CAPÍTULO 1: INTRODUCCIÓN

1.1.- Introducción a los servicios Web

En los últimos años la mayoría de los procesos de negocio han cambiado para dar una mayor flexibilidad, interconectividad y autonomía debido a las condiciones del mercado, a los nuevos modelos organizacionales y a los escenarios de uso de los sistemas de información. En este contexto, Internet y la Web están cambiando la forma en la que se ofrecen los negocios y los servicios a la sociedad global, y en la que estos negocios se relacionan entre si.

Esta tendencia nos lleva a sistemas de información conectados e integrados a través de la infraestructura que proporciona Internet. Internet introduce un nuevo entorno donde el software se va a ofrecer y acceder como servicio. Los **servicios Web** proporcionan la plataforma tecnológica ideal para conseguir la completa integración de los procesos de negocio de una organización con diferentes organizaciones.

Estos servicios consisten de un conjunto de estándares que permiten a los desarrolladores implementar aplicaciones distribuidas, utilizando herramientas muy distintas para crear aplicaciones que utilizan una combinación de módulos de software que son llamados desde diversos sistemas distribuidos en distintas ubicaciones.

La arquitectura de los **servicios Web** es una estructura que permite que ciertos servicios de la red sean dinámicamente descritos, publicados, descubiertos e invocados en un entorno informático distribuido. Los servicios Web son aplicaciones modulares autónomos que pueden ser descritos mediante un lenguaje de descripción de servicio, como el lenguaje **WSDL** (*Web Service Description Language*), publicados al someterlos a las descripciones y políticas de uso en algún registro, utilizando **UDDI** (*Universal Description, Discovery and Integration*) o **LDAP** (*Lightweight Directory Access Protocol*) y localizados al enviar peticiones al registro UDDI y recibir detalles de binding del servicio que se ajusta a los parámetros de la búsqueda de un servicio.

La relación entre los distintos componentes se muestra en la figura de abajo.

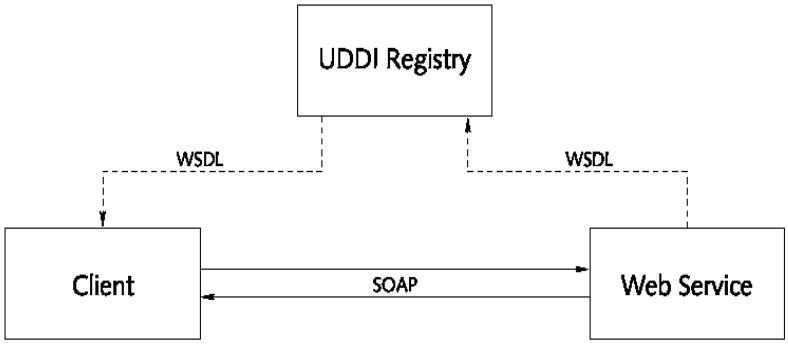

Figura 1. 1: Relación entre los componentes de un servicio Web

El fichero **WSDL** (Web Services Description Language), en formato XML, describe, al ordenador que lo consulta, la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo.

Del mismo modo, al igual que en la Web tenemos buscadores como Google, que nos llevan a las páginas que nos interesan, existe el concepto equivalente a nivel de Servicios Web, que es **UDDI** (Universal Description Discovery Integration). UDDI es un Servicio Web en línea que se puede utilizar desde las aplicaciones para descubrir de forma dinámica otros servicios en línea, todos ellos perfectamente integrados en una interfaz XML simple.

La funcionalidad de los protocolos y lenguajes empleados en estos servicios es la siguiente:

- **XML(eXtensible Markup Language):** Un servicio Web es una aplicación Web creada en XML.
- **WSDL (Web Services Definition Service):** Este protocolo se encarga de describir el servicio Web cuando es publicado. Es el lenguaje XML que los proveedores emplean para describir sus servicios Web.
- **SOAP (Simple Object Access Protocol):** Permite que programas que corren en diferentes sistemas operativos se comuniquen. La comunicación entre las diferentes entidades se realiza mediante mensajes que son rutados en un sobre SOAP.
- **UDDI (Universal Description Discovery and Integration):** Este protocolo permite la publicación y localización de los servicios. Los directorios UDDI actúan como una guía telefónica de servicios Web.

