

Escuela de Informática

Curso: Desarrollo de aplicaciones web

EIF 508

Profesora: Gregorio Villalobos Camacho

Desarrollo en tres capas

Carlos Arguedas

Bidkar Solís

II Ciclo, 2019

# 

**Introducción**

En un mundo el cual la tecnología se desarrolla de manera vertiginosa, la búsqueda del desarrollo de procesos se convierte en parte importante para el cumplimiento del trabajo. Y el área de software no se escapa de ello.

Los procesos de desarrollo de software con las metodologías actuales de Agile ayudan tanto al proceso de desarrollo, como el de mantenimiento. Uno de los procesos que estudiaremos en este documento es la arquitectura por capas.

El es un modelo de desarrollo software con el objetivo principal de la separación (desacoplamiento) de las partes que componen un sistema software: lógica de negocios, capa de presentación y capa de datos.

De tal manera que los cambios en una de las capas interfiera solo a los elementos involucrados en los procesos que soporta. Brindando flexibilidad, agilidad y mantenimiento rápido y eficiente al existir una separación de responsabilidades entre las partes.

Capa de presentación es la capa a simple vista del usuario, presenta el sistema al usuario, comunica la información y captura la información del usuario. También se conoce como interfaz gráfica y debe tener la característica de ser amigable con el usuario. La capa solo se debe comunicar con la capa de negocios.

Capa de negocio es en donde residen a funcionalidad de las operaciones del usuario con el sistema y se envían las respuestas a la capa de vista al final del proceso solicitado. Se denomina capa de negocio debido a que acá es donde radica la lógica del sistema y las reglas que deben cumplire. Esta capa se comunica tanto con la capa de presentación como con la de datos.

Capa de datos es donde residen los datos y es la encargada de acceder a los mismos. Formada por la estructura de la base de datos y encargada de comunicarse con el sistema gestor de la base de datos para almacenar o procesar las solicitudes.

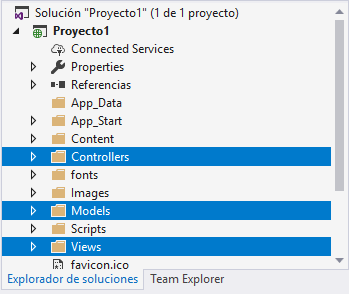
Como ejemplo a desarrollar se presenta un proyecto basado en la tecnología ASP.net con el lenguaje de porgramación C# que soporta el modelo de capas mediante un patrón llamado Modelo -Vista - Controlador.

**Aplicación basado en modelo de tres capas con ASP.Net, usando Modelo-Vista-Controlador**

Para el desarrollo de aplicaciones web la tecnología de Microsoft desarrolló su sistema ASP.net. En el cual con tecnologías y herramientas de Microsoft como IIS como servidor de aplicaciones, .net Core como su librería de desarrollo y Microsoft Visual Studio crea un ambiente completo para desarrollo de aplicaciones.

Como empresa en tecnología este debe de adaptarse a las distintas mejores prácticas de desarrollo y aplicación de arquitecturas. Para el modelo de tres capas en ASP.net tiene un estlo de proyecto llamado MVC el cual pre-configura un ambiente de trabajo con las configuraciones básicas.

Esta configuración básica tomo en cuanta la creación de tres carpetas principales como lo son: Modelo, Controlador y Vistas. Como su nombre lo refiera, en cada carpeta van los archivos con sus respectivos códigos para desarrollar una aplicación por capas.

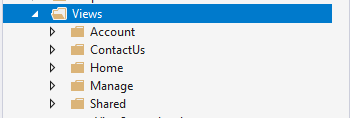


En la carpeta controlador se encuentra los archivos de la capa de negocios, en la cual está el código encargado para el control de la lógica del sistema, acá se encuentran las clases en la cual encontramos los códigos que se comunican con la vista y envía los procesos a la base de datos para crear, modificar, actualizar o eliminar acorde a lo que el cliente indica en las vistas.

