

Sistema de Gestión de Almacenamiento

Integrantes:

- Sebastian Groselj
- Bryan Pino

Profesor: Armando Muñoz

Asignatura: Arquitectura, AQY 4101 - 008D

Índice

Índice	1
Introducción	3
Presentación del Proyecto	3
Alternativas de solución	4
Objetivo general y específicos	4
Viabilidad del proyecto	5
Toma de requisitos	5
Documentación de Requisitos	6
Requerimientos Funcionales	6
Requerimientos No Funcionales	7
Casos de Uso	7
Escenarios de Calidad	8
Estable	8
Descripción	8
Escenario de pruebas	8
Portable	8
Descripción	8
Escenario de pruebas	8
Eficiente	9
Descripción	9
Escenario de pruebas	9
Arquitectura	10
N Capas	10
SOA	10

Vistas	11
Vistas Lógicas	11
Diagrama de Clases	11
Base de Datos	11
Aplicación	12
Conexión	12
Modelo de Datos	13
Negocio	14
Interfaz de Usuario Gráfica de Escritorio (GUI Escritorio)	19
Vista Desarrollo	24
Diagrama de Paquetes	24
Vista Física	25
Diagrama de Despliegue	25
Vista de Procesos	26
Diagrama de Actividades	26
Nº 1 Autenticar Usuario	26
MOCKUPS	27
Login	27
ERP menu	27
Solicitudes	29
Registros	30
Contabilidad	31
Proveedores	32
Sucursales	33
Bodegas	34
Bodega Menú	35
Enviar Producto	36
Recibir Producto	37
Solicitudes	38
Listar productos	39
Actualizar Productos	39
Patrones y Estilos Arquitectónicos	41

Introducción

La empresa “Master Hardware” quiere actualizar su sistema de inventario de las bodegas para automatizar algunos de sus procesos y tener registros de los cambios. Por ello le solicitaron a “Lite Soft” que desarrolle un proyecto para esta necesidad.

Presentación del Proyecto

La empresa “Master Hardware” requiere automatizar su sistema de inventario de las bodegas.

Los productos del inventario son de hardware de computadoras, laptops, accesorios, consumibles de hardware, software en físico, entre otros.

Internamente los productos requieren que se mantengan registros que faciliten su búsqueda y estado. Estos se contabilizan como unidades de un producto. Pueden moverse a diferentes secciones de la bodega.

La ERP o gestión empresarial requiere de la información de la entrada, salida, cantidad y estado de los productos para realizar la logística. Además se encargan de realizar las solicitudes de envío de los productos.

Los bodegueros pueden trasladar productos, recibir productos entrantes y almacenarlos, realizar inspecciones del estado, realizar envíos a los proveedores y a las tiendas por solicitud del ERP.

Se debe mantener registro de los cambios que se realicen en el almacenamiento, para poder realizar trazabilidad.

Alternativas de solución

- a) Realizar un WMS o SGA (Sistema de Gestión de Almacenes) para aplicación. Que posea una Base de datos en un servidor y una o dos aplicaciones que gestionen y den una interfaz gráfica a los usuarios bodegueros y del ERP.
- b) Realizar un WMS o SGA (Sistema de Gestión de Almacenes) para aplicación. Que posea un servidor que se encargue de la gestión junto a una base de datos, además de una o dos aplicaciones que se conecte al servidor para darle al usuario bodeguero y del ERP una interfaz gráfica.
- c) Realizar un SGA (Sistema de gestión de Almacenes) para una página web. Que posea un servidor, base de datos y un sitio web no indexado.

La solución que eligieron “Master Hardware” y “Lite Soft” es la “opción A”. Este cumple con las especificaciones del proyecto, además de poder implementarse con una arquitectura para una única aplicación con una base de datos central.

Objetivo general y específicos

Objetivos Generales:

- Automatizar el SGA (Sistema de Gestión de Almacenes)

Objetivos Específicos:

- Diferenciar los productos por tipos
- Mantener registro de la posición y estado de los productos
- Se pueda mover los productos y conocer su nueva posición
- Mantener a la ERP informada para realizar la logística
- Permitir a la ERP mandar solicitudes que realizara algún bodeguero
- Los bodegueros puedan trasladar productos
- Los bodegueros puedan recibir productos de los proveedores y tiendas
- Los bodegueros pueden recibir una solicitud del ERP para envío de productos
- Los bodegueros pueden realizar inspecciones a los productos
- Mantener registro de cambios

Viabilidad del proyecto

Realizar un WMS o SGA (Sistema de Gestión de Almacenes) para aplicación.

- Técnica: Requiere de servidor de Base de Datos y Dispositivos individuales o compartidos que requieren que el software este compilado para ese hardware, se resuelve fácilmente con el uso de frameworks. Al ser los dispositivos los que se encarguen de las reglas del negocio, simplificará el desarrollo y despliegue del sistema, al minimizar los programas que deben comunicarse para realizar una operación.
- Legal: Por almacenar y registrar información de productos internos de la empresa, no hay complicaciones legales con terceros. Para la cuenta de los usuarios que pueden usar el sistema, estos son de empleados, solo se requiere de una autorización de su parte y que comprenden la necesidad de este permiso y las consecuencias si otro accede al sistema con sus datos.
- Financiera: Hay un costo inicial para el desarrollo e implementación del proyecto, además del pago de los equipos.
- Económica: Los costos de operación son bajos y solo requieren del pago de licencias y energía eléctrica. Mientras que los gastos de mantención pueden ser mayores según el caso, pero estos no deberían ser muy frecuentes.

Toma de requisitos

1. El sistema debe poder mediante una interfaz gráfica poder gestionar los productos, tipos de productos, lista de tiendas y de proveedores. La gestión es solo agregar, editar, deshabilitar o habilitar y eliminar
2. El sistema debe permitir a un ERP enviar una solicitud de envío de producto, este lo podrá ver un bodeguero para encargarse de enviarlo
3. Cualquier modificación debe tener un registro automático, este debe poseer una descripción junto con quien lo realizó y la hora y fecha
4. El sistema debe mostrar el historial de cambios
5. El sistema despliega la lista de productos. También permite filtrar los productos
6. Los bodegueros podrán gestionar los productos. Aunque para realizar envíos requieren de una solicitud del ERP. Pueden recibir, enviar, trasladar y revisar los productos, lo que es agregar, editar y marcar como enviado un producto

