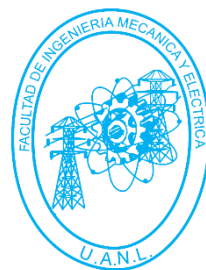




**Universidad Autónoma de
Nuevo León**



**Facultad de Ingeniería
Mecánica y Eléctrica**

Programación Orientada a Objetos

Documentación del Proyecto

Para:

“Comedores Industriales del Pacífico”

Nombre: Michael Antonio Noguera Guzmán

Carrera: Ingeniería en Tecnología de Software

Matrícula: 2077402

Hora: M2

Asesor: Dr. Oscar Rangel Aguilar

Asesor TPOO: Delia Guadalupe Elizondo Sillas



Índice

0.- LLUVIA DE IDEAS	3
1.1- NOMBRE DEL PROYECTO:	3
1.2.- INTEGRANTES	3
2.- OBJETIVOS	3
2.1- RESUMEN	4
2.2- ABSTRACT.....	5
3.- JUSTIFICACIÓN	6
4.- INVESTIGACIÓN	7
5.- REQUERIMIENTOS	10
5.1 REQUERIMIENTOS COMPLEMENTARIOS.....	11
5.5.- ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE(ESTÁNDAR IEEE830) ..	12
6.1- DIAGRAMA ENTIDAD RELATIVA	23
7.- ESPAÑOL ESTRUCTURADO	24
8.- DICCIONARIO DE DATOS.....	33
9.- BASE DE DATOS CODIGO	38
10.-INTERFACES(Mock-ups)	51
10.1-INTERFACES DE ENTRADA.....	57
10.2-INTERFACES DE SALIDA	61
11.- HERRAMIENTAS CASE (Computer Aided Software Engineering)	65
11.- Polimorfismos	66
12.- CONCLUSION GENERAL DEL PROYECTO	69
13.- REFERENCIAS BIBLIOGRAFICAS	70
14.- ANEXOS.....	71
15.- PRONÓSTICO DE TIEMPOS (Gráfica de Gantt)	129

0.- LLUVIA DE IDEAS:

(Todas las ideas son orientadas al comedor industrial)

- » Administración de Detallada: Administración de platillos, productos, ventas y gastos diarios, distribuidores y consumos.
- » Gestor de Empleados: Gestor de identificación de empleados
- » Receptor de Apartados: Recepción de pedidos por plataforma en línea.

1.1- NOMBRE DEL PROYECTO:

- » Gestor Local “Del Pacífico” → Comedores Industriales del Pacífico.

1.2.- INTEGRANTES:

- » 2077402 – Noguera Guzmán Michael Antonio – ITS – M2.

2.- OBJETIVOS:

- » Crear una estandarización en el formato de los registros.
- » Aumentar el tiempo productivo del personal.
- » Disminuir la carga de tiempo al abrir y cerrar el punto comida.
- » Eliminar el uso de los registros en papel innecesarios.
- » Generar perfiles para así identificar por quien fue realizado el registro.
- » Generar un reporte de acceso rápido y detallado para los altos mandos.

2.1- RESUMEN

“Comedores Industriales del Pacífico” es una empresa dedicada a la elaboración, distribución y venta de alimentos en el sector Industrial del Municipio de Baja California, Tijuana. A lo largo de los 2 años de servicio de la empresa, todos los registros y administración de datos se han realizado en papel, lo cual genera un entorpecimiento de los tramites; dado a que en el furor del sector industrial es imposible tener en el área de trabajo cuadernos o papeles, pues más que ayudar terminarían generando un estorbo a la hora de moverse e incluso es peligroso para los datos, pues cualquier accidente terminaría generando una pérdida de los registros, lo cual es altamente negativo para la administración de la mencionada empresa.

La empresa se vería beneficiada de forma inmensa con la implementación de un sistema informático, el cual permita mantener toda la información relacionada a la estación de alimentos, así como la administración de la información requerida para poder mantenerla en un buen estado.

Dado a que la empresa busca una solución para lo anterior se consideró que requería una especie de híbrido entre un ERP y un SGA, dado a que la información relevante para los altos cargos es toda relacionada a las ventas de la comida, mientras que la información relevante para el personal local es relacionada a los sus proveedores de alimentos.

El sistema será elaborado en el IDE Visual Studio 2022, en el que las interfaces graficas se desarrollaran usando el Framework “Bunifu”, y el código del sistema será elaborado en Basic, a su vez la información será almacenada mediante el uso de una base de datos MySQL en Microsoft SQL Server 2019.

Palabras Clave: SQL, Visual Studio, Bunifu, SQL Server, ERP, SGA.

2.2- ABSTRACT:

"Comedores Industriales del Pacífico" is a company dedicated to the preparation, distribution, and sale of food in the Industrial sector of the Municipality of Baja California, Tijuana. Throughout the 2 years of service of the company, all records and data management have been done on paper, which generates a hindrance in the procedures; given that in the furor of the industrial sector it is impossible to have notebooks or papers in the work area, because more than helping they would end up generating a hindrance when moving and it is even dangerous for the data, since any accident would end up generating a loss of records, which is highly negative for the management of the aforementioned company.

The company would benefit immensely from the implementation of a computer system, which allows maintaining all the information related to the food station, as well as the administration of the information required to keep it in good condition.

Given that the company is looking for a solution for the above, it was considered that it required a kind of hybrid between an ERP and a WMS, given that the information relevant to senior management is all related to food sales, while the information relevant to local staff is related to their food providers.

The system will be developed in the IDE Visual Studio 2022, in which the graphical interfaces will be developed using the "Bunifu" Framework, and the system code will be developed in Basic, in turn the information will be stored using a database MySQL in Microsoft SQL Server 2019.

Keywords: SQL, Visual Studio, Bunifu, SQL Server, ERP, SGA.

3.- JUSTIFICACIÓN:

» A lo largo de los aproximadamente 2 años que ha estado la empresa en función, los registros de los distintos puntos de venta se han realizado de forma manual (es decir que se realizan a mano), por lo que la implementación de un sistema de administración de estos datos es vital y podría generar un impacto realmente positivo para la empresa, o por lo menos en la administración local del punto de venta, dado a que es realmente importante una gestión organizada estos datos, dado a que los altos puestos de esta empresa requieren por obvios motivos tener acceso a estos datos, de forma rápida así como detallada.

» Lo anterior es respecto a la alta administración, sin embargo, toda esta captura de datos es realizada a mano, por lo que digitalizar y estandarizar esto genera un beneficio para el encargado de la captura de datos, pues este se ahorra el tiempo de generar una plantilla por día, y es incluso más difícil perder o traspapelar registros entre turnos, lo cual podría suceder algunas veces de forma accidental.

4.- INVESTIGACIÓN

Comedores Industriales del Pacífico:

Esta empresa se ha dedicado a la creación, distribución y venta de comida a nivel industrial, esta se encuentra ubicada en la Ciudad de Tijuana, Baja California. Esta empresa funge con las funciones de distribución de producto de distintas marcas grandes, por ejemplo, Coca-Cola, Sabritas y Lala.

Contexto actual de negocios similares.

En las ciudades industriales grandes como lo es Tijuana es muy común encontrar negocios como este, negocios que se encargan “quitar” responsabilidades a las grandes industrias, ya sea de limpieza o de alimentación.

Existen muchas empresas que se encargan de igual forma que “comedores industriales del pacífico” se encargan del sector alimentario de las grandes industrias.

Comedor Empresarial Bajasin

Somos una empresa iniciando en el proyecto de Servicio de Comedores Industriales, con ideas revolucionarias para el mejoramiento de Calidad en nuestros productos terminados y Servicio al Cliente. Concepto muy bien definido en nuestro personal que tendremos al servicio, con la garantía de que nuestros procesos de capacitación continua, nos da la seguridad de que nuestro desempeño sea satisfactorios para su personal, el cual son nuestro propósito de calidad y servicio, tanto en nuestro producto terminado como en el servicio en nuestras líneas de atención personal al cliente, sin perder la idea de que nuestra meta, es su satisfacción completa.



Cocina Industrial de La Cocina de Mamá

Somos una empresa con más de 20 años de experiencia en la atención de comedores industriales, eventos empresariales y familiares, donde diseñamos menús de acuerdo con las características y necesidades de nuestros clientes. Contamos con certificación ante REPSE como empresa socialmente responsable y cumplida

"La cocina de Mamá"



Cocina industrial... con sabor casero !

Servicio profesional de alimentos para la industria maquiladora, con más de 20 años de experiencia, así como servicios de box lunch, coffee break, mesa de bocadillos y taquizas para eventos empresariales y familiares

Comedor Industrial Coffe Break

Empresa dedicada a la elaboración de comida, damos el servicio de comedor y cafetería con un concepto diferente.

Nuestro servicio va dirigido a Industrias y Escuelas.

Somos una empresa Joven con personal de gran experiencia.



En todos los casos anteriormente mencionados se platico apreciamos que la administración es siempre realizada usando el programa de office Excel, lo cual resulta hasta curioso, dado a que la plataforma no cuenta con ningún tipo de identificación de usuario, ni registros controlados, por lo que es hasta cierto punto interesante, dado a que los registros se cuentan expuestos a modificaciones ilegales y es peligroso para la integridad de los datos almacenados a lo largo de todo el día, semana o mes.

Algunos de los anteriormente mencionados llevan a cabo la administración en un cuaderno o literalmente en un papel.

Efectos de la pandemia Covid-19.

En este caso le negocio vino aprovechando el inicio de la pandemia para arrancar en su carrera, podemos observar como esto beneficio de alguna forma al negocio, dado a que mucho personal había perdido sus trabajos por caídas de negocios, mismo que ahora pertenece a la empresa, lo cual permitió un arranque veloz de la empresa con eficiencia y calidad.

Sistemas Administrativos Existentes.

Sistemas Especializados como tal en este ámbito son pocos o inexistentes, pero existe software dedicado para puntos de venta como abarrotes, en este caso sería el que mas embona con la descripción del negocio.

1.- MANAGEMENT Pro-ERP

<https://www.mproerp.com/>

Este es uno de los preferidos de aquellas abarroteras que tienen varias sucursales, una de las características principales es que integra una Caja de Autoservicio en donde puedes conectar tu báscula y registrar productos por peso, además te muestra ofertas e imágenes de los productos en las pantallas e imprime tus comprobantes.

2.- SICAR POS

<https://www.sicar.mx/sistema-de-abarrotes-punto-de-venta/>

Sicar es un programa desarrollado por una empresa familiar, lo que hace que la asesoría y soporte se sienta más cercana, también ofrece opciones especializadas para tiendas de abarrotes lo que lo hace atractivo al usar una interfaz muy amigable y fácil de entender.

3.- ELEVENTA

<https://www.eleventa.com/>

Eleventa es conocido por su alta trayectoria, ya que antes se llamaba «Abarrotes Punto de Venta» creado desde el 2001 y comercializado desde el año 2007 también es un software especializado en abarroteras. Siempre está en constante actualización para nuevas funciones y mejoras útiles para múltiples sucursales.

5.- REQUERIMIENTOS

Gracias a las constantes pláticas con José Manuel Velarde Hernández (quien fue supervisor general por más de 17 meses en la empresa), logramos dar con múltiples puntos críticos indispensables a la hora de desarrollar la aplicación.

La aplicación debe:

- » Permitir la gestión de usuarios, los cuales fungirán a modo de firma en los registros realizados. Es decir que permita dar de alta nuevos usuarios, y que estos posean una identidad en el sistema.
- » Poder almacenar los registros de las ventas de comida realizada. Así como su desglose detallado a los distintos sectores de venta, específicamente según: el horario, en qué forma se “vendió” la comida, cuantas comidas fueron vendidas en el turno, así como en el día.
- » Poder almacenar las gastos y ventas diarias, con su respectivo motivo y monto. Así como calcular un balance monetario local, mostrar con cuánto dinero se cuenta teóricamente, para así poder llevar la administración de los distintos distribuidores.
- » Poder llevar a cabo un registro de los distribuidores, monto aproximado, día en el que se presentan, número telefónico, nombre, nombre del empleado. Para así poder llevar una administración relacionada a cuánto dinero le podría faltar a la caja, y notificar a los altos cargos.
- » Permitir el acceso a la información, de forma resumida a primera instancia, y de forma detallada si así se desea.
- » Poder llevar a cabo una gestión del horario, de forma automatizada, para así poder llevar una organización laboral de mayor calidad.

5.1 REQUERIMIENTOS COMPLEMENTARIOS

De los requerimientos complementarios tenemos los siguientes, enfocados más en el rendimiento de la aplicación y su aplicación:

- » Una rápida inicialización para iniciar el día.
- » Una fácil y rápida gestión de los datos, permitir las entradas de forma rápida, con atajos de teclado de ser posible.
- » Un acceso veloz e intuitivo a los distintos segmentos de información.
- » Ser rápida para ser ejecutada en pc de bajos recursos.
- » Permitir crear respaldos para almacenar en algún USB

» Visualmente debe de:

Acorde a los colores de la empresa.

Acorde a la tipografía de la empresa.

Ser intuitiva y fácil de usar, aun por personas ajenas al sistema.

Con una cantidad mínima de decoración (evitar sobre carga de imágenes).

» Los reportes deben de:

Incluir la fecha.

Incluir quien realizo la captura de datos.

Resumir la información lo más posible.

Permitir acceso a la información detallada.

5.5.- Especificación de Requisitos de Software (estándar IEEE830)

Requerimientos Funcionales

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Identificación y Autenticación de Usuarios
Características:	Todos los usuarios deben e estar identificados para poder realizar cualquier función del sistema.
Descripción del requerimiento:	El sistema será consultado por los usuarios dependiendo del módulo que se encuentre.
Requerimiento NO funcional:	RNF01 - RNF02 - RNF03
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Alta de Registros
Características:	Los usuarios deberán de estar registrados para poder usar cualquier funcionalidad del sistema.
Descripción del requerimiento:	El registro permitirá al usuario registrarse
Requerimiento NO funcional:	RNF04 - RNF05 - RNF06
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Alta de Registros
Características:	Los usuarios podrán dar de alta registros de distribuidores
Descripción del requerimiento:	El registro permitirá al usuario registrar distribuidores
Requerimiento NO funcional:	RNF04 - RNF05 - RNF06
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Alta de Registros
Características:	Los usuarios podrán dar de alta registros de comidas
Descripción del requerimiento:	El registro permitirá al usuario registrar comidas
Requerimiento NO funcional:	RNF04 - RNF05 - RNF06
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Consulta de Registros
Características:	Los usuarios podrán dar de alta registros de ventas y gastos
Descripción del requerimiento:	El registro permitirá al usuario consultar ventas y gastos
Requerimiento NO funcional:	RNF07 - RNF08
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Consulta de Registros
Características:	Los usuarios podrán pedir registros de comidas
Descripción del requerimiento:	El registro permitirá al usuario consultar registros
Requerimiento NO funcional:	RNF07 - RNF08
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Consulta de Registros
Características:	Los usuarios podrán pedir registros de apuntes
Descripción del requerimiento:	El registro permitirá al usuario consultar apuntes
Requerimiento NO funcional:	RNF07 - RNF08
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RF08
Nombre del Requerimiento:	Modificar Registros
Características:	Los usuarios en todo momento serán capaces de modificar distribuidores
Descripción del requerimiento:	El registro permitirá al usuario modificar Distribuidores
Requerimiento NO funcional:	RNF09 - RNF10
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RF09
Nombre del Requerimiento:	Modificar Registros
Características:	Los usuarios en todo momento serán capaces de modificar Registros de Apuntes
Descripción del requerimiento:	El registro permitirá al usuario modificar Apuntes
Requerimiento NO funcional:	RNF09 - RNF10
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RF10
Nombre del Requerimiento:	Gestionar Reportes
Características:	Los usuarios podrán generar reportes de distintas tablas de datos
Descripción del requerimiento:	El registro permitirá al usuario generar reportes, de ventas y comidas
Requerimiento NO funcional:	RNF09 - RNF10
Prioridad del Requerimiento	ALTA

Requerimientos No Funcionales

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Interfaz del sistema
Características:	El interfaz del sistema presenta un interfaz sencilla y limpia
Descripción del requerimiento:	El sistema posee un interfaz intuitiva y facil de comprender
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Confiabilidad continúa del sistema.
Características:	El sistema posee un desempeño uniforme y veloz
Descripción del requerimiento:	No existen intermitencias entre los distintos procesos, y no genera malfuncionamientos
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Nivel de Usuario
Características:	Garantiza al usuario acceso a la información de acuerdo a su nivel en la empresa
Descripción del requerimiento:	Los usuarios tendrán acceso a la información que ellos tienen refiriendo al nivel de información que este posea
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RNF04
Nombre del Requerimiento:	Desempeño
Características:	Los datos que se trabajan en los distintos niveles, estará siempre disponible
Descripción del requerimiento:	El desempeño de la información almacenada siempre, será lo mas eficiente posible.
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RNF05
Nombre del Requerimiento:	Seguridad en información
Características:	El sistema garantizará a los usuario la mejor seguridad en cuanto a la información que proviene del sistema.
Descripción del requerimiento:	Toda la información tratada a lo largo de los procesos sera procesada con prioridad de seguridad
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RNF06
Nombre del Requerimiento:	Accesibilidad a los datos
Características:	Los usuarios tendrán permiso de acceder a absolutamente toda la información
Descripción del requerimiento:	Los usuarios pueden acceder a la información, unicamente para su lectura en un principio, basado en sus niveles de permisos
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RNF07
Nombre del Requerimiento:	Identificación de Registros
Características:	Todos los registros al ser visualizados contarán con identificadores
Descripción del requerimiento:	Al siempre mostrar los registros podremos ver los datos de la procedencia, es decir la cuenta o usuario que genero el registro
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RNF08
Nombre del Requerimiento:	Hora de los Registros
Características:	Los registros al ser visualizados contarán con identificadores de tiempo
Descripción del requerimiento:	Los registros contarán con la hora específica a la que fueron realizados
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RNF09
Nombre del Requerimiento:	Comprensión de datos
Características:	El sistema contará con sistemas de simplificación
Descripción del requerimiento:	El usuario tendrá acceso únicamente a la información vital para el proceso, simplificando la masa de información.
Prioridad del Requerimiento	ALTA

Identificación del requerimiento:	RNF10
Nombre del Requerimiento:	Mantenimiento
Características:	El sistema contará con la información necesaria para realizar el mantenimiento del sistema
Descripción del requerimiento:	El sistema contará con una documentación y un manual de instalación que permitan realizar operaciones de mantenimiento con el menor esfuerzo posible
Prioridad del Requerimiento	ALTA

Requerimientos para el Sistema

Requerimientos de Hardware

Periféricos:

Teclado

Ratón

Monitor

Equipo (Laptop)

Componentes:

Memoria RAM: 4 GB

Procesador: 2.0 GHz

HDD: 250 GB

Requerimientos de Software

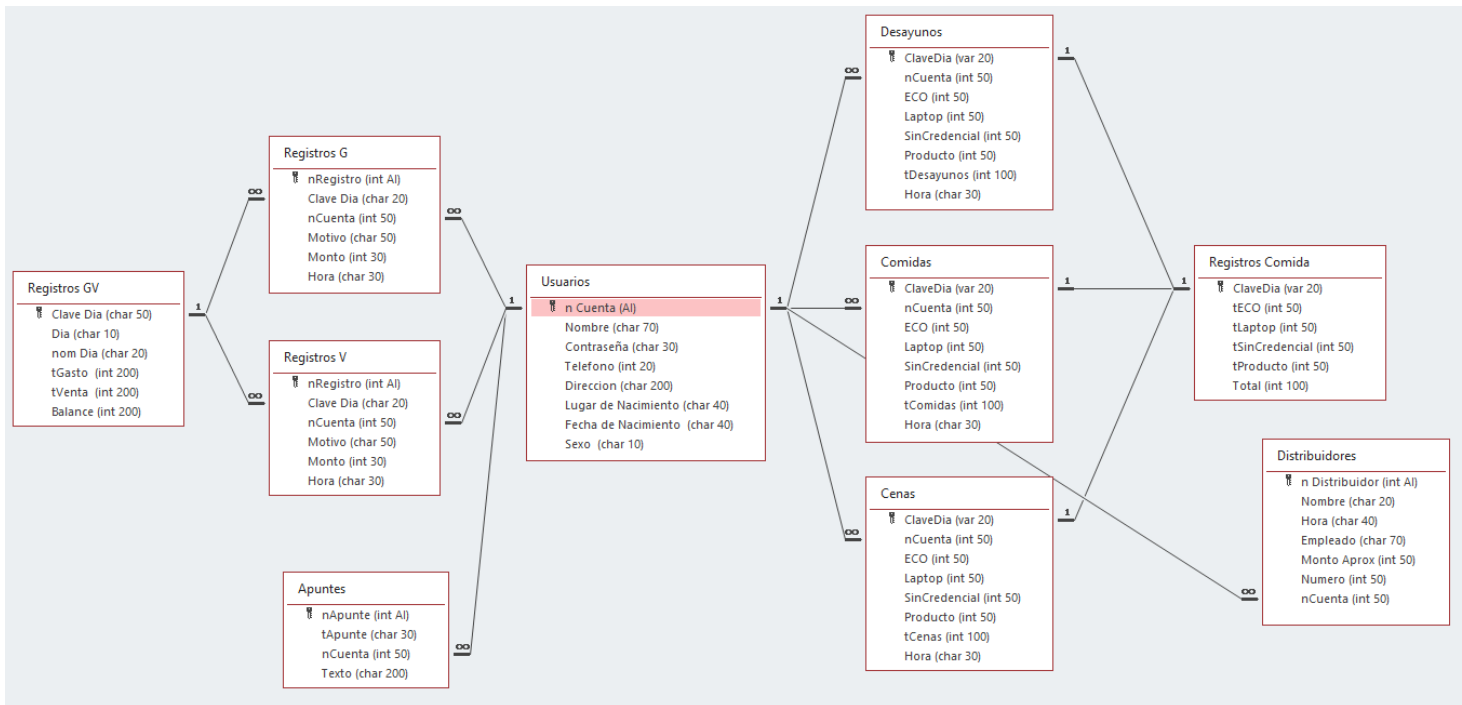
Sistema Operativo: Windows 8 (En adelante)

Microsoft .Net Framework 3.5.2 (En adelante)

Conexión a Internet Local

SQL Server 2018 Express

6.1- DIAGRAMA ENTIDAD RELATIVA



Debido a que se tiene planeado almacenar todos los datos la base de datos (ósea sin eliminar datos), y esta situación se genera en el procesamiento de datos de: las comidas, y los gastos/ventas.