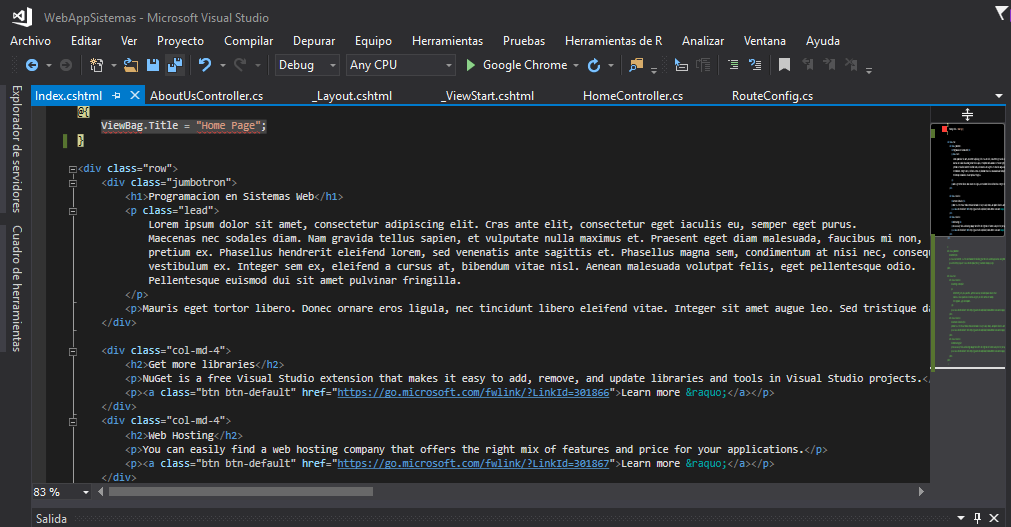
A continuación como se representa el modelo para llenar un formulario de contacto y su comunicación entre el modelo, la vista y el controlador:

Las parte más sencilla del sistema de capas, es la capa de vista en la cual se define la visualización que se presenta al usuario final para la interacción con la lógica del negocio y la base de datos.

En esta capa se presentan las vistas tales como la de home, login, acerca de, contactus, servicios y las demás vistas partes del sistema. En ASP.net se crea una carpeta que contendrá las vistas relacionadas al controlador.



Bajo la carpeta Home, se encuentra un archivo de código Index, el cual es el que inicia la presentación y en el cual se inserta el código HTML, en este caso como aplicación web.



**Agregación de un carrito de compras**

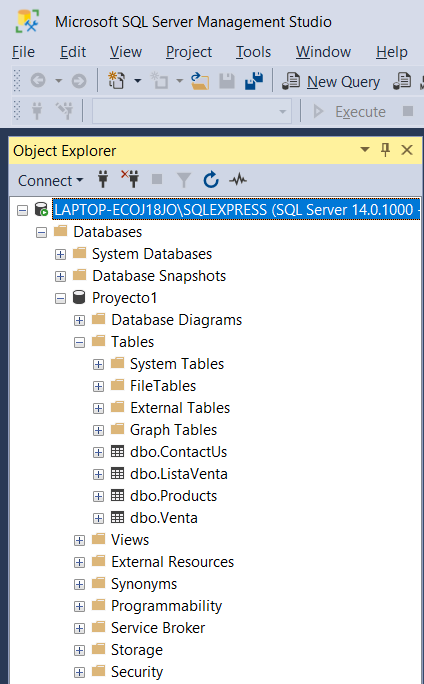
Para elaborar un carrito de compras o cualquier entidad que se desea preservar, conocido como persistencia de datos, se debe de tener un modelo de base de datos. Para nuestra facilidad con MVC .Net Framework Microsoft ayuda mediante una tecnología de mapeo con Entity Framework el cual es un tipo de ORM que mapea las entidades de nuestra base de datos con los modelos de nuestros programas.