Documentación de Requisitos

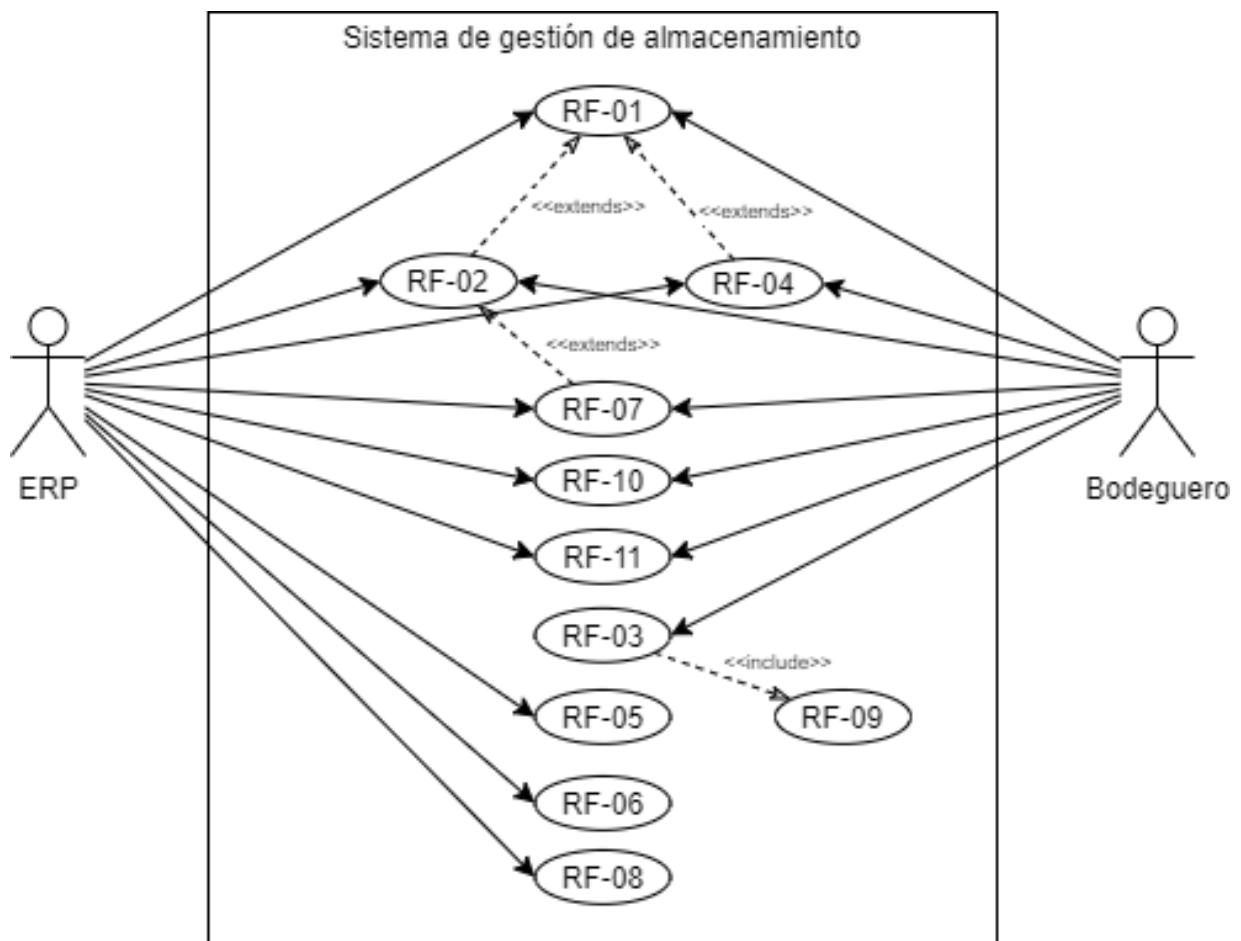
Requerimientos Funcionales

ID	Nombre	Actor	Función
RF-01	Ver Listado de productos	ERP, Bodeguero	Mostrar un listado con todos los productos
RF-02	Filtrar listado de productos	ERP, Bodeguero	Filtrar los productos por categoría
RF-03	Gestionar producto	Bodeguero	Recibir, revisar estado, trasladar y enviar productos
RF-04	Buscar producto	ERP, Bodeguero	Buscar un producto en la lista de productos
RF-05	Enviar Solicitud	ERP	Enviar solicitud para el envío de un producto
RF-06	Ver Contabilidad	ERP	Mostrar funciones para la contabilidad
RF-07	Búsqueda por Stock	ERP, Bodeguero	Mostrará todos los productos con/sin stock y se ordenará por cantidad.
RF-08	Gestionar Mantenedores	ERP	Gestionar la lista de tipo de productos, productos, tiendas y proveedores
RF-09	Registrar Cambios	Sistema	Registrar los cambios de los productos
RF-10	Ver Solicitudes	ERP, Bodeguero	Ver solicitudes de envío de productos, para que el bodeguero los realice o el ERP los cancele
RF-11	Ver Registros	ERP, Bodeguero	Ver el historial con todos los registros de cambios

Requerimientos No Funcionales

ID	Nombre	Descripción
RNF-01	Usabilidad	Interfaz intuitiva, minimalista y amigable
RNF-02	Conexión estable	La conexión del sistema sea estable
RNF-03	Aviso de errores comprensible	Al presentar un error, el sistema muestra un mensaje del error con el motivo y que se requiere para que no ocurra
RNF-04	Interfaz agradable	El sistema utiliza tonos y colores que no cansen la vista

Casos de Uso



Escenarios de Calidad

El sistema requiere ser estable, portable y eficiente.

Estable

Descripción

El sistema funciona correctamente y sin fallos, para todas sus funciones en cualquier situación deben responder de la misma forma.

Escenario de pruebas

Probar las diferentes funcionalidades del sistema con casos de prueba, que registran los resultados de las pruebas, para realizar cambios y correcciones necesarios para que todo esté funcionando correctamente.

Portable

Descripción

El sistema tiene 2 software, el “cliente” y el “servidor”, el cliente funciona en los dispositivos portables y estos permiten conectarse con el software del servidor principal que tiene la base de datos.

Escenario de pruebas

Se debe probar que el “software cliente” esté funcionando en los distintos dispositivos que usarán los usuarios y que estos se conecten al servidor principal para hacer solicitudes a la base de datos. Después se debe probar que la conexión en producción funcione correctamente, por ello se debe realizar pruebas de campo o simular las bodegas para probar que la conexión funcione.

Eficiente

Descripción

El sistema tiene un buen rendimiento del software cliente, donde este no se congela o tarda en realizar acciones. Además el servidor debe poder gestionar muchas conexiones de los clientes.

Escenario de pruebas

Para poner a prueba el software del cliente, hay que utilizarlo durante largos períodos y bajo mucha carga, para ello se deben estar abriendo y cerrando continuamente ventanas, realizar acciones con el servidor, solicitar del servidor muchos datos y en gran cantidad. Mientras se monitorea este programa, vigilando que se esté liberando la memoria del sistema al dejar de usar ventanas o al cerrarse estas, que no se queden ventanas o elementos invisibles hasta que se apague el sistema y que la respuesta se mantenga constante.

Para poner a prueba el software del servidor, hay que realizar muchas conexiones de clientes, la mitad que sean constantes en el tiempo y la otra que estén conectando y desconectando constantemente. Se debe estar vigilando el tiempo de respuesta a las peticiones no varían mucho y el consumo de recursos no se eleve más de lo que pueda soportar el hardware.

Arquitectura

Las arquitecturas elegidas para el proyecto son “N Capas” y “SOA”.

N Capas

N capas permite que el software se divida en diferentes capas que facilitan la realización de funciones al abstraer procesos que son resueltos por capas inferiores. Esto ayuda a tener separado la capa lógica y de diseño, permitir tener un sistema de cliente-servidor, y tener el código más ordenado para realizar cambios y reparaciones.

SOA

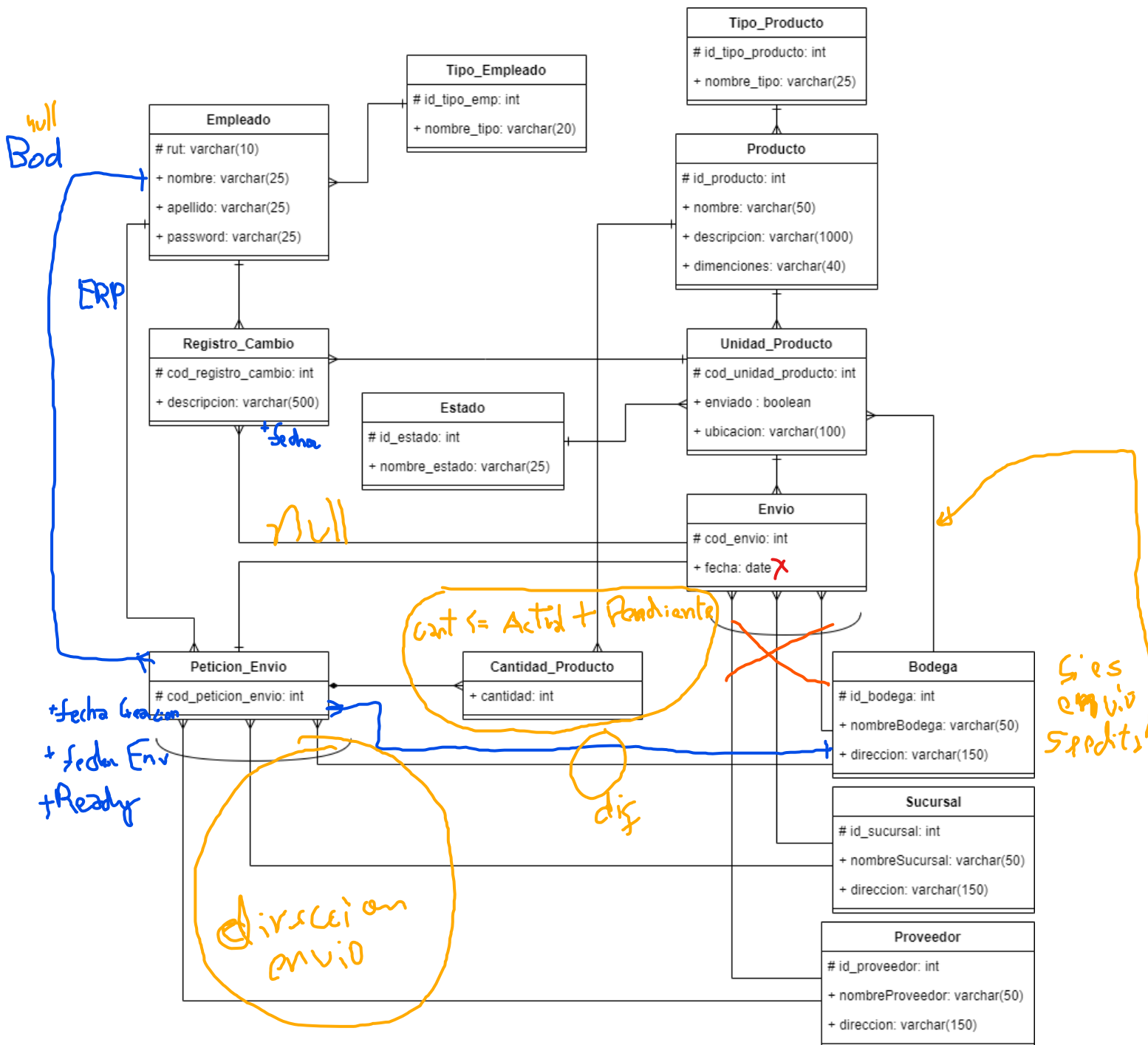
SOA es la “arquitectura orientada a servicios”, esta permite desarrollar módulos con tecnologías independientes al proyecto y poder utilizarlo en este. Esto facilita el desarrollo de los módulos al no necesitar desarrolladores con los conocimientos específicos del lenguaje o herramientas necesarias para el desarrollo del proyecto principal, también permite la utilización de servicios para contratar de módulos ya creados por terceros, o compartir reutilizando los módulos creados con futuros programas necesarios para la empresa.

