INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE ESTUDIOS CIENTÍFICOS Y TECNOLÓGICOS NO. 13
RICARDO FLORES MAGON

TÉCNICO EN INFORMÁTICA

“SISTEMA DE CONTROL DE ALMACÉN ‘ZAPA+’ ”

AUTOR:

RODRIGUEZ PALACIOS, JOSE ANGEL

TRUJANO ORTIZ, LUIS ANTONIO

Índice

Abstract

English

Inventory system

An inventory system is a management tool used to record the quantities of existing merchandise in a business, as well as to determine the cost of goods sold. Through an inventory control system it is possible to know what merchandise is available at a given time and what products are about to run out, as well as determine the rotation levels of the products and identify those close to reaching their expiration date.

The first thing that an inventory control system allows is to enter the quantity, purchase cost, and sale price of each of the products that make up the business stock into a database. In addition, and if necessary, the supplier from whom said merchandise was purchased. Each time new merchandise arrives at the warehouse, the respective entry of the quantities received must be made, so as not to generate inconsistencies between the actual stocks and those registered in the system.

The inventory is associated with the sales record. Therefore, each time a product is sold, the system discounts it from stock, entering the value of the sale. In this way, a continuous record of each item in the inventory is maintained, which allows you to see -at all times- the merchandise available and those that are close to running out. Undoubtedly, relevant information to simplify stock management and help in decision making.

When referring to an inventory system, it is necessary to remember that it is a tool that is included in a point of sale (POS) software. This contributes to ordering income and expenses, ensuring reliable and transparent inventory management, which results in better control of finances and, therefore, of profits.

Español

Sistema de inventario

Un sistema de inventario es una herramienta de gestión empleada para registrar las cantidades de mercancías existentes en un negocio, así como para determinar el costo de los productos vendidos. Mediante un sistema de control de inventarios es posible saber cuánta mercancía se tiene en determinado momento y qué productos están por acabarse,

así como determinar los niveles de rotación de los productos e identificar aquellos próximos a cumplir su fecha de caducidad.

Lo primero que permite hacer un sistema de control de inventarios es ingresar a una base de datos la cantidad, costo de compra, y precio de venta de cada uno de los productos que componen el stock del negocio. además y en caso de ser necesario el proveedor al cual se le compró dicha mercancía. Cada vez que llega nueva mercadería al almacén se debe realizar el respectivo ingreso de las cantidades recibidas, para no generar inconsistencias entre las existencias reales y las registradas en el sistema.

El inventario queda asociado al registro de ventas. Por lo tanto, cada vez que se vende un producto el sistema lo descuenta del stock, ingresando el valor de la venta. Así se mantiene un registro continuo de cada artículo del inventario, lo que permite ver -en todo momento- las mercancías disponibles y las que están próximas a agotarse. Sin duda, información relevante para simplificar la gestión del stock y ayudar en la toma de decisiones.

Al referirnos a un sistema de inventario, es necesario recordar que se trata de una herramienta que se incluye en un software de punto de venta (POS). Esto contribuye a ordenar los ingresos y egresos, asegurando una gestión del inventario fidedigna y transparente, lo que deriva en un mejor control de las finanzas y, por ende, de las utilidades.

1 Estudio de factibilidad

1.1 Introducción

Después de definir la problemática presente y establecer las causas que ameritan de un nuevo sistema, es pertinente realizar un estudio de factibilidad para determinar la infraestructura tecnológica y la capacidad técnica que implica la implantación del sistema en cuestión, así como los costos, beneficios y el grado de aceptación que la propuesta genera en la Institución. Este análisis permitió determinar las posibilidades de diseñar el sistema propuesto y su puesta en marcha, los aspectos tomados en cuenta para este estudio fueron clasificados en tres áreas, las cuales se describen a continuación

1.2 Factibilidad técnica

La Factibilidad Técnica consistió en realizar una evaluación de la tecnología existente en la organización, este estudio estuvo destinado a recolectar información sobre los componentes técnicos que posee la organización y la posibilidad de hacer uso de los mismos en el desarrollo e implementación del sistema propuesto y de ser necesario, los requerimientos tecnológicos que deben ser adquiridos para el desarrollo y puesta en marcha del sistema en cuestión. De acuerdo a la tecnología necesaria para la implantación del Sistema de Seguimiento y Control de las Investigaciones

Tabla 1.1

Hardware
Procesador de 64 bits de 1,8 GHz o más rápido; se recomienda uno de cuatro núcleos o superior. No se admiten los procesadores ARM.
4 GB de memoria, como mínimo. Hay muchos factores que afectan a los recursos usados, se recomiendan 16 GB de RAM para soluciones profesionales típicas.
Windows 365: 2 CPU y 8 GB de RAM, como mínimo. Se recomiendan 4 CPU y 16 GB de RAM.
Espacio en disco duro: mínimo de 850 MB y hasta 210 GB de espacio disponible, en función de las características instaladas; las instalaciones típicas requieren entre 20 y 50 GB de espacio libre. Se recomienda instalar Windows y Visual Studio en una unidad de estado sólido (SSD) para mejorar el rendimiento.
Tarjeta de vídeo que admita una resolución de pantalla mínima de WXGA (1366 x 768); Visual Studio funcionará mejor con una resolución de 1920 x 1080 o superior.

Tabla 1.2

Software
Windows 10, 11 x64
Visual studio 2022
Microsoft SQL Server 2019
SSMS 18.1
Entity Framework Core 3.1

La implementación de la página Web se hará en forma directa ya que no existe un sistema actual. Con las anteriores características técnicas se garantiza el óptimo funcionamiento del sistema.

Base de Datos: se usará el motor de base de datos de SQL server y se usará SQL Server Management Studio

Web: es un dominio de Internet de nivel superior, no oficial, Documento situado en una red informática, al que se accede mediante enlaces de hipertexto.

Hosting: Un host o anfitrión es un ordenador que funciona como el punto de inicio y final de las transferencias de datos. Más comúnmente descrito como el lugar donde reside un sitio Web. Un host de Internet tiene una dirección de Internet única (dirección IP) y un nombre de dominio único o nombre de host.

1.3 Factibilidad operativa

El sistema será administrado por el propietario y empleados bajo una clave personal. Los futuros usuarios del sistema deben tener conocimientos básicos de computación,se instruirá a cada usuario para que realice las funciones en el sistema, el vendedor de la zapatería deberá tener conocimiento básico de computación y encargado del ingreso de la mercadería, el propietario aparte de tener conocimiento básico de computación supervisa la

operación de entrada del producto, genera, analiza y toma decisiones en base al contenido generado por el sistema.

por parte del equipo de trabajo para la realización de este sistemas somos dos personas y nos dividimos al trabajo de manera igual el proyecto está planeado para ser finalizado a mediados de mayo como máximo y cada uno cuenta con el equipo de cómputo necesario para la realización de este proyecto, algo que no contamos de la manera que nos pudiera gustar sería el tiempo y la capacitación educativa.

Para la implementación del sistema se requerirá de un servidor web Host de hardware como son:

Tabla 1.3

Requerimientos
Almacenamiento en disco de 500gb
almacenamiento en memoria 10gb
70 mb de velocidad de descarga
7 mb de velocidad de subida
windows 10 u 11
SQL server

1.4 Factibilidad económica

A continuación se presenta un estudio que dio como resultado la factibilidad económica del desarrollo del nuevo sistema de información. Se determinaron los recursos para desarrollar, implantar, y mantener en operación el sistema programado, haciendo una evaluación donde se puso de manifiesto el equilibrio existente entre los costos intrínsecos del sistema y los beneficios que se derivaron de éste, lo cual permitió observar de una manera más precisa las bondades del sistema propuesto.

