# 5.3 Patrones De Diseño

Cuando se desarrolla una aplicación software es frecuente encontrarse en la situación de tener que volver a resolver problemas similares a otros que ya hemos solucionado anteriormente, y debemos volver a hacerlo partiendo de cero una y otra y otra vez (incluso dentro del mismo proyecto).

Debido a ello y basándose en la programación orientada a objetos surgieron los patrones de diseño, donde cada uno de ellos define la solución para resolver un determinado problema, facilitando además la reutilización del código fuente.

Dependiendo de su finalidad pueden ser:

* **Patrones de creación**: utilizados para crear y configurar de clases y objetos.
* **Patrones estructurales**: su objetivo es desacoplar las interfaces e implementar clases y objetos. Crean grupos de objetos.
* **Patrones de comportamiento**: se centran en la interacción entre asociaciones de clases y objetos definiendo cómo se comunican entre sí.



Antes de seleccionar un patrón de diseño debemos estar seguros de que sea adecuado para el caso de que se trate, teniendo en cuenta que si debemos hacerle más que unos cambios mínimos puede ser señal de que no sea el más adecuado para lo que necesitemos desarrollar.

Otro concepto del que puedes oír hablar es el de **anti-patrones**, que hace referencia a los errores que comúnmente suelen ocurrir al intentar solucionar problemas conocidos.

No es necesario memorizar cómo funcionan todos los patrones, pero sí es importante saber de su existencia para así recurrir a ellos en caso necesario.

Para aprender más sobre este tema te recomendamos encarecidamente el libro **Patrones de Diseño** cuyos autores Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides (conocidos como 'la banda de los cuatro' o 'gang of four'... de aquí en adelante **GoF**) recopilaron 23 de entre los más útiles.

Los patrones que explicamos a continuación han sido extraídos de dicho libro, y aunque nos hemos esforzado en explicarte de la forma más clara y sencilla posible cómo interactúan los diferentes 'engranajes' en cada uno de ellos, es muy recomendable que te hagas con un ejemplar del mismo ya que contiene mucha información importante.

# 5.4 Estándares En Servicios

La definición dada por el World Wide Web Consortium (los responsables de XML) es:«XML describe una clase de objetos de datos llamados documentos XML, y describe parcialmente el comportamiento de los programas que los procesan. Los documentos XML están formados por unidades de almacenamiento llamadas entidades, las cuales contienen datos analizados (parsed) o sin analizar (unparsed). Los datos analizados están formados por caracteres, algunos de los cuales conforman datos y otros etiquetas.»

Aunque esta definición es un poco compleja por la cantidad de elementos involucrados, es importante resaltar que XML permite representar documentos («documento» se refiere no sólo a textos, sino también gráficos, correos electrónicos, transacciones de comercio electrónico, etc.) y que todo documento XML posee una estructura lógica y una física. Físicamente, el documento está compuesto de unidades llamadas entidades. Una entidad, en términos más simples, puede verse como un «trozo de documento», es decir, puede ser un simple carácter o todo el capítulo de un libro. Todo documento tiene una entidad raíz.

En un documento XML es posible hacer referencia a entidades internas o externas, indicando que están definidas en el mismo documento o en otros documentos (que pueden estar en otros archivos del disco duro, en una base de datos, en Internet, etc.), lo que hace muy flexible la construcción de documentos XML.

Lógicamente el documento XML está formado por elementos, que representan la estructura de la información que se está manejando, por ejemplo, si se fuera a representar un libro puede definirse que está formado por los siguientes elementos.

Cada elemento puede constar de otros elementos, así como de frases y palabras (normalmente el contenido del documento). El elemento que contiene a los demás (en el ejemplo, libro), se denomina elemento raíz, indicando que no depende de otros elementos.

Para definir un elemento en XML se tienen etiquetas y contenido, que en XML se denominan datos de tipo carácter.

Las estructuras físicas y lógica de un documento se organizan adecuadamente, para lo cual se han establecido unas normas que definen cuándo un documento XML está bien formado y cuándo es válido, o no.