Vistas

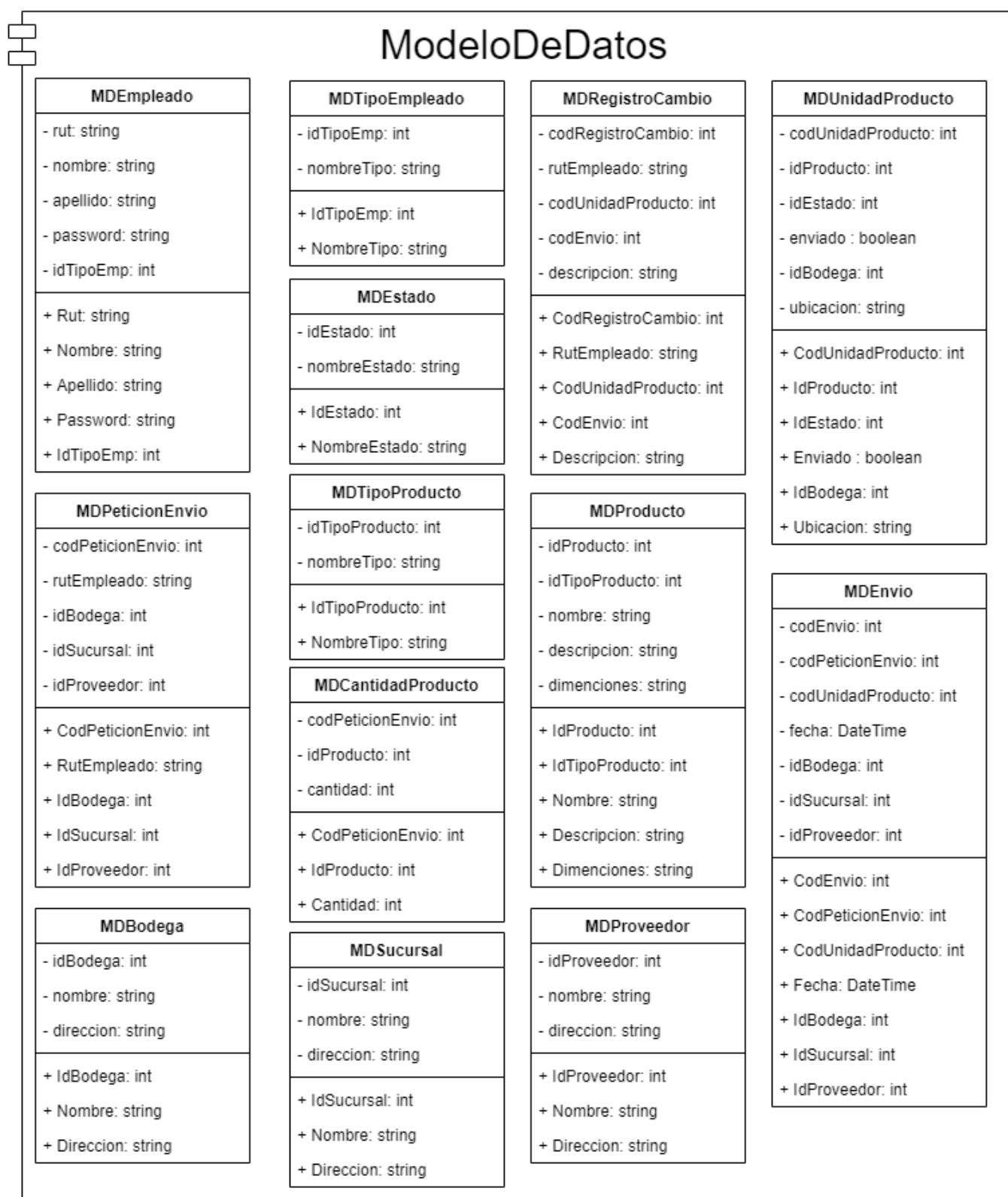
Vistas Lógicas

Diagrama de Clases

Base de Datos

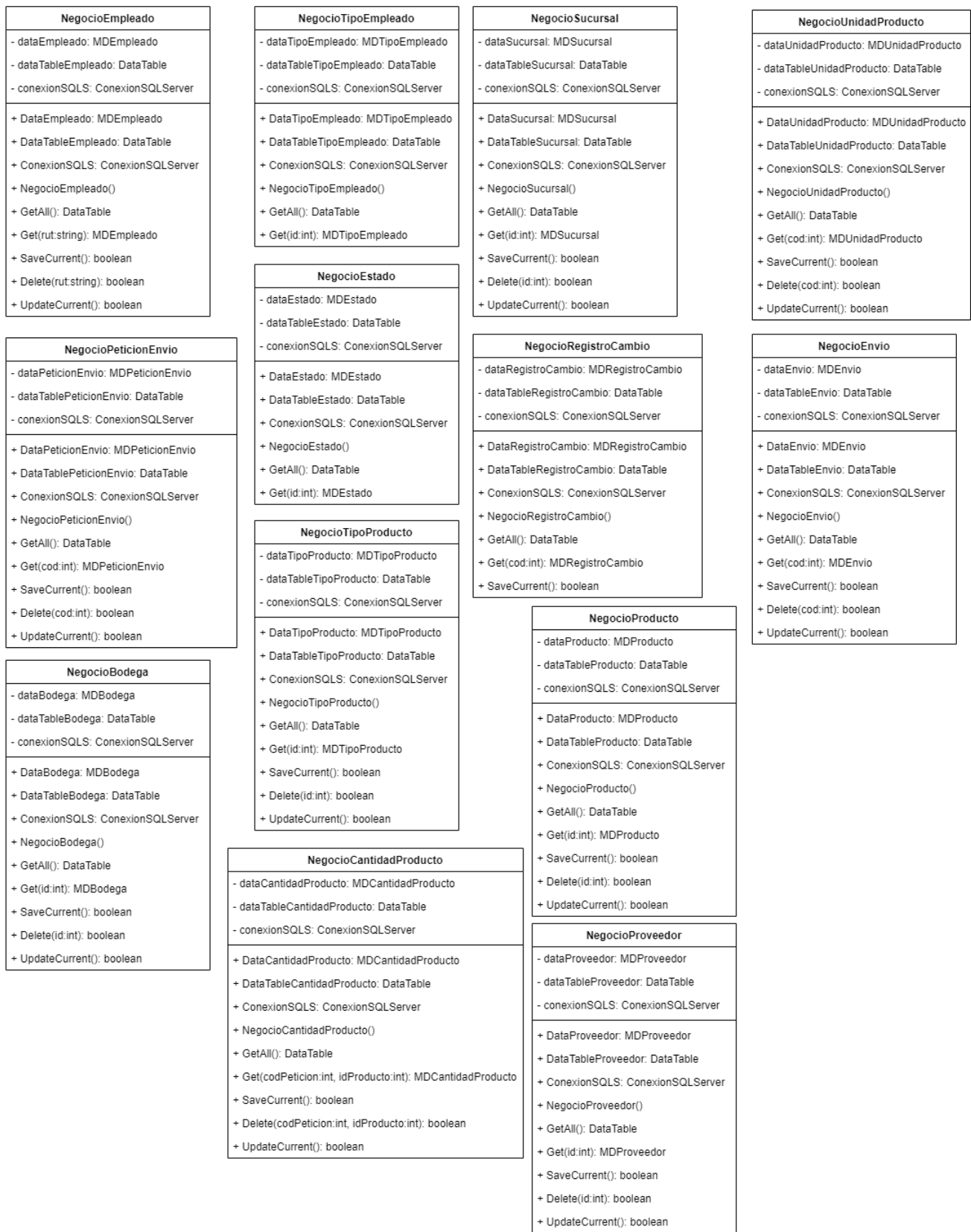


Modelo de Datos



Negocio

Negocio



NegocioEmpleado	NegocioPeticonEnvio
- dataEmpleado: MDEmpleado - dataTableEmpleado: DataTable - conexionSQLS: ConexionSQLServer	- dataPeticonEnvio: MDPeticonEnvio - dataTablePeticonEnvio: DataTable - conexionSQLS: ConexionSQLServer
+ DataEmpleado: MDEmpleado + DataTableEmpleado: DataTable + ConexionSQLS: ConexionSQLServer + NegocioEmpleado() + GetAll(): DataTable + Get(rut:string): MDEmpleado + SaveCurrent(): boolean + Delete(rut:string): boolean + UpdateCurrent(): boolean	+ DataPeticonEnvio: MDPeticonEnvio + DataTablePeticonEnvio: DataTable + ConexionSQLS: ConexionSQLServer + NegocioPeticonEnvio() + GetAll(): DataTable + Get(cod:int): MDPeticonEnvio + SaveCurrent(): boolean + Delete(cod:int): boolean + UpdateCurrent(): boolean

NegocioBodega	NegocioTipoEmpleado
- dataBodega: MDBodega - dataTableBodega: DataTable - conexionSQLS: ConexionSQLServer	- dataTipoEmpleado: MDTipoEmpleado - dataTableTipoEmpleado: DataTable - conexionSQLS: ConexionSQLServer
+ DataBodega: MDBodega + DataTableBodega: DataTable + ConexionSQLS: ConexionSQLServer + NegocioBodega() + GetAll(): DataTable + Get(id:int): MDBodega + SaveCurrent(): boolean + Delete(id:int): boolean + UpdateCurrent(): boolean	+ DataTipoEmpleado: MDTipoEmpleado + DataTableTipoEmpleado: DataTable + ConexionSQLS: ConexionSQLServer + NegocioTipoEmpleado() + GetAll(): DataTable + Get(id:int): MDTipoEmpleado