Luego de realizar un presupuesto se llegaron a las sig cifras

Concepto	Coste
Analisis, Diseño y Desarrollo	8,000.00
Capacitacion al personal	2,000.00
Transportes	80.00
Mantenimiento (en caso de ser necesario)	2,000.00
Hardware y equipo de computo	20,000.00
Total	**32,080.00**

Tabla 1.4

Cabe recalcar que al presupuesto se le restan 2,000.00 de el mantenimiento al comienzo de la vida del sistema pero por cada ciclo de mantenimiento se efectuará el cobro de la tarifa establecida en la tabla anterior

1.5 Costo de personal.

En este tipo de gasto, incluye los generados por el recurso humano, bajo cuya responsabilidad directa está la operación y funcionamiento del sistema y que se muestra en la siguiente tabla:

Recurso humano	Salario mensual	Salario anual
Analista de sistemas	*$10,904*	*130,848.00*
Asistente	*6,542.40*	*78,508.80*
Operadores	*5,888.16*	*70,657.92*
Total	*12,430.56*	*280,014.72*

Tabla 1.5

1.6 Análisis Costo Beneficio del Sistema Propuesto y el Sistema Actual

En la tabla que se muestra a continuación, se pueden visualizar los costos totales tanto del sistema actual como los del sistema propuesto a lo largo de la vida útil, que se estimó para un periodo de cinco (5) años.

Año	Sistema Actual	Sistema Propuesto
1	$420,022.08	$280,014.72
2	$840,044.16	$560,029.44
3	$1,260,066.24	$840,044.16
4	$1,680,088.32	$1,120,058.88
5	$2,100,110.40	$1,400,073.60

Tabla 1.6

1.7 Relación Costo-Beneficio.

El Análisis Costo-Beneficio presenta grandes ventajas para la Organización, ya que la misma cuenta con los recursos técnicos necesarios (hardware y software) para el desarrollo e implantación del nuevo sistema, por lo que no se hará erogación alguna en lo que a tecnología se refiere.

De igual manera, el nuevo sistema trae mejoras significativas para el normal desenvolvimiento de las actividades dentro de la Red de Investigación, reduciendo de esta manera el tiempo de procesamiento y generación de la información, disminuyendo las cargas de trabajo a los usuarios, ya que la velocidad de procesamiento, veracidad y confiabilidad de los procesos y resultados serán los deseados.

Una de las ventajas del sistema propuesto, es que los usuarios podrán plasmar sus necesidades a través del sistema, por lo que se podrá planificar el trabajo a ejecutar en el seguimiento y control, dando respuestas satisfactorias en un tiempo más breve.

Con la implantación del nuevo sistema automatizado, el beneficio más significativo que se adjudicaría la organización sería la información, convirtiéndose de esta manera en la herramienta más poderosa y versátil con que ésta cuente.

Es muy importante destacar que en esta nueva era de la informática, mejor conocida como la "Era de la Información", este recurso es la herramienta de competitividad más utilizada por las organizaciones, y en cualquier caso, tenerla al alcance y en forma oportuna, podría significar ahorro, tanto de tiempo como de dinero.

Además debe tomarse en cuenta el valor que la información tiene en los actuales momentos, siendo el punto de apoyo en el proceso de la toma de decisiones, las

organizaciones que han alcanzado el éxito, se debe en gran parte que esta han otorgado el verdadero valor que debe tener la información dentro de sus procesos.

Con la puesta en marcha de este proyecto se logrará optimizar los procesos que involucra la gestión de la información dentro de la Red Académica de Información Científica, reduciendo de esta manera el empleo de recursos, tanto materiales como humanos, permitiendo obtener una información segura y confiable, dirigida a la consecución de los objetivos y agilizar la toma de decisiones.

Por otra parte, un sistema de información debe contribuir a aumentar la capacidad, el control, la comunicación, disminuir los costos y obtener una ventaja competitiva. Esto recaerá en la disminución de actividades redundantes, proporcionando agilidad en el desempeño de las actividades a un gran número de las áreas involucradas.

Bajo este criterio la alternativa planteada para solucionar la problemática presente y mejorar la situación actual y cumplir con los objetivos de la investigación, es el Sistema de Información

1.8 Beneficios Tangibles

Los beneficios tangibles aportados por el sistema propuesto están dados por los siguientes aspectos:

Reducción de costos en papelerías, mantenimiento y espacio físico.

Ahorro en suministros para los equipos empleados.

1.9 Beneficios Intangibles.

Entre los beneficios intangibles del sistema propuesto se pueden incluir:

Optimizar las actividades dentro del Consejo de Desarrollo Científico y Humanístico, aumentando la productividad del personal que labora en el mismo, repercutiendo por ende en el funcionamiento

Un control y seguimiento de los activos del Consejo de Desarrollo Científico y Humanístico, que permite un mejor y más efectivo empleo de los recursos, tanto materiales como financieros.

La flexibilidad al manejar gran volumen y diversidad de información con rapidez, oportunidad y precisión,lo que ofrece una mejor herramienta de trabajo al personal, que facilitará sus labores.

Generar información más eficiente y confiable, que sirva de apoyo a la toma de decisiones.

Mejor capacidad de búsqueda y actualización de información, reduciendo la fuerza de trabajo en el proceso y control de recursos.

Crear una Sociedad de la Información, a través de la cual se logra la interacción directa de los investigadores, y el intercambio de conocimientos e información, y un mejor uso de recursos

Mayor y mejor aprovechamiento de los recursos tecnológicos instalados.

Capacidad de registrar y almacenar "automáticamente" datos de los registros, estandarizando el mantenimiento de los registros, lo que implica un aumento de la capacidad y seguridad de almacenamiento de registros.

2 Sistema Control de Almacén

2.1 Sistema Vista Cliente

Este sistema engloba la primera pantalla que se verá al entrar a nuestra página web teniendo en el header el nombre de la página web, nuestro carrito (que engloba nuestro sistema de ventas) y el inicio de sesión para el ingreso de clientes o en su defecto de trabajadores.

El body tiene nuestra sidebar de el lado derecho con las divisiones para la organización de los modelos, en el centro y el lado derecho contiene los productos con los que cuenta el almacén, estos podrán ser agregados al carrito para posteriormente ser comprados, contendrán una foto, nombre del modelo, una pequeña descripción y el precio de venta que estos tienen.

Para finalizar el footer tendrá la información de nosotros los programadores.

2.1.1 Sistema Ventas

Este sistema será el encargado de dar de baja los productos que sean vendidos en nuestra página web, estará conformado por el carrito donde se habrán guardado los productos que se quieren comprar y el sistema de pago para finalizar enviando un ticket de compra con su

respectiva información. Se podrá acceder a este sistema dando click en el carrito de compras que se encuentra en el header.

2.2 Sistema Inicio Sesión

Se podrá acceder a este sistema dando click a su respectivo botón desde la vista del cliente.

tendrá un botón de registro para el cliente donde se podrá guardar la información hecha por dicho cliente para mantener un control.Al inicior secion un cliente se le será regresado a la vista de cliente, mientras que por otro lado si inicia sesión un trabajador se le será enviado a la vista trabajador(este tipo de usuario solo podrá ser dado de alta si un administrador lo hace desde el sistema de usuarios en la vista del trabajador)

2.3 Sistema Vista Trabajador

En este sistema se engloba todo lo administrativo referente a nuestro sistema de inventariado, teniendo un header con el nombre de usuario que haya iniciado sesión y un botón para ingresar a la vista del cliente si es necesario.

en en body se tiene el menú donde tenemos las opciones que tiene esta vista donde se destaca el inventario, comprar, usuarios, proveedores haciéndonos ingresar al sistema elegido. Del lado derecho del menú tendremos el ara donde se mostrará la información.