Con el propósito de generar estadísticas a nivel diario y semanal, y mensual, se ha decidido iniciar una tabla que almacene los datos ya procesados diarios, para así no tardar tanto tiempo realizando búsquedas en la tabla día por día, dado a que las estadísticas son una parte importante para el cliente, se busca reducir la información lo máximo posible.

Como petición especial del asesor del proyecto se tomó la decisión de crear un pequeño block de notas para el cajero, planteado para crear breves registros, breves pero críticos registros, como pedidos especiales directos de recursos humanos o directivos, para así no perder datos en papel a la hora de cambio de personal o de turno.

7.- ESPAÑOL ESTRUCTURADO

Inicio de Sesión:

(La función será lanzada directamente por el botón)

SI txtUsuario.value != "" Y txtContra.value != "" ENTONCES

SI ExisteEnTabla (txtUsuario.value) ENTONCES

SI txtContra == Usuario.Contraseña ENTONCES

loginExitoso();

SI NO

NotificarUsuario("Contraseña Incorrecta")

FIN SI

SI NO

NotificarUsuario("Usuario Inexistente")

FIN SI

SI NO

NotificarUsuario("Rellena las cajas")

FIN SI

Creación Usuario:

(La función será lanzada directamente por el botón)

SI txtNombre,txtContra,txtTelefono,txtDire,txtLugar,txtFecha,txtSexo!="" ENTONCES

SI ExisteEnTabla(UsuarioGenerado) ENTONCES

NotificarUsuario("El usuario Generado ya existe")

SI NO

DarDeAlta(Usuario, Nombre, Contra, Telefono, Luga, Fecha, Sexo)

FIN SI

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Mostrar Distribuidor:

(Función se lanza si se accede a la pestaña de distribuidores)

Hacer:

CrearObjeto(regActual(Clave, Nombre))

Si Click(SobreElObjeto) **ENTONCES**

MostrarDatos(nDistribuidor, Nombre, Hora, Empleado, Monto, Telefono)

FIN SI

Mientras (ExistanRegistrosEnTabla())

Alta de Distribuidor:

(La función será lanzada desde mostrar distribuidores, como opción, al hacer click)

SI txtNombre, txtHora, txtEmpleado, txtMonto, txtNumero, txtCuenta != "" **ENTONCES**

SI ExisteEnTabla (Nombre) **ENTONCES**

NotificarUsuario("El distribuidor ya esta dado de alta")

SI NO

DarDeAlta(claveDistrib,Nombre,Hora,Empleado,Monto,Numero,nCuenta)

Reseteartabla()

FIN SI

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Modificación de Distribuidor:

(La función será lanzada desde mostrar distribuidores, como opción, al hacer click)

SI txtNombre, txtHora, txtEmpleado, txtMonto, txtNumero, txtCuenta != "" **ENTONCES**

Actualizar(txtNombre, txtHora, txtEmpleado, txtMonto, txtNumero, txtCuenta)

Reseteartabla()

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Baja de Distribuidor:

(La función será lanzada desde mostrar distribuidores, como opción, al hacer click)

SI txtUsuarioBaja!="" ENTONCES

MostrarRegistro(txtUsuarioBaja)

NotificarUsuario("El registro:"+values+"sera eliminado, ¿desea continuar?")

SI Alerta==TRUE ENTONCES

DarDeBaja(txtUsuarioBaja!)

SI NO

NotificarUsuario("Proceso Cancelado")

FIN SI

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Alta Desayuno:

(La función será lanzada desde Desayuno, como opción, al hacer click)

SI txtECO !="" Y txtLaptop !="" Y txtSinCredencial !="" Y txtProducto!=" ENTONCES

Hora = ExtraerHora()

SI Hora>=10:40 Y Hora <= 11:00 ENTONCES

fueraTiempo=0

SI NO

NotificarUsuario("El registro será almacenado como fuera de tiempo")

fueraTiempo=1

FIN SI

nCuenta = extraerUsuarioActual()

AltaDesayuno(nRegistro,ClaveDia,nCuenta,Motivo,Monto,Hora)

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Alta Comida:

(La función será lanzada desde Comida, como opción, al hacer click)

SI txtECO !="" Y txtLaptop !="" Y txtSinCredencial !="" Y txtProducto!="" ENTONCES

Hora = ExtraerHora()

SI Hora>=10:40 Y Hora <= 11:00 ENTONCES

fueraTiempo=0

SI NO

NotificarUsuario("El registro será almacenado como fuera de tiempo")

fueraTiempo=1

FIN SI

nCuenta = extraerUsuarioActual()

AltaDesayuno(nRegistro,ClaveDia,nCuenta,Motivo,Monto,Hora)

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Alta Cena:

(La función será lanzada desde Cena, como opción, al hacer click)

SI txtECO !="" Y txtLaptop !="" Y txtSinCredencial !="" Y txtProducto!="" ENTONCES

Hora = ExtraerHora()

NotificarUsuario("El registro será almacenado como fuera de tiempo")

fueraTiempo=1

FIN SI

nCuenta = extraerUsuarioActual()

AltaDesayuno(nRegistro,ClaveDia,nCuenta,Motivo,Monto,Hora)

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Alta Gastos:

(La función será lanzada desde "GastosVentas", como opción, al hacer click)

SI motivo!=" " monto!=" " ENTONCES

ClaveDelDia = Primeras3Letras+ExtraerFecha()

ExtraerUsuario()

SI ExisteRegistro(ClaveDelDia) ENTONCES

ActualizarBaseDelDia(ClaveDelDia,0,0,0,0,0,0)

SINO

CrearBaseDelDia(ClaveDelDia,0,0,0,0,0,0)

FIN SI

AltaGasto(ClaveDelDia,nCuenta,Motivo,Monto,Hora)

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Alta Venta:

(La función será lanzada desde "GastosVentas", como opción, al hacer click)

SI motivo!=" " monto!=" " ENTONCES

ClaveDelDia = Primeras3Letras+ExtraerFecha()

ExtraerUsuario()

SI ExisteRegistro(ClaveDelDia) ENTONCES

ActualizarBaseDelDia(ClaveDelDia,0,0,0,0,0,0)

SINO

CrearBaseDelDia(ClaveDelDia,0,0,0,0,0,0)

FIN SI

AltaGasto(ClaveDelDia,nCuenta,Motivo,Monto,Hora)

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Alta Día:

(Función Lanzada Por La Hora, o al cerrar la aplicación)

ClaveDiaG = Primeras3Letras+ExtraerFecha()

A = Select * from tablaDesayunos where ClaveDia == ClaveDiaG

B = Select * from tablaComidas where ClaveDia == ClaveDiaG

C = Select * from tablaCenas where ClaveDia == ClaveDiaG

tECO = A.ECO + B.ECO + C.ECO

tLaptop = A.Laptop + B. Laptop + C. Laptop

tSinCred = A.SinCredencial + B. SinCredencial + C. SinCredencial

tProducto = A.Producto + B. Producto + C. Producto

tTotal = tECO + TLaptop + tSinCredencial – tProducto

SI ExisteRegistro(ClaveDiaG) **ENTONCES**

ActualizarBaseDelDia(ClaveDelDia, tECO, tLaptop, tSinCred, tProducto, Total)

SI NO

notificarUsuario("No existen datos del dia aun, no se creara nada")

FIN SI

Mostrar Día:

(Función Lanzada a la hora de mostrar los datos de los dias)

Hacer:

CrearObjeto(regActual(i,ClaveDia))

Si Click(SobreElObjeto) **ENTONCES**

MostrarDatos(ClaveDia, tECO, tLaptop, tSinCred, tProducto, Total)

FIN SI

Mientras (ExistanRegistrosEnTabla(RegistrosComidas))

Mostrar Apunte:

(Función se lanza si se accede a la pestaña de distribuidores)

Hacer:

CrearObjeto(regActual(Clave, Nombre))

Si Click(SobreElObjeto) ENTONCES

MostrarDatos(nDistribuidor, Nombre, Hora, Empleado, Monto, Telefono)

FIN SI

Mientras (ExistanRegistrosEnTabla())

Dar de alta apunte:

(Función Lanzada Por La Hora, o al cerrar la aplicación)

SI txtTitulo!="" Y txtCuenta!="" Y txtTexto!="" ENTONCES

SI ExisteTitulo(txtTitulo) ENTONCES

Alerta("Este titulo ya existe, ¿desea modificar este registro?")

SI Alerta==Verdadero ENTONCES

ActualizarApunte(txtTitulo.value)

SI NO

NotificaUser("Entonces tendra que cambiar el nombre del titulo")

FIN SI

SI NO

AltaDeApunte(txtTitulo,txtCuenta,txtTexto)

FIN SI

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Borrar Apunte

(Función Lanzada Por La Hora, o al cerrar la aplicación)

SI txtTitulo!="" ENTONCES

Alerta("Este registro será eliminado, ¿seguro que deseas continuar?")

SI Alerta==Verdadero ENTONCES

BorrarApunte(txtTitulo)

NotificarUsuario("Apunte eliminado correctamente")

SI NO

NotificarUsuario("No se ha eliminado el registro")

FIN SI

SI NO

NotificarUsuario("Rellena la caja")

FIN SI

Actualizar Apunte

(Función Lanzada Por La Hora, o al cerrar la aplicación)

SI txtTitulo!="" Y txtCuenta!="" Y txtTexto!="" ENTONCES

Alerta("Los valores serán remplazados, ¿seguro que deseas continuar?")

SI Alerta==Verdadero ENTONCES

ActualizarApunte(txtTitulo,txtCuenta,txtTexto)

NotificarUsuario("Apunte actualizado correctamente")

SI NO

NotificarUsuario("No se han realizado actualizaciones")

FIN SI

SI NO

NotificarUsuario("Rellena todas las cajas")

FIN SI

Apunte Existe

(Lanzado directamente por otras funciones, es una función auxiliar)

valorRecibido

Existe = 0;

Sentencia = Select * from Apuntes where titulo== valorRecibido

mysqli_num_rows(Sentencia)

SI mysqli_num_rows(Sentencia) != 0 ENTONCES

Existe = 1;

FIN SI

8.- DICCIONARIO DE DATOS

Tabla Usuario:

Nombre de la tabla: *Usuarios*

Fecha de Creación: ---

Descripción: Tabla encargada de almacenar todos los datos de los usuarios

Campo:	Tipo:	Tamaño:	Descripción:
nCuenta	Carácter	20	Clave del día
Nombre	Número	50	Número de la cuenta que hizo el registro
Contraseña	Número	50	Número de consumidores de ECO
Telefono	Número	50	Número de registros con tarjeta
Direccion	Número	50	Número de registros o de nuevos, o sin tarjeta
LuNacimiento	Número	50	Número de personas que tomaron producto
FeNacimiento	Número	100	Total de Platos de desayunos consumidos
Hora	Carácter	30	Hora a la que fue realizado el registro

Campos Clave:

ClaveDia

nCuenta

Relaciones:

ClaveDia con el campo **ClaveDia** de la tabla **RegComida**

nCuenta con el campo **nCuenta** de la tabla **Usuarios**

Tabla Desayuno:

Nombre de la tabla: *Desayunos*

Fecha de Creación: ---

Descripción: Tabla encargada de almacenar los valores del desayuno de 1 solo día.

Campo:	Tipo:	Tamaño:	Descripción:
ClaveDia	Carácter	20	Clave del día
nCuenta	Número	50	Número de la cuenta que hizo el registro
ECO	Número	50	Número de consumidores de ECO
Laptop	Número	50	Número de registros con tarjeta
SinCredencial	Número	50	Número de registros o de nuevos, o sin tarjeta
Producto	Número	50	Número de personas que tomaron producto
tDesayunos	Número	100	Total de Platos de desayunos consumidos
Hora	Carácter	30	Hora a la que fue realizado el registro

Campos Clave:

ClaveDia

nCuenta

Relaciones:

ClaveDia con el campo **ClaveDia** de la tabla **RegComida**

nCuenta con el campo **nCuenta** de la tabla **Usuarios**

Tabla Comida:

Nombre de la tabla: Comidas

Fecha de Creación: ---

Descripción: Tabla encargada de almacenar los valores de la comida de 1 solo día

Campo:	Tipo:	Tamaño:	Descripción:
ClaveDia	Carácter	20	Clave del día
nCuenta	Númerico	50	Número de la cuenta que hizo el registro
ECO	Númerico	50	Número de consumidores de ECO
Laptop	Númerico	50	Número de registros con tarjeta
SinCredencial	Númerico	50	Número de registros o de nuevos, o sin tarjeta
Producto	Númerico	50	Número de personas que tomaron producto
tComidas	Númerico	100	Total de Platos de Comidas consumidas
Hora	Carácter	30	Hora a la que fue realizado el registro

Campos Clave:

ClaveDia

nCuenta

Relaciones:

ClaveDia con el campo **ClaveDia** de la tabla **RegComida**

nCuenta con el campo **nCuenta** de la tabla **Usuarios**

Tabla Cena:

Nombre de la tabla: Cenas

Fecha de Creación: ---

Descripción: Tabla encargada de almacenar los valores de la cena de 1 solo día.

Campo:	Tipo:	Tamaño:	Descripción:
ClaveDia	Carácter	20	Clave del día
nCuenta	Númerico	50	Número de la cuenta que hizo el registro
ECO	Númerico	50	Número de consumidores de ECO
Laptop	Númerico	50	Número de registros con tarjeta
SinCredencial	Númerico	50	Número de registros o de nuevos, o sin tarjeta
Producto	Númerico	50	Número de personas que tomaron producto
tCenas	Númerico	100	Total de Platos de Cena consumidos
Hora	Carácter	30	Hora a la que fue realizado el registro

Campos Clave:

ClaveDia

nCuenta

Relaciones:

ClaveDia con el campo **ClaveDia** de la tabla **RegComida**

nCuenta con el campo **nCuenta** de la tabla **Usuarios**

Tabla Registro de Comidas:

Nombre de la tabla: RegComida

Fecha de Creación: ---

Descripción: Tabla encargada de almacenar todos los valores del día.

Campo:	Tipo:	Tamaño:	Descripción:
ClaveDia	Carácter	20	Clave del día
tECO	Númerico	50	Número de consumidores de ECO totales
tLaptop	Númerico	50	Número de registros con tarjeta totales
tSinCredencial	Númerico	50	Número de registros nuevos, sin tarjeta totales
tProducto	Númerico	50	Número de registros por producto totales
Total	Númerico	50	Ventas por platillo totales del día

Campos Clave:

ClaveDia

Relaciones:

ClaveDia con el campo **ClaveDia** de la tablas:
Desayunos, Comidas, Cenas.

Tabla Registro Gastos:

Nombre de la tabla: RegistrosGastos

Fecha de Creación: ---

Descripción: Tabla encargada de almacenar todas las ventas del día.

Campo:	Tipo:	Tamaño:	Descripción:
nRegistro	Númerico	AI	Identificador del número de registro
ClaveDia	Carácter	20	Clave del día
nCuenta	Númerico	50	Número de la cuenta que hizo el registro
Motivo	Carácter	50	Motivo Del Gasto
Monto	Númerico	30	Monto Requerido Para el gasto
Hora	Carácter	50	Ventas por platillo totales del día

Campos Clave:

nRegistro

ClaveDia

nCuenta

Relaciones:

N/A

ClaveDia con el campo **ClaveDia** de la tabla **RegistrosGV**
nCuenta con el campo **nCuenta** de la tabla **Usuarios**

Tabla Registro Ventas:

Nombre de la tabla: RegistrosVentas **Fecha de Creación:** ---

Descripción: Tabla encargada de almacenar todos las ventas del día.

Campo:	Tipo:	Tamaño:	Descripción:
nRegistro	Númerico	AI	Identificador del número de registro
ClaveDia	Carácter	20	Clave del día
nCuenta	Númerico	50	Número de la cuenta que hizo el registro
Motivo	Carácter	50	Motivo Del Venta
Monto	Númerico	30	Monto Obtenido de la Venta
Hora	Carácter	50	Ventas por platillo totales del día

Campos Clave:

nRegistro

ClaveDia

nCuenta

Relaciones:

N/A

ClaveDia con el campo ClaveDia de la tabla RegistrosGV

nCuenta con el campo nCuenta de la tabla Usuarios

Tabla Registro Gastos y Ventas:

Nombre de la tabla: RegistrosGV **Fecha de Creación:** ---

Descripción: Tabla encargada de almacenar todos las ventas del día.

Campo:	Tipo:	Tamaño:	Descripción:
ClaveDia	Carácter	20	Clave del día
Día	Carácter	10	Día de la Semana literal (Lunes, Martes, etc)
tGasto	Númerico	50	Número de la cuenta que hizo el registro
tVenta	Carácter	50	Motivo Del Venta
Balance	Númerico	30	Monto Obtenido de la Venta

Campos Clave:

ClaveDia

Relaciones:

ClaveDia con el campo ClaveDia de la tablas
RegistrosG , RegistrosV

Tabla Registro Apuntes:

Nombre de la tabla: RegistrosApuntes **Fecha de Creación:** ---

Descripción: Tabla encargada de almacenar apuntes del cajero

Campo:	Tipo:	Tamaño:	Descripción:
nApunte	Númerico	AI	Número de apunte registrado
tApuntes	Carácter	30	Será el titulo del apunte realizado
nCuenta	Númerico	50	Es el autor de la cuenta
Texto	Carácter	200	Contenido del apunte

Campos Clave:

nCuenta

Relaciones:

nCuenta con el campo nCuenta de la tabla Usuario

Tabla Distribuidores:

Nombre de la tabla: Distribuidores **Fecha de Creación:** ---

Descripción: Tabla encargada de almacenar los datos de los distribuidores

Campo:	Tipo:	Tamaño:	Descripción:
nDistribuidor	Númerico	AI	nDistribuidor
Nombre	Carácter	20	Nombre de la Distribuidor
Hora	Carácter	40	Hora a la que pasa el distribuidor usualmente
Empleado	Carácter	70	Nombre del empleado de la distribuidora
MontoAprox	Númerico	50	Monto Aproximado requerido por el distribuidor
Numero	Númerico	50	Número Telefónico
nCuenta	Númerico	50	Número de la cuenta que realizo el alta

Campos Clave:

nCuenta

Relaciones:

nCuenta con el campo nCuenta de la tabla Usuario

9.- BASE DE DATOS CODIGO

Código Base de datos:

```
/****** Object: Database [Comedor_Pacifico]  Script Date: 25/06/2022 04:38:18 p. m. *****/  
  
CREATE DATABASE [Comedor_Pacifico]  
  
CONTAINMENT = NONE  
  
ON PRIMARY  
  
( NAME = N'Comedor_Pacifico', FILENAME = N'C:\Program Files\Microsoft SQL  
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Comedor_Pacifico.mdf' , SIZE = 8192KB , MAXSIZE  
= UNLIMITED, FILEGROWTH = 65536KB )  
  
LOG ON  
  
( NAME = N'Comedor_Pacifico_log', FILENAME = N'C:\Program Files\Microsoft SQL  
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Comedor_Pacifico_log.ldf' , SIZE = 8192KB ,  
MAXSIZE = 2048GB , FILEGROWTH = 65536KB )  
  
WITH CATALOG_COLLATION = DATABASE_DEFAULT  
  
GO  
  
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))  
begin  
EXEC [Comedor_Pacifico].[dbo].[sp_fulltext_database] @action = 'enable'  
end  
  
GO  
  
ALTER DATABASE [Comedor_Pacifico] SET ANSI_NULL_DEFAULT OFF  
  
GO  
  
ALTER DATABASE [Comedor_Pacifico] SET ANSI_NULLS OFF  
  
GO  
  
ALTER DATABASE [Comedor_Pacifico] SET ANSI_PADDING OFF  
  
GO  
  
ALTER DATABASE [Comedor_Pacifico] SET ANSI_WARNINGS OFF  
  
GO  
  
ALTER DATABASE [Comedor_Pacifico] SET ARITHABORT OFF  
  
GO
```

```
ALTER DATABASE [Comedor_Pacifico] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [Comedor_Pacifico] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [Comedor_Pacifico] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET DISABLE_BROKER
GO
ALTER DATABASE [Comedor_Pacifico] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
```

```
ALTER DATABASE [Comedor_Pacifico] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [Comedor_Pacifico] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET RECOVERY SIMPLE
GO
ALTER DATABASE [Comedor_Pacifico] SET MULTI_USER
GO
ALTER DATABASE [Comedor_Pacifico] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [Comedor_Pacifico] SET DB_CHAINING OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [Comedor_Pacifico] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [Comedor_Pacifico] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [Comedor_Pacifico] SET ACCELERATED_DATABASE_RECOVERY = OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET QUERY_STORE = OFF
GO
ALTER DATABASE [Comedor_Pacifico] SET READ_WRITE
GO
```


Código Tabla = Apuntes:

```
USE [Comedor_Pacifico]
GO
/***** Object: Table [dbo].[Apuntes]  Script Date: 25/06/2022 04:36:15 p. m. *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Apuntes](
    [nApunte] [int] IDENTITY(1,1) NOT NULL,
    [tApunte] [nvarchar](50) NOT NULL,
    [nCuenta] [int] NOT NULL,
    [Texto] [ntext] NOT NULL,
    [ParaTodos] [int] NOT NULL,
    CONSTRAINT [PK_Apuntes] PRIMARY KEY CLUSTERED
(
    [nApunte] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[Apuntes] WITH CHECK ADD CONSTRAINT [FK_Apuntes_UsuariosComedor]
FOREIGN KEY([nCuenta])
REFERENCES [dbo].[UsuariosComedor] ([nCuenta])
GO

ALTER TABLE [dbo].[Apuntes] CHECK CONSTRAINT [FK_Apuntes_UsuariosComedor]
GO
```

Código Tabla = Desayunos:

```
USE [Comedor_Pacifico]

/***** Object: Table [dbo].[RegDesayunos]  Script Date: 25/06/2022 04:36:54 p. m. *****/

SET ANSI_NULLS ON

SET QUOTED_IDENTIFIER ON

CREATE TABLE [dbo].[RegDesayunos](

    [ClaveDia] [nvarchar](20) NOT NULL,

    [nCuenta] [int] NOT NULL,

    [ECO] [int] NOT NULL,

    [Laptop] [int] NOT NULL,

    [SinCredencial] [int] NOT NULL,

    [PorProducto] [int] NOT NULL,

    [tDesayunos] [int] NOT NULL,

    [Hora] [nvarchar](30) NOT NULL,

    CONSTRAINT [PK_RegDesayunos] PRIMARY KEY CLUSTERED

(

    [ClaveDia] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,

ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)

ON [PRIMARY]

) ON [PRIMARY]

GO


ALTER TABLE [dbo].[RegDesayunos] WITH CHECK ADD CONSTRAINT

[FK_RegDesayunos_UsuariosComedor] FOREIGN KEY([nCuenta])

REFERENCES [dbo].[UsuariosComedor] ([nCuenta])

GO


ALTER TABLE [dbo].[RegDesayunos] CHECK CONSTRAINT

[FK_RegDesayunos_UsuariosComedor]
```

