



---

### Patrones de diseño para sistemas orientados a la toma de decisiones basándose en el Framework MVC para desarrollo de aplicaciones en el Instituto Tecnológico Superior de Pánuco.

*Eric.alvarez@itspanuco.edu.mx (Eric Alvarez Baltierra)<sup>1</sup>, angeles.ahumada@itspanuco.edu.mx (María de los Ángeles Ahumada Cervantes)<sup>2</sup>, patricia.melo@itspanuco.edu.mx (Julia Patricia Melo Morín)<sup>3</sup>.*

<sup>1</sup>Instituto Tecnológico Superior de Pánuco

#### RESUMEN

Hoy en día los programadores de Sitios Web, deben analizar con más detalle la manera en la que realizan todo necesario para cumplir los estándares de diseño en forma óptima debido a que constantemente las tecnologías van cambiando, de tal forma que existe un crecimiento acelerado y las instituciones educativas deben de mantener sus sistemas actualizados, además deben tener las opciones de poder mejorar de manera sencilla y agradable el entorno web en la que proporcionan información día con día a sus usuarios. Se pretende de llevar un análisis de los aspectos que deben de ser considerados en los patrones de diseño en la construcción del Marco de Trabajo de un Sistema web con la utilización de Modelo Vista Controlador.

**Palabras clave:** Sistema, web, modelo, controlador

#### PROBLEMÁTICA

La creciente necesidad de controles para la información generada en el Instituto Tecnológico Superior de Pánuco, ha llevado a la creciente y desproporcionada generación de sistemas de información de diferentes infraestructuras y diferentes capas de trabajo. Estos pueden ser sistemas ejecutables y algunos sistemas basados en la Web.

Los sistemas adquiridos hechos a la medida, hechos por diversas empresas de la región, por lo general solo generan códigos ejecutables, siendo que deben ser orientados a la web. Mas aun, compatibles con diversos sistemas que se tiene semejanzas y pueden compartir información, esto conlleva a tener sistemas aislados, como parte de desarrollos orientados a dar pequeñas soluciones a sistemas.

## INTRODUCCIÓN

Existen grandes bondades con el uso de patrones, ya que se centra la creación de software, siendo estos esquemas los que todos los profesionales de la creación de software los que reconocen cada una de las partes y descripción de un determinado programa. “Los patrones nos ayudan a construir sobre la experiencia colectiva de ingenieros de software experimentados. Estos capturan la experiencia existente y que ha demostrado ser exitosa en el desarrollo de software, y ayudan a promover las buenas prácticas de diseño. Cada patrón aborda un problema específico y recurrente en el diseño o implementación de un sistema de software”. (Daniele M, 2004).

De igual forma el Modelo Vista Controlador es especialmente diseñado para tratar convenciones tales que eficientan el desempeño de los sitios web, haciendo más productiva a la población de las instituciones educativas. Se sabe que la curva de aprendizaje de la metodología de MVC es en realidad muy pronunciada, esto debido a la complejidad técnica que conlleva la realización de sitios utilizando estas técnicas de desarrollo. Además, son importantes los patrones de diseño en la construcción de Marcos de Trabajo debido a que nos permiten desarrollar modelos de software eficientes para la creación rápida de aplicaciones distribuidas, la separación de comportamiento de las interfaces, así como la reducción de llamadas de cliente servidor y la centralización de los objetos del dominio entre otros. Hoy en día las instituciones educativas tienen una creciente necesidad de proporcionar información a los usuarios y contar con la facilidad de mejorar la visualización de su sistema web de información académica en forma rápida y a bajo costo de tal forma que les permita realizar cambios en el entorno en los cuales proporcionar información a sus usuarios.

Todo esto es posible ya que la creación de una sola vez de cualquier código, puede ser reutilizado con pocos cambios para realizar un sistema robusto e inteligente de forma rápida y con calidad.

Los patrones de diseño son una herramienta que brinda flexibilidad y robustez al desarrollo de sistemas de software facilitando el reúso del diseño y la arquitectura, pero un uso excesivo o inadecuado de ellos puede añadir mayor complejidad sin aporte de beneficio alguno. También el contexto en el que se estén utilizando interviene a la hora de seleccionar los patrones, por eso es que este framework no pretende ser inflexible y deja la posibilidad de aplicar otros patrones, según sea cada uno de los casos particulares. (Daniele M, 2004).