NegocioEstado	NegocioTipoProducto
- dataEstado: MDEstado - dataTableEstado: DataTable - conexionSQLS: ConexionSQLServer	- dataTipoProducto: MDTipoProducto - dataTableTipoProducto: DataTable - conexionSQLS: ConexionSQLServer
+ DataEstado: MDEstado + DataTableEstado: DataTable + ConexionSQLS: ConexionSQLServer + NegocioEstado() + GetAll(): DataTable + Get(id:int): MDEstado	+ DataTipoProducto: MDTipoProducto + DataTableTipoProducto: DataTable + ConexionSQLS: ConexionSQLServer + NegocioTipoProducto() + GetAll(): DataTable + Get(id:int): MDTipoProducto + SaveCurrent(): boolean + Delete(id:int): boolean + UpdateCurrent(): boolean

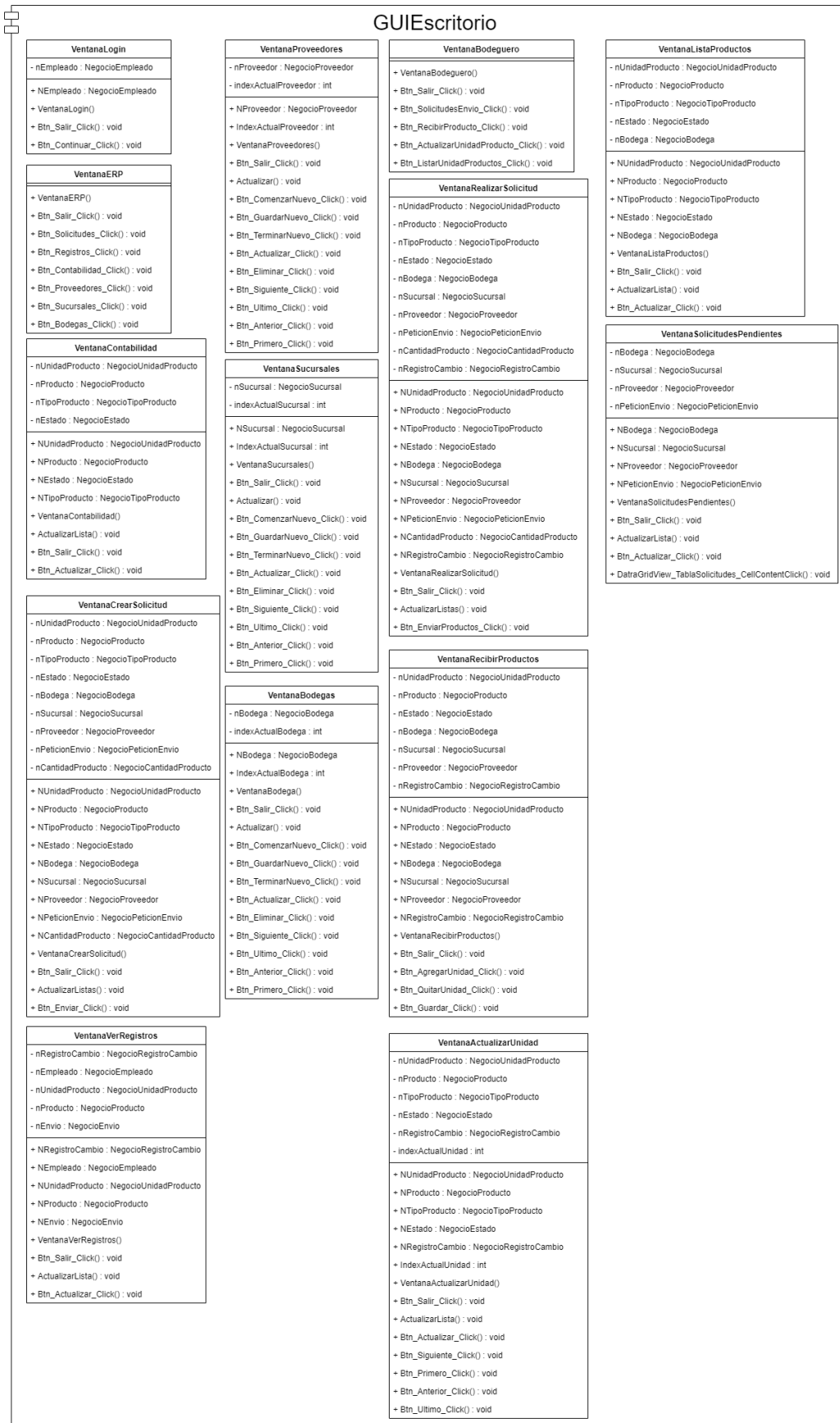
NegocioCantidadProducto	NegocioSucursal
- dataCantidadProducto: MDCantidadProducto - dataTableCantidadProducto: DataTable - conexionSQLS: ConexionSQLServer	- dataSucursal: MDSucursal - dataTableSucursal: DataTable - conexionSQLS: ConexionSQLServer
+ DataCantidadProducto: MDCantidadProducto + DataTableCantidadProducto: DataTable + ConexionSQLS: ConexionSQLServer + NegocioCantidadProducto() + GetAll(): DataTable + Get(codPetición:int, idProducto:int): MDCantidadProducto + SaveCurrent(): boolean + Delete(codPetición:int, idProducto:int): boolean + UpdateCurrent(): boolean	+ DataSucursal: MDSucursal + DataTableSucursal: DataTable + ConexionSQLS: ConexionSQLServer + NegocioSucursal() + GetAll(): DataTable + Get(id:int): MDSucursal + SaveCurrent(): boolean + Delete(id:int): boolean + UpdateCurrent(): boolean