WSDL: Web Services Definition Language

WSDL es el lenguaje común utilizado para la descripción de los servicios Web. Es un lenguaje basado en XML que describe totalmente la forma en la cual los clientes externos pueden interactuar con los servicios Web existentes en una máquina dada, los métodos que soportan y la sintaxis de los protocolos de comunicación (HTTP, SOAP).

En términos generales, un documento WSDL contiene información acerca de la interfaz, la semántica y los aspectos administrativos involucrados en una solicitud (llamado) a un servicio web.

Según el consorcio W3C (World Wide Web Consortium), encargado de la definición del estándar, WSDL es «un formato XML para la descripción de servicios de red, como un conjunto de puntos finales operando sobre mensajes que contienen información orientada a documentos u orientada a procesos». Las operaciones y mensajes se describen en forma abstracta y luego se asocian a un protocolo de red o a un formato de mensaje específico, para definir un punto final. WSDL se extiende para permitir la descripción de los puntos finales y sus mensajes, sin importar qué formatos de mensaje o protocolos de red se estén utilizando para la comunicación.

SOAP: Simple Object Access Protocol

El protocolo SOAP (Simple Object Access Protocol) utiliza mensajes XML para invocar métodos remotos. Un Servicio Web podría interactuar con servicios remotos a través de los métodos get y post de HTTP, pero SOAP es mucho más robusto y flexible.

SOAP es un protocolo liviano, basado en XML, para el intercambio de información estructurada en un ambiente descentralizado y distribuido. Sin embargo, SOAP no define la aplicación, ni la semántica de implementación. En vez de esto, proporciona un modelo de empaquetamiento modular y los mecanismos para la codificación de los datos dentro de los módulos. Esto permite que el protocolo simple se utilice en una amplia variedad de sistemas modulares y que cumpla su propósito primario de facilitar la interoperabilidad entre componentes de software heterogéneos.

SOAP es una especificación para la invocación de métodos en servidores, servicios, componentes y objetos, y codifica la práctica existente de utilizar XML y HTTP como un mecanismo de invocación de métodos.

Un paquete SOAP contiene información que se puede utilizar para invocar un método. La especificación SOAP no define la forma de llamar al método; tampoco maneja recolección de basura distribuida, ni la seguridad de tipos o HTTP bidireccional.

Lo que SOAP permite es el paso de parámetros y comandos entre clientes y servidores de HTTP, independientemente de las plataformas y aplicaciones existentes en el cliente y en el servidor. Los parámetros y los comandos se codifican utilizando XML.

UDDI: Universal Discovery, Description and Integration

Para los servicios Web, XML es el mecanismo por medio del cual los objetos de negocio acuerdan lo que van a hablar unos a otros, SOAP indica cómo van a efectuar la comunicación y UDDI les permite conocer con quién se van a comunicar y dónde encontrar otros objetos de negocio.

En pocas palabras, UDDI es una especificación para un registro distribuido de información acerca de los servicios Web. Define la forma en la cual se publica y descubre información acerca de éstos.

Un registro basado en UDDI es donde se pueden descubrir los servicios Web. El método utilizado por UDDI para el descubrimiento de servicios es tener un registro de aquellos servicios que se encuentran distribuidos a través del Web. En el registro distribuido, los negocios y los servicios se describen utilizando un formato XML común. Los datos estructurados en esos documentos XML son de fácil búsqueda, análisis y manipulación.

Debe anotarse que UDDI/SOAP no son los únicos modelos para el descubrimiento y manejo de mensajes en los servicios Web. EbXML ha sido desarrollado para un fin similar, así como para proporcionar una interfaz de negocio, seguridad robusta y otras funciones que permiten el manejo de transacciones reales de e-business. ebXML y UDDI/SOAP son tecnologías complementarias. Como su nombre lo indica, el estándar de descripción universal, descubrimiento e integración (UDDI), proporciona un mecanismo para que los objetos de negocio se describan a sí mismos y a los tipos de servicios que proporcionan, para luego registrarse y publicarse en un registro UDDI. Tales negocios “públicos” pueden ser buscados, consultados o descubiertos por otros objetos de negocio, utilizando mensajes SOAP. Después de descubrir otros negocios similares con los cuales pueden asociarse, los objetos de negocio pueden utilizar este mecanismo para “integrar” sus servicios con sus “socios” y proporcionar servicios a sus clientes.