
REPORTE FINAL

para

Don Cuco's Little Store

**Realizado por
García García Jonathan Eduardo
García García Yessid Fernando
González Santiesteban Santiago
Guzmán Olvera Jessica
Partida Herrera Alma Karen**

7CV23

Junio 7,2021

Contenido

1	Introducción	4
1.1	Entorno Operativo	4
1.2	Requisitos y dependencias del software	4
1.3	Periféricos	4
2	Módulos	5
1.0	Módulo ABC2 Productos	5
1.1.0	Alta de Productos	6
1.2.0	Baja de Productos	8
1.3.0	Modificación de Productos	9
1.4.0	Entrada de Productos	11
1.5.0	Salida de Productos	13
2.0	Módulo de reportes	14
2.1.0	Reporte de movimientos	15
2.1.1	Reporte de existencias	16
2.1.2	Reporte de Ventas	17
2.1.3	Reporte de Existencia	18
3.0	Módulo de Seguridad	19
3.1.0	Control de Roles y Privilegios	20
3.1.1	Privilegio Reportes	21
3.1.2	Privilegio Entradas	22
3.1.3	Privilegio Salidas	23
3.1.4	Rol solo Lectura	24
3.2.0	Autenticación de Usuario (user/password)	25
3.3.0	ABC de Usuarios/Roles/Privilegios/Contraseñas	26
4.0	Módulo de Alertas	27
5.0	Módulo de Clientes	28
6.0	Módulo de Ventas	30
7.0	Pantalla principal	33
8.0	Diseños	34
8.1.0	Diseño lógico	34
8.1.1	Ejemplo de llamada a procedimiento	49
8.2.0	Diseño conceptual	51
9.0	Manual de usuario	52
9.1	Introducción	52
9.1.1	¿Cómo usar este manual?	52
9.2	Visión general de la aplicación	53
9.2.1	Entrada al sistema	53
9.3	Menú del sistema	54
9.3.1	Ventas	55

9.3.2	Productos	59
9.3.3	Clientes	61
9.3.4	Usuarios	62
9.3.5	Proveedores	63
9.3.6	Compras	64
9.3.7	Entrada/Salida	67
9.4	Reportes	69
9.5	Movimientos	71
9.6	Manual Técnico	71
9.6.1	Ejemplo de llamada a procedimiento	86

1 Introducción

Este documento es el reporte final para la materia de base de datos. Este contiene la Especificación de Requisitos Software (SRS por sus siglas en ingles). Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software.

Los requerimientos especifican qué es lo que el sistema debe hacer (sus funciones) y sus propiedades esenciales y deseables. La captura de los requerimientos tiene como objetivo principal la comprensión de lo que los clientes y los usuarios esperan que haga el sistema. Un requerimiento expresa el propósito del sistema sin considerar como se va a implantar. En otras palabras, los requerimientos identifican el qué del sistema, mientras que el diseño establece el cómo del sistema. El siguiente SRS está estructurado en cuatro módulos principales y un control de mermas especificados en el índice, cada requerimiento se detalla a continuación en la página y modulo correspondiente.

1.1 Entorno Operativo

1. Sistema operativo:
 - Windows 7 o superior
 - Arquitectura X_{86} o X_{64}

1.2 Requisitos y dependencias del software

- Microsoft .NET Framework 4.7

1.3 Periféricos

1. Teclado QWERTY (Recomendado)
2. Mouse ó ratón
3. Pantalla de al menos 11.5" x 7.9" x 0.33"
4. Impresora con soporte para hojas tamaño carta (Opcional)

2 Módulos

1.0 Módulo ABC2 Productos

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: ABC2 es un método que agiliza los procesos de almacenamiento de los productos. Aquí, se clasifican los productos por categoría.

Rol: Administración.

Interfaz/Interacción con el usuario

Se tiene una ventana con los siguientes componentes:

1. **Enlaces o menú de productos:** Se tiene un acceso en la ventana principal hacía las demás interfaces:
 - Ventas
 - Productos
 - Clientes
 - Usuarios
 - Proveedores
 - Compras
 - Entrada
 - Salida
 - Cerrar sesión
2. **Botón Cerrar:** Es el botón que cierra la ventana. Esta acción puede ser por medio de un botón o realizarse desde una equis (x) en el extremo superior derecho de la ventana actual.

Requerimientos funcionales

1. Validar que el usuario seleccione una opción de enlace válida, y te lleve hacía la interfaz correcta, o en su caso, si el usuario selecciona cerrar la pestaña, se cierre correctamente la ventana de ABC2 de Productos.
2. Enlaces dentro de ABC2 de Productos.
3. Que al presionar el botón de cerrar, se cierre la pestaña sin ningún problema.

ABC2 de Productos	Alta de productos
	Baja de productos
	Modificación de productos
	Entrada de producto
	Salida de producto

1.1.0 Alta de Productos

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: El alta de productos, sirve para agregar los productos nuevos que ingresan por primera vez al inventario. Al darlos de alta, deben estar disponibles para todas las operaciones que los involucran como baja de productos, modificación de productos, entrada de productos, salida de productos.

Rol: Administración.

Interfaz/Interacción con el usuario

Se tiene una ventana con los siguientes componentes:

1. **Campo de id:** Este campo permitirá tener un identificador único a cada producto ingresado al inventario.
2. **Nombre de producto:** Campo de texto. Permitirá el ingreso del nombre del producto que se está dando de alta en el inventario.
3. **Descripción:** Campo de texto. Permitirá el ingreso de una descripción al producto que se está dando de alta en el inventario.
4. **Adjuntar imagen:** Se usará, para poder agregar una imagen del producto que se está dando de alta en el inventario.
5. **Proveedor:** Campo de texto. Servirá para poder ingresar el nombre del proveedor del producto.
6. **Botón de confirmación:** Servirá para poder confirmar que se está de acuerdo con los datos ingresados y se procederá a guardar el producto en el inventario. Dar un clic significará aceptar.
7. **Botón Cerrar:** Es el botón que cierra la ventana. Esta acción puede ser por medio de un botón o realizarse desde una equis (x) en el extremo superior derecho de la ventana actual.

Requerimientos funcionales

1. Validar que la información ingresada sea correcta. En caso de presentar algún error, se debe mostrar un cuadro de diálogo con el mensaje según sea el caso:
 - El nombre del producto esta vacío.
 - La descripción del producto esta vacía.
 - No se adjunto ninguna imagen del producto
 - El nombre del proveedor esta vacío.
2. Verificar que los productos ingresados aquí, se suban correctamente en la base de datos del sistema.
3. Validar campos de productos:
 - Eliminar espacios al frente y al final que no sean necesarios
 - Que no contenga simbolos ni caracteres especiales
 - Si se adjunta una imagen verificar que sea un formato de imagen valido
4. Verificar al presionar el botón de cerrar, se cierre la pestaña sin ningún problema.
5. Validar no este registrado ningun producto con el Id o nombre, en caso de que ya se encuentre registrado:
 - Mostrar un mensaje con el texto "Ya está registrado este producto en el sistema."

1.2.0 Baja de Productos

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: La baja de productos, sirve para quitar los productos necesarios, anteriormente ingresados, al inventario. Al darlos de baja, ya no deben de estar visibles para todas las operaciones que los involucran como: alta de productos, modificación de productos, entrada de productos.

Rol: Administración.

Interfaz/Interacción con el usuario

Se tiene una ventana con los siguientes componentes:

1. **Cuadro de búsqueda:** Es un cuadro de búsqueda de productos, que permitirá realizar una búsqueda dentro de todo el inventario, remarcando y mostrando los que coincidan con la búsqueda, o mostrar un mensaje donde indique que no coincide ninguna búsqueda con lo ingresado.
 - **Botón de Ordenar:** Es un filtro que permitirá ordenar los productos en el sistema, lo hará por todos los campos disponibles, id, alfabéticamente, según el nombre del proveedor.
2. En una tabla se muestran los siguientes campos:

Código de producto	Nombre de producto	Categoría del producto	Proveedor	Casilla de selección
Alfanumérico	Texto	Texto	Número flotante	Verdadero/Falso

3. **Casilla de selección:** Permitirá elegir algún, o algunos productos para posteriormente darlos de baja
4. **Cuadro de Diálogo:** Servirá para poder confirmar que se está de acuerdo con el o los productos que se desean dar de baja en ese momento. Se mostrará "¿Desea dar de baja (el nombre del o los productos) seleccionado(s)? y dos opciones de botón, aceptar y cancelar. Al aceptar, se procederá a dar de baja el o los productos en el inventario. Al cancelar, se cerrará el cuadro de diálogo.
5. **Botón Cerrar:** Es el botón que cierra la ventana. Esta acción puede ser por medio de un botón o realizarse desde una equis (x) en el extremo superior derecho de la ventana actual.
6. **Botón Ayuda:** Es el botón que te ofrece ayuda contextual del manual del usuario, necesaria para aprender a usar el Sistema de Inventario.

Requerimientos funcionales

1. Verificar que él, o los productos dados de baja, no aparezcan visibles en todas las operaciones que los involucran como en alta de productos, modificación de productos, entrada de productos.
2. Verificar al presionar el botón de cerrar, se cierre la pestaña sin ningún problema.

1.3.0 Modificación de Productos

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: La modificación de productos sirve para hacer cambios en los datos ingresados a los productos en el inventario. Al hacer una modificación, esta debe mostrarse ya actualizada correctamente en todas las operaciones que los involucran como: alta de productos, baja de productos, entrada de productos, salida de productos.

Rol: Administración

Interfaz/Interacción con el usuario

Se tiene una ventana con los siguientes componentes:

1. **Cuadro de búsqueda:** Es un cuadro de búsqueda de productos, que permitirá realizar una búsqueda dentro de todo el inventario, remarcando y mostrando los que coincidan con la búsqueda, o mostrar un mensaje donde indique que no coincide ninguna búsqueda con lo ingresado.
 - **Botón de Ordenar:** Es un filtro que permitirá ordenar los productos en el sistema, lo hará por todos los campos disponibles, id, alfabéticamente, según el nombre del proveedor.
2. **Campo de id:** Este campo no se podrá modificar, pues es un número asignado únicamente para ese producto.
3. **Campo de texto:** Permitirá modificar el ingreso del nombre del producto que se dio de alta en el inventario.
4. **Campo de texto:** Permitirá modificar el ingreso de una descripción al producto que se dio de alta en el inventario.
5. **Adjuntar imagen:** Se usará, para poder modificar la imagen agregada del producto que se dio de alta en el inventario.
6. **Campo de texto:** Servirá para poder editar el nombre del proveedor del producto.
7. **Botón de confirmación:** Servirá para poder confirmar que se está de acuerdo con los cambios ingresados y se procederá a guardar el producto en el inventario. Dar un clic significará aceptar.
8. **Cuadro de búsqueda:** Es un cuadro de búsqueda de productos, que permitirá realizar una búsqueda dentro de todo el inventario, remarcando y mostrando los que coincidan con la búsqueda, o mostrar un mensaje donde indique que no coincide ninguna búsqueda con lo ingresado.
9. **Botón Cerrar:** Es el botón que cierra la ventana. Esta acción puede ser por medio de un botón o realizarse desde una equis (x) en el extremo superior derecho de la ventana actual.

Requerimientos funcionales

1. Verificar que las modificaciones realizadas, se actualicen y visualicen correctamente en la base de datos del sistema.
2. Validar campos de productos:
 - Eliminar espacios al frente y al final que no sean necesarios
 - Que no contenga símbolos ni caracteres especiales
 - Si se adjunta una imagen verificar que sea un formato de imagen válido
3. Verificar al presionar el botón de cerrar, se cierre la pestaña sin ningún problema.

1.4.0 Entrada de Productos

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: En este módulo se ingresan nuevas cantidades de producto al inventario.

Rol: Administración

Interfaz/Interacción con el usuario

Se tiene una ventana con los siguientes componentes:

1. **Cuadro de búsqueda:** Es un cuadro de búsqueda de productos, que permitirá realizar una búsqueda dentro de todo el inventario, remarcando y mostrando los que coincidan con la búsqueda, o mostrar un mensaje donde indique que no coincide ninguna búsqueda con lo ingresado.
 - **Botón de Ordenar:** Es un filtro que permitirá ordenar los productos en el sistema, lo hará por todos los campos disponibles, id, alfabéticamente, según el nombre del proveedor.
2. **Campo de código:** Este campo permite tener una visualización de un identificador único a cada producto ingresado al inventario.
3. **Campo de texto:** Permitirá visualizar el nombre del producto, que se dio de alta en el inventario.
4. **Campo de texto:** Servirá para visualizar el número de los productos actuales en sistema.
5. **Campo de texto:** Servirá para visualizar el número posterior al movimiento de los productos actuales en sistema.
6. **Botón Cerrar:** Es el botón que cierra la ventana. Esta acción puede ser por medio de un botón o realizarse desde una equis (x) en el extremo superior derecho de la ventana actual.

Requerimientos funcionales

1. Validar que la información ingresada sea correcta. En caso de presentar algún error, se debe mostrar un cuadro de diálogo con el mensaje según sea el caso:
 - La cantidad de ingreso esta vacía.
 - No se ha ingresado una fecha y hora de entrada de productos.
 - No se ingresó un número de lote de los productos, ¿desea continuar?
2. Verificar que las cantidades y los datos ingresados aquí, se suban correctamente en la base de datos del sistema y esta se actualice.
3. Verificar que la fecha sea una fecha preferentemente del día actual. Verificar la hora coincida con la hora actual.
4. Verificar que ninguna de las opciones anteriores marque error y cada una realice lo que se indicó que debe hacer.

5. Verificar al presionar el botón de cerrar, se cierre la pestaña sin ningún problema.
6. Verificar que el botón de ayuda, muestre la ayuda requerida en la ventana de “Entrada de productos”:

1.5.0 Salida de Productos

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: En este módulo se disminuirán las cantidades de producto que se tienen en inventario.

Rol: Administración

Interfaz/Interacción con el usuario

Se tiene una ventana con los siguientes componentes:

1. **Cuadro de búsqueda:** Es un cuadro de búsqueda de productos, que permitirá realizar una búsqueda dentro de todo el inventario, remarcando y mostrando los que coincidan con la búsqueda, o mostrar un mensaje donde indique que no coincide ninguna búsqueda con lo ingresado.
 - **Botón de Ordenar:** Es un filtro que permitirá ordenar los productos en el sistema, lo hará por todos los campos disponibles, id, alfabéticamente
2. **Campo de texto:** Permitirá visualizar el nombre del producto, que se dio de alta en el inventario.
3. **Selector de concepto:** Servirá para poder escribir el motivo por el cual se está dando salida del inventario, el o los productos.
4. **Campo de texto:** Permitirá visualizar las observaciones sobre el producto
5. **Botón Cerrar:** Es el botón que cierra la ventana. Esta acción puede ser por medio de un botón o realizarse desde una equis (x) en el extremo superior derecho de la ventana actual.

Requerimientos funcionales

1. Verificar que él, o los productos dados salida, no aparezcan en ninguna ventana ni en la base de datos del sistema de inventario.
2. Verificar al presionar el botón de cerrar, se cierre la pestaña sin ningún problema.
3. Verificar que el botón de ayuda, muestre la ayuda requerida en la ventana de "Salida de productos":

2.0 Módulo de reportes

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: Este módulo se encarga de presentar la información contenida en el sistema de forma ordenada en formato pdf.

Rol: Administración.

Interfaz/Interacción con el usuario

Ventana para seleccionar el rango de fechas

Se tiene una ventana con los siguientes componentes:

1. **Título:** Nuevo reporte
2. **Campo tipo fecha** con el formato *dd/mm/yyyy*, en la parte superior de este campo se mostrara el texto "Fecha inicial"
3. **Campo tipo fecha** con el formato *dd/mm/yyyy*, en la parte superior de este campo se mostrara el texto "Fecha final"
4. **Botón de confirmación**, un botón en color verde con el texto "Confirmar" que ejecutara la acción de generar^[1] y abrir^[2] el reporte

Requerimientos funcionales

1. Se impide que sean seleccionados rangos de fecha inválidos como son los siguientes casos:
 - La fecha inicial es mayor que la fecha final
 - La fecha final es un día que aún no llega
2. El reporte en formato pdf se abre en la aplicación predeterminada del sistema ^[2]
3. El reporte se genera de la siguiente forma: ^[1]
 - La plantilla de reporte contiene los siguientes datos:
 - **Título:** [Se especifica según el tipo de reporte]
 - **Subtítulo:** Rango de fechas con el formato: (Fecha inicial) - (Fecha final) , ambas con el formato: *dd/mm/yy*
 - **Contenido:** Tabla que muestra la siguiente información que esta entre el rango de fechas seleccionadas ordenada por número de movimiento creciente.
El contenido de esta tabla varia según el tipo de reporte

2.1.0 Reporte de movimientos

Tipo de requerimiento: No Funcional

Prioridad: Alta

Descripción: Permite ver al usuario los movimientos de entrada y salida que se han efectuado en el sistema entre un rango de fechas.

Rol: Administración.

Interfaz/Interacción con el usuario

Al seleccionar este reporte se muestra la **ventana para seleccionar el rango de fechas** especificada en el punto **2.0 "Módulo de reportes"**

Requerimientos funcionales

1. **Título:** Reporte de movimientos
2. Contenido de la tabla de reporte

Fecha	Usuario	Operación	Tipo	Tipo de movimiento
Date	Texto	Texto	Texto	Texto

2.1.1 Reporte de existencias

Tipo de requerimiento: No Funcional

Prioridad: Alta

Descripción: Permite ver al usuario un resumen total de los productos en el inventario.

Rol: Administración.

Interfaz/Interacción con el usuario

Al seleccionar este reporte se muestra la **ventana para seleccionar el rango de fechas** especificada en el punto **2.0 "Módulo de reportes"**

Requerimientos funcionales

1. **Título:** Reporte de entradas
2. Contenido de la tabla de reporte

Código	Nombre	Existencia	Mínimo	Máximo	Status
Alfanumerico	Texto	Alfanumerico	Alfanumerico	Alfanumerico	Alfanumerico

2.1.2 Reporte de Ventas

Tipo de requerimiento: No Funcional

Prioridad: Alta

Descripción: Permite ver al usuario los movimientos de venta que se han efectuado en el sistema entre un rango de fechas.

Rol: Administración.

Interfaz/Interacción con el usuario

Al seleccionar este reporte se muestra la **ventana para seleccionar el rango de fechas** especificada en el punto **2.0 "Módulo de reportes"**

Requerimientos funcionales

1. **Título:** Reporte de salidas
2. Contenido de la tabla de reporte

Fechas	Ticket	Vendedor	Cliente	Importe	Impuesto	Total
Date	Alfanumérico	Texto	Numérico flotante	Texto	Numérico flotante	Numérico flotante

2.1.3 Reporte de Existencia

Tipo de requerimiento: No Funcional

Prioridad: Alta

Descripción: Permite ver al usuario la existencia actual de cada producto ordenado por linea.

Rol: Administración.

Interfaz/Interacción con el usuario

Al seleccionar este reporte se muestra la **ventana para seleccionar el rango de fechas** especificada en el punto **2.0 "Módulo de reportes"**

Requerimientos funcionales

1. **Título:** Reporte de existencia
2. Contenido de la tabla de reporte

Fecha ultimo movimiento	Código de producto	Nombre de producto	Categoría del producto	Existencia actual
Fecha	Alfanumérico	Texto	Texto	Número flotante

3.0 Módulo de Seguridad

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: Módulo Seguridad es un método que da acceso al “ABC de Usuarios/Roles/Privilegios/Contraseñas” para regular el registro de usuarios y el rol a desempeñar con privilegios definidos por el administrador. Aquí, se clasifican los roles y privilegios a desempeñar por el usuario base.

Rol: Administración.

Interfaz/Interacción con el usuario

Se tiene una ventana con los siguientes componentes:

1. **Registro o modificación:** Se tiene un acceso hacia las demás interfaces:
 - ABC de Usuarios/Roles/Privilegios/Contraseñas
 - Autenticación de Usuario (user/password)
2. **Visualizador de usuarios base:** Sirve para poder visualizar a todos los usuarios registrados con sus roles y privilegios.
3. **Botón Terminar:** Es el botón que cierra la ventana. Esta acción puede ser por medio de un botón o realizarse desde una equis (x) en el extremo superior derecho de la ventana actual.
Requerimientos funcionales

Requerimientos funcionales

1. Validar que el administrador ingrese un nuevo usuario o modifique algún rol o privilegio lleve hacia la interfaz correcta, o en su caso, si el Administrador selecciona cerrar la pestaña, se cierre correctamente la ventana de Módulo de Seguridad.
2. Enlaces dentro de Módulo Seguridad

Módulo de Seguridad	ABC de Usuarios/Roles/Privilegios/Contraseñas
	Autenticación de Usuario (user/passwd)
	Control de Roles y Privilegios

3. Que al presionar el botón de cerrar, se cierre la pestaña sin ningún problema.

3.1.0 Control de Roles y Privilegios

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción:Asignación de roles y privilegios a los usuarios base.

Rol: Administración.

Interfaz/Interacción con el usuario

Esta interfaz se especifica mas adelante en el punto **"3.3.0 ABC de Usuarios/Roles/Privilegios/Contraseñas"**

Requerimientos funcionales

1. 1. Validar que la información ingresada sea correcta. En caso de presentar algún error, se debe mostrar un cuadro de diálogo con el mensaje según sea el caso:
 - El nombre del usuario no existe.
 - Rol vacío
 - Sin privilegios (Mensaje diciendo si desea dejar a este usuario sin privilegios)
2. Asignar en la base los nuevos privilegios y rol del usuario además de en caso de haber desmarcado algún privilegio bórralo de la base de ese usuario.

3.1.1 Privilegio Reportes

Tipo de requerimiento: No Funcional

Prioridad: Media

Descripción: Determina si el usuario puede ó no acceder a el módulo de reportes

Rol: Administración/Usuario.

Interfaz/Interacción con el usuario

1. En caso de no contar con este privilegio al momento de intentar acceder al módulo de reportes se muestra el siguiente mensaje:
 - **Título:** Acceso denegado
 - **Texto:** No tiene autorización para usar este módulo
 - **Imagen:** Icono de advertencia amarillo

Requerimientos funcionales

1. Validar que el usuario cuente con este privilegio al intentar acceder a el módulo de reportes

3.1.2 Privilegio Entradas

Tipo de requerimiento: No Funcional

Prioridad: Media

Descripción: Determina si el usuario puede ó no ingresar una entrada de inventario al almacén.

Rol: Administración/Usuario.

Interfaz/Interacción con el usuario

1. En caso de no contar con este privilegio al momento de intentar acceder al módulo de entradas se muestra el siguiente mensaje:
 - **Título:** Acceso denegado
 - **Texto:** No tiene autorización para usar este módulo
 - **Imagen:** Icono de advertencia amarillo

Requerimientos funcionales

1. Validar que el usuario cuente con este privilegio al intentar acceder a el módulo de entradas

3.1.3 Privilegio Salidas

Tipo de requerimiento: No Funcional

Prioridad: Media

Descripción: Determina si el usuario puede ó no acceder a el módulo de salidas

Rol: Administración/Usuario.

Interfaz/Interacción con el usuario

1. En caso de no contar con este privilegio al momento de intentar acceder al módulo de reportes se muestra el siguiente mensaje:
 - **Título:** Acceso denegado
 - **Texto:** No tiene autorización para usar este módulo
 - **Imagen:** Icono de advertencia amarillo

Requerimientos funcionales

1. Validar que el usuario cuente con este privilegio al intentar acceder a el módulo de salidas

3.1.4 Rol solo Lectura

Tipo de requerimiento: No Funcional

Prioridad: Media

Descripción: Determina si el usuario puede ó no generar tanto entradas como salidas.

