# Introdução a Serviços

Marcelo Werneck Barbosa

# Introdução

Conceito de orientação a serviços não é novo.

Podemos imaginar uma cidade orientada a serviços.

Cada empresa provê serviços para diversos consumidores.

# Introdução

Se dependências são muito grandes, negócios podem ser inibidos.

Negócios podem ter que se adequar a algumas convenções como: usar a mesma moeda, empregados falam mesma língua que consumidores, interface entre os negócios, ...

# Introdução

Convenções trazem padrões benéficos aos consumidores sem influenciar cada negócio.

SOA – unidades existem de forma autônoma não isoladas umas das outras. Precisam se adequar a princípios e padronização.

Unidades – são chamadas serviços.

# SOA – Service Oriented Architecture

Paradigma de desenvolvimento

Objetivo é criar módulos funcionais – serviços – com baixo acoplamento e permitindo reutilização

# Acoplamento

Medida da interdependência entre dois ou mais módulos.

Quanto mais acoplados, mais dependentes e vulneráveis a mudanças

# Coesão

Seus elementos internos estão relacionados uns com os outros para executar a função do módulo.

Quanto mais coesos, mais fáceis de manter e menos afetados por mudanças.

# Componentes fracamente acoplados

Web Services devem ser componentes fracamente acoplados.

Podem continuar a funcionar mesmo com alterações na comunicação como

- adição de novos parâmetros
- omissão de parâmetros
- mudança de ordem de parâmetros

# Granularidade

Ajuda a medir a complexidade de um modelo de componentes.

Granularidade fina

- Desce em nível de classes
- Muito detalhado

# Granularidade

Granularidade grossa

- Oculta o modelo de classes básico
- Componentes executam mais de uma função

SOA visa criar componentes de granularidade grossa.

# Serviço

É um componente que atende a uma função de negócio específica.

Recebe requisições e as responde ocultando todo o detalhamento do processamento.

# Serviço

Executa unidades completas do trabalho

- Podem ser “stateless” (não armazenam estado)
- Facilita reutilização

# Manutenção de Estado

Se saída de um componente é afetada pelo seu estado interno (não somente pela entrada), diz-se que é um componente stateful

Exemplo: armazenar código do cliente para mostrar produtos de que gosta

São problemáticos, exigem maior acoplamento, de difícil manutenção e baixa reutilização.

# Como serviços interagem?

Devem ter conhecimento da existência uns dos outros.

Este conhecimento é realizado através de descrições de serviços (*service descriptions*).

# Mensagens

Comunicação através de mensagens.

Mensagens devem ser autônomas

- Unidades independentes de comunicação

Depois que serviço envia uma mensagem, perde controle do que acontece com ela.

# Princípios do Projeto de Serviços

Fracamente acoplados: Mantém uma relação que minimiza dependências e apenas têm conhecimento de outros.

Contrato do serviço: Serviços concordam com um contrato definido por uma ou mais descrições de serviço e documentos relacionados.

# Princípios do Projeto de Serviços

Autonomia: Possuem controle sob a lógica que encapsulam.

Abstração: Escondem lógica do mundo externo.

Reusabilidade: Lógica dividida em serviços para promover reuso.

# Princípios do Projeto de Serviços

Ausência de estado: Serviços minimizam armazenamento sobre informações específicas de suas atividades.

Serviços são projetados para serem encontrados e acessados por mecanismos específicos.