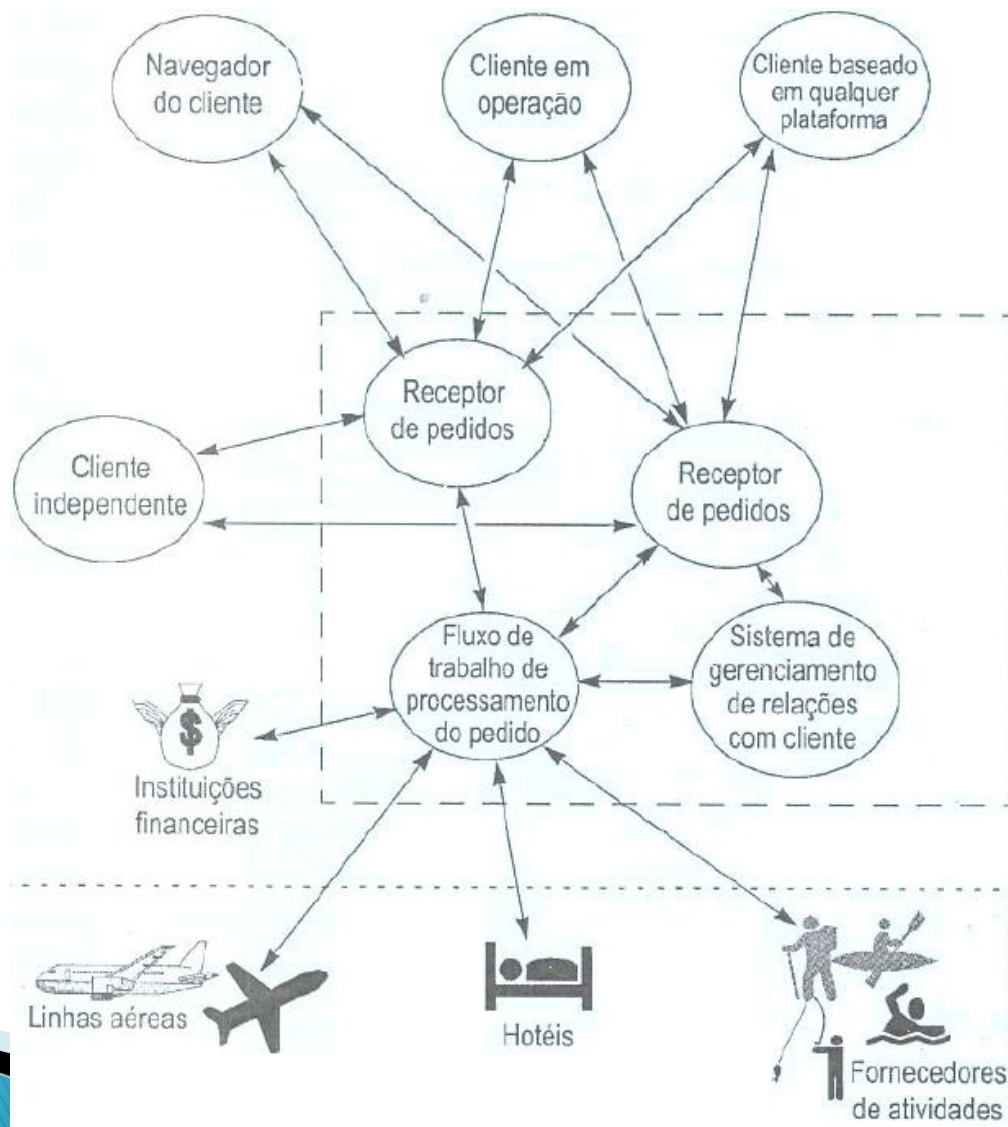
# Cenário de Integração 4

- ▶ Muitas empresas têm a necessidade de expor suas soluções para ser acessadas remotamente em diferentes plataformas de execução (smartphones, PDA, tablet, TV etc) que são incompatíveis com a tecnologia adotadas.
  - ▶ Web services é a solução utilizada para integrar aplicativos incompatíveis.
- 

# Cenário Real – Web Service

- ▶ **Agência de Turismo:**
  - ▶ Fornece catálogo de turismo.
  - ▶ Construir passeios customizados de acordo com o perfil dos clientes.
  - ▶ Rastrear status de pedidos.
  - ▶ Reservar passeios.
  
  - ▶ Todas as operações necessitam Interagir com sistemas de outros sócios – linhas aéreas, hotéis, instrutores de passeios, instituições financeiras etc.
- 

# Cenário Real – Web Service





# Dúvidas e comentários

- ▶ Dúvidas ou comentários, entre em contato pelos canais de comunicação.