Rol: Administración/Usuario.

Interfaz/Interacción con el usuario

1. Si este rol esta activado el usuario no podra generar ninguna entrada ó salida al momento de intentar acceder al módulo de reportes se muestra el siguiente mensaje:
 - **Título:** Acceso denegado
 - **Texto:** No tiene autorización para usar este módulo
 - **Imagen:** Icono de advertencia amarillo

Requerimientos funcionales

1. Validar que el usuario cuente con este privilegio al intentar generar alguna entrada ó salida
2. Si este rol esta activo los permisos **3.1.2** y **3.1.3** quedan invalidados en falso

3.2.0 Autenticación de Usuario (user/password)

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: Método que regula el acceso a ciertas acciones mencionadas mas adelante.

Rol: Administración/Usuario.

Interfaz/Interacción con el usuario

Se tiene una ventana con los siguientes componentes:

1. **Campo Usuario:** Permitirá el ingreso de id del administrador para autenticar el accesos al ABC de usuario
2. **Campo contraseña:** Ingreso de contenido alfanumérico correspondiente al administrador
3. **Botón Nuevo-Usuario:** Habilita ventana para ingreso de datos del nuevo usuario base
4. **Botón confirmación:** para confirmar cambios del usuario

Al acceder el programa mostrara cuál de estos privilegios o roles tiene habilitado el usuario en la parte inferior de la pantalla principal del sistema en color "Violeta oscuro" con hexadecimal "FF650075", el texto en color blanco con el formato: "Rol: [Nombre del rol]".

- **Rol lectura:** para solo ver la documentación
- **Privilegio Entrada:** Poder validar las entradas de los productos ya validado por ABC2 de Productos
- **Privilegio Salida:** Poder validar salida de productos ya validado por ABC2 de Productos
- **Privilegio Reportes:** Permiso para poder levantar reportes de salida o entrada de productos devolución o cancelación del mismo.

Requerimientos funcionales

1. Validar que la información ingresada sea correcta. En caso de presentar algún error, se debe mostrar un cuadro de diálogo con el mensaje según sea el caso:
 - El nombre del usuario no existe
 - El campo contraseña está vacío.
 - Contraseña no valida
2. Verificar que los datos ingresados aquí, se comparen con la base de datos del sistema para así validar privilegios y rol del usuario.
3. Verificar que los campos cuenten con lo requerido en cuanto a espacio o formatos.

3.3.0 ABC de Usuarios/Roles/Privilegios/Contraseñas

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: Método que regula el registro de usuarios y el rol a desempeñar con privilegios definidos por el administrador. Aquí, se clasifican los roles y privilegios a desempeñar por el usuario base.

Rol: Administración.

Interfaz/Interacción con el usuario

Se tiene una ventana con los siguientes componentes:

1. **Campo usuario:** Nombre del nuevo usuario
2. **Campo contraseña:** Contraseña del nuevo usuario
3. **Campo nombre :** Permitirá el ingreso de nombre del nuevo usuario base establecido por el administrador
4. **Menú de Checkbox :** Asignados a cada rol o privilegio para habilitarlos o deshabilitarlos
 - Rol lectura
 - Rol lectura/escritura
 - Privilegio Entrada
 - Privilegio Salida
 - Privilegio Reportes
5. **Botón confirmación:** para confirmar cambios del usuario
6. **Botón cerrar:** Para que así el administrador pueda proseguir al haber terminado de modificar o dar de alta un nuevo usuario

Requerimientos funcionales

1. Validar que la información ingresada sea correcta. En caso de presentar algún error, se debe mostrar un cuadro de diálogo con el mensaje según sea el caso:
 - El nombre del usuario ya asignado.
 - El campo contraseña está vacío.
 - Contraseña no valida
2. Verificar que los nuevos usuarios ingresados aquí, se suban correctamente en la base de datos del sistema.
3. Verificar que los campos cuenten con lo requerido en cuanto a espacio o formatos.

4.0 Módulo de Alertas

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Media

Descripción: Este módulo se encarga de presentar en la pantalla los mensajes de advertencia que estos puedan llegar a tener.

Rol: Administración/Usuario.

Interfaz/Interacción con el usuario

En la ventana principal se muestra una ventana emergente con el texto en letras blancas y la leyenda: "[ALERTA]" seguida de el movimiento que se está realizando para dar aviso y generar una confirmación:

5.0 Módulo de Clientes

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: El módulo de clientes es donde se puede registrar, borrar o ver los clientes existentes, con el fin de agilizar los procesos requeridos.

Rol: Administración.

Interfaz/Interacción con el usuario

Se tiene una ventana con los siguientes componentes:

1. Imagen de cliente: Es un campo que sirve para ingresar una imagen para poder trabajar con un cliente más fácilmente:
2. Nombre: Es un campo de texto que sirve para ingresar el nombre del cliente, el cual vamos a registrar.
3. Notas: Es un cuadro de texto en donde se puede agregar alguna nota de importancia sobre el cliente que se está registrando.
4. Botón Cancelar: Es un botón que anula el proceso de registro de cliente, que se está llevando a cabo.
5. Botón de Confirmar: Es un botón que ratifica la información introducida del cliente, y la guarda en la Base de Datos.
6. Botón Cerrar: Es el botón que cierra la ventana actual. Esta acción se realiza por medio de una equis (x) en el extremo superior derecho de la ventana actual.

Requerimientos funcionales

1. Validar que el administrador al momento de ingresar los datos del cliente, (foto, nombre, notas), se guarde correctamente en el registro de clientes.
2. Validar que "nombre" sea un dato necesario a ingresar, al momento de registrar un cliente de acuerdo a la siguiente lista:
 - En caso de estar vacío este campo, mostrar un cuadro de diálogo con un mensaje indicando que falta introducir el dato.
 - Validar que el "nombre" no contenga símbolos ni caracteres especiales, a excepción de letras con acentos y la letra ñ.

- Validar que no esté registrado ningún otro usuario con el mismo nombre. En caso de que ya se encuentre registrado mostrar un mensaje con el texto “Ya está registrado este usuario en el sistema”.
3. Validar que “imagen de cliente” puede quedar vacía al hacer el registro de un cliente, y no sea necesaria para poder guardar el registro de un cliente. Si se adjunta una imagen, verificar que se esté adjuntando un formato de imagen válido.
 4. Validar que “nota” pueda quedar vacía al hacer el registro de un cliente, y no sea necesaria para poder guardar el registro de un cliente.
 5. Validar que se pueda hacer la modificación del registro de un cliente ya registrado anteriormente
 6. Validar que, al presionar el botón de cancelar, se anule el proceso y se cierre la ventana en la que se estaba trabajando
 7. Validar que, al presionar el botón de cerrar, se cierre la pestaña sin ningún problema.
 8. Validar que la información ingresada del registro de un cliente, se guarde correctamente en la Base de Datos.

6.0 Módulo de Ventas

Tipo de requerimiento: Funcional / Interfaz

Prioridad: Alta

Descripción: El módulo de clientes es donde se puede registrar una venta que realizará la tienda de Don Cuco.

Rol: Administración/Usuario.

Interfaz/Interacción con el usuario

Se tiene una ventana la cual estará dividida en dos partes:

En la mitad izquierda estarán unos accesos rápidos de categorías de los productos que se venden en la tienda. Al dar clic a la categoría de elección, esta mitad de pantalla cambiará para mostrar los productos que se encuentran dentro de la categoría previamente seleccionada. Al estar dentro de una categoría seleccionada, en la parte superior izquierda, habrá un botón de diga “volver a las categorías” para regresar al menú de categorías.

En la otra mitad de la pantalla que se tiene del lado derecho, se muestran los siguientes componentes:

1. **Cliente:** Es una lista desplegable, en la cual se puede seleccionar el cliente a quien se le esté realizando una venta. Para hacer más fácil la búsqueda de un cliente ya existente, hay un botón de “lupa” del lado derecho, en donde al dar clic, se puede ingresar el nombre del cliente a buscar. En caso de no existir el nombre en la Base de Datos, se puede agregar un nuevo cliente, dando clic en el botón “más”.
2. **Vendedor:** Sirve para seleccionar el nombre del vendedor que en ese momento esté realizando una venta.
3. **Etiquetas:** Sirve para categorizar los productos y obtener una mejor visualización de ellos
4. **Venta:** Es una sección en donde al seleccionar un producto (de la otra mitad que se mencionó anteriormente), la información de dicho producto se verá reflejada en esta sección. En caso de no querer seleccionar por medio del menú que se muestra en la otra mitad de la pantalla, se cuenta con un campo “search” para poder buscar el producto por su nombre. La búsqueda irá filtrando lo que encuentre registrado en la base de datos con la palabra o las palabras ingresadas. cuenta con los campos de: Imagen, Nombre, Precio, Cantidad y Total. Los campos de nombre, precio, cantidad y total tienen un filtro para poder acomodar los productos por el orden deseado, de mayor a menor o alfabéticamente según corresponda y según se seleccione.
Al hacer una compra de más de 4 productos, estos se mostrarán por “páginas” identificadas al final de la lista, con color azul la que esté seleccionada en ese momento.
En el campo de “cantidad” habrá un botón de “menos” rojo y un botón de “más” verde, el cual nos ayudará a agilizar el proceso de una venta de más de 1 producto, así como también habrá un cuadro enfrente en donde se muestra dicha cantidad, en el cual también podrá

ingresarse manualmente con el teclado la cantidad del producto deseada para la venta. Al final de “venta” se mostrará en todo momento el total en \$ (MXN) de los productos que se han registrado en la venta.

5. Botón Cobrar: Es un botón que confirma los datos de la venta que se está realizando.

Requerimientos funcionales

1. 1. Validar que, al momento de dar clic en la lista desplegable de clientes, muestre el nombre de un cliente previamente ya registrado. De igual forma validar que el nombre se encuentre al hacer una búsqueda en el botón de búsqueda.

Si el cliente no está registrado y se registra en ese momento, validar que haga correctamente el registro, tomando en cuenta que:

- “nombre” sea un dato necesario a ingresar, al momento de registrar un cliente. En caso de estar vacío este campo, mostrar un cuadro de diálogo con un mensaje indicando que falta introducir el dato.
- Validar que el “nombre” no contenga símbolos ni caracteres especiales, a excepción de letras con acentos y la letra ñ.
- Validar que no esté registrado ningún otro usuario con el mismo nombre. En caso de que ya se encuentre registrado mostrar un mensaje con el texto “Ya está registrado este usuario en el sistema”.
- Validar que “imagen de cliente” puede quedar vacía al hacer el registro de un cliente, y no sea necesaria para poder guardar el registro de un cliente. Si se adjunta una imagen, verificar que se esté adjuntando un formato de imagen válido.
- Validar que “nota” pueda quedar vacía al hacer el registro de un cliente, y no sea necesaria para poder guardar el registro de un cliente.
- Validar que se pueda hacer la modificación del registro de un cliente ya registrado anteriormente.
- Validar que, al presionar el botón de cancelar, se anule el proceso y se cierre la ventana en la que se estaba trabajando
- Validar que, al presionar el botón de cerrar, se cierre la pestaña actual de registro de cliente, sin ningún problema.

- Validar que la información ingresada del registro de un cliente, se guarde correctamente en la Base de Datos.
2. Validar que, al seleccionar un vendedor, se muestre en pantalla dicho nombre del vendedor quien está realizando la venta.
 3. Validar que los productos que se seleccionen, aparezcan en la sección de “venta”.
 4. Validar que el cuadro de “search” haga una búsqueda de cualquier producto deseado, que exista dentro de la Base de Datos.
 5. Validar que “nombre, precio, cantidad y total” hagan el filtrado que se seleccione según la preferencia indicada por el usuario.
 6. Validar que en “cantidad” se aumenten o disminuyan correctamente las cantidades deseadas. Así mismo, verificar que, en caso requerido, también se pueda ingresar la cantidad deseada en el cuadro de texto directamente.
 7. Validar que todos los productos agregados se visualicen en “venta”, no importando la cantidad de productos, puesto que como se mencionó anteriormente, al ser más de 4 productos en una venta, cada 4 productos se irán creando páginas nuevas.
 8. Validar que “total” muestre la cantidad en MXN, de todos los productos que se están agregando a la venta.
 9. Validar que el botón “cobrar” registre correctamente la venta, y los cambios correspondientes, se vean reflejados en la Base de Datos.
 10. Validar que, al estar en las categorías generales de los productos, se pueda ingresar a cualquiera de ellos.
 11. Validar que, al estar en alguna categoría específica, al hacer clic en el botón de “volver a categorías” te regrese a la ventana de categorías generales.

7.0 Pantalla principal

Tipo de requerimiento: Interfaz

Prioridad: Alta

Descripción: Desde esta pantalla se tiene acceso a todos los modulos del sistema.

Rol: Administración/Usuario.

Requerimientos de Interfaz

1. Conjunto de acciones para productos
 - Botón de acceso a **Alta**
 - Botón de acceso a **Baja**
 - Botón de acceso a **Modificar**
 - Botón de acceso a **Entrada**
 - Botón de acceso a **Salida**
2. Conjunto de acciones para usuarios
 - Botón de acceso a **Usuarios, Clientes, Provedores**
 - Botón de acceso a **Cerrar sesión**
3. Apartado de ventas
 - Matriz de productos con **Nombre, categorias y descuentos**
 - Interfaz de cobro con **Cliente, Vendedor, Etiquetas**
 - Tabla de registro de la venta en proceso

Desde esta pantalla se puede activar el modo oscuro

8.0 Diseños

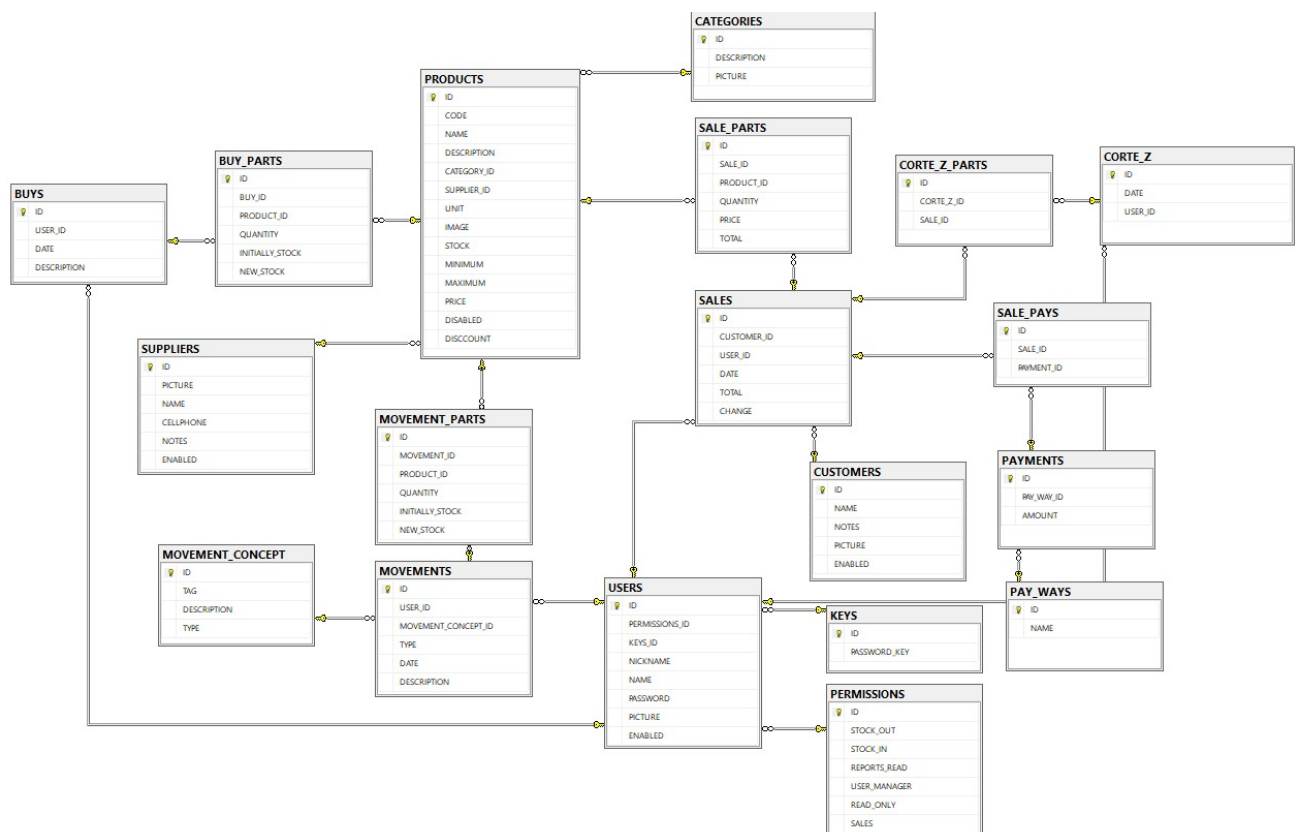
El diseño del proyecto es el proceso de elaboración de la propuesta de trabajo de acuerdo a pautas y procedimientos sistemáticos como ya se mencionó, un buen diseño debe identificar a los beneficiarios y actores claves; establecer un diagnóstico de la situación problema; definir estrategias posibles para enfrentarla y la justificación de la estrategia asumida; objetivos del proyecto (generales y específicos); resultados o productos esperados y actividades y recursos mínimos necesarios.

Al mismo tiempo, la propuesta o diseño debe contemplar la definición de indicadores para realizar el seguimiento y verificación de los resultados que se obtienen, y establecer los factores externos que garantizan su factibilidad y éxito.

Cada uno de los conceptos mencionados: objetivos; estrategia; resultados; productos, actividades, recursos, indicadores y factores externos, se irán describiendo y analizando por separado, a lo largo del desarrollo del documento y en la medida que avancemos en la elaboración del proyecto.

8.1.0 Diseño lógico

El diseño lógico es el proceso de construir un esquema de la información que utiliza la empresa, basándose en un modelo de base de datos específico, independiente del SGBD concreto que se vaya a utilizar y de cualquier otra consideración física.



```

1  /* CREA UNA BASE DE DATOS EN BLANCO*/
2  CREATE DATABASE DON_CUCO
3  GO
4  /* UTILIZA ESTA NUEVA BASE DE DATOS*/
5  USE DON_CUCO
6  GO
7  /*ALMACENA LA INFORMACIÓN SOBRE LAS CATEGORIAS DE PRODUCTOS*/
8  CREATE TABLE CATEGORIES
9  (
10     ID INT PRIMARY KEY IDENTITY(1,1),
11     DESCRIPTION VARCHAR(100) NOT NULL UNIQUE,
12     PICTURE VARCHAR(MAX)
13 );
14 /*INFORMACIÓN DE PROVEEDORES*/
15 CREATE TABLE SUPPLIERS(
16     ID INT PRIMARY KEY IDENTITY(1,1),
17     PICTURE VARCHAR(MAX),
18     NAME VARCHAR(100) NOT NULL,
19     CELLPHONE VARCHAR(100),
20     NOTES VARCHAR(MAX),
21     ENABLED BIT NOT NULL DEFAULT 0
22 );
23
24 /*INFORMACIÓN DE PRODUCTOS*/
25 CREATE TABLE PRODUCTS(
26     ID INT PRIMARY KEY IDENTITY(1,1),
27     CODE VARCHAR(100) UNIQUE NOT NULL,
28     NAME VARCHAR(100) NOT NULL,
29     DESCRIPTION VARCHAR(MAX),
30     CATEGORY_ID INT FOREIGN KEY REFERENCES CATEGORIES NOT NULL,
31     SUPPLIER_ID INT FOREIGN KEY REFERENCES SUPPLIERS NOT NULL,
32     UNIT VARCHAR(100) NOT NULL,
33     IMAGE VARCHAR(MAX),
34     STOCK REAL NOT NULL DEFAULT 0,
35     MINIMUM REAL NOT NULL DEFAULT 0,
36     MAXIMUM REAL NOT NULL DEFAULT 0,
37     PRICE REAL NOT NULL DEFAULT 0,
38     DISABLED BIT NOT NULL DEFAULT 0,
39     DISCCOUNT INTEGER NOT NULL DEFAULT 0
40 );
41 /*INFORMACIÓN/MATRIZ DE PERMISOS DE USUARIO*/
42 CREATE TABLE PERMISSIONS
43 (
44     ID INT PRIMARY KEY IDENTITY (1,1),
45     STOCK_OUT BIT DEFAULT 0,
46     STOCK_IN BIT DEFAULT 0,
47     REPORTS_READ BIT DEFAULT 0,
48     USER_MANAGER BIT DEFAULT 0,
49     READ_ONLY BIT DEFAULT 0,
50     SALES BIT DEFAULT 0,
51 );
52 /* LLAVES,TOKENS DE SEGURIDAD PARA DESENCRIPTAR CONTRASEÑAS DE USUARIO*/
53 CREATE TABLE KEYS(
54     ID INT PRIMARY KEY IDENTITY(1,1),
55     PASSWORD_KEY VARCHAR(37) DEFAULT NEWID()
56 );
57 /*INFORMACIÓN SOBRE USUARIOS DEL SITEMA */

```

```

58 CREATE TABLE USERS(
59     ID INT PRIMARY KEY IDENTITY(1,1),
60     PERMISSIONS_ID INT FOREIGN KEY REFERENCES PERMISSIONS NOT NULL,
61     KEYS_ID INT FOREIGN KEY REFERENCES KEYS NOT NULL,
62     NICKNAME VARCHAR(100) NOT NULL UNIQUE,
63     NAME VARCHAR(100) NOT NULL,
64     PASSWORD VARBINARY(MAX) NOT NULL,
65     PICTURE VARCHAR(MAX) DEFAULT NULL,
66     ENABLED BIT NOT NULL DEFAULT 0
67 );
68 /*LISTADO DE CONCEPTOS DE MOVIMIENTOS*/
69 CREATE TABLE MOVEMENT_CONCEPT
70 (
71     ID INT PRIMARY KEY IDENTITY(1,1),
72     TAG VARCHAR(10) NOT NULL,
73     DESCRIPTION VARCHAR(100),
74     TYPE VARCHAR(1)
75 );
76 /*CABECERA DE LA COMPRA*/
77 CREATE TABLE BUYS( --1 para que te de el id de movement
78     ID INT PRIMARY KEY IDENTITY(1,1),
79     USER_ID INT FOREIGN KEY REFERENCES USERS NOT NULL,
80     DATE DATETIME NOT NULL DEFAULT GETDATE(),
81     DESCRIPTION VARCHAR(100)
82 );
83 /*PARTIDAS DE LA COMPRA*/
84 CREATE TABLE BUY_PARTS(
85     ID INT PRIMARY KEY IDENTITY(1,1),
86     BUY_ID INT FOREIGN KEY REFERENCES BUYS NOT NULL,
87     PRODUCT_ID INT FOREIGN KEY REFERENCES PRODUCTS NOT NULL,
88     QUANTITY REAL NOT NULL DEFAULT 0,
89     INITIALLY_STOCK REAL NOT NULL,
90     NEW_STOCK REAL NOT NULL
91 );
92 /*CABECERA DE EL MOVIMIENTO (E/S)*/
93 CREATE TABLE MOVEMENTS( --1 para que te de el id de movement
94     ID INT PRIMARY KEY IDENTITY(1,1),
95     USER_ID INT FOREIGN KEY REFERENCES USERS NOT NULL,
96     MOVEMENT_CONCEPT_ID INT FOREIGN KEY REFERENCES MOVEMENT_CONCEPT NOT NULL,
97     TYPE VARCHAR(1) NOT NULL,
98     DATE DATETIME NOT NULL DEFAULT GETDATE(),
99     DESCRIPTION VARCHAR(100)
100 );
101 /*PARTIDAS DEL MOVIMIENTO*/
102 CREATE TABLE MOVEMENT_PARTS(
103     ID INT PRIMARY KEY IDENTITY(1,1),
104     MOVEMENT_ID INT FOREIGN KEY REFERENCES MOVEMENTS NOT NULL,
105     PRODUCT_ID INT FOREIGN KEY REFERENCES PRODUCTS NOT NULL,
106     QUANTITY REAL NOT NULL DEFAULT 0,
107     INITIALLY_STOCK REAL NOT NULL,
108     NEW_STOCK REAL NOT NULL
109 );
110 /*REGISTRO DE CLIENTES*/
111 CREATE TABLE CUSTOMERS(
112     ID INT PRIMARY KEY IDENTITY(1,1),
113     NAME VARCHAR(100) NOT NULL,
114     NOTES VARCHAR(MAX),