Para finalizar el footer tiene la información de los programadores.

2.3.1 Sistema Almacen

Este sistema contiene el inventario con el que cuenta el sistema, teniendo como atributos de clase id, nombre, stock, precio compra, precio venta, categoría, división y una imagen (esta será la mostrada en los productos de la vista cliente). En la parte superior se tiene un botón que nos enviará al Sistema Compra Inventario.

de el lado derecho de cada producto se tienen tres opciones:

Editar.- este nos enviará a el Sistema Edición Inventario.

Eliminar.- este dará de baja definitivamente el producto de nuestro sistema.

Detalle.- este mostrará todos los detalles de el producto en cuestión teniendo como formato la vista mostrada en la Vista Cliente

2.3.1.1 Sistema Compra I

Este sistema contiene todos los datos necesarios para dar de alta un nuevo producto en el sistema de inventario (id, nombre, stock, precio compra, precio venta, categoría, división e imagen)

El usuario tiene que agregar uno por uno los valores deseados teniendo la opción de elegir una opción previamente agregada para su facilitación, esto con un pequeño menú que se desplegará al seleccionar el tado que se quiere agregar.

Al final se tiene dos botones:

Agregar.- este confirma el ingreso de el nuevo producto, finalizando este proceso el usuario será regresado a el Sistema Almacen para corroborar que se haya guardado correctamente.

Cancelar.- Este botón cancela el proceso y de igual forma que el botón agregar nos enviará al Sistema Almacen.

2.3.1.2 Sistema Edición I

Este sistema muestra la información de la misma forma que el Sistema Compra con la diferencia que ya trae la información de forma predeterminada del producto elegido (al igual que un código distinto ya que este actualiza y el otro agrega información a nuestra base de datos).

2.3.2 Sistema Agregar Producto

En este sistema se agrega inventario a los productos que ya tenemos en stock, teniendo como parámetros el precio compra, precio venta, fecha compra, proveedores, nombre. Esto se hace con la finalidad de mantener un mayor control sobre el producto que entra.

2.3.2.1 Sistema Proveedores

En este sistema se agrega la información de los proveedores (correo de contacto y razón social) siendo agregados con el botón de “Agregar” ,estos son usados en el detalle compra para mantener un registro de a quien se le fue comprado el producto nuevo.

2.3.3 Sistema Usuarios

Este sistema solo es mostrado al supervisor, con el objetivo de que él sea el único que puede dar de alta trabajadores nuevos para mantener un control más exhaustivo y por ende una mejor seguridad.

Este sistema tiene los mismos parámetros que el login con la diferencia de que aquí se puede asignar un rol específico para la persona deseada, al igual que una lista de usuarios donde se muestra su información y al final de cada usuario opciones para editar, eliminar y detalle que tiene las mismas funciones previamente explicadas en el Sistema Almacén pero modificado a este sistema.

3 Computación en la nube

3.1 Introducción

Cuando se hace referencia a la nube, se está aludiendo a un término con algunos años de historia y que es una forma metafórica de nombrar a Internet. Básicamente la computación en la nube consiste en los servicios ofrecidos a través de la red tales como correo electrónico, almacenamiento, uso de aplicaciones, etc., los cuales son normalmente accesibles mediante un navegador web. Al utilizar estos servicios, la información utilizada y almacenada, así como la mayor´ıa de las aplicaciones requeridas, son procesadas y ejecutadas por un servidor en Internet. Dicho en otras palabras, se trata de una implementación que pretende transformar el arquetipo habitual de la computación y la informática y trasladarla a Internet.

3.2 Características

No es necesario disponer de un equipo potente, tan solo de un aparato con conexión a internet; esto debido a que el dispositivo del usuario no realiza ningún proceso complejo y los ficheros pueden guardarse en la nube. Los servidores en donde se hallan los programas que se utilicen son los encargados de las tareas complicadas que antes se realizaban localmente. Con el uso del Cloud Computing no hay necesidad por parte del usuario de conocer la infraestructura detrás de esta, ya que pasa a ser una abstracción, “una nube” donde las aplicaciones y servicios pueden fácilmente crecer, funcionar rápido y con pocas fallas. Este tipo de servicio se puede pagar según alguna métrica de consumo, no por el equipo usado en sí, sino por uso de CPU/hora como en el caso de Amazon EC2. Entre otras características podemos mencionar: Es auto reparable: En caso de surgir un

fallo, el último respaldo (backup) de la aplicación se convierte automáticamente en la copia primaria y a partir de esta se genera uno nuevo.

3.3 Ventajas

El primero de ellos es el ahorro, tanto en licencias como en la administración del servicio y en los equipos necesarios. Si se cuenta con una infraestructura 100 % basada en “nube computacional” no se requiere instalar ningún tipo de hardware, solo los terminales. En esa simplicidad para el usuario y el hecho de que requiera mucha menor inversión para empezar a trabajar radica la belleza de la tecnología de Cloud Computing.

Por ejemplo, el cambio del software de oficina de una empresa de unos 40,000 usuarios por un servicio de computación en la nube. El cambio toma unos cuantos meses y produce ahorros multimillonarios.

En cuanto al hardware del cliente también hay ahorro, no es necesario escoger entre una computadora portátil o una de escritorio, más barata y a menudo más rápida. En el mundo de la computación en nube, el usuario puede comprar un económico thin client 6 portátil que puede conectar a una pantalla y a un teclado. Entonces, todo lo que necesita es conectarse a su proveedor en la nube y disponer de todo el rendimiento y memoria que desee. Luego, cuando normalmente el consumidor debería reemplazar su obsoleto ordenador portátil, aún podría usar su thin client, porque es el proveedor el que ofrece el rendimiento y no el equipo en sí. Implementación rápida y baja en riesgos. Gracias a una infraestructura de Cloud Computing, es posible comenzar a trabajar muy rápidamente. No es necesario esperar mucho tiempo e invertir grandes cantidades de dinero antes de que un usuario inicie sesión en su nueva solución. Las aplicaciones basadas en tecnología de la nube estarán disponibles en cuestión de pocas semanas, incluso con un alto nivel de personalización.

Actualizaciones automáticas: No afectan negativamente a los recursos de TI. Si se actualiza a la última versión de la aplicación, la nueva tecnoloǵia no obliga al consumidor a decidir entre actualizar o conservar su trabajo, porque las personalizaciones e integraciones se conservan automáticamente durante la actualizaci´on.

Portabilidad de información: Aunque en un principio la mayor´ıa de los proveedores en la nube dirigıan sus servicios a los usuarios corporativos, con el paso del tiempo los usuarios

particulares han comenzado a usar este concepto manera masiva y casi sin darse cuenta con el uso de servicios para teléfonos móviles (smartphones particularmente), tablets, etc.