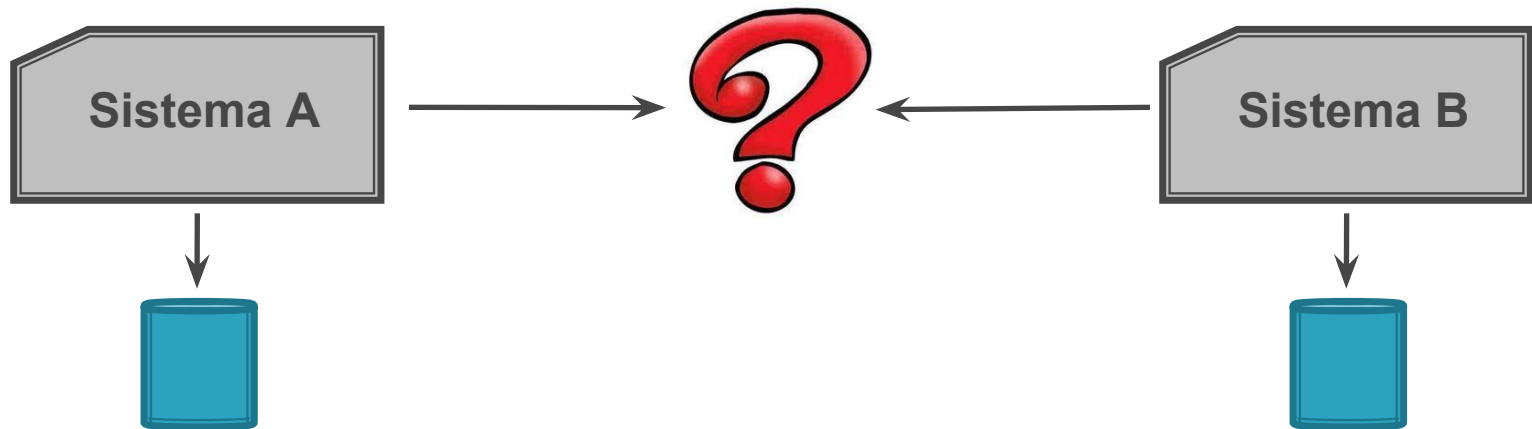
# Arquitetura da Solução – Web Service



Java™

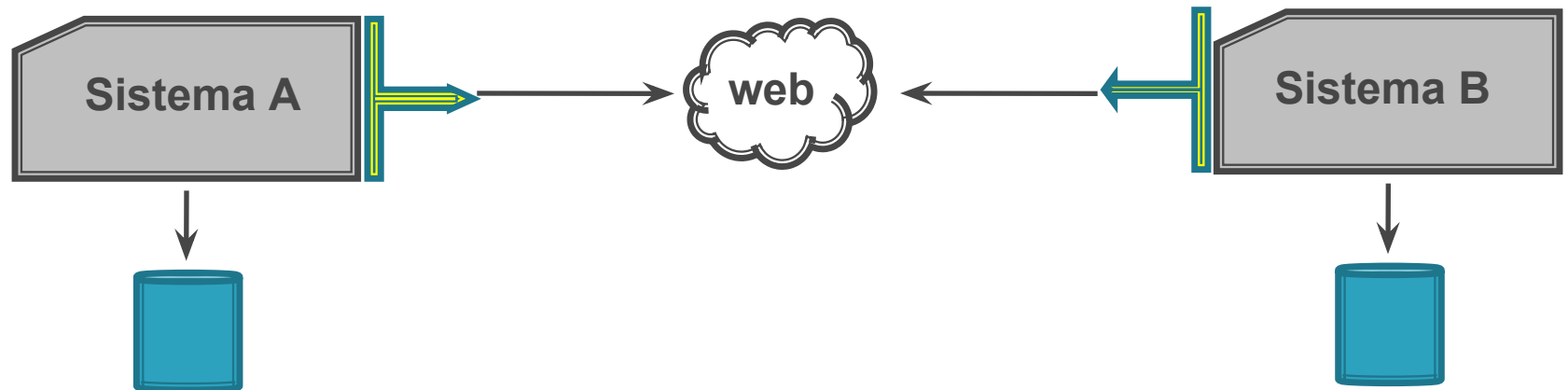
# Arquitetura – Web Service

- ▶ Surge a necessidade de integrar 2 ou mais soluções diferentes dentro de uma mesma corporação:



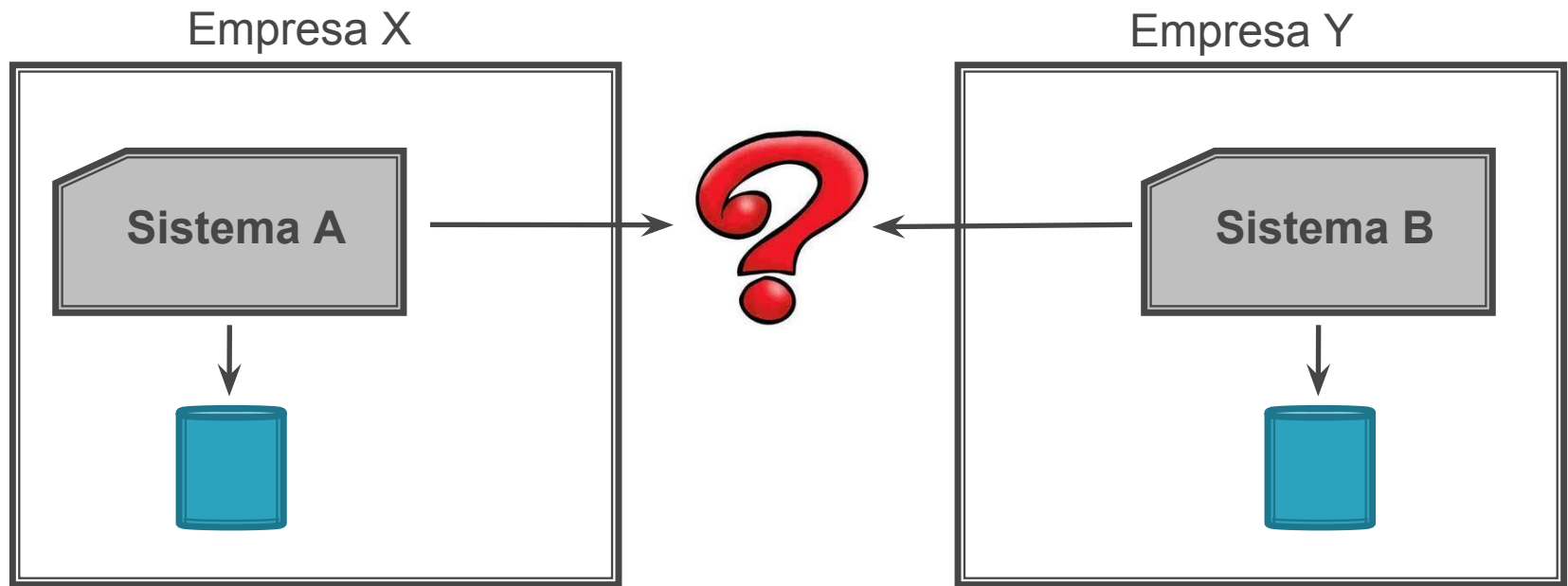
# Arquitetura – Web Service

- ▶ É acrescentado em cada solução uma “camada de comunicação” que expõe as operações existentes como serviço:



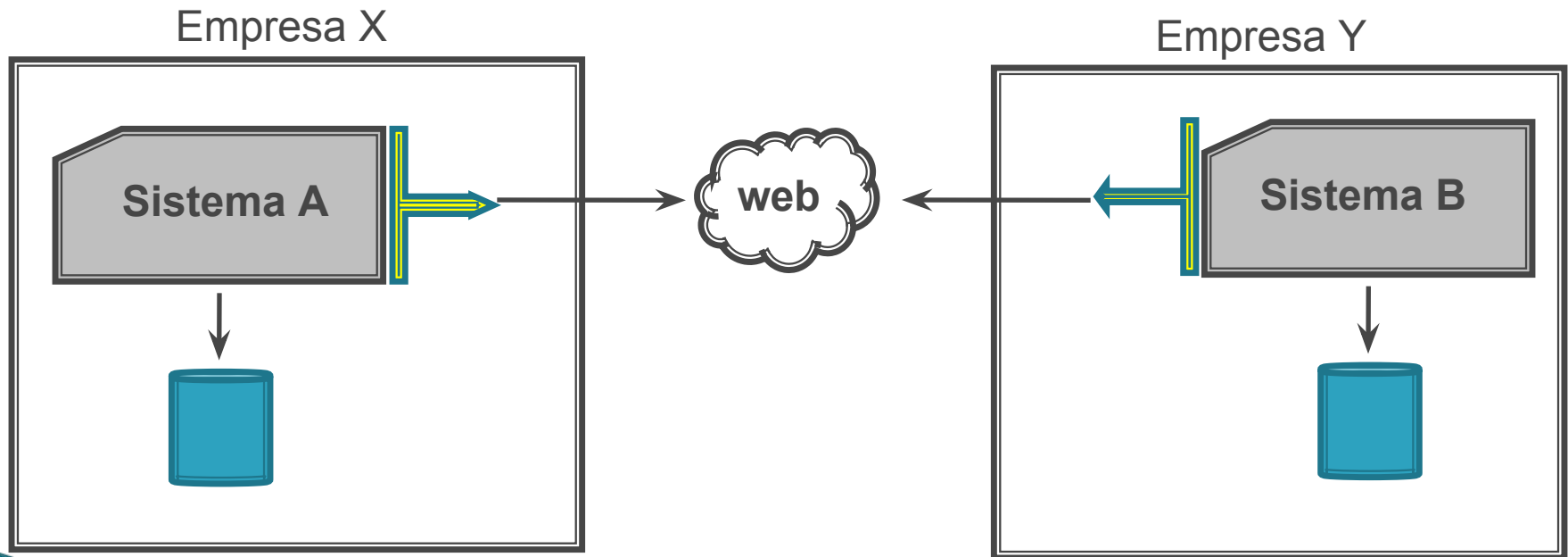
# Arquitetura – Web Service

- ▶ Surge a necessidade de integrar duas ou mais soluções de diferentes corporações:



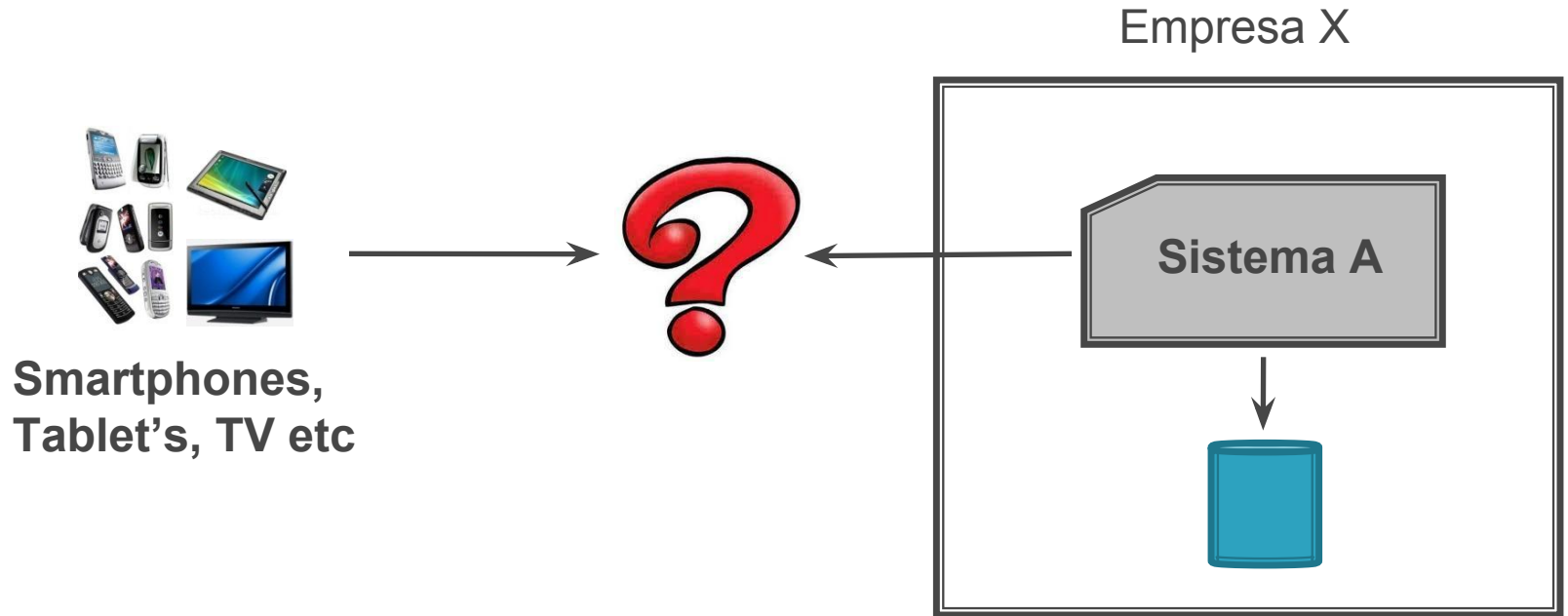
# Arquitetura – Web Service

- ▶ É acrescentado nas soluções uma “camada de comunicação” que expõe as operações existentes como serviço:



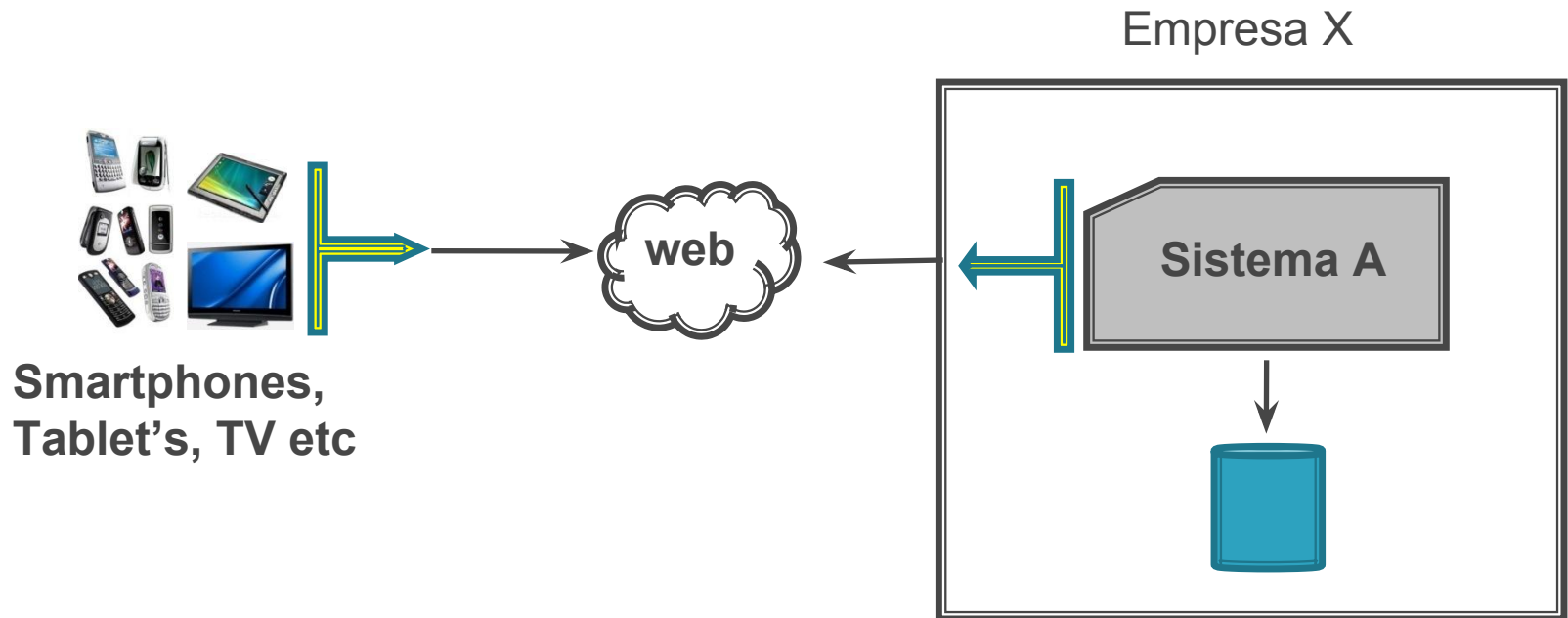
# Arquitetura – Web Service

- ▶ Surge a necessidade de integrar soluções de dispositivos móveis em uma solução corporativa:



# Arquitetura – Web Service

- ▶ É acrescentado na solução da corporação e nas soluções nativas de cada dispositivos uma “camada de comunicação”:



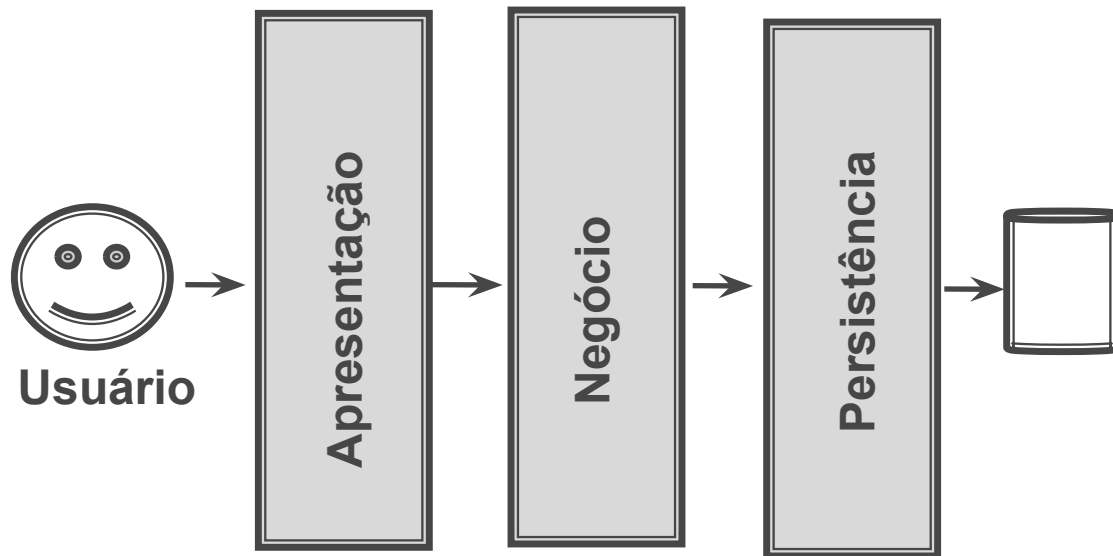


# Arquitetura – Web Service

- ▶ A aplicação que expõe funcionalidades via web service é chamada de “**serviço**”.
- ▶ A aplicação que invoca as funcionalidade de um serviço é chamada de “**consumidor**”.