Una de las razones por las que la investigación en el campo de arquitecturas software atrae cada vez más interés, es la relación directa que existe entre las decisiones arquitectónicas y la satisfacción de ciertos atributos de calidad. La idea que subyace en esta relación es que una arquitectura software, necesita ser diseñada de forma explícita

para que el sistema final satisfaga unos atributos de calidad específicos. (Moreno, 2003)

El presente trabajo de revisión teórica, nos va a proporcionar lo aspectos internos que se deben de tomar en cuenta a la hora de desarrollar diseño de un Sistema Web de información académica a las instituciones educativas.

Un Marco de Trabajo es un diseño reutilizable de todo o parte de un sistema, representado por un conjunto de clases abstractas y la forma en la cual sus instancias interactúan. Un Marco de Trabajo encapsula el patrón de la arquitectura software de un sistema o de alguna de sus partes, que puede además estar parcialmente instanciada.

Las principales ventajas que ofrecen los Marcos de Trabajo son la reducción del coste de los procesos de desarrollo de aplicaciones software para dominios específicos, y la mejora de la calidad del producto final. (Fuentes & Troya, 2004).

Por lo tanto, debemos de desarrollar en forma eficiente los patrones de diseño y los patrones arquitectónicos para que el Marco de Trabajo permita desarrollar soluciones robustas a problemas complejos y con calidad.

## **DESARROLLO**

Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, y describe la esencia de la solución a ese problema, de tal modo que pueda utilizarse esta solución un millón de veces más, sin siquiera hacerlo de la misma manera dos veces. Sabemos que en la mayoría de las instituciones Educativas principalmente escuelas públicas como es la del Instituto Tecnológico Superior de Pánuco, necesitan mantener informado a la comunidad estudiantil con respecto a los aspectos administrativos de su trayectoria académica, por lo que se debe de llevar a cabo aplicaciones que puedan ser instaladas de manera sencilla y eficiente.

Un patrón es: una solución a un problema en un contexto particular recurrente (lo que hace la solución relevante a otras situaciones) enseña (permite entender cómo adaptarlo a la variante particular del problema donde se quiere aplicar) tiene un nombre para referirse al patrón. Los patrones facilitan la reutilización de diseños y arquitecturas software que han tenido éxito. (Alexander, 1977).

Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias). Como objetivo de estos patrones de diseño es la agrupación de la colección de soluciones en el contexto de soluciones diseñadas y validando sus argumentos que se han aplicado con éxito en otras ocasiones. Los patrones de diseño son soluciones de sentido común que deberían formar parte del conocimiento de un diseñador experto. Además, facilitan la comunicación entre diseñadores, pues establecen un marco de referencia

(terminología, justificación). Además, facilitan el aprendizaje al programador inexperto. (Francisco José García Ruiz, 2008).

Los objetivos que ofrecen los patrones de diseño son los siguientes: Reducción de tiempos, disminución del esfuerzo de mantenimiento, aumentar la eficiencia, asegurar la consistencia, aumentar la fiabilidad, proteger la inversión en desarrollos.

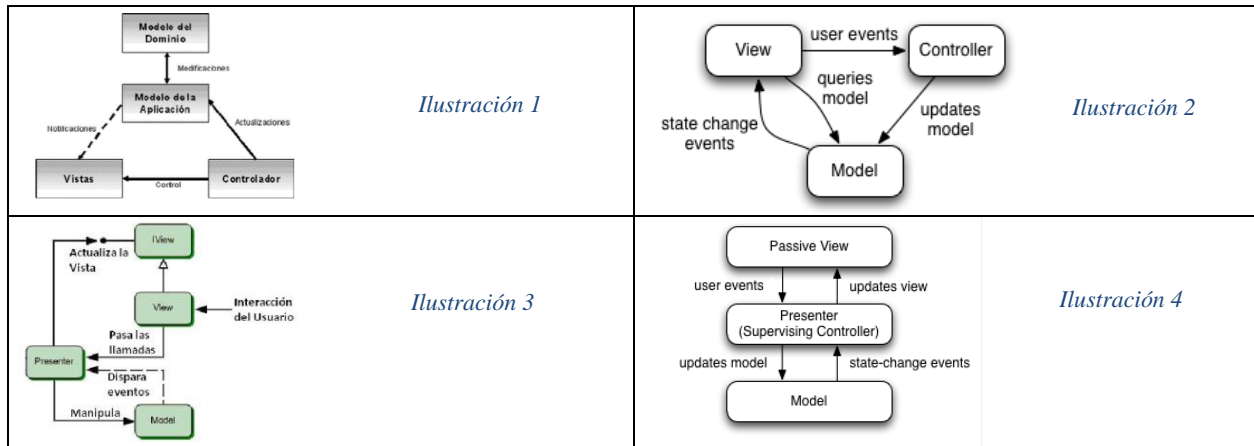
En la presente investigación sobre el desarrollo basado en MVC, se va a describir la forma en que trabajan los patrones de diseño y posteriormente se va a llevar a cabo un análisis para determinar los requerimientos que se van a utilizar en patrón de diseño del Sistema Web para información académica para las instituciones educativas de nuestro país.

