UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR - JOSÉ MIGUEL CARRERA

"Sistema de administración para consultas y ventas nutricionales"

Trabajo de Titulación para optar al Título de:
Técnico Universitario en Informática

Alumnos Integrantes:
Emanuel Elías Tapia Urra
Javier Ignacio Vega Figueroa
Profesor Guía:
Lais San Martín Navarro

2019

RESUMEN

Keywords: Sistema Informático – Consulta nutricional – VB.NET – Compra/Venta de productos

El presente trabajo de título tiene por objetivo presentar la implementación de una aplicación de escritorio para una consulta nutricional privada, la cual no tiene actualmente ningún tipo de soporte informático.

El sistema tiene como objetivo poder apoyar todas las funciones que realiza la empresa tales como, ingresar y almacenar datos sobre los pacientes, sus dietas, ventas de productos, compra, todo esto utilizando bases de datos estructuradas; para que en el sistema se puedan reservar horas, manejar dietas, llevar un conteo de la sesión de cada paciente, asociar dietas a los pacientes, tener un registro de ganancias o pérdidas, entre otras.

A continuación, se describe cada capítulo:

Capítulo 1: Una breve descripción de la organización de la empresa, análisis de la situación actual de la empresa, los problemas detectados y la solución propuesta para solventar los problemas detectados.

Capítulo 2: Descripción del entorno de desarrollo y ejecución, así como también las herramientas utilizadas para desarrollar el sistema. A continuación, se presenta un detalle de todas las tablas que componen la base de datos.

Capítulo 3: Se describe la arquitectura del sistema y se presenta una lista completa de los programas que conforman el sistema y su objetivo. Los programas más relevantes son descritos a continuación en detalle acompañados de diagramas de bloques e imágenes de cada módulo.

El documento finaliza con una conclusión del trabajo realizado, seguido de la bibliografía utilizada y por último el código fuente de programas seleccionados del sistema.

ÍNDICE

INTRODUCCIÓN....1
CAPÍTULO 1: "ASPECTOS RELEVANTES DEL DISEÑO LÓGICO"
1. ASPECTOS RELEVANTES DEL DISEÑO LÓGICO....3
1.1 DESCRIPCIÓN DE LA ORGANIZACIÓN....3
1.2 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL....3
1.2.1 Toma de horas....3
1.2.2 Sesiones....4
1.2.3 Ficha de pacientes....4
1.2.4 Dietas....4
1.2.5 Ventas....5
1.2.6 Productos....5
1.3 PROBLEMAS DETECTADOS....5
1.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO....6
1.3.1 Objetivos....6
1.3.2 Objetivos específicos del sistema propuesto....6
1.3.3 Descripción general de la solución propuesta....6
1.3.4 Estructura funcional del sistema....6
1.3.5 Descripción de fórmulas y cálculos empleados....7
1.3.6 Información que se manejará....7
1.3.7 Modelo de datos....9
1.3.8 Estructura de códigos....9
1.3.9 Condicionantes de diseño....9
CAPÍTULO 2: "MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS"
2. MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS 11
2.1 AMBIENTE COMPUTACIONAL....11
2.1.1 Hardware....11
2.1.2 Software....11
2.2 DESCRIPCIÓN DE ARCHIVOS....12
2.2.1 Paciente....12
2.2.2 Reserva de sesión....12

2.2.3 Dieta 13

2.2.4 Dieta y alimentos 13

2.2.5 Comida 14

2.2.6 Venta 14

2.2.7 Detalle de venta 14

2.2.8 Productos 15

2.2.9 Compra 15

CAPÍTULO 3: "DESCRIPCIÓN DE PROGRAMAS"

3. DESCRIPCIÓN DE PROGRAMAS 18

3.1 DIAGRAMA DE MENÚS 18

3.2 DIAGRAMA DE MODULAR O DE ESTRUCTURA 18

3.3 TABLA DE PROGRAMAS DESARROLLADOS EN EL SISTEMA 19

3.3.1 Tabla de programas 19

3.4 DESCRIPCIÓN DETALLADA DE PROGRAMAS SELECCIONADOS 20

3.4.1 Reserva de sesiones 20

3.4.2 Consultar Pacientes 21

3.4.3 Informe de ventas 22

3.4.4 Dieta alimentos 24

3.4.5 Administrar productos 26

3.4.6 Agregar/Modificar producto 27

CONCLUSIONES 29

BIBLIOGRAFÍA 30

ANEXO 1: CÓDIGO FUENTE 31

1.1 REGISTRO DE SESIONES 31

1.2 CONSULTAR PACIENTE 36

1.3 INFORME DE VENTAS 39

1.4 DIETA ALIMENTOS 40

1.5 ADMINISTRAR PRODUCTOS 50

1.6 AGREGAR/MODIFICAR PRODUCTOS 53

ÍNDICE DE FIGURAS

Figura 1-1 “Fichas de pacientes del smartphone”. 4
Figura 1-2: “Modelo de datos relacional” 9
Figura 3-1: “Diagrama de menús” 18
Figura 3-2: “Diagrama modular” 18
Figura 3-3: “Diagrama de bloques reserva de sesiones” 20
Figura 3-4: “Vista de usuario reserva de sesiones” 21
Figura 3-5: “Diagrama de bloques consultar pacientes” 21
Figura 3-6: “Vista de usuario consultar pacientes” 22
Figura 3-7: “Diagrama de bloques informe de ventas” 23
Figura 3-8: “Vista de usuario informe de ventas” 23
Figura 3-9: “Diagrama de bloques agregar dieta” 24
Figura 3-10: “Diagrama de bloques modificar dieta” 24
Figura 3-11: “Vista de usuario agregar dieta” 25
Figura 3-12: “Vista de usuario modificar dieta” 26
Figura 3-13: “Diagrama de bloques administrar productos” 26
Figura 3-14: “Vista de usuario administrar productos” 27
Figura 3-15: “Diagrama de bloques agregar productos” 27
Figura 3-16: “Diagrama de bloques modificar productos” 27
Figura 3-17: “Vista de usuario agregar productos” 28
Figura 3-18: “Vista de usuario modificar productos” 28

ÍNDICE DE TABLAS

Tabla 1-1: Entidades de información 8
Tabla 2-1: Paciente 12
Tabla 2-2: Reserva de sesión 13
Tabla 2-3: Dieta 13
Tabla 2-4: Dieta y alimentos 14
Tabla 2-5: Comida 14
Tabla 2-6: Venta 14
Tabla 2-7: Detalle de venta 15
Tabla 2-8: Productos 15
Tabla 2-9: Compra 16
Tabla 3-1: “Lista de programas del sistema” 20

INTRODUCCIÓN

Hoy en día las tecnologías de la información están por doquier; por tanto, es de suma importancia que todo proyecto que se quiera realizar tenga a su disposición estas tecnologías que ayudan considerablemente a realizar de una manera más grata todas las operaciones que se realicen en el proyecto.

Es muy difícil de imaginar que existan grandes empresas; como las de retail, que no tengan un sistema informático para poder manejar sus operaciones. Hay empresas y organizaciones que no necesitan de uno; sin embargo, el que una empresa u organización tenga un sistema de informático siempre será de gran ayuda.

Es por lo anterior que la empresa Health Nutrition and Beauty Ltda. desea complementar los servicios y productos que provee con tecnologías de la información.

La empresa Health Nutrition and Beauty Ltda es una empresa que pertenece al área de la salud, la cual busca mejorar la calidad de vida de las personas mediante productos y métodos naturales. Esta organización está en el mercado de la nutrición y salud natural desde el año 2014 y desde entonces ha realizado sus operaciones sin tener sistemas informáticos que le ayuden a realizar sus operaciones aparte de un notebook con el software Excel para poder realizar dietas y un teléfono inteligente.

Uno de los tantos motivos por los que esta empresa necesitaría un sistema informático es por la cantidad de pacientes que ésta posee, de los cuales se tiene información importante que no está organizada, asegurada y es de difícil acceso, por lo cual tener un sistema informático con bases de datos que permita tener toda la información organizada y seguro ayuda en gran medida a todas sus operaciones.

CAPÍTULO 1:

“ASPECTOS RELEVANTES DEL DISEÑO LÓGICO”

1. ASPECTOS RELEVANTES DEL DISEÑO LÓGICO

1.1 DESCRIPCIÓN DE LA ORGANIZACIÓN

La empresa "Health Nutrition Ltda." que es una sociedad Limitada conformada por 2 socios: Diana Andrea Tapia Urra y Pablo David Pozo Tapia, pertenece al área de la salud, estrictamente de la nutrición, con el objetivo de mejorar la calidad de vida de las personas mediante métodos dietas y productos naturales, provee de servicios de consultas nutricionales que tienen una duración aproximada de 15 minutos, las cuales pueden ser por 2 motivos: a) personas que buscan sanar alguna enfermedad mediante métodos naturales, las que buscan bajar de peso o reducir tallas y b) la venta de productos naturales para los fines acordes.

La organización tiene su principal oficina en Dr. Alonso Zumaeta #137, La Calera. También atiende en Arlegui 333 oficina 114, Edificio Marina Centro, Viña del Mar y en Freire 455, Valparaíso. Todas están ubicadas en la Quinta Región. Las dos últimas ubicaciones se tuvieron que abrir ya que la cantidad de personas que eran de Valparaíso y Viña del Mar era considerablemente amplia como para tener sedes en estas ciudades. En las 3 localizaciones trabaja solamente una nutricionista.

1.2 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL

El principal trabajo que se hace en esta empresa es aportar en la sanación variadas enfermedades a través de nutrientes, nutrientes que están en mayor concentración en los productos que se venden aquí, así como también el bajar de peso y reducir tallas mediante dietas y también productos naturales. Todas las operaciones de esta organización las lleva a cabo una sola persona; es decir, las labores de secretaria/recepcionista, nutricionista, vendedora y administración las realiza sola una persona. Para describir de mejor manera la situación actual, se enumerarán y describirán los diferentes tópicos de lo que se realiza en la empresa. Cabe destacar previamente que no existen "fichas de pacientes", sólo es información del paciente que está escrita en una agenda tanto física como la agenda de un teléfono inteligente (smartphone).

1.2.1 Toma de horas

Actualmente la toma de horas se realiza de forma manual; dentro del horario en el que se toman horas, se contestan llamados telefónicos (se pueden tomar horas mediante estos llamados telefónicos o agendar horas futuras cuando el cliente está en una de sus sesiones). Existen 3 lugares en donde se atiende: La Calera, Viña del Mar y Valparaíso. Al tomar un llamado telefónico de La Calera se procede a revisar dentro de una agenda física las horas disponibles para poder asignarle una al paciente y, si es de Viña del Mar o Valparaíso, se revisa la agenda que está dentro del smartphone de la nutricionista. Al tomar la hora se le pregunta el nombre y número telefónico.

1.2.2 Sesiones

Cuando el paciente asiste a las sesiones, la nutricionista pregunta el por qué está asistiendo. Si es un paciente que viene para poder bajar de peso o reducir talla, se agrega la información pertinente a su **"ficha de paciente".** Si es un paciente que viene para sanar alguna enfermedad se le recetan productos naturales para poder ayudar en la mejoría su enfermedad. En ambos casos se anotan sus enfermedades y/o alergias.

1.2.3 Ficha de pacientes

Como se menciona en un comienzo, no existen "fichas de pacientes", sino que en una agenda física o en la agenda de un smartphone se anotan fecha de la sesión, nombre, número de teléfono, enfermedad y productos que está consumiendo. Cuando el paciente está en su sesión, se agregan a su "ficha" datos tales como, peso, altura, edad y medida de abdomen en el caso de los pacientes que van para poder bajar de peso. En el caso de los pacientes que van por enfermedades, se agrega en su "ficha" los productos que se le recetaron. En el caso de los pacientes de La Calera dichas fichas están escritas en un cuaderno escritos con lápiz y papel y, en el caso de Valparaíso/Viña del Mar, están escritas en la agenda del Smartphone.

En la figura 1-1, se puede ver cómo es la agenda del smartphone en la cual están los pacientes de Valparaíso y Viña del Mar mezclados, Se puede obtener de algunas horas el nombre de los pacientes , la hora, enfermedades o alergias, el producto que se le recetó y cantidad en algunos casos, el lugar en donde se atiende (Valparaíso y Viña del Mar), teléfono de contacto y, los casos de las personas que van para bajar de peso y reducir tallas, están sus medidas.

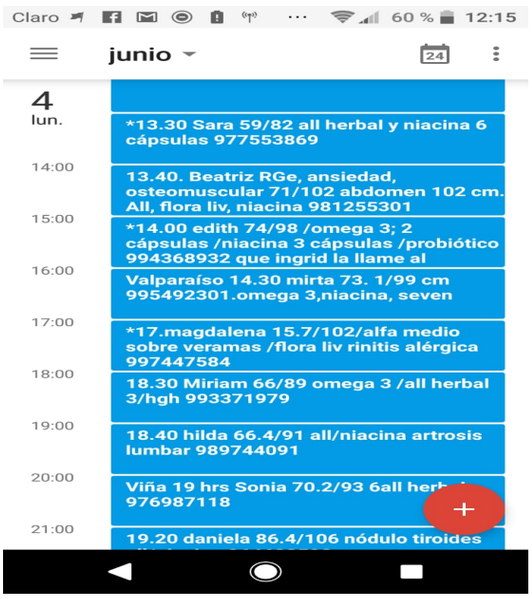

Figura 1-1 "Fichas de pacientes del smartphone".
Fuente: Captura de pantalla agenda de smartphone usado por la nutricionista.

1.2.4 Dietas

Una vez que los pacientes están siendo atendidos en la consulta, se procede a confeccionar las dietas, dietas que se crean en una planilla Excel en la cual se van ingresando los alimentos de manera manual de acuerdo a la situación de cada paciente y sus enfermedades y/o alergias. También existen dietas predeterminadas, que son usadas como

base para otras, las cuales varían dependiendo de las enfermedades y/o alergias que presente el paciente. Estas se basan en diferentes alimentos que se agrupan en comidas tales como desayuno, almuerzo, once, cena, colaciones y los mismos productos que la nutricionista comercializa (aplicados como suplemento).

1.2.5 Ventas

Esta organización posee 4 productos propios, de los cuales se tiene un stock que está anotado en una libreta física. Las ventas se realizan de forma manual, así como también el **control de stock** de productos. Cuando se hace efectiva la venta de un producto se va a la libreta y se tachan del stock los productos vendidos, para así poder llevar la cuenta de cuántos productos se han vendidos y así poder ver cuándo pedir más productos. Dentro de los productos que se comercializa se encuentran: "ALL"," ALL HERBAL"," NIACINA" y "SEVEN SWEET LIFE ", todos estos productos son creados y patentados por la nutricionista.

1.2.6 Productos

En caso de que la nutricionista quiera crear un producto nuevo, ésta llama a la empresa Colón Chile (quien produce y envasa los productos), y les indica las cantidades específicas y el nombre para la creación de éste, además de la cantidad de productos de ese encargo. En cambio, si sólo quiere hacer un encargo de productos que ya están anteriormente creados, sólo les dice el nombre y la cantidad de productos que requiere.

1.3 PROBLEMAS DETECTADOS

- Uno de los principales problemas es claramente la toma de horas, ya que la nutricionista debe revisar en una agenda física las horas disponibles.
- Modificar las horas de consultas, ya que como la agenda es física, para borrar una hora se debe aplicar corrector o se tiene que tachar y escribir de nuevo, lo que lo hace tedioso, así como también borrar en el smartphone para los pacientes de Viña y Valparaíso.
- Acceder de forma lenta a la ficha de los pacientes, ya que éstos se encuentran en un cuaderno o en un smartphone de manera desorganizada.
- Inconsistencia en la información de los pacientes, ya que se encuentran divididos en pacientes de La Calera y Valparaíso/Viña del Mar.
- Pérdida de información de los pacientes al estar toda la información en un cuaderno que puede ser siniestrado al igual que en caso del smartphone.
- Lentitud al crear las dietas, ya que se deben realizar de manera manual en Excel.
- El control de stock es poco confiable, está sujeto a errores de cálculo, y a pérdida de información.
- No existe información sobre si hay ganancias o pérdidas.