3.4 Desventajas

No todo son maravillas en la gran nube, pues existen factores que harán tropezar la confiabilidad de los servicios ofrecidos por esta. Por un lado el crecimiento de esta noción ha fortalecido las ventas de los hoy llamados netbooks, los cuales han sido sacrificados en sus prestaciones f´ısicas como la ausencia de unidad óptica e incluso, en la mayor´ıa de las ocasiones, escaso disco duro; pues la idea central es que el usuario no se llene de perif´ericos y s´olo acceda a su información a través de la red. El concepto es bueno, pero los fallos de los servidores en distintas ocasiones han alertado a muchos usuarios que a´un desconf´ıan de un servicio como este. Es el caso de Google, por ejemplo, con quien en varias ocasiones tanto Gmail como su buscador insigne han sufrido fallas que han dejado a sus usuarios fuera de servicio por unas cuantas horas; las suficientes como para que millones de clientes reclamen y queden con una sensación de que hay algo que no est´a funcionando del todo bien. Otro riesgo importante es la fuga de información, un problema común dada la variedad de los datos que los proveedores en la nube almacenan, lo que implica que en cualquier fuga de información puede ocurrir un significativo impacto. Usar los servicios en la nube implica tener una confianza casi absoluta en el proveedor, dejando en sus manos información importante, resultando atractivo para que los piratas cibernéticos y autores de programas maliciosos apunten a los servicios de computación en la nube con el propósito de buscar datos que puedan robar, vender, manipular o simplemente “mirar”. Dado que la información del cliente debe recorrer diferentes nodos para llegar a su destino, cada uno de ellos resulta un foco de inseguridad. Si se utilizan protocolos seguros como HTTPS por ejemplo, la velocidad total disminuye debido a la sobrecarga que requieren estos protocolos. Por ello, ya que la computación en nube tiene sus miras puestas a convertirse en una herramienta empresarial esencial, tal como se ha consolidado hoy en día el correo electrónico, se necesita introducir patrones de cifrado de datos para mitigar cualquier tipo de riesgo. La disponibilidad de las aplicaciones está atada a la existencia de acceso a Internet. Si un consumidor decide tener todos sus servicios informáticos en la nube, queda sujeto a la cobertura de red; incluso si solo son algunas aplicaciones las que usa en la nube o únicamente información, al carecer de conexión, su productividad se ve anulada o en el mejor caso limitada.

3.5 Tipos de nubes

Nubes públicas: Estas se administran externamente por terceros, los contenidos de distintos clientes pueden encontrarse ubicados en los mismos servidores, sistemas de almacenamiento. Los usuarios finales usan la infraestructura de la nube en todas sus capas y no conocen qué trabajos de otros clientes pueden estar corriendo en el mismo servidor o red.

Nubes privadas: En este caso el proveedor es propietario del servidor, red, y disco y pueden decidir qué usuarios están autorizados a utilizar la infraestructura. Las nubes privadas están en una infraestructura manejada por un solo administrador que controla qué aplicaciones debe correr y donde. Son una buena opción para las compañías que necesitan alta protección de datos y manipulaciones a nivel de servicio.

Nubes híbridas: Aquí se trata de una combinación de nubes públicas y privadas. El cliente está en posesión de unas partes y comparte otras, esto además puede ser de manera controlada. Las nubes híbridas ofrecen la ventaja del escalado proporcionado externamente, bajo demanda, se añade la posibilidad de determinar cómo distribuir las aplicaciones a través de los ambientes diferentes.

3.6 Plataformas de servicios en la nube

3.6.1 Google Apps.

Es uno de los servicios que Google ofrece. Como la mayor´ıa de sus aplicaciones es completamente gratis; aunque también existe una versión de pago especialmente diseñada para clientes empresariales. Google Apps proporciona herramientas eficaces para la gestión y personalización de utilidades para dominios o nombres de Internet. Es decir, Google Apps permite gestionar el correo electr´onico de un dominio (a través de Gmail), mensajer´ıa instant´anea entre miembros de una organización o red (Google Talk), calendario en l´ınea (Google Calendar), edición de Documentos igualmente en lınea (Google Docs) y creación de sitios web profesionales (Google Sites).

3.6.2 Amazon EC2.

Amazon Elastic Compute Cloud es un servicio Web que proporciona capacidad informática con tamaño modificable en la nube. Según la propia Amazon, se ha diseñado con el fin de que la computación web resulte más sencilla a los desarrolladores. Lo interesante de

Amazon es su facilidad para poder escalar de forma horizontal. Esto es agregando mas procesador, mas memoria, mas almacenamiento, o mas instancias, que vendrıan a ser como mas servidores en paralelo. Provee herramientas de recuperación de datos y aislamiento frente a otros procesos realizados en sus máquinas. En este tipo de servicio solo se paga por la capacidad utilizada. Se apoya en las tecnologías de virtualización, lo cual permite utilizar diversos sistemas operativos a través de sus interfaces de servicios Web.

3.6.3 Windows Azure.

Es una plataforma que se ofrece como servicio y alojada en los centros de procesamiento de datos de Microsoft. Ofrece distintos servicios para aplicaciones, desde los que permiten guardar aplicaciones en alguno de los centros de procesamiento de datos de la compañía para que se ejecute sobre su infraestructura en la nube hasta otros de comunicación segura y asociación entre aplicaciones.

4 Programación del Almacén de Zapatos “zapa +”

4.1 Vistas

4.1.1 Acceso

4.1.1.1 Index

```
@model Usuario
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
```

```
    <title>Index</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
</head>
<body>

    @*------ GOOGLEAR ====> form bootstrap 5 ----------------*@

<div class="row mt-5">
    <div class="col-sm-4 offset-sm-4" >
        <div class="col-md-9 col-lg-6 col-xl-5" >
            <img src="https://cdn-icons-png.flaticon.com/512/3596/3596081.png"
                class="img-fluid" alt="Sample image">
        </div>
        <hr />

        <form asp-controller="Acceso" asp-action="Index" method="post">
            <div class="mb-3">
              <label for="exampleInputEmail1" class="form-label">Email address</label>
              <input type="email" class="form-control" asp-for="Correo">
                    @*<div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>*@
            </div>
            <div class="mb-3">
              <label for="exampleInputPassword1" class="form-label">Password</label>
              <input type="password" class="form-control" asp-for="Clave">
            </div>
```

```
            @* <div class="mb-3 form-check">
              <input type="checkbox" class="form-check-input" id="exampleCheck1">
              <label class="form-check-label" for="exampleCheck1">Check me out</label>
            </div>*@
            <div class="form-group">
              <button type="submit" class="btn btn-primary">Iniciar Sesión</button>
              <a asp-action="Registrar" class="btn btn-link">Registrarme</a>
            </div>
        </form>


    </div>
</div>




</body>
   <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
</html>
```

4.1.1.2 Registrar

```
@model Usuario
@{
    Layout = null;
}

<!DOCTYPE html>
```

```
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
</head>
<body>
    <div class="row mt-5">
        <div class="col-sm-4 offset-sm-4" >
            <div class="col-md-9 col-lg-6 col-xl-5" >
                                                                                <img
src="https://cdn-icons.flaticon.com/png/512/3631/premium/3631696.png?token=exp=16525
37782~hmac=3c9a848e95de15e419c420f8a043068d"
                    class="img-fluid" alt="Sample image">
            </div>
            <hr />
            <form asp-action="Registrar" id="frmRegistro">
                <h4>Registro</h4>
                <hr />
                <div asp-validation-summary="ModelOnly" class="text-danger"></div>

                <div class="form-group">
                    <label asp-for="Nombre" class="control-label"></label>
                      <input asp-for="Nombre" class="form-control" required placeholder="Nombre"
/>
                    <span asp-validation-for="Nombre" class="text-danger"></span>
```

```
            </div>
            <div class="form-group">
                <label asp-for="Correo" class="control-label"></label>
                                <input type="email" asp-for="Correo" class="form-control" required
placeholder="Correo electrónico" />
                <span asp-validation-for="Correo" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Clave" class="control-label"></label>
                                <input type="password" asp-for="Clave" class="form-control" required
placeholder="Contraseña" />
                <span asp-validation-for="Clave" class="text-danger"></span>
            </div>


            <div class="form-group">
                <label asp-for="Roles" class="control-label" hidden></label>
                  <select asp-for="Roles" class="form-control" id="exampleFormControlSelect1"
hidden>
                  <option>Cliente</option>
                </select>
                <span asp-validation-for="Roles" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Registrarme" class="btn btn-primary" />
                 <a asp-action="Index" class="btn btn-link">Regresar</a>
            </div>
        </form>
```

```
        </div>
    </div>

</body>
   <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
</html>
```