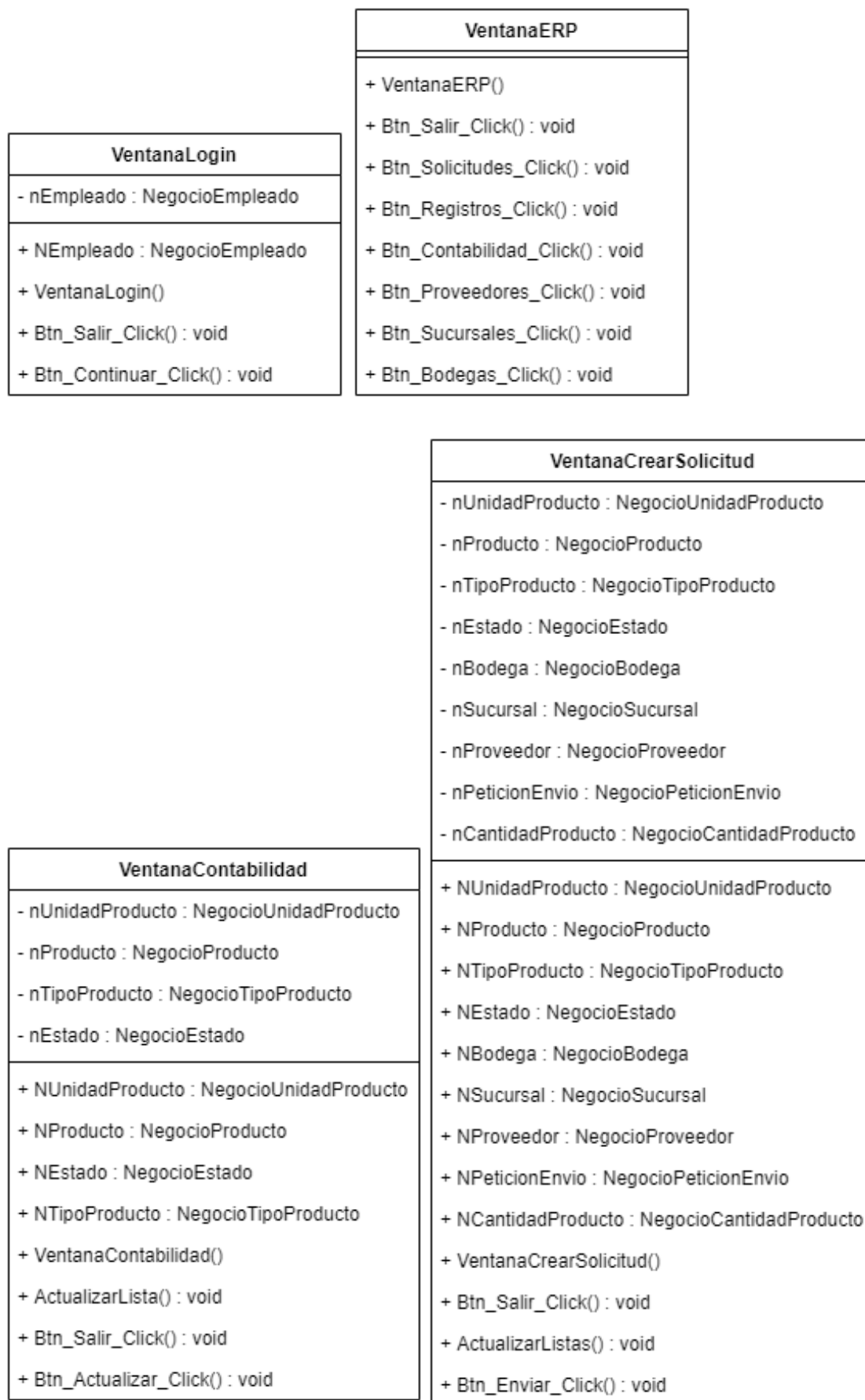
NegocioRegistroCambio	NegocioProducto
- dataRegistroCambio: MDRegistroCambio	- dataProducto: MDProducto
- dataTableRegistroCambio: DataTable	- dataTableProducto: DataTable
- conexionSQLS: ConexionSQLServer	- conexionSQLS: ConexionSQLServer
+ DataRegistroCambio: MDRegistroCambio	+ DataProducto: MDProducto
+ DataTableRegistroCambio: DataTable	+ DataTableProducto: DataTable
+ ConexionSQLS: ConexionSQLServer	+ ConexionSQLS: ConexionSQLServer
+ NegocioRegistroCambio()	+ NegocioProducto()
+ GetAll(): DataTable	+ GetAll(): DataTable
+ Get(cod:int): MDRegistroCambio	+ Get(id:int): MDProducto
+ SaveCurrent(): boolean	+ SaveCurrent(): boolean
	+ Delete(id:int): boolean
	+ UpdateCurrent(): boolean

NegocioProveedor	NegocioUnidadProducto
- dataProveedor: MDProveedor	- dataUnidadProducto: MDUnidadProducto
- dataTableProveedor: DataTable	- dataTableUnidadProducto: DataTable
- conexionSQLS: ConexionSQLServer	- conexionSQLS: ConexionSQLServer
+ DataProveedor: MDProveedor	+ DataUnidadProducto: MDUnidadProducto
+ DataTableProveedor: DataTable	+ DataTableUnidadProducto: DataTable
+ ConexionSQLS: ConexionSQLServer	+ ConexionSQLS: ConexionSQLServer
+ NegocioProveedor()	+ NegocioUnidadProducto()
+ GetAll(): DataTable	+ GetAll(): DataTable
+ Get(id:int): MDProveedor	+ Get(cod:int): MDUnidadProducto
+ SaveCurrent(): boolean	+ SaveCurrent(): boolean
+ Delete(id:int): boolean	+ Delete(cod:int): boolean
+ UpdateCurrent(): boolean	+ UpdateCurrent(): boolean

NegocioEnvio
- dataEnvio: MDEnvio - dataTableEnvio: DataTable - conexionSQLS: ConexionSQLServer
+ DataEnvio: MDEnvio + DataTableEnvio: DataTable + ConexionSQLS: ConexionSQLServer + NegocioEnvio() + GetAll(): DataTable + Get(cod:int): MDEnvio + SaveCurrent(): boolean + Delete(cod:int): boolean + UpdateCurrent(): boolean

Interfaz de Usuario Gráfica de Escritorio (GUI Escritorio)





VentanaVerRegistros	VentanaProveedores
- nRegistroCambio : NegocioRegistroCambio - nEmpleado : NegocioEmpleado - nUnidadProducto : NegocioUnidadProducto - nProducto : NegocioProducto - nEnvio : NegocioEnvio + NRegistroCambio : NegocioRegistroCambio + NEmpleado : NegocioEmpleado + NUnidadProducto : NegocioUnidadProducto + NProducto : NegocioProducto + NEnvio : NegocioEnvio + VentanaVerRegistros() + Btn_Salir_Click() : void + ActualizarLista() : void + Btn_Actualizar_Click() : void	- nProveedor : NegocioProveedor - indexActualProveedor : int + NProveedor : NegocioProveedor + IndexActualProveedor : int + VentanaProveedores() + Btn_Salir_Click() : void + Actualizar() : void + Btn_ComenzarNuevo_Click() : void + Btn_GuardarNuevo_Click() : void + Btn_TerminarNuevo_Click() : void + Btn_Actualizar_Click() : void + Btn_Eliminar_Click() : void + Btn_Siguiente_Click() : void + Btn_Ultimo_Click() : void + Btn_Anterior_Click() : void + Btn_Primer_Click() : void

VentanaSucursales	VentanaBodegas
- nSucursal : NegocioSucursal - indexActualSucursal : int + NSucursal : NegocioSucursal + IndexActualSucursal : int + VentanaSucursales() + Btn_Salir_Click() : void + Actualizar() : void + Btn_ComenzarNuevo_Click() : void + Btn_GuardarNuevo_Click() : void + Btn_TerminarNuevo_Click() : void + Btn_Actualizar_Click() : void + Btn_Eliminar_Click() : void + Btn_Siguiente_Click() : void + Btn_Ultimo_Click() : void + Btn_Anterior_Click() : void + Btn_Primer_Click() : void	- nBodega : NegocioBodega - indexActualBodega : int + NBodega : NegocioBodega + IndexActualBodega : int + VentanaBodega() + Btn_Salir_Click() : void + Actualizar() : void + Btn_ComenzarNuevo_Click() : void + Btn_GuardarNuevo_Click() : void + Btn_TerminarNuevo_Click() : void + Btn_Actualizar_Click() : void + Btn_Eliminar_Click() : void + Btn_Siguiente_Click() : void + Btn_Ultimo_Click() : void + Btn_Anterior_Click() : void + Btn_Primer_Click() : void

