**MODELO – VISTA - CONTROLADOR**

**RESUMEN EJECUTIVO:**

Es un patrón de diseño de software en torno a la interconexión de los tres tipos de componentes principales en un lenguaje de programación como PHP, a menudo con un fuerte enfoque en la programación orientada a objetos (POO). Estos tres tipos de componentes son vagamente llamados modelos, vistas y controladores.

**PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA:**

* **IDENTIFICACIÓN DEL PROBLEMA**

Fuimos a hacer una investigación a un Centro de Salud, en el cual el problema era que había mucha cola al pagar, que los usuarios iban a separar citas y que no tenían un sistema web.

* **PLANTEAMIENTO DEL PROBLEMA**

Luego de haber identificado el problema y en dicho análisis encontramos la falta de soluciones. Planteamos una solución para optimizar y mejorar el servicio en el sector de salud.

Desarrollo de un software aplicando el patrón MVC, para tener el control y el orden en lo que se realizan las citas por paciente. Se evitaría el desorden y las colas en el C.S.

* **COMO IMPLEMENTAR LA SOLUCIÓN**

Implementar el software y utilizar el patrón MVC junto con una base de datos, para guardar los registros de cada atención, satisfecha, incompleta e insatisfecha.

* **JUSTIFICACIÓN DE LA INVESTIGACIÓN**

La presente investigación se enfocará en el gran desempeño y la solidez con la cual podamos utilizar nuestro sistema favoreciendo a los usuarios que trabajan con este patrón MVC.

**DEFINICIÓN DE LOS OBJETIVOS DE LA INVESTIGACIÓN:**

* **OBJETIVO GENERAL**

El objetivo general del patrón mvc es dar soporte a los modelos funcionales y mapas mentales de la información relevante para los usuarios, permitiendo un modelo que facilite la consulta y manejo de los mismos.

* **OBJETIVOS ESPECÍFICOS**

**-**Conocer y aplicar apropiadamente los patrones utilizados en el sistema.  
**-**Explicar y manejar apropiadamente la arquitectura lógica.  
**-**Mostrar la arquitectura utilizando los diagramas de paquetes de UML.

**ALCANCES Y LIMITACIONES DE LA INVESTIGACIÓN:**

* **ALCANCES**

Ser un modelo utilizado al 100% por todos los programadores.

* **LIMITACIONES**

Debe existir por lo menos un caso de uso en donde podamos aplicar nuestro patrón MVC.

**FUNDAMENTO TEÓRICO:**

* **MARCO TEÓRICO**

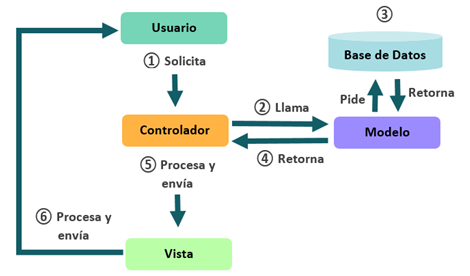
**Patrón: Modelo – Vista – Controlador (M.V.C)**

MVC es un patrón de diseño de software en torno a la interconexión de los tres tipos de componentes principales en un lenguaje de programación como PHP, a menudo con un fuerte enfoque en la programación orientada a objetos (POO). Estos tres tipos de componentes son vagamente llamados modelos, vistas y controladores.

**El modelo** es donde se guarda toda la lógica del negocio de una aplicación, la lógica del negocio puede ser cualquier cosa específica acerca de cómo una aplicación almacena los datos, o utiliza servicios de terceros con el fin de cumplir con sus necesidades.

**La vista** es donde se encontrarán todos los elementos de la interfaz de usuario de una aplicación, esta puede contener código HTML, hojas de estilo CSS y archivos Javascript. Cualquier cosa que el usuario pueda ver, es guardado en la vista.

**El controlador** es el componente encargado de conectar el modelo con la vista. Los controladores aíslan la lógica del negocio de un modelo de los elementos de la interfaz de usuario de una vista y maneja la forma en la que la aplicación responde a la interacción del usuario en la vista.



* El usuario realiza una petición.
* El controlador captura la petición del usuario.
* El controlador llama al modelo.
* El modelo interactúa con la base de datos, y retorna la información al controlador.
* El controlador recibe la información y la envía a la vista.
* La vista procesa la información recibida y la entrega de una manera visualmente entendible al usuario (3).
* **VENTAJAS**

**-**La implementación se realiza de forma modular.  
**-**Sus vistas muestran información actualizada siempre. El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.

**-**Cualquier modificación que afecte al dominio, como aumentar métodos o datos contenidos, implica una modificación sólo en el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.  
**-**Las modificaciones a las vistas no afectan al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.

* **DESVENTAJAS**

**-**MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.

**-**MVC requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los módulos de una aplicación.

**-**La cantidad de archivos que se deben mantener incrementa considerablemente.

**MARCO METODOLÓGICO:**

* **IMPLEMENTACIÓN DE LA SOLUCIÓN PLANTEADA**

Nuestra Implementación para la solución planteada fue crear un software aplicando el patrón MVC para el hospital Sergio E. Bernales, en donde las personas o trabajadores puedan ver los horarios y registrar una cita Vía web, así evitando ir al mismo hospital a sacarlo.

Otra implementación fue utilizar una base de datos para guardar los registros de cada atención que se da al registrarse en dicho hospital.

* **CONCLUSIONES**

MVC es un patrón de diseño enfocado a separar las responsabilidades dentro de nuestra aplicación y es muy utilizado en la web por su enfoque y las ventajas que ofrece con respecto a algunas otras formas o patrones de desarrollo de aplicaciones web.

* **RECOMENDACIONES**

**-**Utilizar MVC como patrón de arquitectura de software ya que permite separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos

**-**Utilizar herramientas conocidas y versiones estables, que posean documentación de apoyo.

**-**Programar siempre pensando en Servicios.

* **REFERENCIAS BIBLIOGRÁFICAS:**

**-** Ramírez Lituma, Abraham. Informe general sobre Patrón MVC.[\*](http://www.fao.org/docrep/field/003/AC587S/AC587S15.htm#refanx10a) Modelo Vista-Controlador. Perú, 1981. 40 p.

**-** Implementación del Patrón MVC. Diseño web Perú. Resultados y Recomendaciones del Proyecto. Proyecto RLA/72/048 “Patrón MVC”.

**-** Juan Antonio. Comisión de Factores recomendables para el (Patrón MVC). Circular. 1980. Proyecto RLA/72

**-** Vergara Vallejo, Juan. Ventajas y Desventajas del Patrón MVC. Informe general. Perú, 2014. 20 p.

* **ANEXOS:**

**-** López Atto Juan .Informe general sobre Modelo vista controlador PHP,HTML,JAVA SCRIPT.[\*](http://www.fao.org/docrep/field/003/AC587S/AC587S15.htm#refanx10a) Aspectos -administrativos. Proyecto INT/79/019 “Modelo Vista Controlador” Perú, 2005. 40 p.

**-** Diseño de Patrones para Proyectos de Software. Desarrollo web. Resultados y Recomendaciones del Proyecto. Proyecto RLA/72/048 “Patrón MVC”.

**-** Morales R., Francisco. Las Ventajas y Desventajas del Patrón MVC. Perú, 1975.