4.1.2 Detallecompra

4.1.2.1 Index

```
@model IEnumerable<ProyectoAula.Models.DetalleCompra>

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.FechaCompra)
            </th>
            <th>
```

```
            @Html.DisplayNameFor(model => model.Precio)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Cantidad)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Total)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Proveedores)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Productos)
        </th>
        <th></th>
    </tr>
</thead>
<tbody>
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.FechaCompra)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Precio)
        </td>
```

```
            <td>
                @Html.DisplayFor(modelItem => item.Cantidad)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Total)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Proveedores.RazonSocial)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Productos.Producto_Nombre)
            </td>
            <td>
                <a asp-action="Edit" asp-route-id="@item.DetalleCompra_ID">Edit</a> |
                <a asp-action="Details" asp-route-id="@item.DetalleCompra_ID">Details</a> |
                <a asp-action="Delete" asp-route-id="@item.DetalleCompra_ID">Delete</a>
            </td>
        </tr>
}
    </tbody>
</table>
```

4.1.2.2 Create

```
@model ProyectoAula.Models.DetalleCompra


@{
    ViewData["Title"] = "Create";
```

```
}

<h1>Create</h1>

<h4>DetalleCompra</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Proveedores" class="control-label"></label>
                                        <select asp-for="Proveedores_ID" class ="form-control" asp-items="ViewBag.Proveedores_ID"></select>
            </div>
            <div class="form-group">
                <label asp-for="Producto_ID" class="control-label"></label>
                                            <select asp-for="Producto_ID" class ="form-control" asp-items="ViewBag.Producto_ID"></select>
            </div>
            <div class="form-group">
                <label asp-for="Precio" class="control-label"></label>
                <input asp-for="Precio" class="form-control" />
                <span asp-validation-for="Precio" class="text-danger"></span>
            </div>
            <div class="form-group">
```

```
            <label asp-for="Cantidad" class="control-label"></label>
            <input asp-for="Cantidad" class="form-control" />
            <span asp-validation-for="Cantidad" class="text-danger"></span>
        </div>
        <div class="form-group">
            <label asp-for="Total" class="control-label"></label>
            <input asp-for="Total" class="form-control" />
            <span asp-validation-for="Total" class="text-danger"></span>
        </div>
        <div class="form-group">
            <input type="submit" value="Create" class="btn btn-primary" />
        </div>
    </form>
  </div>
</div>

<div>
  <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
  @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

4.1.2.3 Edit

```
@model ProyectoAula.Models.DetalleCompra
```

```
@{
    ViewData["Title"] = "Edit";
}

<h1>Edit</h1>

<h4>DetalleCompra</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Edit">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <input type="hidden" asp-for="DetalleCompra_ID" />
            <div class="form-group">
                <label asp-for="FechaCompra" class="control-label"></label>
                <input asp-for="FechaCompra" class="form-control" />
                <span asp-validation-for="FechaCompra" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Proveedores_ID" class="control-label"></label>
                                        <select  asp-for="Proveedores_ID"  class="form-control"
asp-items="ViewBag.Proveedores_ID"></select>
                <span asp-validation-for="Proveedores_ID" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Producto_ID" class="control-label"></label>
```

```
                                        <select asp-for="Producto_ID" class="form-control"
asp-items="ViewBag.Producto_ID"></select>
            <span asp-validation-for="Producto_ID" class="text-danger"></span>
        </div>
        <div class="form-group">
            <label asp-for="Precio" class="control-label"></label>
            <input asp-for="Precio" class="form-control" />
            <span asp-validation-for="Precio" class="text-danger"></span>
        </div>
        <div class="form-group">
            <label asp-for="Cantidad" class="control-label"></label>
            <input asp-for="Cantidad" class="form-control" />
            <span asp-validation-for="Cantidad" class="text-danger"></span>
        </div>
        <div class="form-group">
            <label asp-for="Total" class="control-label"></label>
            <input asp-for="Total" class="form-control" />
            <span asp-validation-for="Total" class="text-danger"></span>
        </div>
        <div class="form-group">
            <input type="submit" value="Save" class="btn btn-primary" />
        </div>
    </form>
  </div>
</div>
```

```
<div>
    <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

4.1.2.4 Delete

```
@model ProyectoAula.Models.DetalleCompra

@{
    ViewData["Title"] = "Delete";
}

<h1>Delete</h1>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>DetalleCompra</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.FechaCompra)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.FechaCompra)
```

```
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Precio)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Precio)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Cantidad)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Cantidad)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Total)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Total)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Proveedores)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Proveedores.Correo)
</dd class>
<dt class = "col-sm-2">
```

```
            @Html.DisplayNameFor(model => model.Productos)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Productos.Categoria)
        </dd class>
    </dl>

    <form asp-action="Delete">
        <input type="hidden" asp-for="DetalleCompra_ID" />
        <input type="submit" value="Delete" class="btn btn-danger" /> |
        <a asp-action="Index">Back to List</a>
    </form>
</div>
```

4.1.2.5 Details

```
@model ProyectoAula.Models.DetalleCompra

@{
    ViewData["Title"] = "Details";
}
<h1>Details</h1>
<div>
    <h4>DetalleCompra</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.FechaCompra)
```

```
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.FechaCompra)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Precio)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Precio)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Cantidad)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Cantidad)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Total)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Total)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Proveedores)
</dt>
<dd class = "col-sm-10">
```

```
            @Html.DisplayFor(model => model.Proveedores.Correo)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Productos)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Productos.Categoria)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="Edit" asp-route-id="@Model.DetalleCompra_ID">Edit</a> |
    <a asp-action="Index">Back to List</a>
</div>
```

4.1.3 Usuarios

4.1.3.1 Index

```
@model IEnumerable<ProyectoAula.Models.Usuario>
@{
    ViewData["Title"] = "Index";
}
<div class="container-fluid">
    <div class="d-sm-flex align-items-center justify-content-between mb-4">
        <h1 class="h3 mb-0 text-gray-800">Usuarios</h1>
                <a asp-action="Create" class="d-none d-sm-inline-block btn btn-sm btn-grad shadow-sm">Nuevo Usuario</a>
    </div>
```

```
<div>
    <table class="table">
        <thead>
            <tr>
                <th>
                    @Html.DisplayNameFor(model => model.Nombre)
                </th>
                <th>
                    @Html.DisplayNameFor(model => model.Correo)
                </th>
                <th>
                    @Html.DisplayNameFor(model => model.Roles)
                </th>
                <th></th>
            </tr>
        </thead>
        <tbody>
    @foreach (var item in Model) {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.Nombre)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Correo)
                </td>
```

```
                <td>
                    @Html.DisplayFor(modelItem => item.Roles)
                </td>
                <td>
                    <a asp-action="Edit" asp-route-id="@item.Usuario_ID">Edit</a> |
                    <a asp-action="Details" asp-route-id="@item.Usuario_ID">Details</a> |
                    <a asp-action="Delete" asp-route-id="@item.Usuario_ID">Delete</a>
                </td>
            </tr>
        }
        </tbody>
    </table>
</div>
</div>
```