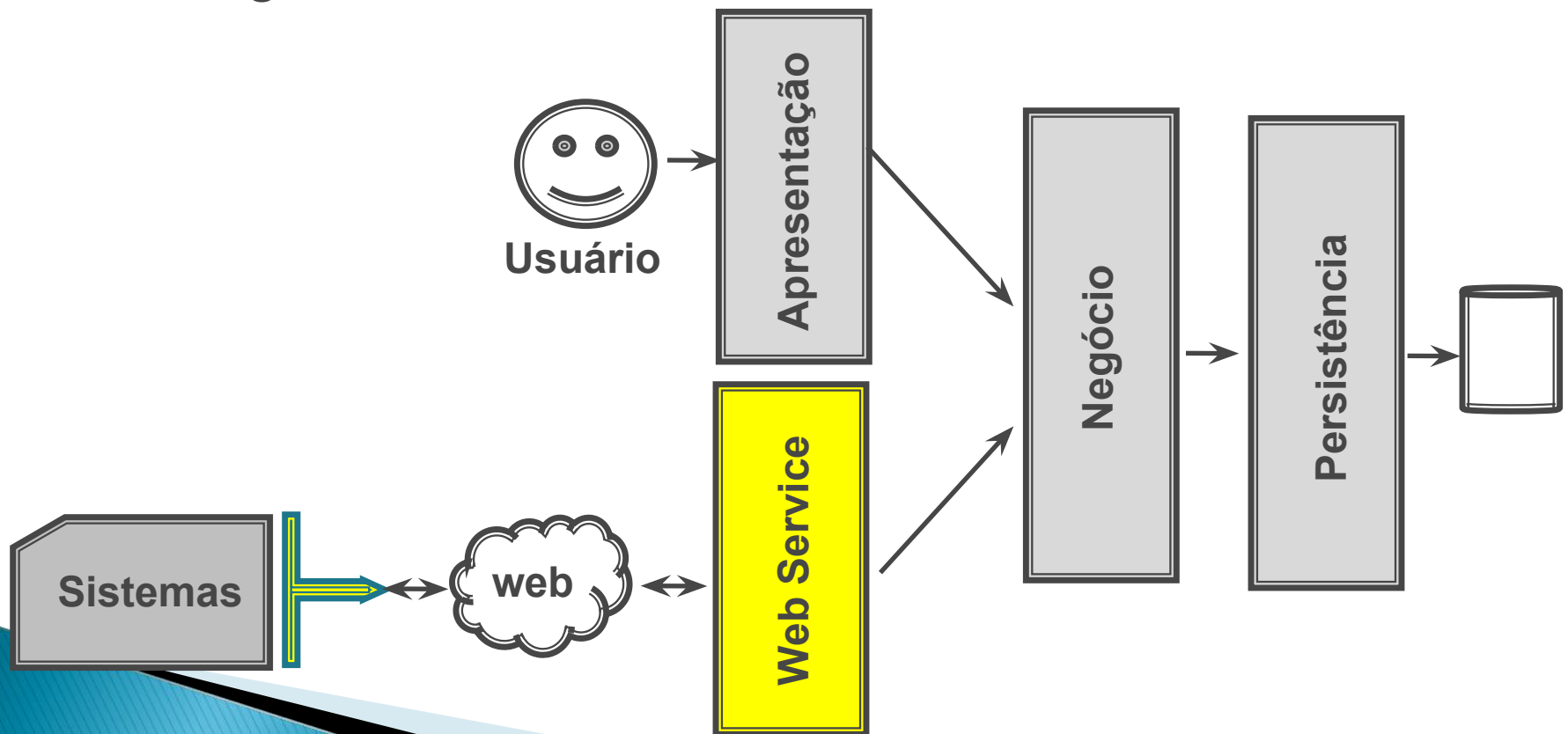
# Arquitetura – Web Service

- ▶ A solução devidamente projetada esta dívida arquiteturalmente em camadas lógicas que organiza suas responsabilidades. (Em caso de dúvidas veja nossos cursos de arquitetura)



# Arquitetura – Web Service

- ▶ Com web services é criado uma camada (responsável por encapsular os detalhes WS) com dependência para a camada de negócio:

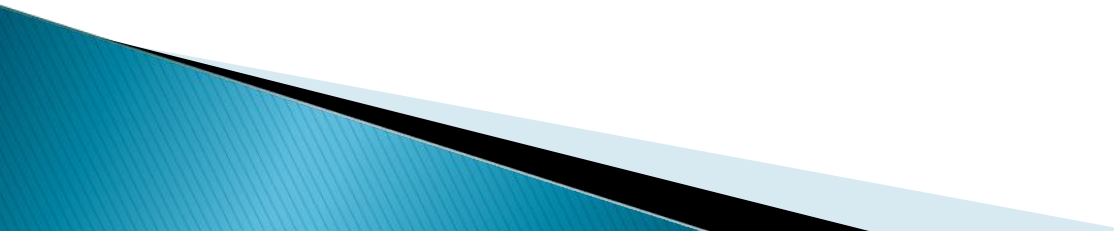


# Benefícios – Web Services



Java™


# Benefícios

- ▶ Interoperabilidade em ambientes heterogêneo:
  - ▶ Como as empresas se desenvolvem ao passar dos tempos, elas adicionam sistemas e soluções requerem diferentes plataformas de diferentes provedores que não se comunicando com as outras.
  - ▶ Permite que diferentes serviços distribuídos sejam executados em variedade de plataformas de software e de arquitetura.
- 

# Benefícios

- ▶ Interoperabilidade em ambientes heterogêneo:
- ▶ Permite que eles sejam escritos em diferentes linguagens e filosofias de programação, independentemente de fornecedor de tecnologia.

# Benefícios

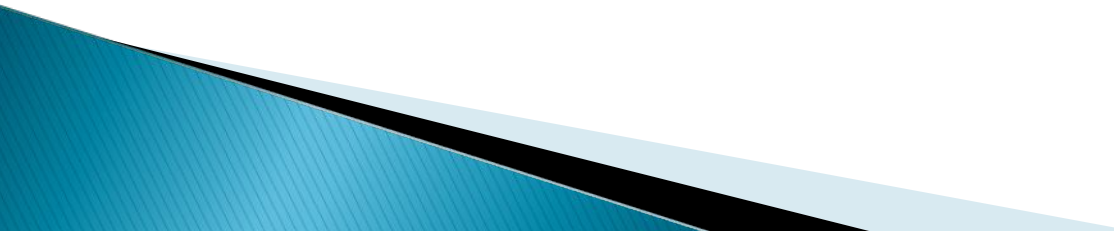
- ▶ Serviços de negócios através da web:
  - ▶ Uma empresa pode usar web services para alavancar as vantagens em âmbito mundial.
  - ▶ Ex: Uma empresa de catálogo de produtos pode colocar disponível seu estoque aos seus fornecedores através de web services de modo a conseguir melhor gerenciamento da cadeia de suprimentos.
- 