1.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO

1.3.1 Objetivos

Confeccionar un software de tipo monousuario stand alone, que permite a la empresa Health Nutrition and Beauty Ltda. poder gestionar de manera efectiva y organizada los servicios nutricionales, centrándose en el registro de consultas, pacientes, manejo de las dietas y ventas de productos (relacionados a la nutrición).

1.3.2 Objetivos específicos del sistema propuesto

- Manejar el proceso de agenda de horas a pacientes, con control de asistencia a las consultas.
- Tener un sistema que permita crear las dietas y vincularlas con los pacientes a través de sus sesiones.
- Implementar una base de datos, en donde se almacene los datos de los pacientes y de la cual se pueda extraer información de manera rápida y eficiente.
- Tener un control de cada sesión de los pacientes para poder ver su progreso a través del tiempo.
- Llevar un conteo del stock de los productos disponibles.
- Llevar de forma ordenada un conteo de las ganancias generadas por ventas de productos

1.3.3 Descripción general de la solución propuesta

El sistema se desarrollará para computadoras personales, se optó por elegir que sea una aplicación de escritorio, ya que, para realizar ciertas operaciones, tales cómo crear dietas, es mucho más cómodo realizarlas con un teclado, ratón y una pantalla de mayor tamaño.

Al iniciar la aplicación, el sistema desplegará una pantalla de acceso en donde el usuario deberá ingresar su nombre de usuario y contraseña.

Luego de autenticar el acceso, se mostrará la pantalla principal. En la parte superior está el menú del programa con todos los módulos del sistema asociados a submenús.

El sistema permitirá realizar todas las operaciones que eran llevadas a cabo de manera manual a una forma digital, permitiendo un control de usuarios y sus sesiones óptimo, creación de dietas y control de las ganancias y/o pérdidas por venta de productos.

1.3.4 Estructura funcional del sistema

El sistema por desarrollar tendrá las siguientes funcionalidades:

- **Mantenedor de Pacientes**: Se podrán hacer funciones básicas respecto a los pacientes (agregar, modificar, consultar y eliminar pacientes).
- **Mantenedor de productos**: Poseerá funciones básicas respecto a los productos (agregar, modificar, eliminar productos, consultar productos y su stock).

- **Mantenedor de dietas**: Se podrá confeccionar dietas nuevas que quedarán almacenadas y podrán ser usadas de forma predefinidas, así como también modificar y/o eliminar las dietas.
- **Mantenedor de Registro de Horas de atención a pacientes**: El sistema podrá agregar, modificar, consultar y eliminar hora de los pacientes.
- **Registro de Sesiones**: Se podrán agregar datos adicionales de los pacientes (estatura, peso, etc.) de la sesión actual.
- **Asignación de dietas**: Se podrán asignar dietas a un paciente en su sesión actual.
- **Registrar venta**: Registra una venta de productos a un paciente.
- **Registrar compra:** Registra las compras de productos de la empresa (para su comercialización).
- **Consultar información histórica**: Se podrá acceder a la información histórica de las sesiones de cada paciente.
- **Consultar ganancia o pérdida**: Mostrará el cálculo de las ganancias o pérdidas de los productos vendidos en un intervalo de fechas, generando así un listado de las ventas y mostrando si hubo ganancia o pérdida.

1.3.5 Descripción de fórmulas y cálculos empleados

Avances Paciente: El único cálculo utilizado a lo largo de sesiones es el índice de masa corporal (IMC), que se determina dividiendo el peso del paciente (kg) por la estatura (m) al cuadrado.

$$IMC = \frac{PESO\ (kg)}{ESTATURA(m) * ESTATURA(m)}$$

Ventas(ganancias/pérdidas): Para el cálculo de ganancia y/o pérdidas por venta de productos se efectúa primero calculando el costo promedio de cada producto que se vende en la empresa. De acuerdo al intervalo dado en el cual se desea visualizar si hubo ganancias o pérdidas, se suman las ventas de productos multiplicando la cantidad de productos vendidos por el precio de venta, a esta cantidad se le resta la misma cantidad de productos vendidos, pero es multiplicado por el promedio de costo del producto.

1.3.6 Información que se manejará

Entradas:

- Datos de Reserva y sesión (día, hora, rut, numero_sesion, estado, nombre, estatura, peso, cintura, cod_dieta, observaciones)
- Datos del Paciente (rut, nombre, apellidos, fecha_nac, celular, email, direccion, enfermedades)
- Datos del Producto (cod_prod, nombre, precio, stock)
- Datos de la Venta ([cod_venta, rut, tipo_pago, fecha, total] detalle [cod_producto, cod_venta, cantidad, precio_venta])
- Datos de la Compra (cod_compra, cod_producto, cantidad, fecha, costo, total)
- Datos de las Dietas ([cod_dieta, nombre, descripción] detalle [cod_dieta, dia, hora, cod_comida, alimentos])
- Datos de tipo de comida (cod_comida, nombre)

Salidas:

- **Horario de consultas nutricionales:** contiene un horario diario en el cual se verán las horas disponibles, confirmadas y pendientes. Esta salida es por pantalla.
- **Lista de dietas:** salida en pantalla contiene una lista de todas las dietas confeccionadas.
- **Consulta dieta específica:** salida en pantalla contiene la información específica de una sola dieta.
- **Información histórica de paciente:** contiene en pantalla todas las consultas de un paciente específico durante el tiempo (mostrando observaciones, tallas, peso, estatura etc).
- **Listado de pacientes:** Contiene en pantalla una lista con toda la información de todos los pacientes (los pacientes pueden ser filtrados por rut, nombre o email).
- **Cálculo de ventas:** Informe por pantalla de las ventas de los productos y balance de ganancias o pérdidas, en un intervalo de fechas.
- **Listado de ventas:** Contiene en pantalla todas las ventas que se han realizado a lo largo del tiempo (las ventas pueden ser filtradas por código de venta o un rango de fechas).
- **Listado de existencias de productos**: Contiene en pantalla el stock de los productos que están a la venta.
- **Listado de compras:** Contiene en pantalla todas las compras que se han realizado a lo largo del tiempo.

Entidades de información

Tabla 1-1: Entidades de información
Fuente: Elaboración propia

PACIENTE	Contiene los datos de los pacientes
RESERVA_SESION	Contiene los datos para registrar, confirmar o eliminar una reserva de hora y datos asociados al seguimiento de las consultas
DIETA	Contiene todas las dietas con sus respectivas descripciones
DIETA_ALIMENTO	Servirá de conexión entre la entidad Dieta y todos los alimentos de una dieta
VENTA	Contiene los datos de una venta de productos
DETALLE_VENTA	Contiene todos los productos de una sola venta
PRODUCTOS	Contiene todos los datos de los productos a la venta
COMPRA	Contiene los datos de compras de productos
COMIDA	Contiene todos los tipos de comida

1.3.7 Modelo de datos

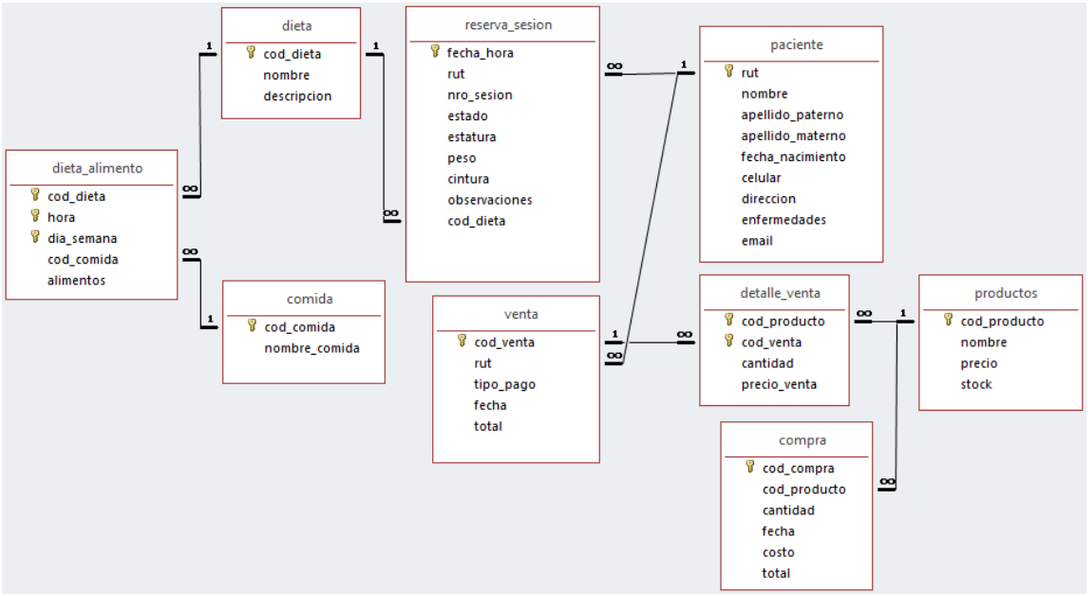

Figura 1-2: "Modelo de datos relacional"
Fuente: Elaboración propia.

Las claves foráneas son identificadas con el campo donde apunta la flecha.

1.3.8 Estructura de códigos

Para poder manejar los datos en este sistema, se usarán los siguientes códigos para poder identificar y manejar de mejor manera los datos.

paciente: La clave principal de pacientes será el Rut de cada paciente.

reserva_sesion: La clave principal de cada reserva_sesion será una clave compuesta por el Rut del paciente y la fecha de la reserva.

Estado: El estado de la hora de reserva puede ser:

"T": Terminada (paciente asistió a la consulta), **"P"**: Por Confirmar y **"C"**: Confirmada

Comida: La clave principal de la tabla comida es cod_comida, un número correlativo, con este se puede identificar los diferentes tipos de comida existentes en un día (almuerzo, desayuno, etc).

Venta: Tipo de pago es un código que permite identificar los diferentes tipos de pago que soporta el sistema.

1.3.9 Condicionantes de diseño

Para poder confeccionar este sistema, ya que es el primero de esta empresa y es monousuario, se tiene un amplio espectro de lenguajes de programación posibles. Por comodidad de programación y porque la interfaz de usuario final es más amigable, el lenguaje utilizado será Visual Basic.NET.

Como este sistema no tendrá un gran volumen de datos y puede ser fácilmente portable, la base de datos a utilizar será Access.

CAPÍTULO 2:

“MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS”

2. MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE ARCHIVOS

2.1 AMBIENTE COMPUTACIONAL

2.1.1 Hardware

Ambiente de desarrollo

Apple MacBook Pro-15

Procesador: Intel Core i7 Quad-Core a 2.6 GHz

RAM: Memoria 16 GB DDR4

Sistema Operativo: Windows 10 64 bits.

Dell inspiron 15-7559

Procesador: Intel Core i7 6700HQ Quad-Core a 2.6 GHz

RAM: Memoria 8 GB DDR3L

Sistema Operativo: Windows 10 64 bits.

Ambiente de Ejecución

HP ProBook 450 G4

Procesador: Intel Core i7 7400U a 2.7 GHz

RAM: Memoria 4 GB DDR4

Sistema Operativo: Windows 10 64 bits

2.1.2 Software

Ambiente de desarrollo

Visual Studio 2017: Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc. Proporciona comodidad a la hora de programar y ofrece distintos módulos para agregar más funcionalidades.

Visual Basic.NET: Se decidió usar este lenguaje de programación con el framework de .NET por el simple hecho de que facilita mucho el traspaso de datos entre formularios además posee funciones integradas bastante útiles.

Windows 10: Este sistema operativo fue elegido para el desarrollo ya que con Visual Studio ofrece grandes avances en soporte multiplataforma respecto a la competencia.

Ambiente de Ejecución

Windows 10: Este sistema operativo fue elegido porque nuestro cliente ya conoce como utilizar este sistema operativo.

2.2 DESCRIPCIÓN DE ARCHIVOS

2.2.1 Paciente

Nombre	paciente.		
Descripción	Esta tabla contiene datos de los pacientes.		
Clave primaria	**rut.**		
Descripción de la tabla			
Nombre	**Tipo**	**Longitud**	**Descripción**
rut	varchar	10	Rut del paciente
nombre	varchar	20	Nombre del paciente
apellido_paterno	varchar	25	Apellido paterno del paciente
apellido_materno	varchar	25	Apellido materno del paciente
fecha_nacimiento	date	N/A	Fecha de nacimiento del paciente
celular	int	9	Celular del paciente
direccion	varchar	50	Dirección del paciente
enfermedades	varchar	200	Descripción de las enfermedades del paciente
email	varchar	60	Email del paciente

Tabla 2-1: Paciente
Fuente: Elaboración propia.

2.2.2 Reserva de sesión

Nombre lógico	Reserva de sesión.		
Nombre físico	reserva_sesion.		
Descripción	Esta tabla contiene todos los datos de las reservas de los pacientes, junto con los datos posteriores de la sesión correspondiente, de las sesiones de éstos.		
Clave primaria	Clave compuesta por los campos **fecha_hora**.		
Clave foránea	cod_dieta a tabla dieta y rut a tabla paciente		
Descripción de la tabla			
Nombre	**Tipo**	**Longitud**	**Descripción**
fecha_hora	datetime	N/A	Fecha y hora de una reserva de sesión
Rut	varchar	10	Rut del paciente
nro_sesion	smallint	3	Número correlativo que indica la sesión a la cual está asistiendo el paciente a lo largo de su historia
estado	char	1	Campo que indica el estado de una reserva, que pueden ser las siguientes: ‘T’: terminada, ‘P’: por confirmar, ‘C’: confirmada

estatura	float	3	Estatura del paciente en la sesión (metros)
peso	float	5	Peso del paciente en la sesión(kilos)
cintura	float	5	Medida de la cintura del paciente en la sesión(centímetros)
observaciones	varchar	1000	Observaciones sobre el paciente
cod_dieta	smallint	4	código de la dieta asignada en la sesión

Tabla 2-2: Reserva de sesión
Fuente: Elaboración propia.

2.2.3 Dieta

Nombre	dieta		
Descripción	Esta tabla contiene la descripción general de todas las dietas disponibles para ser asignadas a una reserva_sesion.		
Clave primaria	**cod_dieta.**		
Descripción de la tabla			
Nombre	**Tipo**	**Longitud**	**Descripción**
cod_dieta	smallint	4	Código incremental usado para identificar una dieta
nombre	varchar	20	Nombre de la dieta
descripción	varchar	200	Descripción sobre lo que trata dicha dieta, así como también para qué enfermedades y/o alergias es más apta

Tabla 2-3: Dieta
Fuente: Elaboración propia.

2.2.4 Dieta y alimentos

Nombre lógico	Dieta y alimentos (detalles de una dieta).		
Nombre físico	dieta_alimento.		
Descripción	Esta tabla contiene el detalle de cada una de las dietas, vale decir, cada comida que constituye una dieta.		
Clave primaria	Clave compuesta por los campos **cod_dieta + hora + dia_semana**.		
Clave foránea	cod_comida a tabla comida.		
Descripción de la tabla			
Nombre	**Tipo**	**Longitud**	**Descripción**
cod_dieta	smallint	4	Código de la dieta
hora	datetime	N/A	Hora a la que se debe ingerir cada comida
dia_semana	tinyint	2	Código que indica el día al que pertenece el alimento (Ej: “lunes = 1 ,” martes = 2”, etc.)
cod_comida	tinyint	2	Código de los tipos de comidas existentes ya sea desayuno, almuerzo, cena, etc

alimentos	varchar	150	Todos los alimentos que pertenecen a una comida, para esa dieta específica.

