**Cuestionario 2 de Sistemas Distribuidos**

**1- Defina Servicio Web**

Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones que pueden estar desarrolladas incluso en diferentes lenguajes de programación y ejecutadas sobre cualquier plataforma.

Un Web Service es ofrecido por una aplicación que expone su lógica a clientes de cualquier plataforma mediante una interfaz.

Lo interesante de un Web Service es XML, que provee un lenguaje común para describir llamadas procedimentales remotas, servicios web y directorios.

**2- ¿Qué es SOA? ¿Qué es lo que permite?**

El Service-Oriented Architecture es un tipo de arquitectura de software que se basa en la integración de aplicaciones mediante servicios.

SOA permite la creación de una arquitectura que se integra o forma parte de sistemas de información y/o aplicaciones que pueden llegar a ser bastante escalables, que permiten modelar todo el proceso de negocio de alguna organización y que son capaces de exponer e invocar servicios web. Esto nos permite tener una rápida y fácil integración de nuestra información.

**3- ¿Qué son SOAP y REST relacionados con SOA?**

Representan dos tipos diferentes para implementar una arquitectura SOA.

**4- Defina REST.**

REST es una interfaz para conectar varios sistemas basados en el protocolo HTTP y nos sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON.

**5- Defina SOAP.**

SOAP es un protocolo de mensajería basado en XML que puede utilizarse por una variedad de protocolos de transporte como podría ser HTTP. Los servicios SOAP también son conocidos como Web Services que basan su comunicación bajo el protocolo SOAP.

**6- Explique la forma en que se manifiestan los diferentes estados durante el ciclo de vida para SOAP y para REST.**

Con SOAP, el movimiento por los diferentes estados se realiza interactuando con un extremo único del servicio que puede encapsular y proporcionar acceso a muchas operaciones y tipos de mensaje.

Con REST, se permite un número limitado de operaciones y dichas operaciones se aplican a recursos representados y direccionados por direcciones HTTP. Los mensajes capturan el estado actual o el requerido del recurso.

**7- ¿Qué es lo que SOAP le permite a los diseñadores?**

SOAP permite encapsular la complejidad del sistema, dando lugar a interfaces generadas automáticamente que permiten facilitar el diseño del sistema.

**8- Defina SOAP como protocolo.**

Los servicios SOAP funcionan por lo general por el protocolo HTTP (el más común cuando se invoca un Web Service), pero no están limitados a ese protocolo, sino que puede ser enviado por FTP, TCP, entre otros.

**9- ¿Cómo es SOAP relacionado con la robustez?**

SOAP es el mejor protocolo para la comunicación Server to Server o Partner to Partner ya que es un protocolo muy robusto. Este tiene un tipado mucho más fuerte, permite agregar atributos, definir espacios de nombre y evita la ambigüedad.

**10- ¿Qué tipo de tecnología es REST? ¿Qué tipos de datos transmite?**

REST es una tecnología más flexible que transporta datos por medio del protocolo HTTP, pero este permite utilizar los diversos métodos que proporciona HTTP para comunicarse. REST permite transmitir prácticamente cualquier tipo de datos, ya que el tipo de datos está definido por el Header Content-Type (que nos permite mandar XML, JSON, Binarios, al contrario de SOAP que solo permite transmitir datos XML).

**11- ¿Con qué modelo está relacionado REST?**

REST se utiliza para obtener datos en aplicaciones WEB que funcionan principalmente con el Modelo MVC del lado del cliente.

**12- ¿Cuál de los dos es fuertemente acoplado?**

SOAP. Esto le permite ser testeado y depurado antes de correr la aplicación.

**13- Nombre una característica saliente entre REST y SOAP.**

SOAP: las operaciones son definidas como puertos WSDL.

REST: las operaciones se definen en los mensajes.

**14- Nombre una ventaja para REST y SOAP.**

SOAP: Mayor privacidad.

REST: Bajo consumo de recursos.

**15- Nombre una desventaja para REST y SOAP.**

SOAP: Los clientes necesitan puertos dedicados para diferentes tipos de notificaciones.

REST: Posee pocas herramientas de desarrollo.

**16- Nombre una característica tecnológica para REST y SOAP.**

SOAP: se centra en el diseño de aplicaciones distribuidas.

REST: se centra en la escalabilidad y el rendimiento a gran escala para sistemas distribuidos hipermedia.

**17- Nombre los protocolos involucrados para REST y SOAP.**

SOAP: SMTP, HTTP POST, MQ.