# Benefícios

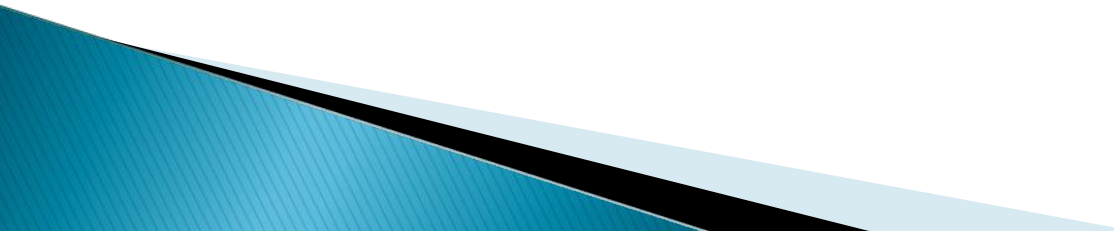
- ▶ Integração com sistemas existentes:
- ▶ A maioria das empresas possuem uma quantidade enorme de informações armazenados no sistemas de informações existentes e o custo para substituir isso é tão absurdo que não exista possibilidades de descartar sistemas legados.



# Benefícios

- ▶ Liberdade de opção:
  - ▶ O uso de padrões, convenções e diretrizes no desenvolvimento de web services abriram um grande mercado de ferramentas, produtos e tecnologias. Isso oferece aos desenvolvedores uma ampla variedade de opções (free, open source e comercial) na seleção de um produto para a criação de web services.
- 

# Benefícios

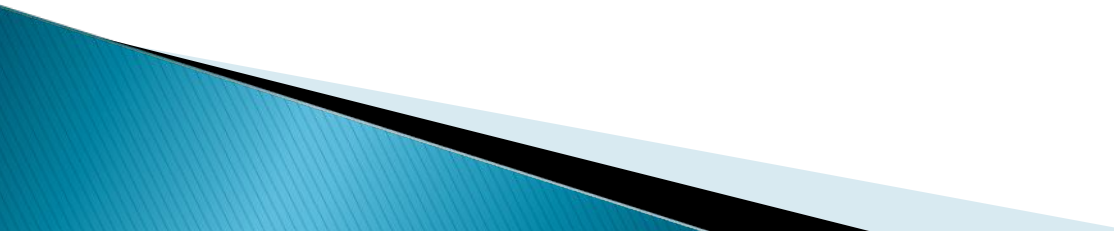
- ▶ Suportam qualquer tipo de aplicativos clientes:
  - ▶ Suportam qualquer tipo de dispositivo como aplicativo cliente de um web service devido independência de plataforma, tecnologias e provedores.
- 

# **Web Services – Desenvolvimento Orientado a Interfaces**



**Java™**

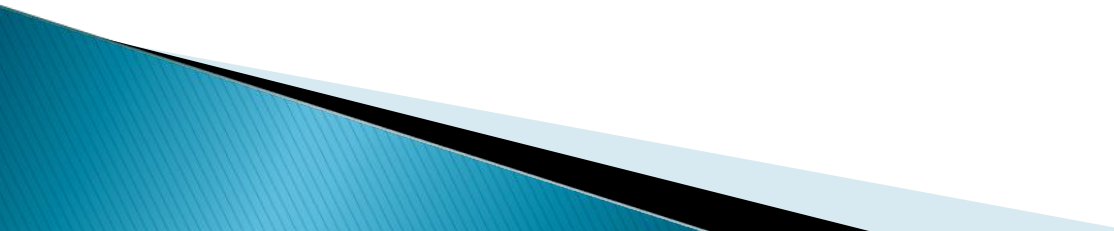
# Componentes via Interfaces

- ▶ Web services devem seguir uma boa prática de desenvolvimento chamado **Desenvolvimento Orientado a Interface**.
  - ▶ Que refere-se ao conceito de criar um componente de software que deve **encapsular e esconder** detalhes de implementação do seus serviços, promovendo **fácil manutenção, substituição e reutilização**.
- 

# Componentes via Interfaces

- ▶ Para aqueles que não sabem desse conceito, vejam:
- ▶ [http://en.wikipedia.org/wiki/Interface-based\\_programming](http://en.wikipedia.org/wiki/Interface-based_programming)
- ▶ [http://pt.wikipedia.org/wiki/Interface\\_\(programa%C3%A7%C3%A3o\\_orientada\\_a\\_objetos\)](http://pt.wikipedia.org/wiki/Interface_(programa%C3%A7%C3%A3o_orientada_a_objetos))

# Componentes via Interfaces

- ▶ Os primeiros exercícios SOAP e REST serão feitos com interfaces seguindo esse padrão.
  - ▶ Outros exercícios serão feitos sem interfaces para reduzir o tempo dos exercícios.
  - ▶ Lembrando que um web services de real em produção deverá seguir essa boa prática.
- 

# Dúvidas e comentários

- ▶ Dúvidas ou comentários, entre em contato pelos canais de comunicação.

