**Introdução**

* Familiaridade com as ‘palavras-chaves‘ do SOA;
* Entender como os as aplicações de SOA composites implementação Oracle Soa Suite 11g.

**Pré-Requisitos – Palavras Chaves**

**XML Fundamentals:** descreve os tipos de estrutura de dados.

**XSD(XML schema**): É uma alternativa DTD baseada em XML. Ele define um bloco valido no XML. Como: atributos/elementos/filhos/valores/etc..

**XPath:** Query Language(permite construir expressões que recorrem um documento XML de modo parecido a uma expressão regular).

**WS:** solução usada a integração de sistema e na comunicação entre diferentes aplicações.

### UDDI: [*Universal Description Discovery and Integration*](http://en.wikipedia.org/wiki/Universal_Description_Discovery_and_Integration) *Protocolo* desenvolvido pra organização e registro de web service:

* **Publicação:** divulga serviços;
* **Descoberta:** procura e encontra serviços;
* **Ligação:** liga e interagi com o serviço;

Basicamente: distribuir/publicar/contém informações WS, ou seja, BUSCA. Ele é como o OER, que tem todas as informações do WS, e devolve o endpoint do serviço.

**WSDL:** *Web Service Description Language*, usada para descrever um WS. Funciona como um contrato do serviço;

**SOAP:** *Simple Object Acess Protocol*, um protocolo para trocas de informações estruturadas em uma plataforma descentralizada e distribuída. Baseia em camadas de aplicações (Stub/Skeleton).

**STUB:** funciona semelhante a um "proxy" para o objeto remoto. Quando um objeto local invoca um método num objeto remoto, o "stub" fica responsável por enviar a chamada ao método para o objeto remoto.

"stub" quando é invocado:

* Iniciar conexão com a "Virtual Machine" que contém o objeto remoto.
* Escrever e transmitir os parâmetros para a "Virtual Machine" remota.
* Esperar pelos resultados da invocação do método.
* Ler os resultados retornados.
* Retornar os valores ao objeto que executou a chamada .

**SKELETON:** correspondente ao "stub". O "skeleton" é responsável por enviar a chamada ao objeto remoto.

"skeleton" quando recebe uma chamada:

* Ler os parâmetros enviados pelo "stub"
* Invocar o método no objeto remoto
* Escrever e transmitir o resultado ao objeto que executou a chamada

**Protocols using stub/skeleton approach**

* **RPC** - Remote Procedure Call; protocolo que comunica processos de um programa PC chamar um procedimento em outro PC (por um endereço).*RPC (Cliente <-> Servidor)*
* **CORBA** - Common Object Request Broker Architecture;
* **RMI** - Remote Method Invocation;
* **ETC..**

**Proxy:** Contém a orquestração/coreografia expondo tudo.

**2- Introdução aos Conceitos de SOA (Arquitetura Orientada a Serviço)**

Basicamente SOA, é uma integração funcional dos sistemas, em volta de seus negócios e processos, atrás da comunicação por serviços.

**Serviços:**

Serviços são blocos da construção de SOA.

* Representa as funcionalidades dos negócios;
* Descreve a definição da sua interface;
* Abstrai a implementação;
* Utiliza protocolos padronizados, para realizar a interoperabilidade e desassociar as funções.

**Web Services**

Existe duas versões a primeira a antiga WS 1.0 conhecida como WS. E a atual WS-\* 2.0, nesta especificação temos: Discovery / Security/ Reliability/ Transactions e Management.

**SOA e Web Services**

SOA é a aproximação da arquitetura de serviços com o design.

WebServices são várias maneiras de se implentar SOA, por:

* Endereços (endpoints/address) interagindo os serviços;
* A escala de SOA permite vc fazer usar a arquitetura de cross (onde vários sistemas de diferentes linguagens se comunicam através de endereços (xsd...);

***IMPORTANTE:***  WebService (a partir da versão 2.0 WS-\*) são bem completos. Porém as comapnhias podem escolher desenvolver seus blocos em REST (*Representational State Transfer*).

REST = arquitetura

RESTFUL = implementação de REST

**REST:** termo se originou no ano de 2000, em uma tese de doutorado1 (PHD) sobre a web escrita por Roy Fielding, um dos principais autores da especificação do protocolo HTTP que é utilizado por sites da internet. ***REST não é um padrão, não é um protocolo e sim uma arquitetura. REST*** vê cada aplicação web como um conjunto de recursos, que representam um estado particular de um aplicativo. Quando você acessa este recurso, você está transferindo o estado (conteúdo), e talvez alterando o seu estado.

**RESTFUL:** Os sistemas que seguem os princípios REST são frequentemente chamados de RESTful. RESTful é uma implementação de um "web service" simples utilizando o HTTP e os princípios REST.

Outros pontos fortes de SOA:

* Reusability;
* Interoperability;
* Scalability;
* Cost Efficiency;

***Abordagens de implementação SOA***

* Top-Down: Business centric-approach = da interface com o usuário até o serviço mais baixo do sistema (de cima para baixo);
* Bottom-Up: Technical centric-approach = o serviço mais baixo do sistema até a interface com o usuário (de baixo para cima)