MVC es un patrón de diseño que considera dividir una aplicación en tres módulos, que se identifican en forma clara y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador.

**El modelo:** Son clases que expresan del mundo real la base de datos que se requiere procesar, por ejemplo, un sistema de Nominas de recursos humanos tendrá un modelo que representa el Sueldo, pagos realizados, faltas etc. Es una forma de expresar los datos que se estarán manejando sin importar como se mostrarán y sin tener ninguna relación con cualquier otra entidad de la misma aplicación.

**Las vistas:** Las vistas de la misma manera que los modelos son clases, estas se encargan de mostrar por medio de código de HTML la información al usuario de la información obtenida de los modelos. Por lo general las vistas esta relacionadas a un modelo, pero dependiendo de la orientación de actividad, puede una vista estar relacionada a dos o mas modelos. Si ejemplificamos una vista mostrando valores como la lista de valores, otra vista se muestran valores de otro aspecto. La información de una vista la obtiene del modelo solamente la información que es necesario desplegar y la actualización es realizada cada vez que existe una notificación generada por el modelo de la aplicación.

**El controlador:** Es una clase especial que tiene funciones especiales como la de controlar el flujo de la aplicación por medio de mensajes externos, así de datos introducidos por medio de la URL o por medio de la misma página web o por opciones de menú dadas por el usuario. A partir de mensajes, el controlador puede encargarse de modificar el modelo o de cerrar o abrir vistas. El controlador como su nombre lo indica controla además en acceso al modelo ya las vistas, sin embargo, el modelo y las vistas no conocen la existencia del controlador. (Bascón, 2004)



Definiendo el patrón Modelo Vista Presentador MVP tiene una derivación del tradicional patrón MVC. Como se verá Ilustración 3 esta variante del patrón busca un desacople de mayores características con la interfaz de usuario y así mismo con la lógica de la propia aplicación.

La ejecución del patrón modelo vista controlador presenta se puede ver en la Ilustración número 4: El modelo donde se centra el modelo de negocios es tal, que la vista mostrará la tecnología de interface de usuario a la vez que el presentador será encargado de desacoplar la comunicación que se tiene entre el Modelo y la Vista. Un detalle de suma importancia es que al implementar este tipo de patrón será tal el desacople entre tecnología de interfaz de usuario y lógica, que se puede desarrollar una interface que intermedio la vista con la definición del presentador para acudir y consumir sus acciones, sin embargo nunca existirá un medio de información entre ellos ya que se declara la tecnología de interface del usuario en la interface., (González, 2014).

Modelo-Vista-Presentador (MVP) es un patrón que surge para auxiliar a la realización pruebas automáticas de las interfaces gráficas, para ello se ideó la codificación del interfaz de usuario lo más simple posible, contando con un menor código y realizando las mismas funciones, Disminuyendo considerablemente las pruebas del sistema. En vez se cuenta con una lógica de la interfaz de usuario, y se realiza en una clase separada, esto se conoce como presentador, que no depende de los componentes de la interfaz gráfica por lo que es más fácil la realización de pruebas.

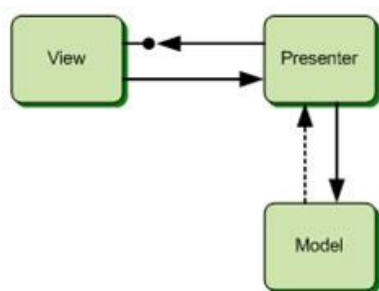
Ya contando con este patrón, el diseño se realizará con las pruebas necesarias, para poner un ejemplo, se cuenta con un sistema de control de personal en formato WEB que permita verificar los resultados de la aplicación así poder validar con los directivos de la empresa antes de poner en marcha la aplicación.

La clase Presentador presente una idea básica, la de hacer de intermediario entre la Vista, hablando como interface grafica y el modelo de datos. La vista contará con suficientes métodos en los que le pasan exclusivamente los que valores que se deben

meter y los datos en los componentes gráficos, como Botones, Cuadros de texto, etc. generan de la misma forma métodos GET para obtener el contenido de esos componentes.