Tabla 2-4: Dieta y alimentos
Fuente: Elaboración propia.

2.2.5 Comida

Nombre	comida.		
Descripción	Tabla que contiene todas las posibles comidas a lo largo de un día (desayuno, colación, almuerzo, etc.).		
Clave primaria	**cod_comida.**		
Descripción de la tabla			
Nombre	**Tipo**	**Longitud**	**Descripción**
cod_comida	tinyint	1	Código identificador del tipo de comida
nombre_comida	varchar	10	Nombre del tipo de comida (desayuno, almuerzo, etc)

Tabla 2-5: Comida
Fuente: Elaboración propia.

2.2.6 Venta

Nombre	venta.		
Descripción	Esta tabla contiene los datos de una venta de producto(s).		
Clave primaria	**cod_venta.**		
Clave foránea	rut a tabla paciente.		
Descripción de la tabla			
Nombre	**Tipo**	**Longitud**	**Descripción**
cod_venta	mediumint	6	Código incremental usado para identificar una venta
Rut	varchar	10	Rut del paciente
tipo_pago	tinyint	1	Codigo que indica el tipo de pago de una venta, con las opciones: “1= Efectivo” o “2 = Tbk”
Fecha	datetime	N/A	Fecha y hora de una venta
Total	mediumint	5	Total de la venta

Tabla 2-6: Venta
Fuente: Elaboración propia.

2.2.7 Detalle de venta

Nombre lógico	Detalle de venta.		
Nombre físico	detalle_venta.		
Descripción	(intersección) Esta tabla es el detalle de cada producto en una venta.		
Clave primaria	Clave compuesta por los campos **cod_venta + cod_prod.**		
Descripción de la tabla			
Nombre	**Tipo**	**Longitud**	**Descripción**
cod_venta	mediumint	6	Código identificador de una venta
cod_prod	tinyint	2	Código identificador de un producto
Cantidad	tinyint	2	Cantidad que se venderá del producto
precio_venta	mediumint	5	Precio al que se venderá el producto

Tabla 2-7: Detalle de venta
Fuente: Elaboración propia.

2.2.8 Productos

Nombre	productos.		
Descripción	Esta tabla contiene todos los productos que vende la nutricionista.		
Clave primaria	**cod_prod.**		
Descripción de la tabla			
Nombre	**Tipo**	**Longitud**	**Descripción**
cod_prod	tinyint	2	Código incremental usado para identificar un producto
nombre	varchar	15	Nombre del producto
precio_actual	mediumint	5	Precio actual del producto
stock	tinyint	3	Cantidad del producto en bodega

Tabla 2-8: Productos
Fuente: Elaboración propia.

2.2.9 Compra

Nombre	compra.		
Descripción	Esta tabla contiene los datos de la compra de productos al proveedor.		
Clave primaria	**cod_compra.**		
Clave foránea	cod_prod a tabla productos		
Descripción de la tabla			
Nombre	**Tipo**	**Longitud**	**Descripción**
cod_compra	mediumint	6	Código incremental usado para identificar una compra
cod_prod	tinyint	2	Código identificador del producto que se compra
cantidad	smallint	3	Cantidad que se comprará del producto
fecha	date	N/A	Fecha en la que se hizo la compra
costo_unitario	mediumint	5	Valor de compra del producto
total	mediumint	6	Total de la compra

Tabla 2-9: Compra
Fuente: Elaboración propia.

CAPÍTULO 3:

“Descripción de programas”

3. DESCRIPCIÓN DE PROGRAMAS

3.1 DIAGRAMA DE MENÚS

A continuación, se presenta el diagrama de menús del sistema.

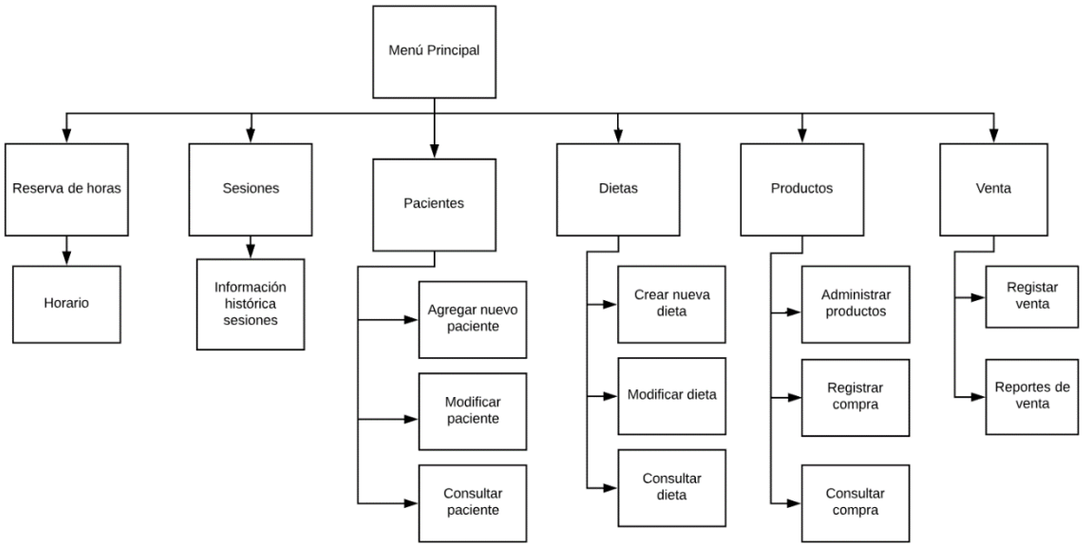

Figura 3-1: "Diagrama de menús"
Fuente: Elaboración propia.

3.2 DIAGRAMA DE MODULAR O DE ESTRUCTURA

A continuación, se presenta el diagrama de estructura del sistema, mostrando todos los módulos del programa y cómo estos se relacionan entre sí.

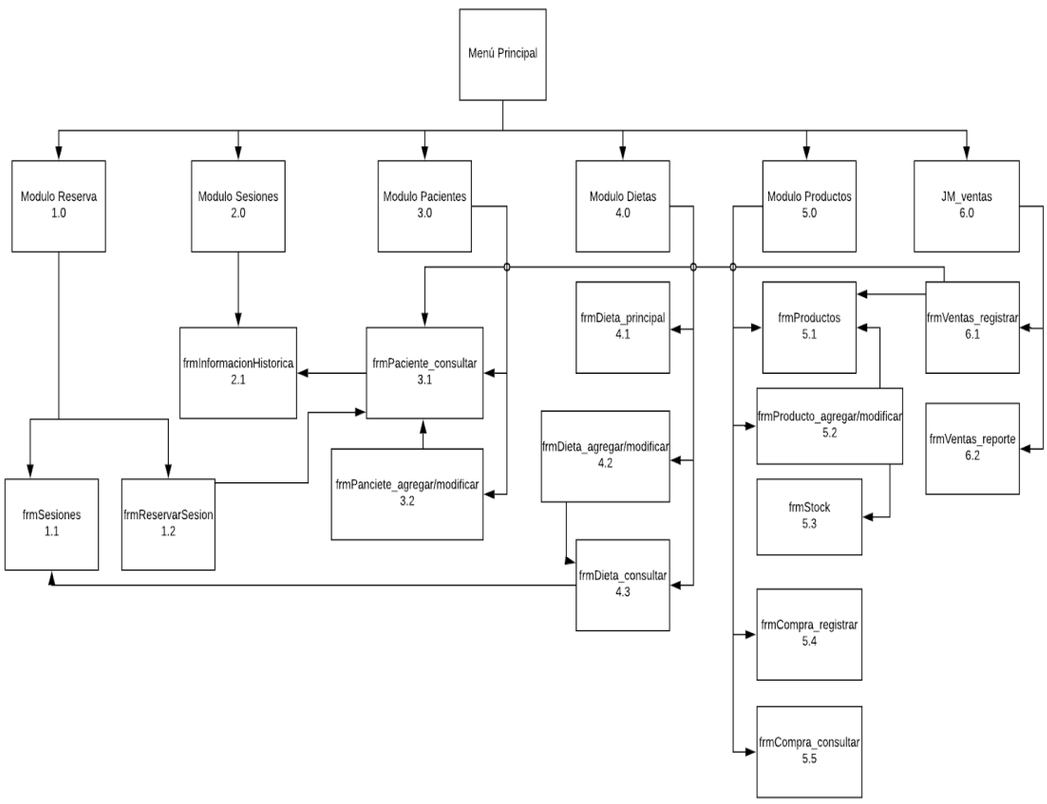

Figura 3-2: "Diagrama modular"
Fuente: Elaboración propia.

3.3 TABLA DE PROGRAMAS DESARROLLADOS EN EL SISTEMA

3.3.1 Tabla de programas

En la siguiente tabla se mostrarán todos los programas utilizados en el sistema, cada uno con su respectivo objetivo. Los nombres de programas que están antecedidos por un * serán descritos detalladamente en el punto 3.4.

LOGIN	Permite validar el usuario y entrar al programa principal
MENÚ PRINCIPAL	Permite al usuario acceder a casi todos los otros programas del sistema.
***RESERVA DE SESIÓN**	Permite al usuario consultar las reservas de cualquier fecha y verlas por día, ingresar nuevas reservas, cambiar su estado (Confirmada, Por Confirmar, Terminada) e ir a la sesión previamente registrada.
SESIÓN	Permite registrar todos los datos de la sesión de un paciente en específico.
INFORMACIÓN HISTÓRICA	Permite ver todas las sesiones con sus respectivos datos de un paciente en específico.
***CONSULTAR PACIENTES**	Despliega una lista con todos los pacientes y sus respectivos datos. Desde este formulario se puede eliminar un paciente y llamar al programa de agregar/modificar.
INGRESAR/MODIFICAR PACIENTE	Dependiendo de los parámetros con los que se invoque este programa, se puede registrar un nuevo paciente en el sistema ó modificar alguno de los pacientes ya registrados.
AGREGAR/MODIFICAR DIETA	Dependiendo de los parámetros con los que se llame a este formulario, se puede agregar los datos generales de la dieta (nombre y descripción) o modificar estos datos de una dieta ya existente.
***AGREGAR/MODIFICAR DIETA ALIMENTOS**	Dependiendo de los parámetros con los que se llame a este formulario se puede agregar los alimentos a una nueva dieta o modificar los alimentos de una ya existente.
CONSULTAR DIETAS	Permite al usuario ver una lista con todas las dietas (datos generales), dando la posibilidad de eliminar o ver el detalle de una dieta.
***ADMINISTRAR PRODUCTOS**	Permite al usuario ver una lista con todos los productos y su respectiva información, se puede eliminar un producto y llamar al programa de agregar/modificar un producto
***AGREGAR/MODIFICAR PRODUCTO**	Permite al usuario agregar o modificar un producto.
REGISTRAR COMPRA	Permite al usuario registrar una nueva compra, mostrando un listado de todos los productos para hacer más cómoda la selección de qué producto compró la nutricionista.
CONSULTAR COMPRAS	Permite al usuario ver una lista con todas las compras, dando la posibilidad de eliminar una en específico.

REGISTRAR VENTA	Permite al usuario registrar una nueva venta, dando la posibilidad de ir ingresando el detalle de los productos que vendió.
***INFORME DE GANANCIAS/PÉRDIDAS**	Permite al usuario poder visualizar las ganancias y/o pérdidas por venta de productos de acuerdo a un intervalo de fechas.
AGREGAR/RESTAR STOCK	En caso de que la realidad no coincida lo que dice el sistema respecto a la cantidad de productos en existencia, este programa permite agregar o restar stock manualmente.

Tabla 3-1: "Lista de programas del sistema"
Fuente: Elaboración propia.

3.4 DESCRIPCIÓN DETALLADA DE PROGRAMAS SELECCIONADOS

3.4.1 Reserva de sesiones

Nombre: frmReservarSesion

Objetivo: Permite al usuario consultar las reservas de cualquier fecha y verlas por día, ingresar nuevas reservas, cambiar su estado (Confirmada, Por Confirmar, Terminada) e ir a la sesión previamente registrada.

Referencia código fuente: Anexo 1.1 RESERVA DE SESIONES

Diagrama de bloque:

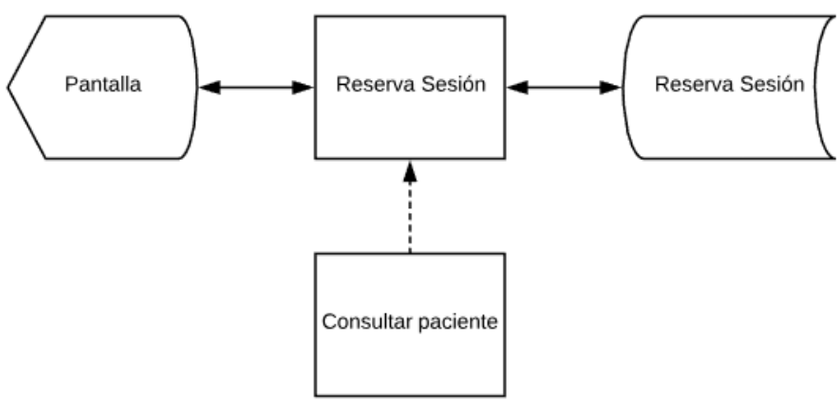

Figura 3-3: "Diagrama de bloques reserva de sesiones"
Fuente: Elaboración propia.

*Nota: El programa frmReservarSesion invoca el programa "Consultar paciente" para extraer los datos del paciente. Se indica mediante línea punteada.

Reglas del proceso:

Se presenta en la pantalla (figura 3-4) una opción de "revisar horario" (ubicada en la parte superior izquierda de la pantalla) en la que se selecciona una fecha y de acuerdo a la fecha seleccionada el programa obtiene desde la tabla de reserva_sesion los datos de esa fecha y se rellenan automáticamente las horas que están disponibles y las tomadas para ese día. Una vez seleccionada la fecha, con las horas disponibles, el usuario puede reservar una consulta y para ello es necesario ingresar los datos del paciente. Para ingresar los datos se accede al formulario de consulta pacientes a través del botón "Buscar paciente" (sector de la pantalla "Datos de la reserva") del mismo formulario de reserva, una vez abierto el formulario de consulta de pacientes (figura 3-6 del programa Consultar Paciente), se elige

el paciente o, de no existir, se puede agregar uno nuevo desde la misma pantalla. Una vez ingresada la hora se graba inmediatamente en la tabla reserva_sesion.

Con la hora tomada es posible eliminar la hora, para que quede nuevamente disponible y también está la opción de ir a la sesión de dicha hora tomada y proceder con la consulta. También con las horas tomadas se puede cambiar su estado a Confirmada, Por Confirmar y Terminada (automáticamente al terminar la sesión la hora quedará en estado de Terminada).

