

**UNIVERSIDAD DE EL SALVADOR**  
**FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE**



**Diseño de Sistema de Ventas**  
**Versión 2.**

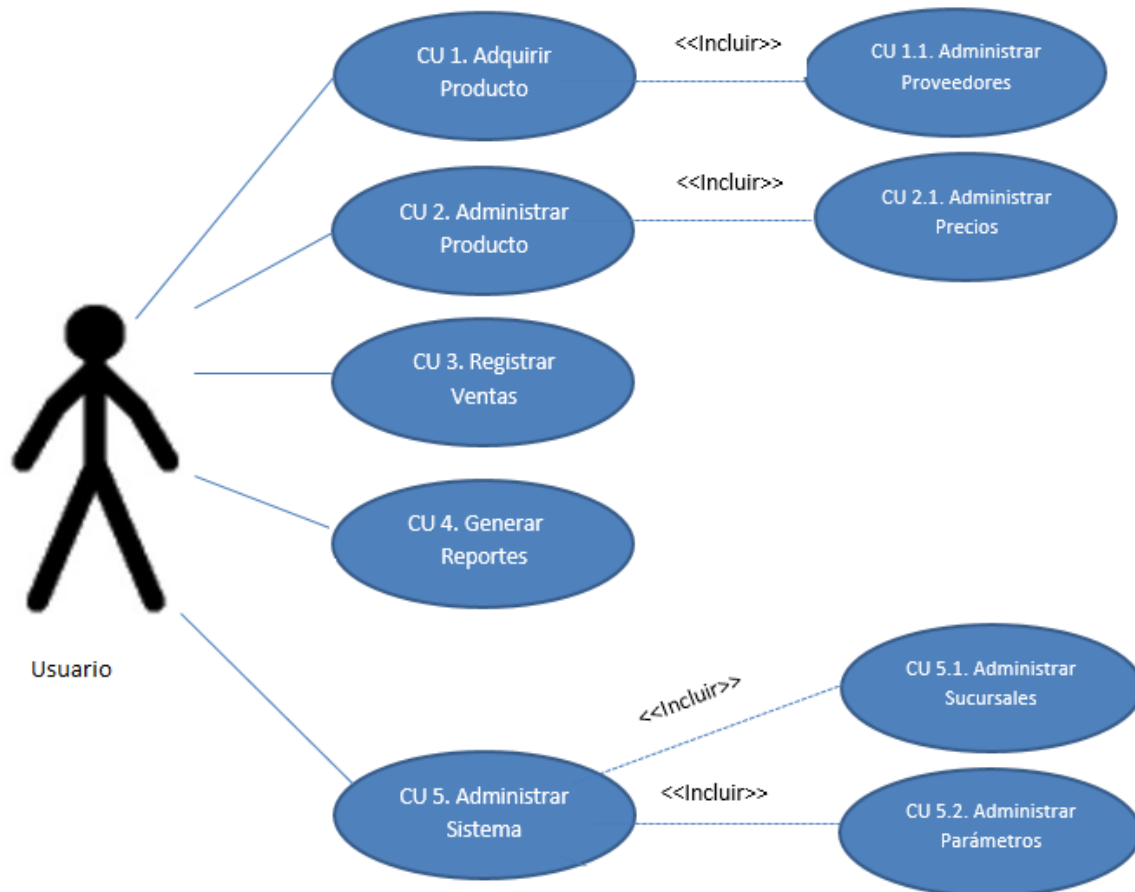
**Alvarez Aragón, Rogelio Antonio**

**López Díaz, Walter Hazael**

**Castro Solís, Francisco Nahum**

**García Arriola, Wilfredo Enrique**

## Casos de Uso de Sistema de Ventas versión 2.0



### Descripción del Sistema Ventas

El sistema de ventas es un software diseñado para ayudar a las pequeñas y medianas empresas en el manejo y control de todo lo concerniente a sus ventas, manejo de productos y recolección de datos que permitan evaluar el rendimiento del negocio. El software será implementado en dos versiones; en la versión 2.0 se podrán realizar todas las tareas antes descritas e implementadas en la versión 1.0. Las mejoras implementadas en la versión 2.0 será la posibilidad de asignar diferentes precios a los productos, realizar ventas tipo borrador y crédito fiscal para lo cual se podrán configurar todos estos valores constantes. De igual manera

se podrán generar reportes físicos al momento de consolidar las ventas; se permitirá también el manejo de distintos usuarios cada uno con tareas específicas en el sistema y la administración de más de una sucursal.

## **CU 1 Adquirir Producto.**

Al adquirir nuevos productos, estos serán registrados en el sistema junto con los detalles de la compra, actualizando el inventario con el nuevo producto que se agrega. Cada producto tendrá un código identificador único que será generado por el mismo sistema para evitar duplicados. El registro de cada nuevo producto contiene los detalles y precio de dicho producto. La compra se hará a proveedores registrados en el sistema.

**Actores:** Encargado de tienda.

**Funciones:** F1

**Condición Previa:** Productos comprados no presentes en el sistema.

**Condición Posterior:** Productos registrados e inventario actualizado.

### **Curso normal de eventos**

#### Acción del actor

1. El usuario registra en el sistema los productos con todos los detalles a partir de un registro físico de compra.
2. A la vez que se agrega cada producto, se especifica el precio y otros detalles propios del producto.

#### Respuesta del Sistema

3. El sistema genera el código de identificación único para cada producto.
4. Se muestra un mensaje de confirmación.

## CU 1.1 Administrar Proveedores (Incluido en CU 1)

El sistema permitirá agregar nuevos proveedores con sus datos, estos consisten en información general del proveedor y sus contactos. Además se podrá modificar y eliminar cada proveedor.

**Actores:** Encargado de Tienda.

**Funciones:** F2.2

**Condición Previa:** Necesidad de agregar nuevos proveedores.

**Condición Posterior:** Registro de proveedores actualizado.

### Curso normal de eventos

Acción del actor	Respuesta del Sistema
1. Se accede al panel de administración de proveedores en el sistema y se podrá agregar, modificar o eliminar los proveedores	2. Se muestra un mensaje de confirmación.