4.1.3.2 Create

```
@model ProyectoAula.Models.Usuario
@{
    ViewData["Title"] = "Create";
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
</head>
```

```
<body>


    @*------ GOOGLEAR ====> form bootstrap 5 ----------------*@
    <div class="row mt-5">
        <div class="col-sm-4 offset-sm-4" >
                        <div  class="col-md-9  col-lg-6  col-xl-5"  >                                <img
src="https://cdn-icons.flaticon.com/png/512/3631/premium/3631696.png?token=exp=16525
37782~hmac=3c9a848e95de15e419c420f8a043068d"
                class="img-fluid" alt="Sample image">
            </div>
            <hr />
            <form asp-action="Registrar" id="frmRegistro">
                <h4>Registro</h4>
                <hr />
                <div asp-validation-summary="ModelOnly" class="text-danger"></div>
                <div class="form-group">
                    <label asp-for="Nombre" class="control-label"></label>
                      <input asp-for="Nombre" class="form-control" required placeholder="Nombre"
/>
                    <span asp-validation-for="Nombre" class="text-danger"></span>
                </div>
                <div class="form-group">
                    <label asp-for="Correo" class="control-label"></label>
                                <input  type="email"  asp-for="Correo"  class="form-control"  required
placeholder="Correo electrónico" />
                    <span asp-validation-for="Correo" class="text-danger"></span>
                </div>
```

```
            <div class="form-group">
                <label asp-for="Clave" class="control-label"></label>
                        <input type="password" asp-for="Clave" class="form-control" required placeholder="Contraseña" />
                <span asp-validation-for="Clave" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Roles" class="control-label"></label>
                                                <select asp-for="Roles" class="form-control" id="exampleFormControlSelect1">
                  <option>Administrador</option>
                  <option>Empleado</option>
                </select>
                <span asp-validation-for="Roles" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Registrarme" class="btn btn-primary" />
            </div>
        </form>
    </div>
  </div>
</body>
  <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
</html>
```

4.1.3.3 Delete

```
@model ProyectoAula.Models.Usuario
```

```
@{
    ViewData["Title"] = "Delete";
}
<h1>Delete</h1>
<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Usuario</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Nombre)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Nombre)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Correo)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Correo)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Roles)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Roles)
```

```
        </dd>
    </dl>

    <form asp-action="Delete">
        <input type="hidden" asp-for="Usuario_ID" />
        <input type="submit" value="Delete" class="btn btn-danger" /> |
        <a asp-action="Index">Back to List</a>
    </form>
</div>
```

4.1.3.4 Details

```
@model ProyectoAula.Models.Usuario
@{
    ViewData["Title"] = "Details";
}
<h1>Details</h1>
<div>
    <h4>Usuario</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Nombre)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Nombre)
        </dd>
        <dt class = "col-sm-2">
```

```
            @Html.DisplayNameFor(model => model.Correo)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Correo)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Clave)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Clave)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Roles)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Roles)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="Edit" asp-route-id="@Model.Usuario_ID">Edit</a> |
    <a asp-action="Index">Back to List</a>
</div>
```

4.1.3.5 Edit

```
@model ProyectoAula.Models.Usuario
@{
```

```
    ViewData["Title"] = "Edit";
}
<h1>Edit</h1>
<h4>Usuario</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Edit">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <input type="hidden" asp-for="Usuario_ID" />
            <div class="form-group">
                <label asp-for="Nombre" class="control-label"></label>
                <input asp-for="Nombre" class="form-control" />
                <span asp-validation-for="Nombre" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Correo" class="control-label"></label>
                <input asp-for="Correo" class="form-control" />
                <span asp-validation-for="Correo" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Clave" class="control-label"></label>
                <input asp-for="Clave" class="form-control" />
                <span asp-validation-for="Clave" class="text-danger"></span>
            </div>
            <div class="form-group">
```

```
            <label asp-for="Roles" class="control-label"></label>
            <input asp-for="Roles" class="form-control" />
            <span asp-validation-for="Roles" class="text-danger"></span>
        </div>
        <div class="form-group">
            <input type="submit" value="Save" class="btn btn-primary" />
        </div>
    </form>
  </div>
</div>
<div>
  <a asp-action="Index">Back to List</a>
</div>
@section Scripts {
  @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

4.1.4 Productos

Index

```
@model IEnumerable<ProyectoAula.Models.Productos>
@{
  ViewData["Title"] = "Index";
}
<h1>Index</h1>
<p>
  <a asp-action="Create">Create New</a>
</p>
```

```
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Producto_Nombre)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Marca)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Division)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Categoria)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Color)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Talla)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Stock)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Precio_Compra)
```

```
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Precio_Venta)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
@foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Producto_Nombre)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Marca)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Division)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Categoria)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Color)
            </td>
            <td>
```

```
            @Html.DisplayFor(modelItem => item.Talla)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Stock)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Precio_Compra)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Precio_Venta)
        </td>
        <td>
            <a asp-action="Edit" asp-route-id="@item.Producto_ID">Edit</a> |
            <a asp-action="Details" asp-route-id="@item.Producto_ID">Details</a> |
            <a asp-action="Delete" asp-route-id="@item.Producto_ID">Delete</a>
        </td>
    </tr>
}
    </tbody>
</table>
Create
@model ProyectoAula.Models.Productos
@{
    ViewData["Title"] = "Create";
}
<h1>Create</h1>
```

```
<h4>Productos</h4>

<hr />

<div class="row">

    <div class="col-md-4">

        <form asp-action="Create">

            <div asp-validation-summary="ModelOnly" class="text-danger"></div>

            <div class="form-group">

                <label asp-for="Producto_Nombre" class="control-label"></label>

                <input asp-for="Producto_Nombre" class="form-control" />

                <span asp-validation-for="Producto_Nombre" class="text-danger"></span>

            </div>

            <div class="form-group">

                <label asp-for="Marca" class="control-label"></label>

                <input asp-for="Marca" class="form-control" />

                <span asp-validation-for="Marca" class="text-danger"></span>

            </div>

            <div class="form-group">

                <label asp-for="Division" class="control-label"></label>

                <input asp-for="Division" class="form-control" />

                <span asp-validation-for="Division" class="text-danger"></span>

            </div>

            <div class="form-group">

                <label asp-for="Categoria" class="control-label"></label>

                <input asp-for="Categoria" class="form-control" />

                <span asp-validation-for="Categoria" class="text-danger"></span>

            </div>
```

```
<div class="form-group">
    <label asp-for="Color" class="control-label"></label>
    <input asp-for="Color" class="form-control" />
    <span asp-validation-for="Color" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="Talla" class="control-label"></label>
    <input asp-for="Talla" class="form-control" />
    <span asp-validation-for="Talla" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="Stock" class="control-label"></label>
    <input asp-for="Stock" class="form-control" />
    <span asp-validation-for="Stock" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="Precio_Compra" class="control-label"></label>
    <input asp-for="Precio_Compra" class="form-control" />
    <span asp-validation-for="Precio_Compra" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="Precio_Venta" class="control-label"></label>
    <input asp-for="Precio_Venta" class="form-control" />
    <span asp-validation-for="Precio_Venta" class="text-danger"></span>
</div>
<div class="form-group">
```