Como ya se ha planteado el presentador tiene como función el entablar un enlace entre el modelo y la vista, y concederá de inteligencia a la vista. Como la meta es poder probarlo fácilmente, el Presentador recoge las interfaces que deben implementar el modelo y la vista, con los métodos públicos a los que el Presentador debe llamar.

En las siguientes ilustración 5 se observa claramente la forma en que se desarrolla cada uno de los patrones de diseño y la interacción con sus componentes.



Los expertos del grupo de GoF catalogaron los patrones de Diseño clasificación de la siguiente manera (Margain Fuentes, 2009) .

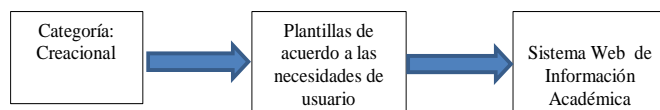
**Creacionales:** Patrones creacionales tratan con las formas de crear instancias de objetos. El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.

**Estructurales:** Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos.

**Comportamiento:** Los patrones de comportamiento nos ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos. (Tedeschi, 2009).

**Análisis:** De acuerdo al análisis realizado se pretende desarrollar un patrón de diseño del Ensamblado de un sistema Web de información académica que pueda ser utilizado en cualquier institución educativa que pueda ser utilizado para adaptarse a diferentes ambientes de trabajo. A continuación, se describe los elementos que contendrá el Patrón de diseño:

Ilustración 6



Las tecnologías que se utilizarán para el desarrollo del patrón de diseño del sistema web de información académica son: El lenguaje de un patrón se define como "La especificación de una serie de elementos (patrones) y sus relaciones (con otros patrones) de modo que nos permiten describir buenas soluciones a los diferentes problemas que aparecen en un contexto específico." Si bien notaciones como UML o los diagramas de flujo ayudan a explicar un determinado patrón, no constituyen un lenguaje de patrones (Tello, 2009).

El lenguaje unificado de diagrama o notación (UML) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros.

**Spring** Es parte de JAVA para desarrollo de aplicaciones empresariales es un framework muy poderoso. Gracias a Spring podremos centrarnos en resolver la lógica de negocio de nuestra aplicación mientras nos despreocupamos de las interconexiones y el funcionamiento a bajo nivel de nuestra arquitectura. Spring une la aplicación desarrollada en ambiente web de las aplicaciones puramente de escritorio, por ofrecer grandiosos beneficios al no tener la necesidad de modificar el código para disfrutar de las funcionalidades y grandes beneficios que ofrece el IDE Spring.

Esto indica que las dependencias se introducen en el objeto que las necesite, ya que, de esta forma, el objeto no debe estar ocupado en crear esas instancias. Así mismo al utiliza la AOP aplica aspectos al código que se basa en incluir más funcionalidad extra a las clases además de métodos sin alterar tanto el comportamiento como el código original del método o clase (Walls, 2011). Hoy en día, más y más desarrolladores quieren escribir aplicaciones transaccionales distribuidas para las empresas y aprovechar la velocidad, la seguridad y la fiabilidad de la tecnología del lado del servidor. Si ya se encuentra trabajando en esta área, sabrá que, en el rápido mundo de la tecnología de la información y el comercio electrónico, las aplicaciones empresariales tienen que ser diseñadas, construidas y producidas por menos dinero, con más velocidad y con menos recursos que antes.

**Java EE:** El Java EE de Oracle es muy utilizado para desarrollar aplicaciones puramente empresariales estas aplicaciones se dividen en capas dependiendo de su funcionalidad como pueden ser: Componentes de la capa de cliente corriendo en la máquina cliente. Componentes de la capa web corriendo en el servidor Java EE. Componentes de la capa de negocio corriendo en el servidor Java EE.

Software de la capa EIS (Enterprise Information System) corriendo en un servidor EIS, con bases de datos, Java EE en la actualidad el lenguaje de programación es uno de

los lenguajes que están siendo utilizados por los programadores para las aplicaciones Web por lo que se va a utilizar en el diseño del mismo. La utilización de frameworks y patrones de diseño mejorar sustancialmente la aplicación, por este motivo se propone el uso de Hibernate, una herramienta que permite realizar el mapeo de una base de datos relacional al paradigma de la programación orientada a objetos, esto facilitará el acceso a los datos, ya que, una tabla en la base de datos puede ser vista como una clase; por otro lado Spring, un framework que permite implementar de forma eficiente el patrón de diseño MVC, el cual como su nombre lo indica, está compuesto por el modelo, la vista y el controlador.