## CU 2 Administrar Productos

La tarea de administrar productos se realizará por el encargado de la tienda o algún empleado autorizado, este puede agregar un producto, así mismo, modificar y eliminar dicho producto. Los productos a eliminar solo podrán ser suprimidos si no contiene ningún registro vigente, si esa condición no se cumple, el producto no podrá ser eliminado. El registro de un producto nuevo se realizará por medio de su código para identificarlo y si existe se actualizará.

**Actores:** Encargado de Tienda o Empleado.

**Funciones:** F2.1, F2.3

**Condición Previa:** Necesidad de realizar una modificación a productos.

**Condición Posterior:** Productos actualizados.

### Curso normal de eventos

Acción del actor	Respuesta del Sistema
1. El encargado elige en el menú la opción de administrar producto, donde podrá agregar, buscar, modificar o eliminar el producto.	2. El responde de acuerdo a la acción que sea solicitada y válida.
3. Con la información requerida de cada producto se agrega al sistema.	4. Si el producto no existe se agrega, si existe se modifica en la base de datos o se elimina si es requerido.
	5. Se muestra un mensaje de confirmación.

## CU 2.1 Administrar Precios (Incluido en CU 2)

El sistema permitirá agregar distintos precios al producto en inventario, los cuales serán distintos si el producto se vende por unidad o por mayoreo. Estos precios serán modificados dependiendo de las leyes vigentes en ese momento.

**Actores:** Encargado de Tienda o Empleado asignado.

**Funciones:** F2.4

**Condición Previa:** Necesidad de agregar precios a productos.

**Condición Posterior:** Precios registrados.

### Curso normal de eventos

Acción del actor	Respuesta del Sistema
1. Se accede al panel de administración de productos.	2. Se muestra todas las opciones para administrar productos.
3. Se elige la opción de agregar o modificar los precios de un producto y se elige guardar cambios.	4. El sistema guardara los cambios y mostrara un mensaje en pantalla.

## CU 3 Registrar Venta

Se realiza y registra una venta al detalle, para la actual versión, se calculan impuestos si se elige una venta con factura, de lo contrario solo define el monto de venta informal. Se brinda una interfaz que permita vender los diferentes productos, se captura cada producto con su código de barra y su identificador único, se selecciona el tipo de venta y el sistema registra la venta.

**Actores:** Encargado de Venta.

**Funciones:** F3.1, F3.2

**Condición Previa:** Necesita registrar una venta

**Condición Posterior:** Venta registrada e inventario actualizado

### Curso normal de eventos

Acción del actor	Respuesta del Sistema
1. Se inicia una venta en el sistema	
2. Se agrega cada uno de los productos vendidos, ya sea por código de barra y correlativo o por búsqueda de productos en el inventario.	3. Se muestran los productos elegidos para vender, su código de barra correlativo, nombre y descripción.
4. Se escoge la opción de venta informal o de precio si es con factura para el producto y se verifica y guarda la venta.	5. El sistema calcula impuestos si es venta con factura, si es venta sin factura muestra los totales sin impuesto y se actualiza el sistema.

## Generar Reporte (Incluido en CU 4)

El Sistema al momento de consolidar las ventas, tendrá la opción de generar un reporte con los datos solicitados e imprimirlos.

**Actores:** Encargado de Tienda o Empleado asignado.

**Funciones:** F4.2, F4.3

**Condición Previa:** Necesidad de generar reporte de datos.

**Condición Posterior:** Reporte listo para imprimir.

### Curso normal de eventos

#### Acción del actor

1. Se accede al panel de consolidar ventas.
3. Se elige la opción de generar un reporte con los datos necesarios.

#### Respuesta del Sistema

2. Se muestra todas las opciones.
4. El sistema genera un reporte listo para imprimir.



## CU 5.1 Administrar Sucursales (Incluido en CU 5)

El administrador del Sistema tiene tareas únicas tales como: editar datos de sucursales y dar de baja alguna cuando está estéril fuera de funciones y cerrada y/o agregar una nueva sucursal.

**Actores:** Administrador de la Tienda.

**Funciones:** F5.2

**Condición Previa:** Necesidad de editar sucursal y/o agregar sucursal.

**Condición Posterior:** Datos de sucursal actualizados.

### Curso normal de eventos

Acción del actor	Respuesta del Sistema
1. El administrador elige la opción de administrar datos de sucursal para editar datos teniendo la opción de dar de baja si la sucursal está ya fuera de funciones y cerrada. Además cuenta con la opción de agregar una nueva sucursal por posible crecimiento del negocio.	2. El sistema muestra las opciones de editar y agregar sucursal.
3. Se registran los datos de sucursales modificados y/o la nueva sucursal agregada.	4. Se muestra un mensaje de confirmación.

## CU 5.2 Administrar Parámetros (Incluido en CU 5)

El administrador del Sistema en general es el encargado de realizar y registrar cada edición de los parámetros porcentuales de IVA, pago a cuenta, renta y percepción, (con el fin de mantener al día todo lo referente a contabilidad).