```

```

115         PICTURE VARCHAR(MAX),
116         ENABLED BIT NOT NULL DEFAULT 0
117     );
118     /*LISTADO DE LAS FORMAS DE PAGO DISPONIBLES*/
119     CREATE TABLE PAY_WAYS(
120         ID INT PRIMARY KEY IDENTITY(1,1),
121         NAME VARCHAR(50)
122     );
123     /*LISTADO DE PAGOS RECIBIDOS */
124     CREATE TABLE PAYMENTS
125     (
126         ID INT PRIMARY KEY IDENTITY(1,1),
127         PAY_WAY_ID INT FOREIGN KEY REFERENCES PAY_WAYS,
128         AMOUNT FLOAT NOT NULL DEFAULT 0
129     );
130     /*CABECERA DE UNA VENTA*/
131     CREATE TABLE SALES(
132         ID INT PRIMARY KEY IDENTITY(1,1),
133         CUSTOMER_ID INT FOREIGN KEY REFERENCES CUSTOMERS NOT NULL,
134         USER_ID INT FOREIGN KEY REFERENCES USERS NOT NULL,
135         DATE DATETIME DEFAULT GETDATE(),
136         TOTAL REAL,
137         CHANGE REAL
138     );
139     /*PARTIDAS DE UNA VENTA*/
140     CREATE TABLE SALE_PARTS(
141         ID INT PRIMARY KEY IDENTITY(1,1),
142         SALE_ID INT FOREIGN KEY REFERENCES SALES NOT NULL,
143         PRODUCT_ID INT FOREIGN KEY REFERENCES PRODUCTS NOT NULL,
144         QUANTITY REAL,
145         PRICE REAL,
146         TOTAL REAL
147     );
148     /*PAGOS REALIZADOS EN UNA VENTA ESPECIFICA*/
149     CREATE TABLE SALE_PAYS(
150         ID INT PRIMARY KEY IDENTITY(1,1),
151         SALE_ID INT FOREIGN KEY REFERENCES SALES NOT NULL,
152         PAYMENT_ID INT FOREIGN KEY REFERENCES PAYMENTS
153     );
154     GO
155     /*REGISTRO DE CORTE DE CAJA DE CIERRE*/
156     CREATE TABLE CORTE_Z(
157         ID INT PRIMARY KEY IDENTITY(1,1),
158         DATE DATETIME,
159         USER_ID INT FOREIGN KEY REFERENCES USERS NOT NULL);
160     GO
161     /*LISTADO DE VENTAS POR CORTE DE CAJA*/
162     CREATE TABLE CORTE_Z_PARTS(
163         ID INT PRIMARY KEY IDENTITY(1,1),
164         CORTE_Z_ID INT FOREIGN KEY REFERENCES CORTE_Z NOT NULL,
165         SALE_ID INT FOREIGN KEY REFERENCES SALES NOT NULL,
166     );
167     GO
168     /*REALIZA UN NUEVO CORTE DE CAJA, INSERTA LA LLAVE DE LAS VENTAS QUE
        PARTICIPAN EN ESTE CORTE*/
169     CREATE PROCEDURE SP_CORTE_Z(@USER_ID INT)
170     AS

```

```

171 BEGIN
172     DECLARE @CORTE_Z_ID INT
173     INSERT INTO CORTE_Z (DATE, USER_ID) VALUES (GETDATE(), @USER_ID);
174     SELECT @CORTE_Z_ID = SCOPE_IDENTITY();
175     INSERT INTO CORTE_Z_PARTS (CORTE_Z_ID, SALE_ID) SELECT @CORTE_Z_ID, ID FROM
        SALES WHERE NOT EXISTS
176     (SELECT ID FROM CORTE_Z_PARTS CZ WHERE CZ.SALE_ID = SALES.ID)
177     SELECT @CORTE_Z_ID
178 END;
179 GO
180 /*RETORNA UNA VISTA DE LAS VENTAS QUE AÚN NO ESTAN AGRUPADAS EN ALGÚN CORTE DE
        CAJA*/
181 CREATE VIEW VIEW_GETCORTEZ_SALES
182 AS
183 select SALES.ID, SALES.DATE as 'Fecha', SALES.TOTAL,
184 SALES.CHANGE, USERS.NAME as 'User', CUSTOMERS.NAME as 'Customers'
185 From sales
186 inner join USERS ON USERS.ID = SALES.USER_ID
187 inner join CUSTOMERS ON CUSTOMERS.ID = SALES.CUSTOMER_ID
188 WHERE NOT EXISTS
189 (SELECT ID FROM CORTE_Z_PARTS CZ WHERE CZ.SALE_ID = SALES.ID)
190 GO
191 /*RETORNA UN LISTADO DE LOS PRODUCTOS DE LAS VENTAS QUE AÚN NO ESTAN AGRUPADAS
        EN ALGÚN CORTE DE CAJA*/
192 CREATE VIEW VIEW_GETCORTEZ_PRODUCTS
193 AS
194 select SALE_ID, NAME, QUANTITY, SALE_PARTS.PRICE,
195 TOTAL/1.16 AS 'IMPUESTO',
196 TOTAL-TOTAL/1.16 AS 'IMPORTE',
197 TOTAL
198 From SALE_PARTS
199 INNER JOIN PRODUCTS ON PRODUCTS.ID = SALE_PARTS.PRODUCT_ID
200 WHERE NOT EXISTS
201 (SELECT ID FROM CORTE_Z_PARTS CZ WHERE CZ.SALE_ID = SALE_PARTS.SALE_ID)
202 GO
203 /*RETORNA UN LISTADO DE TODOS LOS PAGOS QUE SE HICIERON EN LAS VENTAS QUE AÚN
        NO ESTAN AGRUPADAS EN ALGÚN CORTE DE CAJA*/
204 CREATE VIEW VIEW_PAYMENTS_CORTE_Z
205 AS
206 select SALE_ID, NAME, AMOUNT
207 From SALE_PAYS
208 INNER JOIN PAYMENTS ON PAYMENTS.ID = SALE_PAYS.PAYMENT_ID
209 INNER JOIN PAY_WAYS ON PAY_WAYS.ID = PAYMENTS.PAY_WAY_ID
210 WHERE NOT EXISTS
211 (SELECT ID FROM CORTE_Z_PARTS CZ WHERE CZ.SALE_ID = SALE_PAYS.SALE_ID)
212 GO
213 /*INSERTA LA CABECERA DE UNA VENTA NUEVA Y RETORNA SU ID*/
214 CREATE PROCEDURE SP_SAVE_SALE (@CUSTOMER_ID INT, @USER_ID INT, @TOTAL FLOAT,
        @CHANGE FLOAT)
215 AS
216 BEGIN
217     INSERT INTO SALES (CUSTOMER_ID, USER_ID, DATE, TOTAL, CHANGE) VALUES (@CUSTOMER_ID,
        @USER_ID, GETDATE(), @TOTAL, @CHANGE);
218     SELECT SCOPE_IDENTITY();
219 END
220 GO
221 /*INSERTA UNA PARTIDA DE UNA VENTA NUEVA Y RETORNA SU ID*/

```

```

222 CREATE PROCEDURE SP_SAVE_SALE_PART
223 (@SALE_ID INT ,@PRODUCT_ID INT,@PRICE FLOAT,@QUANTITY FLOAT,@TOTAL FLOAT)
224 AS
225 BEGIN
226 --UPDATE PRODUCTS SET STOCK-=@QUANTITY WHERE ID=@PRODUCT_ID
227 INSERT INTO SALE_PARTS
228 (SALE_ID,PRODUCT_ID,QUANTITY,PRICE,TOTAL)
229 VALUES (@SALE_ID,@PRODUCT_ID,@QUANTITY,@PRICE,@TOTAL);
230 SELECT SCOPE_IDENTITY();
231 END
232 GO
233 /*INSERTA LOS MÉTODOS DE PAGO DE UNA VENTA NUEVA*/
234 CREATE PROCEDURE SP_SAVE_SALE_PAYS(@SALE_ID INT,@PAYMENT_ID FLOAT)
235 AS
236 BEGIN
237 INSERT INTO SALE_PAYS(SALE_ID,PAYMENT_ID) VALUES (@SALE_ID,@PAYMENT_ID);
238 END
239 GO
240 /*INSERTA LA CANTIDAD EN DINERO DE UN MÉTODO DE PAGO DE UNA VENTA NUEVA*/
241 CREATE PROCEDURE SP_SAVE_PAYMENT(@PAY_WAY_ID INT,@AMOUNT FLOAT)
242 AS
243 BEGIN
244 INSERT INTO PAYMENTS(PAY_WAY_ID,AMOUNT) VALUES (@PAY_WAY_ID,@AMOUNT);
245 SELECT SCOPE_IDENTITY();
246 END
247 GO
248 --INSERTA LOS TIPOS DE PAGO POR DEFECTO
249 INSERT INTO PAY_WAYS (NAME) VALUES('Cash');
250 INSERT INTO PAY_WAYS (NAME) VALUES('Vale');
251 INSERT INTO PAY_WAYS (NAME) VALUES('Card');
252 --Loguea al usuario pidiendo como parametros Nickname y Password y
    desencriptando a su vez la contraseña para validarla
253 GO
254 CREATE PROCEDURE SP_LOGIN (@NICKNAME VARCHAR(100), @PASSWORD NVARCHAR(50))
255 AS
256 BEGIN
257     DECLARE @USER_ID INT;
258     SELECT @USER_ID=ID FROM USERS WHERE NICKNAME=@NICKNAME
259     IF(@USER_ID>0)
260     BEGIN
261         DECLARE @PASSWORD_KEY VARCHAR(37);
262         DECLARE @REAL_PASSWORD VARCHAR(50);
263         DECLARE @KEY_ID INT;
264         SELECT @KEY_ID=KEYS_ID FROM USERS WHERE ID=@USER_ID
265
266         SELECT @PASSWORD_KEY=PASSWORD_KEY FROM KEYS WHERE ID=@KEY_ID
267
268         SELECT @REAL_PASSWORD=CONVERT(NVARCHAR(MAX), DECRYPTBYPASSPHRASE
            (@PASSWORD_KEY, PASSWORD)) FROM USERS WHERE ID=@USER_ID
269         IF (@REAL_PASSWORD=@PASSWORD)
270         BEGIN
271             SELECT 1 AS 'LOGIN_STATUS';
272             RETURN;
273         END
274     END
275     SELECT 0 AS 'LOGIN_STATUS';
276 END;

```

```

277 GO
278 --Registra y/o Actualiza al usuario pidiendo como parametros nickname,
      contraseña,Nobre y una imagen.
279 --- Encripta la contraseña por seguridad del ususario
280 CREATE PROCEDURE SP_REGISTER(@NICKNAME VARCHAR(100), @PASSWORD NVARCHAR(50),
      @NAME VARCHAR(100),@PICTURE VARCHAR(MAX),@ENABLED BIT,@ID INT)
281 AS
282 BEGIN
283     IF(@ID<=0)
284     BEGIN
285         DECLARE @PERMISSIONS_ID INT;
286         DECLARE @KEY_ID INT;
287         DECLARE @KEY VARCHAR(MAX);
288         DECLARE @ENCRYPTED_PASSWORD VARBINARY(MAX);
289
290         INSERT INTO PERMISSIONS DEFAULT VALUES
291         SELECT @PERMISSIONS_ID=SCOPE_IDENTITY();
292
293         INSERT INTO KEYS DEFAULT VALUES
294         SELECT @KEY_ID=SCOPE_IDENTITY();
295         SELECT @KEY =PASSWORD_KEY FROM KEYS WHERE ID=@KEY_ID
296
297         SELECT @ENCRYPTED_PASSWORD = ENCRYPTBYPASSPHRASE(@KEY, @PASSWORD);
298
299         INSERT INTO USERS(PERMISSIONS_ID,KEYS_ID,NICKNAME,NAME,PASSWORD,PICTURE,
      ENABLED)
300         VALUES(@PERMISSIONS_ID,@KEY_ID,@NICKNAME,@NAME,@ENCRYPTED_PASSWORD,
      @PICTURE,@ENABLED)
301     END ELSE
302     BEGIN
303
304         SELECT @KEY_ID=KEYS_ID FROM USERS WHERE ID=@ID
305         SELECT @KEY =PASSWORD_KEY FROM KEYS WHERE ID=@KEY_ID
306         SELECT @ENCRYPTED_PASSWORD = ENCRYPTBYPASSPHRASE(@KEY, @PASSWORD);
307         UPDATE USERS SET NICKNAME=@NICKNAME,NAME=@NAME,PASSWORD=
      @ENCRYPTED_PASSWORD,PICTURE=@PICTURE,ENABLED=@ENABLED
308         WHERE ID=@ID
309
310     END
311 END;
312 GO
313
314 -----PRODUCTS-----
315 GO
316 --Obtiene la existencia de un producto mendante su Id
317 CREATE PROCEDURE SP_GETSTOCK(@ID INT)
318 AS
319 BEGIN
320     SELECT STOCK FROM PRODUCTS WHERE ID=@ID
321 END;
322 GO
323 ---Obtiene los productos donde DISABLED sea igual a cero y ordnandolos por su
      nombre
324 CREATE PROCEDURE SP_GET_PRODUCTS
325 AS
326 BEGIN
327     SELECT * FROM PRODUCTS WHERE DISABLED=0 ORDER BY NAME

```



```

328 END;
329 GO
330 --Obtiene los productos por su codigo, donde DISABLED sea igual a cero y
      ordnandolos por su nombre
331 CREATE PROCEDURE SP_FIND_PRODUCT(@CODE VARCHAR(100))
332 AS
333 BEGIN
334     SELECT * FROM PRODUCTS WHERE DISABLED=0 AND CODE=@CODE ORDER BY NAME
335 END;
336 GO
337 --Obtiene todos los productos de la tabla Products
338 CREATE PROCEDURE SP_GET_PRODUCT_BY_ID (@ID INT)
339 AS
340 BEGIN
341     SELECT * FROM PRODUCTS WHERE ID=@ID
342 END
343 GO
344 ---Procedure para la busqueda de un producto donde el patrón del codigo o del
      nombre coincida
345 CREATE PROCEDURE SP_SEARCH_PRODUCT(@SEARCH VARCHAR(200))
346 AS
347 BEGIN
348     SELECT * FROM PRODUCTS WHERE DISABLED=0 AND CODE LIKE '%' + @SEARCH + '%' OR
      NAME LIKE '%' + @SEARCH + '%' ORDER BY NAME
349 END;
350 GO
351 --Obtiene los productos de una categoria
352 CREATE PROCEDURE SP_GET_PRODUCTS_BY_CATEGORY(@CATEGORY_ID INT)
353 AS
354 BEGIN
355     SELECT * FROM PRODUCTS WHERE DISABLED=0 AND CATEGORY_ID=@CATEGORY_ID ORDER BY
      NAME
356 END;
357 GO
358 --Obtiene los productos por la categoria buscada
359 CREATE PROCEDURE SP_GET_PRODUCTS_BY_CATEGORY_FIND(@CATEGORY_ID INT, @SEARCH
      VARCHAR(MAX))
360 AS
361 BEGIN
362     SELECT * FROM PRODUCTS WHERE DISABLED=0 AND CATEGORY_ID=@CATEGORY_ID AND NAME
      LIKE '%' + @SEARCH + '%' ORDER BY NAME
363 END;
364 GO
365 --Obtiene las categorias y las ordena por su descripcion
366 CREATE PROCEDURE SP_GET_CATEGORIES
367 AS
368 BEGIN
369     SELECT * FROM CATEGORIES order by DESCRIPTION
370 END;
371 GO
372 --Busca las categorias donde el patrón de la descripción categoria o de la
      descripción del producto coincida ordenandolos por su descripción
373 CREATE PROCEDURE SP_FIND_CATEGORIES (@SEARCH VARCHAR(MAX))
374 AS
375 BEGIN
376     SELECT DISTINCT CATEGORIES.ID, CATEGORIES.DESCRPTION FROM CATEGORIES
      INNER JOIN PRODUCTS ON PRODUCTS.CATEGORY_ID=CATEGORIES.ID
377

```

```

378         WHERE CATEGORIES.DESCRPTION LIKE '%' + @SEARCH + '%' OR PRODUCTS.DESCRPTION
          LIKE '%' + @SEARCH + '%' order by CATEGORIES.DESCRPTION, CATEGORIES.ID
379     END;
380     GO
381     ---Obtiene los categoria por la descripción
382     CREATE PROCEDURE SP_GET_CATEGORY_BY_NAME (@NAME VARCHAR(100))
383     AS
384     BEGIN
385         SELECT *FROM CATEGORIES WHERE DESCRIPTION=@NAME
386     END;
387     GO
388     ---Obtiene los categoria por el Id
389     CREATE PROCEDURE SP_GET_CATEGORY_BY_ID (@ID INT)
390     AS
391     BEGIN
392         SELECT *FROM CATEGORIES WHERE ID=@ID
393     END;
394     GO
395     ---Obtiene todos los proveedores de la tabla SUPPLIERS
396     CREATE PROCEDURE SP_GET_SUPPLIERS
397     AS
398     BEGIN
399         SELECT *FROM SUPPLIERS
400     END;
401     GO
402     ---Registra y actualizacion de productos con los paramentros de id en caso de
      actualizar,codigo,nombre,etc...
403     CREATE PROCEDURE SP_ABC_PRODUCT (@ID INT, @CODE VARCHAR(100), @NAME VARCHAR(100)
      , @DESCRIPTION VARCHAR(MAX),
404     @CATEGORY_ID INT, @SUPPLIER_ID INT, @UNIT VARCHAR(100), @IMAGE VARCHAR(MAX),
      @STOCK REAL, @MINIMUM REAL, @MAXIMUM REAL,
405     @PRICE REAL, @DISABLED BIT, @DISCCOUNT REAL)
406     AS
407     BEGIN
408         IF (@ID <= 0)
409         BEGIN
410             INSERT INTO PRODUCTS (CODE, NAME, DESCRIPTION, CATEGORY_ID, SUPPLIER_ID,
              UNIT, IMAGE, STOCK, MINIMUM, MAXIMUM, PRICE, DISABLED, DISCCOUNT)
411             VALUES (@CODE, @NAME, @DESCRIPTION, @CATEGORY_ID, @SUPPLIER_ID, @UNIT, @IMAGE
              , @STOCK, @MINIMUM, @MAXIMUM, @PRICE, @DISABLED, @DISCCOUNT)
412         END ELSE
413         BEGIN
414             UPDATE PRODUCTS SET CODE=@CODE, NAME=@NAME, DESCRIPTION=@DESCRIPTION,
              CATEGORY_ID=@CATEGORY_ID,
415             SUPPLIER_ID=@SUPPLIER_ID, UNIT=@UNIT, IMAGE=@IMAGE, STOCK=@STOCK, MINIMUM=
              @MINIMUM, MAXIMUM=@MAXIMUM, PRICE=@PRICE, DISABLED=@DISABLED, DISCCOUNT
              =@DISCCOUNT
416             WHERE ID=@ID
417         END
418     END;
419     GO
420     ---Registra y actualizacion de cateogrias con los paramentros de id en caso
      de actualizar, descripción e imagen
421     CREATE PROCEDURE SP_ABC_CATEGORY (@ID INT, @DESCRIPTION VARCHAR(100), @PICTURE
      VARCHAR(MAX))
422     AS
423     BEGIN

```

```

424         IF (@ID <= 0)
425         BEGIN
426             INSERT INTO CATEGORIES (DESCRIPTION, PICTURE) VALUES (@DESCRIPTION,
427                             @PICTURE)
428         END ELSE
429         BEGIN
430             UPDATE CATEGORIES SET DESCRIPTION = @DESCRIPTION, PICTURE = @PICTURE WHERE
431                 ID = @ID
432         END
433     END;
434 GO
435 -----ALERTS-----
436 CREATE VIEW VIEW_OVERSTOCKED
437 AS
438 SELECT * FROM PRODUCTS WHERE STOCK >= MAXIMUM;
439 GO
440 CREATE VIEW VIEW_UNDERSTOCKED
441 AS
442 SELECT * FROM PRODUCTS WHERE STOCK <= MINIMUM;
443 GO
444 CREATE VIEW VIEW_ALL_ALERTS
445 AS
446 SELECT * FROM PRODUCTS WHERE STOCK >= MAXIMUM OR STOCK <= MINIMUM;
447 GO
448 -----MOVEMENTS-----
449 GO
450 ---Obtiene todos las compras de la tabla BUYS mediante su Id
451 CREATE PROCEDURE SP_GET_BUY_BY_ID (@ID INT)
452 AS
453 BEGIN
454     SELECT * FROM BUYS WHERE ID = @ID;
455 END;
456 GO
457 ---Obtiene todos los movimientos de la tabla MOVEMENTS
458 CREATE PROCEDURE SP_GET_MOVEMENT (@ID INT)
459 AS
460 BEGIN
461     SELECT * FROM MOVEMENTS WHERE ID = @ID;
462 END;
463 GO
464 ---Obtiene todos las partes de los movimientos de la tabla MOVEMENT_PARTS
465 CREATE PROCEDURE SP_GET_MOVEMENTPART_BY_ID (@ID INT)
466 AS
467 BEGIN
468     SELECT * FROM MOVEMENT_PARTS WHERE ID = @ID;
469 END;
470 GO
471 ---Obtiene todos las partes de las compras de la tabla BUY_PARTS
472 CREATE PROCEDURE SP_GET_BUY_PART_BY_ID (@ID INT)
473 AS
474 BEGIN
475     SELECT * FROM BUY_PARTS WHERE ID = @ID;
476 END;
477 GO
478 ---Obtiene todos las partes de las compras de la tabla BUY_PARTS ?????????
479     Esta repetido?
480 CREATE PROCEDURE SP_GET_BUY_PARTS_BY_ID (@ID INT)