```
            <input type="submit" value="Create" class="btn btn-primary" />
        </div>
    </form>
</div>
</div>
<div>
    <a asp-action="Index">Back to List</a>
</div>
@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
Delete
@model ProyectoAula.Models.Productos
@{
    ViewData["Title"] = "Delete";
}
<h1>Delete</h1>
<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Productos</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Producto_Nombre)
        </dt>
        <dd class = "col-sm-10">
```

```
    @Html.DisplayFor(model => model.Producto_Nombre)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Marca)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Marca)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Division)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Division)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Categoria)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Categoria)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Color)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Color)
</dd>
```

```
    <dt class = "col-sm-2">
        @Html.DisplayNameFor(model => model.Talla)
    </dt>
    <dd class = "col-sm-10">
        @Html.DisplayFor(model => model.Talla)
    </dd>
    <dt class = "col-sm-2">
        @Html.DisplayNameFor(model => model.Stock)
    </dt>
    <dd class = "col-sm-10">
        @Html.DisplayFor(model => model.Stock)
    </dd>
    <dt class = "col-sm-2">
        @Html.DisplayNameFor(model => model.Precio_Compra)
    </dt>
    <dd class = "col-sm-10">
        @Html.DisplayFor(model => model.Precio_Compra)
    </dd>
    <dt class = "col-sm-2">
        @Html.DisplayNameFor(model => model.Precio_Venta)
    </dt>
    <dd class = "col-sm-10">
        @Html.DisplayFor(model => model.Precio_Venta)
    </dd>
</dl>
<form asp-action="Delete">
```

```
        <input type="hidden" asp-for="Producto_ID" />
        <input type="submit" value="Delete" class="btn btn-danger" /> |
        <a asp-action="Index">Back to List</a>
    </form>
</div>
Edit
@model ProyectoAula.Models.Productos
@{
    ViewData["Title"] = "Edit";
}
<h1>Edit</h1>
<h4>Productos</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Edit">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <input type="hidden" asp-for="Producto_ID" />
            <div class="form-group">
                <label asp-for="Producto_Nombre" class="control-label"></label>
                <input asp-for="Producto_Nombre" class="form-control" />
                <span asp-validation-for="Producto_Nombre" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Marca" class="control-label"></label>
                <input asp-for="Marca" class="form-control" />
```

```
    <span asp-validation-for="Marca" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="Division" class="control-label"></label>
    <input asp-for="Division" class="form-control" />
    <span asp-validation-for="Division" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="Categoria" class="control-label"></label>
    <input asp-for="Categoria" class="form-control" />
    <span asp-validation-for="Categoria" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="Color" class="control-label"></label>
    <input asp-for="Color" class="form-control" />
    <span asp-validation-for="Color" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="Talla" class="control-label"></label>
    <input asp-for="Talla" class="form-control" />
    <span asp-validation-for="Talla" class="text-danger"></span>
</div>
<div class="form-group">
    <label asp-for="Stock" class="control-label"></label>
    <input asp-for="Stock" class="form-control" />
    <span asp-validation-for="Stock" class="text-danger"></span>
```

```
            </div>
            <div class="form-group">
                <label asp-for="Precio_Compra" class="control-label"></label>
                <input asp-for="Precio_Compra" class="form-control" />
                <span asp-validation-for="Precio_Compra" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Precio_Venta" class="control-label"></label>
                <input asp-for="Precio_Venta" class="form-control" />
                <span asp-validation-for="Precio_Venta" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Save" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>
<div>
    <a asp-action="Index">Back to List</a>
</div>
@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
Details
@model ProyectoAula.Models.Productos
@{
```

```
    ViewData["Title"] = "Details";
}
<h1>Details</h1>
<div>
    <h4>Productos</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Producto_Nombre)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Producto_Nombre)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Marca)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Marca)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Division)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Division)
        </dd>
        <dt class = "col-sm-2">
```

```
    @Html.DisplayNameFor(model => model.Categoria)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Categoria)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Color)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Color)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Talla)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Talla)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Stock)
</dt>
<dd class = "col-sm-10">
    @Html.DisplayFor(model => model.Stock)
</dd>
<dt class = "col-sm-2">
    @Html.DisplayNameFor(model => model.Precio_Compra)
</dt>
```

```
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Precio_Compra)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Precio_Venta)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Precio_Venta)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="Edit" asp-route-id="@Model.Producto_ID">Edit</a> |
    <a asp-action="Index">Back to List</a>
</div>
```

4.1.5 Proveedores

Index

```
@model IEnumerable<ProyectoAula.Models.Proveedores>
@{
    ViewData["Title"] = "Index";
}
<div class="container-fluid">
    <div class="d-sm-flex align-items-center justify-content-between mb-4">
        <h1 class="h3 mb-0 text-gray-800">Proveedores</h1>
                <a asp-action="Create" class="d-none d-sm-inline-block btn btn-sm btn-grad shadow-sm">Nuevo Proveedor</a>
```

```
</div>
<div>
    <table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.RazonSocial)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Telefono)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Correo)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
@foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.RazonSocial)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Telefono)
            </td>
```

```
            <td>
                @Html.DisplayFor(modelItem => item.Correo)
            </td>
            <td>
                <a asp-action="Edit" asp-route-id="@item.Proveedores_ID">Editar</a> |
                <a asp-action="Details" asp-route-id="@item.Proveedores_ID">Detalles</a> |
                <a asp-action="Delete" asp-route-id="@item.Proveedores_ID">Eliminar</a>
            </td>
        </tr>
    }
    </tbody>
    </table>
    </div>
</div>
Create
@model ProyectoAula.Models.Proveedores
@{
    ViewData["Title"] = "Create";
}
<h1>Create</h1>
<h4>Proveedores</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

```
            <div class="form-group">
                <label asp-for="RazonSocial" class="control-label"></label>
                <input asp-for="RazonSocial" class="form-control" />
                <span asp-validation-for="RazonSocial" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Telefono" class="control-label"></label>
                <input asp-for="Telefono" class="form-control" />
                <span asp-validation-for="Telefono" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Correo" class="control-label"></label>
                <input asp-for="Correo" class="form-control" />
                <span asp-validation-for="Correo" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Create" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>
<div>
    <a asp-action="Index">Back to List</a>
</div>
@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
```

```
}
```

Delete

```
@model ProyectoAula.Models.Proveedores
@{
    ViewData["Title"] = "Delete";
}
<h1>Delete</h1>
<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Proveedores</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.RazonSocial)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.RazonSocial)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Telefono)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Telefono)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Correo)
```

```
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Correo)
        </dd>
    </dl>
    <form asp-action="Delete">
        <input type="hidden" asp-for="Proveedores_ID" />
        <input type="submit" value="Delete" class="btn btn-danger" /> |
        <a asp-action="Index">Back to List</a>
    </form>
</div>
Edit
@model ProyectoAula.Models.Proveedores
@{
    ViewData["Title"] = "Edit";
}
<h1>Edit</h1>
<h4>Proveedores</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Edit">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <input type="hidden" asp-for="Proveedores_ID" />
            <div class="form-group">
                <label asp-for="RazonSocial" class="control-label"></label>
```

```
            <input asp-for="RazonSocial" class="form-control" />
            <span asp-validation-for="RazonSocial" class="text-danger"></span>
        </div>
        <div class="form-group">
            <label asp-for="Telefono" class="control-label"></label>
            <input asp-for="Telefono" class="form-control" />
            <span asp-validation-for="Telefono" class="text-danger"></span>
        </div>
        <div class="form-group">
            <label asp-for="Correo" class="control-label"></label>
            <input asp-for="Correo" class="form-control" />
            <span asp-validation-for="Correo" class="text-danger"></span>
        </div>
        <div class="form-group">
            <input type="submit" value="Save" class="btn btn-primary" />
        </div>
    </form>
  </div>
</div>
<div>
  <a asp-action="Index">Back to List</a>
</div>
@section Scripts {
  @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
Details
```

```
@model ProyectoAula.Models.Proveedores

@{
    ViewData["Title"] = "Details";
}

<h1>Details</h1>

<div>
    <h4>Proveedores</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.RazonSocial)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.RazonSocial)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Telefono)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Telefono)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Correo)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Correo)
```

```
        </dd>
    </dl>
</div>
<div>
    <a asp-action="Edit" asp-route-id="@Model.Proveedores_ID">Edit</a> |
    <a asp-action="Index">Back to List</a>
</div>
```