**Actores:** Administrador de la Tienda.

**Funciones:** F5.1

**Condición Previa:** Necesidad de editar parámetros.

**Condición Posterior:** Parámetros actualizados.

### Curso normal de eventos

Acción del actor

Respuesta del Sistema

1. El administrador elige la opción de administrar parámetros pudiendo editarlos

2. El sistema muestra las opciones de editar parámetros.

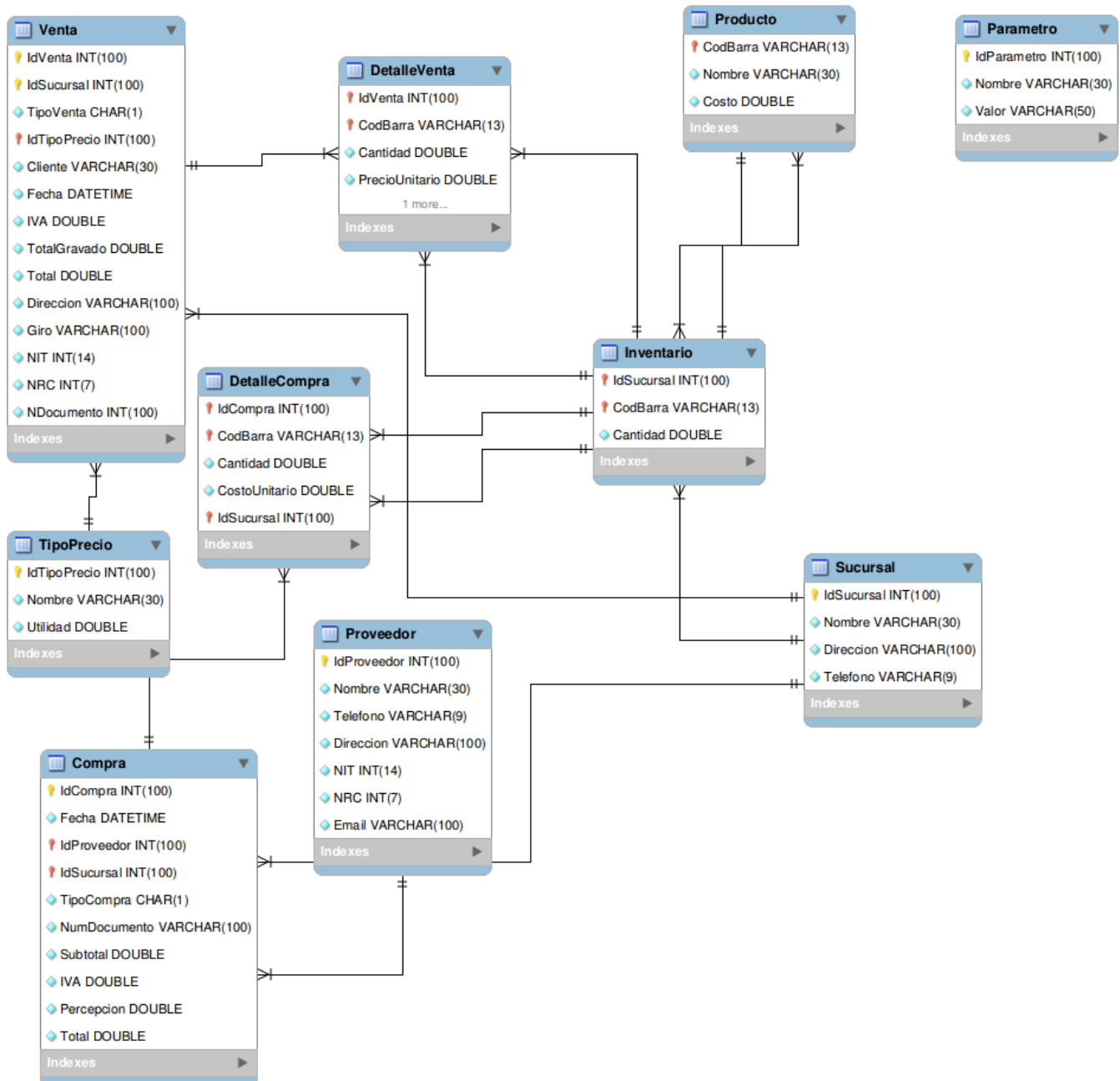
3. Se registran los parámetros modificados.

4. Se muestra un mensaje de confirmación.

## Funciones que realiza el sistema.

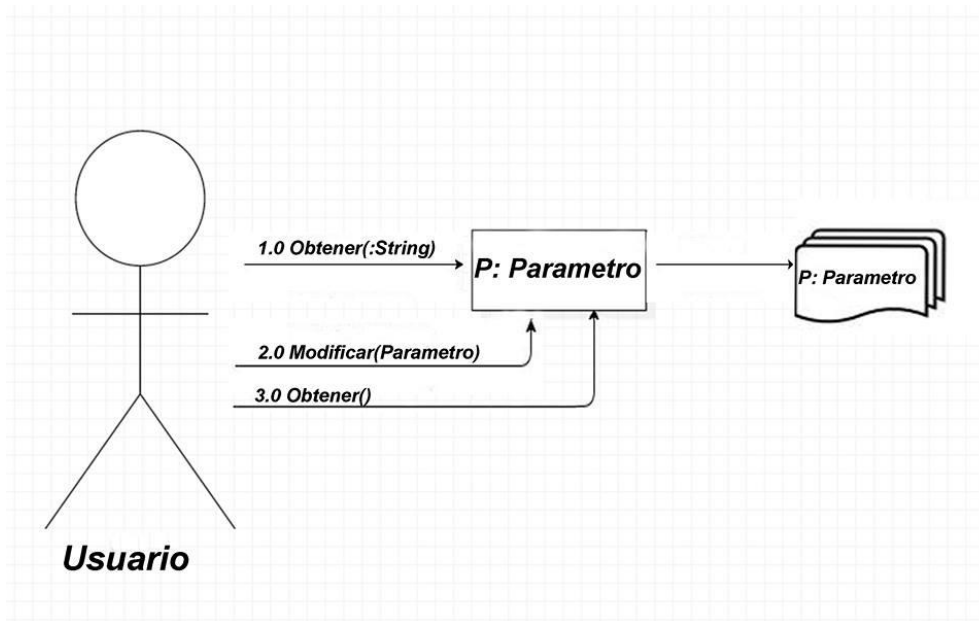
- F1. 1    Agregar productos y sus detalles junto a su respectivo identificador único generado por el sistema, siendo este capaz de garantizar la persistencia de datos en una base.
- F2.1    Modificar y eliminar productos que no tengan registrada ninguna venta.
- F2.2    Registrar nuevos proveedores junto a sus datos generales y contactos para la comunicación. Permitir agregar, modificar y eliminar proveedores con restricciones.
- F2.3    Modificar y eliminar un producto del sistema, verificando que al eliminar, dicho producto no posea registro de ventas, de lo contrario la eliminación es negada.
- F2.4    Administrar lo diferentes precios que un producto pueda tener.
- F3.1    Registra una venta que con solo colocar el código de barras y el correlativo, pueda identificarse de forma única el producto.
- F3.2    Registra los datos de venta y actualiza el inventario.
- F4.1    Proporcionar al usuario un reporte diario de forma digital, que será almacenado en una base de datos.
- F5.1    Administrar datos y variables que el sistema utiliza a la hora de generar las ventas.
- F5.2    Administrar las diferentes sucursales que la tienda tiene, permitiendo agregar, editar o dar de baja cada una de ellas.

