El presente manual está dirigido a los programadores que deseen continuar realizando mejoras a la plataforma PROMED, se detalla las acciones y procesos a los cuales tienen acceso y se da un detalle gráficamente

**Modelo Vista Controlador (MVC)**

Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

• El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

• La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.

• El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

**El modelo:** Es el responsable de acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento. Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor". Lleva un registro de las vistas y controladores del sistema. Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.)

**El controlador:** Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.). Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar".

**Las vistas:** Son las responsables de recibir datos del modelo y los muestra al usuario. Tienen un registro de su controlador asociado (normalmente porque además lo instancia). Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes)

**Ventajas de utilización del MVC**

Una separación total entre lógica de negocio y presentación. A esto se le pueden aplicar opciones como el multilenguaje, distintos diseños de presentación, etc. sin alterar la lógica de negocio. La separación de capas como presentación, lógica de negocio, acceso a datos es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y más fácilmente mantenibles, lo que al final resulta en un ahorro de tiempo en desarrollo en posteriores proyectos. Al existir la separación de vistas, controladores y modelos es más sencillo realizar labores de mejora como:

• Agregar nuevas vistas.

• Agregar nuevas formas de recolectar las órdenes del usuario (interpretar sus modelos mentales).

• Modificar los objetos de negocios bien sea para mejorar el performance o para migrar a otra tecnología.

• Las labores de mantenimiento también se simplifican y se reduce el tiempo necesario para ellas. Las correcciones solo se deben hacer en un solo lugar y no en varios como sucedería si tuviésemos una mezcla de presentación e implementación de la lógica del negocio.

• Las vistas también son susceptibles de modificación sin necesidad de provocar que todo el sistema se paralice. Adicionalmente el patrón MVC propende a la especialización de cada rol del equipo, por tanto en cada liberación de una nueva versión se verán los resultados.

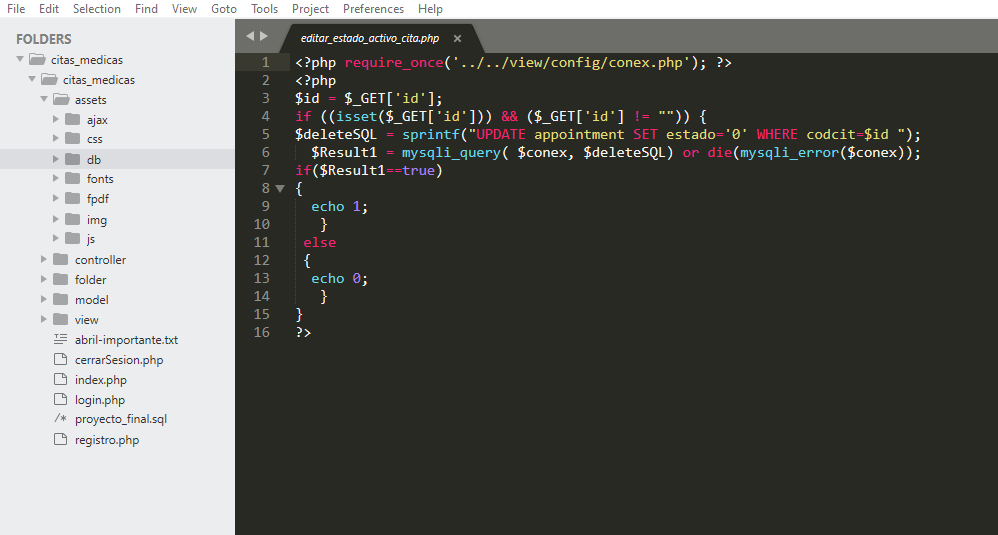
**Orígenes del MVC**

El estilo fue descrito por primera vez en 1979 por Trygve Reenskaug, entonces trabajando en Smalltalk en laboratorios de investigación de Xerox.