VentanaBodeguero
+ VentanaBodeguero() + Btn_Salir_Click() : void + Btn_SolicitudesEnvio_Click() : void + Btn_RecibirProducto_Click() : void + Btn_ActualizarUnidadProducto_Click() : void + Btn_ListarUnidadProductos_Click() : void

VentanaRealizarSolicitud
- nUnidadProducto : NegocioUnidadProducto - nProducto : NegocioProducto - nTipoProducto : NegocioTipoProducto - nEstado : NegocioEstado - nBodega : NegocioBodega - nSucursal : NegocioSucursal - nProveedor : NegocioProveedor - nPeticonEnvio : NegocioPeticonEnvio - nCantidadProducto : NegocioCantidadProducto - nRegistroCambio : NegocioRegistroCambio
+ NUnidadProducto : NegocioUnidadProducto + NProducto : NegocioProducto + NTipoProducto : NegocioTipoProducto + NEstado : NegocioEstado + NBodega : NegocioBodega + NSucursal : NegocioSucursal + NProveedor : NegocioProveedor + NPeticonEnvio : NegocioPeticonEnvio + NCantidadProducto : NegocioCantidadProducto + NRegistroCambio : NegocioRegistroCambio + VentanaRealizarSolicitud() + Btn_Salir_Click() : void + ActualizarListas() : void + Btn_EnviarProductos_Click() : void

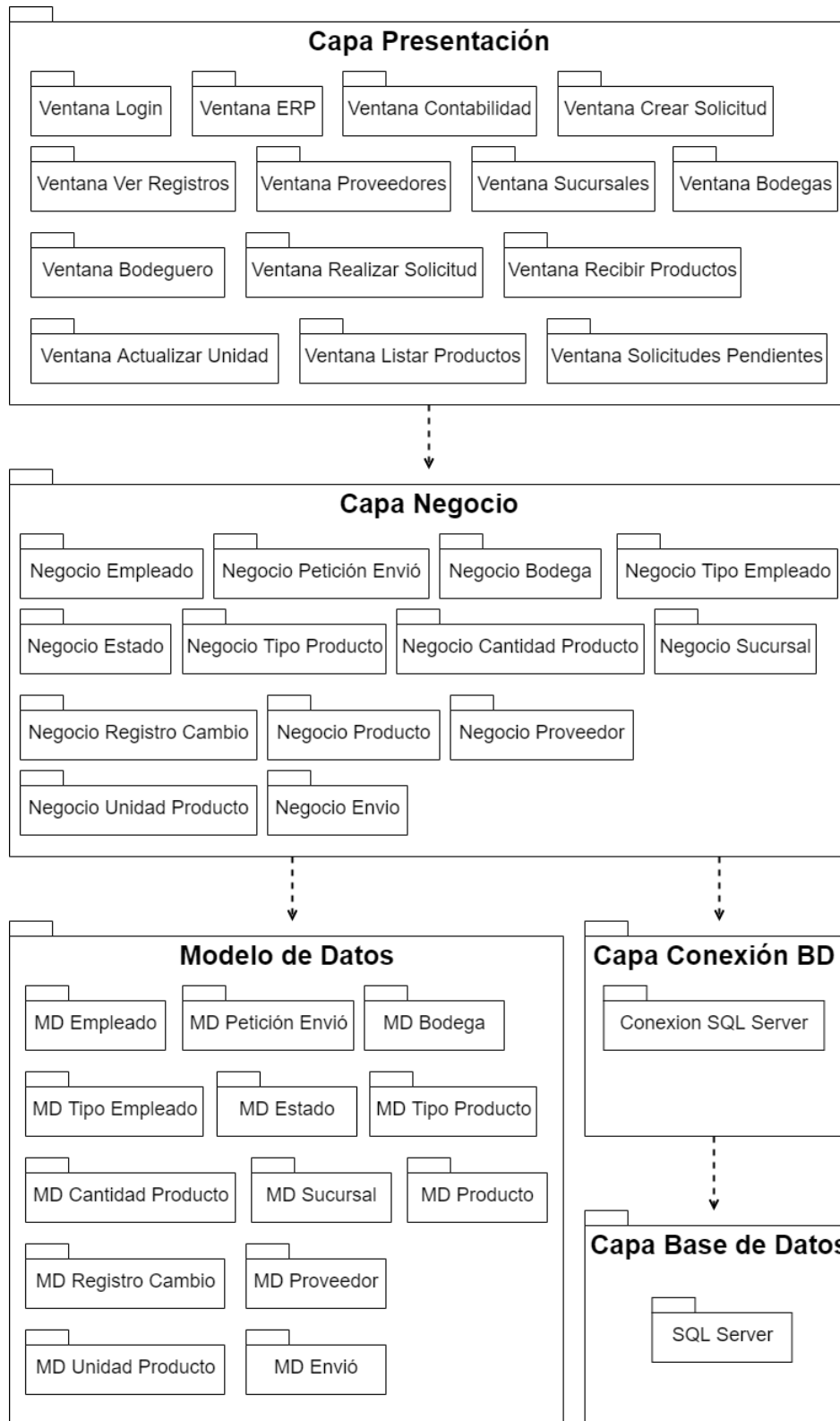
VentanaRecibirProductos
- nUnidadProducto : NegocioUnidadProducto - nProducto : NegocioProducto - nEstado : NegocioEstado - nBodega : NegocioBodega - nSucursal : NegocioSucursal - nProveedor : NegocioProveedor - nRegistroCambio : NegocioRegistroCambio
+ NUnidadProducto : NegocioUnidadProducto + NProducto : NegocioProducto + NEstado : NegocioEstado + NBodega : NegocioBodega + NSucursal : NegocioSucursal + NProveedor : NegocioProveedor + NRegistroCambio : NegocioRegistroCambio + VentanaRecibirProductos() + Btn_Salir_Click() : void + Btn_AgregarUnidad_Click() : void + Btn_QuitarUnidad_Click() : void + Btn_Guardar_Click() : void

VentanaActualizarUnidad	
- nUnidadProducto : NegocioUnidadProducto - nProducto : NegocioProducto - nTipoProducto : NegocioTipoProducto - nEstado : NegocioEstado - nRegistroCambio : NegocioRegistroCambio - indexActualUnidad : int	
+ NUnidadProducto : NegocioUnidadProducto + NProducto : NegocioProducto + NTipoProducto : NegocioTipoProducto + NEstado : NegocioEstado + NRegistroCambio : NegocioRegistroCambio + IndexActualUnidad : int + VentanaActualizarUnidad() + Btn_Salir_Click() : void + ActualizarLista() : void + Btn_Actualizar_Click() : void + Btn_Siguiente_Click() : void + Btn_Primer_Click() : void + Btn_Anterior_Click() : void + Btn_Ultimo_Click() : void	
VentanaListaProductos	
- nUnidadProducto : NegocioUnidadProducto - nProducto : NegocioProducto - nTipoProducto : NegocioTipoProducto - nEstado : NegocioEstado - nBodega : NegocioBodega	
+ NUnidadProducto : NegocioUnidadProducto + NProducto : NegocioProducto + NTipoProducto : NegocioTipoProducto + NEstado : NegocioEstado + NBodega : NegocioBodega + VentanaListaProductos() + Btn_Salir_Click() : void + ActualizarLista() : void + Btn_Actualizar_Click() : void	

VentanaSolicitudesPendientes
- nBodega : NegocioBodega - nSucursal : NegocioSucursal - nProveedor : NegocioProveedor - nPeticonEnvio : NegocioPeticonEnvio
+ NBodega : NegocioBodega + NSucursal : NegocioSucursal + NProveedor : NegocioProveedor + NPeticonEnvio : NegocioPeticonEnvio + VentanaSolicitudesPendientes() + Btn_Salir_Click() : void + ActualizarLista() : void + Btn_Actualizar_Click() : void + DataGridView_TablaSolicitudes_CellContentClick() : void