Código Tabla = Comidas:

```
USE [Comedor_Pacifico]
```

```
/***** Object: Table [dbo].[RegComidas] Script Date: 25/06/2022 04:36:44 p. m. *****/
```

```
SET ANSI_NULLS ON
```

```
SET QUOTED_IDENTIFIER ON
```

```
CREATE TABLE [dbo].[RegComidas](
```

```
    [ClaveDia] [nvarchar](20) NOT NULL,
```

```
    [nCuenta] [int] NOT NULL,
```

```
    [ECO] [int] NOT NULL,
```

```
    [Laptop] [int] NOT NULL,
```

```
    [SinCredencial] [int] NOT NULL,
```

```
    [PorProducto] [int] NOT NULL,
```

```
    [tComidas] [int] NOT NULL,
```

```
    [Hora] [nvarchar](30) NOT NULL,
```

```
CONSTRAINT [PK_RegComidas] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [ClaveDia] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)  
ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
ALTER TABLE [dbo].[RegComidas] WITH CHECK ADD CONSTRAINT  
[FK_RegComidas_UsuariosComedor] FOREIGN KEY([nCuenta])
```

```
REFERENCES [dbo].[UsuariosComedor] ([nCuenta])
```

```
GO
```

```
ALTER TABLE [dbo].[RegComidas] CHECK CONSTRAINT [FK_RegComidas_UsuariosComedor]
```

Código Tabla = Cenas:

USE [Comedor_Pacifico]

/***** Object: Table [dbo].[RegCenas] Script Date: 25/06/2022 04:36:36 p. m. *****/

SET ANSI_NULLS ON

SET QUOTED_IDENTIFIER ON

CREATE TABLE [dbo].[RegCenas](

[ClaveDia] [nvarchar](20) NOT NULL,

[nCuenta] [int] NOT NULL,

[ECO] [int] NOT NULL,

[Laptop] [int] NOT NULL,

[SinCredencial] [int] NOT NULL,

[PorProducto] [int] NOT NULL,

[tCenas] [int] NOT NULL,

[Hora] [nvarchar](20) NOT NULL,

CONSTRAINT [PK_RegCenas] PRIMARY KEY CLUSTERED

(

[ClaveDia] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]

) ON [PRIMARY]

ALTER TABLE [dbo].[RegCenas] WITH CHECK ADD CONSTRAINT

[FK_RegCenas_UsuariosComedor] FOREIGN KEY([nCuenta])

REFERENCES [dbo].[UsuariosComedor] ([nCuenta])

ALTER TABLE [dbo].[RegCenas] CHECK CONSTRAINT [FK_RegCenas_UsuariosComedor]

Código Tabla = Registros Totales:

```
USE [Comedor_Pacifico]
```

```
/****** Object: Table [dbo].[RegistrosTotales]   Script Date: 25/06/2022 04:37:33 p. m. *****/
```

```
SET ANSI_NULLS ON
```

```
SET QUOTED_IDENTIFIER ON
```

```
CREATE TABLE [dbo].[RegistrosTotales](
```

```
    [ClaveDia] [nvarchar](20) NOT NULL,
```

```
    [tEco] [int] NOT NULL,
```

```
    [tLaptop] [int] NOT NULL,
```

```
    [tSinCredencial] [int] NOT NULL,
```

```
    [tProducto] [int] NOT NULL,
```

```
    [Total] [bigint] NOT NULL,
```

```
CONSTRAINT [PK_RegistrosTotales] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [ClaveDia] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)  
ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
ALTER TABLE [dbo].[RegistrosTotales] WITH CHECK ADD CONSTRAINT  
[FK_RegistrosTotales_RegistrosTotales] FOREIGN KEY([ClaveDia])
```

```
REFERENCES [dbo].[RegistrosTotales] ([ClaveDia])
```

```
ALTER TABLE [dbo].[RegistrosTotales] CHECK CONSTRAINT [FK_RegistrosTotales_RegistrosTotales]
```

Código Tabla = Gastos:

USE [Comedor_Pacifico]

/***** Object: Table [dbo].[RegistrosG] Script Date: 25/06/2022 04:37:11 p. m. *****/

SET ANSI_NULLS ON

SET QUOTED_IDENTIFIER ON

CREATE TABLE [dbo].[RegistrosG](

[nRegistro] [int] IDENTITY(1,1) NOT NULL,

[ClaveDia] [nvarchar](20) NOT NULL,

[nCuenta] [int] NOT NULL,

[Motivo] [nvarchar](50) NOT NULL,

[Monto] [int] NOT NULL,

[Hora] [nvarchar](50) NOT NULL,

CONSTRAINT [PK_RegistrosGa] PRIMARY KEY CLUSTERED

(

[nRegistro] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]

) ON [PRIMARY]

ALTER TABLE [dbo].[RegistrosG] WITH CHECK ADD CONSTRAINT [FK_RegistrosGa_RegistrosGV]
FOREIGN KEY([ClaveDia])

REFERENCES [dbo].[RegistrosGV] ([ClaveDia])

ALTER TABLE [dbo].[RegistrosG] CHECK CONSTRAINT [FK_RegistrosGa_RegistrosGV]

ALTER TABLE [dbo].[RegistrosG] WITH CHECK ADD CONSTRAINT
[FK_RegistrosGa_UsuariosComedor] FOREIGN KEY([nCuenta])

REFERENCES [dbo].[UsuariosComedor] ([nCuenta])

ALTER TABLE [dbo].[RegistrosG] CHECK CONSTRAINT [FK_RegistrosGa_UsuariosComedor]

Código Tabla = Ventas:

USE [Comedor_Pacifico]

/***** Object: Table [dbo].[RegistrosV] Script Date: 25/06/2022 04:37:45 p. m. *****/

SET ANSI_NULLS ON

SET QUOTED_IDENTIFIER ON

CREATE TABLE [dbo].[RegistrosV](

[nRegistro] [int] IDENTITY(1,1) NOT NULL,

[ClaveDia] [nvarchar](20) NOT NULL,

[nCuenta] [int] NOT NULL,

[Motivo] [nvarchar](50) NOT NULL,

[Monto] [int] NOT NULL,

[Hora] [nvarchar](50) NOT NULL,

CONSTRAINT [PK_RegistrosV] PRIMARY KEY CLUSTERED

(

[nRegistro] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]

) ON [PRIMARY]

ALTER TABLE [dbo].[RegistrosV] WITH CHECK ADD CONSTRAINT [FK_RegistrosV_RegistrosGV]
FOREIGN KEY([ClaveDia])

REFERENCES [dbo].[RegistrosGV] ([ClaveDia])

ALTER TABLE [dbo].[RegistrosV] CHECK CONSTRAINT [FK_RegistrosV_RegistrosGV]

ALTER TABLE [dbo].[RegistrosV] WITH CHECK ADD CONSTRAINT
[FK_RegistrosV_UsuariosComedor] FOREIGN KEY([nCuenta])

REFERENCES [dbo].[UsuariosComedor] ([nCuenta])

ALTER TABLE [dbo].[RegistrosV] CHECK CONSTRAINT [FK_RegistrosV_UsuariosComedor]

Código Tabla = Gastos y Ventas:

USE [Comedor_Pacifico]

/***** Object: Table [dbo].[RegistrosGV] Script Date: 25/06/2022 04:37:22 p. m. *****/

SET ANSI_NULLS ON

SET QUOTED_IDENTIFIER ON

```
CREATE TABLE [dbo].[RegistrosGV](
    [ClaveDia] [nvarchar](20) NOT NULL,
    [Dia] [nvarchar](10) NOT NULL,
    [nomDia] [nvarchar](20) NOT NULL,
    [tGasto] [int] NOT NULL,
    [tVenta] [int] NOT NULL,
    [Balance] [int] NOT NULL,
    CONSTRAINT [PK_RegistrosGV] PRIMARY KEY CLUSTERED
(
    [ClaveDia] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
```


Código Tabla = Distribuidores:

USE [Comedor_Pacifico]

/***** Object: Table [dbo].[Distribuidores] Script Date: 25/06/2022 04:36:23 p. m. *****/

SET ANSI_NULLS ON

SET QUOTED_IDENTIFIER ON

CREATE TABLE [dbo].[Distribuidores](

[nDistribuidor] [int] IDENTITY(1,1) NOT NULL,

[NombreDistribudor] [nvarchar](50) NOT NULL,

[DiasAsistencia] [nvarchar](50) NOT NULL,

[HoraAsistencia] [nvarchar](50) NOT NULL,

[NombreEmpleado] [nvarchar](50) NOT NULL,

[NumeroTelefonico] [nvarchar](20) NULL,

[MontoAprox] [int] NOT NULL,

[nCuenta] [int] NOT NULL,

CONSTRAINT [PK_Distribuidores] PRIMARY KEY CLUSTERED

(

[nDistribuidor] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]

) ON [PRIMARY]

ALTER TABLE [dbo].[Distribuidores] WITH CHECK ADD CONSTRAINT
[FK_Distribuidores_UsuariosComedor] FOREIGN KEY([nCuenta])

REFERENCES [dbo].[UsuariosComedor] ([nCuenta])

ALTER TABLE [dbo].[Distribuidores] CHECK CONSTRAINT [FK_Distribuidores_UsuariosComedor]

Código Tabla = Usuarios:

```
USE [Comedor_Pacifico]
```

```
GO
```

```
/***** Object: Table [dbo].[UsuariosComedor] Script Date: 25/06/2022 04:37:56 p. m. *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[UsuariosComedor](
```

```
    [nCuenta] [int] IDENTITY(1,1) NOT NULL,
```

```
    [NombreUsuario] [nvarchar](30) NOT NULL,
```

```
    [Nombre] [nvarchar](50) NOT NULL,
```

```
    [Apellidos] [nvarchar](50) NOT NULL,
```

```
    [Telefono] [nvarchar](20) NULL,
```

```
    [Rol] [nvarchar](20) NOT NULL,
```

```
    [Year] [int] NOT NULL,
```

```
    [Contrasena] [nvarchar](30) NOT NULL,
```

```
CONSTRAINT [PK_UsuariosComedor] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [nCuenta] ASC
```

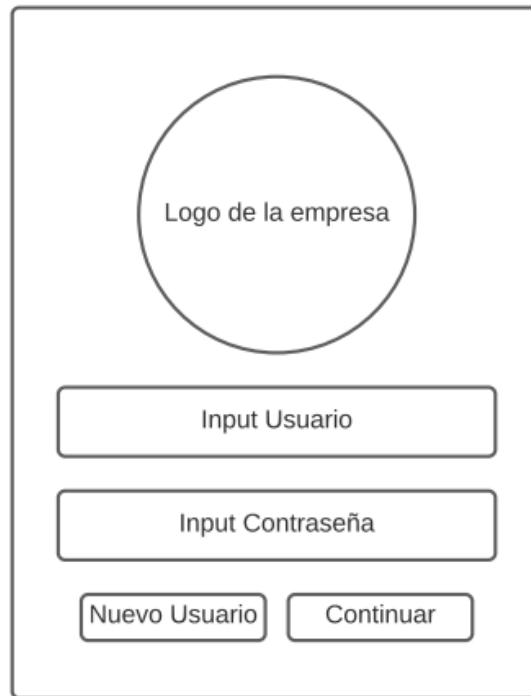
```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)  
ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

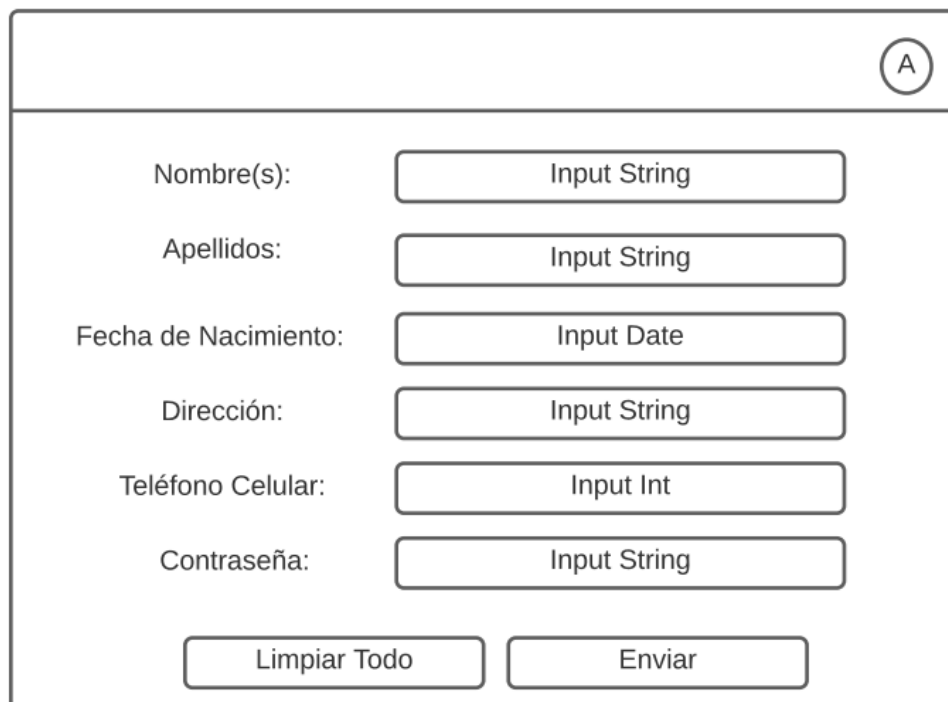
10-INTERFACES(Mock-ups)

Pantalla de Inicio de Sesión:



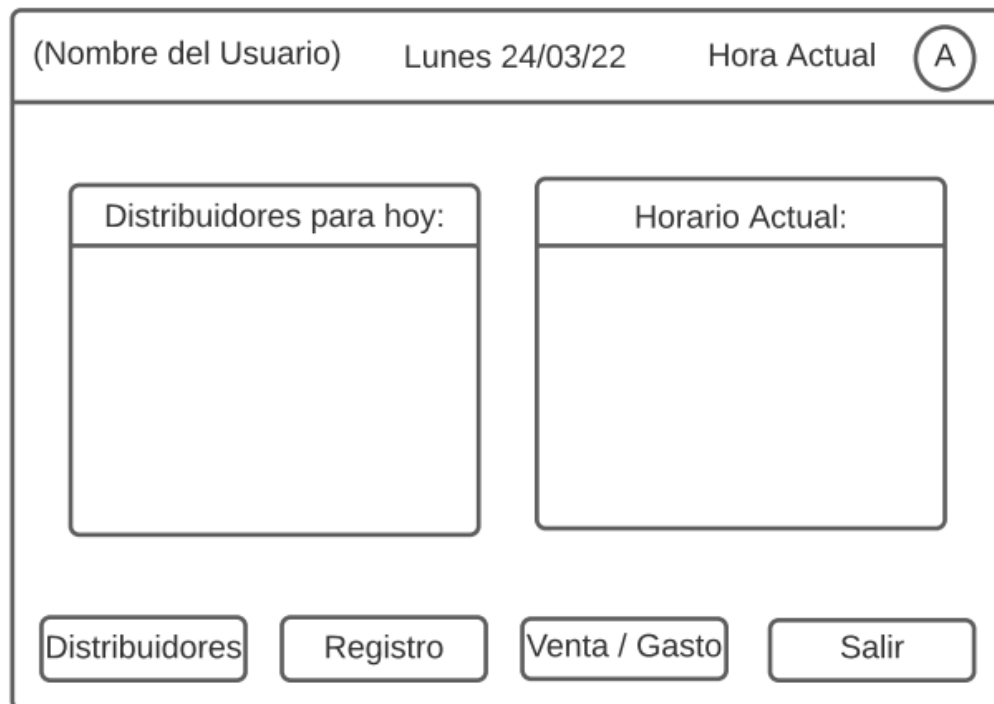
A mockup of a login screen. It features a large circle at the top containing the text "Logo de la empresa". Below the circle are two rectangular input fields: "Input Usuario" and "Input Contraseña". At the bottom, there are two buttons: "Nuevo Usuario" and "Continuar".

Pantalla de Creación de Cuenta:



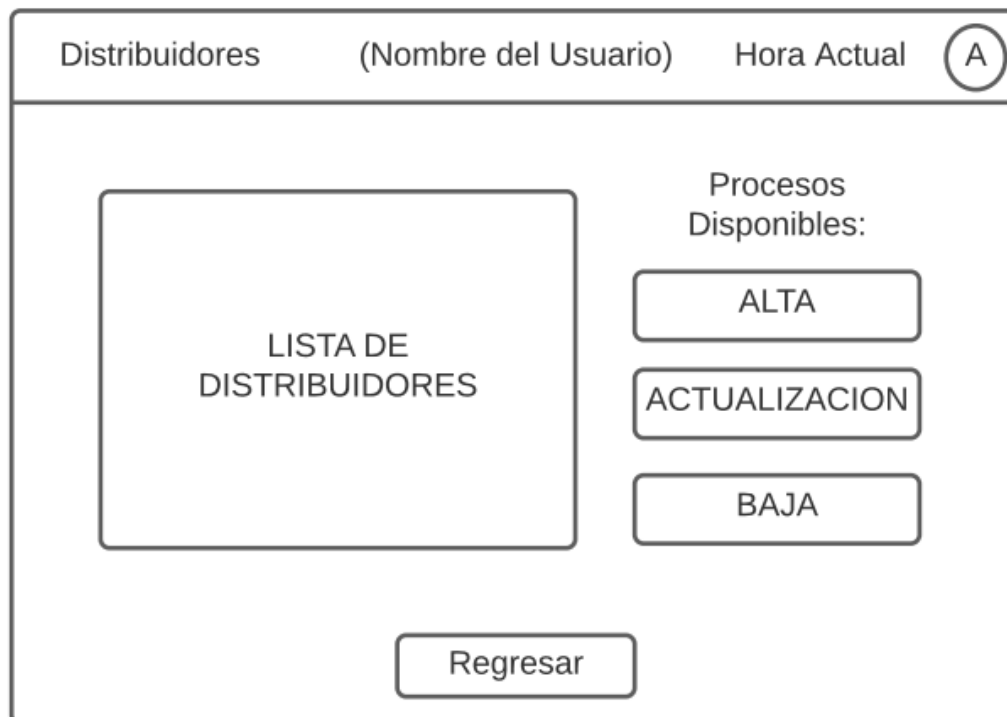
A mockup of a user creation screen. It has a header bar with a circular icon containing the letter "A". Below the header, there are six rows of labels and input fields: "Nombre(s):" with an "Input String" field, "Apellidos:" with an "Input String" field, "Fecha de Nacimiento:" with an "Input Date" field, "Dirección:" with an "Input String" field, "Teléfono Celular:" with an "Input Int" field, and "Contraseña:" with an "Input String" field. At the bottom, there are two buttons: "Limpiar Todo" and "Enviar".

Pantalla Principal de la aplicación:




The main application screen features a header bar with three elements: a placeholder for the user's name '(Nombre del Usuario)', the current date 'Lunes 24/03/22', and the current time 'Hora Actual' next to a circular icon containing the letter 'A'. Below the header, the screen is divided into two columns. The left column has a title 'Distribuidores para hoy:' above a large empty rectangular box. The right column has a title 'Horario Actual:' above another large empty rectangular box. At the bottom of the screen, there is a horizontal row of four buttons: 'Distribuidores', 'Registro', 'Venta / Gasto', and 'Salir'.

Pantalla de Distribuidores:




The distributors screen has a header bar with three elements: a 'Distribuidores' title, a placeholder for the user's name '(Nombre del Usuario)', and the current time 'Hora Actual' next to a circular icon containing the letter 'A'. The main content area is split into two sections. On the left, a large rectangular box is labeled 'LISTA DE DISTRIBUIDORES'. On the right, under the heading 'Procesos Disponibles:', there are three buttons stacked vertically: 'ALTA', 'ACTUALIZACION', and 'BAJA'. At the bottom center of the screen is a 'Regresar' button.

Pantalla de alta de Distribuidores:

Alta de Distribuidores	(Usuario)	24/03/22	
Nombre Distribuidor:	<input type="text" value="Input String"/>		
Dias que asistiura de la semana	<input type="text" value="Input String"/>		
Hora Aproximada que asiste:	<input type="text" value="Input String"/>		
Nombre el empleado:	<input type="text" value="Input String"/>		
Telefono del empleado:	<input type="text" value="Input Int"/>		
Monto aproximado requerido:	<input type="text" value="Input Int"/>		
<input type="button" value="Volver"/>		<input type="button" value="Guardar"/>	

Pantalla de Actualización de Distribuidores:

Actualizacion de Distribuidores	(Usuario)	24/03/22	
Nombre Distribuidor:	<input type="text" value="String (BLOQUEADO)"/>		
Dias que asistiura de la semana	<input type="text" value="Input String"/>		
Hora Aproximada que asiste:	<input type="text" value="Input String"/>		
Nombre el empleado:	<input type="text" value="Input String"/>		
Telefono del empleado:	<input type="text" value="Input Int"/>		
Monto aproximado requerido:	<input type="text" value="Input Int"/>		
<input type="button" value="Volver"/>		<input type="button" value="Guardar"/>	

Pantalla de Registro de Comidas:

Registro de Alimentos

(Usuario)

Lunes 24/03/22

A

Desayunos:

Registro en laptop:

Input

Anotados sin credencial

Input

Anotados ECO Recycling:

Input

Anotados por producto:

Input

Volver

Guardar

Ver Registros

Pantalla Gastos y Ventas:

Gastos y ventas

(Usuario)

Lunes 24/03/22

A

(Gastos / Ventas)

Gastos

Motivo:

Input (char)

Ventas

Cantidad:

Input(int)

Volver

Ver Detalles

Pantalla de Apuntes:

The wireframe for the 'Pantalla de Apuntes' screen features a header bar with the title 'Apuntes', a user identifier '(Usuario)', the date 'Lunes 24/03/22', and a circular profile icon labeled 'A'. The main content area is divided into two sections. On the left, a box titled 'Lista de Apuntes:' contains a large, empty rectangular area for displaying a list of notes. On the right, three buttons are stacked vertically: 'Nuevo', 'Ver / Editar', and 'Eliminar'. At the bottom of the screen, two buttons are positioned side-by-side: 'Volver' and 'Ver Detalles'.

Pantalla de Nuevo Apunte:

The wireframe for the 'Pantalla de Nuevo Apunte' screen has a header bar with the title 'Nuevo Apunte', a user identifier '(Usuario)', the date '24/03/22', and an empty circular profile icon. The main content area contains two input fields. The first is labeled 'Titulo:' and is represented by a rounded rectangular box containing the text 'Input String'. The second is labeled 'Contenido:' and is represented by a larger rounded rectangular box containing the text 'Texto'. At the bottom of the screen, two buttons are positioned side-by-side: 'Volver' and 'Guardar'.

Pantalla de Actualización de Apuntes:

Ver / Editar Apunte

(Usuario)

24/03/22

Titulo:

Input String (BLOQUEADO)

Contenido:

Texto

Volver

Guardar

10.1-INTERFACES DE ENTRADA

Pantalla Inicio de Sesión



The login screen features a central logo with a chef's hat, a blue building, and a fork and spoon. Below the logo are input fields for 'Usuario:' and 'Contraseña:'. The password field has a 'Ver' button. At the bottom are 'Crear Nueva Cuenta' and 'Ingresar' buttons. A red 'X' button is in the top right corner.

Usuario:

Contraseña:

Ver

Crear Nueva Cuenta Ingresar

Pantalla Alta Distribuidor



The 'Nuevo Distribuidor' screen displays user information (MICNOG02) and the date (Mar - 14/Jun/2022). It includes fields for distributor name, attendance days (Lunes to Sábado), arrival time (Hora, Min, Am/Pm), employee name, employee phone, and required amount. 'Volver' and 'Continuar' buttons are at the bottom.

Usuario: MICNOG02 Fecha: Mar - 14/Jun/2022

Nuevo Distribuidor

Nombre del Distribuidor:

Días que asistirá: Lunes Martes Miércoles Jueves Viernes Sábado | Limpiar

Hora Aprox. de Llegada: Hora Min (Am/Pm)


Nombre del Empleado:

Teléfono del Empleado:

Monto Aprox. Requerido:

Volver Continuar

Pantalla Alta Registro



Usuario:
MICNOG02

Registros Comida

Fecha:
Mar - 14/Jun/2022

Opciones de Turno → **Desayunos** **Comida** **Cena**

Registros en Laptop:

Sin Credencial:

ECO Recycling:

Por Producto:

Continuar

Volver

Ver Detalles

Pantalla Alta Gastos y Ventas



Usuario:
MICNOG02

Gastos y Ventas

Fecha:
Mar - 14/Jun/2022

Opciones → **Gasto** **Venta**

Motivo:

Cantidad:

Continuar

Volver

Ver Detalles

Pantalla Alta Apuntes



Usuario:
MICNOG02

Apuntes

Fecha:
Mar - 14/Jun/2022

Nuevo Apunte

Título del apunte:

Texto del Apunte:

¿Editable por todos? ☐

Volver

Continuar

Pantalla Alta Usuario

Crear Nuevo Usuario 

Nombre(s):

Apellidos:

Teléfono:

Rol (Área):

Año de Nacimiento:

Contraseña:

Repetir Contraseña:

Volver a Inicio de Sesión

Crear Cuenta

Pantalla Actualización Distribuidor



Usuario:
MICNOG02

Distribuidores

Fecha:
Mar - 14/Jun/2022

[Ver](#) / [Eliminar](#) / [Editar Distribuidor](#)

Nombre del Distribuidor:

Distribuidor Fruta

Días que asistirá:

Lunes

Martes

Miércoles

Jueves

Viernes

Sábado

|

Limpiar

Hora Aprox. de Llegada:

Hora

12

Min

30

(Am/Pm)

PM

Nombre del Empleado:

Fernando Hinojosa

Teléfono del Empleado:

1231231231

Monto Aprox. Requerido:

19591

Volver

Eliminar

Continuar

Pantalla Actualización Apuntes



Usuario:
MICNOG02

Apuntes

Fecha:
Mar - 14/Jun/2022

[Ver](#) / [Eliminar](#) / [Editar Apunte](#)

Título del apunte:

Pendientes

Texto del Apunte:

Barrer

¿Editable por todos?

☐

Volver

Eliminar

Continuar

10.2-INTERFACES DE SALIDA

Pantalla Lista de Distribuidores

Usuario: MICNOG02

Distribuidores

Fecha: Mar - 14/Jun/2022

Lista de Distribuidores

No tienes Distribuidores, ¡Intenta dar de alta alguno!

Opciones:

Nuevo Distribuidor

← 1/0 → # Distribuidores: 0

Volver

Pantalla Lista de Apuntes

Usuario: EDGOCE04

Apuntes

Fecha: Mié - 15/Jun/2022

Lista de Apuntes

No tienes Apuntes, ¡Intenta dar de alta alguno!

Opciones:

Nuevo Apunte

← 1/0 → # Apuntes: 0

Volver

Pantalla Lista de Gastos



Usuario:
MICNOG02

Gastos y Ventas

Registros del día de hoy:

Gasto		Venta	
nCuenta	Motivo	Monto	Hora
¡No has realizado ningun registro!			

Balance: 0

VolverFecha en Especifico

Pantalla Lista de Ventas



Usuario:
MICNOG02

Gastos y Ventas

Registros del día de hoy:

Gasto		Venta	
nCuenta	Motivo	Monto	Hora
¡No has realizado ningun registro!			

Balance: 0

VolverFecha en Especifico

Pantalla Lista de Registros de Comidas



Usuario: MICNOG02

Registros Comida


Registros del día de Hoy

Fecha: Mié - 15/Jun/2022

Turno	Desayunos	Comidas	Cenas	Totales
ECO	-	-	-	0
Laptop	-	-	-	0
Sin Credencial	-	-	-	0
Por Producto	-	-	-	0
Total	-	-	-	0

VolverFecha en Especifico

Pantalla Principal



Usuario: MICNOG02

Fecha: Mar - 14/Jun/2022

Hora: 05:11 pm ✕

Distribuidores

COCA COLA DISTRIB

08:10 AM

Horario Laboral

07:00 AM

12:00 PM

07:15 AM

12:30 PM

07:30 AM

12:00 PM

08:00 AM

01:00 PM

08:30 AM

01:30 PM

09:00 AM

02:00 PM

10:00 AM

03:00 PM

11:30 AM

03:30 PM

DistribuidoresRegistrosVentas y GastosApuntes

Pantalla Auxiliar Seleccionador de Fecha



Usuario: MICNOG02

Gastos y Ventas

Fecha: Mar - 14/Jun/2022

Seleccione su fecha:

Fecha: 31/05/2022 ▼

Volver Aceptar

11.- HERRAMIENTAS CASE (Computer Aided Software Engineering)

No.	Tipo de Herramienta	Herramienta	Versión
1	Alto Nivel	Microsoft SQL Server	18
2	Alto Nivel	Visual Studio 2022	17.2
3	Medio Nivel	Adobe Photoshop	CS6
4	Bajo Nivel	Google Chrome	101
5	Bajo Nivel	Microsoft Excell	Office 365
6	Bajo Nivel	Microsoft Word	Office 365
7	Bajo Nivel	Microsoft Access	Office 365

Usos:

No #1 (Microsoft SQL Server): Desarrollo y Gestión de las Bases de Datos

No #2 (Visual Studio 2022): Construcción del Programa

No #3 (Adobe Photoshop): Desarrollo de Fondos y Recursos Visuales

No #4 (Google Chrome): Investigación y Accesos a Documentación

No #5 (Microsoft Excel): Desarrollo de análisis de iteraciones

No #6 (Microsoft Word): Desarrollo de Documentación

No #7 (Microsoft Access): Desarrollo del Análisis para la Base de Datos

11.- Polimorfismos

Pantalla Lista de Gastos



Usuario:
MICNOG02

Gastos y Ventas

Fecha:
Mar - 14/Jun/2022

Registros del: 14/06/2022

Gasto **Venta**

nCuenta	Motivo	Monto	Hora
1	Azucar	150	05:00 pm

Balance: 1350

Volver

Fecha en Especifico

Pantalla Lista de Ventas



Usuario:
MICNOG02

Gastos y Ventas

Fecha:
Mar - 14/Jun/2022

Registros del: 14/06/2022

Gasto **Venta**

nCuenta	Motivo	Monto	Hora
1	Cierre Caja	1500	05:01 pm

Balance: 1350

Volver

Fecha en Especifico

Pantalla Lista de Registros de Comidas

 Usuario: MICNOG02

Registros Comida

Fecha: Mar - 14/Jun/2022


Registros del día de Hoy

Turno	Desayunos	Comidas	Cenas	Totales
ECO	1	2	3	6
Laptop	1	2	3	6
Sin Credencial	1	2	3	6
Por Producto	1	2	3	6
Total	4	8	12	24

Volver

Fecha en Especifico

Pantalla Lista de Distribuidores

 Usuario: MICNOG02

Distribuidores

Fecha: Mar - 14/Jun/2022

Lista de Distribuidores

1.- Distribuidor Fruta

2.- COCA COLA DISTRIB

Opciones:

Nuevo Distribuidor

← 1/1 → # Distribuidores: 2

Volver

Pantalla Lista de Apuntes



Usuario:
MICNOG02

Fecha:
Mar - 14/Jun/2022

Apuntes

Lista de Apuntes

1.- Pendientes

←

1/1

→

Apuntes: 1

Opciones:

Nuevo Apunte

Volver

12.- CONCLUSION GENERAL DEL PROYECTO

A lo largo del proyecto pude darme cuenta de la mayoría de las situaciones que establece el cliente tienden a crear planteamientos en los que se tienen que realizar más pasos de los que a simple vista se logran analizar, estos casos, pocas veces podrán ser vistos por el usuario, pero aun así cumplen un rol muy relevante a la hora de trabajar como un programador, es curioso ver como en muchos casos esta programación es incluso más compleja que la programación final que se ha de realizar pues si nosotros como programadores dejamos estos errores libres, podemos generar situaciones de errores fatales, en lo que el proyecto tranquilamente podría quedar inútil.

El empleamiento de las herramientas try catch fue completamente novedoso para mí como programador, dado a que nunca había tenido que contemplar situaciones en las que el usuario era capaz de romper por completo un programa, pero aun así es muy útil empezar a desarrollar con la idea en la mente de que el usuario siempre va a poner nuestro programa en situaciones en las que este puede mal funciona.

Pasando un poco más al trato con la empresa, fue interesante apreciar como todos podemos de alguna forma u otra comunicarnos para lograr lo que la empresa exactamente busca, es un proceso complejo pero si se realiza bien, tendremos un fuerte base de inicio para empezar a trabajar de forma segura, sin tener que pensar si algo que realizamos será del agrado de nuestro cliente, y aunque en algunos casos el paso de la comunicación con el cliente no haya sido muy buena, nosotros debemos de velar por realizar el trabajo lo mejor posible, para que aun así nuestro programa logre satisfacer las expectativas de nuestro cliente.

13.- REFERENCIAS BIBLIOGRAFICAS

- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/developing-apps/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/developing-apps/programming/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/developing-apps/programming/computer-resources/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/developing-apps/programming/drives-directories-files/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/developing-apps/accessing-data>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/developing-apps/creating-and-using-components>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/developing-apps/windows-forms/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/arrays/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/collection-initializers/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/constants-enums/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/events/>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/error-types>
- » <https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/objects-and-classes/>

14.- Anexos

Código del Formulario (Inicio de Sesión):

