

Faculdade SENAI FATESG

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Componente Curricular: Arquitetura e projeto de Software

Docente: Elisabete Tio Hato

Aluno: Jhonathan dos reis santa rosa

## SOA

Esse conceito de arquitetura busca disponibilizar as funcionalidades de um sistema como um serviço.

Seu principal objetivo é ser mais flexível em atender às necessidades do mercado.

Os usuários dessa abordagem ganham em flexibilidade, Agilidade e redução de custos na reutilização de serviços. O alinhamento com as regras de negócio é um dos maiores Benefícios desse tipo de arquitetura. Podemos elencar outras Vantagens como:

- A diminuição do tempo de desenvolvimento;
- O baixo acoplamento entre as partes do sistema facilita a manutenção;
- O isolamento da estrutura de um serviço traz flexibilidade durante mudanças;
- Facilidade de agregar novas tecnologias a plataformas;
- E a possibilidade de reutilização de componentes.

Para implementar o SOA com sucesso é importante ter uma infraestrutura bem desenvolvida, arquitetura de referência consolidada e uma política de governança para direcionar os trabalhos ajudando as empresas a estarem preparadas para evoluir em tecnologia e em rentabilidade, diminuindo restrições da tecnologia para os líderes de negócio, também possibilita assegurar uma estrutura flexível e reutilizável.

### Tipos de serviço em SOA

De acordo com Thomas Erl serviços podem ser classificados como:

- Entity Services, empregados em operações de CRUD (inclusão, exclusão, alteração e / ou consulta a informações);
- Utility Services, contemplando funcionalidades que não estejam diretamente relacionadas ao negócio (log, envio de e-mail, etc.);
- Task Services, utilizados na automação de processos de negócio.

Tais estruturas costumam implementar a composição de serviços, com o consumo de Entity e/ou Utility Services;

- Orchestrated Task Services, os quais incorporam lógica de orquestração e controlam o fluxo em composições de serviços que envolvam Entity, Utility e Task Services.

Exemplos:

**Reusabilidade** = A implementação de um novo serviço deve levar em conta a possibilidade de reuso do mesmo. Logo, chances consideráveis de utilização de um Web Service imediatamente ou, até mesmo, a médio prazo constituem boas justificativas para a adoção deste tipo de estratégia.

**Autonomia** = A ideia de autonomia está ligada à independência de um serviço em relação a influências externas. Logo, um nível alto de autonomia é uma característica extremamente desejável na implementação de serviços.

## Aplicações Distribuídas

É um sistema que possui componentes localizados em computadores interligados em rede e que se comunicam e coordenam suas ações através da troca de mensagens entre os componentes. É comum pensarmos em componentes como sendo apenas servidores e máquinas conectadas em rede, mas quando falamos de componentes podem ser componentes de hardware (servidores) ou de software (aplicações e serviços).

Os componentes interligados em rede podem estar separados por qualquer distância. No mesmo edifício de uma empresa, na mesma cidade, mesmo estado, no mesmo país ou até em continentes diferentes.

São alguns exemplos de sistemas distribuídos:

A internet é o maior exemplo de sistema distribuído

Qualquer aplicação intranet

Qualquer aplicação mobile

Aplicações e serviços baseados na Computação em Nuvem

Os componentes de um sistema distribuídos são um grande desafio.

Em sistemas centralizados isso não existe pois há um único componente que é o próprio sistema centralizado. Em SD é um pouco diferente pois temos diversos componentes espalhados, ou distribuídos por diferentes servidores, que podem ser heterogêneos,

com diferentes sistemas operacionais, de diferentes fabricantes,  
com

diferentes capacidades de processamento.

Daí surgem diversos desafios na computação distribuída, entre os  
quais podemos citar:

- Segurança
- Escalabilidade
- Desempenho
- Comunicação entre os componentes
- Tolerância a falhas

Todos estes aspectos devem ser considerados quando trabalhamos  
com sistemas distribuídos. Para um bom funcionamento de  
aplicações de sistema a solução envolve não apenas projeto e  
arquitetura do software, mas também infraestrutura na qual o  
software será executado.