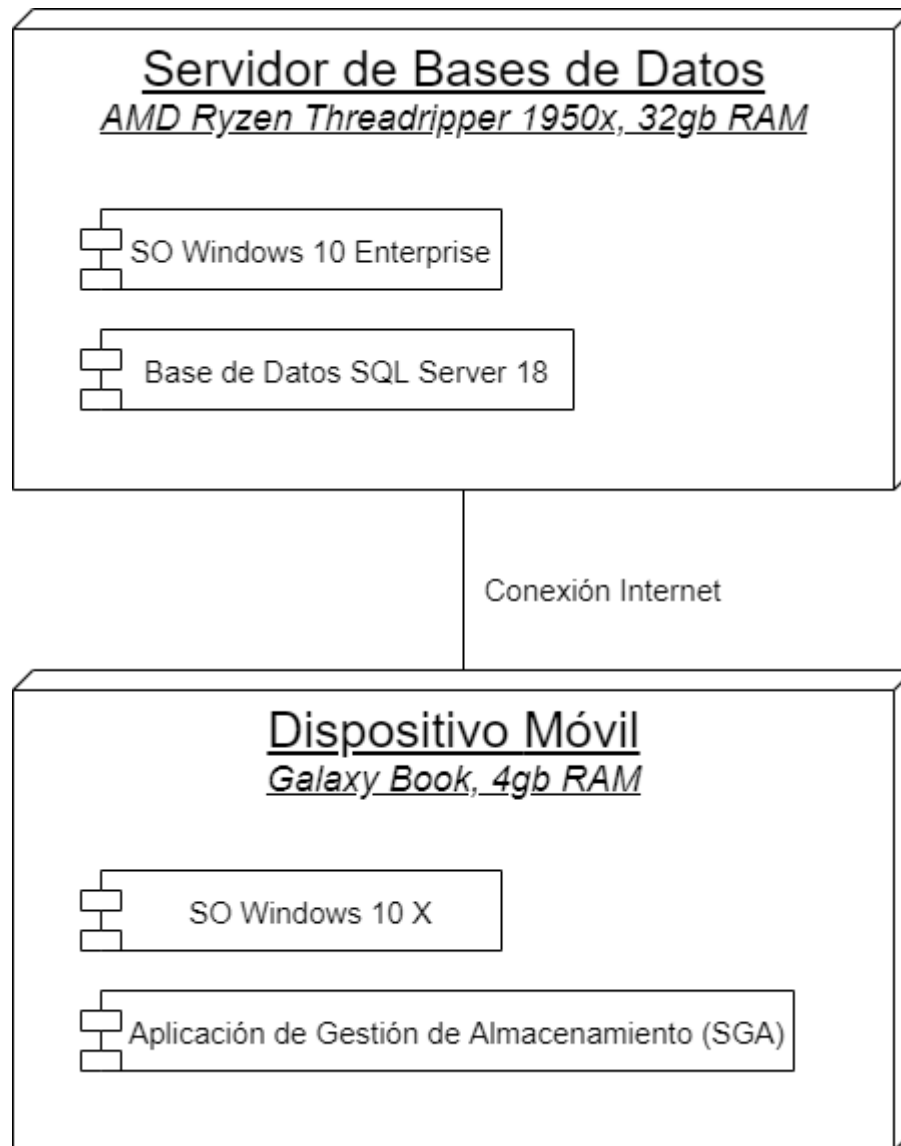
Vista Desarrollo

Diagrama de Paquetes



Vista Física

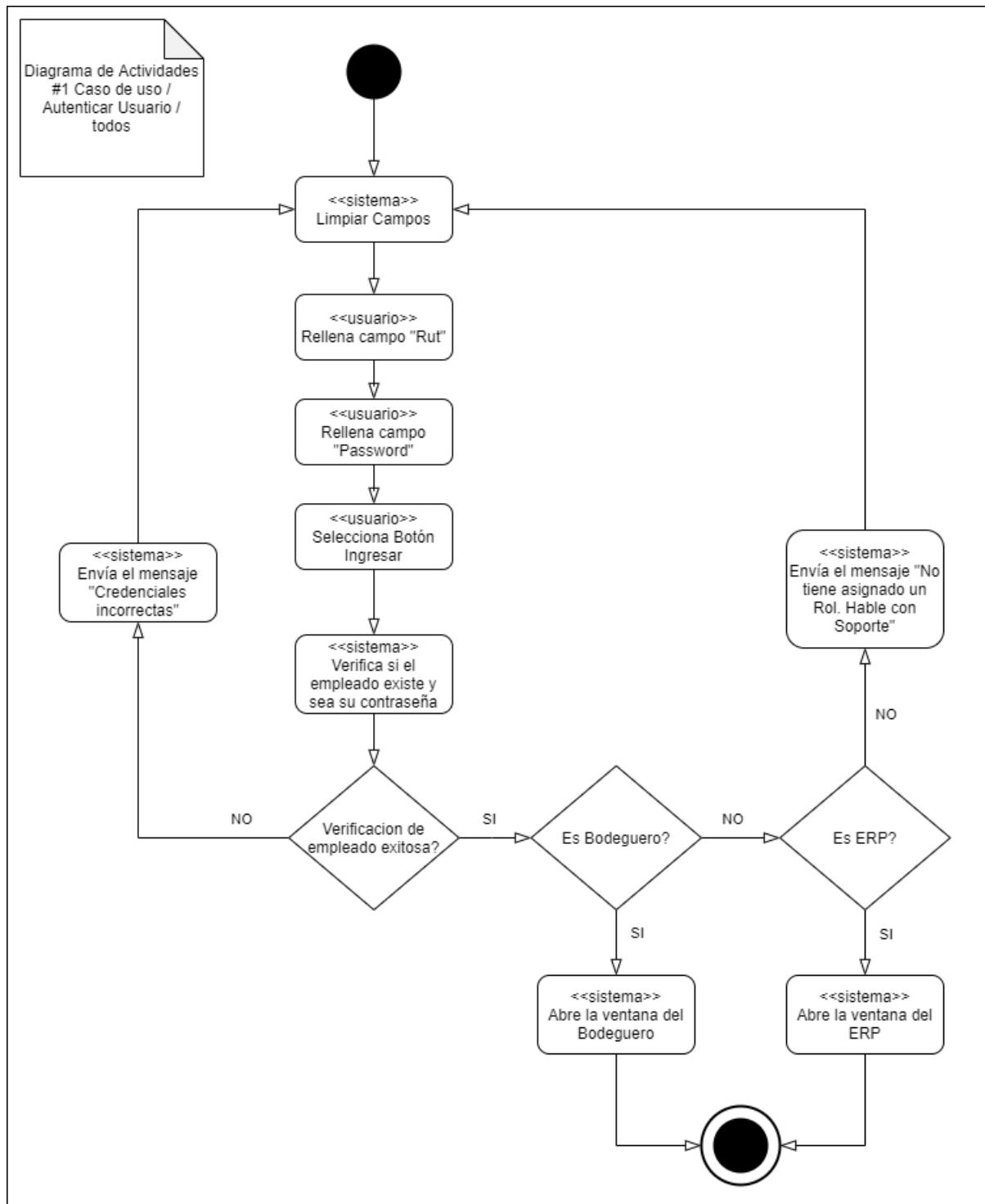
Diagrama de Despliegue



Vista de Procesos

Diagrama de Actividades

Nº 1 Autenticar Usuario



MOCKUPS

Login

○ ○ ○

LOGIN



USER

input

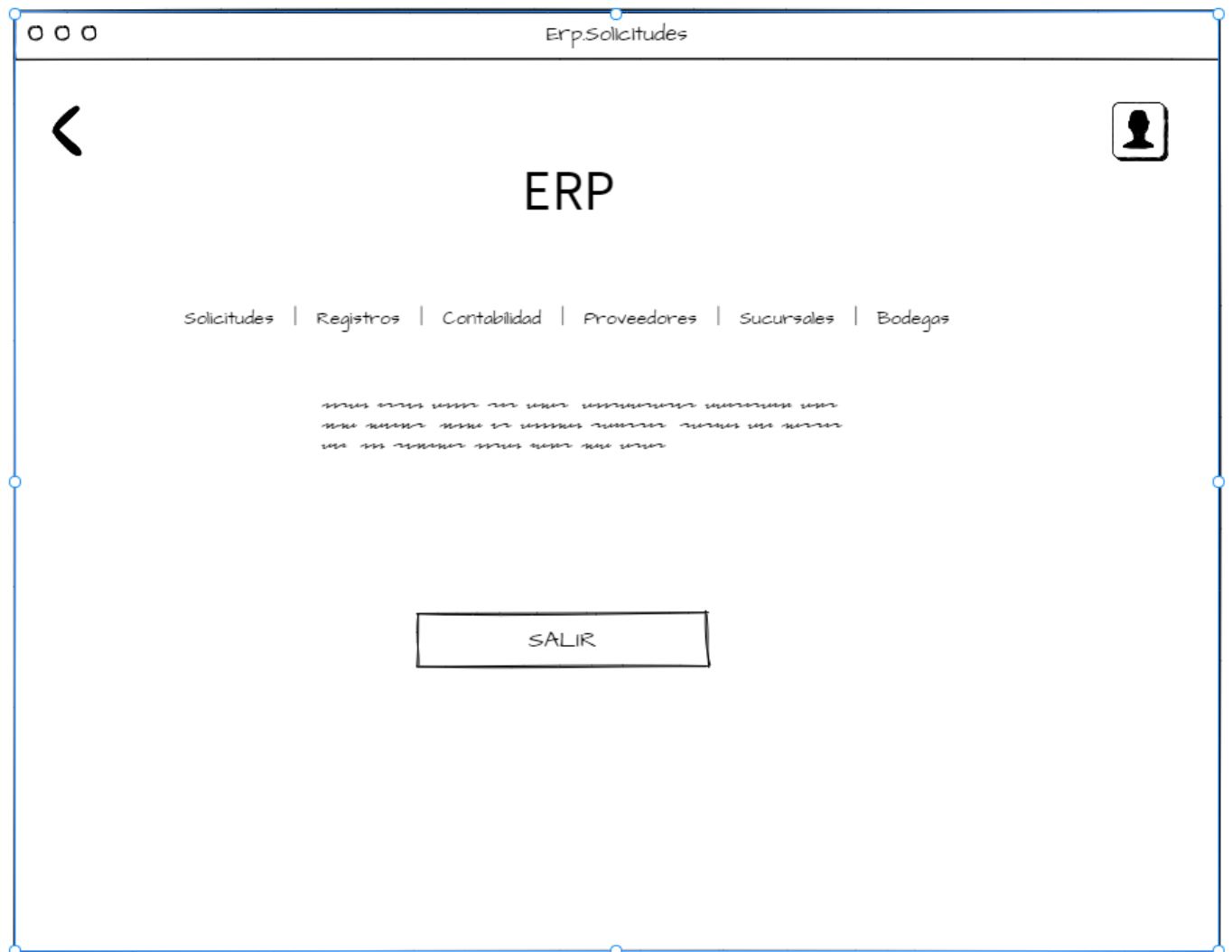
PASSWORD

input

INGRESAR

SALIR

ERP menú



Solicitudes

Erp.Solicitudes

<

Solicitudes

Solicitudes | Registros | Contabilidad | Proveedores | Sucursales | Bodegas

Lorem ipsum dolor sit amet consectetur adipiscing elit
 sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
 Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Enviar	PRODUCTO	CANTIDAD
<input checked="" type="checkbox"/>	8GB RAM DDR4	150/350
<input type="checkbox"/>	DISCO DURO 7200RPM 1TB	90/300
<input checked="" type="checkbox"/>	AMD RX 570 gigabyte	65/120

Enviar productos

SALIR


Registros

ooo

Erp.Solicitudes

<

Registros



Solicitudes | **Registros** | Contabilidad | Proveedores | Sucursales | Bodegas

...

Estado	PRODUCTO	CANTIDAD
modificado	8GB RAM DDR4	150/350
Eliminado	DISCO DURO 7200RPM 1TB	90/300
modificado	AMD RX 570 gigabyte	65/120

SALIR

Contabilidad

ooo


Erp.Solicitudes

<

Contabilidad

Solicitudes | Registros | Contabilidad | Proveedores | Sucursales | Bodegas

...



Estado	PRODUCTO	CANTIDAD	BODEGA
NUEVO	8GB RAM DDR4	300	LA GRANJA
OPENBOX	8GB RAM DDR4	50	LA GRANJA
NUEVO	DISCO DURO 7200RPM 1TB	300	LA GRANJA
NUEVO	AMD RX 570 gigabyte	50	LA FLORIDA
DAÑADO	AMD RX 570 gigabyte	10	LA FLORIDA


SALIR

Proveedores

o o o

Erp.Solicitudes

<



Proveedores

Solicitudes | Registros | Contabilidad | Proveedores | Sucursales | Bodegas

ID

input

NOMBRE

input

DIRECCION

input

AGREGAR

ELIMINAR

MODIFICAR

SALIR

<|

<

input

>


>|

Sucursales

o o o

Erp.Solicitudes

<



Sucursales

Solicitudes | Registros | Contabilidad | Proveedores | Sucursales | Bodegas

ID

input

NOMBRE

input

DIRECCION

input

AGREGAR

ELIMINAR

MODIFICAR

SALIR

<|

<

input

>


>|

Bodegas

o o o

Erp.Solicitudes

<



Bodegas

Solicitudes | Registros | Contabilidad | Proveedores | Sucursales | Bodegas

ID

Input

NOMBRE

Input

DIRECCION

Input

AGREGAR

ELIMINAR

MODIFICAR

SALIR

<|

<

>

>|

Bodega Menú




Enviar Producto

ooo

Erp.Solicitudes

<

Enviar Producto



Enviar productos | recibir productos | solicitudes | Listar productos | Actualizar productos

Peticion Envio

Input

Destino

Input

PRODUCTO