```
1      Public Class InicioSesion
2      Private Sub ElButton5_Click(sender As Object, e As EventArgs) Handles ElButton5.Click
3      StartUpForm.Close()
4      End Sub
5      Private Sub ElButton3_MouseDown(sender As Object, e As MouseEventArgs) Handles
6      ElButton3.MouseDown
7      txtContrasena.UseSystemPasswordChar = False
8      End Sub
9      Private Sub ElButton3_MouseUp(sender As Object, e As MouseEventArgs) Handles
10     ElButton3.MouseUp
11     txtContrasena.UseSystemPasswordChar = True
12     End Sub
13
14     Private Sub ElButton1_Click(sender As Object, e As EventArgs) Handles ElButton1.Click
15     CrearNuevoUsuario.Show()
16
17     txtContrasena.Text = ""
18     txtUsuario.Text = ""
19
20     Me.Opacity = 0
21     Me.Close()
22     End Sub
23     Private Sub ElButton2_Click(sender As Object, e As EventArgs) Handles ElButton2.Click
24     If txtUsuario.Text = "" Then
25     AlertaTexto.Left = 95
26     AlertaTexto.Text = "    INGRESE UN USUARIO"
27     MarcarError(txtUsuario)
28     Else
29     QuitarError(txtUsuario)
30     End If
31
32     If txtContrasena.Text = "" Then
33     AlertaTexto.Left = 69
34     AlertaTexto.Text = "    INGRESE UNA CONTRASEÑA"
35     MarcarError(txtContrasena)
36     Else
37     QuitarError(txtContrasena)
38     End If
39
40     If txtUsuario.Text = "" And txtContrasena.Text = "" Then
41     AlertaTexto.Left = 100
42     AlertaTexto.Text = "RELLENA TODOS LOS DATOS"
43     ElseIf txtUsuario.Text <> "" And txtContrasena.Text <> "" Then
```

```

42         VerificarUsuario()
43     End If
44
45
46 End Sub
47
48 Public Sub VerificarUsuario()
49     Dim Usuario As String = "NOHAY"
50
51     Try
52         Usuario = Me.UsuariosComedorTableAdapter.BuscarUsuario
53             (txtUsuario.Text)
54
55     Catch ex As Exception
56         Usuario = "NOHAY"
57     End Try
58
59     If Usuario = "NOHAY" Then
60         AlertaTexto.Text = "    USUARIO INEXISTENTE"
61         MarcarError(txtUsuario)
62     Else
63         ModuloExterno.NumeroCuenta = CInt(Usuario)
64         QuitarError(txtUsuario)
65         VerificarContra()
66     End If
67 End Sub
68
69 Public Sub VerificarContra()
70     If Me.UsuariosComedorTableAdapter.BuscarContraseña(txtUsuario.Text) <>
71         txtContraseña.Text Then
72         AlertaTexto.Text = "CONTRASEÑA INCORRECTA"
73         MarcarError(txtContraseña)
74     Else
75         QuitarError(txtContraseña)
76         Arranque()
77     End If
78 End Sub
79
80 Public Sub Arranque()
81     ModuloExterno.NombreUsuario = txtUsuario.Text.ToUpper()
82     PantallaPrincipal.Show()
83     CrearNuevoUsuario.Close()
84     Me.Close()
85 End Sub

```



```

85 'Codigo del Inicio Difuminado
86 Private Sub InicioSesion_Load(sender As Object, e As EventArgs) Handles MyBase.Load
87 'TODO: esta línea de código carga datos en la tabla
    'Comedor_PacificoDataSet.UsuariosComedor' Puede moverla o quitarla según sea necesario.
88 Me.UsuariosComedorTableAdapter.Fill
    (Me.Comedor_PacificoDataSet.UsuariosComedor)
89 ShowFormStart(Me)
90 End Sub
91 'Fin del Codigo de Difuminado
92
93 Private Sub txtUsuario_TextChanged(sender As Object, e As EventArgs) Handles
    txtUsuario.TextChanged
94 QuitarError(txtUsuario)
95 End Sub
96 Private Sub txtContrasena_TextChanged(sender As Object, e As EventArgs) Handles
    txtContrasena.TextChanged
97 QuitarError(txtContrasena)
98 End Sub
99 End Class

```

Código del Formulario (Crear Nuevo Usuario):

```
1      Public Class CrearNuevoUsuario
2      Private Sub ElButton2_Click(sender As Object, e As EventArgs) Handles ElButton2.Click
3      InicioSesion.Show()
4
5          txtNombre.Text = ""
6          txtApellidos.Text = ""
7          txtTelefono.Text = ""
8          txtRol.Text = ""
9          txtContrasena.Text = ""
10         txtOtraContra.Text = ""
11
12         Me.Opacity = 0
13         Me.Close()
14     End Sub
15
16     Private Sub ElButton5_Click(sender As Object, e As EventArgs) Handles ElButton5.Click
17     StartUpForm.Close()
18     End Sub
19
20     Private Sub ElButton1_Click(sender As Object, e As EventArgs) Handles ElButton1.Click
21     Dim Campos() = {txtNombre, txtApellidos, txtTelefono, txtRol, txtYear, txtContrasena,
22                     txtOtraContra}
23
24         Dim flag As Integer = 0
25         For iterador As Integer = 0 To 6
26             If Campos(iterador).Text = "" Then
27                 flag = 1
28                 MarcarError(Campos(iterador))
29             End If
30         Next
31
32         If flag = 0 Then
33             SegundaValidacion()
34         Else
35             MensajeDeError(MessageBox, "Rellene todos los campos", 323)
36         End If
37     End Sub
38
39     Public Sub SegundaValidacion()
40     Dim flag As Integer = 0
41
42         If txtContrasena.Text <> txtOtraContra.Text Then
43             MarcarError(txtContrasena)
44             MarcarError(txtOtraContra)
```

```

44             MensajeDeError(ErrorBox, "Las Contraseñas Deben de Coincidir", 454)
45         flag = 1
46     End If
47
48     If (VerificaInts(txtYear.Text)) Or (txtYear.Text).Length <> 4 Then
49         MarcarError(txtYear)
50         MensajeDeError(ErrorBox, "Ingrese un Año de Nacimiento Válido", 454)
51         flag = 1
52     End If
53
54     If (VerificaInts(txtTelefono.Text)) Or (txtTelefono.Text).Length <> 10 Then
55         MarcarError(txtTelefono)
56         MensajeDeError(ErrorBox, "Ingrese un Número Telefónico Válido", 454)
57         flag = 1
58     End If
59
60     If flag = 0 Then
61         ErrorBox.Text = ""
62         TerceraValidacion()
63     End If
64
65 End Sub
66
67 Public Sub TerceraValidacion()
68     Dim Nombre As String = (txtNombre.Text).Substring(0, 3).ToUpper()
69     Dim Apellido As String = (txtApellidos.Text).Substring(0, 3).ToUpper()
70
71
72     Dim Year As String = (txtYear.Text).Substring(2)
73     Dim User As String = Nombre + Apellido + Year
74     lblUsuario.Text = User
75
76     'Comprobación de cuenta
77
78     Dim existe As String = "NOHAY"
79
80     Try
81         existe = Me.UsuariosComedorTableAdapter.BuscarUsuario
82             (lblUsuario.Text)
83
84     Catch ex As Exception
85         existe = "NOHAY"
86     End Try
87
88     If existe = "NOHAY" Then
89         ModuloExterno.NombreUsuario = User
90         CuartaValidacion()

```

```

89         Else
90             MensajeDeError(ErrorBox, "El usuario con estos valores ya existe", 454)
91         End If
92
93     End Sub
94
95     Public Sub CuartaValidacion()
96         Dim Campos() = {txtNombre, txtApellidos, txtTelefono, txtRol, txtYear, txtContrasena,
97             txtOtraContra}
98         Dim Labels() = {lblNombre, lblApellidos, lblTelefono, lblRol, lblYear, lblContra}
99
100         For iterador As Integer = 0 To 5
101             Labels(iterador).Text = Campos(iterador).Text
102         Next
103
104         PnlTextBox.Visible = False
105         PnlDatos.Visible = True
106         PnlDatos.Left = 292
107         PnlDatos.Top = 95
108
109         ElButton1.Visible = False
110         ElButton2.Visible = False
111
112         ElButton3.Visible = True
113         ElButton4.Visible = True
114     End Sub
115
116     Public Sub Arranque()
117         PantallaPrincipal.Show()
118         InicioSesion.Close()
119         Me.Close()
120     End Sub
121
122     Private Sub CrearNuevoUsuario_Load(sender As Object, e As EventArgs) Handles
123         MyBase.Load
124         'TODO: esta línea de código carga datos en la tabla
125         'Comedor_PacificoDataSet UsuariosComedor' Puede moverla o quitarla según sea necesario.
126         Me.UsuariosComedorTableAdapter.Fill
127             (Me.Comedor_PacificoDataSet.UsuariosComedor)
128
129         ShowFormStart(Me)
130     End Sub
131
132     Private Sub txtNombre_TextChanged(sender As Object, e As EventArgs) Handles
133         txtNombre.TextChanged
134         QuitarError(txtNombre)
135     End Sub

```

```

130 Private Sub txtApellidos_TextChanged(sender As Object, e As EventArgs) Handles
txtApellidos.TextChanged
131 QuitarError(txtApellidos)
132 End Sub
133 Private Sub txtTelefono_TextChanged(sender As Object, e As EventArgs) Handles
txtTelefono.TextChanged
134 QuitarError(txtTelefono)
135 End Sub
136 Private Sub txtRol_TextChanged(sender As Object, e As EventArgs) Handles
txtRol.TextChanged
137 QuitarError(txtRol)
138 End Sub
139 Private Sub txtYear_TextChanged(sender As Object, e As EventArgs) Handles
txtYear.TextChanged
140 QuitarError(txtYear)
141 End Sub
142 Private Sub txtContrasena_TextChanged(sender As Object, e As EventArgs) Handles
txtContrasena.TextChanged
143 QuitarError(txtContrasena)
144 End Sub
145 Private Sub txtOtraContra_TextChanged(sender As Object, e As EventArgs) Handles
txtOtraContra.TextChanged
146 QuitarError(txtOtraContra)
147 End Sub
148
149 Private Sub ElButton4_Click(sender As Object, e As EventArgs) Handles ElButton4.Click
150 ElButton1.Visible = True
151 ElButton2.Visible = True
152
153 ElButton3.Visible = False
154 ElButton4.Visible = False
155
156 PnlTextBox.Visible = True
157 PnlDatos.Visible = False
158 End Sub
159
160 Private Sub ElButton3_Click(sender As Object, e As EventArgs) Handles ElButton3.Click
161 'txtNombre, txtApellidos, txtTelefono, txtRol, txtYear, txtContrasena, txtOtraContra}
162 Me.UsuariosComedorTableAdapter.AltaUsuarioComedor
    (ModuloExterno.NombreUsuario, txtNombre.Text, txtApellidos.Text,
        txtTelefono.Text, txtRol.Text, CInt(txtYear.Text), txtContrasena.Text)
163 Me.UsuariosComedorTableAdapter.Fill
    (Me.Comedor_PacificoDataSet.UsuariosComedor)
164
165 Arranque()
End Sub

```

166

167 End Class

Código del Formulario (Pantalla Principal):