## Diagrama Entidad Relación de Sistema de Ventas versión 2.0

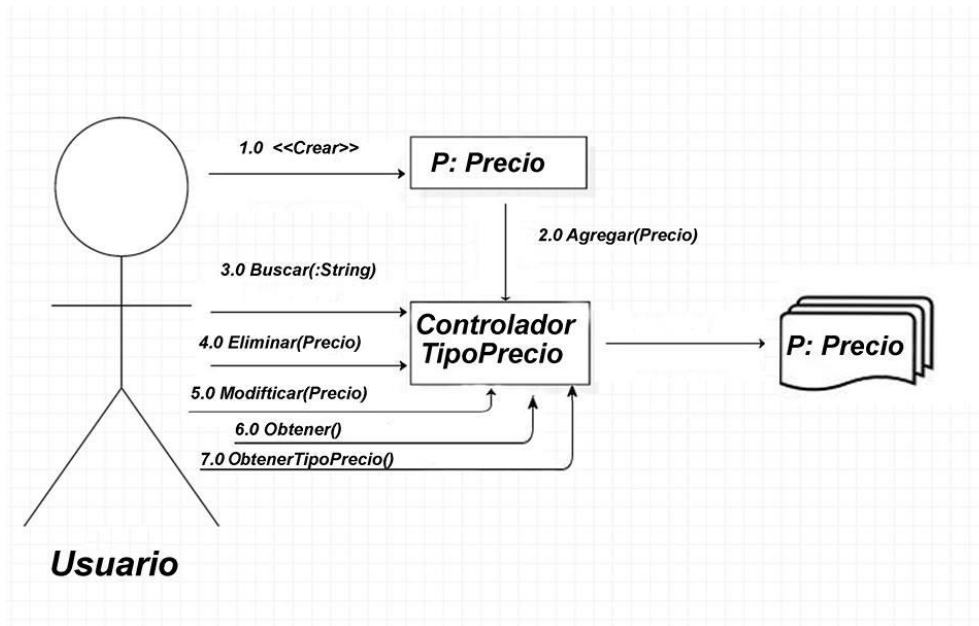


# Diagramas de colaboración de Sistema de Ventas version 2.0

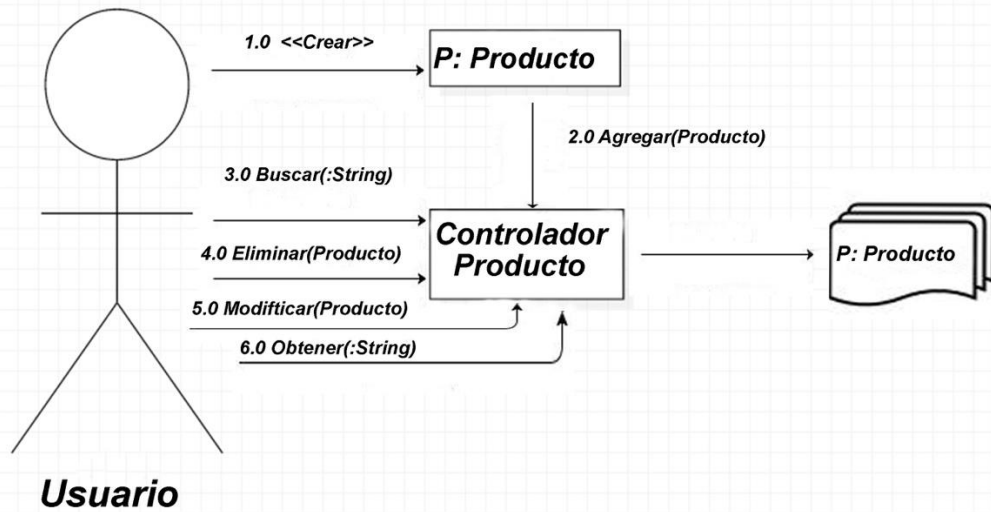
## Administrar Parámetros



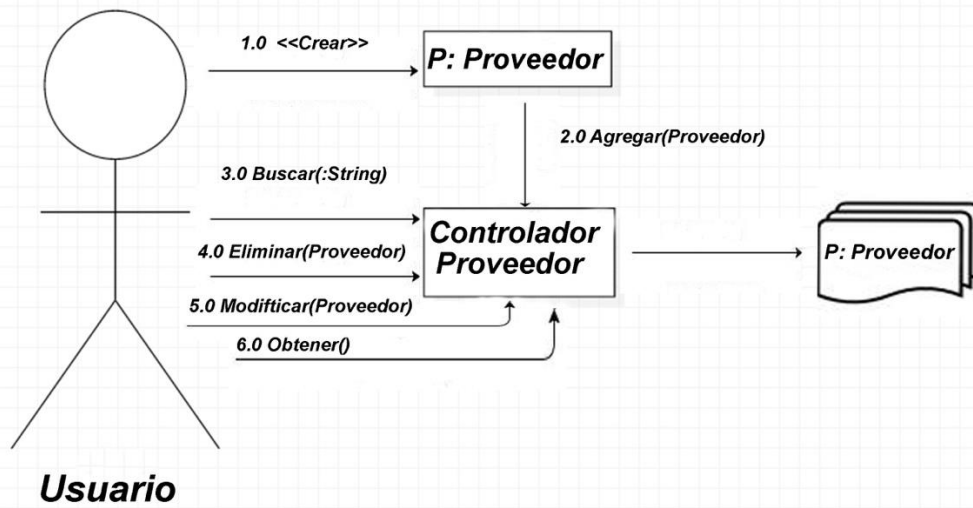
## Administrar Precios



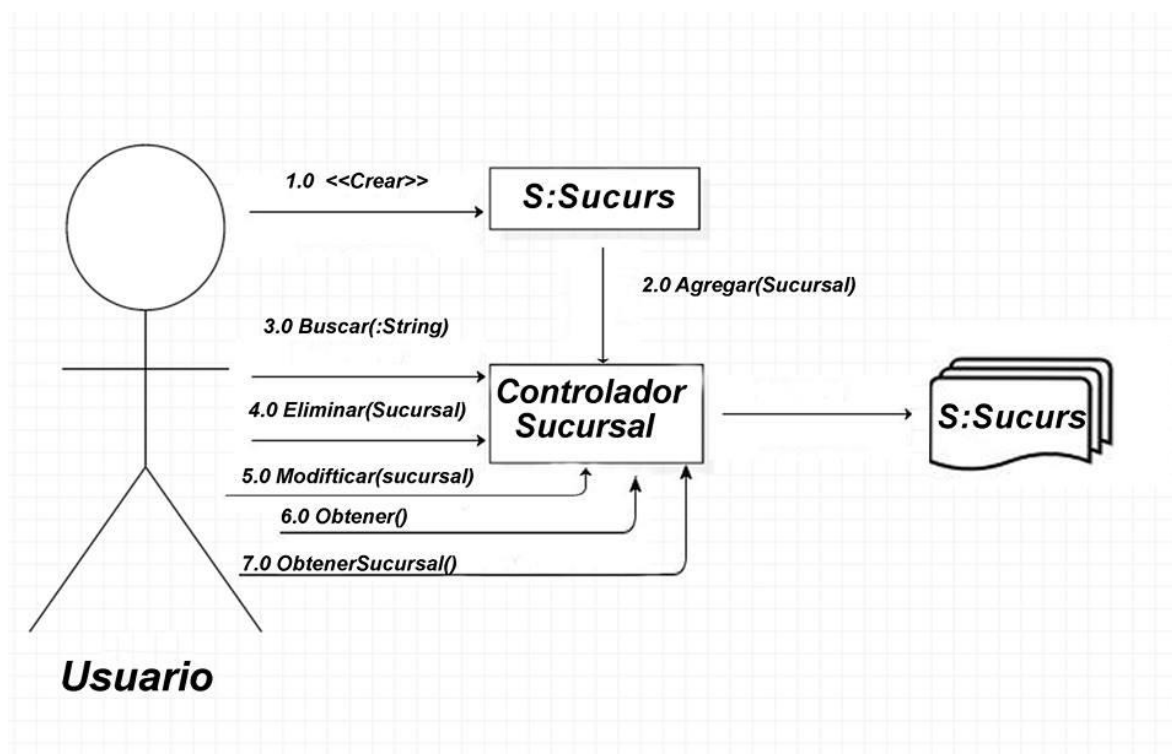
## Administrar Productos



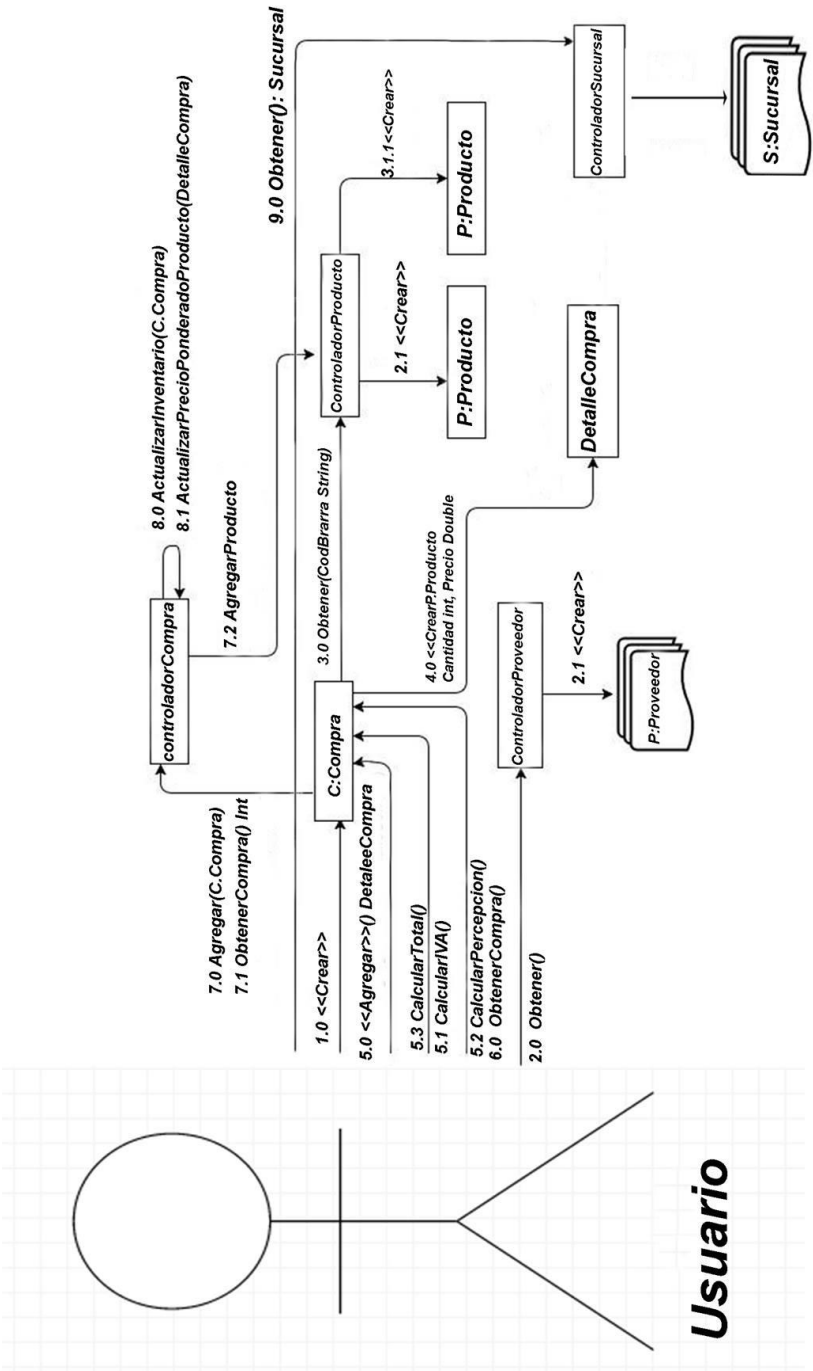
## Administrar Proveedores



## Administrar Sucursales

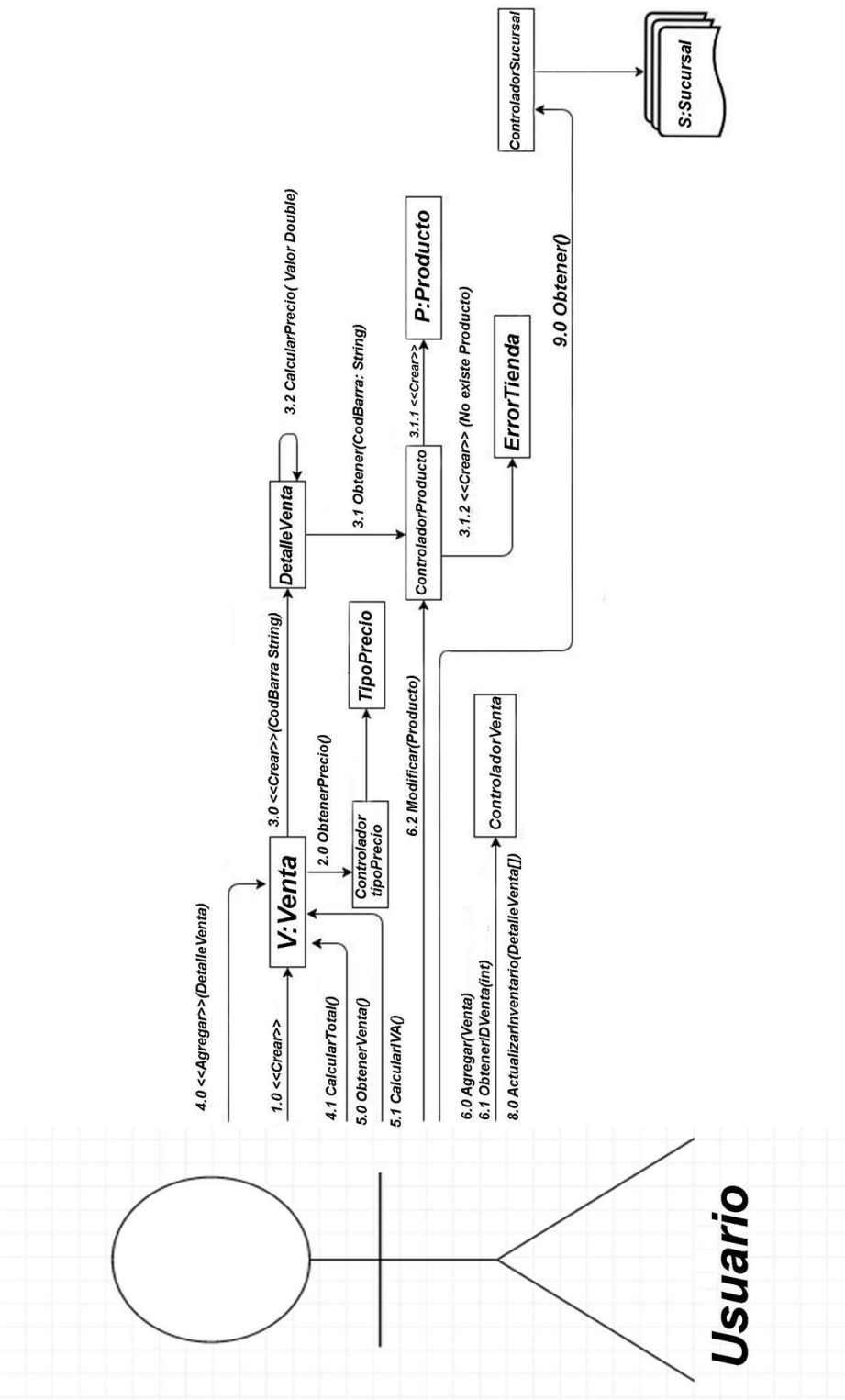


Registrar Compra

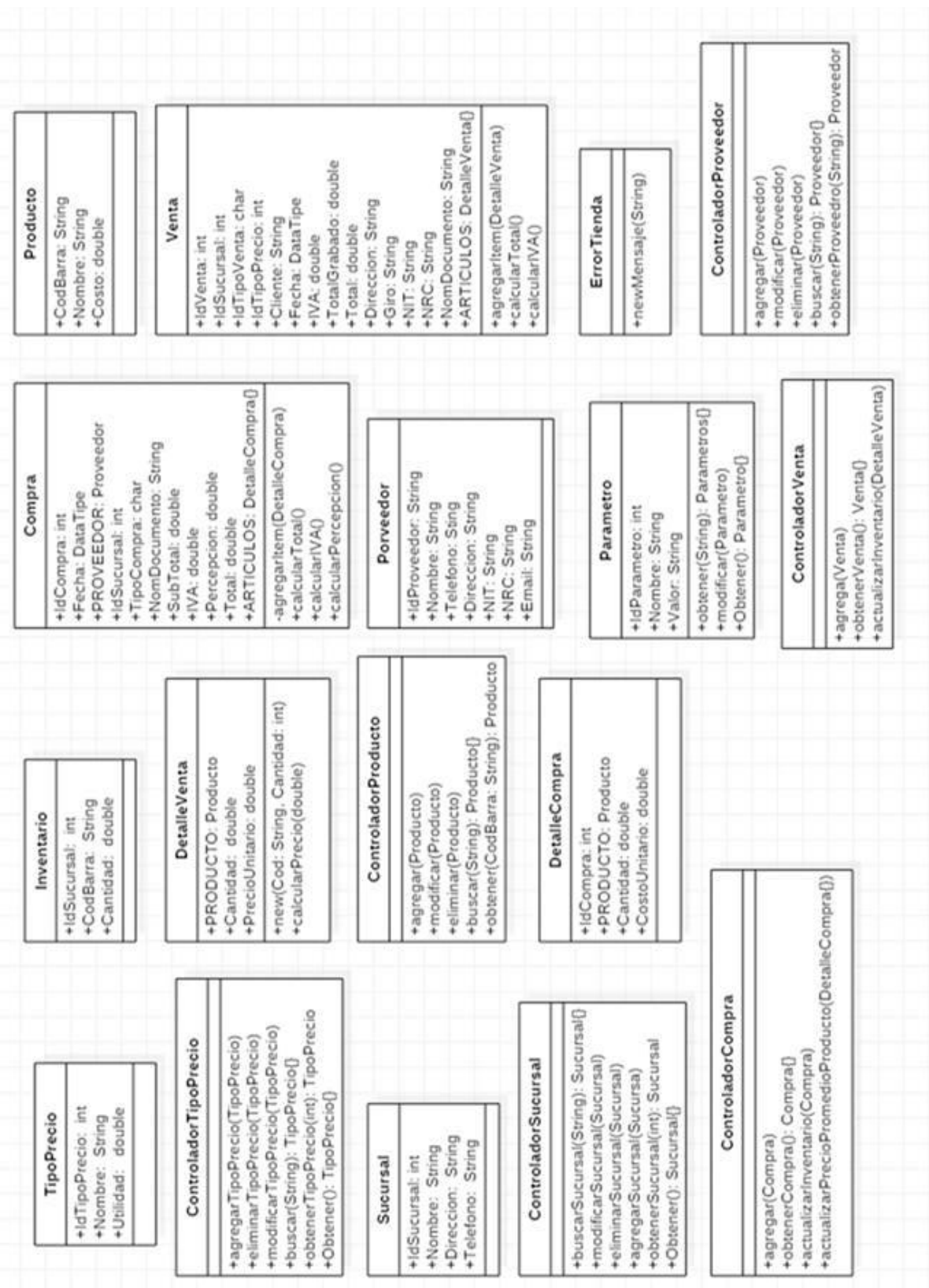




Registrar Venta



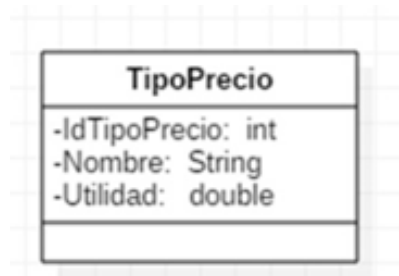