ID	NOMBRE	CANTIDAD
01	8GB RAM DDR4	0/10
02	8GB RAM DDR4	0/10
03	DISCO DURO	0/10
04	AMD RX 570	0/10
05	AMD RX 570	0/10

UNIDADES

<input type="checkbox"/>	ID	ESTADO	NOMBRE
<input type="checkbox"/>	P01	nuevo	MSI
<input type="checkbox"/>	P02	openbox	WINPY
<input type="checkbox"/>	P03	nuevo	SPDIGITAL
<input type="checkbox"/>	P04	usado	ALLTEC
<input type="checkbox"/>	P05	nuevo	PC EXPRES

ENVIAR

CANCELAR

<|

<



//

>

>

Recibir Producto

000 Erp.Solicitudes



Recibir Producto

Enviar productos | recibir productos | Solicitudes | Listar productos | Actualizar productos

label

label

label

AGREGAR

BORRAR

<input type="checkbox"/>	ID	NOMBRE	CANTIDAD
<input type="checkbox"/>	01	8GB RAM DDR4	20
<input type="checkbox"/>	02	8GB RAM DDR3	15
<input type="checkbox"/>	03	DISCO DURO	15
<input type="checkbox"/>	04	AMD RX 570	10


SALIR

Solicitudes

ERP SOLICITUDES

<

SOLICITUDES



Enviar productos | recibir productos | Solicitudes | Listar productos | Actualizar productos

SOLICITUD	NOMBRE	CANTIDAD
01	8GB RAM DDR4	20
02	8GB RAM DDR3	15
03	DISCO DURO	15
04	AMD RX 570	10

SALIR

Listar productos

Site Title

<

LISTA DE PRODUCTOS

Enviar productos | recibir productos | Solicitudes de ERP | Listar productos

FILTRAR POR : ☐ PRECIO ☐ CANTIDAD ☐ ABC

PRODUCTO	CANTIDAD	PRECIO
MEMORIA RAM 8GB	300	\$35'000
DISCOS DURO 1TB	5'00	\$30'000
PROCESADORES I5 9+00F	150	\$15'0000
TARJETAS GRAFICA 2060	75	\$+00000

Actualizar Productos

Site Title

<

ACTUALIZAR PRODUCTOS

Enviar productos | recibir productos | Solicitudes de ERP | Listar productos | Actualizar productos

FILTRAR POR : ☐ PRECIO ☐ CANTIDAD ☐ ABC

<input type="checkbox"/>	PRODUCTO	CANTIDAD	PRECIO
<input type="checkbox"/>	MEMORIA RAM 8GB	300	\$35000
<input type="checkbox"/>	DISCOS DURO 1TB	500	\$30000
<input type="checkbox"/>	PROCESADORES I5 9+00F	150	\$150000
<input type="checkbox"/>	TARJETAS GRAFICA 2060	75	\$+00000

Patrones y Estilos Arquitectónicos

El patrón de diseño a utilizar es **Facade**. Este es un patrón de diseño estructural, enfocado en utilizar módulos que interactúan entre ellos. Esto reduce la complejidad del sistema al subdividirlo en módulos, minimizando las comunicaciones y dependencias entre estos. Este patrón principalmente debe tener una interfaz simple que oculte la complejidad interna de los subsistemas y sus conexiones. Este funciona bien con las arquitecturas **N Capas** y **SOA**, porque puede utilizar estos para generar los módulos.