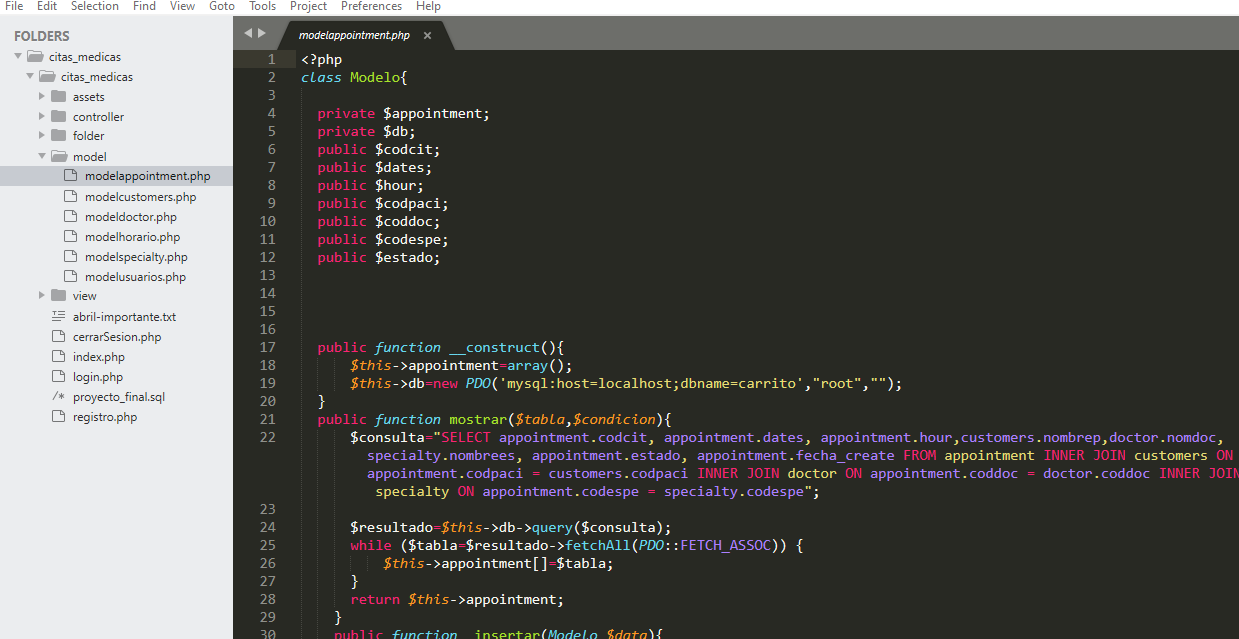
**MYSQL**

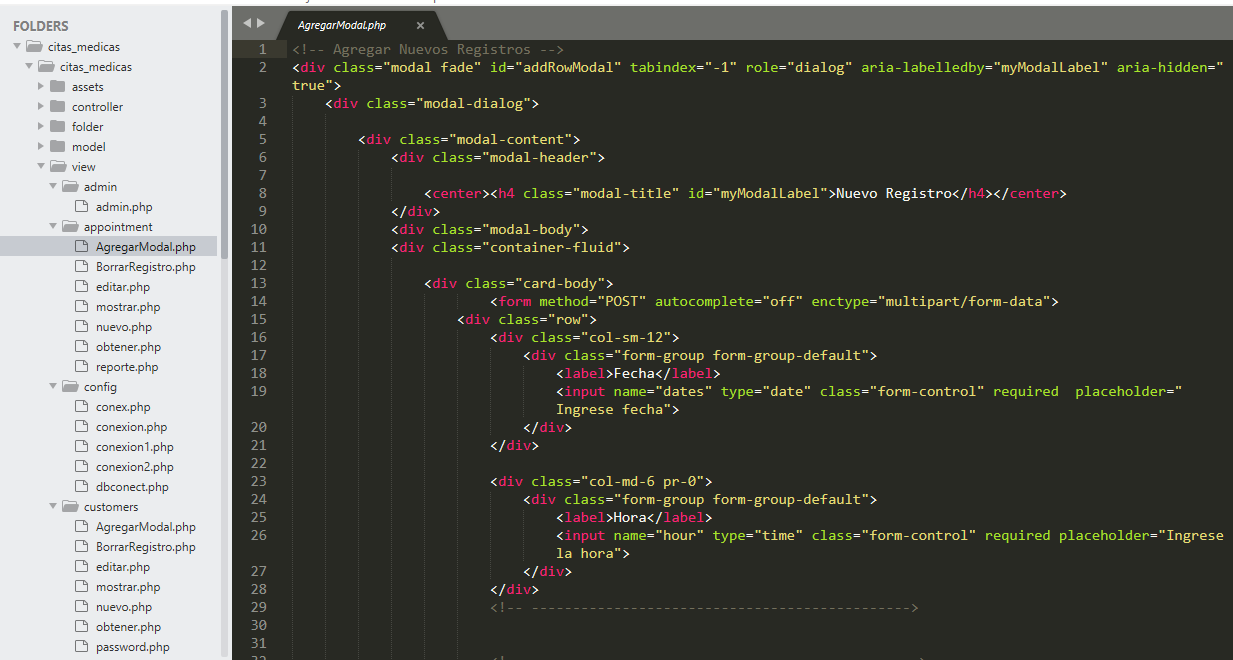
Es uno de los sistemas de administración de bases de datos relacional más popular de la actualidad, de código abierto, usa el estándar SQL Structure Query Language (Lenguaje de Consulta Estructurado), el cual es un lenguaje de programación para bases de datos relacional como Oracle, Sql server, Db2, Postgres etc. Este motor de base de datos permite crear bases de datos, tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas y realizar muchas operaciones, etc. MYSQL se ejecuta en diferentes plataformas tales como: Linux, Windows, Mac OS, Solaris, AIX de IBM etc.

Adentro de la carpeta Access se detalla todos los recursos que se utilizaran: CSS, Image, JS, Fuentes



Tomando en cuenta que se ha utilizado el modelo MVC para tener un mayor orden y compresión del código en la carpeta Core // App se encuentra el Modelo y Vistas





Siguiendo MVC se tiene la parte del controlador en la carpeta Core // Controller

