

textOsFtextTOsFliningLFliningTLFtextosflininglftabulartabproportionalprosuperiorSup
superiorSup
fontspechyperref

TITLE

Author1

University

LA ARQUITECTURA PIRAMIDAL PARA SISTEMAS DE INFORMACIÓN DISTRIBUIDOS

¹Michael Rojas Rodríguez, ²Elizabeth Acosta Gonzaga, ³Jesús Antonio Álvarez Cedillo

1, 2, 3, Dpto. de Posgrado, CIDETEC IPN, U. P. Adolfo López Mateos, Av. Juan de Dios Bátiz s/n casi esq.

Miguel Othón de Mendizábal, Edif. del CIDETEC, Col. Nva. Industrial Vallejo, Del. Gustavo A. Madero, 07700, México, D. F.

¹michaelrojasr@gmail.com, ²egonzaga03@yahoo.com.mx, ³jaalvarez@ipn.mx

Resumen. En la actualidad muchos de los grandes sistemas son pensados para operar en forma distribuida, con el fin de centralizar la información y así poder llevar cada vez más orden y control en la operación de los comercios, el problema es que normalmente no se piensa en que tan fácil o difícil sería para un comercio adaptarse al sistema. En este trabajo vamos a referirnos principalmente a la arquitectura de sistemas distribuidos y orientados a servicios, en los cuales la misma arquitectura pueda plantearse para diferentes problemas a través de un patrón que sirva como base para generar soluciones tomando sus elementos. Este trabajo tiene el objetivo dar a conocer un patrón base y flexible que permita ser implementado en sistemas distribuidos con información centralizada que pretenda ofrecer servicios a una gran cantidad de comercios sin importar que algún comercio maneje alguna infraestructura especial, es decir se pretende que existan componentes adaptables a diferentes necesidades. En resumen tener un patrón genérico a soluciones distribuidas orientadas a servicios.

Palabras Clave: Sistema de Información Distribuido, Patrones Arquitectónicos, Arquitectura de Software.

Un sistema de información distribuido es un sistema en el cual sus componentes transmiten información, del tipo que sea mediante mensajes, pueden intervenir varios actores, los cuales de alguna manera participan en el proceso de circulación de la información entre ellos, de forma independiente el uno del otro.

Un sistema de información distribuido es una colección de elementos de que se encuentran físicamente separados y no comparten una memoria común, se comunican entre sí a través del intercambio de mensajes utilizando un medio de comunicación.

Un sistema de información distribuido debe contar con una arquitectura base que organice sus elementos, la Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución [3].

Cuando un problema o situación tiende a presentarse en forma recurrente o existen situaciones que pueden adaptarse de una misma manera, es pauta para utilizar un patrón que sirva como guía en la generación de soluciones,

un patrón de arquitectura de software describe un problema particular y recurrente que surge en un contexto específico, y presenta un esquema genérico y probado de su solución [2].

Los patrones arquitectónicos expresan un esquema estructural de la organización del sistema, es decir, un esqueleto o estructura de un sistema, dicha estructura consiste en subsistemas así como de sus responsabilidades e interrelaciones [1]. Aun cuando un patrón arquitectónico contiene la forma o imagen de un sistema, no es una arquitectura como tal, un patrón arquitectónico es un concepto que captura elementos esenciales de una arquitectura [4].

Por consecuencia al no ser una arquitectura como tal, un patrón al ser implementado permite que varias arquitecturas lo contengan y compartan características comunes.

1

El Patrón Piramidal

El patrón piramidal pretende ser un modelo a soluciones basadas en elementos que se comunican entre sí, con la característica de que la comunicación sea jerárquica. Es decir que exista un orden entre elementos y que cada elemento se comunique solo con un elemento del siguiente nivel según el orden en el que se encuentren.

El patrón piramidal se orienta a sistemas distribuidos que como se mencionó requieren de una comunicación jerárquica, ya sea por diseño o por seguridad o para evitar inconsistencia de datos, así también esta orientado principalmente para aplicaciones orientadas a servicios.

Por mucho tiempo la forma piramidal se ha utilizado para denotar jerarquía entre los niveles que la conforman. El patrón piramidal denota tres niveles en su estructura, el nivel base, el nivel de interpretación y el nivel fuente, La figura 1 muestra los niveles que conforman al patrón piramidal.



Figura 1, Niveles del Patrón Piramidal

El nivel base es el primer nivel de la pirámide y del cual se generan las peticiones o solicitudes de procesamiento de datos. El nivel base contiene todos los componentes que interactúan directamente con el usuario final.

El nivel de interpretación, es el que tiene los componentes que validan con reglas de negocio las peticiones del nivel base, para procesarlas y enviarlas al siguiente nivel (Nivel Fuente).

El nivel fuente es el último en la pirámide y contiene el procesamiento final de la petición, este nivel contiene componentes que proveen diversos servicios de terceros y por consecuencia la arquitectura de este nivel se considera como una caja negra.

2.

Características de los Niveles

Nivel Base:

1.

Los componentes que lo conforman se encuentran instalados de forma local.

2.