```

```

478 AS
479 BEGIN
480 SELECT * FROM BUY_PARTS WHERE BUY_ID=@ID;
481 END;
482 GO
483 ---Obtiene todos las partes de los movimientos de la tabla MOVEMENT_PARTS DADO
      UN ID DE MOVIMIENTO
484 CREATE PROCEDURE SP_GET_MOVEMENTPART(@ID INT)
485 AS
486 BEGIN
487 SELECT * FROM MOVEMENT_PARTS WHERE MOVEMENT_ID=@ID;
488 END;
489 GO
490 ---Obtiene todos los conceptos de la tabla MOVEMENT_CONCEPT
491 CREATE PROCEDURE SP_GET_MOVEMENTCONCEPT(@ID INT)
492 AS
493 BEGIN
494 SELECT * FROM MOVEMENT_CONCEPT WHERE ID =@ID;
495 END;
496 GO
497 ---Registra las compras con los paramentros de id del usuario,fecha y
      descripción
498 CREATE PROCEDURE SP_ADD_BUY(
499 @USER_ID INT,
500 @DATE DATETIME,
501 @DESCRIPTION VARCHAR(100))
502 AS
503 BEGIN
504     INSERT INTO BUYS (USER_ID,DATE,DESCRIPTION)
505     VALUES (@USER_ID,@DATE,@DESCRIPTION);
506     SELECT SCOPE_IDENTITY();
507 END;
508 GO
509 ---Registra las partes de las compras con los paramentros de id del movimeinto
      ,id del producto y cantidad
510 ---ademas de que ajusta la existencia de los productos que s even afectados por
      los movimientos
511 CREATE PROCEDURE SP_ADD_BUY_PARTS(
512 @BUY_ID INT,
513 @PRODUCT_ID INT,
514 @QUANTITY REAL)
515 AS
516 BEGIN
517     DECLARE @INITIALLY_STOCK REAL
518     DECLARE @NEW_STOCK REAL
519     SELECT @INITIALLY_STOCK = STOCK FROM PRODUCTS WHERE ID=@PRODUCT_ID
520     SELECT @NEW_STOCK=@INITIALLY_STOCK+@QUANTITY
521     INSERT INTO BUY_PARTS (BUY_ID,PRODUCT_ID,QUANTITY,INITIALLY_STOCK,
522     NEW_STOCK)
523     VALUES (@BUY_ID,@PRODUCT_ID,@QUANTITY,@INITIALLY_STOCK,@NEW_STOCK);
524     SELECT SCOPE_IDENTITY();
525 END;
526 GO
527 ---Registra los movimientos con los paramentros de id del usuario,id del
      concepto,tipo,fecha y descripción
528 CREATE PROCEDURE SP_ADD_MOVEMENT(
      @USER_ID INT,

```

```

529     @MOVEMENT_CONCEPT_ID INT,
530     @TYPE VARCHAR(1),
531     @DATE_M DATETIME,
532     @DESCRIPTION VARCHAR(100))
533 AS
534 BEGIN
535     INSERT INTO MOVEMENTS (USER_ID,MOVEMENT_CONCEPT_ID,TYPE,DATE,DESCRIPTION)
536     VALUES(@USER_ID,@MOVEMENT_CONCEPT_ID,@TYPE,@DATE_M,@DESCRIPTION);
537     SELECT SCOPE_IDENTITY();
538 END;
539 GO
540 --Registra las partes de los movimientos con los paramentros de id del
541 --movimeinto,id del producto y cantidad
542 --ademas de que ajusta la existencia de los productos que s even afectados por
543 --los movimientos
544 CREATE PROCEDURE SP_ADD_MOVENT_PARTS(
545     @MOVEMENT_ID INT,
546     @PRODUCT_ID INT,
547     @QUANTITY REAL,
548     @TYPE VARCHAR(1))
549 AS
550 BEGIN
551     DECLARE @INITIALLY_STOCK REAL
552     DECLARE @NEW_STOCK REAL
553     SELECT @INITIALLY_STOCK = STOCK FROM PRODUCTS WHERE ID=@PRODUCT_ID
554     IF(@TYPE='S')
555     BEGIN
556         SELECT @QUANTITY=@QUANTITY*-1
557     END
558     SELECT @NEW_STOCK=@INITIALLY_STOCK+@QUANTITY
559     UPDATE PRODUCTS SET STOCK=@NEW_STOCK WHERE ID=@PRODUCT_ID
560     INSERT INTO MOVEMENT_PARTS (MOVEMENT_ID,PRODUCT_ID,QUANTITY,
561     INITIALLY_STOCK,NEW_STOCK)
562     VALUES(@MOVEMENT_ID,@PRODUCT_ID,@QUANTITY,@INITIALLY_STOCK,@NEW_STOCK);
563     SELECT SCOPE_IDENTITY();
564 END;
565 GO
566 --Registra los conceptos de movimientos con los paramentros de tag,
567 --descripción y tipo
568 CREATE PROCEDURE SP_ADD_MOVEMENTCONCEPT(
569     @TAG VARCHAR(10),
570     @DESCRIPTION VARCHAR(100),
571     @TYPE VARCHAR(1))
572 AS
573 BEGIN
574     INSERT INTO MOVEMENT_CONCEPT (TAG,DESCRIPTION,TYPE)
575     VALUES(@TAG,@DESCRIPTION,@TYPE);
576 END;
577 GO
578 --Obtiene todos las compras de la tabla BUYS
579 CREATE VIEW VIEW_GET_ALL_BUYS
580 AS
581 SELECT * FROM BUYS ;
582 GO
583 --Obtiene todos los movimientos de la tabla MOVEMENTS
584 CREATE VIEW VIEW_GETALLMOVEMENTS
585 AS

```

```

582 SELECT * FROM MOVEMENTS ;
583 GO
584 ---Obtiene todos los conceptos de movimientos de la tabla MOVEMENT_CONCEPT
585 CREATE VIEW VIEW_GETALLMOVEMENTCONCEPT
586 AS
587 SELECT * FROM MOVEMENT_CONCEPT ;
588 GO
589 GO
590 ---Obtiene todas las partes de los movimientos de la tabla MOVEMENT_PARTS
591 CREATE VIEW VIEW_GETALLMOVEMENTPART
592 AS
593 SELECT * FROM MOVEMENT_PARTS ;
594 -----CUSTOMER-----
595 GO
596 ---Registra y actualizacion de categorias con los parametros de id en caso
    de actualizar, notas e imagen ademas de enabled para eliminar al cliente
597 CREATE PROCEDURE SP_ABC_CUSTOMER (@ID INT, @NAME VARCHAR(100), @NOTES VARCHAR(
    MAX), @PICTURE VARCHAR(MAX), @ENABLED BIT)
598 AS
599 BEGIN
600     IF (@ID <= 0)
601     BEGIN
602         INSERT INTO CUSTOMERS (NAME, NOTES, PICTURE, ENABLED ) VALUES (@NAME, @NOTES
            , @PICTURE, @ENABLED )
603     END ELSE
604     BEGIN
605         UPDATE CUSTOMERS SET NAME= @NAME, NOTES=@NOTES, PICTURE=@PICTURE, ENABLED
            =@ENABLED WHERE ID=@ID
606     END
607 END;
608 GO
609 ---Obtiene los clientes de la tabla CUSTOMERS mediante su id
610 CREATE PROCEDURE SP_SEARCHCUSTOMER (@ID INT)
611 AS
612 BEGIN
613 SELECT * FROM CUSTOMERS WHERE ID= @ID;
614 END;
615
616 GO
617 ---Obtiene todos los clientes de la tabla CUSTOMERS que no esten eliminados
618 CREATE VIEW VIEW_GETALL
619 AS
620 SELECT * FROM CUSTOMERS WHERE ENABLED= 0;
621
622
623 -----Supplier-----
624 GO
625 -----Registra y actualizacion de proveedores con los parametros de id en
    caso de actualizar, nombre, telefono, etc...
626 CREATE PROCEDURE SP_ABC_SUPPLIERS (@ID INT, @NAME VARCHAR(100), @CELLPHONE
    VARCHAR(100), @NOTES VARCHAR(MAX), @PICTURE VARCHAR(MAX), @ENABLED BIT)
627 AS
628 BEGIN
629     IF (@ID <= 0)
630     BEGIN
631         INSERT INTO SUPPLIERS (NAME, CELLPHONE, NOTES, PICTURE, ENABLED) VALUES (
            @NAME, @CELLPHONE, @NOTES, @PICTURE, @ENABLED)

```

```

632         END ELSE
633         BEGIN
634             UPDATE SUPPLIERS SET NAME=@NAME,CELLPHONE=@CELLPHONE,NOTES=@NOTES,
                PICTURE=@PICTURE,ENABLED=@ENABLED WHERE ID=@ID;
635     END
636 END;
637 GO
638 ---Obtiene a todos los proveedores de la tabla SUPPLIERS
639 CREATE VIEW VIEW_GETALLSUPPLIERS
640 AS
641 SELECT * FROM SUPPLIERS WHERE ENABLED=0;
642 GO
643 ---Obtiene a los proveedores de la tabla SUPPLIERS mediante su Id
644 CREATE PROCEDURE SP_GETIDSUPPLIERS(@ID INT)
645 AS
646 SELECT * FROM SUPPLIERS WHERE ID=@ID;
647 GO
648 /*INSERTA O ACTUALIZA UN DETERMINADO REGISTRO DE PERMISOS GUIANDOSE POR SU ID
        */
649 CREATE PROCEDURE SP_ABC_PERMISSIONS
650 ( @ID INT,@STOCK_OUT BIT,@STOCK_IN BIT ,@REPORTS_READ BIT,@USER_MANAGER BIT,
        @READ_ONLY BIT,@SALES BIT)
651 AS
652 BEGIN
653 IF(@ID<=0)
654 BEGIN
655     INSERT INTO PERMISSIONS (STOCK_OUT,STOCK_IN,REPORTS_READ,USER_MANAGER,
        READ_ONLY,SALES)VALUES(@STOCK_OUT,@STOCK_IN ,@REPORTS_READ ,
        @USER_MANAGER,
656     @READ_ONLY,@SALES)
657 END ELSE
658 BEGIN
659     UPDATE PERMISSIONS SET STOCK_OUT=@STOCK_OUT,STOCK_IN=@STOCK_IN,
        REPORTS_READ=@REPORTS_READ,USER_MANAGER=@USER_MANAGER,READ_ONLY=
        @READ_ONLY,SALES=@SALES
660     WHERE ID=@ID
661 END
662 END;
663 GO
664 /*RECUPERA UN REGISTRO DE PERMISOS POR SU ID*/
665 CREATE PROCEDURE GET_PERMISSIONS_BY_ID (@ID INT)
666 AS
667 BEGIN
668     SELECT *FROM PERMISSIONS WHERE ID=@ID
669 END;
670 ----USERS-----
671 GO
672 /*OBTIENE LA INFORMACIÓN UN USUARIO POR SU NICKNAME*/
673 CREATE PROCEDURE GET_USER_BY_NICKNAME (@NICKNAME VARCHAR(100))
674 AS
675 BEGIN
676
677     DECLARE @USER_ID INT;
678     SELECT @USER_ID=ID FROM USERS WHERE NICKNAME=@NICKNAME
679     DECLARE @PASSWORD_KEY VARCHAR(37);
680     DECLARE @REAL_PASSWORD VARCHAR(50);
681     DECLARE @KEY_ID INT;

```

```

682     SELECT @KEY_ID=KEYS_ID FROM USERS WHERE ID=@USER_ID
683     SELECT @PASSWORD_KEY=PASSWORD_KEY FROM KEYS WHERE ID=@KEY_ID
684     SELECT @REAL_PASSWORD=CONVERT(NVARCHAR(MAX), DECRYPTBYPASSPHRASE(
        @PASSWORD_KEY, PASSWORD)) FROM USERS WHERE ID=@USER_ID
685     SELECT ID,PERMISSIONS_ID,KEYS_ID,NICKNAME,NAME,@REAL_PASSWORD,PICTURE,ENABLED
686     FROM USERS WHERE ID=@USER_ID
687 END;
688 GO
689 /*OBTIENE LA INFORMACIÓN UN USUARIO POR SU ID*/
690 CREATE PROCEDURE GET_USER_BY_ID (@ID INT)
691 AS
692 BEGIN
693     DECLARE @PASSWORD_KEY VARCHAR(37);
694     DECLARE @REAL_PASSWORD VARCHAR(50);
695     DECLARE @KEY_ID INT;
696     SELECT @KEY_ID=KEYS_ID FROM USERS WHERE ID=@ID
697     SELECT @PASSWORD_KEY=PASSWORD_KEY FROM KEYS WHERE ID=@KEY_ID
698     SELECT @REAL_PASSWORD=CONVERT(NVARCHAR(MAX), DECRYPTBYPASSPHRASE(
        @PASSWORD_KEY, PASSWORD)) FROM USERS WHERE ID=@ID
699     SELECT ID,PERMISSIONS_ID,KEYS_ID,NICKNAME,NAME,@REAL_PASSWORD,PICTURE,ENABLED
700     FROM USERS WHERE ID=@ID
701 END;
702 GO
703 GO
704 /*RETORNA TODOS LOS USUARIOS QUE NO HAN SIDO ELIMINADOS*/
705 CREATE VIEW VIEW_GETALLUSERS
706 AS
707 SELECT * FROM USERS WHERE ENABLED=0;
708 GO
709 /*RETORNA LA INFORMACIÓN DE NOMBRE Y STOCK DE TODOS LOS PRODUCTOS*/
710 CREATE VIEW VIEW_GET_STOCK
711 AS
712 SELECT
713 code,NAME,stock,MINIMUM,MAXIMUM
714 FROM products where DISABLED=0
715 GO
716 /*RETORNA UN REPORTE DE VENTAS ENTRE UN RANGO DE FECHAS O BIEN TODOS LOS
    REGISTROS*/
717 CREATE PROCEDURE SALES_REPORT(@BEGIN DATETIME,@END DATETIME,@ALL BIT)
718 AS
719 BEGIN
720     select SALES.ID,SALES.DATE as 'Fecha',SALES.TOTAL,SALES.CHANGE,USERS.NAME as '
        User',CUSTOMERS.NAME as 'Customers'
721     From sales
722     inner join USERS ON USERS.ID=SALES.USER_ID
723     inner join CUSTOMERS ON CUSTOMERS.ID=SALES.CUSTOMER_ID
724     WHERE @ALL=1 OR (SALES.DATE>=@BEGIN AND SALES.DATE<=@END)
725     END
726     GO
727     /*RETORNA UN REPORTE DE MOVIMIENTOS POR USUARIO ENTRE UN RANGO DE FECHAS O
        BIEN TODOS LOS REGISTROS*/
728     CREATE PROCEDURE SP_MOVENTS_REGISTER (@BEGIN DATETIME,@END DATETIME,@ALL BIT)
729     AS
730     BEGIN
731     SELECT 'Ticket #' +CAST(SALES.ID AS VARCHAR(MAX)) AS ID,USERS.NAME, DATE AS '
        Fecha', 'Venta'AS TYPE FROM SALES
732     INNER JOIN USERS ON USERS.ID=SALES.USER_ID

```



```

733 WHERE @ALL=1 OR (SALES.DATE>=@BEGIN AND SALES.DATE<=@END)
734 UNION
735 SELECT 'Movimiento #' + CAST(MOVEMENTS.ID AS VARCHAR(MAX)) as ID, USERS.NAME,
       DATE AS 'Fecha',
736 IIF(TYPE='S', 'Salida', 'Entrada') AS TYPE
737 FROM MOVEMENTS
738 INNER JOIN USERS ON USERS.ID=MOVEMENTS.USER_ID
739 WHERE @ALL=1 OR (MOVEMENTS.DATE>=@BEGIN AND MOVEMENTS.DATE<=@END)
740 UNION
741 SELECT 'Compra #' + CAST(BUYS.ID AS VARCHAR(MAX)) as ID, USERS.NAME, DATE AS '
       Fecha', 'Compra' AS TYPE
742 FROM BUYS
743 INNER JOIN USERS ON USERS.ID=BUYS.USER_ID
744 WHERE @ALL=1 OR (BUYS.DATE>=@BEGIN AND BUYS.DATE<=@END)
745 END;

```

8.1.1 Ejemplo de llamada a procedimiento

```

1  AppData.SQL.EXEC("SP_ABC_PRODUCT", CommandType.StoredProcedure,
2  new SqlParameter("ID", Id),
3  new SqlParameter("CODE", Code),
4  new SqlParameter("NAME", Name),
5  new SqlParameter("DESCRIPTION", Description),
6  new SqlParameter("CATEGORY_ID", Category.Id),
7  new SqlParameter("SUPPLIER_ID", Supplier.Id),
8  new SqlParameter("STOCK", Stock),
9  new SqlParameter("UNIT", Unit),
10 new SqlParameter("IMAGE", Picture),
11 new SqlParameter("MINIMUM", Minimum),
12 new SqlParameter("MAXIMUM", Maximum),
13 new SqlParameter("PRICE", Price),
14 new SqlParameter("DISABLED", Disabled),
15 new SqlParameter("DISCCOUNT", Disccount)
16 );
17
18 using (IReader reader = AppData.SQL.Read("SP_GET_PRODUCT_BY_ID", CommandType.
       StoredProcedure,
19 new SqlParameter("ID", Id)))
20 {
21     if (reader.Read())
22     {
23         producto = new Product()
24         {
25             Id = Convert.ToInt32(reader[0]),
26             Code = Convert.ToString(reader[1]),
27             Name = Convert.ToString(reader[2]),
28             Description = Convert.ToString(reader[3]),
29             Category = Category.GetById(Convert.ToInt32(reader[4])),
30             Supplier = Supplier.GetById(Convert.ToInt32(reader[5])),
31             Unit = Convert.ToString(reader[6]),
32             Picture = reader[7].ToString(),
33             Stock = Convert.ToSingle(reader[8]),
34             Minimum = Convert.ToSingle(reader[9]),
35             Maximum = Convert.ToSingle(reader[10]),
36             Price = Convert.ToSingle(reader[11]),
37             Disabled = Convert.ToBoolean(reader[12]),

```

```
38         Disccount = Convert.ToInt32(reader[13])
39     };
40 }
41 }
```

8.2.0 Diseño conceptual

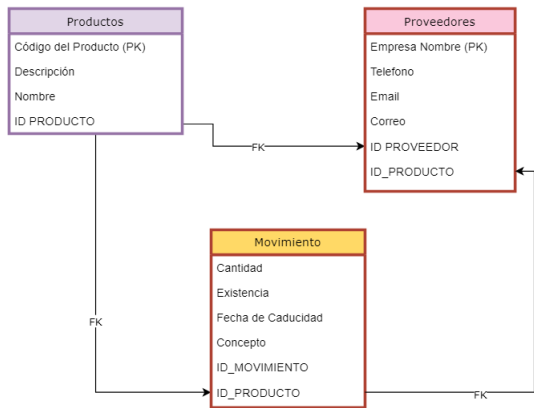


Figure 8.1: Primera Versión tabla 1

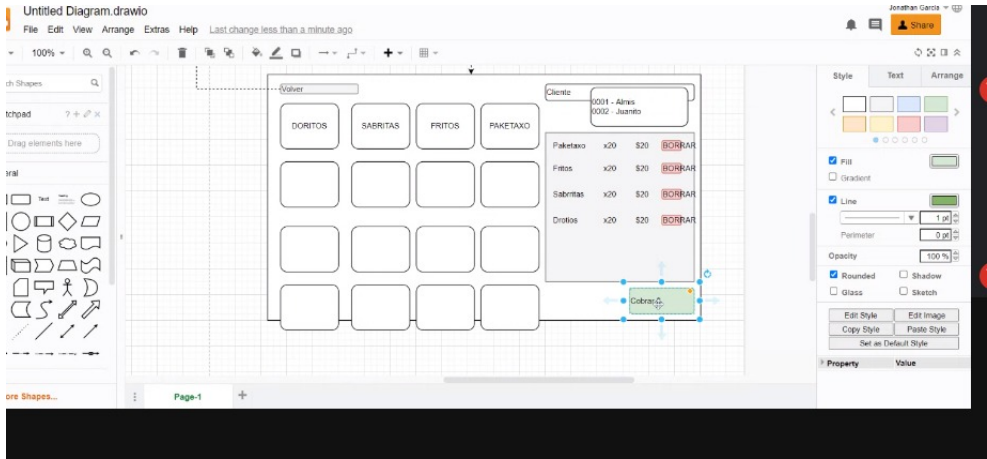


Figure 8.2: Boceto Productos

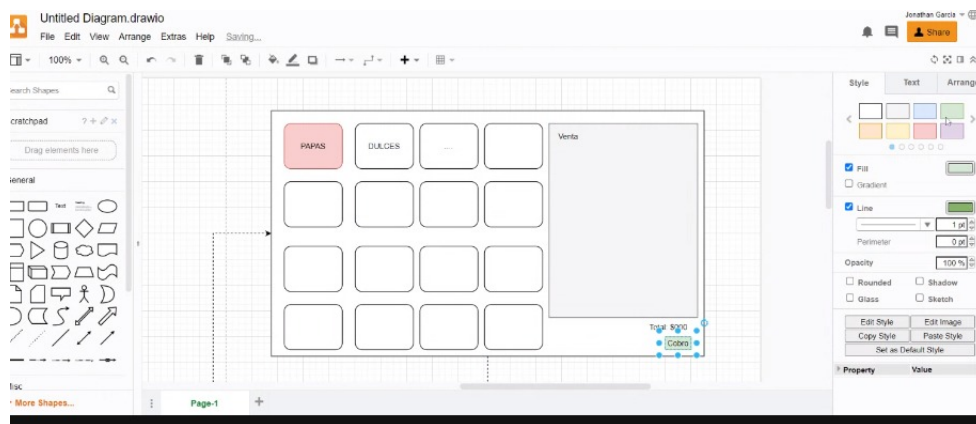


Figure 8.3: Boceto Productos

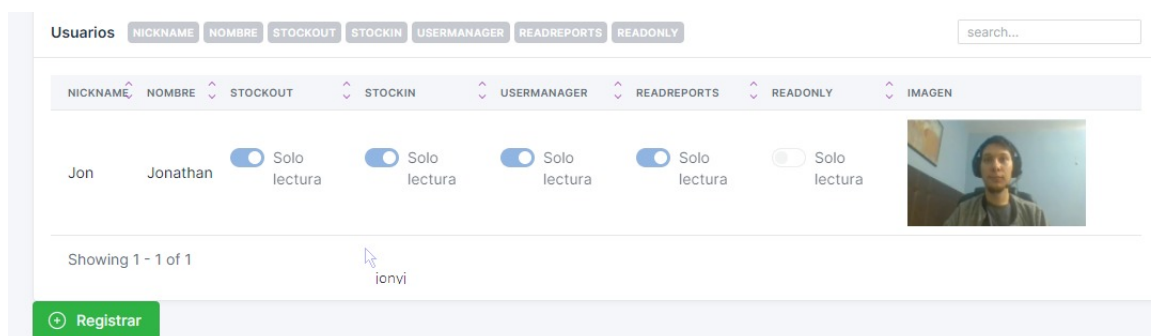


Figure 8.4: Boceto final de usuarios

9.0 Manual de usuario

9.1 Introducción

El aplicativo web es un software que permite a nuestros clientes realizar el almacenamiento y la consulta de los datos registrados en la base de datos. Nos permitirá tomar el control de cada producto que se vaya ingresando de los proveedores, así como los productos que vayan de salida de cada cliente.

9.1.1 ¿Cómo usar este manual?

La interface principal presenta la opción para que ingrese el nombre, contraseña y tipo de usuario. Entre las opciones principales constarán las siguientes:

- Módulo Administrador
- Módulo Usuario

Cuando se ingresa a cualquiera de las opciones anteriormente mencionadas se podrá escoger entre varias funciones, todo esto referente a la opción escogida.

9.2 Visión general de la aplicación

9.2.1 Entrada al sistema

Para poder entrar en la aplicación es obligatorio identificarse, para ello es necesario introducir el usuario y la contraseña.