# Diagrama de clases de Sistema de Ventas version 2.0



## Descripción del Diagrama de Clases de Sistema de Ventas versión 2.0

### TIPOPRECIO

**Descripción:** Esta clase se utilizará para la creación de objetos del tipo tipoprecio, los cuales tendrán un identificador, un nombre y un porcentaje de utilidad.

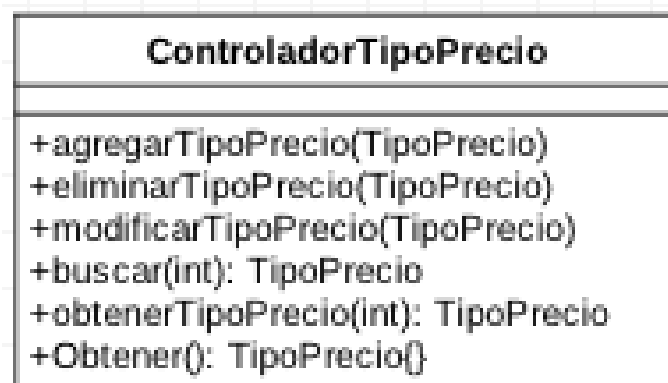


#### Métodos:

No contiene métodos.

### TIPOPRECIO CONTROLADOR

**Descripción:** En esta clase tenemos aquellos métodos que necesitamos para realizar tareas administrativas con los tipos de precio, tales como agregar, eliminar, modificar, buscar y obtener.

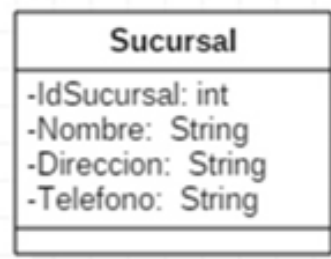


Métodos	
Nombre:	AgregarTipoPrecio(:TipoPrecio)
Parámetros:	Un objeto TipoPrecio.