Pueden estar instalados en diferentes escenarios.

3.

No importa el lenguaje en el que se hayan desarrollado.

4.

No importa el sistema operativo sobre el que se encuentren instalados.

5.

Se comunican solo con el nivel de interpretación a través de un lenguaje o protocolo de comunicación estándar, como XML, por HTTP POST, GET o SOAP.

6.

Por el tipo de comunicación con el nivel de interpretación los componentes del nivel base deben tener salida a internet o intranet.

Nivel de Interpretación

1.

Es un nivel en el que sus componentes se encuentren centralizados en un servidor.

2.

Los componentes de este nivel cuentan con reglas de negocio que validan las peticiones del nivel base y para generar una respuesta entendible para este.

3.

Los componentes de este nivel se comunican con los componentes del nivel base a través del mismo protocolo o lenguaje de comunicación.

4.

Este nivel decide si la petición que se hace en el nivel base puede ser enviada al nivel fuente o no. Si la petición no puede ser procesada se genera el mensaje correspondiente y se envía de regreso la petición al nivel base, de lo contrario si la petición si puede ser procesada se genera un mensaje entendible para el nivel fuente y se envía.

5.

Los componentes de este nivel se comunican con los componentes del nivel fuente a través de un medio que los componentes del nivel fuente establecen.

Nivel Fuente

1.

En este nivel se contienen los componentes que proveen diversos servicios de terceros.

2.

Este nivel es el último nivel de la pirámide y es donde se procesa la respuesta final de la petición que se genera en el nivel fuente y que el nivel de interpretación considera correcto.

3.

La arquitectura de los componentes o la operación de estos se considera como una caja negra ya que provienen de servicios de terceros.

3.

Flujo de Operación y Características del Patrón Piramidal

El flujo de la operativa para un sistema basado en el patrón piramidal es de forma jerárquica, de modo que interactúan los componentes por niveles enviando y recibiendo mensajes, a continuación se enlista el flujo de operación.

1.

Un usuario a través de un componente del nivel base genera una petición.

2.

El componente del nivel base genera un mensaje que cumple con características comunes entendibles para algún componente del nivel de interpretación, se empaqueta el mensaje y se envía a través de un protocolo de comunicación.

3.

El componente del nivel de interpretación que fue invocado por el componente del nivel base, recibe el mensaje y lo interpreta.

4.

El componente del nivel de interpretación valida el mensaje y decide si puede ser procesado y enviado a algún componente del nivel fuente o si debe ser retornado al nivel base.

5.

Una vez que el mensaje que envía el componente del nivel base es correcto el componente del nivel de interpretación, genera un nuevo mensaje entendible para el componente correspondiente del nivel de servicios o nivel fuente, es por esto que este nivel se le llama nivel de interpretación.

6.

El componente o servicio del nivel fuente recibe la petición del nivel de interpretación, la procesa y genera un nuevo mensaje con la respuesta para enviarlo al componente del nivel de interpretación.

7.

El componente del nivel de interpretación recibe la respuesta del nivel fuente, la interpreta y genera un nuevo mensaje con el lenguaje o protocolo establecido con el nivel base.

8.

Finalmente el nivel base recibe la respuesta del procesamiento, la interpreta y genera las acciones necesarias para notificar dicha respuesta al componente que dio origen a la solicitud.

4.

Beneficios del Patrón Piramidal

En general el flujo de las partículas en el espacio hacia afuera de la malla, su movimiento puede verse afectado por varias situaciones, como son:

.

Facilita la colocación de la solución sobre cualquier plataforma, sin importar el lenguaje de programación, es debido a que todos los niveles se comunican a través de protocolos estándar de comunicación.

.

Los componentes del nivel base pueden ser cualquier "cosa" (cualquier medio), que tenga la facilidad de conectarse a internet o intranet y que pueda ejecutar un programa.

.

Se puede tener control tanto del nivel base como del nivel de interpretación.

.

No tiene restricciones sobre el modo en el que se comuniquen los niveles de la pirámide, puede ser cualquier protocolo de comunicación estándar, como SOAP.

.
Es posible hacer llegar a los usuarios finales de una forma organizada y segura las ventajas que ofrecen servicios de terceros.

.
Al tener centralizados los componentes del nivel de interpretación, se tiene mayor control de las peticiones que se hacen a los servicios de terceros y así también se tiene control de las respuestas que ofrecen dichos servicios, En este nivel se debe llevar el registro y administración de las peticiones y respuestas.

.
Los usuarios para tener acceso a un servicio pueden estar en cualquier sitio en el que se encuentre un componente del nivel base, por ejemplo si un componente del nivel base es una aplicación para celular, el usuario podría acceder a un servicio desde su dispositivo.

.
Por la estructura de niveles cualquier operación realizada en el nivel base, se registra en el nivel de interpretación y posteriormente se puede acceder al registro de dichas operaciones a través de un componente base que haga una petición de consulta al nivel de interpretación.