```
1 Public Class PantallaPrincipal
2 Private Sub PantallaPrincipal_Load(sender As Object, e As EventArgs) Handles MyBase.Load
3 'TODO: esta línea de código carga datos en la tabla
4 'Comedor_PacificoDataSet.UsuariosComedor' Puede moverla o quitarla según sea necesario.
5 Me.UsuariosComedorTableAdapter.Fill
6 (Me.Comedor_PacificoDataSet.UsuariosComedor)
7
8 'TODO: esta línea de código carga datos en la tabla
9 'Comedor_PacificoDataSet.Distribuidores' Puede moverla o quitarla según sea necesario.
10 Me.DistribuidoresTableAdapter.Fill
11 (Me.Comedor_PacificoDataSet.Distribuidores)
12
13 'TODO: esta línea de código carga datos en la tabla
14 'Comedor_PacificoDataSet.RegistrosTotales' Puede moverla o quitarla según sea necesario.
15 Me.RegistrosTotalesTableAdapter.Fill
16 (Me.Comedor_PacificoDataSet.RegistrosTotales)
17
18 'TODO: esta línea de código carga datos en la tabla
19 'Comedor_PacificoDataSet.RegistrosGV' Puede moverla o quitarla según sea necesario.
20 Me.RegistrosGVTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegistrosGV) 11
21
22 ModuloExterno.ShowFormStart(Me)
23 ModuloExterno.ShowFormStart(Me)
24 LbUsuario.Text = ModuloExterno.NombreUsuario
25 ModuloExterno.NumeroCuenta = Me.UsuariosComedorTableAdapter.BuscarUsuario
26 (ModuloExterno.NombreUsuario)
27
28
29 timer2.Start()
30 LbHora.Text = getHora()
31 LbFecha.Text = getDia()
32
33 MensajesHorario()
34
35
36 'SetNuevoDia
37 Dim ExisteDia As String = ""
38 Try
39 ExisteDia = Me.RegistrosGVTableAdapter.ExisteDia(getClaveDia())
40 Catch ex As Exception
41 ExisteDia = ""
42 End Try
43
44
45 If ExisteDia = "" Then
46 Me.RegistrosGVTableAdapter.IniciarDia(getClaveDia(), getDiaConLetra
47 (), getDiaEspecifico(), 0, 0, 0)
48
49 Me.RegistrosGVTableAdapter.Fill
50 (Me.Comedor_PacificoDataSet.RegistrosGV)
51
52
53 Me.RegistrosTotalesTableAdapter.InicioDiaRegistros (ModuloExterno.getClaveDia)
```

```

37         Me.RegistrosTotalesTableAdapter.Fill
           (Me.Comedor_PacificoDataSet.RegistrosTotales)
38     End If
39
40     'Inicia el contador de días que
41     Dim díaSemanal() = {"Lunes", "Martes", "Miércoles", "Jueves", "Viernes",
           "Sabado"}
42
43     Dim dia As String = DateTime.Now.ToString("dddd").Substring(0, 1).ToUpper
           () + DateTime.Now.ToString("dddd").Substring(1)
44
45     Dim posicionDelCodigo As Integer
46
47     For iterador As Integer = 0 To 5
48         If díaSemanal(iterador) = dia Then
49             posicionDelCodigo = iterador
50         End If
51     Next
52
53     Dim numeroDeDistribuidores As Integer
54
55     Try
56         numeroDeDistribuidores = Me.DistribuidoresTableAdapter.PerdidaPaquete ()
57     Catch ex As Exception
58         numeroDeDistribuidores = 0
59     End Try
60
61     Dim PsObjetos() = {psObjeto1, psObjeto2, psObjeto3, psObjeto4, psObjeto5, psObjeto6, psObjeto7,
           psObjeto8}
62     Dim PsCasillas() = {Hora1, Hora2, Hora3, Hora4, Hora5, Hora6, Hora7, Hora8}
63
64     If numeroDeDistribuidores <> 0 Then
65         Dim arregloSinPerdidas As New ArrayList
66         arregloSinPerdidas.Clear()
67
68         For NeoIterador As Integer = 1 To numeroDeDistribuidores
69             Dim cadenaSemanal As String
70             cadenaSemanal = Me.DistribuidoresTableAdapter.getAsistencia
           (NeoIterador)
71
72             If cadenaSemanal <> "" Then
73                 Dim NeoArray() = cadenaSemanal.Split("-")
74                 If NeoArray(posicionDelCodigo) = 1 Then 75             'Agregar a la lista semanal
75
76                     arregloSinPerdidas.Add(NeoIterador)
77                 End If
78             End If

```



```

79             Next
80
81             'Mandar el arreglo a ordenar
82             For NeoIterador As Integer = 0 To arregloSinPerdidas.Count() - 2
83                 For IteradorEspecifico As Integer = 0 To
(arregloSinPerdidas.Count() - 2)
84
85                     If CodificaHorarios(arregloSinPerdidas.Item
(IteradorEspecifico)) > CodificaHorarios
(arregloSinPerdidas.Item(IteradorEspecifico + 1)) Then
86
87                         Dim temporal As Integer = arregloSinPerdidas.Item
(IteradorEspecifico)
88
89                         arregloSinPerdidas.Item(IteradorEspecifico) =
arregloSinPerdidas.Item(IteradorEspecifico + 1)
90
91                         arregloSinPerdidas.Item(IteradorEspecifico + 1) =
temporal
92
93                     End If
94                 Next
95             Next
96
97             'Inicio Mostrar en Display
98
99             For IteradorDisplay = 0 To 7
100
101                 If (IteradorDisplay + 1) > arregloSinPerdidas.Count Then
102                     PsObjetos(IteradorDisplay).Visible = False
103                     PsCasillas(IteradorDisplay).Visible = False
104                 Else
105
106                     Dim Distribuidor =
Me.DistribuidoresTableAdapter.getNombreDistribuidor
(arregloSinPerdidas.Item(IteradorDisplay))
107                     Dim Monto As String = Me.DistribuidoresTableAdapter.getMonto
(arregloSinPerdidas.Item(IteradorDisplay))
108                     Dim Nombre As String =
Me.DistribuidoresTableAdapter.getNombreEmpleado
(arregloSinPerdidas.Item(IteradorDisplay))
109                     Dim Telefono As String =
Me.DistribuidoresTableAdapter.getNumeroTelefonico
(arregloSinPerdidas.Item(IteradorDisplay))
110                     PsObjetos(IteradorDisplay).Text = Distribuidor
111                     PsCasillas(IteradorDisplay).Text =
Me.DistribuidoresTableAdapter.getHoraAsistencia
(arregloSinPerdidas.Item(IteradorDisplay))
112
113             Next
114
115         End Sub
116
117     End Class

```

```

109
110         PsObjetos(IteradorDisplay).AccessibleDescription = "Datos del
            Distribuidor: " + Distribuidor + Chr(13) + Chr(13) +
            "Nombre del Empleado: " + Nombre + Chr(13) + "Número Telefónico: " + Telefono +
            Chr(13) + Chr(13) + "Monto Requerido: " + Monto + " $"
111     End If
112     Next
113     psObjeto33.visible = False
114     Else
115     For NeoIterador As Integer = 0 To 7
116     PsObjetos(NeoIterador).Visible = False
117     PsCasillas(NeoIterador).Visible = False
118     Next
119
120     psObjeto33.visible = True
121     End If
122
123 End Sub
124
125 Public Function CodificaHorarios(ByVal ValorBusqueda As Integer) 126
127
128         Dim Cadena As String = Me.DistribuidoresTableAdapter.getHoraAsistencia
            (ValorBusqueda)
129
130         Dim strHora As String = Cadena.Substring(0, 2)
131         Dim strMin As String = Cadena.Substring(3, 2)
132         Dim strMeridiano As String = Cadena.Substring(6, 2) 132
133         Dim resultado As Integer = 0
134
135         If strMeridiano = "AM" Then
136             resultado = Integer.Parse(strHora) * 60 + Integer.Parse(strMin)
137         Else
138             resultado = Integer.Parse(strHora) * 60 + Integer.Parse(strMin) + 720
139         End If
140
141     Return resultado
142 End Function
143
144 Private Sub ElButton5_Click(sender As Object, e As EventArgs) Handles ElButton5.Click
145     StartUpForm.Close()
146 End Sub
147
148 Private WithEvents timer2 As New Windows.Forms.Timer With {.Interval = 1000}
149 Private Sub timer2_Tick(sender As Object, e As EventArgs) Handles timer2.Tick
150     LbHora.Text = ModuloExterno.getHora()
151 End Sub
152
153 Private Sub ElButton1_Click(sender As Object, e As EventArgs) Handles ElButton1.Click
154     Distribuidores.Show()

```

```

155     Me.Close()
156 End Sub
157
158 Private Sub ElButton3_Click(sender As Object, e As EventArgs) Handles ElButton3.Click
159     Registros.Show()
160     Me.Close()
161 End Sub
162
163 Private Sub ElButton2_Click(sender As Object, e As EventArgs) Handles ElButton2.Click
164     VentasGastos.Show()
165     Me.Close()
166 End Sub
167
168 Private Sub ElButton4_Click(sender As Object, e As EventArgs) Handles ElButton4.Click
169     Apuntes.Show()
170     Me.Close()
171 End Sub
172
173 Private Sub PantallaPrincipal_Closed(sender As Object, e As EventArgs) Handles Me.Closed
174     Me.Opacity = 0
175 End Sub
176
177 Private Sub RegistrosGVBindingNavigatorSaveItem_Click(sender As Object, e As EventArgs)
178     Me.Validate()
179     Me.RegistrosGVBindingSource.EndEdit()
180     Me.TableAdapterManager.UpdateAll(Me.Comedor_PacificoDataSet) 181
182 End Sub
183
184 Private Sub MuestraPropiedades(sender As Object, e As EventArgs) Handles psObjeto1.Click,
    psObjeto2.Click, psObjeto3.Click, psObjeto4.Click, psObjeto5.Click, psObjeto6.Click, psObjeto7.Click,
    psObjeto8.Click,
    BTN1.Click, BTN2.Click, BTN3.Click, BTN4.Click, BTN5.Click, BTN6.Click,
    BTN7.Click, BTN8.Click, BTN9.Click, BTN10.Click, BTN11.Click, BTN12.Click, BTN13.Click,
    BTN14.Click, BTN15.Click, BTN16.Click
185     MsgBox(sender.AccessibleDescription)
186 End Sub
187 Private Sub MensajesHorario()
188     Dim Cajitas() = {BTN1, BTN2, BTN3, BTN4, BTN5, BTN6, BTN7, BTN8, BTN9, BTN10, BTN11,
    BTN12, BTN13, BTN14, BTN15, BTN16}
189     Dim Descripciones(15) As String
190
191     Descripciones(0) = "Inicio del Día: " + Chr(13) + Chr(13) + "Abrir Caja" + Chr(13) +
    "Limpiar Mesa de Trabajo" + Chr(13) + "Rellenar Refrigeradores" + Chr(13) + "Preparar Ordenes
    RH"
192     Descripciones(15) = "Final del Día:" + Chr(13) + Chr(13) + "Rellena
    Refrigeradores" + Chr(13) + "Inicia Cierre de Caja" + Chr(13) +
    "Limpieza General de Área de Trabajo."
193     For NeoIterador As Integer = 0 To 15
194         Cajitas(NeoIterador).AccessibleDescription = Descripciones (NeoIterador)
195     Next
196 End Sub

```

197 End Class

Código del Formulario (Distribuidores):

```
1 Public Class Distribuidores
2
3     Public Hojas As Integer = 0
4     Public HojaActual As Integer = 1
5     Public PseudoHoja As Integer = 0
6
7         Public DistribuidorSeleccionado As Integer = 0
8
9     Private Sub Distribuidores_Load(sender As Object, e As EventArgs) Handles MyBase.Load
10         'TODO: esta línea de código carga datos en la tabla
11         'Comedor_PacificoDataSet.Distribuidores' Puede moverla o quitarla según sea necesario.
12         Me.DistribuidoresTableAdapter.Fill
13         (Me.Comedor_PacificoDataSet.Distribuidores)
14         ShowFormStart(Me)
15         LbFecha.Text = getDia()
16
17         Usuario.Text = ModuloExterno.NombreUsuario
18
19         Despliegue()
20     End Sub
21
22     Private Sub Distribuidores_Closed(sender As Object, e As EventArgs) Handles Me.Closed
23         Me.Opacity = 0
24     End Sub
25
26     Private Sub ElButton23_Click(sender As Object, e As EventArgs)
27         PnlLista.Visible = False
28         PnlNuevo.Visible = True
29
30         PnlLista.Left = 44
31         PnlLista.Top = 110
32     End Sub
33
34     Private Sub ElButton4_Click(sender As Object, e As EventArgs) Handles ElButton4.Click
35         'Me.Opacity = 0
36         Label20.Text = "Nuevo Distribuidor"
37
38         txtNombreDistr.Text = ""
39         txtNombreDistr.BackgroundColor = Color.White
40         txtNombreDistr.TextColour = Color.Black
41
42         txtNombreDistr.Enabled = True
43
44         BtnEliminar.Visible = False
45         BtnEliminar.Enabled = False
46
47         PnlLista.Visible = False
```

```

48     PnlNuevo.Visible = True
49
50     PnlNuevo.Left = 44
51     PnlNuevo.Top = 110
52     'ModuloExterno.ShowFormStart(Me)
53 End Sub
54
55 Private Sub ElButton24_Click(sender As Object, e As EventArgs) Handles ElButton24.Click
56     LimpiarInputs()
57
58     PnlNuevo.Visible = False
59     PnlLista.Visible = True
60
61     BtnEliminar.Enabled = False
62     BtnEliminar.Visible = False
63
64     PnlLista.Left = 44
65     PnlLista.Top = 110
66 End Sub
67
68     Private Sub ElButton3_Click(sender As Object, e As EventArgs)
69         Label20.Text = "Editar Distribuidor"
70
71         txtNombreDistr.Text = "NombreDistribuidor"
72         txtNombreDistr.BackgroundColor = Color.Gainsboro
73         txtNombreDistr.TextColour = Color.White
74
75         txtNombreDistr.Enabled = False
76
77
78     PnlNuevo.Visible = True
79     PnlLista.Visible = False
80
81     PnlNuevo.Left = 44
82     PnlNuevo.Top = 110
83 End Sub
84
85 'Btn del Dia
86 Private Sub ActivarBoton(obj As Object, e As EventArgs) Handles
    BtnLunes.Click, BtnMartes.Click, BtnMiercoles.Click, BtnMiercoles.Click, BtnJueves.Click,
    BtnViernes.Click, BtnSabado.Click
87     PermaActiva(obj)
88 End Sub
89
90 Private Sub ElButton14_Click(sender As Object, e As EventArgs) Handles ElButton14.Click
91     Dim Botones() = {BtnLunes, BtnMartes, BtnMiercoles, BtnJueves, BtnViernes, BtnSabado}
92     For iterador As Integer = 0 To 5
93         Reinicia(Botones(iterador))
94     Next
95 End Sub

```

```

96
97 Private Sub ElButton7_Click(sender As Object, e As EventArgs) Handles ElButton7.Click
98 Dim flag As Integer = 0
99 Dim Campos() = {txtNombreDistr, txtHora, txtMin, txtMeridiano, txtNombreEmpleado, txtTelefono,
txtMonto}
100
101 For iterador As Integer = 0 To 6
102 If (Campos(iterador).Text = "") Then
103 MarcarError(Campos(iterador))
104 flag = 1
105 End If
106 Next
107
108 If flag Then
109 MensajeDeError(ErrorBox, "Rellena todos los campos", 380)
110 Else
111 SegundaVerificacion()
112 End If
113 End Sub
114
115 Public Sub SegundaVerificacion()
116 Dim Flag As Boolean = False
117
118 If VerificaInts(txtMonto.Text) Then
119 MarcarError(txtMonto)
120 MensajeDeError(ErrorBox, "Los Valores deben de ser numéricos", 460)
121 Flag = True
122 ElseIf Integer.Parse(txtMonto.Text) <= 0 Then
123 MarcarError(txtMonto)
124 MensajeDeError(ErrorBox, "Monto debe de ser mayor a 0", 400)
125 Flag = True
126 End If
127
128 If VerificaInts(txtTelefono.Text) Then
129 MarcarError(txtTelefono)
130 MensajeDeError(ErrorBox, "Los Valores deben de ser numéricos", 460)
131 Flag = True
132 ElseIf (txtTelefono.Text).Length = 9 Then
133 MarcarError(txtTelefono)
134 MensajeDeError(ErrorBox, "El teléfono debe de ser de 10 dígitos", 490)
135 Flag = True
136 End If
137
138 If (txtNombreEmpleado.Text).Length <= 10 Then
139 MarcarError(txtNombreEmpleado)

```

```

140 MensajeDeError(MessageBox, "El Nombre del Empleado debe de ser mayor a 10 caracteres", 760)
141     Flag = True
142     End If
143
144     If Not (((txtMeridiano.Text).ToUpper() = "AM") Or
145 ((txtMeridiano.Text).ToUpper() = "PM")) Then
146         MarcarError(txtMeridiano)
147         146 MensajeDeError(MessageBox, "El formato del meridiano debe de ser AM o PM", 600)
148         Flag = True
149         End If
150
151     If VerificaInts(txtMin.Text) Then
152         MarcarError(txtMin)
153         MensajeDeError(MessageBox, "Los Valores deben de ser numéricos", 460)
154         Flag = True
155         ElseIf Integer.Parse(txtMin.Text) <= 0 Or Integer.Parse(txtMin.Text) >= 60 Then
156             MarcarError(txtMin)
157             156 MensajeDeError(MessageBox, "Los minutos deben de ser mayor a 0 y menor a 60", 610)
158             Flag = True
159             End If
160
161     If VerificaInts(txtHora.Text) Then
162         MarcarError(txtHora)
163         MensajeDeError(MessageBox, "Los Valores deben de ser numéricos", 460)
164         Flag = True
165         ElseIf Integer.Parse(txtHora.Text) <= 0 Or Integer.Parse(txtHora.Text) >= 13 Then
166             MarcarError(txtHora)
167             166 MensajeDeError(MessageBox, "La hora debe ser mayor que 0 y menor que 12", 540)
168             Flag = True
169             End If
170
171     If (txtNombreDistr.Text).Length <= 7 Then
172         171 MensajeDeError(MessageBox, "El Nombre del distribuidor debe de ser mayor a 7
173 caracteres", 800)
174         MarcarError(txtNombreDistr)
175         Flag = True
176         End If
177
178     If Not Flag Then
179         If VerificarDiasSemana() Then
180             'Proceso de Confirmación
181             CreadorDeStringSemanal()
182             MessageBox.Text = ""
183             MessageBoxDias.Text = ""
184             Confirmacion()
185             Else
186             MessageBoxDias.Text = "ERROR"
187             MensajeDeError(MessageBox, "Seleccione por lo menos un día", 460)

```



```

187             End If
188
189         End If
190     End Sub
191
192 Public Sub Confirmacion()
193     Dim Campos() = {txtNombreDistr, txtNombreEmpleado, txtTelefono, txtMonto}
194     Dim Labels() = {LblNombreDist, LblNombreEmpl, LblTelefono, LblMonto}
195
196     For iterador As Integer = 0 To 3
197         Labels(iterador).Text = Campos(iterador).Text
198     Next
199
200     txtHora.Text = VerificaCeros(txtHora.Text)
201     txtMin.Text = VerificaCeros(txtMin.Text)
202
203     LblHoraFinal.Text = txtHora.Text + ":" + txtMin.Text + " " +
        (txtMeridiano.Text).ToUpper()
204
205     If Label20.Text = "Nuevo Distribuidor" Then
206         LblTipoRegistro.Text = "Nuevo Distribuidor"
207     Else
208         LblTipoRegistro.Text = "Edición Distribuidor"
209     End If
210
211     'Convierte de array a texto
212     LblDiasSemana.Text = CreadorDeStringSemanalTextual() 213
214     PnlConfirmacion.Visible = True
215     PnlNuevo.Visible = False
216
217     PnlConfirmacion.Left = 44
218     PnlConfirmacion.Top = 110
219
220     End Sub
221     Public Function VerificaCeros(Verificar As String)
222         Dim NumeroCreado As Integer = Integer.Parse(Verificar)
223         If (NumeroCreado <= 9) Then
224             Return "0" + NumeroCreado.ToString()
225         End If
226
227     Return Verificar
228 End Function
229
230 Public Function CreadorDeStringSemanalTextual()
231     Dim DiasSemanales As String = ""
232
233     Dim Botones() = {BtnLunes, BtnMartes, BtnMiercoles, BtnJueves, BtnViernes, BtnSabado}
234     Dim Dias() As String = {"Lunes", "Martes", "Miércoles", "Jueves",
        "Viernes", "Sabado"}

```

```

235
236         Dim PuntoFinal As Integer = -1
237
238         For iterador As Integer = 0 To 5
239             If Botones(iterador).Enabled = False Then
240                 PuntoFinal = PuntoFinal + 1
241             End If
242         Next
243
244         Dim ContadorPalPunto = 0
245
246         For iterador As Integer = 0 To 5
247             If Botones(iterador).Enabled = False Then
248                 If ContadorPalPunto <> PuntoFinal Then
249                     DiasSemanales += Dias(iterador) + ", "
250                 Else
251                     DiasSemanales += Dias(iterador) + "."
252                 End If
253
254                 ContadorPalPunto += 1
255
256             End If
257         Next
258
259         Return DiasSemanales
260     End Function
261
262     Public Function VerificarDiasSemana()
263         Dim Botones() = {BtnLunes, BtnMartes, BtnMiercoles, BtnJueves, BtnViernes, BtnSabado}
264         Dim Flag As Integer = 0
265
266         For iterador As Integer = 0 To 5
267             If Botones(iterador).Enabled = False Then
268                 Flag = 1
269             End If
270         Next
271
272         Return Flag
273     End Function
274
275     Public Function CreadorDeStringSemanal()
276         Dim Botones() = {BtnLunes, BtnMartes, BtnMiercoles, BtnJueves, BtnViernes, BtnSabado}
277         Dim Cadena As String = ""
278         For iterador As Integer = 0 To 5
279             If Botones(iterador).Enabled = False Then
280                 Cadena += "1-"
281             Else
282                 Cadena += "0-"
283             End If

```