Devuelve:	Nada
Estático:	Sí
Descripción:	Este método realiza la tarea de agregar un tipo precio a la base.
Nombre:	ModificarTipoPrecio(:TipoPrecio)
Parámetros:	Un objeto TipoPrecio.
Devuelve:	Nada
Estático:	Sí
Descripción:	Modifica la información de un tipo de precio existente, el usuario selecciona cual será el tipo de precio a modificar y el sistema utiliza este método para realizar la tarea.
Nombre:	EliminarTipoPrecio(:TipoPrecio)
Parámetros:	Un objeto TipoPrecio.
Devuelve:	Nada
Estático:	Sí
Descripción:	Elimina un tipo de precio de la base de datos..
Nombre:	Buscar(int):TipoPrecio
Parámetros:	Un String.
Devuelve:	Un tipo de precio
Estático:	Sí
Descripción:	El usuario busca a través de un ID y el sistema utiliza este método para devolver al usuario una lista de resultados de tipos de precio
Nombre:	ObtenerTipoPrecio(int):TipoPrecio
Parámetros:	Un ID de tipo de precio
Devuelve:	Un Tipo de Precio
Estático:	Sí
Nombre:	ObtenerTipoPrecio():TipoPrecio[]
Parámetros:	Nada

Devuelve:	Una colección de objetos de Tipo de Precio
Estático:	Sí
Descripción:	Obtiene todos los tipos de precio registrados en el sistema.

## SUCURSAL

**Descripcion:** Esta clase se utilizara para crear objetos del tipo sucursal los cuales tendrán un identificador, un nombre, una dirección y un teléfono.

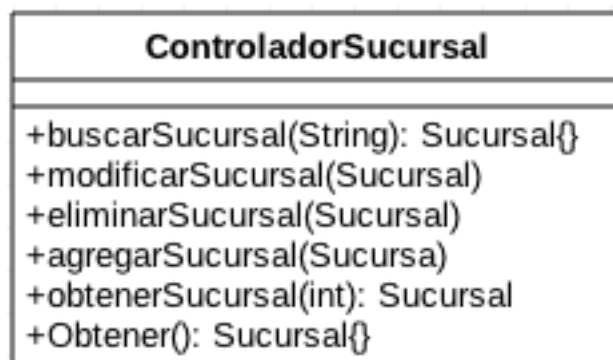


### Métodos:

No contiene métodos.