Vista de usuario:

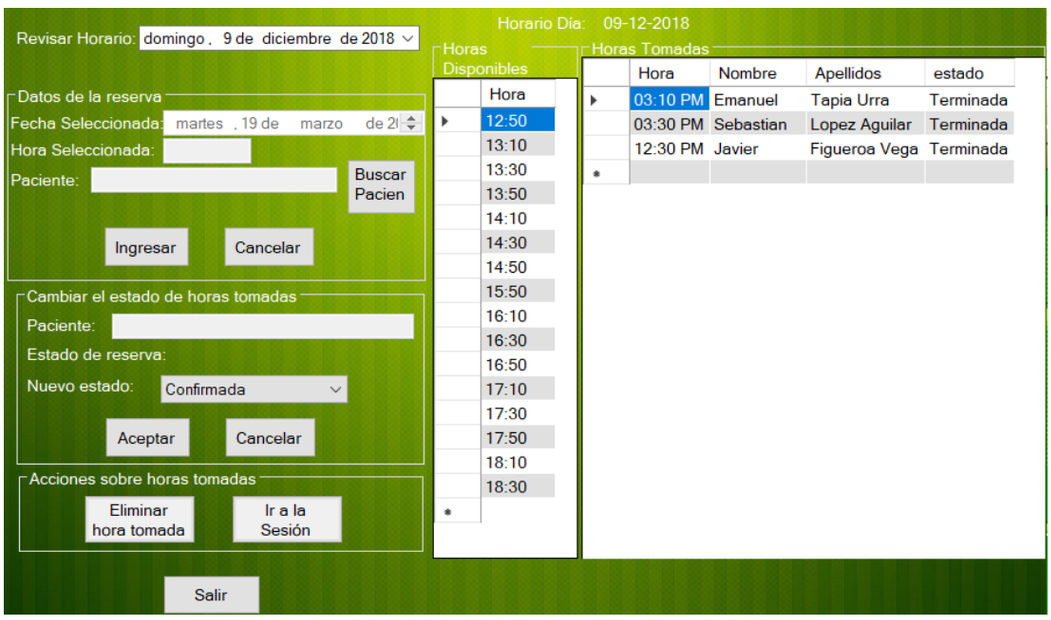

Figura 3-4: "Vista de usuario reserva de sesiones"
Fuente: Elaboración propia.

3.4.2 Consultar Pacientes

Nombre: frmPaciente_consultar

Objetivo: Permite al usuario poder ver todos los pacientes registrados en el sistema con su respectiva información, también se puede eliminar pacientes de la lista y posee botones abrir otros programas que permiten agregar nuevos pacientes ó modificarlos.

Referencia código fuente: Anexo 1.2 CONSULTAR PACIENTE

Diagrama de bloque:

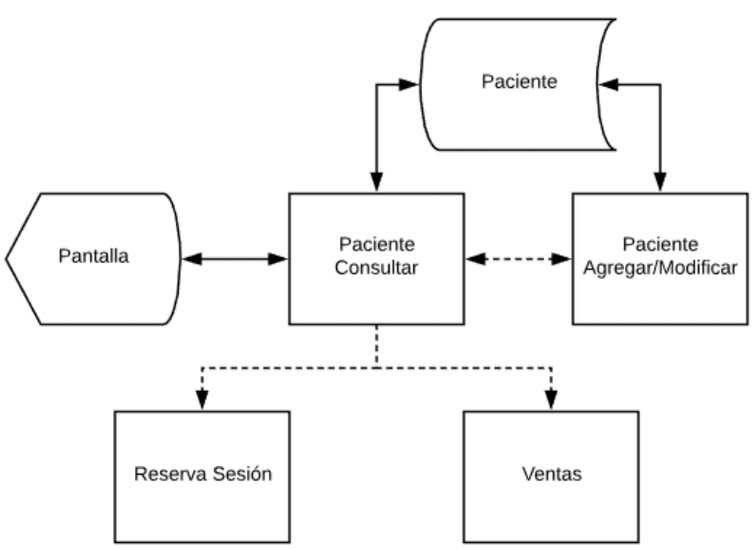

Figura 3-5: "Diagrama de bloques consultar pacientes"
Fuente: Elaboración propia.

*Nota: El programa consultar paciente invoca a otro programa llamado ingresar/modificar paciente en el cual se ingresa información de pacientes nuevos y/o modifica información de pacientes ya ingresados (Es el mismo programa para ambas opciones ya que dependiendo del parámetro de invocación varía su propósito). También el programa Paciente consultar es invocado por el programa Reserva sesión y ventas para poder obtener información de los pacientes. Se indican mediante líneas punteadas.

Reglas del proceso:

Es posible abrir este programa desde el menú principal, así como también desde el programa de reserva de sesiones y el de ventas. Si el programa es invocado desde reserva de sesión ó ventas, aparecerá un nuevo botón de "selección" para poder extraer los datos del paciente seleccionado.

En todos los métodos para acceder a este programa se despliega una lista de todos los pacientes registrados en el sistema (figura 3-6) con su respectiva información extraida desde la tabla paciente.

La lista de pacientes posee filtros de RUT, Nombre y Correo electrónico para que sea más fácil de encontrar los pacientes en caso de tener demasiados.

Existe un botón para poder agregar nuevos pacientes al sistema y si se selecciona un paciente aparecerán botones que darán la opción de eliminar dicho paciente ó modificar su información.

Vista de usuario:

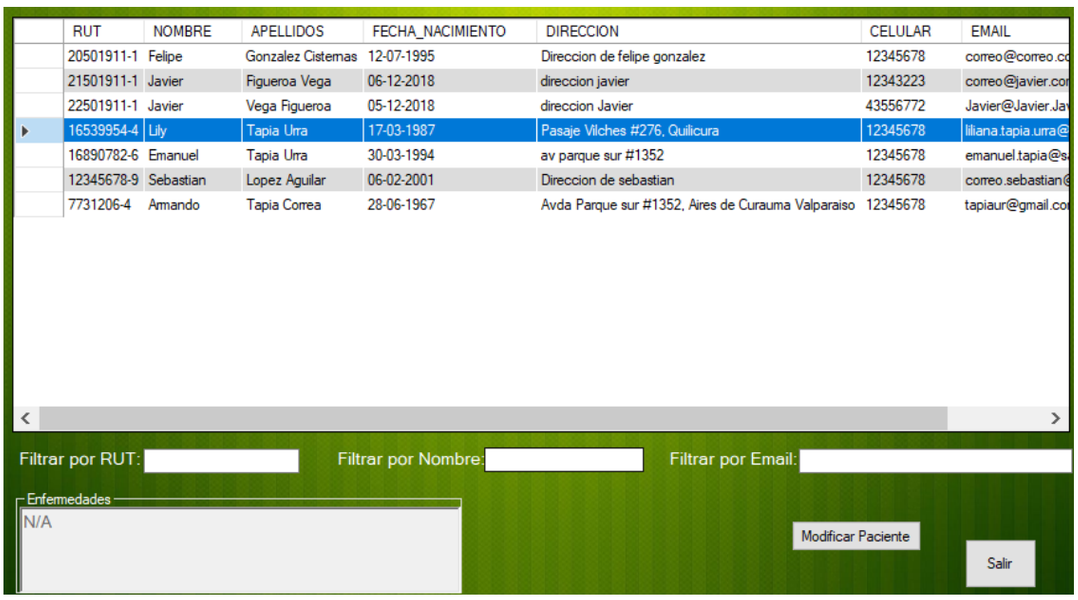

Figura 3-6: "Vista de usuario consultar pacientes"
Fuente: Elaboración propia.

3.4.3 Informe de ventas

Nombre: frmVentas_reporte

Objetivo: Permite al usuario poder visualizar las ganancias y/o pérdidas por venta de productos de acuerdo a un intervalo de fechas que puede ser seleccionado por el usuario.

Referencia código fuente: Anexo 1.3 INFORME DE VENTAS

Diagrama de bloque:

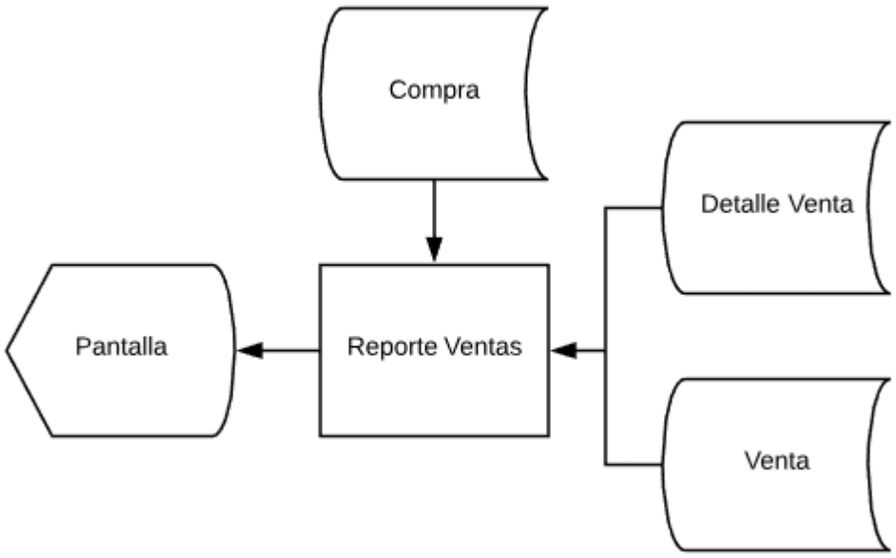

Figura 3-7: "Diagrama de bloques informe de ventas"
Fuente: Elaboración propia.

Reglas del proceso:

El programa despliega la pantalla (figura 3-8), que permite al usuario seleccionar dos campos tipo fecha uno "Desde" y otro "Hasta", en los cuales el usuario puede seleccionar el intervalo dentro del cual desea generar un reporte de ventas, para poder visualizar las ganancias y/o pérdidas.

Para poder calcularlo, previamente se obtiene un promedio de costo de los productos para que así se pueda tener un cálculo más acercado a la realidad. Los datos para poder cálcular este promedio son extraidos desde la tabla de compra.

Luego de acuerdo al intervalo dado, se buscan los datos de las ventas en la tabla venta y detalle_venta que se efectuaron en dicho intervalo, con los datos de las ventas realizadas en ese intervalo se acumulan los datos de valor de la venta final.

Una vez teniendo el promedio de costo de los productos vendidos y los ingresos de los productos vendidos, se procede a calcular si hubo ganancia ó pérdida, mostrando la información en la pantalla (figura 3-8) en lado superior derecho y en la parte de debajo de la pantalla se muestra un detalle de las transacciones (ventas) de productos.

Vista de usuario:

Fecha	Nombre	Código_Venta	Producto	Cantidad	precio_venta	Total
11-12-2018	Lily Tapia Urra	22	All	15	23000	345000
11-12-2018	Javier Vega Figueroa	23	Vitamina C	20	15000	300000
11-12-2018	Javier Vega Figueroa	23	Niacina	15	20000	300000
11-12-2018	Javier Vega Figueroa	23	All Herbal	20	25000	500000
11-12-2018	Javier Vega Figueroa	23	All	5	23000	115000
10-12-2018	Emanuel Tapia Urra	24	All	2	23000	46000
10-12-2018	Emanuel Tapia Urra	24	All Herbal	3	25000	75000
10-12-2018	Emanuel Tapia Urra	24	Niacina	5	20000	100000
10-12-2018	Emanuel Tapia Urra	24	Vitamina C	23	15000	345000
13-12-2018	Lily Tapia Urra	27	Niacina	15	20000	300000
13-12-2018	Lily Tapia Urra	27	Vitamina C	17	15000	255000

Figura 3-8: "Vista de usuario informe de ventas"
Fuente: Elaboración propia.

3.4.4 Dieta alimentos

Nombre: frmDieta_principal

Objetivo: Este formulario tiene dos formas de ser llamado ya sea desde el agregar dieta o el de modificarla.

Agregar: Permite al usuario crear una nueva dieta desde cero, pudiendo agregar la hora, el tipo de comida y el dia en el cual se deben ingerir los alimentos.

Modificar: Permite al usuario modificar los alimentos, la hora, el tipo de comida y el dia en el que deben ser ingeridos. En caso de que la dieta estuviera a medio terminar también se puede completar en este mantenedor.

Referencia código fuente: Anexo 1.4 DIETA ALIMENTOS

Diagrama de bloque:

Agregar:

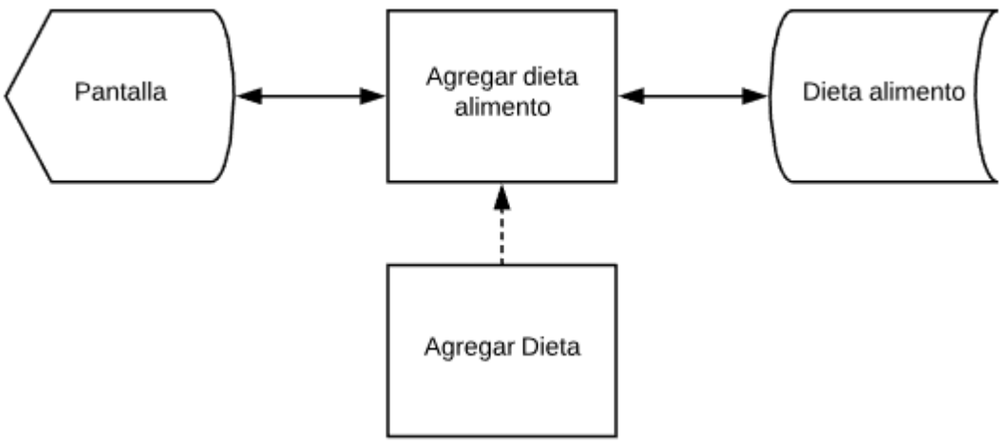

Figura 3-9: "Diagrama de bloques agregar dieta"
Fuente: Elaboración propia.

*Nota: El programa frmDieta_agregar invoca el programa "frmDieta_principal" enviando datos generales de una dieta. Se indica mediante línea punteada.

Modificar:

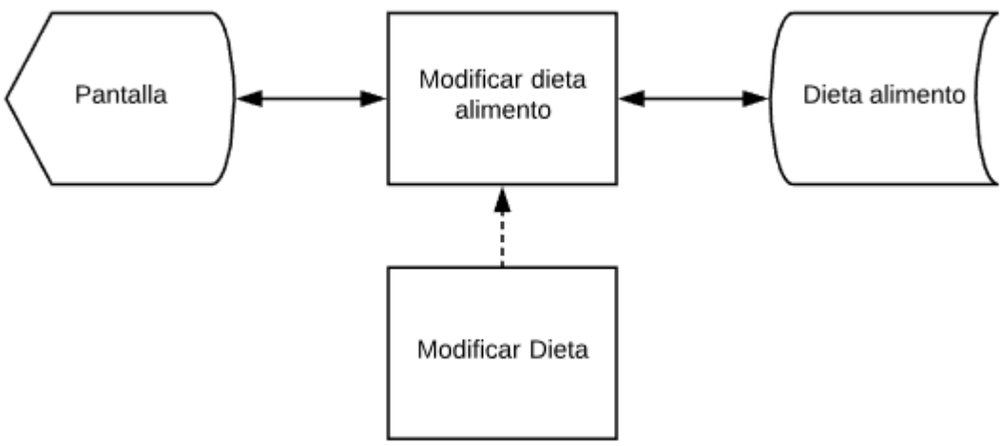

Figura 3-10: "Diagrama de bloques modificar dieta"
Fuente: Elaboración propia.

*Nota: El programa frmDieta_agregar invoca el programa "frmDieta_principal" pasando datos generales de una dieta. Se indica mediante línea punteada.

Reglas del proceso:

Agregar: Si se desea agregar una dieta nueva, se ejecuta frmDieta_agregar el cual despliga la pantalla para registrar el código, el nombre y la descripción de la dieta; estos datos son enviados a través de parámetros al formulario el cual se está describiendo(frmDieta_principal) , se despliega la pantalla del detalle de dieta (figura 3-11) , en este formulario se le permite al usuario la selección el día de la semana, de la hora y que tipo de comida(almuerzo, once, suplemento, etc.) tendrá ese grupo de alimentos, validando antes que cuando se clickee el boton agregar que despliega al texto del nuevo alimento, exista una un tipo de comida y una hora, en caso de que no sea asi se le manda un aviso de que campo le falta. Luego al presionar el botón guardar los registros son almacenados en la tabla dieta_alimento de la base de datos.