Para el proyecto se plantea la elaboración de un carrito de compras para lo cual vamos desarrollamos las siguientes entidades:

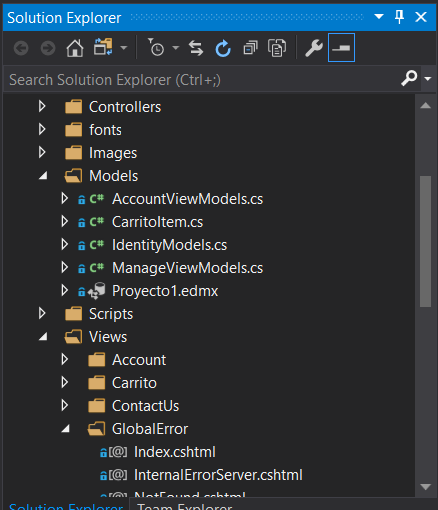
Producto: el cual es la entidad con los atributos para cada producto que podamos realizar las compras

Venta: el cual es el carrito como tal, registra la operación de la venta con el nombre del producto, y el cálculo del costo

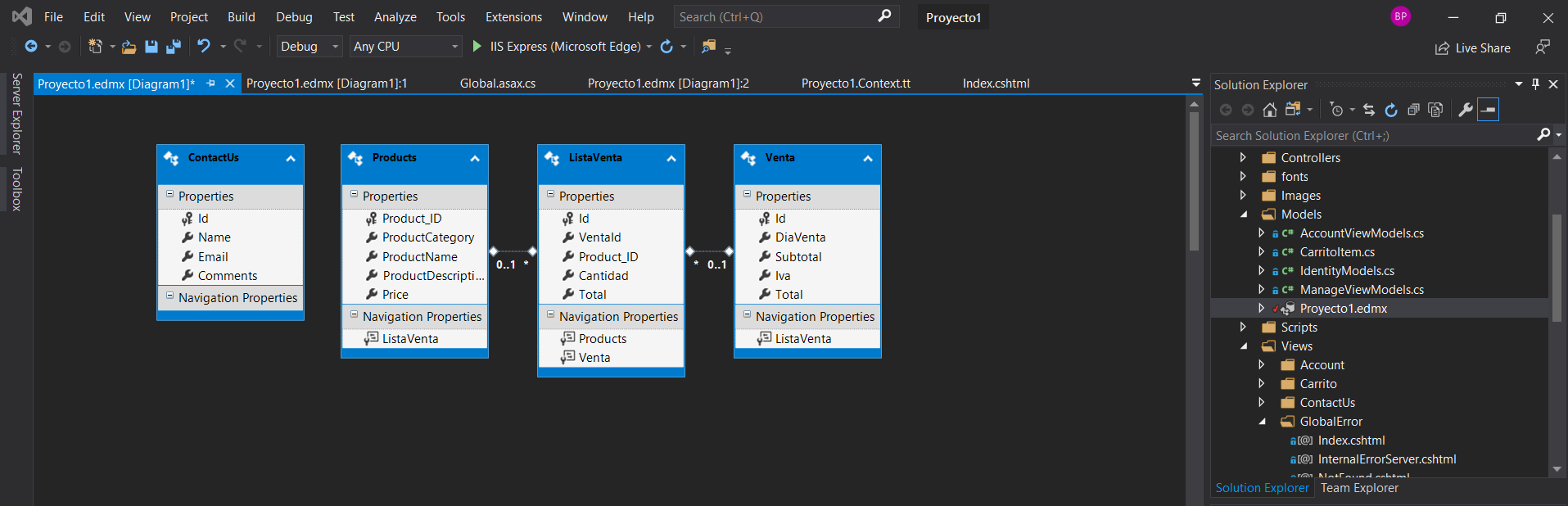
ListaVenta: es el histórico de las ventas realizadas que se va mostrar para obtener los detalles de las ventas



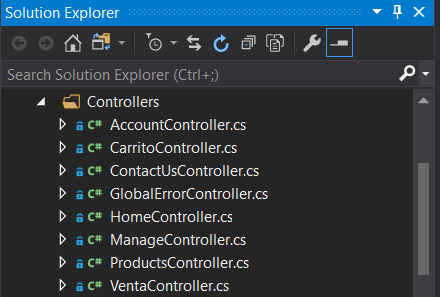
Una vez desarrollados las entidades en la base de datos en nuestro proyecto debemos de desarrollar los modelos, que son las representaciones de las entidades en manera de código, los cuales como lo indica el tipo de tecnología MVC, estos modelos los representa la letra M que son los modelos representativos de las entidades.



