

Desarrollo de sistemas web con n-capas y metodologías ágiles en la universidad tecnológica de Tehuacán

Christian Galicia

C.Galicia
Universidad Tecnológica de Tehuacán, Tecamachalco, Puebla .

M. Ramos.,V.Aguilera.,(eds.). Ciencias de la Ingeniería y Tecnología, Handbook -©ECORFAN- Valle de Santiago, Guanajuato, 2013.

Abstract

El presente artículo describe la experiencia, procesos de desarrollo y evolución de la metodología empleada en los diferentes sistemas web desarrollados en la Universidad Tecnológica de Tehuacán y como parte del proceso de vinculación con empresas, así como de los resultados obtenidos.

16 Introducción

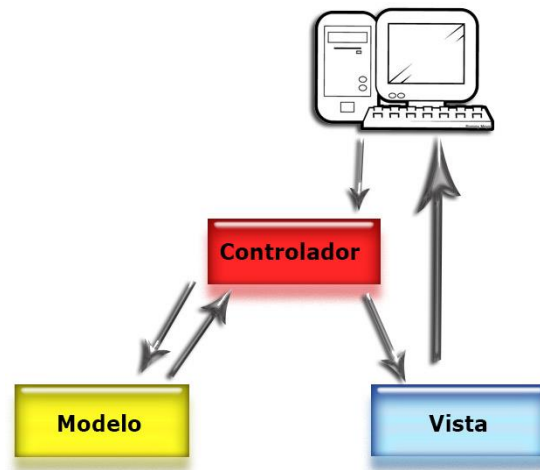
El desarrollo de software ha evolucionado en los últimos años dado por las exigencias y evolución de las propias empresas, las cuales requieren el desarrollo de aplicaciones mas ágiles, rápidas y con la capacidad de adaptarse a los cambios de las organizaciones, hasta hace poco el desarrollo de sistemas manejaba procesos muy rigurosos, con roles, actividades y documentación muy detallada, estas metodologías son totalmente funcionales para proyectos grandes, pero no suelen aplicarse bien en proyectos pequeños para los cuales existen metodologías mas ágiles.

Por otra parte en los sistemas de información web (SIW)[4] no existe una metodología completamente aceptada que permita guiar los procesos de desarrollo en los SIW aunque la tendencia dice que las metodologías ágiles son una posible solución para estos sistemas a diferencia de las metodologías tradicionales que imponen procesos de desarrollo muy pesados y burocráticos; afirma Fowler (2001)[2], Overmyer[5] en cambio dice que el desarrollo de aplicaciones web maneja un ciclo de desarrollo muy corto por lo cual las metodologías ágiles son perfectas para estos sistemas.

La arquitectura N-capas es una técnica que se aplica perfectamente al desarrollo web ya que ésta se puede separar en varias capas de manera muy fácil por la misma naturaleza y evolución de esta tecnología, actualmente el estilo más utilizado es el de 3 capas (Fig. 16):

1. Vista: en la cual interactúa el usuario. Controlador: la cual contiene las reglas de negocios.
2. Modelo: la cual contiene el acceso a los datos.

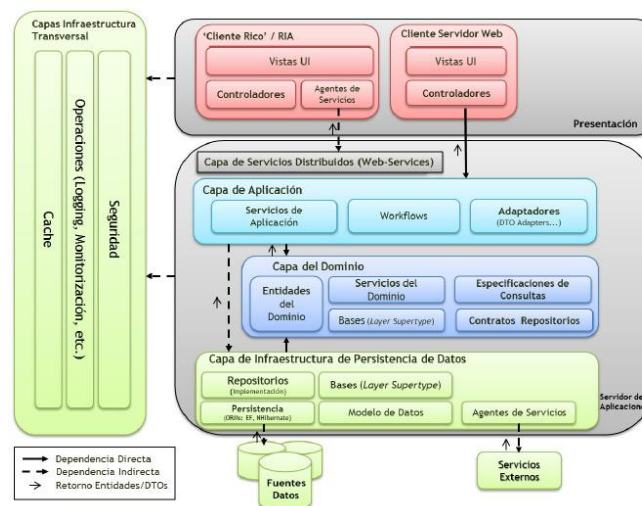
Figura 16 Modelo Vista Controlador (MVC)