Figure 9.5: Ingreso de Usuario y contraseña

Botones disponibles:

Ingresar: Una vez introducido el usuario y la contraseña, pulsar este botón para acceder al menú principal dónde una vez dentro del sistema se nos desplegara una vista.

9.3 Menú del sistema

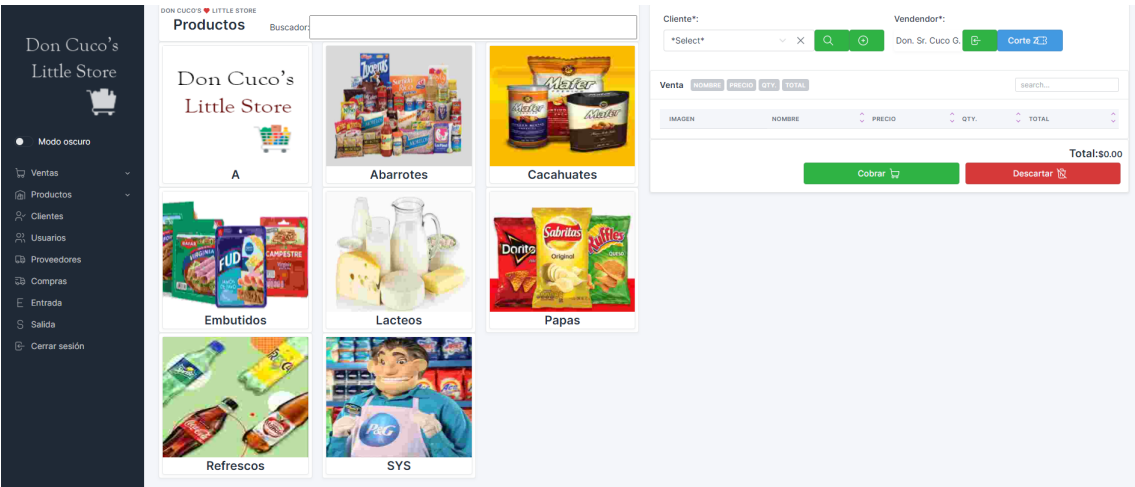


Figure 9.6: Menú en la Parte de Productos

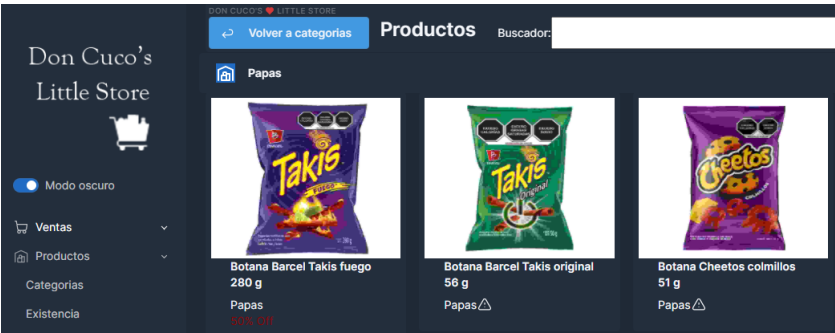


Figure 9.7: Vista de tema oscuro

En esta parte podemos ver que tenemos acceso a diferentes partes de la aplicación:

9.3.1 Ventas

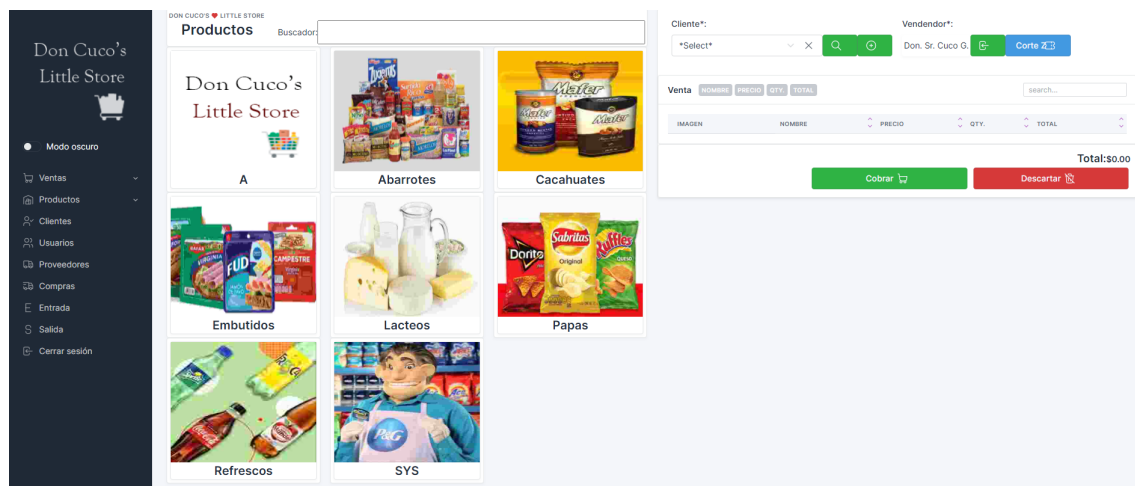


Figure 9.8: Vista de ventas

Se tiene una ventana la cual estará dividida en dos partes:

- En una parte se encuentran los accesos de rápido de categorías de los productos que se venden en la tienda.
- Al dar clic a la categoría de elección, esta mitad de pantalla cambiará para mostrar los productos que se encuentran dentro de la categoría previamente seleccionada.

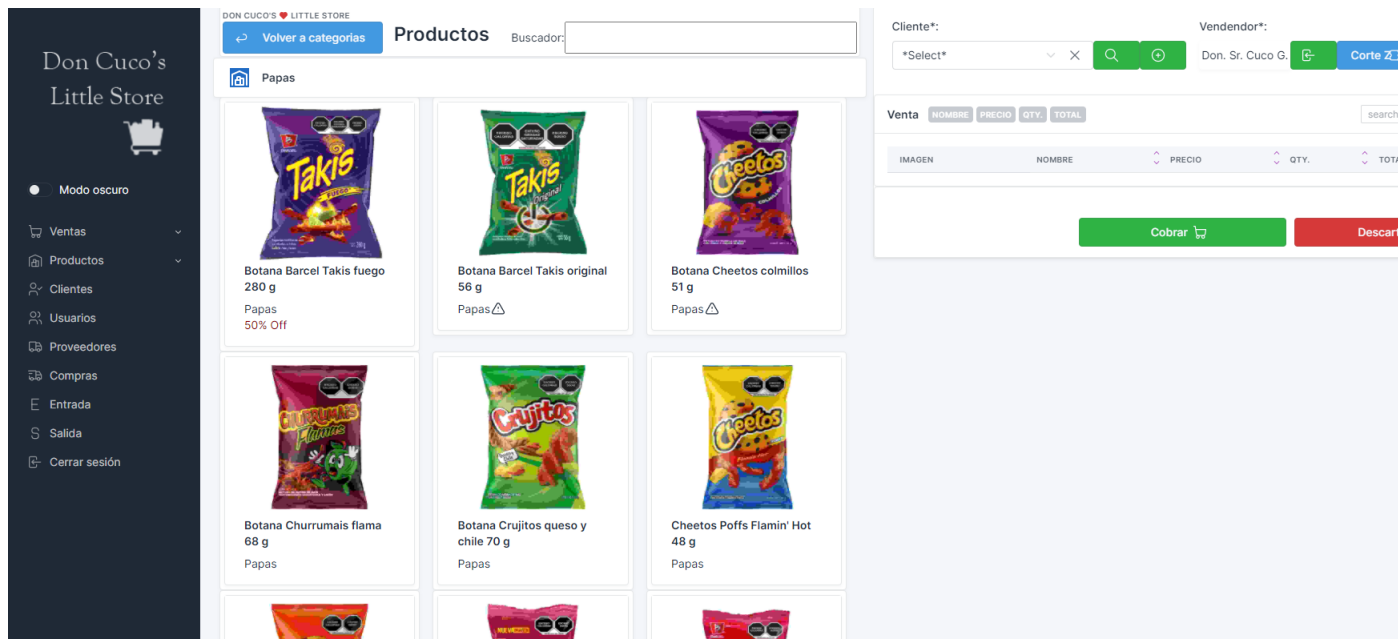


Figure 9.9: Vista de Categoría de Papas

Al estar dentro de una categoría seleccionada, en la parte superior izquierda, habrá un botón de diga “volver a las categorías” para regresar al menú de categorías.

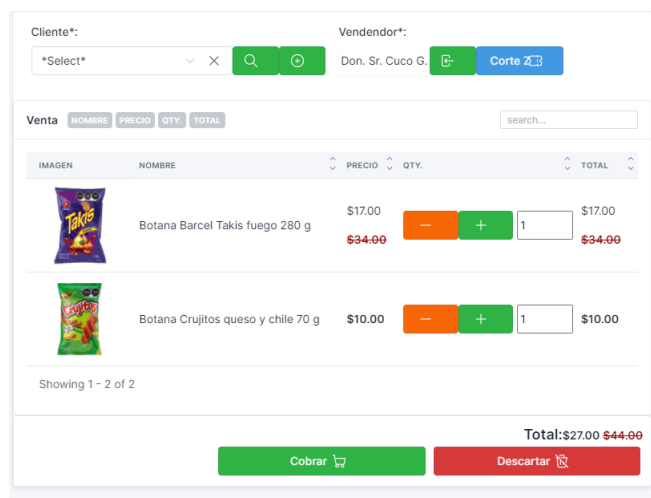


Figure 9.10: Vista de seleccion de productos

En esta vista se encuentra:

Cliente: Es una lista desplegable, en la cual se puede seleccionar el cliente a quien se le esté realizando una venta. Para hacer más fácil la búsqueda de un cliente ya existente, hay un botón de “lupa” del lado derecho, en donde al dar clic, se puede ingresar el nombre del cliente a buscar. En caso de no existir el nombre en la Base de Datos, se puede agregar un nuevo cliente, dando clic en el botón “más”.

Vendedor: Sirve para seleccionar el nombre del vendedor que en ese momento esté realizando una venta.

Venta: Es una sección en donde al seleccionar un producto (de la otra mitad que se mencionó anteriormente), la información de dicho producto se verá reflejada en esta sección. En caso de no querer seleccionar por medio del menú que se muestra en la otra mitad de la pantalla, se cuenta con un campo “search” para poder buscar el producto por su nombre. La búsqueda irá filtrando lo que encuentre registrado en la base de datos con la palabra o las palabras ingresadas. cuenta con los campos de: Imagen, Nombre, Precio, Cantidad y Total.

The screenshot displays a sales interface. At the top, there are fields for 'Cliente*' (with a dropdown menu showing 'Jonathan') and 'Vendedor*' (with a dropdown menu showing 'Don. Sr. Cuco G.'). Below these are search buttons with magnifying glass and plus icons. The main section is titled 'Venta' and contains a table with columns: IMAGEN, NOMBRE, PRECIO, QTY., and TOTAL. The table lists two products: 'Botana Barcel Takis fuego 280 g' and 'Botana Crujitos queso y chile 70 g'. Each product row includes an image, the product name, the unit price, a quantity selector (with minus, plus, and a text input field), and the total price. At the bottom, there is a summary bar showing 'Total: \$27.00 \$44.00' and two buttons: 'Cobrar' (green) and 'Descartar' (red).

IMAGEN	NOMBRE	PRECIO	QTY.	TOTAL
	Botana Barcel Takis fuego 280 g	\$17.00	1	\$17.00
	Botana Crujitos queso y chile 70 g	\$10.00	1	\$10.00

Showing 1 - 2 of 2

Total: \$27.00 \$44.00

Cobrar Descartar

Figure 9.11: Cobro de productos

En esta ventana se abrirá cuándo se selecciona un producto que no hay en existencia.

Aquí se encuentran dos botones:

Cancelar: Cancelara la opción que se haya seleccionado

Ok: Cerrará la ventana

Aquí se abrirá una ventana de confirmación de los productos seleccionados para la venta

- Tenemos los siguientes botones:
- Cancelar: Cancelara la opción de cobro
- Confirmar: Confirmará la venta de los productos seleccionados y de cobrará y abrirá lo siguiente:

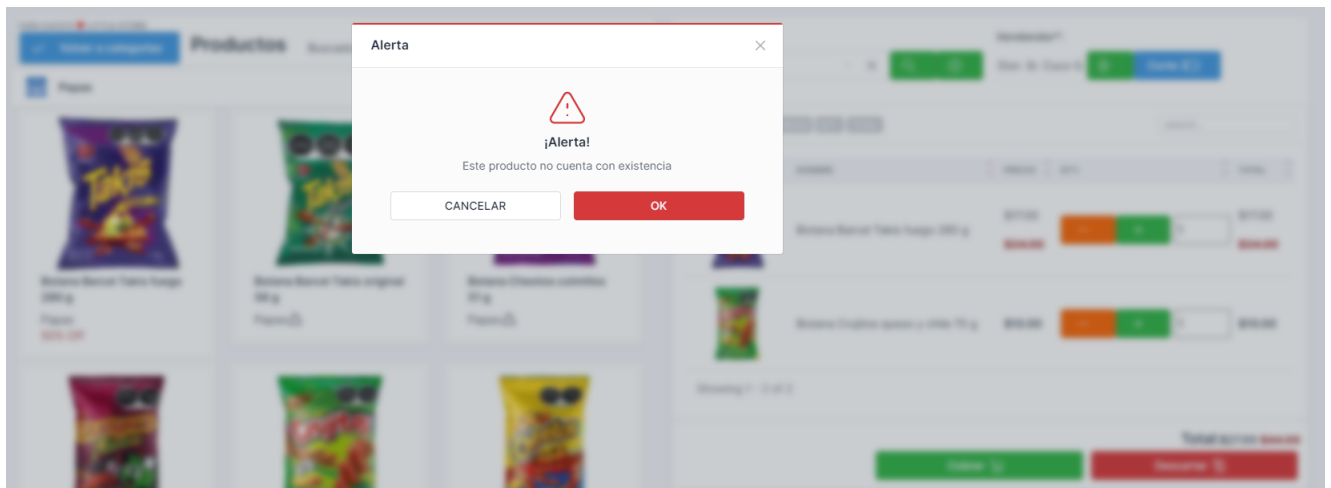


Figure 9.12: Ventana de Alerta

Finalizar venta

Efectivo

Efectivo Tarjeta Vale \$

\$ 30

Efectivo

Efectivo Tarjeta Vale \$

\$ 0

Efectivo

Efectivo Tarjeta Vale \$

\$ 0

Total \$27.00

Cubierto: \$30.00

Por cubrir: \$0.00

Cancelar

+ Confirmar

Figure 9.13: Ventana de cobro

Venta concluida Vendedor: Don. Sr. Cuco G. Ticket: # 1004 09-06-2021 04:57			
Cant.	Descripción	Precio	Importe
1	Botana Barcel Takis fuego 280 g	\$17.00	\$17.00
1	Botana Crujitos queso y chile 70 g	\$10.00	\$10.00
IMPUESTO			\$3.72
IMPORTE			\$23.28
TOTAL			\$27.00
VEINTISIETE MXN 00 /100			
Su pago:		Efectivo	\$30.00
		Efectivo	\$0.00
		Efectivo	\$0.00
Su cambio:			\$3.00
~~~~~ GRACIAS ~~~~~			

Figure 9.14: Ticket de venta realizada

Se confirma la venta realizada de los productos realizados

### 9.3.2 Productos

Esta parte de la aplicación abrirá la sección de productos y sus categorías que hay de estos mismos.

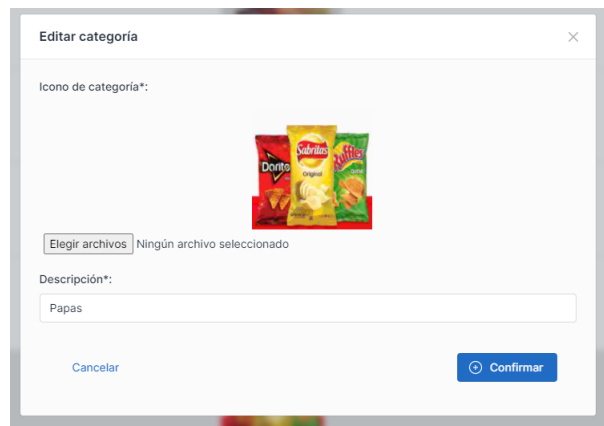


Figure 9.15: Ventana de Alta de una Categoría

En esta ventana se abrirá cuándo queramos dar de alta una categoría diferente que tengamos en nuestra tienda.

Cuenta con las siguientes opciones:

- Elegir archivos: Vista previa que tendremos para seleccionar más rápido un producto a la hora de venta
- Descripción: Una breve descripción de lo que es la categoría
- Cancelar: Cancelará la opción de dar de alta un producto
- Confirmar: Se creará la nueva categoría en nuestro sistema

Diferentes productos que podemos tener en nuestra aplicación

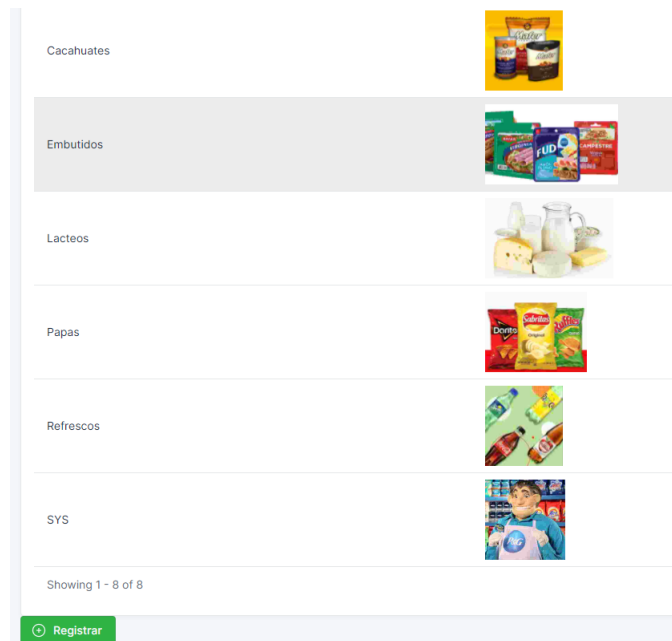


Figure 9.16: Categorías de los Productos

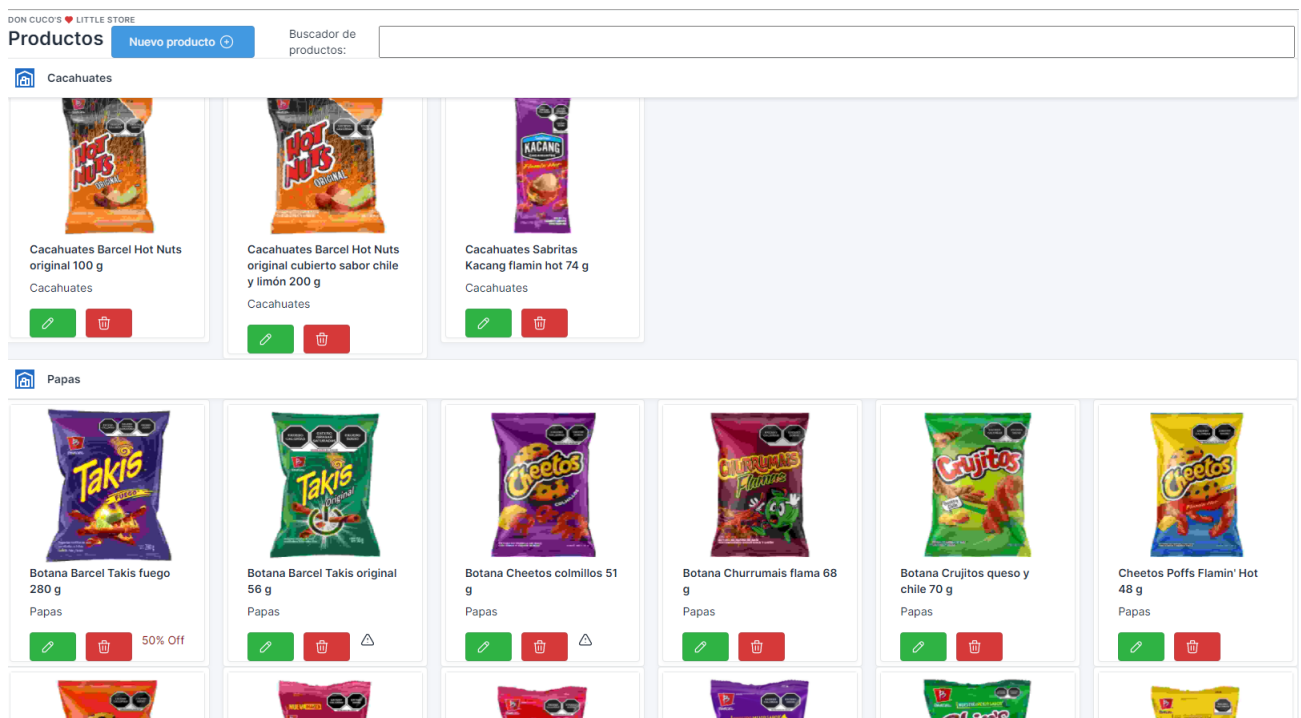


Figure 9.17: Vista de productos seleccionados ya caducados

### 9.3.3 Clientes







Clientes			search...
CLIENTES	NOTAS	IMAGEN	
Alma Partida Herrera			
Jonathan	No paga		
Jessica		Don Cuco's Little Store 	
Santiago Gonzalez Santiesteban	Enviar a calle 2, #145 casa morada		
Yessid		Don Cuco's Little Store 	
Showing 1 - 5 of 5			
<a href="#">+ Registrar</a>			

Figure 9.18: Vista general de nuestros clientes

Aquí encontraremos nuestra sección de clientes de nuestro menú este cuenta con un boton de registrar que nos abrirá la siguiente ventana:

Registrar cliente

Imagen de usuario*:  


Elegir archivos

Ningún archivo seleccionado

Nombre*:

Notas*:

[Cancelar](#)[+ Confirmar](#)

Figure 9.19: Ventana para dar de alta a un cliente

' Esta ventana tiene los siguientes botones:

- Elegir archivo: Podremos subir una imagen para identificar nuestros clientes

- Nombre: En esta se pondrá el nombre de nuestros clientes
- Cancelar: Cancelará la alta de un cliente
- Confirmar: Dará de alta nuestro cliente que queramos registrar

### 9.3.4 Usuarios

Figure 9.20: Ventana de Dar de alta un usuario

En esta parte se definirán los roles y privilegios que tendrán nuestros usuarios que vayan atender.

En esta parte damos de alta a nuestros usuarios, al igual cuenta con una opción para definir una imagen y nos sea más fácil identificar.

- Elegir archivos: Vista previa de imagen de nuestro usuario
- Nombre: Nombre de nuestro usuario
- Botones de privilegios:
- Cancelar: Cancelará la opción de dar de alta un producto
- Confirmar: Se creará la nueva categoría en nuestro sistema

Aquí se podrá ver nuestros usuarios y que privilegios tendrán dentro de la aplicación, esta vista solo será por el Administrador.

Usuarios

NICKNAME

NOMBRE

PRIVILEGIO SALIDAS

PRIVILEGIO ENTRADAS




DERECHOS SOBRE USUARIOS

PRIVILEGIO REPORTE

SOLO LECTURA

PRIVILEGIO VENTAS

search...

NICKNAME	NOMBRE	PRIVILEGIO SALIDAS	PRIVILEGIO ENTRADAS	DERECHOS SOBRE USUARIOS	PRIVILEGIO REPORTE	SOLO LECTURA	PRIVILEGIO VENTAS	IMAGEN
CUCO	Don. Sr. Cuco G.	<div><div></div>Privilegio Salidas</div>	<div><div></div>Privilegio Entradas</div>	<div><div></div>Derechos sobre usuarios</div>	<div><div></div>Privilegio Reporte</div>	<div><div></div>Solo lectura</div>	<div><div></div>Privilegio Ventas</div>	<div></div>
Jon	Jonathan	<div><div></div>Privilegio Salidas</div>	<div><div></div>Privilegio Entradas</div>	<div><div></div>Derechos sobre usuarios</div>	<div><div></div>Privilegio Reporte</div>	<div><div></div>Solo lectura</div>	<div><div></div>Privilegio Ventas</div>	<div><div>Don Cuco's Little Store</div><div></div></div>
CAJERO01	Cajero de la caja 1	<div><div></div>Privilegio Salidas</div>	<div><div></div>Privilegio Entradas</div>	<div><div></div>Derechos sobre usuarios</div>	<div><div></div>Privilegio Reporte</div>	<div><div></div>Solo lectura</div>	<div><div></div>Privilegio Ventas</div>	<div><div>Don Cuco's Little Store</div><div></div></div>


Showing 1 - 3 of 3

Registrar

Figure 9.21: Usuarios/Roles

Registrar proveedor

Imagen de usuario*:



Elegir archivos

Ningún archivo seleccionado

Nombre*:

Coca Cola Company

Celular*:

800-704-4400

Notas*:

El camión pasa el día 17 del mes

Cancelar

Confirmar

Figure 9.22: Venta para dar de Alta a nuestros Proveedores

### 9.3.5 Proveedores

Cuenta con las siguientes opciones:

- Elegir archivos: Vista previa que tendremos para identificar con facilidad nuestro proveedor
- Nombre: Nombre de la empresa de nuestro proveedor
- Celular: Tener el contacto de nuestro proveedor
- Cancelar: Cancelará la opción de dar de alta un producto

- Confirmar: Se creará la nueva categoría en nuestro sistema







Proveedores <span>NOMBRE</span> <span>CELULAR</span> <span>NOTAS</span>				search...
NOMBRE	CELULAR	NOTAS	IMAGEN	
Coca Cola Company	800-704-4400	El camión pasa el día 17 del mes		
La Coste(a	(52) (52) 58363636			
Sabritas	01800 90 19 500			
Barcel	01 800 910 2030	Preguntar antes por las promociones		
Peñafiel	443355774			
Nutrioli				
Showing 1 - 6 of 6				
<a href="#">Registar</a>				

Figure 9.23: Vista de nuestra lista de proveedores

Vista rápida de nuestros proveedores con su respectivo número

## 9.3.6 Compras

Compras

FECHAUSUARIOOBSERVACIONES

search...

FECHA	USUARIO	OBSERVACIONES
6/7/2021 12:18:01 AM	Don. Sr. Cuco G.	
6/7/2021 4:59:00 PM	Don. Sr. Cuco G.	Se compro una caja
6/7/2021 5:34:40 PM	Don. Sr. Cuco G.	Unos takis
6/7/2021 9:10:32 PM	Don. Sr. Cuco G.	Compra de una caja de frijoles

Showing 1 - 4 of 4

Partidas de la compra

CÓDIGOPRODUCTOPROVEEDORCANTIDADEXISTENCIA INICIALEXISTENCIA ACTUALVALOR

search...

código	PRODUCTO	PROVEEDOR	CANTIDAD	EXISTENCIA INICIAL	EXISTENCIA ACTUAL	VALOR
00750101700436	Frijoles	La Coste[a	50	0	50	695
00750103045995	Cacahuates Barcel Hot Nuts original 100 g	Barcel	10	4	14	170

Showing 1 - 2 of 2

Registrar

Figure 9.24: Vista de nuestras compras que se han realizado por nuestros usuarios

En esta parte de la aplicación podemos tener mejor control de lo que saldrá de nuestra tienda y



los gastos que se generarán.  
La ventana tiene los siguientes botones:

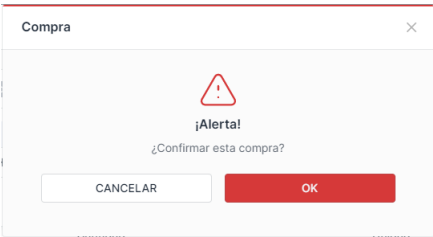


Figure 9.25: Ventana de alerta para confirmar nuestra compra

- Cancelar: Cancelará la compra
- Ok: Seguirá la compra
- X: Cerrara la operación

The image shows a "Nueva compra" (New purchase) window. It features a table of items with columns for "CODIGO", "NOMBRE", "CANTIDAD", "ACTUAL", and "POSTERIOR". Below the table, there are input fields for "Artículo*", "Cantidad*", "Unidad*", "Observaciones*", and "Proveedor*", along with buttons for "Seleccionar", "Confirmar compra", and a search icon.

Movimientos	CODIGO	NOMBRE	CANTIDAD	ACTUAL	POSTERIOR
	18	Cacahuates Barcel Hot Nuts original cubierto sabor chile y limón 200 g	1	3	4
	29	Cacahuates Sabritas Kacang flamin hot 74 g	1	1	2
	23	Cheetos Torciditos Torciditos 145 g	1	4	5
	6	Papas fritas Barcel Chips sabor jalapeño 350 g	1	-10	-9

Showing 1 - 4 of 4

Artículo*:  Cantidad*:  Unidad*:  Seleccionar

Observaciones*:  Proveedor*:  Confirmar compra

Figure 9.26: Ventana de nueva compra

Corte Z

Don Cucu's

Little Store



Corte Z-1006

Fecha:

09/06/2021 17:05

Ticket Inicial:

1004

Ticket Final:

1004

Numero de Tickets:

1.00

Ticket Promedio:

\$27.00

Ingresos

Concepto	Importe
Cash	\$30.00
Total	\$30.00

Ventas por ticket

Hora	Ticket	Total	Cambio
09/06/2021 16:57	1004	\$27.00	\$3.00
		\$27.00	\$3.00

Figure 9.27: Ticket de nuestra compra

### 9.3.7 Entrada/Salida

CODIGO	NOMBRE	CANTIDAD	ACTUAL	POSTERIOR
00750047802118	Botana Crujitos queso y chile 70 g	1	1	2
00750047801797	Cheetos Poffs Flamin' Hot 48 g	1	1	2
00750101700436	Frijoles	1	50	51
00075752800186	Papas fritas Barcel Chips fuego 55 g	1	0	1

Showing 1 - 4 of 4

Artículo*: Cacahuates Barcel Hot Nuts orig  Cantidad*: 1 Unidad*: Pieza

Observaciones*:  Concepto*:

Confirmar movimiento →

Figure 9.28: Entrada de nuestros productos

Se tiene una ventana con los siguientes componentes:

1. **Cuadro de búsqueda:** Es un cuadro de búsqueda de productos, que permitirá realizar una búsqueda dentro de todo el inventario, remarcando y mostrando los que coinciden con la búsqueda, o mostrar un mensaje donde indique que no coincide ninguna búsqueda con lo ingresado.
  - **Botón de Ordenar:** Es un filtro que permitirá ordenar los productos en el sistema, lo hará por todos los campos disponibles, id, alfabéticamente, según el nombre del proveedor.
2. **Campo de código:** Este campo permite tener una visualización de un identificador único a cada producto ingresado al inventario.
3. **Campo de texto:** Permitirá visualizar el nombre del producto, que se dio de alta en el inventario.
4. **Campo de texto:** Servirá para visualizar el número de los productos actuales en sistema.
5. **Campo de texto:** Servirá para visualizar el número posterior al movimiento de los productos actuales en sistema.
6. **Botón Cerrar:** Es el botón que cierra la ventana. Esta acción puede ser por medio de un botón o realizarse desde una equis (x) en el extremo superior derecho de la ventana actual.

Se tiene una ventana con los siguientes componentes:

1. **Cuadro de búsqueda:** Es un cuadro de búsqueda de productos, que permitirá realizar una búsqueda dentro de todo el inventario, remarcando y mostrando los que coinciden con la búsqueda, o mostrar un mensaje donde indique que no coincide ninguna búsqueda con lo ingresado.

Movimientos					search...	
FECHA	USUARIO	CONCEPTO	TIPO	OBSERVACIONES		
6/6/2021 9:52:31 PM	Don. Sr. Cuco G.	VTA - Venta	S	Ticket #0		
6/6/2021 9:53:55 PM	Don. Sr. Cuco G.	VTA - Venta	S	Ticket #2		
6/6/2021 9:55:21 PM	Don. Sr. Cuco G.	VTA - Venta	S	Ticket #3		
6/6/2021 11:48:26 PM	Don. Sr. Cuco G.	VTA - Venta	S	Ticket #4		
6/6/2021 11:49:53 PM	Don. Sr. Cuco G.	VTA - Venta	S	Ticket #5		
6/6/2021 11:51:40 PM	Don. Sr. Cuco G.	VTA - Venta	S	Ticket #6		
6/7/2021 12:01:57 AM	Don. Sr. Cuco G.	VTA - Venta	S	Ticket #7		
6/7/2021 12:11:48 AM	Don. Sr. Cuco G.	VTA - Venta	S	Ticket #8		
6/7/2021 12:13:36 AM	Don. Sr. Cuco G.	VTA - Venta	S	Ticket #9		
6/7/2021 12:15:06 AM	Don. Sr. Cuco G.	VTA - Venta	S	Ticket #10		
Showing 1 - 10 of 25					1 2 Last	

Partes del movimiento						search...	
CÓDIGO	PRODUCTO	CANTIDAD	EXISTENCIA INICIAL	EXISTENCIA ACTUAL	VALOR		
00750047801797	Cheetos Poffs Flamin' Hot 48 g	1	4	3	-10		
00750047802118	Botana Crujitos queso y chile 70 g	1	4	3	-10		
00750100016437	Botana Barcel Takis original 56 g	1	-4	-5	-13.5		
00750047802198	Botana Cheetos colmillos 51 g	1	1	-1	-10		
Showing 1 - 4 of 4							

⌂ Registrar

Figure 9.29: Salida de nuestros productos

- Campo de texto:** Permitirá visualizar el nombre del producto, que se dio de alta en el inventario.
- Registrar:** Registrara una nueva salida del producto que seleccionemos
- Botón Cerrar:** Es el botón que cierra la ventana. Esta acción puede ser por medio de un botón o realizarse desde una equis (x) en el extremo superior derecho de la ventana actual.

## 9.4 Reportes

Reporte de ventas

Desde

←

June 2021

→

SUN

MON

TUE

WED

THU

FRI

SAT

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

1

2

3

4

5

6

7

8

9

10

Todas las fechas

Cancelar

Hasta

←

June 2021

→

SUN

MON

TUE

WED

THU

FRI

SAT

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

1

2

3

4

5

6

7

8

9

10

Confirmar

Figure 9.30: Filtro de reportes

Esta ventana nos dará la opción de checar de que fecha a que fecha queremos ver los movimientos, la existencia de nuestros productos y el reporte de ventas de nuestra tienda.

Existencia

CÓDIGO

NOMBRE

EXISTENCIA

MÍNIMO

MÁXIMO

search...

CÓDIGO	NOMBRE	EXISTENCIA	MÍNIMO	MÁXIMO	STATUS
7501011101454	Sabritas Originales - 42 g	15.00	10.00	20.00	Ok
00000007500761	Refresco Coca Cola botella de 600 ml	18.00	25.00	35.00	inventario mínimo
00750105530377	Refresco Fanta sabor naranja 600 ml	7.00	5.00	20.00	Ok
00750101116775	Rancheritos Max 240 g	2.00	3.00	5.00	inventario mínimo
00750101700436	Frijoles	50.00	12.00	20.00	Sobre inventario
00750100039840	Papas fritas Barcel Chips sabor jalapeño 350 g	10.00	7.00	21.00	inventario mínimo
00075752802703	Botana Barcel Takis fuego 280 g	19.00	25.00	35.00	inventario mínimo
00075752800186	Papas fritas Barcel Chips fuego 55 g	0.00	5.00	20.00	inventario mínimo
00750100026404	Papas fritas Barcel Chips sal de mar 60 g	8.00	5.00	20.00	Ok
00075752804420	Papas fritas Barcel Chips adobadas 55 g	10.00	15.00	30.00	inventario mínimo
00750105533998	Refresco Sidral Mundet sabor manzana 600 ml	9.00	15.00	20.00	inventario mínimo
00750105530562	Refresco Sprite sabor lima-limón 600 ml	9.00	15.00	30.00	inventario mínimo
00750107383050	Agua mineral Peñafiel 600 ml	43.00	16.00	30.00	Sobre inventario
00750107112007	Refresco Squirt sabor toronja 2 l	12.00	5.00	10.00	Sobre inventario
00750107112030	Refresco Squirt sabor toronja 3 l	7.00	2.00	5.00	inventario mínimo
00750105534000	Refresco Sidral Mundet sabor manzana 2 l	7.00	15.00	20.00	inventario mínimo
00750107112018	Refresco Squirt sabor toronja 600 ml	3.00	15.00	20.00	inventario mínimo

Figure 9.31: Reporte de Existencia de nuestros productos

09/06/2021 16:58

**Registro de movimientos por usuario**  
04/07/2021 al 16/06/2021

Don Cuco's Little Store

Fecha	Usuario que realizó el movimiento	Operación	Tipo
06/06/2021 21:52	Don. Sr. Cuco G.	Movimiento #1	Salida
06/06/2021 21:53	Don. Sr. Cuco G.	Ticket #2	Venta
06/06/2021 21:53	Don. Sr. Cuco G.	Movimiento #2	Salida
06/06/2021 21:55	Don. Sr. Cuco G.	Ticket #3	Venta
06/06/2021 21:55	Don. Sr. Cuco G.	Movimiento #3	Salida
06/06/2021 23:48	Don. Sr. Cuco G.	Ticket #4	Venta
06/06/2021 23:48	Don. Sr. Cuco G.	Movimiento #4	Salida
06/06/2021 23:49	Don. Sr. Cuco G.	Ticket #5	Venta
06/06/2021 23:49	Don. Sr. Cuco G.	Movimiento #5	Salida
06/06/2021 23:51	Don. Sr. Cuco G.	Ticket #6	Venta
06/06/2021 23:51	Don. Sr. Cuco G.	Movimiento #6	Salida
07/06/2021 00:01	Don. Sr. Cuco G.	Ticket #7	Venta
07/06/2021 00:01	Don. Sr. Cuco G.	Movimiento #7	Salida
07/06/2021 00:11	Don. Sr. Cuco G.	Ticket #8	Venta
07/06/2021 00:11	Don. Sr. Cuco G.	Movimiento #8	Salida
07/06/2021 00:13	Don. Sr. Cuco G.	Ticket #9	Venta
07/06/2021 00:13	Don. Sr. Cuco G.	Movimiento #9	Salida
07/06/2021 00:15	Don. Sr. Cuco G.	Ticket #10	Venta
07/06/2021 00:15	Don. Sr. Cuco G.	Movimiento #10	Salida
07/06/2021 00:18	Don. Sr. Cuco G.	Compra #1	Compra
07/06/2021 00:18	Don. Sr. Cuco G.	Movimiento #11	Entrada
07/06/2021 00:29	Don. Sr. Cuco G.	Ticket #11	Venta
07/06/2021 00:29	Don. Sr. Cuco G.	Movimiento #12	Salida

Figure 9.32: Reporte de movimientos de nuestros productos

6/9/2021

**Reporte de ventas**  
05/07/2021 al 16/06/2021

Don Cuco's Little Store

Ventas del periodo: 22    Total ventas: \$2,317.37    Ticket Promedio: \$105.34

Fecha	Ticket	Vendedor	Cliente	Importe	Impuesto	Total
06/06/2021	2	Don. Sr. Cuco G.	Cliente Genérico	\$34.91	\$5.59	\$40.50
06/06/2021	3	Don. Sr. Cuco G.	Cliente Genérico	\$46.12	\$7.38	\$53.50
06/06/2021	4	Don. Sr. Cuco G.	Cliente Genérico	\$14.57	\$2.33	\$16.90
06/06/2021	5	Don. Sr. Cuco G.	Cliente Genérico	\$2.59	\$0.41	\$3.00
06/06/2021	6	Don. Sr. Cuco G.	Cliente Genérico	\$2.59	\$0.41	\$3.00
07/06/2021	7	Don. Sr. Cuco G.	Cliente Genérico	\$12.93	\$2.07	\$15.00
07/06/2021	8	Don. Sr. Cuco G.	Cliente Genérico	\$2.59	\$0.41	\$3.00
07/06/2021	9	Don. Sr. Cuco G.	Cliente Genérico	\$1.29	\$0.21	\$1.50
07/06/2021	10	Don. Sr. Cuco G.	Cliente Genérico	\$1.29	\$0.21	\$1.50
07/06/2021	11	Don. Sr. Cuco G.	Cliente Genérico	\$306.03	\$48.97	\$355.00
07/06/2021	12	Don. Sr. Cuco G.	Cliente Genérico	\$560.34	\$89.66	\$650.00
07/06/2021	13	Don. Sr. Cuco G.	Cliente Genérico	\$9.05	\$1.45	\$10.50
07/06/2021	14	Don. Sr. Cuco G.	Cliente Genérico	\$129.31	\$20.69	\$150.00
07/06/2021	15	Don. Sr. Cuco G.	Cliente Genérico	\$40.52	\$6.48	\$47.00
07/06/2021	16	Don. Sr. Cuco G.	Jonathan	\$97.08	\$15.53	\$112.61
07/06/2021	17	Don. Sr. Cuco G.	Cliente Genérico	\$13.79	\$2.21	\$16.00
07/06/2021	18	Don. Sr. Cuco G.	Jessica	\$116.82	\$18.69	\$135.51
07/06/2021	19	Don. Sr. Cuco G.	Alma Partida Herrera	\$66.29	\$10.61	\$76.90
07/06/2021	20	Don. Sr. Cuco G.	Jonathan	\$254.22	\$40.68	\$294.90
07/06/2021	21	Don. Sr. Cuco G.	Cliente Genérico	\$188.31	\$30.13	\$218.44
07/06/2021	22	Jonathan	Cliente Genérico	\$73.80	\$11.81	\$85.61
09/06/2021	1004	Don. Sr. Cuco G.	Jonathan	\$23.28	\$3.72	\$27.00

Figure 9.33: Reporte de Ventas de nuestros productos

## 9.5 Movimientos

Movimientos

FECHAUSUARIOCONCEPTOTIPOOBSERVACIONES

search...

FECHA	USUARIO	CONCEPTO	TIPO	OBSERVACIONES
6/7/2021 12:18:05 AM	Don. Sr. Cuco G.	COM - Compra	E	Compra #1
6/7/2021 4:59:00 PM	Don. Sr. Cuco G.	COM - Compra	E	Compra #2
6/7/2021 5:34:40 PM	Don. Sr. Cuco G.	COM - Compra	E	Compra #3
6/7/2021 9:10:32 PM	Don. Sr. Cuco G.	COM - Compra	E	Compra #4
6/7/2021 9:11:46 PM	Don. Sr. Cuco G.	A+ - Ajuste	E	
6/9/2021 5:02:48 PM	Don. Sr. Cuco G.	COM - Compra	E	Compra #1002

Showing 1 - 6 of 6

Partes del movimiento

CÓDIGOPRODUCTOCANTIDADEXISTENCIA INICIALEXISTENCIA ACTUALVALOR

search...

CÓDIGO	PRODUCTO	CANTIDAD	EXISTENCIA INICIAL	EXISTENCIA ACTUAL	VALOR
--------	----------	----------	--------------------	-------------------	-------

Registrar

Figure 9.34: Movimiento de existencia de productos

Movimientos

FECHAUSUARIOCONCEPTOTIPOOBSERVACIONES

search...

FECHA	USUARIO	CONCEPTO	TIPO	OBSERVACIONES
6/7/2021 12:18:05 AM	Don. Sr. Cuco G.	COM - Compra	E	Compra #1
6/7/2021 4:59:00 PM	Don. Sr. Cuco G.	COM - Compra	E	Compra #2
6/7/2021 5:34:40 PM	Don. Sr. Cuco G.	COM - Compra	E	Compra #3
6/7/2021 9:10:32 PM	Don. Sr. Cuco G.	COM - Compra	E	Compra #4
6/7/2021 9:11:46 PM	Don. Sr. Cuco G.	A+ - Ajuste	E	
6/9/2021 5:02:48 PM	Don. Sr. Cuco G.	COM - Compra	E	Compra #1002

Showing 1 - 6 of 6

Partes del movimiento

CÓDIGOPRODUCTOCANTIDADEXISTENCIA INICIALEXISTENCIA ACTUALVALOR

search...

CÓDIGO	PRODUCTO	CANTIDAD	EXISTENCIA INICIAL	EXISTENCIA ACTUAL	VALOR
00750107383050	Agua mineral Peñafiel 600 ml	10	25	35	110
00750047802088	Botana Churruimalis flama 68 g	10	13	23	100

Showing 1 - 2 of 2

Registrar

Figure 9.35: Partes de los movimientos de productos

En esta parte se visualizarán los movimientos realizados por el usuario, se verá quién hizo el ajuste o movimiento del productos y cuántos quedan en existencia actual e inicial.

## 9.6 Manual Técnico

```
1  /* CREA UNA BASE DE DATOS EN BLANCO*/
2  CREATE DATABASE DON_CUCO
3  GO
4  /* UTILIZA ESTA NUEVA BASE DE DATOS*/
5  USE DON_CUCO
6  GO
```

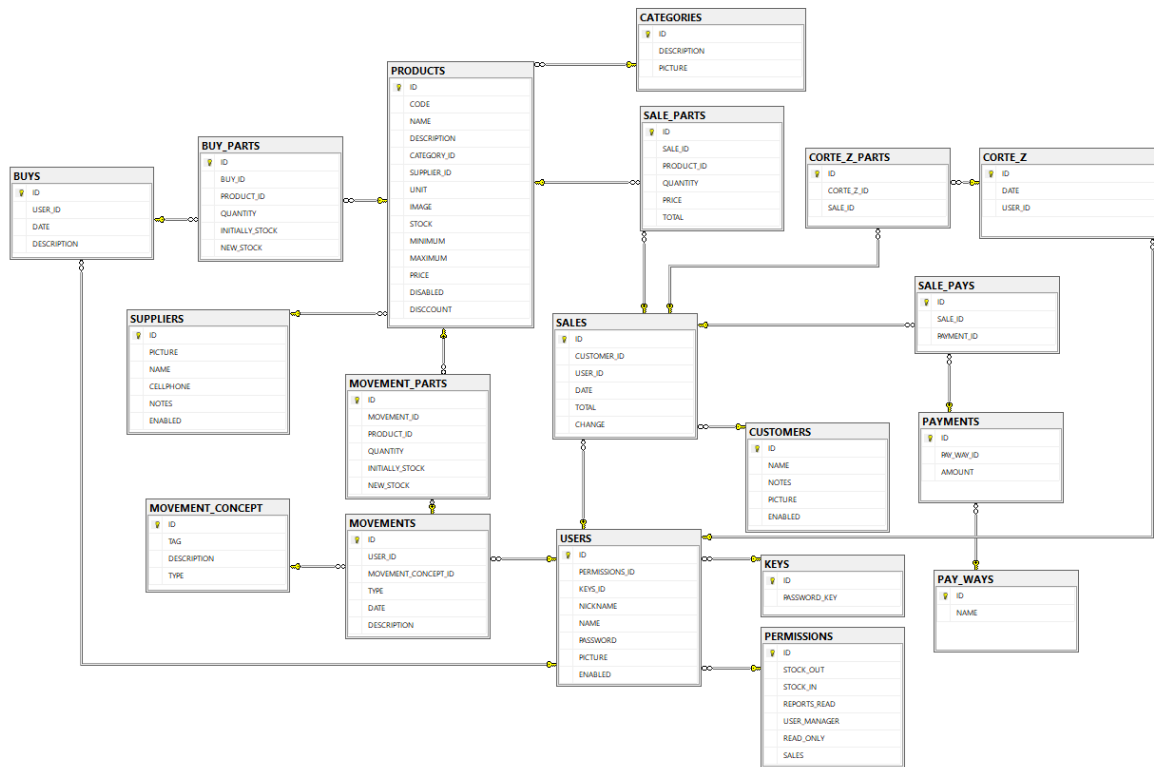


Figure 9.36: Diseño lógico