```

284             Next
285
286         Return Cadena
287     End Function
288
289     'Btn Cambio Texto
290     Private Sub txtNombreDistr_TextChanged(obj As Object, e As EventArgs) Handles
        txtNombreDistr.TextChanged, txtHora.TextChanged, txtMin.TextChanged,
        txtMeridiano.TextChanged, txtNombreEmpleado.TextChanged,
        txtTelefono.TextChanged, txtMonto.TextChanged
291         QuitarError(obj)
292     End Sub
293
294     Private Sub ElButton8_Click(sender As Object, e As EventArgs) Handles ElButton8.Click
295         PnlConfirmacion.Visible = False
296         PnlNuevo.Visible = True
297     End Sub
298
299     Private Sub ElButton16_Click(sender As Object, e As EventArgs) Handles ElButton16.Click
300         PnlConfirmacion.Visible = False
301         PnlNuevo.Visible = True
302         'Aqui se da de alta el distribuidor
303         Dim NumeroDistribuidor As Integer = -1
304         If Label20.Text = "Ver / Eliminar / Editar Distribuidor" Then
305             'Actualizacion
306
307             Me.TableAdapterManager.DistribuidoresTableAdapter.ActualizarDistribuidor
        (txtNombreDistr.Text, CreadorDeStringSemanal(), LblHoraFinal.Text,
        LblNombreEmpl.Text, LblTelefono.Text, LblMonto.Text, ModuloExterno.NumeroCuenta,
        DistribuidorSeleccionado)
308
309         Else
310             'Nuevo
311             Try
312                 NumeroDistribuidor =
                    Me.TableAdapterManager.DistribuidoresTableAdapter.ExisteDistrib
                        udor(txtNombreDistr.Text)
313             Catch ex As Exception
314                 NumeroDistribuidor = -1
315             End Try
316
317             If NumeroDistribuidor = -1 Then
318                 Me.TableAdapterManager.DistribuidoresTableAdapter.AltaDistribuidor (txtNombreDistr.Text,
                    CreadorDeStringSemanal(), LblHoraFinal.Text,
                    LblNombreEmpl.Text, LblTelefono.Text, LblMonto.Text, ModuloExterno.NumeroCuenta)
319                 LimpiarInputs()
320                 CrearPseudoArreglo()

```

```

321         Despliegue()
322     Else
323         MensajeDeError(ErrorBox, "El Nombre del distribuidor ya se encuentra ocupado", 780)
324     End If
325 End If
326
327 End Sub
328
329 Public Sub LimpiarInputs()
330     Dim Campos() = {txtNombreDistr, txtHora, txtMin, txtMeridiano, txtNombreEmpleado, txtTelefono,
331                     txtMonto}
332     For iterador As Integer = 0 To 6
333         Campos(iterador).Text = ""
334     Next
335
336     Dim Botones() = {BtnLunes, BtnMartes, BtnMiercoles, BtnJueves, BtnViernes, BtnSabado}
337     For iterador As Integer = 0 To 5
338         Reinicia(Botones(iterador))
339     Next
340 End Sub
341
342 Private Sub ElButton9_Click(sender As Object, e As EventArgs) Handles ElButton9.Click
343     PantallaPrincipal.Show()
344     Me.Close()
345 End Sub
346
347
348 Private Sub panelHojas_TextChanged(sender As Object, e As EventArgs) Handles
349     panelHojas.TextChanged
350     If HojaActual = Hojas Then
351         nextHoja.Enabled = False
352     Else
353         nextHoja.Enabled = True
354     End If
355
356     If HojaActual = 1 Then
357         backHoja.Enabled = False
358     Else
359         backHoja.Enabled = True
360     End If
361 End Sub
362
363 Dim tDistribuidores As Double = 0
364 Public Sub Despliegue()
365     HojaActual = 1
366     tDistribuidores =
367     Me.TableAdapterManager.DistribuidoresTableAdapter.nDistribuidores()

```

```

367
368         If tDistribuidores = 0 Then
369             Dim Casillas = {psObjeto1, psObjeto2, psObjeto3, psObjeto4, psObjeto5, psObjeto6}
370
371             For iteradorReal As Integer = 0 To 5
372
373                 Casillas(iteradorReal).Visible = False
374                 Casillas(iteradorReal).Enabled = False
375
376             Next
377             psObjeto33.Visible = True
378             Else
379             psObjeto33.Visible = False
380             End If
381
382 nDistribuidores.Text = "# Distribuidores: " + tDistribuidores.ToString() 383
384     Hojas = Math.Ceiling(tDistribuidores / 6)
385     PseudoHoja = Hojas
386
387     panelHojas.Text = "1/" + Hojas.ToString()
388
389     If HojaActual = Hojas Then
390         nextHoja.Enabled = False
391     Else
392         nextHoja.Enabled = True
393     End If
394
395     CrearPseudoArreglo()
396     MostrarElementos()
397 End Sub
398
399 Private Sub nextHoja_Click(sender As Object, e As EventArgs) Handles nextHoja.Click
400     HojaActual = HojaActual + 1
401     panelHojas.Text = HojaActual.ToString() + "/" + Hojas.ToString()
402     PseudoHoja = PseudoHoja - 1
403     MostrarElementos()
404 End Sub
405
406 Private Sub backHoja_Click(sender As Object, e As EventArgs) Handles backHoja.Click
407     HojaActual = HojaActual - 1
408     panelHojas.Text = HojaActual.ToString() + "/" + Hojas.ToString()
409     PseudoHoja = PseudoHoja + 1

```

```

410     MostrarElementos()
411 End Sub
412
413     Dim PaquetesPerdidos As Integer = 0
414
415     Dim arregloSinPerdidas As New ArrayList
416
417 Public Sub CrearPseudoArreglo()
418     ObtenerPerdidaDePaquetes()
419
420     arregloSinPerdidas.Clear()
421
422     Dim pseudoIterador = 0
423     Dim ContadorCasilla = 0
424
425
426     If tDistribuidores <> 0 Then
427     For iteradorReal As Integer = 1 To PaquetesPerdidos + tDistribuidores 428
429         Dim na As String =
430             Me.TableAdapterManager.DistribuidoresTableAdapter.getNombreDist
431             ribuidor(iteradorReal)
432
433         If na <> "" Then
434             arregloSinPerdidas.Add(iteradorReal)
435         End If
436     Next
437     End If
438
439 End Sub
440
441 Public Sub MostrarElementos()
442
443     Dim iteradorBase As Integer = (6 * (PseudoHoja - 1)) + 1
444     'Fin
445     Dim Casillas = {psObjeto1, psObjeto2, psObjeto3, psObjeto4, psObjeto5, psObjeto6}
446
447     If tDistribuidores <> 0 Then
448     For iteradorReal As Integer = 0 To 5
449
450         'Se acabo el ultimo registro
451         If (iteradorBase + iteradorReal - 1) >= arregloSinPerdidas.Count
452             Then
453                 For neoIterador = iteradorReal To 5
454                     Casillas(neoIterador).Visible = False
455                     Casillas(neoIterador).Enabled = False
456                 Next
457                 iteradorReal = 5
458                 Else
459                 Dim nombre As String =

```

```

Me.TableAdapterManager.DistribuidoresTableAdapter.getNombreD
istribuidor(arregloSinPerdidas.Item(iteradorBase + iteradorReal - 1))
459
460 Casillas(iteradorReal).Visible = True
461 Casillas(iteradorReal).Enabled = True
462 Casillas(iteradorReal).AccessibleName =
arregloSinPerdidas.Item(iteradorBase + iteradorReal - 1)
463 Casillas(iteradorReal).Text = (iteradorBase +
iteradorReal).ToString() + ".- " + nombre
464 End If
465 Next
466 End If
467 End Sub
468
469 Public Sub ObtenerPerdidaDePaquetes()
470 Dim LimitePaquete As Integer = 0
471
472 Try
473 LimitePaquete =
Me.TableAdapterManager.DistribuidoresTableAdapter.PerdidaPaquete()
474
475 Catch ex As Exception
476 LimitePaquete = 0
477 End Try
478
479 If LimitePaquete <> 0 Then
480
481 For BuscadorPerdidos As Integer = 1 To LimitePaquete
482 Dim Hueco As String =
Me.TableAdapterManager.DistribuidoresTableAdapter.getNombreDist
ribuidor(BuscadorPerdidos)
483 If Hueco = "" Then
484 PaquetesPerdidos += 1
485 End If
486 Next
487 End If
488
489
490 End Sub
491
492 Private Sub LanzaVerEditar(sen As Object, e As EventArgs) Handles psObjeto1.Click, psObjeto2.Click,
psObjeto6.Click, psObjeto5.Click, psObjeto4.Click, psObjeto3.Click
493 DistribuidorSeleccionado = Integer.Parse(sen.AccessibleName) 494
495 Label20.Text = "Ver / Eliminar / Editar Distribuidor" 496
497 BtnEliminar.Enabled = True
498 BtnEliminar.Visible = True
499
500 txtNombreDistr.Text =
Me.TableAdapterManager.DistribuidoresTableAdapter.getNombreDistribuidor
(DistribuidorSeleccionado)
501 txtNombreDistr.BackgroundColor = Color.Gainsboro

```

```

502     txtNombreDistr.TextColour = Color.White
503
504     procesadorDePseudoCadena
505     (Me.TableAdapterManager.DistribuidoresTableAdapter.getAsistencia
506     (DistribuidorSeleccionado))
507
508     Dim NeoCadena As String =
509     Me.TableAdapterManager.DistribuidoresTableAdapter.getHoraAsistencia
510     (DistribuidorSeleccionado)
511
512     txtHora.Text = NeoCadena.Substring(0, 2)
513     txtMin.Text = NeoCadena.Substring(3, 2)
514     txtMeridiano.Text = NeoCadena.Substring(6, 2)
515     txtNombreEmpleado.Text =
516     Me.TableAdapterManager.DistribuidoresTableAdapter.getNombreEmpleado (DistribuidorSeleccionado)
517     txtTelefono.Text =
518     Me.TableAdapterManager.DistribuidoresTableAdapter.getNumeroTelefonico
519     (DistribuidorSeleccionado)
520     txtMonto.Text =
521     Me.TableAdapterManager.DistribuidoresTableAdapter.getMonto
522     (DistribuidorSeleccionado)
523
524     txtNombreDistr.Enabled = False
525
526     PnlNuevo.Visible = True
527     PnlLista.Visible = False
528
529     PnlNuevo.Left = 44
530     PnlNuevo.Top = 110
531     End Sub
532
533     Public Sub procesadorDePseudoCadena(Cadena As String)
534     Dim NeoArray() = Cadena.Split("-")
535
536     Dim Botones() = {BtnLunes, BtnMartes, BtnMiercoles, BtnJueves, BtnViernes, BtnSabado}
537
538     For iterador As Integer = 0 To 5
539     If NeoArray(iterador) = "1" Then
540     Botones(iterador).Enabled = False
541     Else
542     Botones(iterador).Enabled = True
543     End If
544     Next
545     End Sub
546
547     Private Sub BtnEliminar_Click(sender As Object, e As EventArgs) Handles BtnEliminar.Click
548     Me.TableAdapterManager.DistribuidoresTableAdapter.BorrarDistribuidor (DistribuidorSeleccionado)
549     Me.DistribuidoresTableAdapter.Fill
550     (Me.Comedor_PacificoDataSet.Distribuidores)
551
552

```



```
543         CrearPseudoArreglo()
544         Despliegue()
545         LimpiarInputs()
546
547         PnlNuevo.Visible = False
548         PnlConfirmacion.Visible = False
549         PnlLista.Visible = True
550
551         BtnEliminar.Enabled = False
552         BtnEliminar.Visible = False
553
554         PnlLista.Left = 44
555         PnlLista.Top = 110
556     End Sub
557 End Class
```

Código del Formulario (Registros):

```
1      Public Class Registros
2      Private Sub ElButton1_Click(sender As Object, e As EventArgs) Handles ElButton1.Click
3      PantallaPrincipal.Show()
4      Me.Close()
5      End Sub
6
7      Private Sub Registros_Load(sender As Object, e As EventArgs) Handles MyBase.Load
8      'TODO: esta línea de código carga datos en la tabla
9      'Comedor_PacificoDataSet.RegistrosTotales' Puede moverla o quitarla según sea necesario.
10     Me.RegistrosTotalesTableAdapter.Fill
11     (Me.Comedor_PacificoDataSet.RegistrosTotales)
12     'TODO: esta línea de código carga datos en la tabla
13     'Comedor_PacificoDataSet.RegCenas' Puede moverla o quitarla según sea necesario.
14     Me.RegCenasTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegCenas)
15     'TODO: esta línea de código carga datos en la tabla
16     'Comedor_PacificoDataSet.RegComidas' Puede moverla o quitarla según sea necesario.
17     Me.RegComidasTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegComidas)
18     'TODO: esta línea de código carga datos en la tabla
19     'Comedor_PacificoDataSet.RegDesayunos' Puede moverla o quitarla según sea necesario.
20     Me.RegDesayunosTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegDesayunos)
21     ShowFormStart(Me)
22     LbFecha.Text = getDia()
23     Usuario.Text = ModuloExterno.NombreUsuario
24     VerificarRegistros()
25     End Sub
26
27     Private Sub Registros_Closed(sender As Object, e As EventArgs) Handles Me.Closed
28     Me.Opacity = 0
29     End Sub
30
31     'Habilitar Y Desabilitar Opciones
32     Private Sub LblDesayunos_Click(sender As Object, e As EventArgs) Handles LblDesayunos.Click
33     Desaparece(0)
34     End Sub
35     Private Sub LblComidas_Click(sender As Object, e As EventArgs) Handles LblComidas.Click
36     Desaparece(1)
37     End Sub
38     Private Sub LblCenas_Click(sender As Object, e As EventArgs) Handles LblCenas.Click
39     Desaparece(2)
40     End Sub
41
42     'Flag = 0→Desayuno, 1→Comida, 2→Cena
43     Public FlagGlobal As Integer = 0
44
45     Public Sub Desaparece(ByVal Elemento As Integer)
46     Dim Opciones() = {LblDesayunos, LblComidas, LblCenas} 42
47     For iterador As Integer = 0 To 2
48     If iterador = Elemento Then
49     FlagGlobal = Elemento
```

```

46         Habilita(Opciones(iterador))
47     Else
48         Desabilita(Opciones(iterador))
49     End If
50     Next
51
52 End Sub
53
54 Private Sub ElButton3_Click(sender As Object, e As EventArgs) Handles ElButton3.Click
55     Dim Campos() = {txtLaptop, txtCredencial, txtEco, txtProducto}
56
57     Dim flag As Boolean = False
58
59     For iterador As Integer = 0 To 3
60         If Campos(iterador).Text = "" Then
61             MarcarError(Campos(iterador))
62             flag = True
63         End If
64     Next
65
66     If flag Then
67         MensajeDeError(ErrorBox, "Rellena todos los campos", 500)
68     Else
69         If LblDesayunos.Visible = False And LblComidas.Visible = False And LblCenas.Visible = False
70             Then
71                 MensajeDeError(ErrorBox, "Ya se han dado de alta todos los registros", 670)
72             Else
73                 SegundaValidacion()
74             End If
75         End If
76     End Sub
77
78 Private Sub SegundaValidacion()
79     Dim Campos() = {txtLaptop, txtCredencial, txtEco, txtProducto} 80     Dim Labels() = {LblLaptop,
80     LblCredencial, LblECO, LblProducto}
81
82     Dim flag As Boolean = False
83
84     For iterador As Integer = 0 To 3
85         If VerificaInts(Campos(iterador).Text) Then
86             MarcarError(Campos(iterador))
87             flag = True
88         ElseIf Integer.Parse(Campos(iterador).Text) <= 0 Then
89             MarcarError(Campos(iterador))
90             flag = True
91         End If
92     Next
93

```

```

94         If flag Then
95             MensajeDeError(ErrorBox, "Todos los valores deberán ser numéricos y mayores de 0", 850)
96         Else
97             ErrorBox.Text = ""
98             PnlCajas.Visible = False
99             PnlConfirmacion.Visible = True
100            PnlConfirmacion.Left = 95
101
102            For iterador As Integer = 0 To 3
103                Labels(iterador).Text = Campos(iterador).Text
104            Next
105
106            Dim Opciones() = {LblDesayunos, LblComidas, LblCenas}
107
108            For iterador As Integer = 0 To 2
109                If Opciones(iterador).ForeColor = Color.Black Then
110                    lblRegistro.Text = Opciones(iterador).Text
111                End If
112            Next
113
114        End If
115    End Sub
116
117    'Codigo TextChange
118    Private Sub txtLaptop_TextChanged(sender As Object, e As EventArgs) Handles txtLaptop.TextChanged
119        QuitarError(txtLaptop)
120    End Sub
121    Private Sub txtCredencial_TextChanged(sender As Object, e As EventArgs) Handles
122        txtCredencial.TextChanged
123        QuitarError(txtCredencial)
124    End Sub
125    Private Sub txtEco_TextChanged(sender As Object, e As EventArgs) Handles txtEco.TextChanged
126        QuitarError(txtEco)
127    End Sub
128    Private Sub txtProducto_TextChanged(sender As Object, e As EventArgs) Handles
129        txtProducto.TextChanged
130        QuitarError(txtProducto)
131    End Sub
132    Private Sub ElButton4_Click(sender As Object, e As EventArgs) Handles ElButton4.Click
133        PnlCajas.Visible = True
134        PnlConfirmacion.Visible = False
135    End Sub
136
137    Private Sub ElButton5_Click(sender As Object, e As EventArgs) Handles ElButton5.Click
138
139        Dim TotalTurno As Integer = 0
140        TotalTurno = CInt(txtLaptop.Text) + CInt(txtCredencial.Text) + CInt
141            (txtEco.Text) + CInt(txtProducto.Text)

```

```

142     If LblDesayunos.ForeColor = Color.Black And LblDesayunos.ForeColor <> Color.Yellow Then
143
144         Try
145             Flag =
146                 Me.TableAdapterManager.RegDesayunosTableAdapter.ExisteDesayuno
147                 (ModuloExterno.getClaveDia)
148         Catch ex As Exception
149             Flag = -1
150         End Try
151
152         If Flag = -1 Then
153             Me.TableAdapterManager.RegDesayunosTableAdapter.AltaDesayuno
154             (ModuloExterno.getClaveDia, ModuloExterno.NumeroCuenta, txtEco.Text, txtLaptop.Text,
155             txtCredencial.Text,
156             txtProducto.Text, TotalTurno, getHora)
157         Me.RegDesayunosTableAdapter.Fill
158         (Me.Comedor_PacificoDataSet.RegDesayunos)
159
160         Else
161             MensajeDeError(ErrorBox, "Ya se ha dado de alta el registro",
162             850)
163         End If
164
165     ElseIf LblComidas.ForeColor = Color.Black And LblComidas.ForeColor <> Color.Yellow Then
166
167         Try
168             Flag = Me.TableAdapterManager.RegComidasTableAdapter.ExisteComida
169             (ModuloExterno.getClaveDia)
170         Catch ex As Exception
171             Flag = -1
172         End Try
173
174         If Flag = -1 Then
175             Me.TableAdapterManager.RegComidasTableAdapter.AltaComida
176             (ModuloExterno.getClaveDia, ModuloExterno.NumeroCuenta,
177             txtEco.Text, txtLaptop.Text, txtCredencial.Text,
178             txtProducto.Text, TotalTurno, getHora)
179         Me.RegComidasTableAdapter.Fill
180         (Me.Comedor_PacificoDataSet.RegComidas)
181
182         Else
183             MensajeDeError(ErrorBox, "Ya se ha dado de alta el registro",
184             850)
185         End If
186
187     ElseIf LblCenas.ForeColor = Color.Black And LblCenas.ForeColor <> Color.Yellow Then
188
189         Try
190             Flag = Me.TableAdapterManager.RegCenasTableAdapter.ExisteCena
191             (ModuloExterno.getClaveDia)
192         Catch ex As Exception
193             Flag = -1

```

```

179         End Try
180
181     If Flag = -1 Then
182         Me.TableAdapterManager.RegCenasTableAdapter.AltaCena
            (ModuloExterno.getClaveDia, ModuloExterno.NumeroCuenta, txtEco.Text, txtLaptop.Text,
            txtCredencial.Text,
            txtProducto.Text, TotalTurno, getHora)
183         Me.RegCenasTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegCenas)
184     Else
185         MensajeDeError(ErrorBox, "Ya se ha dado de alta el registro",
            850)
186     End If
187     Else
188         MensajeDeError(ErrorBox, "Ya se ha dado de alta el registro", 850)
189     End If
190
191 'Actualizar El Registro Diario
192 If Flag = -1 Then
193
194     Dim Eco As Integer =
        Me.TableAdapterManager.RegistrosTotalesTableAdapter.getEco
        (ModuloExterno.getClaveDia)
195     Dim Laptop As Integer =
        Me.TableAdapterManager.RegistrosTotalesTableAdapter.getLaptop
        (ModuloExterno.getClaveDia)
196     Dim Credencial As Integer =
        Me.TableAdapterManager.RegistrosTotalesTableAdapter.getSinCredencia
        l(ModuloExterno.getClaveDia)
197     Dim Producto As Integer =
        Me.TableAdapterManager.RegistrosTotalesTableAdapter.getProducto
        (ModuloExterno.getClaveDia)
198     Dim TotalDia As Integer =
        Me.TableAdapterManager.RegistrosTotalesTableAdapter.getTotal
        (ModuloExterno.getClaveDia)
199
200         Me.RegistrosTotalesTableAdapter.Fill
            (Me.Comedor_PacificoDataSet.RegistrosTotales)
201
202     Eco += CInt(txtEco.Text)
203     Laptop += CInt(txtLaptop.Text)
204     Credencial += CInt(txtCredencial.Text)
205     Producto += CInt(txtProducto.Text)
206     TotalDia += TotalTurno
207
208     Dim Total As Integer =
209     Me.TableAdapterManager.RegistrosTotalesTableAdapter.ActualizarDia
        (Eco, Laptop, Credencial, Producto, TotalDia,
        ModuloExterno.getClaveDia)
210 End If
211 'Si todo sale bien

```

```

212 Dim Campos() = {txtLaptop, txtCredencial, txtEco, txtProducto} 213
214 For iterador As Integer = 0 To 3
215 Campos(iterador).Text = ""
216 Next
217
218 PnlCajas.Visible = True
219 PnlConfirmacion.Visible = False
220
221 Desaparece(0)
222 VerificarRegistros()
223 'Aqui va el codigo de alta
224 End Sub
225
226 'Codigo Hover
227 Private Sub LblDesayunos_MouseEnter(sender As Object, e As EventArgs) Handles
LblDesayunos.MouseEnter
228 If Not FlagGlobal = 0 Then
229 LblDesayunos.ForeColor = Color.PowderBlue
230 End If
231 End Sub
232 Private Sub LblDesayunos_MouseLeave(sender As Object, e As EventArgs) Handles
LblDesayunos.MouseLeave
233 If Not FlagGlobal = 0 Then
234 LblDesayunos.ForeColor = Color.DarkGray
235 End If
236 End Sub
237 Private Sub LblComidas_MouseEnter(sender As Object, e As EventArgs) Handles
LblComidas.MouseEnter
238 If Not FlagGlobal = 1 Then
239 LblComidas.ForeColor = Color.PowderBlue
240 End If
241 End Sub
242 Private Sub LblComidas_MouseLeave(sender As Object, e As EventArgs) Handles
LblComidas.MouseLeave
243 If Not FlagGlobal = 1 Then
244 LblComidas.ForeColor = Color.DarkGray
245 End If
246 End Sub
247 Private Sub LblCenas_MouseEnter(sender As Object, e As EventArgs) Handles
LblCenas.MouseEnter
248 If Not FlagGlobal = 2 Then
249 LblCenas.ForeColor = Color.PowderBlue
250 End If
251 End Sub
252 Private Sub LblCenas_MouseLeave(sender As Object, e As EventArgs) Handles
LblCenas.MouseLeave
253 If Not FlagGlobal = 2 Then
254 LblCenas.ForeColor = Color.DarkGray
255 End If
256 End Sub
257

```

