

# Programação Orientada a Serviços

**Prof. Diego Cirilo**

**Aula 02:** Arquitetura Orientada a Serviços

# Programação Orientada a Serviços

- Service Oriented Architecture (SOA)
- Sistemas monolíticos
- Arquitetura Cliente - Serviço
- Componentes reutilizáveis
- Manutenção
- Custo

# Origem

- Desenvolvimento da web nos anos 90;
- Acesso a informações fora da rede local;
- Pessoas acessavam serviços através de um navegador;
- Um programa poderia acessar um serviço?
- Thomas Erl

# O que é um serviço?

um ato ou desempenho oferecido de uma parte para outra. Embora o processo possa ser vinculado a um produto físico, o desempenho é essencialmente intangível e normalmente não resulta na posse de qualquer um dos fatores de produção (LOVELOCK *et. al.*, 1996)

# O que é um serviço?

- Representa logicamente uma atividade de negócio repetível e com um resultado específico;
- É auto contida;
- É uma *caixa-preta* para os consumidores, no sentido de que o consumidor não precisa saber o funcionamento interno do serviço.
- Pode ser composto por outros serviços.

# Serviços

- Serviços apresentam as seguintes características:
  - Entradas
  - Saídas
  - Objetivos
  - Transformações
  - Recursos

# E orientação a serviços?

- Uma forma de pensar o seu projeto através da comunicação entre um conjunto de serviços bem definidos e seus clientes.
- Não é uma tecnologia, e sim um estilo de arquitetura de software.
- Facilita o reuso de componentes de software.

# Características

- Neutro quanto a fornecedores;
- Direcionado para o negócio;
- Focado em corporações;
- Centrado na composição.

# Objetivos estratégicos

- Aumento da interoperabilidade intrínseca
- Aumento da federação
- Aumento das opções de diversificação de fornecedores
- Aumento do alinhamento entre negócio e tecnologia
- Aumento do retorno sobre investimento (ROI)
- Aumento da agilidade organizacional
- Redução da carga de trabalho da TI

# Benefícios

- Maior agilidade de negócio, menor *time-to-market*
  - Reusabilidade
  - Eficiência
- Aproveitamento de funcionalidades em novas aplicações
- Melhor colaboração entre gestão e TI
  - Representa melhor as estruturas dentro de uma empresa, como setores e responsáveis.
  - Facilita a comunicação usando termos em comum, ex. gerar um orçamento de seguro, calcular retorno de investimento, etc.
- Padronização

# Benefícios

- Manutenção
- Escalabilidade
- Independência de plataforma
- Independência de funcionalidades

# Problemas

- *Overhead*
- Maior complexidade
- Maior custo inicial

# Princípios do projeto SOA

- Contrato de serviço padronizado
- Baixo acoplamento
- Abstração
- Reusabilidade
- Autonomia
- Ausência de estado (*stateless*)
- Visibilidade
- Composição

# Dúvidas?