```

7  ALMACENA LA INFORMACION SOBRE LAS CATEGORIAS DE PRODUCTOS\\
8  CREATE TABLE CATEGORIES
9  (
10     ID INT PRIMARY KEY IDENTITY(1,1),
11     DESCRIPTION VARCHAR(100) NOT NULL UNIQUE,
12     PICTURE VARCHAR(MAX)
13 );
14 /*INFORMACION DE PROVEEDORES*/
15 CREATE TABLE SUPPLIERS(
16     ID INT PRIMARY KEY IDENTITY(1,1),
17     PICTURE VARCHAR(MAX),
18     NAME VARCHAR(100) NOT NULL,
19     CELLPHONE VARCHAR(100),
20     NOTES VARCHAR(MAX),
21     ENABLED BIT NOT NULL DEFAULT 0
22 );
23
24 /*INFORMACION DE PRODUCTOS*/
25 CREATE TABLE PRODUCTS(
26     ID INT PRIMARY KEY IDENTITY(1,1),
27     CODE VARCHAR(100) UNIQUE NOT NULL,
28     NAME VARCHAR(100) NOT NULL,
29     DESCRIPTION VARCHAR(MAX),
30     CATEGORY_ID INT FOREIGN KEY REFERENCES CATEGORIES NOT NULL,
31     SUPPLIER_ID INT FOREIGN KEY REFERENCES SUPPLIERS NOT NULL,
32     UNIT VARCHAR(100) NOT NULL,

```



```

33     IMAGE VARCHAR(MAX),
34     STOCK REAL NOT NULL DEFAULT 0,
35     MINIMUM REAL NOT NULL DEFAULT 0,
36     MAXIMUM REAL NOT NULL DEFAULT 0,
37     PRICE REAL NOT NULL DEFAULT 0,
38     DISABLED BIT NOT NULL DEFAULT 0,
39     DISCCOUNT INTEGER NOT NULL DEFAULT 0
40 );
41 /*INFORMACION/MATRIZ DE PERMISOS DE USUARIO*/
42 CREATE TABLE PERMISSIONS
43 (
44     ID INT PRIMARY KEY IDENTITY (1,1),
45     STOCK_OUT BIT DEFAULT 0,
46     STOCK_IN BIT DEFAULT 0,
47     REPORTS_READ BIT DEFAULT 0,
48     USER_MANAGER BIT DEFAULT 0,
49     READ_ONLY BIT DEFAULT 0,
50     SALES BIT DEFAULT 0,
51 );
52 /* LLAVES,TOKENS DE SEGURIDAD PARA DESENCRIPTAR PASSWORDS DE USUARIO*/
53 CREATE TABLE KEYS(
54     ID INT PRIMARY KEY IDENTITY(1,1),
55     PASSWORD_KEY VARCHAR(37) DEFAULT NEWID()
56 );
57 /*INFORMACION SOBRE USUARIOS DEL SITEMA */
58 CREATE TABLE USERS(
59     ID INT PRIMARY KEY IDENTITY(1,1),
60     PERMISSIONS_ID INT FOREIGN KEY REFERENCES PERMISSIONS NOT NULL,
61     KEYS_ID INT FOREIGN KEY REFERENCES KEYS NOT NULL,
62     NICKNAME VARCHAR(100) NOT NULL UNIQUE,
63     NAME VARCHAR(100) NOT NULL,
64     PASSWORD VARBINARY(MAX) NOT NULL,
65     PICTURE VARCHAR(MAX) DEFAULT NULL,
66     ENABLED BIT NOT NULL DEFAULT 0
67 );
68 /*LISTADO DE CONCEPTOS DE MOVIMIENTOS*/
69 CREATE TABLE MOVEMENT_CONCEPT
70 (
71     ID INT PRIMARY KEY IDENTITY(1,1),
72     TAG VARCHAR(10) NOT NULL,
73     DESCRIPTION VARCHAR(100),
74     TYPE VARCHAR(1)
75 );
76 /*CABECERA DE LA COMPRA*/
77 CREATE TABLE BUYS( --1 para que te de el id de movement
78     ID INT PRIMARY KEY IDENTITY(1,1),
79     USER_ID INT FOREIGN KEY REFERENCES USERS NOT NULL,
80     DATE DATETIME NOT NULL DEFAULT GETDATE(),
81     DESCRIPTION VARCHAR(100)
82 );
83 /*PARTIDAS DE LA COMPRA*/
84 CREATE TABLE BUY_PARTS(
85     ID INT PRIMARY KEY IDENTITY(1,1),
86     BUY_ID INT FOREIGN KEY REFERENCES BUYS NOT NULL,
87     PRODUCT_ID INT FOREIGN KEY REFERENCES PRODUCTS NOT NULL,
88     QUANTITY REAL NOT NULL DEFAULT 0,
89     INITIALLY_STOCK REAL NOT NULL,

```

```

90     NEW_STOCK REAL NOT NULL
91 );
92 /*CABECERA DE EL MOVIMIENTO (E/S)*/
93 CREATE TABLE MOVEMENTS( --1 para que te de el id de movement
94     ID INT PRIMARY KEY IDENTITY(1,1),
95     USER_ID INT FOREIGN KEY REFERENCES USERS NOT NULL,
96     MOVEMENT_CONCEPT_ID INT FOREIGN KEY REFERENCES MOVEMENT_CONCEPT NOT NULL,
97     TYPE VARCHAR(1) NOT NULL,
98     DATE DATETIME NOT NULL DEFAULT GETDATE(),
99     DESCRIPTION VARCHAR(100)
100 );
101 /*PARTIDAS DEL MOVIMIENTO*/
102 CREATE TABLE MOVEMENT_PARTS(
103     ID INT PRIMARY KEY IDENTITY(1,1),
104     MOVEMENT_ID INT FOREIGN KEY REFERENCES MOVEMENTS NOT NULL,
105     PRODUCT_ID INT FOREIGN KEY REFERENCES PRODUCTS NOT NULL,
106     QUANTITY REAL NOT NULL DEFAULT 0,
107     INITIALLY_STOCK REAL NOT NULL,
108     NEW_STOCK REAL NOT NULL
109 );
110 /*REGISTRO DE CLIENTES*/
111 CREATE TABLE CUSTOMERS(
112     ID INT PRIMARY KEY IDENTITY(1,1),
113     NAME VARCHAR(100) NOT NULL,
114     NOTES VARCHAR(MAX),
115     PICTURE VARCHAR(MAX),
116     ENABLED BIT NOT NULL DEFAULT 0
117 );
118 /*LISTADO DE LAS FORMAS DE PAGO DISPONIBLES*/
119 CREATE TABLE PAY_WAYS(
120     ID INT PRIMARY KEY IDENTITY(1,1),
121     NAME VARCHAR(50)
122 );
123 /*LISTADO DE PAGOS RECIBIDOS */
124 CREATE TABLE PAYMENTS
125 (
126     ID INT PRIMARY KEY IDENTITY(1,1),
127     PAY_WAY_ID INT FOREIGN KEY REFERENCES PAY_WAYS,
128     AMOUNT FLOAT NOT NULL DEFAULT 0
129 );
130 /*CABECERA DE UNA VENTA*/
131 CREATE TABLE SALES(
132     ID INT PRIMARY KEY IDENTITY(1,1),
133     CUSTOMER_ID INT FOREIGN KEY REFERENCES CUSTOMERS NOT NULL,
134     USER_ID INT FOREIGN KEY REFERENCES USERS NOT NULL,
135     DATE DATETIME DEFAULT GETDATE(),
136     TOTAL REAL,
137     CHANGE REAL
138 );
139 /*PARTIDAS DE UNA VENTA*/
140 CREATE TABLE SALE_PARTS(
141     ID INT PRIMARY KEY IDENTITY(1,1),
142     SALE_ID INT FOREIGN KEY REFERENCES SALES NOT NULL,
143     PRODUCT_ID INT FOREIGN KEY REFERENCES PRODUCTS NOT NULL,
144     QUANTITY REAL,
145     PRICE REAL,
146     TOTAL REAL

```

```

147 );
148 /*PAGOS REALIZADOS EN UNA VENTA ESPECIFICA*/
149 CREATE TABLE SALE_PAYS(
150     ID INT PRIMARY KEY IDENTITY(1,1),
151     SALE_ID INT FOREIGN KEY REFERENCES SALES NOT NULL,
152     PAYMENT_ID INT FOREIGN KEY REFERENCES PAYMENTS
153 );
154 GO
155 /*REGISTRO DE CORTE DE CAJA DE CIERRE*/
156 CREATE TABLE CORTE_Z(
157     ID INT PRIMARY KEY IDENTITY(1,1),
158     DATE DATETIME,
159     USER_ID INT FOREIGN KEY REFERENCES USERS NOT NULL);
160 GO
161 /*LISTADO DE VENTAS POR CORTE DE CAJA*/
162 CREATE TABLE CORTE_Z_PARTS(
163     ID INT PRIMARY KEY IDENTITY(1,1),
164     CORTE_Z_ID INT FOREIGN KEY REFERENCES CORTE_Z NOT NULL,
165     SALE_ID INT FOREIGN KEY REFERENCES SALES NOT NULL,
166 );
167 GO
168 /*REALIZA UN NUEVO CORTE DE CAJA, INSERTA LA LLAVE DE LAS VENTAS QUE
    PARTICIPAN EN ESTE CORTE*/
169 CREATE PROCEDURE SP_CORTE_Z(@USER_ID INT)
170 AS
171 BEGIN
172     DECLARE @CORTE_Z_ID INT
173     INSERT INTO CORTE_Z(DATE,USER_ID) VALUES(GETDATE(),@USER_ID);
174     SELECT @CORTE_Z_ID=SCOPE_IDENTITY();
175     INSERT INTO CORTE_Z_PARTS(CORTE_Z_ID,SALE_ID) SELECT @CORTE_Z_ID,ID FROM
        SALES WHERE NOT EXISTS
176         (SELECT ID FROM CORTE_Z_PARTS CZ WHERE CZ.SALE_ID=SALES.ID)
177     SELECT @CORTE_Z_ID
178 END;
179 GO
180 /*RETORNA UNA VISTA DE LAS VENTAS QUE AUN NO ESTAN AGRUPADAS EN ALGUN CORTE DE
    CAJA*/
181 CREATE VIEW VIEW_GETCORTEZ_SALES
182 AS
183 select SALES.ID,SALES.DATE as 'Fecha',SALES.TOTAL,
184 SALES.CHANGE,USERS.NAME as 'User',CUSTOMERS.NAME as 'Customers'
185 From sales
186 inner join USERS ON USERS.ID=SALES.USER_ID
187 inner join CUSTOMERS ON CUSTOMERS.ID=SALES.CUSTOMER_ID
188 WHERE NOT EXISTS
189 (SELECT ID FROM CORTE_Z_PARTS CZ WHERE CZ.SALE_ID=SALES.ID)
190 GO
191 /*RETORNA UN LISTADO DE LOS PRODUCTOS DE LAS VENTAS QUE AUN NO ESTAN AGRUPADAS
    EN ALGUN CORTE DE CAJA*/
192 CREATE VIEW VIEW_GETCORTEZ_PRODUCTS
193 AS
194 select SALE_ID,NAME,QUANTITY,SALE_PARTS.PRICE,
195 TOTAL/1.16 AS 'IMPUESTO',
196 TOTAL-TOTAL/1.16 AS 'IMPORTE',
197 TOTAL
198 From SALE_PARTS
199 INNER JOIN PRODUCTS ON PRODUCTS.ID=SALE_PARTS.PRODUCT_ID

```

```

200 WHERE NOT EXISTS
201 (SELECT ID FROM CORTE_Z_PARTS CZ WHERE CZ.SALE_ID=SALE_PARTS.SALE_ID)
202 GO
203 /*RETORNA UN LISTADO DE TODOS LOS PAGOS QUE SE HICIERON EN LAS VENTAS QUE AUN
    NO ESTAN AGRUPADAS EN ALGUN CORTE DE CAJA*/
204 CREATE VIEW VIEW_PAYMENTS_CORTE_Z
205 AS
206 select SALE_ID,NAME,AMOUNT
207 From SALE_PAYS
208 INNER JOIN PAYMENTS ON PAYMENTS.ID=SALE_PAYS.PAYMENT_ID
209 INNER JOIN PAY_WAYS ON PAY_WAYS.ID=PAYMENTS.PAY_WAY_ID
210 WHERE NOT EXISTS
211 (SELECT ID FROM CORTE_Z_PARTS CZ WHERE CZ.SALE_ID=SALE_PAYS.SALE_ID)
212 GO
213 /*INSERTA LA CABECERA DE UNA VENTA NUEVA Y RETORNA SU ID*/
214 CREATE PROCEDURE SP_SAVE_SALE (@CUSTOMER_ID INT ,@USER_ID INT ,@TOTAL FLOAT ,
    @CHANGE FLOAT)
215 AS
216 BEGIN
217 INSERT INTO SALES(CUSTOMER_ID,USER_ID,DATE,TOTAL,CHANGE) VALUES (@CUSTOMER_ID,
    @USER_ID,GETDATE(),@TOTAL,@CHANGE);
218 SELECT SCOPE_IDENTITY();
219 END
220 GO
221 /*INSERTA UNA PARTIDA DE UNA VENTA NUEVA Y RETORNA SU ID*/
222 CREATE PROCEDURE SP_SAVE_SALE_PART
223 (@SALE_ID INT ,@PRODUCT_ID INT ,@PRICE FLOAT ,@QUANTITY FLOAT ,@TOTAL FLOAT)
224 AS
225 BEGIN
226 --UPDATE PRODUCTS SET STOCK-=@QUANTITY WHERE ID=@PRODUCT_ID
227 INSERT INTO SALE_PARTS
228 (SALE_ID,PRODUCT_ID,QUANTITY,PRICE,TOTAL)
229 VALUES (@SALE_ID,@PRODUCT_ID,@QUANTITY,@PRICE,@TOTAL);
230 SELECT SCOPE_IDENTITY();
231 END
232 GO
233 /*INSERTA LOS MÉTODOS DE PAGO DE UNA VENTA NUEVA*/
234 CREATE PROCEDURE SP_SAVE_SALE_PAYS(@SALE_ID INT ,@PAYMENT_ID FLOAT)
235 AS
236 BEGIN
237 INSERT INTO SALE_PAYS(SALE_ID,PAYMENT_ID) VALUES (@SALE_ID,@PAYMENT_ID);
238 END
239 GO
240 /*INSERTA LA CANTIDAD EN DINERO DE UN MÉTODO DE PAGO DE UNA VENTA NUEVA*/
241 CREATE PROCEDURE SP_SAVE_PAYMENT(@PAY_WAY_ID INT ,@AMOUNT FLOAT)
242 AS
243 BEGIN
244 INSERT INTO PAYMENTS(PAY_WAY_ID,AMOUNT) VALUES (@PAY_WAY_ID,@AMOUNT);
245 SELECT SCOPE_IDENTITY();
246 END
247 GO
248 --INSERTA LOS TIPOS DE PAGO POR DEFECTO
249 INSERT INTO PAY_WAYS (NAME) VALUES('Cash');
250 INSERT INTO PAY_WAYS (NAME) VALUES('Vale');
251 INSERT INTO PAY_WAYS (NAME) VALUES('Card');
252 --Loguea al usuario pidiendo como parametros Nickname y Password y
    desencriptando a su vez la contraseña para validarla

```

```

253 GO
254 CREATE PROCEDURE SP_LOGIN (@NICKNAME VARCHAR(100), @PASSWORD NVARCHAR(50))
255 AS
256 BEGIN
257     DECLARE @USER_ID INT;
258     SELECT @USER_ID=ID FROM USERS WHERE NICKNAME=@NICKNAME
259     IF(@USER_ID>0)
260     BEGIN
261         DECLARE @PASSWORD_KEY VARCHAR(37);
262         DECLARE @REAL_PASSWORD VARCHAR(50);
263         DECLARE @KEY_ID INT;
264         SELECT @KEY_ID=KEYS_ID FROM USERS WHERE ID=@USER_ID
265
266         SELECT @PASSWORD_KEY=PASSWORD_KEY FROM KEYS WHERE ID=@KEY_ID
267
268         SELECT @REAL_PASSWORD=CONVERT(NVARCHAR(MAX), DECRYPTBYPASSPHRASE
                (@PASSWORD_KEY, PASSWORD)) FROM USERS WHERE ID=@USER_ID
269         IF (@REAL_PASSWORD=@PASSWORD)
270         BEGIN
271             SELECT 1 AS 'LOGIN_STATUS';
272             RETURN;
273         END
274     END
275     SELECT 0 AS 'LOGIN_STATUS';
276 END;
277 GO
278 --Registra y/o Actualiza al usuario pidiendo como parametros nickname,
279     contraseña,Nobre y una imagen.
280 --- Encripta la contraseña por seguridad del ususario
281 CREATE PROCEDURE SP_REGISTER(@NICKNAME VARCHAR(100), @PASSWORD NVARCHAR(50),
282     @NAME VARCHAR(100),@PICTURE VARCHAR(MAX),@ENABLED BIT,@ID INT)
283 AS
284 BEGIN
285     IF(@ID<=0)
286     BEGIN
287         DECLARE @PERMISSIONS_ID INT;
288         DECLARE @KEY_ID INT;
289         DECLARE @KEY VARCHAR(MAX);
290         DECLARE @ENCRYPTED_PASSWORD VARBINARY(MAX);
291
292         INSERT INTO PERMISSIONS DEFAULT VALUES
293         SELECT @PERMISSIONS_ID=SCOPE_IDENTITY();
294
295         INSERT INTO KEYS DEFAULT VALUES
296         SELECT @KEY_ID=SCOPE_IDENTITY();
297         SELECT @KEY =PASSWORD_KEY FROM KEYS WHERE ID=@KEY_ID
298
299         SELECT @ENCRYPTED_PASSWORD = ENCRYPTBYPASSPHRASE(@KEY, @PASSWORD);
300
301         INSERT INTO USERS(PERMISSIONS_ID,KEYS_ID,NICKNAME,NAME,PASSWORD,PICTURE,
302             ENABLED)
303         VALUES(@PERMISSIONS_ID,@KEY_ID,@NICKNAME,@NAME,@ENCRYPTED_PASSWORD,
304             @PICTURE,@ENABLED)
305     END ELSE
306     BEGIN
307         SELECT @KEY_ID=KEYS_ID FROM USERS WHERE ID=@ID

```

```

305     SELECT @KEY =PASSWORD_KEY FROM KEYS WHERE ID=@KEY_ID
306     SELECT @ENCRYPTED_PASSWORD = ENCRYPTBYPASSPHRASE(@KEY, @PASSWORD);
307     UPDATE USERS SET NICKNAME=@NICKNAME,NAME=@NAME,PASSWORD=
        @ENCRYPTED_PASSWORD,PICTURE=@PICTURE,ENABLED=@ENABLED
308     WHERE ID=@ID
309
310     END
311 END;
312 GO
313
314 -----PRODUCTS-----
315 GO
316 --Obtiene la existencia de un producto mendante su Id
317 CREATE PROCEDURE SP_GETSTOCK(@ID INT)
318 AS
319 BEGIN
320     SELECT STOCK FROM PRODUCTS WHERE ID=@ID
321 END;
322 GO
323 ---Obtiene los productos donde DISABLED sea igual a cero y ordnandolos por su
        nombre
324 CREATE PROCEDURE SP_GET_PRODUCTS
325 AS
326 BEGIN
327     SELECT * FROM PRODUCTS WHERE DISABLED=0 ORDER BY NAME
328 END;
329 GO
330 ---Obtiene los productos por su codigo, donde DISABLED sea igual a cero y
        ordnandolos por su nombre
331 CREATE PROCEDURE SP_FIND_PRODUCT(@CODE VARCHAR(100))
332 AS
333 BEGIN
334     SELECT * FROM PRODUCTS WHERE DISABLED=0 AND CODE=@CODE ORDER BY NAME
335 END;
336 GO
337 ---Obtiene todos los productos de la tabla Products
338 CREATE PROCEDURE SP_GET_PRODUCT_BY_ID (@ID INT)
339 AS
340 BEGIN
341     SELECT *FROM PRODUCTS WHERE ID=@ID
342 END
343 GO
344 ---Procedure para la busqueda de un producto donde el patrón del codigo o del
        nombre coincida
345 CREATE PROCEDURE SP_SEARCH_PRODUCT(@SEARCH VARCHAR(200))
346 AS
347 BEGIN
348     SELECT * FROM PRODUCTS WHERE DISABLED=0 AND CODE LIKE '%' + @SEARCH + '%' OR
        NAME LIKE '%' + @SEARCH + '%' ORDER BY NAME
349 END;
350 GO
351 ---Obtiene los productos de una categoria
352 CREATE PROCEDURE SP_GET_PRODUCTS_BY_CATEGORY(@CATEGORY_ID INT)
353 AS
354 BEGIN
355     SELECT * FROM PRODUCTS WHERE DISABLED=0 AND CATEGORY_ID=@CATEGORY_ID ORDER BY
        NAME

```

```

356 END;
357 GO
358 ---Obtiene los productos por la categoria buscada
359 CREATE PROCEDURE SP_GET_PRODUCTS_BY_CATEGORY_FIND(@CATEGORY_ID INT,@SEARCH
    VARCHAR(MAX))
360 AS
361 BEGIN
362 SELECT * FROM PRODUCTS WHERE DISABLED=0 AND CATEGORY_ID=@CATEGORY_ID AND NAME
    LIKE '%' + @SEARCH + '%' ORDER BY NAME
363 END;
364 GO
365 ---Obtiene las categorias y las ordena por su descripcion
366 CREATE PROCEDURE SP_GET_CATEGORIES
367 AS
368 BEGIN
369     SELECT *FROM CATEGORIES order by DESCRIPTION
370 END;
371 GO
372 ---Busca las categorias donde el patrón de la descripción categoria o de la
    descripción del producto coincida ordenandolos por su descripción
373 CREATE PROCEDURE SP_FIND_CATEGORIES (@SEARCH VARCHAR(MAX))
374 AS
375 BEGIN
376     SELECT DISTINCT CATEGORIES.ID,CATEGORIES.DESCRPTION FROM CATEGORIES
377     INNER JOIN PRODUCTS ON PRODUCTS.CATEGORY_ID=CATEGORIES.ID
378     WHERE CATEGORIES.DESCRPTION LIKE '%' + @SEARCH + '%' OR PRODUCTS.DESCRPTION
        LIKE '%' + @SEARCH + '%' order by CATEGORIES.DESCRPTION,CATEGORIES.ID
379 END;
380 GO
381 ---Obtiene los categoria por la descripción
382 CREATE PROCEDURE SP_GET_CATEGORY_BY_NAME (@NAME VARCHAR(100))
383 AS
384 BEGIN
385     SELECT *FROM CATEGORIES WHERE DESCRIPTION=@NAME
386 END;
387 GO
388 ---Obtiene los categoria por el Id
389 CREATE PROCEDURE SP_GET_CATEGORY_BY_ID (@ID INT)
390 AS
391 BEGIN
392     SELECT *FROM CATEGORIES WHERE ID=@ID
393 END;
394 GO
395 ---Obtiene todos los proveedores de la tabla SUPPLIERS
396 CREATE PROCEDURE SP_GET_SUPPLIERS
397 AS
398 BEGIN
399     SELECT *FROM SUPPLIERS
400 END;
401 GO
402 ---Registra y actualizacion de productos con los paramentros de id en caso de
    actulizar,codigo,nombre,etc...
403 CREATE PROCEDURE SP_ABC_PRODUCT (@ID INT,@CODE VARCHAR(100),@NAME VARCHAR(100)
    ,@DESCRIPTION VARCHAR(MAX),
404 @CATEGORY_ID INT,@SUPPLIER_ID INT,@UNIT VARCHAR(100),@IMAGE VARCHAR(MAX),
    @STOCK REAL,@MINIMUM REAL,@MAXIMUM REAL,
405 @PRICE REAL,@DISABLED BIT,@DISCCOUNT REAL)

```