Sin embargo nada impide ampliar el número de capas la ventaja es que esto nos permite diferenciar y separar los tipos de tareas a realizar ofreciendo un diseño que maximiza la reutilización y mantenibilidad del sistema.

Las capas se suelen agrupar por niveles y cada nivel puede tener sub-capas[1], dichas capas realizan a su vez tareas específicas ya que cuentan con roles muy específicos lo cual nos proporciona una clara idea de donde esta cada componente e incluso cada tipo de tecnología empleada.

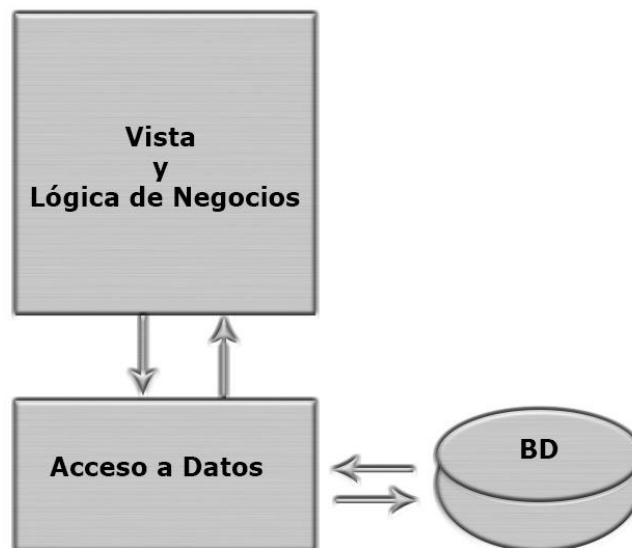
Figura 16.1 Arquitectura N-capas [1]



16.1 Desarrollo

La Universidad Tecnológica de Tehuacán es una universidad relativamente nueva por lo que tiene muchas necesidades en lo que se refiere a sistemas informáticos, gran parte de estos sistemas son desarrollados por parte de la carrera de TIC. Cuando se inicio el desarrollo de estos proyectos no existía una metodología completamente definida, los proyectos se desarrollaban bajo una metodologías clásica de cascada pero con algunas vertientes: requerimientos, diseño, implementación, verificación y mantenimiento, No se tenía una programación definida, cada desarrollo era diferente uno de otro. Dadas las características particulares de la organización lo cambios y evolución de sus procesos son constantes, los sistemas se debían adecuar, el proceso de actualización constante así como el tiempo disponible para estas tareas complicaba la situación, otra dificultad fue la estructura de programación que no permitía realizar cambios fáciles al código sin una repercusión a otras partes del mismo. A causa de estas particularidades la organización decidió busca procesos más eficientes; el primer paso fue el desarrollo de aplicaciones con arquitectura orientada a objetos, con el objetivo de reducir los tiempo y la reutilización de código, el primer logro fue la creación de una clase manejadora para la base de datos que facilitó el acceso a datos y que tenía la capacidad de ser reutilizable en varios proyectos de la organización (Fig. 16.1).

Figura 16.2 Arquitectura 2-capas



Esta clase o capa se ajustó para poder ser reutilizable en múltiples proyectos lo cual acortó el tiempo de desarrollo y creó la base para la reutilización de código (característica particular de la POO).