## **CONCLUSIONES**

En esta investigación se realizaron estudios para ver aspectos importantes que se debe de tomar en cuenta en la creación de patrones de diseño en los marcos de trabajo, para construir un sistema web de información académica para instituciones educativas. En la actualidad existen muchos sistemas de este tipo solo que solo son acoplados a una determinada situación y no tienen forma de que el usuario modifique el entorno, solo el programador lo puede realizar.

Se llevó a cabo un estudio sobre el modelo MVC Y MVP con la finalidad de resaltar los aspectos que se deben considerar al desarrollar el sistema web para información académica de Instituciones Educativas. La idea de este proyecto es que el usuario pueda mejorar los aspectos de diseño del Sistema Web. Como se puede observar se llevó a cabo un estudio detallado de las tecnologías de Java EE, Spring, Hibernate, de los cuales se determina que son muy eficiente en el manejo de Páginas Web además de la seguridad que poseen el uso de los recursos, así como, la optimización de los mismos.

Se analizaron las tecnologías que van a ser utilizadas para la creación de los patrones de diseños considerando los marcos de trabajo, así como, la forma en que va a ser representado los requerimiento del usuario, como es el caso de UML, el cual será utilizado para representar los requerimientos del usuario, la tecnología Spring analizando las ventajas que posee esta herramienta, además, de el manejador de Base de Datos que se va utilizar en la realización de dicha aplicación, con la finalidad de que el programador haga conciencia y las utilice cuando desarrolle el sistema web y así pueda disminuir los costos en la realización del mismo, de tal forma que pueda ser presentado a las autoridades educativas y desarrollado con la mayor calidad posible.

## REFERENCIAS

- Alexander, C. I. (1977). *Pattern Language. Towns. Buildings. Construction*. New York, NY.
- Bascón, E. (2004). El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing. *Academia Edu. Acta Nova Vol.2*, 8.
- Castro, P. (2012). *Actualidad TIC*. México. D.F.: Revista del Instituto Tecnológico de Informática.
- Daniele M, R. D. (2004). CONSTRUCCIÓN DE UN FRAMEWORK PARA LA ENSEÑANZA DE. *Universidad Politécnica de Valencia revista electrónica*, 1.
- Francisco José García Ruiz, J. F. (2008). Estudio, Comparativa y Aplicación Práctica de Metodologías de. Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/619/1/00804tfc.pdf>
- Fuentes, L., & Troya, J. M. (2004). *Desarrollo de Software Basado en Componentes*. Malagá.
- González, G. (2014). [http://www.ciberaula.com/articulo/disenio\\_patrones\\_j2ee](http://www.ciberaula.com/articulo/disenio_patrones_j2ee). Obtenido de Patrones de Diseño en aplicaciones Web: [http://www.ciberaula.com/articulo/disenio\\_patrones\\_j2ee](http://www.ciberaula.com/articulo/disenio_patrones_j2ee)
- Google. (2014). *Google Apps for Education*. Recuperado el agosto 2014, de <http://www.google.com/intx/es-419/enterprise/apps/education/products.html>
- Margain Fuentes, M. d. (2009). Metodología de Aprendizaje Colaborativo. *Investigación y Ciencia*, 20-28.
- Moreno, A. M. (2003). Patrones de Usabilidad: Mejora de la Usabilidad del Software desde el. En M. I. Sánchez. Madrid: In JISBD.
- Oracle Corporation. (2013, 03 07). *Oracle*. Recuperado el 01 28, 2015, de Conozca más sobre la tecnología Java: <https://www.java.com/es/about/>
- Pestalozzi, J. H. (1988). *cartas sobre educacion infantil*. tecnos.
- Red, J. a. (s.f.). *PLATAFORMA DE APLICACIONES*. Recuperado el 01 30, 2015, de [http://es.redhat.com/pdf/jboss/JBoss\\_Ent\\_app\\_platform\\_ES\\_web.pdf](http://es.redhat.com/pdf/jboss/JBoss_Ent_app_platform_ES_web.pdf)
- Tedeschi, N. (2009). Patrones de Diseño. Microsoft Developer Network. *Uruguay*.
- Tello, J. C. (2009). *Patrones de diseño: ejemplo de aplicación en los Generative Learning Object*. Obtenido de UNIVERSIDAD DE MURCIA: <https://www.um.es/ead/red/M10/caceres.pdf>
- The Java EE 6 Tutorial*. (s.f.). Recuperado el 01 26, 2015, de docs.oracle.com: <http://docs.oracle.com/javaee/6/tutorial/doc/bnaay.html>
- Walls, C. (2011). *Spring in Action. Spring in Action. Third Edition. Graig Waslls Maning*. Manning.