```

406 AS
407 BEGIN
408     IF (@ID <= 0)
409     BEGIN
410         INSERT INTO PRODUCTS (CODE, NAME, DESCRIPTION, CATEGORY_ID, SUPPLIER_ID,
411             UNIT, IMAGE, STOCK, MINIMUM, MAXIMUM, PRICE, DISABLED, DISCCOUNT)
412         VALUES (@CODE, @NAME, @DESCRIPTION, @CATEGORY_ID, @SUPPLIER_ID, @UNIT, @IMAGE
413             , @STOCK, @MINIMUM, @MAXIMUM, @PRICE, @DISABLED, @DISCCOUNT)
414     END ELSE
415     BEGIN
416         UPDATE PRODUCTS SET CODE=@CODE, NAME=@NAME, DESCRIPTION=@DESCRIPTION,
417             CATEGORY_ID=@CATEGORY_ID,
418             SUPPLIER_ID=@SUPPLIER_ID, UNIT=@UNIT, IMAGE=@IMAGE, STOCK=@STOCK, MINIMUM=
419             @MINIMUM, MAXIMUM=@MAXIMUM, PRICE=@PRICE, DISABLED=@DISABLED, DISCCOUNT
420             =@DISCCOUNT
421         WHERE ID=@ID
422     END
423 END;
424 GO
425 ---Registra y actualizacion de categorias con los paramentros de id en caso
426 de actualizar, descripción e imagen
427 CREATE PROCEDURE SP_ABC_CATEGORY (@ID INT, @DESCRIPTION VARCHAR(100), @PICTURE
428     VARCHAR(MAX))
429 AS
430 BEGIN
431     IF (@ID <= 0)
432     BEGIN
433         INSERT INTO CATEGORIES (DESCRIPTION, PICTURE) VALUES (@DESCRIPTION,
434             @PICTURE)
435     END ELSE
436     BEGIN
437         UPDATE CATEGORIES SET DESCRIPTION=@DESCRIPTION, PICTURE=@PICTURE WHERE
438             ID=@ID
439     END
440 END;
441 GO
442 -----ALERTS-----
443 CREATE VIEW VIEW_OVERSTOCKED
444 AS
445 SELECT * FROM PRODUCTS WHERE STOCK >= MAXIMUM;
446 GO
447 CREATE VIEW VIEW_UNDERSTOCKED
448 AS
449 SELECT * FROM PRODUCTS WHERE STOCK <= MINIMUM;
450 GO
451 CREATE VIEW VIEW_ALL_ALERTS
452 AS
453 SELECT * FROM PRODUCTS WHERE STOCK >= MAXIMUM OR STOCK <= MINIMUM;
454 GO
455 -----MOVEMENTS-----
456 GO
457 ---Obtiene todos las compras de la tabla BUYS mediante su Id
458 CREATE PROCEDURE SP_GET_BUY_BY_ID (@ID INT)
459 AS
460 BEGIN
461     SELECT * FROM BUYS WHERE ID = @ID;
462 END;

```



```

454 GO
455 ---Obtiene todos los movimientos de la tabla MOVEMENTS
456 CREATE PROCEDURE SP_GET_MOVEMENT(@ID INT)
457 AS
458 BEGIN
459 SELECT * FROM MOVEMENTS WHERE ID =@ID;
460 END;
461 GO
462 ---Obtiene todas las partes de los movimientos de la tabla MOVEMENT_PARTS
463 CREATE PROCEDURE SP_GET_MOVEMENTPART_BY_ID(@ID INT)
464 AS
465 BEGIN
466 SELECT * FROM MOVEMENT_PARTS WHERE ID=@ID;
467 END;
468 GO
469 ---Obtiene todas las partes de las compras de la tabla BUY_PARTS
470 CREATE PROCEDURE SP_GET_BUY_PART_BY_ID(@ID INT)
471 AS
472 BEGIN
473 SELECT * FROM BUY_PARTS WHERE ID=@ID;
474 END;
475 GO
476 ---Obtiene todas las partes de las compras de la tabla BUY_PARTS ?????????
    Esta repetido?
477 CREATE PROCEDURE SP_GET_BUY_PARTS_BY_ID(@ID INT)
478 AS
479 BEGIN
480 SELECT * FROM BUY_PARTS WHERE BUY_ID=@ID;
481 END;
482 GO
483 ---Obtiene todas las partes de los movimientos de la tabla MOVEMENT_PARTS DADO
    UN ID DE MOVIMIENTO
484 CREATE PROCEDURE SP_GET_MOVEMENTPART(@ID INT)
485 AS
486 BEGIN
487 SELECT * FROM MOVEMENT_PARTS WHERE MOVEMENT_ID=@ID;
488 END;
489 GO
490 ---Obtiene todos los conceptos de la tabla MOVEMENT_CONCEPT
491 CREATE PROCEDURE SP_GET_MOVEMENTCONCEPT(@ID INT)
492 AS
493 BEGIN
494 SELECT * FROM MOVEMENT_CONCEPT WHERE ID =@ID;
495 END;
496 GO
497 ---Registra las compras con los paramentros de id del usuario,fecha y
    descripción
498 CREATE PROCEDURE SP_ADD_BUY(
499 @USER_ID INT,
500 @DATE DATETIME,
501 @DESCRIPTION VARCHAR(100))
502 AS
503 BEGIN
504     INSERT INTO BUYS (USER_ID,DATE,DESCRIPTION)
505     VALUES(@USER_ID,@DATE,@DESCRIPTION);
506     SELECT SCOPE_IDENTITY();
507 END;

```

```

508 GO
509 ---Registra las partes de las compras con los paramentros de id del movimeinto
    ,id del producto y cantidad
510 --ademas de que ajusta la existencia de los productos que s even afectados por
    los movimientos
511 CREATE PROCEDURE SP_ADD_BUY_PARTS(
512 @BUY_ID INT,
513 @PRODUCT_ID INT,
514 @QUANTITY REAL)
515 AS
516 BEGIN
517     DECLARE @INITIALLY_STOCK REAL
518     DECLARE @NEW_STOCK REAL
519     SELECT @INITIALLY_STOCK = STOCK FROM PRODUCTS WHERE ID=@PRODUCT_ID
520     SELECT @NEW_STOCK=@INITIALLY_STOCK+@QUANTITY
521     INSERT INTO BUY_PARTS (BUY_ID,PRODUCT_ID,QUANTITY,INITIALLY_STOCK,
        NEW_STOCK)
522     VALUES(@BUY_ID,@PRODUCT_ID,@QUANTITY,@INITIALLY_STOCK,@NEW_STOCK);
523     SELECT SCOPE_IDENTITY();
524 END;
525 GO
526 ---Registra los movimientos con los paramentros de id del usuario,id del
    concepto,tipo,fecha y descripción
527 CREATE PROCEDURE SP_ADD_MOVEMENT(
528 @USER_ID INT,
529 @MOVEMENT_CONCEPT_ID INT,
530 @TYPE VARCHAR(1),
531 @DATE_M DATETIME,
532 @DESCRIPTION VARCHAR(100))
533 AS
534 BEGIN
535     INSERT INTO MOVEMENTS (USER_ID,MOVEMENT_CONCEPT_ID,TYPE,DATE,DESCRIPTION)
536     VALUES(@USER_ID,@MOVEMENT_CONCEPT_ID,@TYPE,@DATE_M,@DESCRIPTION);
537     SELECT SCOPE_IDENTITY();
538 END;
539 GO
540 ---Registra las partes de los movimientos con los paramentros de id del
    movimeinto,id del producto y cantidad
541 --ademas de que ajusta la existencia de los productos que s even afectados por
    los movimientos
542 CREATE PROCEDURE SP_ADD_MOVENT_PARTS(
543 @MOVEMENT_ID INT,
544 @PRODUCT_ID INT,
545 @QUANTITY REAL,
546 @TYPE VARCHAR(1))
547 AS
548 BEGIN
549     DECLARE @INITIALLY_STOCK REAL
550     DECLARE @NEW_STOCK REAL
551     SELECT @INITIALLY_STOCK = STOCK FROM PRODUCTS WHERE ID=@PRODUCT_ID
552     IF(@TYPE='S')
553     BEGIN
554         SELECT @QUANTITY=@QUANTITY*-1
555     END
556     SELECT @NEW_STOCK=@INITIALLY_STOCK+@QUANTITY
557     UPDATE PRODUCTS SET STOCK=@NEW_STOCK WHERE ID=@PRODUCT_ID
558     INSERT INTO MOVEMENT_PARTS (MOVEMENT_ID,PRODUCT_ID,QUANTITY,

```

```

INITIALLY_STOCK,NEW_STOCK)
559     VALUES (@MOVEMENT_ID,@PRODUCT_ID,@QUANTITY,@INITIALLY_STOCK,@NEW_STOCK);
560     SELECT SCOPE_IDENTITY();
561 END;
562 GO
563 ---Registra los conceptos de movimientos con los paramentros de tag,
    descripción y tipo
564 CREATE PROCEDURE SP_ADD_MOVEMENTCONCEPT(
565     @TAG VARCHAR(10),
566     @DESCRIPTION VARCHAR(100),
567     @TYPE VARCHAR(1))
568 AS
569 BEGIN
570     INSERT INTO MOVEMENT_CONCEPT (TAG,DESCRIPTION,TYPE)
571     VALUES (@TAG,@DESCRIPTION,@TYPE);
572 END;
573 GO
574 ---Obtiene todos las compras de la tabla BUYS
575 CREATE VIEW VIEW_GET_ALL_BUYS
576 AS
577 SELECT * FROM BUYS ;
578 GO
579 ---Obtiene todos los movimientos de la tabla MOVEMENTS
580 CREATE VIEW VIEW_GETALLMOVEMENTS
581 AS
582 SELECT * FROM MOVEMENTS ;
583 GO
584 ---Obtiene todos los conceptos de movimientos de la tabla MOVEMENT_CONCEPT
585 CREATE VIEW VIEW_GETALLMOVEMENTCONCEPT
586 AS
587 SELECT * FROM MOVEMENT_CONCEPT ;
588 GO
589 GO
590 ---Obtiene todos las partes de los movimientos de la tabla MOVEMENT_PARTS
591 CREATE VIEW VIEW_GETALLMOVEMENTPART
592 AS
593 SELECT * FROM MOVEMENT_PARTS ;
594 -----CUSTOMER-----
595 GO
596 ---Registra y actualizacion de cateogorias con los paramentros de id en caso
    de actulizar,notas e imagen ademas de enabled para eliminar al cliente
597 CREATE PROCEDURE SP_ABC_CUSTOMER (@ID INT, @NAME VARCHAR(100),@NOTES VARCHAR(
    MAX),@PICTURE VARCHAR(MAX),@ENABLED BIT)
598 AS
599 BEGIN
600     IF (@ID <= 0)
601     BEGIN
602         INSERT INTO CUSTOMERS (NAME,NOTES,PICTURE,ENABLED ) VALUES (@NAME,@NOTES
            ,@PICTURE,@ENABLED )
603     END ELSE
604     BEGIN
605         UPDATE CUSTOMERS SET NAME= @NAME,NOTES=@NOTES,PICTURE=@PICTURE,ENABLED
            =@ENABLED WHERE ID=@ID
606     END
607 END;
608 GO
609 ---Obtiene los clientes de la tabla CUSTOMERS mediante su id

```

```

610 CREATE PROCEDURE SP_SEARCHCUSTOMER(@ID INT)
611 AS
612 BEGIN
613 SELECT * FROM CUSTOMERS WHERE ID= @ID;
614 END;
615
616 GO
617 ---Obtiene todos los clientes de la tabla CUSTOMERS que no esten eliminados
618 CREATE VIEW VIEW_GETALL
619 AS
620 SELECT * FROM CUSTOMERS WHERE ENABLED= 0;
621
622
623 -----Supplier-----
624 GO
625 -----Registra y actualizacion de proveedores con los paramentros de id en
        caso de actulizar,nombre,telefono,etc...
626 CREATE PROCEDURE SP_ABC_SUPPLIERS(@ID INT,@NAME VARCHAR(100),@CELLPHONE
        VARCHAR(100),@NOTES VARCHAR(MAX),@PICTURE VARCHAR(MAX),@ENABLED BIT)
627 AS
628 BEGIN
629     IF(@ID<=0)
630     BEGIN
631         INSERT INTO SUPPLIERS(NAME,CELLPHONE,NOTES,PICTURE,ENABLED) VALUES (
            @NAME,@CELLPHONE,@NOTES,@PICTURE,@ENABLED)
632     END ELSE
633     BEGIN
634         UPDATE SUPPLIERS SET NAME=@NAME,CELLPHONE=@CELLPHONE,NOTES=@NOTES,
            PICTURE=@PICTURE,ENABLED=@ENABLED WHERE ID=@ID;
635     END
636 END;
637 GO
638 ---Obtiene a todos los proveedores de la tabla SUPPLIERS
639 CREATE VIEW VIEW_GETALLSUPPLIERS
640 AS
641 SELECT * FROM SUPPLIERS WHERE ENABLED=0;
642 GO
643 ---Obtiene a los proveedores de la tabla SUPPLIERS mediante su Id
644 CREATE PROCEDURE SP_GETIDSUPPLIERS(@ID INT)
645 AS
646 SELECT * FROM SUPPLIERS WHERE ID=@ID;
647 GO
648 /*INSERTA O ACTUALIZA UN DETERMINADO REGISTRO DE PERMISOS GUIANDOSE POR SU ID
        */
649 CREATE PROCEDURE SP_ABC_PERMISSIONS
650 ( @ID INT,@STOCK_OUT BIT,@STOCK_IN BIT ,@REPORTS_READ BIT,@USER_MANAGER BIT,
        @READ_ONLY BIT,@SALES BIT)
651 AS
652 BEGIN
653 IF(@ID<=0)
654 BEGIN
655     INSERT INTO PERMISSIONS (STOCK_OUT,STOCK_IN,REPORTS_READ,USER_MANAGER,
        READ_ONLY,SALES)VALUES(@STOCK_OUT,@STOCK_IN ,@REPORTS_READ ,
        @USER_MANAGER,
656     @READ_ONLY,@SALES)
657 END ELSE
658 BEGIN

```

```

659         UPDATE PERMISSIONS SET STOCK_OUT=@STOCK_OUT,STOCK_IN=@STOCK_IN,
            REPORTS_READ=@REPORTS_READ,USER_MANAGER=@USER_MANAGER,READ_ONLY=
            @READ_ONLY,SALES=@SALES
660     WHERE ID=@ID
661 END
662 END;
663 GO
664 /*RECUPERA UN REGISTRO DE PERMISOS POR SU ID*/
665 CREATE PROCEDURE GET_PERMISSIONS_BY_ID (@ID INT)
666 AS
667 BEGIN
668     SELECT *FROM PERMISSIONS WHERE ID=@ID
669 END;
670 ----USERS-----
671 GO
672 /*OBTIENE LA INFORMACIÓN UN USUARIO POR SU NICKNAME*/
673 CREATE PROCEDURE GET_USER_BY_NICKNAME (@NICKNAME VARCHAR(100))
674 AS
675 BEGIN
676
677     DECLARE @USER_ID INT;
678     SELECT @USER_ID=ID FROM USERS WHERE NICKNAME=@NICKNAME
679     DECLARE @PASSWORD_KEY VARCHAR(37);
680     DECLARE @REAL_PASSWORD VARCHAR(50);
681     DECLARE @KEY_ID INT;
682     SELECT @KEY_ID=KEYS_ID FROM USERS WHERE ID=@USER_ID
683     SELECT @PASSWORD_KEY=PASSWORD_KEY FROM KEYS WHERE ID=@KEY_ID
684     SELECT @REAL_PASSWORD=CONVERT(NVARCHAR(MAX), DECRYPTBYPASSPHRASE(
        @PASSWORD_KEY, PASSWORD)) FROM USERS WHERE ID=@USER_ID
685     SELECT ID,PERMISSIONS_ID,KEYS_ID,NICKNAME,NAME,@REAL_PASSWORD,PICTURE,ENABLED
686     FROM USERS WHERE ID=@USER_ID
687 END;
688 GO
689 /*OBTIENE LA INFORMACIÓN UN USUARIO POR SU ID*/
690 CREATE PROCEDURE GET_USER_BY_ID (@ID INT)
691 AS
692 BEGIN
693     DECLARE @PASSWORD_KEY VARCHAR(37);
694     DECLARE @REAL_PASSWORD VARCHAR(50);
695     DECLARE @KEY_ID INT;
696     SELECT @KEY_ID=KEYS_ID FROM USERS WHERE ID=@ID
697     SELECT @PASSWORD_KEY=PASSWORD_KEY FROM KEYS WHERE ID=@KEY_ID
698     SELECT @REAL_PASSWORD=CONVERT(NVARCHAR(MAX), DECRYPTBYPASSPHRASE(
        @PASSWORD_KEY, PASSWORD)) FROM USERS WHERE ID=@ID
699     SELECT ID,PERMISSIONS_ID,KEYS_ID,NICKNAME,NAME,@REAL_PASSWORD,PICTURE,ENABLED
700     FROM USERS WHERE ID=@ID
701 END;
702 GO
703 GO
704 /*RETORNA TODOS LOS USUARIOS QUE NO HAN SIDO ELIMINADOS*/
705 CREATE VIEW VIEW_GETALLUSERS
706 AS
707 SELECT * FROM USERS WHERE ENABLED=0;
708 GO
709 /*RETORNA LA INFORMACIÓN DE NOMBRE Y STOCK DE TODOS LOS PRODUCTOS*/
710 CREATE VIEW VIEW_GET_STOCK
711 AS

```

```

712 SELECT
713 code,NAME,stock,MINIMUM,MAXIMUM
714 FROM products where DISABLED=0
715 GO
716 /*RETORNA UN REPORTE DE VENTAS ENTRE UN RANGO DE FECHAS O BIEN TODOS LOS
    REGISTROS*/
717 CREATE PROCEDURE SALES_REPORT(@BEGIN DATETIME,@END DATETIME,@ALL BIT)
718 AS
719 BEGIN
720 select SALES.ID,SALES.DATE as 'Fecha',SALES.TOTAL,SALES.CHANGE,USERS.NAME as '
    User',CUSTOMERS.NAME as 'Customers'
721 From sales
722 inner join USERS ON USERS.ID=SALES.USER_ID
723 inner join CUSTOMERS ON CUSTOMERS.ID=SALES.CUSTOMER_ID
724 WHERE @ALL=1 OR (SALES.DATE>=@BEGIN AND SALES.DATE<=@END)
725 END
726 GO
727 /*RETORNA UN REPORTE DE MOVIMIENTOS POR USUARIO ENTRE UN RANGO DE FECHAS O
    BIEN TODOS LOS REGISTROS*/
728 CREATE PROCEDURE SP_MOVENTS_REGISTER (@BEGIN DATETIME,@END DATETIME,@ALL BIT)
729 AS
730 BEGIN
731 SELECT 'Ticket #' +CAST(SALES.ID AS VARCHAR(MAX)) AS ID,USERS.NAME, DATE AS '
    Fecha', 'Venta' AS TYPE FROM SALES
732 INNER JOIN USERS ON USERS.ID=SALES.USER_ID
733 WHERE @ALL=1 OR (SALES.DATE>=@BEGIN AND SALES.DATE<=@END)
734 UNION
735 SELECT 'Movimiento #' +CAST(MOVEMENTS.ID AS VARCHAR(MAX)) as ID,USERS.NAME,
    DATE AS 'Fecha',
736 IIF(TYPE='S','Salida','Entrada') AS TYPE
737 FROM MOVEMENTS
738 INNER JOIN USERS ON USERS.ID=MOVEMENTS.USER_ID
739 WHERE @ALL=1 OR (MOVEMENTS.DATE>=@BEGIN AND MOVEMENTS.DATE<=@END)
740 UNION
741 SELECT 'Compra #' +CAST(BUYS.ID AS VARCHAR(MAX)) as ID,USERS.NAME, DATE AS '
    Fecha', 'Compra' AS TYPE
742 FROM BUYS
743 INNER JOIN USERS ON USERS.ID=BUYS.USER_ID
744 WHERE @ALL=1 OR (BUYS.DATE>=@BEGIN AND BUYS.DATE<=@END)
745 END;

```

---

### 9.6.1 Ejemplo de llamada a procedimiento

```

1  AppData.SQL.EXEC("SP_ABC_PRODUCT", CommandType.StoredProcedure,
2  new SqlParameter("ID", Id),
3  new SqlParameter("CODE", Code),
4  new SqlParameter("NAME", Name),
5  new SqlParameter("DESCRIPTION", Description),
6  new SqlParameter("CATEGORY_ID", Category.Id),
7  new SqlParameter("SUPPLIER_ID", Supplier.Id),
8  new SqlParameter("STOCK", Stock),
9  new SqlParameter("UNIT", Unit),
10 new SqlParameter("IMAGE", Picture),
11 new SqlParameter("MINIMUM", Minimum),
12 new SqlParameter("MAXIMUM", Maximum),
13 new SqlParameter("PRICE", Price),
14 new SqlParameter("DISABLED", Disabled),

```

```
15     new SqlParameter("DISCCOUNT", Disccount)
16 );
17
18 using (IReader reader = AppData.SQL.Read("SP_GET_PRODUCT_BY_ID", CommandType.
19     StoredProcedure,
20     new SqlParameter("ID", Id)))
21 {
22     if (reader.Read())
23     {
24         producto = new Product()
25         {
26             Id = Convert.ToInt32(reader[0]),
27             Code = Convert.ToString(reader[1]),
28             Name = Convert.ToString(reader[2]),
29             Description = Convert.ToString(reader[3]),
30             Category = Category.GetById(Convert.ToInt32(reader[4])),
31             Supplier = Supplier.GetById(Convert.ToInt32(reader[5])),
32             Unit = Convert.ToString(reader[6]),
33             Picture = reader[7].ToString(),
34             Stock = Convert.ToSingle(reader[8]),
35             Minimum = Convert.ToSingle(reader[9]),
36             Maximum = Convert.ToSingle(reader[10]),
37             Price = Convert.ToSingle(reader[11]),
38             Disabled = Convert.ToBoolean(reader[12]),
39             Disccount = Convert.ToInt32(reader[13])
40         };
41     }
42 }
```

---