Modificar: Si se desea modificar, se ejecuta el mismo frmDieta_agregar (con diferentes parámetros) de agregar esta vez con el botón de buscar todas las dietas registradas con su respectivo nombre y descripción en caso de que estos campos deseen ser modificados; estos datos son enviados a través de parámetros al formulario el cual se está describiendo(frmDieta_principal), se despliega la pantalla del detalle de dieta (figura 3-11), cargándose así todos los datos del detalle de la dieta (respectivos alimentos, su hora y que tipo de comida son), dando posibilidad al usuario de modificar los alimentos o agregar los restantes en caso de que la dieta no esté completa (validando antes que cuando se clickee el boton agregar que despliega al texto del nuevo alimento, exista una un tipo de comida y una hora, en caso de que no sea asi se le manda un aviso de que campo le falta), Luego al presionar el botón guardar los registros son almacenados en la tabla dieta_alimento de la base de datos.

Vista de usuario:

Agregar:

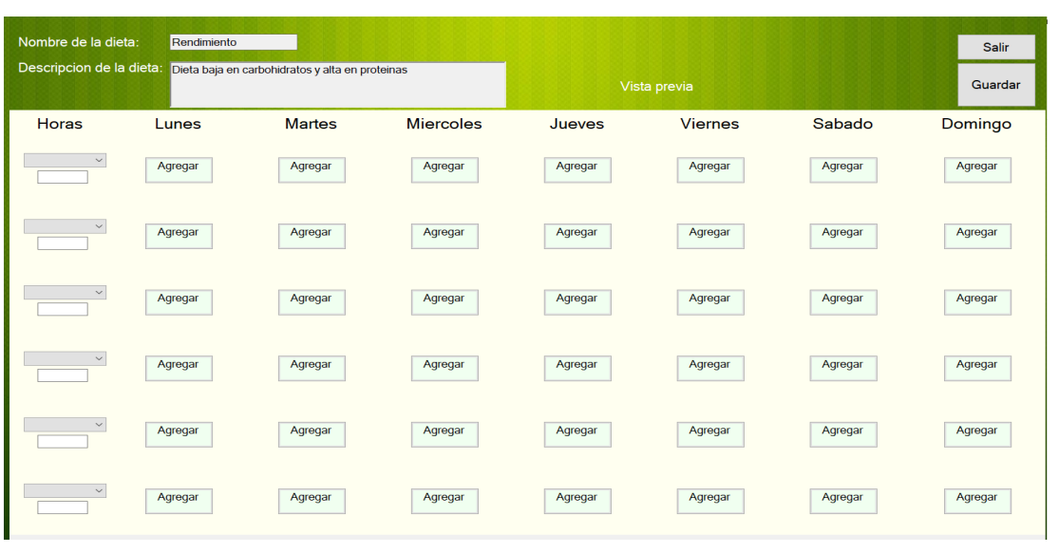

Figura 3-11: “Vista de usuario agregar dieta”
Fuente: Elaboración propia.

Modificar:

Figura 3-12: "Vista de usuario modificar dieta"
Fuente: Elaboración propia.

3.4.5 Administrar productos

Nombre: frmProductos

Objetivo: Permite al usuario visualizar toda la información de los productos actualmente en el sistema, además de ser un gestor para la creación, modificación y eliminación de éstos.

Referencia código fuente: Anexo 1.5 ADMINISTRAR PRODUCTOS

Diagrama de bloque:

Figura 3-13: "Diagrama de bloques administrar productos"
Fuente: Elaboración propia.

Reglas del proceso:

El programa, al ser llamado (figura 3-14), carga todos los datos de la tabla productos en una grilla, dando la posibilidad de hacer tres acciones: agregar, modificar y eliminar.

En caso de que se quiera agregar un producto, se presiona el botón agregar y se llama al formulario frmProdutos_agregar el cual posee los campos necesarios para realizar la acción.

En caso de que se quiera modificar un producto, se selecciona dicho producto en la tabla, y se presiona el botón modificar; luego se llama al formulario frmProdcutos_agregar, pero esta vez se le envían los datos del producto en forma de parámetro (polimorfismo), lo que permite al usuario cambiar dichos datos y posteriormente modificar la base de datos.

En caso de que se quiera eliminar un producto, se selecciona dicho producto en la tabla, y se presiona el botón eliminar. Emerge un mensaje de alerta para cerciorarse de que el usuario está seguro, si el usuario confirma la eliminación, se elimina el registro de la tabla productos de la base de datos.

Vista de usuario:

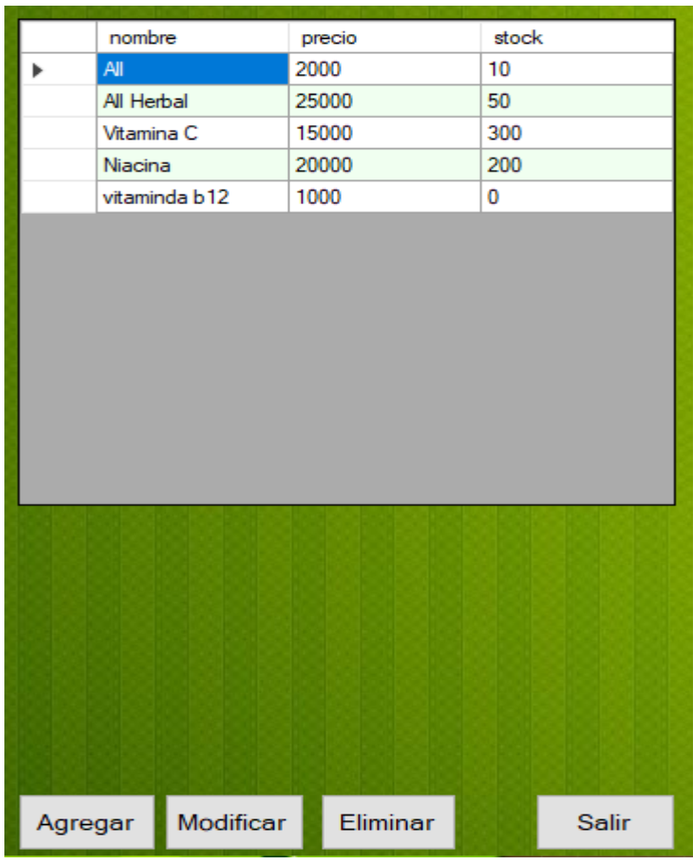

Figura 3-14: "Vista de usuario administrar productos"
Fuente: Elaboración propia.

3.4.6 Agregar/Modificar producto

Nombre: frmProductos_agregar

Objetivo: Permite al usuario agregar o modificar un producto.

Referencia código fuente: Anexo 1.6 AGREGAR/MODIFICAR PRODUCTOS

Diagrama de bloque:

Agregar:

Figura 3-15: "Diagrama de bloques agregar productos"
Fuente: Elaboración propia.

Modificar:

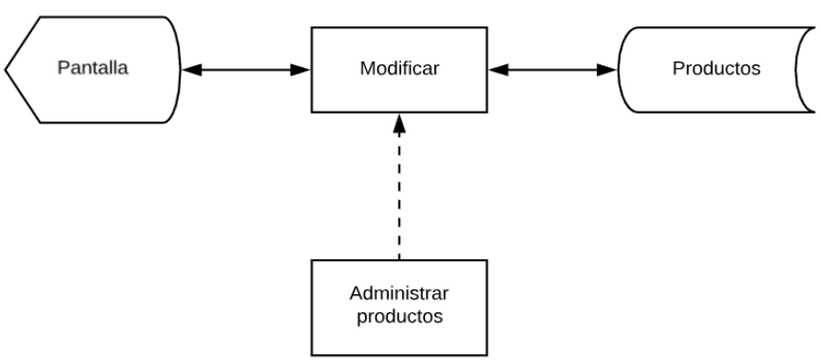

Figura 3-16: "Diagrama de bloques modificar productos"
Fuente: Elaboración propia.

*Nota: El programa frmProductos invoca el programa “frmProductos_agregar” pasando datos del producto que se desea modificar. Se indica mediante línea punteada.

Reglas del proceso:

Agregar: En caso de que este formulario sea llamado sin ningún parámetro, todos los campos quedan en blanco. Una vez estén todos los campos con datos, se presiona el botón de agregar y este inserta el registro de ese producto en la tabla base de datos.

Modificar: En caso de que este formulario sea llamado con parámetro(datos del producto), se rellenan todos los campos con esos datos, dando acceso al usuario de cambiar el nombre y el precio, en caso de que se quiera modificar el stock, se debe presionar el botón que está al lado del campo del stock, lo que llama al formulario frmStock y desde ahí se modifica; una vez cambiados los campos que se desean modificar se debe presionar el botón modificar y este modifica el registro de ese producto en la base de datos.

Vista de usuario:

Agregar:

Figura 3-17: “Vista de usuario agregar productos”
Fuente: Elaboración propia.

Modificar:

Figura 3-18: “Vista de usuario modificar productos”
Fuente: Elaboración propia.

CONCLUSIONES

Este trabajo de título fue realizado durante todo el año académico del 2018, siendo el primer semestre exclusivo para el análisis y el diseño de nuestro sistema; el segundo semestre se dio inicio a la construcción del código de todos los formularios. Al principio el lenguaje de desarrollo seleccionado fue Java SE, aunque la implementación de ciertos aspectos que se querían lograr se tornaron un poco complicados e incómodos de programar; luego en un ramo académico se tuvo la oportunidad de aprender VisualBasic.NET y la programación en base a capas, y al ver que era más sencillo y todos esos inconvenientes se agilizaban, se decidió migrar todo el sistema a dicho lenguaje de programación.

Para la implementación de este lenguaje fue necesaria documentación extra proporcionada por Internet, ya que lo aprendido en clases en un principio fue útil, pero con el tiempo se tornó insuficiente.

También hubo un cambio de motor de base de datos, migrando de MySQL a Access, esto por dos razones: el flujo de datos que se mueve en el sistema no es mayor y, además, esta base de datos es de fácil transporte para el usuario ya que es un simple archivo.

Para terminar, una de las posibles mejoras sería expandir la toma de horas a una aplicación web, asi los pacientes ya no tendrían que llamar cada vez que pidieran una hora, y la nutricionista solo tendría que ver la lista de pacientes que va a atender en el dia, lo que aliviana su trabajo y también lo agiliza.

BIBLIOGRAFÍA

Visual Studio tutorials | Visual Basic:

https://docs.microsoft.com/en-us/visualstudio/get-started/visual-basic/?view=vs-2017

Documentación de uso, instalación y programación en Microsoft Visual Studio enfocado en el lenguaje Visual Basic.

Framework de Microsoft Visual Basic .NET ver 4.7.2:

https://docs.microsoft.com/en-us/dotnet/api/system.data?view=netframework-4.7.2

Documentación sobre el framework Visual Basic .NET.

Stack over flow:

https://es.stackoverflow.com

Sitio de preguntas y respuestas para programadores y profesionales de la informática.

Social Microsoft:

https://social.msdn.microsoft.com/Forums/es-ES/home

Sitio de Microsft sobre preguntas y respuestas para programadores.