4.1.6 Ventas

4.1.6.1 Index

```
@model IEnumerable<ProyectoAula.Models.DetalleVenta>
@{
    ViewData["Title"] = "Index";
}
<div class="container-fluid">
<div class="d-sm-flex align-items-center justify-content-between mb-4">
        <h1 class="h3 mb-0 text-gray-800">Venta</h1>
        <a asp-action="Create" class="d-none d-sm-inline-block btn btn-sm btn-grad
shadow-sm">Crear Venta</a>
    </div>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.FechaVenta)
            </th>
            <th>
```

```
            @Html.DisplayNameFor(model => model.PrecioVenta)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Cantidad)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Total)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Productos)
        </th>
        <th></th>
    </tr>
</thead>
<tbody>
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.FechaVenta)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.PrecioVenta)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Cantidad)
        </td>
```

```
            <td>
                @Html.DisplayFor(modelItem => item.Total)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Productos.Categoria)
            </td>
            <td>
                <a asp-action="Edit" asp-route-id="@item.DetalleVenta_ID">Edit</a> |
                <a asp-action="Details" asp-route-id="@item.DetalleVenta_ID">Details</a> |
                <a asp-action="Delete" asp-route-id="@item.DetalleVenta_ID">Delete</a>
            </td>
        </tr>
}
    </tbody>
</table>
</div>
```

4.1.6.2 Create

```
@model ProyectoAula.Models.DetalleVenta
@{
    ViewData["Title"] = "Create";
}
<h1>Crear</h1>
<h4>Venta</h4>
<hr />
<div class="row">
    <div class="col-md-4">
```

```
        <form asp-action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Producto_ID" class="control-label">Nombre de Producto</label>
                <select asp-for="Producto_ID" class ="form-control"
asp-items="ViewBag.Producto_ID"></select>
            </div>
            <div class="form-group">
                <label asp-for="PrecioVenta" class="control-label"></label>
                <input asp-for="PrecioVenta" class="form-control" />
                <span asp-validation-for="PrecioVenta" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Cantidad" class="control-label"></label>
                <input asp-for="Cantidad" class="form-control" />
                <span asp-validation-for="Cantidad" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Create" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>
<div>
    <a asp-action="Index">Back to List</a>
</div>
```

```
@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

4.1.6.3 Delete

```
@model ProyectoAula.Models.DetalleVenta
@{
    ViewData["Title"] = "Details";
}
<h1>Details</h1>
<div>
    <h4>DetalleVenta</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.FechaVenta)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.FechaVenta)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.PrecioVenta)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.PrecioVenta)
        </dd>
        <dt class = "col-sm-2">
```

```
            @Html.DisplayNameFor(model => model.Cantidad)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Cantidad)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Total)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Total)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Productos)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Productos.Categoria)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="Edit" asp-route-id="@Model.DetalleVenta_ID">Edit</a> |
    <a asp-action="Index">Back to List</a>
</div>
```

4.1.6.4 Details

```
@model ProyectoAula.Models.DetalleVenta
@{
```

```
    ViewData["Title"] = "Details";
}

<h1>Details</h1>
<div>
    <h4>DetalleVenta</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.FechaVenta)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.FechaVenta)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.PrecioVenta)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.PrecioVenta)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Cantidad)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Cantidad)
        </dd>
```

```
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Total)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Total)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Productos)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Productos.Categoria)
        </dd>
    </dl>
</div>
<div>
    <a asp-action="Edit" asp-route-id="@Model.DetalleVenta_ID">Edit</a> |
    <a asp-action="Index">Back to List</a>
</div>
```

4.1.6.5 Edit

```
@model ProyectoAula.Models.DetalleVenta
@{
    ViewData["Title"] = "Edit";
}
<h1>Edit</h1>
<h4>DetalleVenta</h4>
<hr />
```

```
<div class="row">
    <div class="col-md-4">
        <form asp-action="Edit">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <input type="hidden" asp-for="DetalleVenta_ID" />
            <div class="form-group">
                <label asp-for="FechaVenta" class="control-label"></label>
                <input asp-for="FechaVenta" class="form-control" />
                <span asp-validation-for="FechaVenta" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Producto_ID" class="control-label"></label>
                <select asp-for="Producto_ID" class="form-control"
asp-items="ViewBag.Producto_ID"></select>
                <span asp-validation-for="Producto_ID" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="PrecioVenta" class="control-label"></label>
                <input asp-for="PrecioVenta" class="form-control" />
                <span asp-validation-for="PrecioVenta" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Cantidad" class="control-label"></label>
                <input asp-for="Cantidad" class="form-control" />
                <span asp-validation-for="Cantidad" class="text-danger"></span>
            </div>
```

```
            <div class="form-group">
                <label asp-for="Total" class="control-label"></label>
                <input asp-for="Total" class="form-control" />
                <span asp-validation-for="Total" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Save" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>
<div>
    <a asp-action="Index">Back to List</a>
</div>
@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

Conclusiones

Este programa está hecho en lenguaje C# para la parte administrativa en un archivo Asp.net core Mvc, usando de igual forma Html para el diseño de esta.

Para la creación de este no es necesaria la posesión de equipo muy avanzado, ya que es un proyecto “básico” por lo que nuestro estudio de factibilidad determina que es un proyecto con un buen costo beneficio, ya que este optimiza las actividades dentro del Consejo de Desarrollo Científico y Humanístico, aumentando la productividad del personal que labora en el mismo, repercutiendo por ende en el funcionamiento.

En la investigación del proyecto se explica el diseño minimalista al que se buscó llegar, teniendo las opciones necesarias en los lugares necesarios, manteniendo la pantalla lo más limpia posible.

y al final la programación en la nube nos muestra nuestras características, ventajas, desventajas, y las plataformas en las que se puede trabajar por lo que se mantiene un buen control sobre todos los aspectos de este programa que tiene como objetivo el facilitar, recrear y reemplazar la parte administrativa de un almacén agregando una página web de pormedio.

Webgrafía

- https://www.mecalux.com.mx/blog/tipos-de-inventario
- https://www.ionos.es/startupguide/gestion/que-es-un-inventario/
- https://www.sana-commerce.com/es/conceptos-de-comercio-electronico/que-es-terminal-punto-de-venta/#:~:text=Un%20POS%20es%20una%20agrupaci%C3%B3n,acuerdo%20con%20sus%20necesidades%20comerciales.
- https://docs.microsoft.com/es-es/visualstudio/releases/2022/system-requirements
- https://aws.amazon.com/es/?nc2=h_lg
- www.microsoft.com/latam/windowsazure/

Bibliografía

- James F. Kurose, Keith W. Ross. Computer networking Pearson/Addison Wesley, 2008