En la carpeta Models es donde se crean los códigos en el lenguaje C# para desarrollar los modelos, y podemos observar el modelo de la base de datos como Proyecto1 el cual tiene la representación de la base de datos.

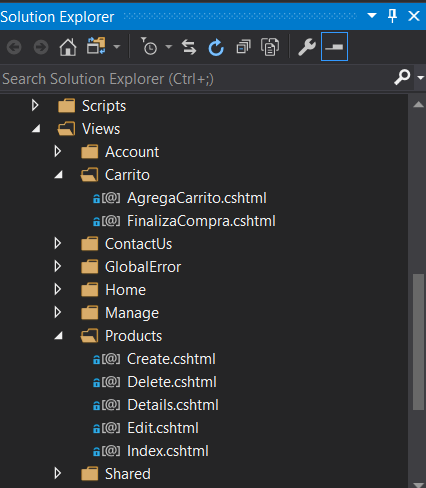


Estos modelos son unidos a un controlador el cual es el encargado de la interacción entre las vistas y el modelo con el cliente. Cuando una persona agrega un producto nuevo al sistema el controlador es el que decide cual vista presentar en conjunto con el ruteador. Y las acciones posibles son enviadas al controlador mediante la vista.



El controlador tiene las conexiones con la base de datos que le permite, haciendo referencia a la acción anterior de agregar, hacer la agregación a la entidad respectiva. En este caso ejecuta la acción para agregar un nuevo producto a la tabla de Productos.

Cuando un controlador es creado este solicita si se desea asociar a un modelo, y además de ello la creación de las vistas para el CRUD. Para lo cual crea un conjunto de carpetas que tienen el código de la representación de las vistas con las opciones CRUD.



**Dificultades**

El sistema para la elaboración de los modelos de entidad no es fácil de realizar si no se cuenta con la tecnología del marco de trabajo, en este caso cada vez que se inicializa un modelo existe una manera de aplicarlo, pero la tecnología de base de datos debe ser SQL Server.

Hay que contar con bastante claridad los principios básicos del Modelo-Vista-Controlador para no perderse entre tanto código modularizado y además de los códigos pre-hechos por el marco de trabajo para el manejo de la configuración.

Si bien los marcos de trabajo ayudan a desarrollar sistemas más rápidos, también causan que los programas sean un poco más densos y pesados con las ayudas de objetos preconfigurados y hay que depender de su base para continuar con el desarrollo propio.

**Conclusión**

El modelo de tres capas permite tener un proyecto de cierta manera limpio y ordenado, además de ello se acopla a las metodologías de ágiles en la administración de proyectos. Como es un sistema compuesto por porciones que se acoplan durante su desarrollo este método permite que se vaya desarrollando de manera iterativa.

En la parte de mantenimiento de la aplicación es un modelo muy fácil de manejar ya que permite que el cambio a un cierto artefacto el impacto sea menor, ya que el desacoplamiento busca el mínimo impacto ante cambios requeridos.

El modelo de capas brinda una flexibilidad y separación de responsabilidades que permite al desarrollador un mejor manejo y control del sistema y de los cambios. Además de que la manera de presentarlo a las unidades de negocios es más sencillo al ir desplegando cada capa en unidades lógicas.

El modelo de capas además aporta a las medidas de seguridad, cada capa tiene su seguridad implementada por aparte lo cual hace más fácil su desarrollo, monitorea y control. Al tener cada capa sus métodos de seguridad implica un poco más de dificultad para los atacantes y aporta capas de abstracción para la protección.