ANEXO 1: CÓDIGO FUENTE

1.1 REGISTRO DE SESIONES

```
Imports System.Windows.Forms.MonthCalendar
Imports TrabajoTitulo_v1.Paciente
Imports TrabajoTitulo_v1.frmPaciente_consultar
Imports System.Data.DataTable
Public Class frmReservaSesion
  Dim pacienteWithData As Paciente
  Dim bsnNegocios As New BsnReservaSesion
  Dim dtp As New DateTimePicker
  Dim fecha As New Date
  Public Sub New()
    InitializeComponent()
    dtpFechaSeleccionada.ShowUpDown = True
    lblHorarioActual.Text = dtpFechaHorario.Value.Date
    crearHorasDisponibles()
    updateTabla()
    dgvHoras.Columns(0).Visible = False
    dgvHoras.Columns(1).Visible = False
    dgvHoras.Columns(5).Visible = False
    dgvHoras.Columns(7).Visible = False
    dgvHoras.Columns(8).Visible = False
    dgvHoras.Columns(9).Visible = False
    dgvHoras.Columns(10).Visible = False
    dgvHoras.Columns(11).Visible = False
    'Fill combobox
    cmbEstados.DisplayMember = "Text"
    cmbEstados.ValueMember = "Value"
    Dim dtcmb As New DataTable
    dtcmb.Columns.Add("Text", GetType(String))
    dtcmb.Columns.Add("Value", GetType(String))
    dtcmb.Rows.Add("Confirmada", "Confirmada")
    dtcmb.Rows.Add("Por Confirmar", "Por Confirmar")
    'dtcmb.Rows.Add("Terminada", "Terminada")
    cmbEstados.DataSource = dtcmb
    cmbaux.DataSource =
bsnNegocios.getOnlyHoraNoDisponible(dtpFechaHorario.Value)
    cmbaux.ValueMember = "fecha_hora"
    cmbaux.DisplayMember = "fecha_hora"
  End Sub

  Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
BtnBuscarPaciente.Click
    Dim frmConsultar As New FrmPaciente_consultar(Operaciones.consultarreserva)
    frmConsultar.ShowDialog()
    pacienteWithData = frmConsultar.globalP
    Try
      TxtNombre.Text = pacienteWithData.nombre & " " & pacienteWithData.apellidos
    Catch
    End Try
  End Sub
  Public Sub updateTabla()
    dgvHoras.DataSource =
bsnNegocios.getHorasNoDisponibles(dtpFechaHorario.Value)
  End Sub
```

```
Private Sub crearHorasDisponibles()
    Dim tabla As New DataTable("Tabla")
    tabla.Clear()
    tabla.Columns.Add("Fecha", Type.GetType("System.DateTime"))
    tabla.Columns.Add("Hora", Type.GetType("System.String"))
    fecha = dtpFechaHorario.Value.Date
    fecha = fecha.AddHours(12)
    fecha = fecha.AddMinutes(30)
    dtp.Value = fecha
    Dim aux As Boolean
    cmbaux.DataSource =
bsnNegocios.getOnlyHoraNoDisponible(dtpFechaHorario.Value)
    cmbaux.ValueMember = "fecha_hora"
    cmbaux.DisplayMember = "fecha_hora"
    For i As Integer = 0 To 18
        If i > 0 Then
            dtp.Value = dtp.Value.AddMinutes(20)
        End If
        'Rella el cmb con la base de datos (sólo muestra las horas no disponibles)
        'la idea es obtener los datos y compararlos con el que se esta agregando a la lista
de disponibles
        aux = True
        For j As Integer = 0 To cmbaux.Items.Count - 1
            cmbaux.SelectedIndex = j
            If dtp.Value = cmbaux.SelectedValue Then
                aux = False
            End If
        Next
        If (aux) Then
            tabla.Rows.Add(dtp.Value, dtp.Value.ToShortTimeString)
        End If
    Next
    dgvHorasDisponibles.DataSource = tabla
    dgvHorasDisponibles.Columns(0).Visible = False
End Sub

Private Sub dgvHorasDisponibles_CellClick(sender As Object, e As
DataGridViewCellEventArgs) Handles dgvHorasDisponibles.CellClick
    Try
        dtpFechaSeleccionada.Value =
dgvHorasDisponibles.SelectedRows.Item(0).Cells(0).Value
        TxtHoraSeleccionada.Text = dtpFechaSeleccionada.Value.ToShortTimeString
    Catch
    End Try
End Sub
Private Sub ActualizarDataGrids()
    lblHorarioActual.Text = dtpFechaHorario.Value.Date
    crearHorasDisponibles()
    Dim FechasDisp As DateTime
    FechasDisp = dtpFechaHorario.Value
    dgvHoras.DataSource = bsnNegocios.getHorasNoDisponibles(FechasDisp)
End Sub
Private Sub dtpFechaHorario_ValueChanged(sender As Object, e As EventArgs)
Handles dtpFechaHorario.ValueChanged
    ActualizarDataGrids()
End Sub
```

```
Private Sub dgvHoras_CellClick(sender As Object, e As DataGridViewCellEventArgs)
Handles dgvHoras.CellClick
    Try
        'cell 1 = rut del paciente
        'cell 0 = fecha 3=nom 4=ape
        TxtNombrePacienteEstado.Text =
dgvHoras.SelectedRows.Item(0).Cells(3).Value & "
" & dgvHoras.SelectedRows.Item(0).Cells(4).Value & " " &
dgvHoras.SelectedRows.Item(0).Cells(2).Value
        Dim st As String
        st = dgvHoras.SelectedRows.Item(0).Cells(6).Value
        If st = "Por Confirmar" Then
            lblEstadoReserva.Text = "Por Confirmar"
        End If
        If st = "Terminada" Then
            lblEstadoReserva.Text = "Terminada"
        End If
        If st = "Confirmada" Then
            lblEstadoReserva.Text = "Confirmada"
        End If
    Catch
        TxtNombrePacienteEstado.Text = ""
        lblEstadoReserva.Text = ""
    End Try
End Sub
Private Sub BtnAceptarCambioEstado_Click(sender As Object, e As EventArgs)
Handles BtnAceptarCambioEstado.Click
    Dim reserva_sesion As New Reserva_sesion
    If (TxtNombrePacienteEstado.Text = "" And lblEstadoReserva.Text = "") Then
        MsgBox("Para cambiar el estado: DEBE seleccionar una fila completa",
MsgBoxStyle.Information, "Alerta")
    End If
    Try
        reserva_sesion.fecha_hora = dgvHoras.SelectedRows.Item(0).Cells(0).Value
        reserva_sesion.estado = cmbEstados.SelectedValue
        If (TxtNombrePacienteEstado.Text <> "" And lblEstadoReserva.Text <> "") Then
            bsnNegocios.updateEstadoReserva(reserva_sesion)
            MsgBox("Estado cambiando satisfactoriamente", MsgBoxStyle.Information,
"Alerta")
        End If
    Catch
    End Try
    updateTabla()
    TxtNombrePacienteEstado.Text = ""
    lblEstadoReserva.Text = ""
End Sub
Private Sub btnIngresarNuevaReserva_Click(sender As Object, e As EventArgs)
Handles btnIngresarNuevaReserva.Click
    Dim reserva_sesion As New Reserva_sesion
    'If MessageBox.Show("¿ Está segura que desea salir ?", "Alerta",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question) =
Windows.Forms.DialogResult.OK Then
    '   Application.ExitThread()
    'End If
    If (TxtNombre.Text = "") Then
```

```
        MsgBox("Se necesita buscar un paciente primero", MsgBoxStyle.Information,
"Alerta")
      Else
        If TxtHoraSeleccionada.Text = "" Then
          MsgBox("Se necesita seleccionar una hora disponible",
MsgBoxStyle.Information, "Alerta")
        Else
          reserva_sesion.fecha_hora = dtpFechaSeleccionada.Value
          reserva_sesion.rut = pacienteWithData.rut
          reserva_sesion.estado = "Por Confirmar"
          reserva_sesion.observaciones = "N/A"
          reserva_sesion.cod_dieta = vbNull 'dieta 1 means no data dieta
          bsnNegocios.setReservaSesionReservando(reserva_sesion)
          MsgBox("Hora Ingresada correctamente", MsgBoxStyle.Information, "Alerta")
          'TODO ACTUALIZAR AMBAS TABLAS
          ActualizarDataGrids()
          dtpFechaSeleccionada.Value = Today
          TxtHoraSeleccionada.Text = ""
          TxtNombre.Text = ""
        End If
      End If
    End Sub
    Private Function getDatosFromDgvHoras()
      Dim reserva_sesionWithData As New Reserva_sesion
      Try
        reserva_sesionWithData.fecha_hora =
dgvHoras.SelectedRows.Item(0).Cells(0).Value
        reserva_sesionWithData.rut = dgvHoras.SelectedRows.Item(0).Cells(1).Value
        reserva_sesionWithData.nro_sesion =
Integer.Parse(dgvHoras.SelectedRows.Item(0).Cells(5).Value)
        reserva_sesionWithData.estado =
dgvHoras.SelectedRows.Item(0).Cells(6).Value
        reserva_sesionWithData.estatura =
Decimal.Parse(dgvHoras.SelectedRows.Item(0).Cells(7).Value)
        reserva_sesionWithData.peso =
Decimal.Parse(dgvHoras.SelectedRows.Item(0).Cells(8).Value)
        reserva_sesionWithData.cintura =
Decimal.Parse(dgvHoras.SelectedRows.Item(0).Cells(9).Value)
        reserva_sesionWithData.observaciones =
dgvHoras.SelectedRows.Item(0).Cells(10).Value
        reserva_sesionWithData.cod_dieta =
Integer.Parse(dgvHoras.SelectedRows.Item(0).Cells(11).Value)
      Catch
      End Try
      Return reserva_sesionWithData
    End Function
    Private Sub btnSesiones_Click(sender As Object, e As EventArgs) Handles
btnSesiones.Click
      Try
        Dim reserva_sesion As New Reserva_sesion
        reserva_sesion = getDatosFromDgvHoras()
        Dim frmSesiones As New
frmSesiones(dgvHoras.SelectedRows.Item(0).Cells(3).Value,
dgvHoras.SelectedRows.Item(0).Cells(4).Value, reserva_sesion)
        frmSesiones.ShowDialog()
        ActualizarDataGrids()
      Catch ex As Exception
```

```
        MsgBox("Para ir a las sesiones: DEBE seleccionar una fila completa",
MsgBoxStyle.Information, "Información")
      End Try
   End Sub

   Private Sub btnCancelarNuevaReserva_Click(sender As Object, e As EventArgs)
Handles btnCancelarNuevaReserva.Click
      dtpFechaSeleccionada.Value = Today
      TxtHoraSeleccionada.Text = ""
      TxtNombre.Text = ""
   End Sub

   Private Sub BtnCancelarCambioEstado_Click(sender As Object, e As EventArgs)
Handles BtnCancelarCambioEstado.Click
      TxtNombrePacienteEstado.Text = ""
      lblEstadoReserva.Text = ""
   End Sub

   Private Sub frmReservaSesion_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
      Dim style As New DataGridViewCellStyle
      style.ForeColor = Color.Black
      dgvHorasDisponibles.DefaultCellStyle = style
      dgvHoras.DefaultCellStyle = style
   End Sub

   Private Sub btnEliminarHora_Click(sender As Object, e As EventArgs) Handles
btnEliminarHora.Click
      Dim fechaToDelete As DateTime
      Try
         fechaToDelete = dgvHoras.SelectedRows.Item(0).Cells(0).Value
         bsnNegocios.deleteReservaSesion(fechaToDelete)
         ActualizarDataGrids()
      Catch ex As Exception
         MsgBox("Error al eliminar la reserva: DEBE seleccionar una fila completa",
MsgBoxStyle.Information, "Información")
      End Try
   End Sub

   Private Sub btnSalir_Click(sender As Object, e As EventArgs) Handles btnSalir.Click
      Me.Dispose()
   End Sub
End Class
```

1.2 CONSULTAR PACIENTE

```
Public Class frmPaciente_consultar
   Dim bsnNegocios As New BsnPacientes
   Dim paciente As New Paciente
   Public consultarPaciente As New Paciente
   Dim oper As Integer
   Public globalP As New Paciente

   Public Sub New(operacion As operaciones)
      InitializeComponent()
      oper = operacion
      dgv_pacientes.DataSource = bsnNegocios.getPaciente()
      btn_modificar.Visible = False
      btn_eliminar.Visible = False
      dgv_pacientes.Columns(7).Visible = False
      'dgv_pacientes.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill
      'dgv_pacientes.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells
      'dgv_pacientes.ScrollBars = ScrollBars.Both
      'dgv_pacientes.AlternatingRowsDefaultCellStyle.BackColor = Color.Honeydew
      'dgv_pacientes.DefaultCellStyle.WrapMode = DataGridViewTriState.True
      BtnSelecionar.Visible = False
      If oper = operaciones.consultar Then
         btn_agregar.Visible = True
         BtnSelecionar.Visible = True
      End If
      If oper = operaciones.consultarreserva Then
         btn_eliminar.Visible = False
         btn_modificar.Visible = False
         BtnSelecionar.Visible = True
      End If
      If oper = operaciones.modificar Then
         btn_eliminar.Visible = False
         btn_agregar.Visible = False
         BtnSelecionar.Visible = False
      End If
   End Sub
   Private Sub ActualizarDGV()
      dgv_pacientes.DataSource = bsnNegocios.getPaciente()
   End Sub
   Private Sub btn_eliminar_Click(sender As Object, e As EventArgs) Handles
btn_eliminar.Click
      If MessageBox.Show("¿ Está seguro de eliminar el paciente?", "Alerta",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question) =
Windows.Forms.DialogResult.OK Then
         Try
            bsnNegocios.deletePaciente(paciente)
         Catch ex As Exception
         End Try
         ActualizarDGV()
         rtxt_enfermedades.Text = ""
         txtFiltroEmail.Text = ""
         txtFiltroNombre.Text = ""
         txtFiltroRut.Text = ""
      End If
   End Sub
```

```
Private Sub btn_agregar_Click(sender As Object, e As EventArgs) Handles btn_agregar.Click
    Dim pacienteNoData As New Paciente
    Dim ingresar As New FrmPaciente_ingresar(operaciones.ingresar, pacienteNoData)
    ingresar.ShowDialog()
    ActualizarDGV()
    txtFiltroEmail.Text = ""
    txtFiltroNombre.Text = ""
    txtFiltroRut.Text = ""
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Me.Dispose()
End Sub
Public Function getDatosFromDgv() As Paciente
    Dim p As New Paciente
    Try
        p.rut = dgv_pacientes.SelectedRows.Item(0).Cells(0).Value
        p.nombre = dgv_pacientes.SelectedRows.Item(0).Cells(1).Value
        p.apellidos = dgv_pacientes.SelectedRows.Item(0).Cells(2).Value
        p.fechaNacimiento = dgv_pacientes.SelectedRows.Item(0).Cells(3).Value
        p.direccion = dgv_pacientes.SelectedRows.Item(0).Cells(4).Value
        p.celular = Integer.Parse(dgv_pacientes.SelectedRows.Item(0).Cells(5).Value)
        p.email = dgv_pacientes.SelectedRows.Item(0).Cells(6).Value
        p.enfermedades = dgv_pacientes.SelectedRows.Item(0).Cells(7).Value
    Catch
    End Try
    Return p
End Function
Private Sub BtnSelecionar_Click(sender As Object, e As EventArgs) Handles BtnSelecionar.Click
    Dim paciente2 As New Paciente
    paciente2 = getDatosFromDgv()
    globalP = paciente2
    Me.Dispose()
End Sub
Private Sub btn_modificar_Click(sender As Object, e As EventArgs) Handles btn_modificar.Click
    Dim paciente As New Paciente
    paciente = getDatosFromDgv()
    Dim formIngresarPaciente As New FrmPaciente_ingresar(operaciones.modificar, paciente)
    formIngresarPaciente.ShowDialog()
    rtxt_enfermedades.Text = ""
    ActualizarDGV()
    txtFiltroEmail.Text = ""
    txtFiltroNombre.Text = ""
    txtFiltroRut.Text = ""
End Sub
Private Sub txtFiltroNombre_TextChanged(sender As Object, e As EventArgs) Handles txtFiltroNombre.TextChanged
    Dim k As New DataView(bsnNegocios.getPaciente)
    k.RowFilter = String.Format("NOMBRE LIKE '%{0}%'", txtFiltroNombre.Text)
    dgv_pacientes.DataSource = k
End Sub
Private Sub txtFiltroRut_TextChanged(sender As Object, e As EventArgs) Handles txtFiltroRut.TextChanged
    Dim k As New DataView(bsnNegocios.getPaciente)
```

```
        k.RowFilter = String.Format("RUT LIKE '%{0}%'", txtFiltroRut.Text)
        dgv_pacientes.DataSource = k
    End Sub
    Private Sub txtFiltroEmail_TextChanged(sender As Object, e As EventArgs) Handles
txtFiltroEmail.TextChanged
        Dim k As New DataView(bsnNegocios.getPaciente)
        k.RowFilter = String.Format("EMAIL LIKE '%{0}%'", txtFiltroEmail.Text)
        dgv_pacientes.DataSource = k
    End Sub
    Private Sub dgv_pacientes_CellClick(sender As Object, e As
DataGridViewCellEventArgs) Handles dgv_pacientes.CellClick
        Try
            rtxt_enfermedades.Text = dgv_pacientes.SelectedRows.Item(0).Cells(7).Value
            If rtxt_enfermedades.Text <>
dgv_pacientes.SelectedRows.Item(0).Cells(7).Value Then
                rtxt_enfermedades.Text = ""
            End If
            If oper = operaciones.consultar Then
                consultarPaciente.rut = dgv_pacientes.SelectedRows.Item(0).Cells(0).Value
                consultarPaciente.nombre =
dgv_pacientes.SelectedRows.Item(0).Cells(1).Value
                consultarPaciente.apellidos =
dgv_pacientes.SelectedRows.Item(0).Cells(2).Value
            Else
                paciente.rut = dgv_pacientes.SelectedRows.Item(0).Cells(0).Value
                btn_modificar.Visible = True
                If oper = operaciones.modificar Then
                Else
                    btn_eliminar.Visible = True
                End If
            End If
        Catch
            rtxt_enfermedades.Text = ""
            btn_modificar.Visible = False
            btn_eliminar.Visible = False
        End Try
    End Sub
End Class
```

1.3 INFORME DE VENTAS

```
Public Class frmReporteVentas
   Dim bsnNegocios As New BsnVentas
   Public Sub New()
      InitializeComponent()
   End Sub
   Private Sub btnGenerarReporte_Click(sender As Object, e As EventArgs) Handles
btnGenerarReporte.Click
      Dim valor As Integer
      Dim valor2 As Integer
      Dim valor3 As Integer
      Dim aux As String
      'se agrega + 1 dia para solucionar como se recorre
      dgvReporteVentas.DataSource =
bsnNegocios.getVentasDetalleventas(dtpDesde.Value.Date,
dtpHasta.Value.Date.AddDays(1))
      'txtCostos.Text = bsnNegocios.getCostosByRangeDate(dtpDesde.Value.Date,
dtpHasta.Value.Date.AddDays(1)).ToString
      valor2 = bsnNegocios.getCostosByRangeDate(dtpDesde.Value.Date,
dtpHasta.Value.Date.AddDays(1))
      aux = valor2.ToString("n0")
      txtCostos.Text = aux
      'txtIngresos.Text = bsnNegocios.getIngresosByRangeDate(dtpDesde.Value.Date,
dtpHasta.Value.Date.AddDays(1)).ToString
      valor3 = bsnNegocios.getIngresosByRangeDate(dtpDesde.Value.Date,
dtpHasta.Value.Date.AddDays(1))
      aux = valor3.ToString("n0")
      txtIngresos.Text = aux
      valor = valor3 - valor2
      Dim gananciaPerdida As String = valor.ToString("n0")
      txtGananciaPerdida.Text = gananciaPerdida
      If valor > 0 Then
         txtGananciaPerdida.BackColor = Color.LightGreen
      Else
         txtGananciaPerdida.BackColor = Color.White
         txtGananciaPerdida.ForeColor = Color.Red
      End If
   End Sub

   Private Sub frmReporteVentas_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
      Dim style As New DataGridViewCellStyle
      style.ForeColor = Color.Black
      dgvReporteVentas.DefaultCellStyle = style
   End Sub
   Private Sub btnSalir_Click(sender As Object, e As EventArgs) Handles btnSalir.Click
      Me.Dispose()
   End Sub
End Class
```