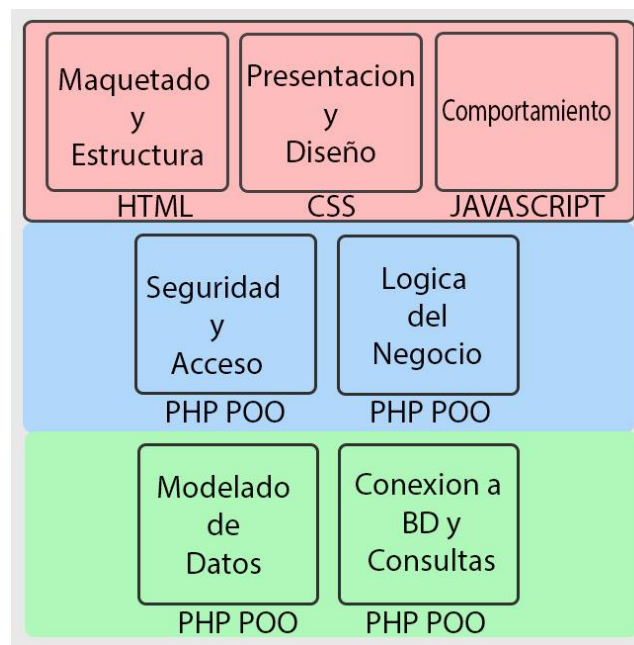
A partir del éxito de esta primera fase se decidió implementar el modelo MVC junto con la programación orientada a objeto en los sucesivos proyectos lo cual atrajo beneficios al desarrollo de software en la institución al tener piezas de código que fácilmente pueden ser reutilizables en múltiples proyectos. Dadas las características de la institución y el crecimiento rápido de la universidad, los cambios en los sistemas empezaron a ser más frecuentes y la metodología tradicional en cascada dejó de ser una manera útil de programar para este tipo de proyectos como lo afirma Fowler [2] a pesar de que el software se terminaba con mayor rapidez surgían los problemas de las actualizaciones, por lo cual se decidió ocupar metodologías ágiles para el desarrollo, metodologías con iteraciones incrementales que facilitarían la refactorización del código de manera más constante y rápida ya que uno de los problemas que se tiene es el tiempo disponible (8 horas a la semana para el desarrollo y mantenimiento del software).

Tabla 16 Evolución del proceso de desarrollo de software en la universidad tecnológica de Tehuacán.

Nombre	Cambios
Sistema para la evaluación a docentes por parte del alumnado	-Desarrollo en cascada -Programación secuencial -1 capa
Sistema de alta de calificación para profesores	-Desarrollo en cascada -Programación secuencial -2 capa
Administrador de contenido de la página de la universidad	-Desarrollo en cascada -Programación Orientada a Objetos -2 capas
Portal web para el ENDCUT con administrador de contenido	-Desarrollo con metodologías ágiles -Programación orientada a objetos -2 capas
Sistema de control de puntos de monitoreo de cuencas tributarias por Alternativas y procesos de Participación S.A.C	-Desarrollo con metodologías ágiles -Programación orientada a objetos -3 capas (MVC)
Sistema para el préstamo, control y búsqueda de libros de la biblioteca de la UTT	-Desarrollo con metodologías ágiles -Programación orientada a objetos -N-capas

El siguiente proceso de evolución y eficiencia llegó cuando la universidad empezó a ofrecer servicios de vinculación con empresas, las solicitudes de software se incrementaron y el tiempo para el desarrollo de estos seguía siendo el mismo a demás del problema de mantenimiento y actualización de los sistemas ya existentes, de ahí el siguiente paso fue el cambio al patrón MVC al cual se le incorporaron 2 sub-capas, (metodología sugerida por César de la Torre 2010-pag 38) [1] una en la vista y otra en el modelo con lo cual ya se tenía un estructura solida para el desarrollo del software lo cual incremento la funcionalidad del propio sistema, (el tiempo de desarrollo del sistema fue de solo un mes) .