REST: HTTP GET, HTTP POST, HTTP PUT, HTTP DEL.

**18- Respecto a la seguridad que puede argumentar entre REST y SOAP.**

SOAP posee WS Security con comunicación origen a destino segura.

REST posee HTTPS con comunicación punto a punto segura.

**19- Explique páginas blancas, amarillas y verdes y ¿con qué están relacionadas?**

El registro de un negocio en UDDI (Universal Description, Discovery and Integration) tiene tres partes:

* Páginas blancas - incluye información general de un negocio: nombre del negocio, descripción y dirección.
* Páginas amarillas - incluye datos clasificatorios del negocio o del servicio que ofrece: su industria, producto o código geográfico basado en la taxonomía estándar.
* Páginas verdes - incluye información técnica sobre los servicios.

**20- ¿Cómo es el stack de Web Services?**

UDDI - Ofrece un directorio de servicios en Internet.

WSDL - Ofrece un modo de definir los servicios.

SOAP - Permite invocar métodos de los servicios.

XML y XML Schema - Permite a los consumidores de servicios enviar y recibir mensajes a y de los servicios.

HTTP, SMTP, TCP, … - Son protocolos abiertos de Internet. Dan soporte a las capas superiores.

**21- A través de qué concepto la UDDI establece una distinción entre la abstracción y la implementación?**

UDDI establece una distinción similar entre la abstracción y la implementación con el concepto de tModels. La estructura tModel, abreviatura de "Technology Model", representa huellas digitales técnicas, interfaces y tipos abstractos de metadatos. El resultado de los tModels son las plantillas de enlace, que son la implementación concreta de uno o más tModels. Dentro de una plantilla de enlace se registra el punto de acceso de una implementación particular de un tModel. Del mismo modo que el esquema de WSDL permite separar la interfaz y la implementación, UDDI ofrece un mecanismo que permite publicar por separado los tModels de las plantillas de enlace que hacen referencia a ellos.

**22- ¿Cómo es el proceso de publicación de la UDDI? (A grandes rasgos).**

UDDI es un registro público diseñado para almacenar de forma estructurada información sobre empresas y los servicios que éstas ofrecen. A través de UDDI, se puede publicar y descubrir información de una empresa y de sus servicios.

A través de un conjunto de llamadas a API XML basadas en SOAP, se puede interactuar con UDDI.

La información de UDDI se aloja en nodos de operador. Los operadores del host deben replicar datos entre ellos a través de un canal seguro, para conseguir la redundancia de la información en el registro UDDI. Se pueden publicar los datos en un nodo y descubrirlos en otro tras la réplica.

**23- ¿Cómo hace WDSL para distinguir a los puertos de los mensajes?**

En WSDL, la definición abstracta de puntos finales y de mensajes se separa de la instalación concreta de red o de los enlaces del formato de datos. Esto permite la reutilización de definiciones abstractas: mensajes, que son descripciones abstractas de los datos que se están intercambiando y tipos de puertos, que son colecciones abstractas de operaciones.

Un documento WSDL utiliza los siguientes elementos en la definición de servicios de red:

* **Message**: definición abstracta y escrita de los datos que se están comunicando.
* **Port:** punto final único que se define como la combinación de un enlace y una dirección de red

**24- ¿WSDL cómo define a los servicios? Contestar con detalle.**

Un documento WSDL define los servicios como colecciones de endpoints o puertos.

La definición abstracta de puertos y mensajes es separada de su asociación a una implementación concreta o formato de datos. Esto permite la reutilización de mensajes que son descripciones abstractas de los datos intercambiados y tipos de puertos que son colecciones abstractas de operaciones.

Un puerto se define asociando una dirección de red a un enlace reutilizable y una colección de puertos definen un servicios.

Por lo tanto, un documento WSDL usa los siguientes elementos en la definición de servicios:

* Tipos: proporciona la definición de tipos de datos XSD.
* Mensaje: representan una definición abstracta de los datos.
* Operación: una descripción abstracta de una acción.
* Tipo de puerto: un conjunto de operaciones abstractas soportado por uno o más endpoints.

WSDL tiene cuatro primitivas que un endpoint puede soportar:

* **One-way:** el endpoint recibe un mensaje.
* **Request-Response:** el endpoint recibe un mensaje y responde con otro.
* **Solicit-Response:** el endpoint envía un mensaje y recibe otro.
* **Notification:** el endpoint recibe un mensaje.