1.4 DIETA ALIMENTOS

```
Imports System.Drawing.Imaging
Imports System.IO
Imports System.Text.RegularExpressions
Public Class frmDieta_principal
   Dim bsnDieta As New BsnDietas
   Dim detalleDieta As New detalleDieta
   Dim data As New DataTable
   Dim Lista As New List(Of detalleDieta)
   Dim oper As Integer
   Public Sub New(dieta As Dieta, operacion As operaciones)
      oper = operacion
      InitializeComponent()
      txtNombre.Text = dieta.nombre
      rtxtDescripcion.Text = dieta.descripcion
      txtNombre.ReadOnly = True
      rtxtDescripcion.ReadOnly = True
      detalleDieta.cod_dieta = dieta.cod_dieta
      btnCapturar.Visible = False
      txtImagen.Visible = False
      Button1.Visible = False
      picCaptura.Visible = False
      If oper = operaciones.ingresar Then
         data = bsnDieta.getNombreComida
         For fil As Integer = 1 To Layout.RowCount - 1
            Dim cmb As New ComboBox
            Dim Panel As New Panel
            Dim txt As New TextBox
            cmb.Name = "horac" & fil
            txt.Name = "horat" & fil
            cmb.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
            cmb.Text = "Ingrese Comida"
            For Each row As DataRow In data.Rows
               cmb.Items.Add(row("nombre_comida"))
            Next
            cmb.Size = New System.Drawing.Size(115, 25)
            cmb.Location = New System.Drawing.Point(17, 22)
            cmb.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
            txt.Size = New System.Drawing.Size(70, 24)
            txt.Location = New System.Drawing.Point(36, 50)
            txt.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
            AddHandler txt.KeyPress, AddressOf numeros
            Panel.Name = "Panel" & fil
            Panel.Controls.Add(cmb)
            Panel.Controls.Add(txt)
            Layout.Controls.Add(Panel, 0, fil)
         Next
         For fila As Integer = 1 To Layout.RowCount - 1
            For col As Integer = 1 To Layout.ColumnCount - 1
               Dim btn As New Button
               btn.Name = "btn" & fila & col
               btn.Text = "Agregar Alimentos"
```

```
                btn.UseVisualStyleBackColor = True
                btn.BackColor = System.Drawing.Color.Honeydew
                btn.Anchor = AnchorStyles.None
                btn.Location = New System.Drawing.Point(54, 40)
                btn.Size = New System.Drawing.Size(96, 43)
                btn.Font = New System.Drawing.Font("Microsoft Sans Serif", 10.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
                Layout.Controls.Add(btn, col, fila)
                AddHandler btn.Click, AddressOf focusRtxt
            Next
        Next
    End If
    If oper = operaciones.modificar Then
        Dim data As New DataTable
        Dim comidas As New DataTable
        Dim comida As String
        Dim detalle As DataTable
        Dim contt As Integer = 0
        comidas = bsnDieta.getNombreComida()
        data = bsnDieta.getHoras(detalleDieta.cod_dieta)

        Dim cont As Integer = 1
        For Each row As DataRow In data.Rows
            Dim cmb As New ComboBox
            Dim Panel As New Panel
            Dim txt As New TextBox
            txt.Size = New System.Drawing.Size(70, 24)
            txt.Location = New System.Drawing.Point(36, 50)
            txt.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
            If row("hora").ToString.Substring(12, 1) = ":" Then
                txt.Text = "      " & row("hora").ToString.Substring(11, 4)
            Else
                txt.Text = "     " & row("hora").ToString.Substring(11, 5)
            End If
            cmb.Name = "horac" & cont
            txt.Name = "horat" & cont
            cmb.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
            For Each roww As DataRow In comidas.Rows
                cmb.Items.Add(roww("nombre_comida"))
            Next
            cmb.Text = row("nombre_comida").ToString
            cmb.Size = New System.Drawing.Size(115, 25)
            cmb.Location = New System.Drawing.Point(17, 22)
            cmb.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
            AddHandler txt.KeyPress, AddressOf numeros
            Panel.Name = "Panel" & cont
            Panel.Controls.Add(cmb)
            Panel.Controls.Add(txt)
            Layout.Controls.Add(Panel, 0, cont)
            cont = cont + 1
        Next
        For a = cont To 6
```

```
            Dim cmb As New ComboBox
            Dim Panel As New Panel
            Dim txt As New TextBox
            cmb.Name = "horac" & a
            txt.Name = "horat" & a
            cmb.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
            cmb.Text = "Ingrese Comida"
            For Each roww As DataRow In comidas.Rows
               cmb.Items.Add(roww("nombre_comida"))
            Next
            cmb.Size = New System.Drawing.Size(115, 25)
            cmb.Location = New System.Drawing.Point(17, 22)
            cmb.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
            txt.Size = New System.Drawing.Size(70, 24)
            txt.Location = New System.Drawing.Point(36, 50)
            txt.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
            AddHandler txt.KeyPress, AddressOf numeros
            Panel.Name = "Panel" & a
            Panel.Controls.Add(cmb)
            Panel.Controls.Add(txt)
            Layout.Controls.Add(Panel, 0, a)
         Next
         Dim aux As String
         detalle = bsnDieta.getDetalleDieta(detalleDieta.cod_dieta)
         Dim Ctl As Control
         Dim horax As String
         For fila As Integer = 1 To Layout.RowCount - 1
            For col As Integer = 1 To Layout.ColumnCount - 1
               Dim panell As New Panel
               panell = Layout.Controls.Container.GetControlFromPosition(0, fila)
               For xx = 0 To panell.Controls.Count - 1
                  Ctl = panell.Controls.Item(xx)
                  If TypeOf (Ctl) Is TextBox Then
                     Dim Control As TextBox = Ctl
                     aux = (Control.Text.Trim)
                  End If
               Next
               For Each fil As DataRow In detalle.Rows
                  If fil("hora").ToString.Substring(12, 1) = ":" Then
                     horax = fil("hora").ToString.Substring(11, 4)
                  Else
                     horax = fil("hora").ToString.Substring(11, 5)
                  End If
                  If Layout.Controls.Container.GetControlFromPosition(col, 0).Text =
fil("dia") And horax = aux Then
                     Dim rtxt As New RichTextBox
                     rtxt.Text = fil("alimentos")
                     rtxt.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
                     rtxt.Size = New System.Drawing.Size(241, 102)
                     Layout.Controls.Add(rtxt, col, fila)
                  End If
```

```
            Next

        Next
        contt = contt + 1
        If (cont - 1) = contt Then
            Exit For
        End If
    Next
    For fila As Integer = 1 To Layout.RowCount - 1
        For col As Integer = 1 To Layout.ColumnCount - 1
            If TypeOf Layout.Controls.Container.GetControlFromPosition(col, fila) Is
RichTextBox Then
            Else
                Dim btn As New Button
                btn.Name = "btn" & fila & col
                btn.Text = "Agregar Alimentos"
                btn.UseVisualStyleBackColor = True
                btn.BackColor = System.Drawing.Color.Honeydew
                btn.Anchor = AnchorStyles.None
                btn.Location = New System.Drawing.Point(54, 40)
                btn.Size = New System.Drawing.Size(96, 43)
                btn.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
                Layout.Controls.Add(btn, col, fila)
                AddHandler btn.Click, AddressOf focusRtxt
            End If
        Next
    Next
End If
If oper = operaciones.consultar Then
    txtImagen.Text =
"C:\Users\Emanuel\source\repos\TrabajoTitulo_v1\TrabajoTitulo_v1\Dieta\Dietas\" &
txtNombre.Text & ".png"
    'btnCapturar.Visible = True
    btnGuardar.Enabled = False
    Button1.Visible = False
    Dim data As New DataTable
    Dim comidas As New DataTable
    Dim detalle As DataTable
    Dim contt As Integer = 0
    comidas = bsnDieta.getNombreComida()
    data = bsnDieta.getHoras(detalleDieta.cod_dieta)
    Dim cont As Integer = 1
    For Each row As DataRow In data.Rows
        Dim lbl As New Label
        Dim Panel As New Panel
        Dim txt As New TextBox
        txt.Size = New System.Drawing.Size(70, 24)
        txt.Location = New System.Drawing.Point(36, 50)
        txt.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
        If row("hora").ToString.Substring(12, 1) = ":" Then
            txt.Text = "      " & row("hora").ToString.Substring(11, 4)
        Else
            txt.Text = "     " & row("hora").ToString.Substring(11, 5)
        End If
```

```
            lbl.Name = "horac" & cont
            txt.Name = "horat" & cont
            lbl.Size = New System.Drawing.Size(115, 25)
            lbl.Location = New System.Drawing.Point(17, 22)
            lbl.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
            lbl.Text = "   " & row("nombre_comida").ToString
            AddHandler txt.KeyPress, AddressOf numeros
            Panel.Name = "Panel" & cont
            txt.ReadOnly = True
            Panel.Controls.Add(lbl)
            Panel.Controls.Add(txt)
            Layout.Controls.Add(Panel, 0, cont)
            cont = cont + 1
        Next
        For a = cont To 6
            Dim lbl As New Label
            Dim Panel As New Panel
            Dim txt As New TextBox
            lbl.Name = "horac" & a
            txt.Name = "horat" & a
            lbl.Text = "No hay comida"
            txt.Text = "     00:00"
            lbl.Size = New System.Drawing.Size(115, 25)
            lbl.Location = New System.Drawing.Point(17, 22)
            lbl.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
            txt.Size = New System.Drawing.Size(70, 24)
            txt.Location = New System.Drawing.Point(36, 50)
            txt.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
            AddHandler txt.KeyPress, AddressOf numeros
            Panel.Name = "Panel" & a
            txt.ReadOnly = True
            Panel.Controls.Add(lbl)
            Panel.Controls.Add(txt)
            Layout.Controls.Add(Panel, 0, a)
        Next
        Dim aux As String
        detalle = bsnDieta.getDetalleDieta(detalleDieta.cod_dieta)
        Dim Ctl As Control
        Dim horax As String
        For fila As Integer = 1 To Layout.RowCount - 1
            For col As Integer = 1 To Layout.ColumnCount - 1
                Dim panell As New Panel
                panell = Layout.Controls.Container.GetControlFromPosition(0, fila)
                For xx = 0 To panell.Controls.Count - 1
                    Ctl = panell.Controls.Item(xx)
                    If TypeOf (Ctl) Is TextBox Then
                        Dim Control As TextBox = Ctl
                        aux = (Control.Text.Trim)
                    End If
                Next
                For Each fil As DataRow In detalle.Rows
                    If fil("hora").ToString.Substring(12, 1) = ":" Then
```

```
                    horax = fil("hora").ToString.Substring(11, 4)
                Else
                    horax = fil("hora").ToString.Substring(11, 5)
                End If
                If Layout.Controls.Container.GetControlFromPosition(col, 0).Text =
fil("dia") And horax = aux Then
                    Dim rtxt As New RichTextBox
                    rtxt.Text = fil("alimentos")
                    rtxt.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
                    rtxt.Size = New System.Drawing.Size(241, 102)
                    rtxt.BackColor = System.Drawing.Color.White
                    Layout.Controls.Add(rtxt, col, fila)
                    rtxt.ReadOnly = True
                End If
            Next
        Next
        contt = contt + 1
        If (cont - 1) = contt Then
            Exit For
        End If
    Next
  End If
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles btnSalir.Click
  Me.Dispose()
End Sub
Private Sub numeros(sender As Object, e As KeyPressEventArgs)
  Dim myTxt As TextBox = sender
  Dim re As New Regex("[^0-9pma:\b]", RegexOptions.IgnoreCase)
  If (e.KeyChar.ToString = ":") And (myTxt.Text.Contains(e.KeyChar.ToString))
Then
    e.Handled = True
    Exit Sub
  End If
  e.Handled = re.IsMatch(e.KeyChar)
End Sub
Private Sub focusRtxt(ByVal sender As Object, ByVal e As System.EventArgs)
  Dim myBtn As Button = sender
  Dim panel As New Panel
  Dim y, x As Integer
  Dim rtxt As New RichTextBox
  y = Integer.Parse(myBtn.Name.Substring(3, 1))
  x = Integer.Parse(myBtn.Name.Substring(4))
  panel = Layout.Controls.Container.GetControlFromPosition(0, y)
  Dim Ctl As Control
  Dim Txt As Control
  Dim xx As Integer
  Dim sw1, sw2 As Integer
```

```
  For xx = 0 To panel.Controls.Count - 1
```

```
        Ctl = panel.Controls.Item(xx)
        If TypeOf (Ctl) Is TextBox Then
            Dim Control As TextBox = Ctl
            If Control.Text.Trim <> "" Then
                If Control.Text.Trim.Substring(1, 1) = ":" Then
                    sw2 = 0
                Else
                    If Integer.Parse(Control.Text.Trim.Substring(0, 2)) < 24 Then
                        sw2 = 0
                    Else
                        MsgBox("Un día no tiene mas de 24 horas ", MsgBoxStyle.Information,
"Error")
                        Control.Text = ""
                        sw2 = 1
                    End If
                End If
            Else
                MsgBox("Debe ingresar un hora ", MsgBoxStyle.Information, "Error")
                sw1 = 1
            End If
        End If
        If TypeOf (Ctl) Is ComboBox Then
            Dim Control As ComboBox = Ctl
            If Control.Text <> "" Then
                sw2 = 0
            Else
                MsgBox("Debe ingresar un tipo de comida ", MsgBoxStyle.Information,
"Error")
                sw2 = 1
            End If
        End If
    Next
    If sw1 = 0 And sw2 = 0 Then
        Layout.Controls.Remove(myBtn)
        'rtxt.Text = "Fila " & y & "Columna" & x
        rtxt.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
        rtxt.Size = New System.Drawing.Size(241, 102)
        Layout.Controls.Add(rtxt, x, y)
    End If
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Layout.RowCount = Layout.RowCount + 1
    Layout.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent,
16.54216!))
    Dim cmb As New ComboBox
    Dim Panel As New Panel
    Dim txt As New TextBox
    cmb.Name = "horac" & Layout.RowCount - 1
    txt.Name = "horat" & Layout.RowCount - 1
    cmb.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList
    For Each row As DataRow In data.Rows
        cmb.Items.Add(row("nombre_comida"))
    Next
    cmb.Size = New System.Drawing.Size(115, 25)
```

```
        cmb.Location = New System.Drawing.Point(17, 12)
        cmb.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
        txt.Size = New System.Drawing.Size(70, 24)
        txt.Location = New System.Drawing.Point(36, 40)
        txt.Font = New System.Drawing.Font("Microsoft Sans Serif", 9.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
        AddHandler txt.KeyPress, AddressOf numeros
        Panel.Name = "Panel" & Layout.RowCount - 1
        Panel.Controls.Add(cmb)
        Panel.Controls.Add(txt)
        Layout.Controls.Add(Panel, 0, Layout.RowCount - 1)
        For col As Integer = 1 To Layout.ColumnCount - 1
            Dim btn As New Button
            btn.Name = "btn" & Layout.RowCount - 1 & col
            btn.Text = "Agregar Alimentos"
            btn.UseVisualStyleBackColor = True
            btn.BackColor = System.Drawing.Color.Honeydew
            btn.Anchor = AnchorStyles.None
            btn.Location = New System.Drawing.Point(54, 30)
            btn.Size = New System.Drawing.Size(96, 43)
            btn.Font = New System.Drawing.Font("Microsoft Sans Serif", 10.2!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
            Layout.Controls.Add(btn, col, Layout.RowCount - 1)
            AddHandler btn.Click, AddressOf focusRtxt

        Next
    End Sub
    Private Sub btnGuardar_Click(sender As Object, e As EventArgs) Handles
btnGuardar.Click
        Dim Ctl As Control
        Dim Panel As Panel
        If oper = operaciones.ingresar Then
            For fila As Integer = 1 To Layout.RowCount - 1
                For col As Integer = 1 To Layout.ColumnCount - 1
                    If TypeOf (Layout.Controls.Container.GetControlFromPosition(col, fila)) Is
RichTextBox Then
                        detalleDieta.alimentos =
Layout.Controls.Container.GetControlFromPosition(col, fila).Text
                        If detalleDieta.alimentos <> "" Then
                            Panel = Layout.Controls.Container.GetControlFromPosition(0, fila)
                            detalleDieta.dia =
Layout.Controls.Container.GetControlFromPosition(col, 0).Text
                            For i = 0 To Panel.Controls.Count - 1
                                Ctl = Panel.Controls.Item(i)
                                If TypeOf (Ctl) Is TextBox Then
                                    Dim Control As TextBox = Ctl
                                    detalleDieta.hora = Control.Text.Trim
                                End If
                                If TypeOf (Ctl) Is ComboBox Then
                                    Dim Control As ComboBox = Ctl
                                    detalleDieta.nom_comida = Control.Text
                                    detalleDieta.cod_comida =
bsnDieta.getTipoComida(detalleDieta.nom_comida)
                                End If
```

```
                Next
                bsnDieta.setDetalleDieta(detalleDieta)
            End If
        End If
    Next
  Next
  MsgBox("Se agregó correctamente ", MsgBoxStyle.Information, "Felicitaciones")
  Me.Dispose()
End If
If oper = operaciones.modificar Then
  Dim detalle As DataTable
  Dim horax, aux As String
  Dim sw As Integer = 0
  For fila As Integer = 1 To Layout.RowCount - 1
    For col As Integer = 1 To Layout.ColumnCount - 1
      If TypeOf (Layout.Controls.Container.GetControlFromPosition(col, fila)) Is
RichTextBox Then
        detalleDieta.alimentos                                                =
Layout.Controls.Container.GetControlFromPosition(col, fila).Text
        If detalleDieta.alimentos <> "" Then
          Panel = Layout.Controls.Container.GetControlFromPosition(0, fila)
          detalleDieta.dia                                                    =
Layout.Controls.Container.GetControlFromPosition(col, 0).Text
          For i = 0 To Panel.Controls.Count - 1
            Ctl = Panel.Controls.Item(i)
            If TypeOf (Ctl) Is TextBox Then
              Dim Control As TextBox = Ctl
              detalleDieta.hora = Control.Text.Trim
              aux = (Control.Text.Trim)
            End If
            If TypeOf (Ctl) Is ComboBox Then
              Dim Control As ComboBox = Ctl
              detalleDieta.nom_comida = Control.Text
              detalleDieta.cod_comida                                         =
bsnDieta.getTipoComida(detalleDieta.nom_comida)
            End If
          Next
          sw = 0
          detalle = bsnDieta.getDetalleDieta(detalleDieta.cod_dieta)
          For Each fil As DataRow In detalle.Rows
            If fil("hora").ToString.Substring(12, 1) = ":" Then
              horax = fil("hora").ToString.Substring(11, 4)
            Else
              horax = fil("hora").ToString.Substring(11, 5)
            End If
            If Layout.Controls.Container.GetControlFromPosition(col, 0).Text =
fil("dia") And horax = aux Then
              bsnDieta.updateDetalleDieta(detalleDieta, horax)
              sw = 1
            End If
          Next
          If sw = 0 Then
            bsnDieta.setDetalleDieta(detalleDieta)
          End If
        End If
      End If
    Next
```

```
        Next
        MsgBox("Se modificó correctamente ", MsgBoxStyle.Information,
"Felicittaciones")
        Me.Dispose()
      End If
    End Sub

    Private Sub btnCapturar_Click(sender As Object, e As EventArgs) Handles
btnCapturar.Click
      Dim gr As Graphics = Me.CreateGraphics
      ' Tamaño de lo que queremos copiar
      Dim fSize As Size = Me.Layout.Size
      ' Creamos el bitmap con el área que vamos a capturar
      ' En este caso, con el tamaño del formulario actual
      Dim bm As New Bitmap(fSize.Width, fSize.Height, gr)
      ' Un objeto Graphics a partir del bitmap
      Dim gr2 As Graphics = Graphics.FromImage(bm)
      ' Copiar el área de la pantalla que ocupa el formulario
      gr2.CopyFromScreen(300, 400, 0, 0, fSize)
      Me.picCaptura.Image = bm
      Me.picCaptura.Visible = True
      If String.IsNullOrEmpty(Me.txtImagen.Text) Then
        MessageBox.Show("Debes indicar el nombre del fichero",
                "Guardar",
                MessageBoxButtons.OK,
                MessageBoxIcon.Exclamation)
        Exit Sub
      End If
      Dim ext As String = Path.GetExtension(Me.txtImagen.Text).ToLower()
      Select Case ext
        Case ".png"
          Me.picCaptura.Image.Save(Me.txtImagen.Text, ImageFormat.Png)
          MsgBox("Captura guardada", MsgBoxStyle.Information, "¡Bien!")
        Case Else
          Me.picCaptura.Image.Save(Me.txtImagen.Text)
          MsgBox("Captura guardada", MsgBoxStyle.Information, "¡Bien!")
      End Select
    End Sub
End Class
```