```

258 Private Sub ElButton2_Click(sender As Object, e As EventArgs) Handles ElButton2.Click
259 'MuestraLosRegistros
260 MostrarDatos(getClaveDia())
261 End Sub
262
263 Private Sub ElButton6_Click(sender As Object, e As EventArgs) Handles ElButton6.Click
264 PnlCajas.Visible = True
265 PnlConfirmacion.Visible = False
266 PnlRegistros.Visible = False
267 End Sub
268
269 Private Sub ElButton7_Click(sender As Object, e As EventArgs) Handles ElButton7.Click
270 'Desplegar Fecha
271 PnlRegistros.Visible = False
272
273 SelectorFecha.Visible = True
274
275 SelectorFecha.Left = 67
276 SelectorFecha.Top = 95
277 End Sub
278
279 Private Sub ElButton8_Click(sender As Object, e As EventArgs) Handles ElButton8.Click
280 'Mostrar Selector defecah
281 PnlRegistros.Visible = True
282
283 SelectorFecha.Visible = False
284
285 SelectorFecha.Left = 67
286 SelectorFecha.Top = 95
287 End Sub
288
289 Private Sub ElButton9_Click(sender As Object, e As EventArgs) Handles ElButton9.Click
290 'Desplegar de la nueva fecha
291 Dim claveBusqueda As String = SelectorDeFecha.Value 292
293 Dim NeoClaveBusqueda As String = claveBusqueda.Replace("/", "") 294
295 Dim claveBusquedaA As String = NeoClaveBusqueda.Substring(0, 4) 296 Dim claveBusquedaB As
String = NeoClaveBusqueda.Substring(6, 2) 297
298 Dim FinalclaveBusqueda As String = claveBusquedaA + claveBusquedaB 299
300 If FinalclaveBusqueda = getClaveDia() Then
301 Label15.Text = "Registros del día de Hoy"
302 Else
303 Label15.Text = "Registros del: " + claveBusqueda.Substring(0, 10)
304 End If
305
306 MostrarDatos(FinalclaveBusqueda)
307
308 SelectorFecha.Visible = False
309 End Sub
310
311 Public Sub MostrarDatos(ByVal CadenaEnviada)
312 TablaRegistros.Height = 32 + 30 * 5

```



```

313
314 Me.RegDesayunosTableAdapter.RellenaDesayuno
      (Me.Comedor_PacificoDataSet.RegDesayunos, CadenaEnviada)
315 Me.RegComidasTableAdapter.RellenaComidas
      (Me.Comedor_PacificoDataSet.RegComidas, CadenaEnviada)
316 Me.RegCenasTableAdapter.RellenaCenas(Me.Comedor_PacificoDataSet.RegCenas, CadenaEnviada)
317
318 Me.RegistrosTotalesTableAdapter.RellenaTotales
      (Me.Comedor_PacificoDataSet.RegistrosTotales, CadenaEnviada)
319
320 PnlCajas.Visible = False
321 PnlConfirmacion.Visible = False
322 PnlRegistros.Visible = True
323
324 PnlRegistros.Left = 24
325 PnlRegistros.Top = 98
326
327 TablaRegistros.Rows.Clear()
328
329 Dim Textos() As String = {"ECO", "Laptop", "Sin Credencial", "Por Producto", "Total"}
330 For iterador As Integer = 0 To 4
331     TablaRegistros.Rows.Add()
332     TablaRegistros.Item(0, iterador).Value = Textos(iterador) 333
334     TablaRegistros.Item(1, iterador).Value = DataGridView1.Item(iterador, 0).Value
335     TablaRegistros.Columns(iterador).DefaultCellStyle.Alignment =
        DataGridViewContentAlignment.MiddleCenter
336
337     TablaRegistros.Item(2, iterador).Value = DataGridView2.Item(iterador, 0).Value
338     TablaRegistros.Item(3, iterador).Value = DataGridView3.Item(iterador, 0).Value
339     TablaRegistros.Item(4, iterador).Value = DataGridView4.Item(iterador, 0).Value
340     Next
341
342 TablaRegistros.Columns(0).DefaultCellStyle.Alignment = DataGridViewContentAlignment.TopLeft
343
344 For iterador As Integer = 0 To 4
345     TablaRegistros.Rows.Add()
346     TablaRegistros.Item(0, iterador).Value = Textos(iterador) 347
348     TablaRegistros.Rows(4).DefaultCellStyle.BackColor =
        Color.LightSteelBlue
349     TablaRegistros.Rows(4).DefaultCellStyle.Font = New Drawing.Font
        ("Century Gothic", 17, FontStyle.Regular, GraphicsUnit.Point)
350     TablaRegistros.Rows(4).DefaultCellStyle.ForeColor =
        Color.DarkSlateBlue
351     Next
352 End Sub
353
354 Public Sub VerificarRegistros()
355     Dim Desayunos As Integer = -1
356     Try
357         Desayunos = Me.RegDesayunosTableAdapter.ExisteDesayuno(getClaveDia)
358     Catch ex As Exception

```

```

359         Desayunos = -1
360     End Try
361
362     If Desayunos <> -1 Then
363         LblDesayunos.Visible = False
364         LblDesayunos.ForeColor = Color.Yellow
365         LblNoDesayuno.Visible = True
366     End If
367
368
369     Dim Comidas As Integer = -1
370     Try
371         Comidas = Me.RegComidasTableAdapter.ExisteComida(getClaveDia)
372     Catch ex As Exception
373         Comidas = -1
374     End Try
375
376     If Comidas <> -1 Then
377         LblComidas.Visible = False
378         LblComidas.ForeColor = Color.Yellow
379         LblNoComida.Visible = True
380     End If
381
382     Dim Cenas As Integer = -1
383     Try
384         Cenas = Me.RegCenasTableAdapter.ExisteCena(getClaveDia)
385     Catch ex As Exception
386         Cenas = -1
387     End Try
388
389     If Cenas <> -1 Then
390         LblCenas.Visible = False
391         LblCenas.ForeColor = Color.Yellow
392         LblNoCena.Visible = True
393     End If
394 End Sub
395 End Class

```

Código del Formulario (Gastos y Ventas):

```
1 Public Class VentasGastos
2 Private Sub ElButton1_Click(sender As Object, e As EventArgs) Handles ElButton1.Click
3 PantallaPrincipal.Show()
4 Me.Close()
5 End Sub
6
7 Private Sub VentasGastos_Load(sender As Object, e As EventArgs) Handles MyBase.Load
8 'TODO: esta línea de código carga datos en la tabla
9 'Comedor_PacificoDataSet.RegistrosGV' Puede moverla o quitarla según sea necesario.
10 Me.RegistrosGVTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegistrosGV)
11 'TODO: esta línea de código carga datos en la tabla
12 'Comedor_PacificoDataSet.RegistrosV' Puede moverla o quitarla según sea necesario.
13 Me.RegistrosVTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegistrosV)
14 'TODO: esta línea de código carga datos en la tabla
15 'Comedor_PacificoDataSet.RegistrosG' Puede moverla o quitarla según sea necesario.
16 Me.RegistrosGTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegistrosG)
17 ShowFormStart(Me)
18 LbFecha.Text = getDia()
19 Usuario.Text = ModuloExterno.NombreUsuario
20
21 txtMotivo.Text = ""
22 txtCantidad.Text = ""
23 PnlCajas.Visible = True
24 PnlConfirmacion.Visible = False
25
26 End Sub
27
28 Private Sub VentasGastos_Closed(sender As Object, e As EventArgs) Handles Me.Closed
29 Me.Opacity = 0
30 End Sub
31
32 Private Sub ElButton4_Click(sender As Object, e As EventArgs) Handles ElButton4.Click
33 'Validacion de Si Estan rellenos
34 Dim flag As Boolean = False
35
36 If txtMotivo.Text = "" Then
37 MarcarError(txtMotivo)
38 flag = True
39 End If
40
41 If txtCantidad.Text = "" Then
42 MarcarError(txtCantidad)
43 flag = True
44 End If
45
46 If flag Then
47 MensajeDeError(MessageBox, "Rellene todos los campos", 700)
48 Else
49 SegundaValidacion()
```

```

47         End If
48     End Sub
49     Public Sub SegundaValidacion()
50         Dim flag As Boolean
51
52         If VerificaInts(txtCantidad.Text) Then
53             MensajeDeError(ErrorBox, "Ingrese un valor numérico", 700)
54             MarcarError(txtCantidad)
55             flag = True
56
57         ElseIf Integer.Parse(txtCantidad.Text) <= 0 Then
58             MensajeDeError(ErrorBox, "Ingrese un valor numérico mayor a 0", 730)
59             MarcarError(txtCantidad)
60             flag = True
61         End If
62
63         If (txtMotivo.Text).Length <= 5 Then
64             MensajeDeError(ErrorBox, "El Motivo debe de ser mayor a 5 caracteres", 860)
65             MarcarError(txtMotivo)
66             flag = True
67         End If
68
69         If Not flag Then
70             Confirmacion()
71         End If
72     End Sub
73     Public Sub Confirmacion()
74
75         lblMotivo.Text = txtMotivo.Text
76         lblCantidad.Text = txtCantidad.Text + "$ m.n"
77         lblUsuario.Text = Usuario.Text
78
79         If lblTipoRegistro.ForeColor = Color.Black Then
80             lblRegistro.Text = "Gasto"
81         Else
82             lblRegistro.Text = "Venta"
83         End If
84
85         PnlConfirmacion.Left = 212
86         PnlCajas.Visible = False
87         PnlConfirmacion.Visible = True
88
89         ErrorBox.Text = ""
90     End Sub
91
92     Private Sub ElButton6_Click(sender As Object, e As EventArgs) Handles ElButton6.Click
93         PnlCajas.Visible = True
94         PnlConfirmacion.Visible = False
95     End Sub
96
97     Private Sub ElButton7_Click(sender As Object, e As EventArgs) Handles ElButton7.Click

```

➤

```

98      'Balanceo
99      Dim tGasto As Integer = Me.RegistrosGVTableAdapter.getGasto(getClaveDia)
100
101
102      Dim tVenta As Integer = Me.RegistrosGVTableAdapter.getVenta(getClaveDia)
103
104      Dim Balance As Integer = Me.RegistrosGVTableAdapter.getBalance
105      (getClaveDia)
106
107      'Aqui va el codigo de alta en gasto, o venta 106
108      If lblTipoRegistro.ForeColor = Color.Black Then
109          Me.RegistrosGTableAdapter.AltaGasto(getClaveDia,
110              ModuloExterno.NumeroCuenta, txtMotivo.Text, txtCantidad.Text, getHora())
111          tGasto += CInt(txtCantidad.Text)
112          Balance = Balance - CInt(txtCantidad.Text)
113          Me.RegistrosGTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegistrosG)
114      Else
115          Me.RegistrosVTableAdapter.AltaVenta(getClaveDia,
116              ModuloExterno.NumeroCuenta, txtMotivo.Text, txtCantidad.Text, getHora())
117          tVenta += CInt(txtCantidad.Text)
118          Balance = Balance + CInt(txtCantidad.Text)
119          Me.RegistrosVTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegistrosV)
120      End If
121
122      Me.RegistrosGVTableAdapter.ActualizarDia(tGasto, tVenta, Balance, getClaveDia)
123
124      'Codigo de Refrescar tablas ambas
125      Me.RegistrosGVTableAdapter.Fill(Me.Comedor_PacificoDataSet.RegistrosGV) 123
126      txtMotivo.Text = ""
127      txtCantidad.Text = ""
128      PnlCajas.Visible = True
129      PnlConfirmacion.Visible = False
130
131      Desabilita(Label3)
132      Habilita(lblTipoRegistro)
133      End Sub
134
135      Private Sub QuitarMarcas(sender As Object, e As EventArgs) Handles txtMotivo.TextChanged,
136      txtCantidad.TextChanged
137      QuitarError(sender)
138      End Sub
139
140      Private Sub lblTipoRegistro_Click(sender As Object, e As EventArgs) Handles lblTipoRegistro.Click
141      Desabilita(Label3)
142      Habilita(lblTipoRegistro)
143      End Sub
144
145      Private Sub Label3_Click(sender As Object, e As EventArgs) Handles Label3.Click
146      Habilita(Label3)
147      Desabilita(lblTipoRegistro)

```

```

145         End Sub
146
147         Private Sub EfectoHover(sender As Object, e As EventArgs) Handles
Label3.MouseEnter, lblTipoRegistro.MouseEnter, VentaLblRegistro.MouseEnter,
        GastoLblRegistro.MouseEnter
148             If sender.ForeColor = Color.DarkGray Then
149                 sender.ForeColor = Color.PowderBlue
150             End If
151         End Sub
152
153         Private Sub Label3_MouseLeave(sender As Object, e As EventArgs) Handles
Label3.MouseLeave, lblTipoRegistro.MouseLeave, VentaLblRegistro.MouseLeave,
        GastoLblRegistro.MouseLeave
154             If sender.ForeColor = Color.PowderBlue Then
155                 sender.ForeColor = Color.DarkGray
156             End If
157         End Sub
158
159         Private Sub VentaLblRegistro_Click(sender As Object, e As EventArgs) Handles VentaLblRegistro.Click
160             Habilita(VentaLblRegistro)
161             Desabilita(GastoLblRegistro)
162
163             If nRegistrosV = 0 Then
164                 NoHayRegistros.Visible = True
165             Else
166                 NoHayRegistros.Visible = False
167             End If
168
169             TablaVentas.Visible = True
170             TablaGastos.Visible = False
171
172             TablaVentas.Left = 45
173             TablaVentas.Top = 123
174
175             TablaGastos.Left = 1200
176             TablaGastos.Top = 700
177         End Sub
178
179         Private Sub GastoLblRegistro_Click(sender As Object, e As EventArgs) Handles GastoLblRegistro.Click
180             Habilita(GastoLblRegistro)
181             Desabilita(VentaLblRegistro)
182
183             If nRegistrosG = 0 Then
184                 NoHayRegistros.Visible = True
185             Else
186                 NoHayRegistros.Visible = False
187             End If
188
189             TablaGastos.Left = 45
190             TablaGastos.Top = 123
191

```

```

192     TablaVentas.Left = 1200
193     TablaVentas.Top = 700
194
195     TablaVentas.Visible = False
196     TablaGastos.Visible = True
197     End Sub
198
199     Dim nRegistrosG As Integer = 0
200     Dim nRegistrosV As Integer = 0
201
202     Private Sub ElButton5_Click(sender As Object, e As EventArgs) Handles ElButton5.Click
203         PnlCajas.Visible = False
204         PnlConfirmacion.Visible = False
205         PnlRegistros.Visible = True
206
207         PnlRegistros.Left = 67
208         PnlRegistros.Top = 95
209
210         GastoLblRegistro_Click(GastoLblRegistro, EventArgs.Empty)
211         'Aqui se llenaran ambos dataGrip con los datos del dia de hoy 212
212         Try
213             nRegistrosG = Me.RegistrosGTableAdapter.getRegistrosGastos (getClaveDia())
214         Catch ex As Exception
215             nRegistrosG = 0
216         End Try
217
218         Me.RegistrosGTableAdapter.FillBy(Me.Comedor_PacificoDataSet.RegistrosG, getClaveDia())
219         Me.RegistrosVTableAdapter.FillBy(Me.Comedor_PacificoDataSet.RegistrosV, getClaveDia())
220
221         'CreadorDeEspacio
222         Try
223             nRegistrosV = Me.RegistrosVTableAdapter.getRegistrosVentas (getClaveDia())
224         Catch ex As Exception
225             nRegistrosG = 0
226         End Try
227
228         TablaGastos.Height = 32 + 30 * nRegistrosG
229         TablaVentas.Height = 32 + 30 * nRegistrosV
230
231         If nRegistrosG = 0 Then
232             NoHayRegistros.Visible = True
233         Else
234             NoHayRegistros.Visible = True
235         End If
236
237         Dim Balance As Integer = 0
238         Try
239             Balance = Me.RegistrosGVTableAdapter.getBalance(getClaveDia())
240         Catch ex As Exception
241             Balance = 0
242         End Try
243

```

```

244
245         Label16.Text = "Balance: " + Balance.ToString()
246
247     TablaGastos.Visible = True
248 End Sub
249
250 Private Sub ElButton2_Click(sender As Object, e As EventArgs) Handles ElButton2.Click
251     TablaGastos.Visible = False
252     TablaVentas.Visible = False
253
254     PnlCajas.Visible = True
255     PnlConfirmacion.Visible = False
256     PnlRegistros.Visible = False
257
258     PnlCajas.Left = 172
259     PnlCajas.Top = 149
260 End Sub
261
262 Private Sub ElButton3_Click(sender As Object, e As EventArgs) Handles ElButton3.Click
263     'Mostrar Selector defecah
264     PnlRegistros.Visible = False
265
266     SelectorFecha.Visible = True
267
268     SelectorFecha.Left = 67
269     SelectorFecha.Top = 95
270 End Sub
271
272 Private Sub ElButton8_Click(sender As Object, e As EventArgs) Handles ElButton8.Click,
ElButton9.Click
273 '
274 GastoLblRegistro_Click(GastoLblRegistro, EventArgs.Empty)
275 PnlRegistros.Visible = True
276
277     SelectorFecha.Visible = False
278
279     PnlRegistros.Visible = True
280
281     PnlRegistros.Left = 67
282     PnlRegistros.Top = 95
283
284     SelectorFecha.Visible = False
285
286     If sender.text = "Aceptar" Then
287         'Poner de titulo el nuevo
288
289         Dim claveBusqueda As String = SelectorDeFecha.Value
290         Label15.Text = "Registros del: " + claveBusqueda.Substring(0, 10)
291         claveBusqueda = claveBusqueda.Replace("/", "")
292

```



```

293     Dim claveBusquedaA As String = claveBusqueda.Substring(0, 4) 294 Dim claveBusquedaB As String =
claveBusqueda.Substring(6, 2) 295
296         claveBusqueda = claveBusquedaA + claveBusquedaB
297         GastoLblRegistro_Click(GastoLblRegistro, EventArgs.Empty)
298     'Aqui se llenaran ambos dataGrip con los datos del dia de hoy 299
300         Try
301             nRegistrosG = Me.RegistrosGTableAdapter.getRegistrosGastos
                (claveBusqueda)
302         Catch ex As Exception
303             nRegistrosG = 0
304         End Try
305
306     Me.RegistrosGTableAdapter.FillBy
        (Me.Comedor_PacificoDataSet.RegistrosG, claveBusqueda)
307     Me.RegistrosVTableAdapter.FillBy
        (Me.Comedor_PacificoDataSet.RegistrosV, claveBusqueda)
308
309     'CreadorDeEspacio
310     Try
311         nRegistrosV = Me.RegistrosVTableAdapter.getRegistrosVentas
            (claveBusqueda)
312     Catch ex As Exception
313         nRegistrosG = 0
314     End Try
315
316     TablaGastos.Height = 32 + 30 * nRegistrosG
317     TablaVentas.Height = 32 + 30 * nRegistrosV
318
319     If nRegistrosG = 0 Then
320         NoHayRegistros.Visible = True
321     Else
322         NoHayRegistros.Visible = True
323     End If
324
325     Dim Balance As Integer = 0
326     Try
327         Balance = Me.RegistrosGVTableAdapter.getBalance(claveBusqueda)
328     Catch ex As Exception
329         Balance = 0
330     End Try
331
332     Label16.Text = "Balance: " + Balance.ToString()
333 End If
334 End Sub
335
336 End Class

```

Código del Formulario (Apuntes):

```
1 Public Class Apuntes
2
3     Public Hojas As Integer = 0
4     Public HojaActual As Integer = 1
5     Public PseudoHoja As Integer = 0
6
7     Private Sub ElButton1_Click(sender As Object, e As EventArgs) Handles ElButton1.Click
8         PantallaPrincipal.Show()
9         Me.Close()
10    End Sub
11
12    Private Sub Apuntes_Load(sender As Object, e As EventArgs) Handles MyBase.Load
13        'TODO: esta línea de código carga datos en la tabla
14        'Comedor_PacificoDataSet.Apuntes' Puede moverla o quitarla según sea necesario.
15        Me.ApuntesTableAdapter.Fill(Me.Comedor_PacificoDataSet.Apuntes)
16        ShowFormStart(Me)
17        LbFecha.Text = getDia()
18        Usuario.Text = ModuloExterno.NombreUsuario
19
20        Despliegue()
21    End Sub
22
23    Private Sub Apuntes_Closed(sender As Object, e As EventArgs) Handles Me.Closed
24        Me.Opacity = 0
25    End Sub
26
27    Private Sub ElButton23_Click(sender As Object, e As EventArgs)
28        PnlLista.Visible = False
29        PnlNuevo.Visible = True
30        PnlNuevo.Left = 44
31        PnlNuevo.Top = 110
32    End Sub
33
34    Private Sub ElButton4_Click(sender As Object, e As EventArgs) Handles ElButton4.Click
35        Label20.Text = "Nuevo Apunte"
36
37        txtTitulo.Text = ""
38        txtContenido.Text = ""
39        CheckBoxEditable.Checked = False
40
41        txtTitulo.BackgroundColor = Color.White
42        txtTitulo.TextColour = Color.Black
43
44        txtTitulo.Enabled = True
45
46        BtnEliminar.Visible = False
47        BtnEliminar.Enabled = False
48
49        PnlLista.Visible = False
```