Figura 16.3 Arquitectura N-capas empleado en los proyectos de la UTT



Nuestro proyectos más recientes ya manejan 7 capas, entre estos proyecto se encuentra el *Sistema para préstamo, control y búsqueda de libros de la biblioteca de la UTT*, el proceso de desarrollo está calculado para un mes y medio lo cual demuestra una mayor agilidad en el desarrollo a comparación de los métodos empleados con anterioridad, algo también ya demostrado por Highsmith [3] en su tabla comparativa de metodologías ágiles.

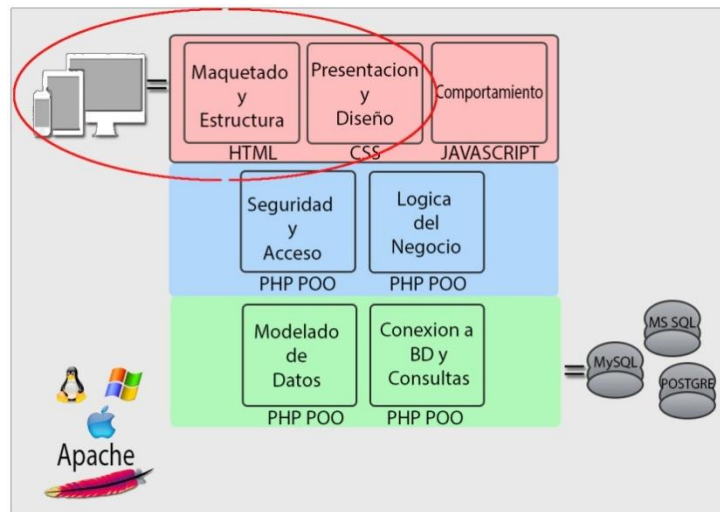
La reutilización de código y manejo de capas es un proceso que ahorra tiempo de programación y mejora la calidad del software creado en la institución ya que solo se tienen que ajustar y cambiar algunas capas, el framework de trabajo realiza lo demás.

16.2 Conclusiones

La incorporación de más capas a nuestro modelo de trabajo reduce el tiempo de desarrollo junto con las metodologías ágiles empleadas, al igual que ha optimizado los tiempos de mantenimiento y actualización del software, ya que la propia metodología trabaja con iteraciones cortas lo cual produce resultados más rápidos.

El siguiente paso para el desarrollo de nuestros proyectos y en el cual ya están haciendo pruebas es ocupar la tecnología responsiva, la cual se implementara en las sub-capas, esta tecnología permitirá hacer que nuestros sistemas tengan la capacidad de ser ajustados a cualquier resolución y por tal ser funcionales desde cualquier dispositivo lo interesante viene en que las modificaciones en los sistemas serán mínimos, solo se tendrían que hacer cambios en la sub-capa de presentación, diseño y se ajustaría la de maquetado y estructura, estos cambios traerían grandes beneficios por el hecho de que nuestros desarrollos serian multidispositivos y multiresoluciones (Fig. 16.4). Al trabajar en capas y sub-capas obtenemos el beneficio de que nuestros desarrollos tengan la capacidad de ajustarse a cualquier cambio o nueva tecnología en el mercado sin hacer cambios o procesos complicados.

Figura 16.4 Evolución de la Arquitectura N-capas empleado en el desarrollo de software en la UTT



Por último está considerado migrar los sistemas desarrollados con anterioridad al nuevo framework de trabajo con el objetivo de aplicar los beneficios a todos los desarrollos de la universidad.

16.3 Referencias

César de la Torre Llorente, U. Z. (2010). Guia de Arquitectura N-Capas Orientado al Dominio con Net 4.0.

Fowler, M. (2001). *The New Methodology*. Retrieved from <http://www.martinfowler.com/articles/newMethodology.html>.

Highsmith, J. (2002). Agile Software Development Ecosystems. In J. Highsmith.

Muñoz, A. (2003). *Sistemas de informacion en las empresas*.

Overmyer, S. (2000). What's Different about Requirements Engineering for Web Sites? In S. Overmyer.