1.5 ADMINISTRAR PRODUCTOS

```
Imports System.Text.RegularExpressions
Public Class frmProductos
    Dim bsnProducto As New BsnProducto
    Dim oper As Integer
    Public detalle As New detalleVenta
    Public nombre As String
    Dim producto As New Producto
    Public Sub New(operacion As operaciones, tabla As DataTable)
        oper = operacion
        InitializeComponent()
        If oper = operaciones.consultarprod Then
            dgvProductos.DataSource = tabla
        Else
            dgvProductos.DataSource = bsnProducto.getProductos()
        End If
        dgvProductos.Columns(0).Visible = False
        dgvProductos.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill
        dgvProductos.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells
        dgvProductos.ScrollBars = ScrollBars.Both
        dgvProductos.AlternatingRowsDefaultCellStyle.BackColor = Color.Honeydew
        dgvProductos.DefaultCellStyle.WrapMode = DataGridViewTriState.True
        txtPrecio_venta.Visible = False
        lbl.Visible = False
        txtCantidad.Visible = False
        lblCantidad.Visible = False
        btnAgregarProducto.Visible = False
        txtNombre.Visible = False
        txtPrecio.Visible = False
        txtStock.Visible = False
        lblNombre.Visible = False
        LblPrecio.Visible = False
        lblStock.Visible = False
        If oper = operaciones.consultarprod Then
            btnAgregar.Visible = False
            btnEliminar.Visible = False
            btnModificar.Visible = False
            txtNombre.Enabled = False
            txtPrecio.Enabled = False
            txtStock.Enabled = False
            txtPrecio_venta.Visible = True
            lbl.Visible = True
            txtCantidad.Visible = True
            lblCantidad.Visible = True
            txtNombre.Visible = True
            txtPrecio.Visible = True
            txtStock.Visible = True
            lblNombre.Visible = True
            LblPrecio.Visible = True
            lblStock.Visible = True

        End If
    End Sub
```

```
Private Sub ActualizarDGV()
    dgvProductos.DataSource = bsnProducto.getProductos()
End Sub
Private Sub btnAgregar_Click(sender As Object, e As EventArgs) Handles
btnAgregar.Click
    Dim product As New Producto
    Dim agregar As New frmProductos_agregar(operaciones.ingresar, product)
    agregar.ShowDialog()
    Me.ActualizarDGV()
End Sub
Private Sub btnModificar_Click(sender As Object, e As EventArgs) Handles
btnModificar.Click
    If txtNombre.Text <> "" And txtPrecio.Text <> "" And txtStock.Text <> "" Then
        producto.nombre = txtNombre.Text
        producto.precio = txtPrecio.Text
        producto.stock = txtStock.Text
        Dim productom As New frmProductos_agregar(operaciones.modificar, producto)
        productom.ShowDialog()
    Else
        MsgBox("Seleccione un producto", MsgBoxStyle.Information, "Error")
    End If
End Sub
Private Sub btnELiminar_Click(sender As Object, e As EventArgs) Handles
btnEliminar.Click
    If txtNombre.Text <> "" And txtPrecio.Text <> "" And txtStock.Text <> "" Then
        Dim productod As New Producto
        productod.cod_producto                                                =
Integer.Parse(dgvProductos.SelectedRows.Item(0).Cells(0).Value)

        If MessageBox.Show("¿ Está seguro de eliminar el producto?", "Alerta",
MessageBoxButtons.OKCancel,              MessageBoxIcon.Question)              =
Windows.Forms.DialogResult.OK Then
            Try
                bsnProducto.deleteProducto(productod)
                MsgBox("El producto se elimino satisfactoriamente",
MsgBoxStyle.Information, "¡Bien!")
                txtStock.Text = ""
                txtNombre.Text = ""
                txtPrecio.Text = ""
                Me.ActualizarDGV()
            Catch ex As Exception
                MessageBox.Show(ex.Message)
            End Try
        End If
    Else
        MsgBox("Seleccione un producto", MsgBoxStyle.Information, "Error")
    End If
End Sub

Private Sub dgvProductos_Click(sender As Object, e As EventArgs) Handles
dgvProductos.Click
    Try
        If oper = operaciones.consultar Then

            txtNombre.Text = dgvProductos.SelectedRows.Item(0).Cells(1).Value
            txtPrecio.Text = dgvProductos.SelectedRows.Item(0).Cells(2).Value
            txtStock.Text = dgvProductos.SelectedRows.Item(0).Cells(3).Value
```

```
            producto.cod_producto                                                    =
Integer.Parse(dgvProductos.SelectedRows.Item(0).Cells(0).Value)

            btnAgregar.Visible = True
            btnEliminar.Visible = True
            btnModificar.Visible = True
            txtStock.Enabled = False
        End If
        If oper = operaciones.consultarprod Then
            txtNombre.Text = dgvProductos.SelectedRows.Item(0).Cells(1).Value
            txtPrecio.Text = dgvProductos.SelectedRows.Item(0).Cells(2).Value
            txtStock.Text = dgvProductos.SelectedRows.Item(0).Cells(3).Value
            detalle.cod_producto                                                     =
Integer.Parse(dgvProductos.SelectedRows.Item(0).Cells(0).Value)
            nombre = dgvProductos.SelectedRows.Item(0).Cells(1).Value
            btnAgregarProducto.Visible = True
            txtPrecio_venta.Text = dgvProductos.SelectedRows.Item(0).Cells(2).Value
        End If
    Catch
    End Try
  End Sub

  Private Sub Button1_Click(sender As Object, e As EventArgs)
    btnAgregar.Visible = True
    btnEliminar.Visible = False
    btnModificar.Visible = False
    txtStock.Text = ""
    txtNombre.Text = ""
    txtPrecio.Text = ""
    txtStock.Enabled = True
  End Sub

  Private Sub txtPrecio_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
txtPrecio.KeyPress
    Dim re As New Regex("[^0-9\b]", RegexOptions.IgnoreCase)
    e.Handled = re.IsMatch(e.KeyChar)
  End Sub
  Private Sub txtStock_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
txtStock.KeyPress
    Dim re As New Regex("[^0-9\b]", RegexOptions.IgnoreCase)
    e.Handled = re.IsMatch(e.KeyChar)
  End Sub

  Private Sub btnSalir_Click(sender As Object, e As EventArgs) Handles btnSalir.Click

    Me.Dispose()

  End Sub

  Private Sub txtPrecio_venta_KeyPress(sender As Object, e As KeyPressEventArgs)
Handles txtPrecio_venta.KeyPress
    Dim re As New Regex("[^0-9\b]", RegexOptions.IgnoreCase)
    e.Handled = re.IsMatch(e.KeyChar)
  End Sub
  Private Sub txtCantidad_KeyPress(sender As Object, e As KeyPressEventArgs)
Handles txtCantidad.KeyPress
    Dim re As New Regex("[^0-9\b]", RegexOptions.IgnoreCase)
    e.Handled = re.IsMatch(e.KeyChar)
```

```
    End Sub

    Private Sub btnAgregarProducto_Click(sender As Object, e As EventArgs) Handles
btnAgregarProducto.Click
        If txtPrecio_venta.Text <> "" And txtCantidad.Text <> "" Then
            If Integer.Parse(txtStock.Text) > Integer.Parse(txtCantidad.Text) Then
                detalle.cantidad = Integer.Parse(txtCantidad.Text)
                detalle.precio_venta = Integer.Parse(txtPrecio_venta.Text)

                Me.Dispose()

            Else
                MsgBox("la cantidad de productos no puede ser mayor al stock actual",
MsgBoxStyle.Information, "Error")
            End If
        Else
            MsgBox("Debe ingresar todos los datos", MsgBoxStyle.Information, "Error")
        End If
    End Sub

    Private Sub frmProductos_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

    End Sub


End Class
```

1.6 AGREGAR/MODIFICAR PRODUCTOS

```
Imports System.Text.RegularExpressions
Public Class frmProductos_agregar
    Dim bsnProducto As New BsnProducto
    Dim oper As Integer
    Dim product As New Producto
    Public Sub New(operacion As operaciones, producto As Producto)
        InitializeComponent()
        oper = operacion
        If oper = operaciones.ingresar Then
            btnStock.Visible = False
        Else
            btnAgregar.Text = "Modificar"
            btnStock.Visible = True
            txtStock.ReadOnly = True
            txtNombre.Text = producto.nombre
            txtPrecio.Text = producto.precio
            txtStock.Text = producto.stock
            product.cod_producto = producto.cod_producto
        End If
    End Sub
    Private Sub btnAgregar_Click(sender As Object, e As EventArgs) Handles
btnAgregar.Click
        If oper = operaciones.ingresar Then
            If txtStock.Text <> "" And txtNombre.Text <> "" And txtPrecio.Text <> "" Then
                Dim productoa As New Producto
                productoa.nombre = txtNombre.Text
                productoa.precio = Integer.Parse(txtPrecio.Text)
```

```
            productoa.stock = Integer.Parse(txtStock.Text)
            Try
               bsnProducto.setProducto(productoa)
               MsgBox("El producto se ingreso satisf0actoriamente",
MsgBoxStyle.Information, "¡Bien!")
               txtStock.Text = ""
               txtNombre.Text = ""
               txtPrecio.Text = ""
            Catch ex As Exception
               MessageBox.Show(ex.Message)
            End Try
         Else MsgBox("Debe ingresar todos los campos", MsgBoxStyle.Information,
"Error")
         End If
      End If
      If oper = operaciones.modificar Then
         product.nombre = txtNombre.Text
         product.stock = Integer.Parse(txtStock.Text)
         product.precio = Integer.Parse(txtPrecio.Text)
         Try
            bsnProducto.updateProducto(product)
            MsgBox("El producto se modifico satisfactoriamente",
MsgBoxStyle.Information, "¡Bien!")
            txtStock.Text = ""
            txtNombre.Text = ""
            txtPrecio.Text = ""
         Catch ex As Exception
            MessageBox.Show(ex.Message)
         End Try
      End If
   End Sub

   Private Sub txtPrecio_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
txtPrecio.KeyPress
      Dim re As New Regex("[^0-9\b]", RegexOptions.IgnoreCase)
      e.Handled = re.IsMatch(e.KeyChar)
   End Sub
   Private Sub txtStock_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
txtStock.KeyPress
      Dim re As New Regex("[^0-9\b]", RegexOptions.IgnoreCase)
      e.Handled = re.IsMatch(e.KeyChar)
   End Sub
   Private Sub btnStock_Click(sender As Object, e As EventArgs) Handles
btnStock.Click
      Dim stock As New FrmStock
      stock.ShowDialog()
      txtStock.Text = bsnProducto.getStock(product)
   End Sub
   Private Sub btnSalir_Click(sender As Object, e As EventArgs) Handles btnSalir.Click
      Me.Dispose()
   End Sub
End Class
```