.
Los componentes del nivel base solo se preocupan por establecer una sola forma de comunicación, es decir la que establezcan con el nivel de interpretación, por definición el nivel base no tiene forma de comunicarse con el nivel fuente, lo que da ventaja a integrar en los componentes del nivel base solo un protocolo de comunicación.

.
Al estar centralizado el nivel de interpretación y al ser servicios los del nivel fuente, los componentes del nivel base puede acceder en cualquier momento a los servicios.

.
Al ser un lenguaje o protocolo estándar el que se utiliza para comunicarse con el nivel de interpretación se facilita la integración de nuevos tipos o escenarios en el nivel fuente.



Figura 2, Patrón Piramidal, un enfoque genérico

5.

Flujo de Implementación del Patrón Piramidal

1.

Se deben especificar las reglas de negocio a nivel base de datos, así como establecer restricciones y normalización de la misma.

2.

Se sugiere generar procedimientos almacenados y transacciones para la manipulación de la base de datos y no agregar queries al código de la aplicación.

3.

Posteriormente es conveniente iniciar con el análisis y programación del Web Service que se instalará en el nivel de interpretación, en este Web Service se deben establecer reglas de negocio acorde a las necesidades.

4.

Establecer protocolos de comunicación y conexión con servicios de terceros mismos que se pretenden consumir y que se encuentran en el último nivel de la pirámide (nivel fuente).

5.

Se debe establecer el lenguaje y medio de comunicación para la interacción entre el nivel de interpretación y el nivel base.

.

Se recomienda utilizar el lenguaje XML, y así generar esquemas bien definidos con tags que representen los datos que se desean manipular o consultar.

.

Se recomienda que cada método del Web Service reciba parámetros de tipo String en este caso se recomienda recibir solo un parámetro encriptado por algún método genérico y no usar un método de encriptación propio de algún lenguaje, este parámetro contendrá el XML con la información de la operación que se realiza.

6.

Una vez establecidos los XML de intercambio se debe generar el programa cliente el cual será el que se instale en las PC's de los usuarios finales y de donde se generarán las peticiones, este programa representa el nivel base de la pirámide.

.

En estas vistas existirán validaciones, interacción con hardware y en general lo necesario para generar los datos de las peticiones y para recibir la respuesta a las mismas.

.

Las vistas del nivel base generan los XML de petición para invocar el Web Service que se encuentra en el nivel de interpretación, así también reciben los XML de respuesta para generar el flujo correspondiente.

Figura 3, Flujo de Implementación del Patrón Piramidal

6.

Conclusiones

El patrón piramidal es un conjunto de herramientas orientadas a la creación arquitectónica de soluciones distribuidas, buscando siempre un punto genérico

con el fin de hacer genéricas a las soluciones como tal, y por lo tanto poder aplicarlas a diversos proyectos, principalmente aquellos que pretendan masificar soluciones

.

El patrón piramidal permite crear estructuras arquitectónicas basadas en tres Niveles conformados en forma jerárquica.

Cada Nivel cuenta con una funcionalidad en específico, el Nivel Base que es el primer nivel de la pirámide y del que nacen las peticiones, el Nivel de Interpretación él se segundo nivel de la pirámide y es el primero que recibe las peticiones del Nivel Base y en el cual se hacen las validaciones de reglas de negocio y manejo de bases de datos, por último se encuentra el Nivel Fuente el cual contiene servicios de terceros y al cual finalmente llegan las peticiones para ser procesadas.

Con el patrón piramidal no debe haber limitaciones de lenguajes de programación ya que la comunicación entre los Niveles debe estar basada en protocolos de comunicación estándar como SOAP y los mensajes pueden estar conformados por cadenas por ejemplo XML lo cual también es estándar.

Por este punto es posible utilizar diferentes lenguajes de programación entre los Niveles. El patrón piramidal en consecuencia puede ser aplicado a soluciones basadas principalmente en servicios.

Referencias

[1] DAVID GARLAN, Software Architecture: A Roadmap". En Anthony Finkelstein (ed.), the future of software engineering, ACM Press (2000).

[2] ALEXANDER WOLF, Succeedings of the Second International Software Architecture Workshop (ISAW-2). ACM SIGSOFT Software Engineering Notes (1997).

[3] MARY SHAW, DAVID GARLAN, Software Architecture: Perspectives on an emerging discipline. Upper Saddle River, Prentice Hall, (1996).

[4] DOUGLAS SCHMIDT, MICHAEL STAL, HANS ROHNERT AND FRANK BUSCHMANN, Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects, Volume 5, ISBN: 0471606952, (2007).

Referencias

[1] <http://www.robotis.com/xen/darwin>

[2] Brushless DC (BLDC) Motor Fundamentals, Padmaraja Yedamale Microchip Technology Inc.

[3] Técnicas de control para motores Brushless Comparativa entre conmutación Trapezoidal, conmutación Sinusoidal y Control Vectorial, Roger Juanpere Tolrà.

Referencias

- [1] Albert Einstein, Isaac Newton, Marie Curie, Galileo Galilei, Charles Darwin **(mayo - junio, 2025)** *La teoría de la evolución biológica. Boletín UPIITA. año 19, (108) 2025* [liga del artículo](#)