Pila de interoperabilidad de Servicio Web	**UDDI** (Universal Description, Discovery and Integration)
	SOAP (Simple Object Access Protocol)
	XML (eXtensible Markup Language)
	Protocolos comunes de internet (**HTTP, TCP/IP**)

Figura 1. 2: Pila de interoperabilidad en servicios Web

1.1.1.- Ventajas e inconvenientes de los servicios Web

Los servicios Web en todo su conjunto ofrecen las **siguientes ventajas** tecnológicas que han provocado que hayan tenido mucho éxito:

Interoperabilidad

Cualquier servicio Web puede interactuar con cualquier otro servicio Web. El protocolo SOAP permite que cualquier servicio pueda ser ofrecido o utilizado independientemente del lenguaje o ambiente en que se haya desarrollado.

Omnipresencia

Los servicios Web se comunican utilizando HTTP y XML. Cualquier dispositivo que trabaje con éstas tecnologías puede tanto ser un cliente del servicio como servidor en algunas circunstancias.

Mínimo Esfuerzo

Los conceptos detrás de los servicios de Web son fáciles de comprender y se ofrecen Herramientas de Desarrollo específicas por [WebLogic], Sun, Apache los que permiten a los programadores implementar rápidamente servicios Web con SOAP.

Uno de los **problemas a solucionar para los servicios Web** es la gestión de sistemas altamente distribuidos ya que algunos servicios Web delegan funcionalidades a otras de más bajo nivel. Además aparecen problemas de seguridad que se expondrán más adelante.

1.1.2.- Seguridad

Aunque las expectativas alrededor de la tecnología de servicios Web son grandes, también puede tener sus **riesgos** ya que los servicios Web hacen uso de las mismas tecnologías que han sido atacadas en tantas ocasiones. Si los usamos, la seguridad de una empresa puede verse comprometida. La **ausencia de técnicas de seguridad estándar** es un obstáculo para la adopción de la tecnología. La calidad de un servicio Web es un parámetro que no está demasiado claro, pero cuya medida es fundamental a la hora de desarrollar un servicio maduro. Esta tecnología está en desarrollo y la mayoría de los protocolos en los que se basa, son recomendaciones a estándar.

Actualmente, los **servicios Web** están siendo ampliamente aceptados por las empresas para el desarrollo de software de uso interno. De este modo, los servicios pueden implementar toda su funcionalidad y permanecer seguros tras el *firewall* de la empresa. Los desarrollos actuales no ayudan a la cooperación entre las empresas ya que no hay ningún estándar establecido sobre las técnicas de seguridad.

Debido a la tecnología que es usada por los **servicios Web**, y en concreto al **uso de SOAP**, las **técnicas de seguridad convencionales** que se han venido usando en Internet, **ya no son suficientes**. Con SOAP, cada mensaje simple que se intercambia realiza múltiples saltos y es rutado a través de numerosos puntos antes de que alcance su destino final. Es por ello que los servicios Web necesitan tecnologías que protejan los mensajes desde el principio hasta el final.

Existen un conjunto de técnicas que se pueden usar para garantizar la seguridad a nivel de mensaje. Estas son:

- **Cifrado de XML** - Evita que los datos se vean expuestos a lo largo de su recorrido.

- **Firma Digital XML** - Asocia los datos del mensaje al usuario que emite la firma, de modo que este usuario es el único que puede modificar dichos datos.

- **SAML y la Autorización** - SAML (*Security Assertion Mark-up Language*) hace posible que los servicios Web intercambien información de autentificación y autorización entre ellos, de modo que un servicio Web confíe en un usuario autentificado por otro servicio Web.

- **Validación de datos** - Permite que los servicios Web reciban datos dentro de los rangos esperados.

- Además, también hay técnicas que permiten mantener la **seguridad a otros niveles**. La seguridad en **UDDI** permite autentificar todas las entidades que toman parte en la publicación de un servicio Web: proveedor, agente y consumidor del servicio. De este modo, nadie podrá registrar servicios en el papel de un proveedor o hacer uso de ellos sin contar con los permisos adecuados.