```

49     PnlNuevo.Visible = True
50     PnlNuevo.Left = 44
51     PnlNuevo.Top = 110
52     End Sub
53
54     Private Sub ElButton24_Click(sender As Object, e As EventArgs) Handles ElButton24.Click
55         PnlNuevo.Visible = False
56         PnlLista.Visible = True
57         PnlLista.Left = 44
58         PnlLista.Top = 110
59         End Sub
60
61     Private Sub ElButton3_Click(sender As Object, e As EventArgs)
62         Label20.Text = "Editar Apunte"
63
64         txtTitulo.Text = "TituloApunte"
65         txtTitulo.BackgroundColor = Color.Gainsboro
66         txtTitulo.TextColour = Color.White
67
68         txtTitulo.Enabled = False
69
70         PnlNuevo.Visible = True
71         PnlLista.Visible = False
72
73         PnlNuevo.Left = 44
74         PnlNuevo.Top = 110
75         End Sub
76
77     Private Sub ElButton7_Click(sender As Object, e As EventArgs) Handles ElButton7.Click
78         'Arranca Primer Validacion
79         Dim flag As Boolean = False
80
81         If txtContenido.Text = "" Then
82             MarcarError(txtContenido)
83             flag = True
84         End If
85
86         If txtTitulo.Text = "" Then
87             MarcarError(txtTitulo)
88             flag = True
89         End If
90
91         If flag Then
92             'Error
93             MessageBox.Text = "          Rellena todos los campos"
94
95         Else
96             MessageBox.Text = ""
97             'Confirmacion
98             LblAutor.Text = ModuloExterno.NombreUsuario
99             LblContenido.Text = txtContenido.Text
100            LblTitulo.Text = txtTitulo.Text

```

```

100
101         If CheckBoxEditable.Checked Then
102             LblPermisos.Text = "Todos los usuarios podrán ver y editar el apunte."
103         Else
104             LblPermisos.Text = "Solo el autor podra editar el apunte."
105         End If
106
107         PnlNuevo.Visible = False
108         PnlConfirmacion.Visible = True
109
110         PnlConfirmacion.Left = 44
111         PnlConfirmacion.Top = 110
112     End If
113
114 End Sub
115
116 Private Sub txtTitulo_TextChanged(sender As Object, e As EventArgs) Handles txtTitulo.TextChanged
117     QuitarError(txtTitulo)
118 End Sub
119
120 Private Sub txtContenido_TextChanged(sender As Object, e As EventArgs) Handles
121     txtContenido.TextChanged
122     QuitarError(txtContenido)
123 End Sub
124
125 Private Sub ElButton8_Click(sender As Object, e As EventArgs) Handles ElButton8.Click
126     PnlConfirmacion.Visible = False
127     PnlNuevo.Visible = True
128
129     PnlNuevo.Left = 44
130     PnlNuevo.Top = 110
131 End Sub
132
133 Private Sub ElButton9_Click(sender As Object, e As EventArgs) Handles ElButton9.Click
134
135     Dim paraTodos As Integer
136     If CheckBoxEditable.Checked Then
137         paraTodos = 1
138     Else
139         paraTodos = 0
140     End If
141
142     If Label20.Text = "Ver / Eliminar / Editar Apunte" Then
143         Me.TableAdapterManager.ApuntesTableAdapter.ActualizarApunte
144             (txtTitulo.Text, txtContenido.Text, paraTodos,
145             DistribuidorSeleccionado)
146         Despliegue()
147         limpiar()
148
149         PnlConfirmacion.Visible = False
150         PnlNuevo.Visible = False

```

```

148         PnlLista.Visible = True
149     Else
150         'Aqui va la alta del apunte
151         Dim ExisteTitulo As String = "NoHay"
152
153         Try
154             ExisteTitulo =
155                 Me.TableAdapterManager.ApuntesTableAdapter.ExisteTitulo (txtTitulo.Text)
156         Catch ex As Exception
157             ExisteTitulo = "NoHay"
158         End Try
159         If ExisteTitulo = "NoHay" Then
160             'Alta
161             Me.TableAdapterManager.ApuntesTableAdapter.AltaApunte
162                 (txtTitulo.Text, ModuloExterno.NumeroCuenta, txtContenido.Text, paraTodos)
163             Despliegue()
164             limpiar()
165
166             PnlConfirmacion.Visible = False
167             PnlNuevo.Visible = True
168
169         Else
170             MsgBox.Text = "No se puede repetir el titulo de un apunte"
171             PnlConfirmacion.Visible = False
172             PnlNuevo.Visible = True
173         End If
174     End If
175 End Sub
176
177 Public Sub limpiar()
178     txtTitulo.Text = ""
179     txtContenido.Text = ""
180     CheckBoxEditable.Checked = False
181 End Sub
182
183 Dim tDistribuidores As Double
184
185 Public Sub Despliegue()
186     tDistribuidores = Me.TableAdapterManager.ApuntesTableAdapter.nApuntes() 187
188     nDistribuidores.Text = "# Apuntes: " + tDistribuidores.ToString() 189
189     If tDistribuidores = 0 Then
190         Dim Casillas = {psObjeto1, psObjeto2, psObjeto3, psObjeto4, psObjeto5, psObjeto6}
191
192         For iteradorReal As Integer = 0 To 5
193
194             Casillas(iteradorReal).Visible = False
195             Casillas(iteradorReal).Enabled = False
196
197         Next
198

```

```

199         psObjeto33.Visible = True
200     Else
201         psObjeto33.Visible = False
202     End If
203
204     Hojas = Math.Ceiling(tDistribuidores / 6)
205     PseudoHoja = Hojas
206
207     panelHojas.Text = "1/" + Hojas.ToString()
208
209     If HojaActual = Hojas Then
210         nextHoja.Enabled = False
211     Else
212         nextHoja.Enabled = True
213     End If
214
215     CrearPseudoArreglo()
216     MostrarElementos()
217 End Sub
218 Private Sub nextHoja_Click(sender As Object, e As EventArgs) Handles nextHoja.Click
219     HojaActual = HojaActual + 1
220     panelHojas.Text = HojaActual.ToString() + "/" + Hojas.ToString()
221     PseudoHoja = PseudoHoja - 1
222     MostrarElementos()
223 End Sub
224
225 Private Sub backHoja_Click(sender As Object, e As EventArgs) Handles backHoja.Click
226     HojaActual = HojaActual - 1
227     panelHojas.Text = HojaActual.ToString() + "/" + Hojas.ToString()
228     PseudoHoja = PseudoHoja + 1
229     MostrarElementos()
230 End Sub
231
232 Public Sub MostrarElementos() 233
234     Dim iteradorBase As Integer = (6 * (PseudoHoja - 1)) + 1
235     'Fin
236     Dim Casillas = {psObjeto1, psObjeto2, psObjeto3, psObjeto4, psObjeto5, psObjeto6}
237
238     If tDistribuidores <> 0 Then
239         For iteradorReal As Integer = 0 To 5
240
241             'Se acabo el ultimo registro
242             If (iteradorBase + iteradorReal - 1) >= arregloSinPerdidas.Count
243
244                 Then
245                     For neoIterador = iteradorReal To 5
246                         Casillas(neoIterador).Visible = False
247                         Casillas(neoIterador).Enabled = False
248                     Next
249                     iteradorReal = 5
250                 Else

```

```

249         Dim nombre As String =
                Me.TableAdapterManager.ApuntesTableAdapter.getTitulo
                    (arregloSinPerdidas.Item(iteradorBase + iteradorReal - 1))
250
251         Casillas(iteradorReal).Visible = True
252         Casillas(iteradorReal).Enabled = True
253         Casillas(iteradorReal).AccessibleName =
                    arregloSinPerdidas.Item(iteradorBase + iteradorReal - 1)
254         Casillas(iteradorReal).Text = (iteradorBase +
                    iteradorReal).ToString() + ".- " + nombre
255     End If
256     Next
257 End If
258 End Sub
259
260 Dim PaquetesPerdidos As Integer = 0
261
262 Dim arregloSinPerdidas As New ArrayList
263
264 Public Sub CrearPseudoArreglo()
265     ObtenerPerdidaDePaquetes()
266
267     arregloSinPerdidas.Clear()
268
269     Dim pseudoIterador = 0
270     Dim ContadorCasilla = 0
271     If tDistribuidores <> 0 Then
272     For iteradorReal As Integer = 1 To PaquetesPerdidos + tDistribuidores 273
274         Dim na As String =
                Me.TableAdapterManager.ApuntesTableAdapter.getTitulo
                    (iteradorReal)
275
276         If na <> "" Then
277
278             arregloSinPerdidas.Add(iteradorReal)
279
280         End If
281     Next
282     End If
283
284 End Sub
285
286 Public Sub ObtenerPerdidaDePaquetes()
287
288     Dim LimitePaquete As Integer = 0
289
290     Try
291         LimitePaquete =
                Me.TableAdapterManager.ApuntesTableAdapter.ApuntesPerdidos()
292

```

```

293     Catch ex As Exception 294
LimitePaquete = 0
295         End Try
296         If LimitePaquete <> 0 Then
297             For BuscadorPerdidos As Integer = 1 To LimitePaquete
298                 Dim Hueco As String =
Me.TableAdapterManager.ApuntesTableAdapter.getTitulo
(BuscadorPerdidos)
299                 If Hueco = "" Then
300                     PaquetesPerdidos += 1
301                 End If
302             Next
303         End If
304     End Sub
305
306     Dim DistribuidorSeleccionado = -1
307
308     Private Sub LanzaVerEditar(sen As Object, e As EventArgs) Handles psObjeto1.Click, psObjeto2.Click,
psObjeto6.Click, psObjeto5.Click, psObjeto4.Click, psObjeto3.Click
309         DistribuidorSeleccionado = Integer.Parse(sen.AccessibleName)
310
311         Label20.Text = "Ver / Eliminar / Editar Apunte"
312
313         BtnEliminar.Enabled = True
314         BtnEliminar.Visible = True
315
316         txtTitulo.Text = Me.TableAdapterManager.ApuntesTableAdapter.getTitulo (DistribuidorSeleccionado)
317         txtTitulo.BackgroundColor = Color.Gainsboro
318         txtTitulo.TextColour = Color.White
319
320         txtTitulo.Enabled = False
321
322         Dim paraTodos As Integer =
Me.TableAdapterManager.ApuntesTableAdapter.getParaTodos
(DistribuidorSeleccionado)
323
324         txtContenido.Text =
Me.TableAdapterManager.ApuntesTableAdapter.getTextoApunte
(DistribuidorSeleccionado)
325
326         Dim autor As Integer =
Me.TableAdapterManager.ApuntesTableAdapter.getAutor
(DistribuidorSeleccionado)
327
328         If (ModuloExterno.NumeroCuenta = autor) Then
329             ElButton7.Enabled = True
330             BtnEliminar.Enabled = True
331         Else
332             ElButton7.Enabled = False
333             BtnEliminar.Enabled = False
334

```



```

335         End If
336
337         If paraTodos = 1 Or ModuloExterno.NumeroCuenta = autor Then
338             txtContenido.Enabled = True
339         Else
340             txtContenido.Enabled = False
341         End If
342
343         If paraTodos = 1 Then
344             CheckBoxEditable.Checked = True
345         Else
346             CheckBoxEditable.Checked = False
347         End If
348         PnlNuevo.Visible = True
349         PnlLista.Visible = False
350
351         PnlNuevo.Left = 44
352         PnlNuevo.Top = 110
353     End Sub
354
355     Private Sub BtnEliminar_Click(sender As Object, e As EventArgs) Handles BtnEliminar.Click
356         Me.TableAdapterManager.ApuntesTableAdapter.BorrarApunte
357             (DistribuidorSeleccionado)
358         Me.ApuntesTableAdapter.Fill(Me.Comedor_PacificoDataSet.Apuntes) 358
359         CrearPseudoArreglo()
360         Despliegue()
361         limpiar()
362
363         PnlNuevo.Visible = False
364         PnlLista.Visible = True
365
366         BtnEliminar.Enabled = False
367         BtnEliminar.Visible = False
368
369         PnlLista.Left = 44
370         PnlLista.Top = 110
371     End Sub
372 End Class

```

Código Auxiliar(Modulo Externo):

```
1  Module ModuloExterno
2  Public NombreUsuario As String
3  Public NumeroCuenta As Integer
4
5      Public Function MarcarError(ByVal Objeto)
6          Objeto.BorderColor = Color.Red
7          Objeto.TextColour = Color.Red
8
9      Return 0
10     End Function
11
12     Public Function QuitarError(ByVal Objeto)
13         Objeto.BorderColor = Color.DimGray
14         Objeto.TextColour = Color.Black
15         Return 0
16     End Function
17
18     Public Function VerificaInts(ByVal Cadena As String)
19         Try
20             Convert.ToInt64(Cadena)
21             Return 0
22         Catch ex As Exception
23             Return 1
24         End Try
25
26     End Function
27
28     Public Function MensajeDeError(ByVal Objeto As Object, Cadena As String, Longitud As Integer)
29         Objeto.Visible = True
30         Objeto.left = (1049 - Longitud) / 2
31         Objeto.text = Cadena
32         Return 0
33     End Function
34
35     'Difuminado Global
36     Public y As Integer
37     Public ObjetoAparecedor As Object
38
39     Private WithEvents ShowForm As New Windows.Forms.Timer With {.Interval = 10}
40     Public Function ShowFormStart(ByVal Objeto As Object)
41         ShowForm.Start()
42         ObjetoAparecedor = Objeto
43         y = -1
44         Return 0
45     End Function
46
47     Private Sub ShowForm_Tick(sender As Object, e As EventArgs) Handles ShowForm.Tick
48         y += 20
```

```

49
50         ObjetoAparecedor.Opacity = y / 100

51
52         If y >= 100 Then
53             ShowForm.Stop()
54         End If
55     End Sub
56     'Fin Difuminado Global
57
58     'Creacion Hora
59     Public Function getHora()
60         Dim Hora As String = DateTime.Now.ToString("HH")
61         Dim Min As String = Format(DateTime.TimeOfDay, "mm")
62         Dim Estatus As String = "-"
63
64         If Hora < 13 Then
65             Estatus = " am"
66         Else
67             Hora = Hora - 12
68             Estatus = " pm"
69
70             If CInt(Hora) < 10 Then
71                 Hora = "0" + Hora
72             End If
73
74         End If
75
76         Dim Resultado As String = Hora + ":" + Min + Estatus
77         Return Resultado
78     End Function
79
80     'Creacion Fecha
81     Public Function getDia()
82
83         Dim nDia As String = DateTime.Now.ToString("dd")
84         Dim Mes As String = DateTime.Now.ToString("MMM").Substring(0, 1).ToUpper
85             () + DateTime.Now.ToString("MMM").Substring(1)
86
87         Dim Year As String = DateTime.Now.ToString("yyyy") 86
88         Dim Fecha As String = nDia + "/" + Mes + "/" + Year 88
89
90             Dim dia As String = DateTime.Now.ToString("ddd").Substring(0, 1).ToUpper
91
92             Dim DiaTotal As String = Dia + " - " + Fecha
93
94         Return DiaTotal.Replace(".", "")

```

```

95      End Function

96
97      Public Sub Desabilita(ByVal Objeto)
98      Objeto.ForeColor = Color.DarkGray
99      Objeto.Cursor = Cursors.Hand
100     End Sub

101
102     Public Sub Habilita(ByVal Objeto)
103     Objeto.ForeColor = Color.Black
104     Objeto.Cursor = Cursors.No
105     End Sub

106
107     Public Sub PermaActiva(ByVal Objeto)
108     Objeto.enabled = False
109     End Sub

110
111     Public Sub Reinicia(ByVal Objeto)
112     Objeto.enabled = True
113     Objeto.Cursor = Cursors.Hand
114     End Sub

115
116     Public Function getClaveDia()
117     Dim Fecha As String = DateTime.Now.ToString("ddMMyy")
118     Return Fecha
119     End Function

120
121     Public Function getDiaEspecifico()
122     Dim DiaEspecifico As String = DateTime.Now.ToString("dddd").Substring(0,
        1).ToUpper() + DateTime.Now.ToString("dddd").Substring(1, DateTime.Now.ToString("dddd").Length
        - 1)

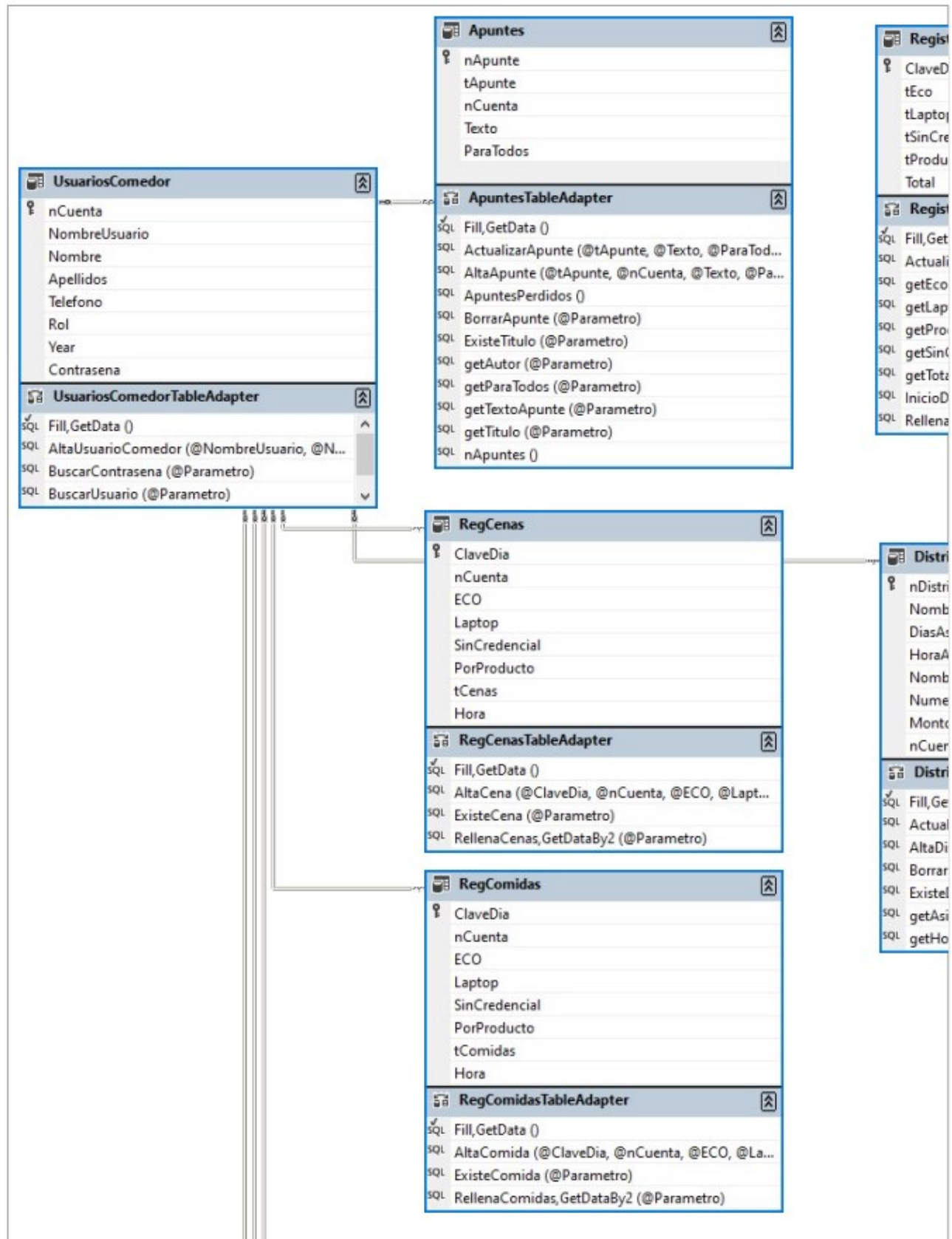
123
124     Return DiaEspecifico
125     End Function

126
127     Public Function getDiaConLetra()
128     Dim nDia As String = DateTime.Now.ToString("dd")
129     Dim Mes As String = DateTime.Now.ToString("MMM").Substring(0, 1).ToUpper
        () + DateTime.Now.ToString("MMM").Substring(1)

130     Dim Year As String = DateTime.Now.ToString("yyyy") 131
132
133     Return nDia + "/" + Mes + "/" + Year
134     End Function
135
136 End Module

```

Módulo de Bases de Datos (Esquema DataSet):



trosTotales
lia
p
edencial
cto
trosTotalesTableAdapter
Data ()
izarDia (@tEco, @tLaptop, @tSinCredencia...
(@Parametro)
top (@Parametro)
ducto (@Parametro)
Credencial (@Parametro)
sl (@Parametro)
liaRegistros (@ClaveDia)
sTotales,GetDataBy7 (@Parametros)

ibuidores
ibuidor
reDistribuidor
sistencia
sistencia
reEmpleado
roTelefonico
oAprox
nta
ibuidoresTableAdapter
tData ()
lizarDistribuidor (@NombreDistribuidor, ...
istribuidor (@NombreDistribuidor, @Dia...
Distribuidor (@Parametro)
Distribuidor (@Parametro)
istencia (@Parametro)
raAsistencia (@Parametro)

RegDesayunos	
	ClaveDia
	nCuenta
	ECO
	Laptop
	SinCredencial
	PorProducto
	tDesayunos
	Hora

RegDesayunosTableAdapter	
	Fill,GetData ()
	AltaDesayuno (@ClaveDia, @nCuenta, @ECO, @...
	ExisteDesayuno (@Parametro)
	RellenaDesayuno,GetDataBy2 (@Parametro)

RegistrosG	
	nRegistro
	ClaveDia
	nCuenta
	Motivo
	Monto
	Hora

RegistrosGTableAdapter	
	Fill,GetData ()
	AltaGasto (@ClaveDia, @nCuenta, @Motivo, @...
	FillBy,GetDataBy1 (@Parametro)
	getRegistrosGastos (@Parametro)

RegistrosV	
	nRegistro
	ClaveDia
	nCuenta
	Motivo
	Monto
	Hora

RegistrosVTableAdapter	
	Fill,GetData ()
	AltaVenta (@ClaveDia, @nCuenta, @Motivo, @M...
	FillBy,GetDataBy1 (@Parametro)
	getRegistrosVentas (@Parametro)

Regis	
	ClaveC
	Dia
	nomDi
	tGasto
	tVenta
	Balanc

Regis	
	Fill,Get
	Actual
	ExisteE
	getBal.
	getGas
	getVen
	IniciarI

trosGV
dia
ia
ie
trosGVTableAdapter
tData ()
izarDia (@tGasto, @tVenta, @Balance, @Pa...
dia (@Parametro)
ance (@Parametro)
sto (@Parametro)
ita (@Parametro)
Dia (@ClaveDia, @Dia, @nomDia, @tGasto...

15.- Pronóstico de Tiempos (Gráfica de Gantt)

Actividades	Semanas de Trabajo															
	Febrero				Marzo				Abril				Mayo			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1.- Lluvia de Ideas																
2.- Nombre del Proyecto y Nombre del Modelo																
3.- Objetivo del Proyecto																
4.- Justificación																
5.- Investigación																
6.- Requerimientos																
7.- Diagrama UML																
8.- Español Estructurado																
9.- Diccionario de Datos																
10.- Base de Datos																
11.- Interfaces																
12.- Herramientas Case																
13.- Polimorfismo																
14.- Conclusiones																
15.- Bibliografía																
16.- Anexos																