  Estas técnicas las podemos resumir en la siguiente tabla.

Encriptación XML:	Evita que los datos se vean expuestos a lo largo de su recorrido. Xenc (*XML encryption*)
Firma Digital XML:	Asocia los datos del mensaje al usuario que emite la firma, de modo que este usuario es el único que puede modificar dichos datos. XML-SIG (*XML Signature*)
XKMS y los Certificados:	XKMS (*XML Key Management Specification*) define servicios Web que se pueden usar para chequear la confianza de un certificado de usuario. XACML (*eXtensible access control markup language*) vocabulario para especificar políticas de acceso.
SAML y la Autorización:	SAML (*Security Assertion Mark-up Language*) hace posible que los servicios Web intercambien información de autentificación y autorización entre ellos, de modo que un servicio Web confíe en un usuario autentificado por otro servicio Web.
Validación de datos:	Permite que los servicios Web reciban datos dentro de los rangos esperados.

Tabla 1.1: Técnicas de protección en servicios Web

De este modo podemos encuadrar nuestra implementación en el marco que acabamos de exponer.

1.2.- Organización de la memoria

Toda la documentación que se presenta en este libro está dividida en los siguientes capítulos:

• **Capítulo 1: "Introducción",** proporciona una visión general de la situación actual en la que se enmarca el proyecto.

• **Capítulo 2: "Diseño Global del proyecto",** proporciona una visión general del diseño de toda la aplicación al completo.

• **Capítulo 3: "J2ME",** realiza un estudio de la tecnología **J2ME** (*Java 2 Platform Micro Edition*) que se usa en la realización del cliente de la aplicación.

• **Capítulo 4: "XML",** describe el formato de datos **XML** que subyace en la información que transmitimos durante el intercambio de mensajes que se da en nuestra implementación.

• **Capítulo 5: "Protocolo HTTP",** estudiamos el protocolo de transporte **HTTP** que será el que usemos en la conexión del cliente con la aplicación que proporciona el servicio.

• **Capítulo 6: "SOAP",** describe el protocolo de transporte **SOAP** que es el que se usa para transmitir los mensajes entre las entidades que forman la aplicación que da el servicio.

• **Capítulo 7: "Conceptos de seguridad",** detallamos las características que deben tener los sistemas para que se consideren seguros. Además ajustamos esas características a las necesidades del comercio electrónico.

• **Capítulo 8: "Introducción a SAML"**, introduce la especificación SAML, repasando las características que incorpora.

• **Capítulo 9: "SAML",** realiza un estudio extenso de la especificación **SAML** detallando las partes que lo forman y explicando sus elementos.

• **Capítulo 10: "Seguridad en SAML",** describe los posibles riesgos de seguridad que puede tener la implementación así como los posibles ataques que puede sufrir.

• **Capítulo 11: "Diseño de la implementación",** describe en detalle el diseño de la aplicación que hemos desarrollado.

• **Capítulo 12: "Pruebas",** realiza un estudio de cómo se comporta la implementación cuando la sometemos a pruebas de uso. Para ello se utilizara un software llamado **JMeter** que realizara dichas pruebas.

• **Capítulo 13: "Instalación y Uso",** proporciona una guía completa de instalación y uso de la aplicación a nivel de usuario.

• **Capítulo 14: "Planos de Código",** es el conjunto de todos los códigos desarrollados en la aplicación.

• **Capítulo 15: "Temporización y Presupuesto",** describe en detalle el tiempo dedicado en la realización del proyecto y los costes incurridos en el desarrollo de la aplicación.

• **Capítulo 16: "Conclusiones",** ofrece una visión general de las conclusiones extraídas del proyecto una vez terminado éste.

• **Capítulo 17: "Líneas Futuras",** describe las posibles ampliaciones que se podrían realizar sobre la aplicación desarrollada para que adquiera una mayor funcionalidad.

• **Capítulo 18: "Referencias Bibliográficas",** describe en detalle toda la documentación empleada en la realización del proyecto.

• **Capítulo 19: "Glosario",** ofrece una relación detallada de todos los términos utilizados en la documentación del proyecto.