## CONTROLADOR SUCURSAL

**Descripción:** En esta clase tenemos aquellos métodos que necesitamos para realizar tareas administrativas con las sucursales, tales como agregar, eliminar, modificar, buscar y obtener.

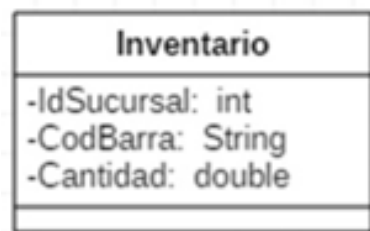


Métodos	
Nombre:	AgregarSucursal(:Sucursal)
Parámetros:	Un objeto Sucursal.
Devuelve:	Nada
Estático:	Sí
Descripción:	Este método realiza la tarea de agregar una sucursal a la base.
Nombre:	ModificarSucursal(:Sucursal)
Parámetros:	Un objeto Sucursal
Devuelve:	Nada
Estático:	Sí
Descripción:	Modifica la información de una sucursal existente, el usuario selecciona cual será la sucursal a modificar y el sistema utiliza este método para realizar la tarea.
Nombre:	EliminarSucursal(:Sucursal)
Parámetros:	Un objeto Sucursal.
Devuelve:	Nada
Estático:	Sí
Descripción:	Elimina una sucursal de la base de datos..
Nombre:	BuscarSucursal(:String):Sucursal[]
Parámetros:	Un String.
Devuelve:	Una lista de Sucursales
Estático:	Sí
Descripción:	El usuario busca a través de un criterio y el sistema utiliza este método para devolver al usuario una lista de resultados de Sucursales
Nombre:	ObtenerSucursal(int):Sucursal
Parámetros:	Un ID de sucursal.
Devuelve:	Una Sucursal identificada por el ID
Estático:	Sí
Descripción:	El usuario busca a través de un ID de una sucursal y el sistema utiliza este método para devolver al usuario la información .
Nombre:	Obtener():Sucursal[]

Parámetros:	Nada
Devuelve:	Una lista de todas las sucursales
Estático:	Sí
Descripción:	El usuario recibe la lista de todas las sucursales registradas en el sistema.

## INVENTARIO

**Descripción:** contiene los datos necesarios para actualizar el inventario de un producto según la sucursal seleccionada.

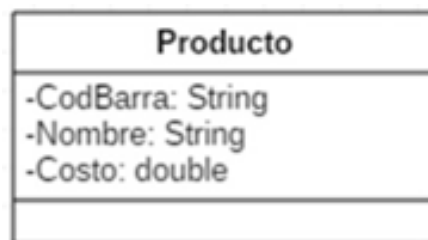


Metodos:

No contiene métodos.

## PRODUCTO

**Descripción:** Esta clase se utilizará para la creación de objetos del tipo producto, los cuales tendrán un código de barra, un nombre y costo.



**Métodos:**

No contiene métodos.

## PRODUCTO CONTROLADOR

**Descripción:** En esta clase tenemos aquellos métodos que necesitamos para realizar tareas administrativas con los productos, tales como agregar, buscar o eliminar productos.



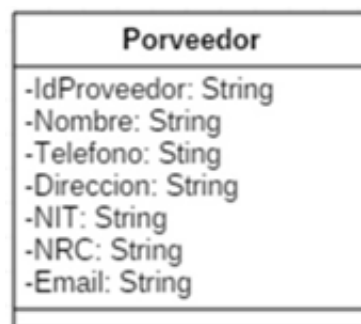
Métodos	
Nombre:	Agregar(:Producto)
Parámetros:	Un objeto Producto.
Devuelve:	Nada
Estático:	Sí
Descripción:	Este método realiza la tarea de agregar productos a la base. Cuando se realiza una compra internamente el sistema crea un objeto Producto el cual es agregado a la correspondiente sucursal seleccionada.
Nombre:	Modificar(:Producto)
Parámetros:	Un objeto Producto.
Devuelve:	Nada
Estático:	Sí
Descripción:	Modifica la información de un producto existente, el usuario selecciona cual será el producto a modificar y en la sucursal seleccionada y el sistema utiliza este método para realizar la tarea.
Nombre:	Eliminar(:Producto)
Parámetros:	Un objeto Producto.
Devuelve:	Nada
Estático:	Sí
Descripción:	Elimina un producto de la sucursal especificada.
Nombre:	Buscar(:String):Producto[]
Parámetros:	Un String.



Devuelve:	Una lista de productos
Estático:	Sí
Descripción:	El usuario busca a través de un criterio y el sistema utiliza este método para devolver al usuario una lista de resultados de productos.
Nombre:	Obtener(CodBarra:String):Producto
Parámetros:	Un código de barras.
Devuelve:	Un producto
Estático:	Sí
Descripción:	El usuario busca a través del código de barras y sistema utiliza este método para devolver al usuario la información del producto.

## PROVEEDOR

**Descripción:** En esta clase tenemos la información necesaria de todos los proveedores, los cuales tendrán un Identificador, un nombre, un teléfono, dirección, NIT, NRC y Email del proveedor.

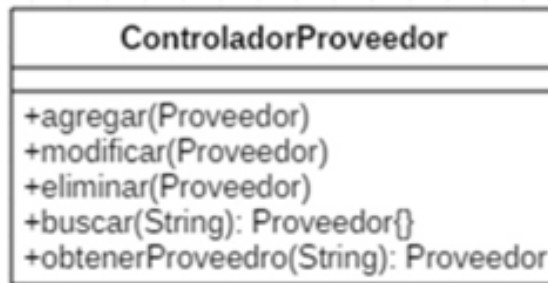


### Métodos:

No contiene métodos.

## PROVEEDOR CONTROLADOR

**Descripción:** En esta clase se tienen los métodos que son necesarios para realizar tareas administrativas con los proveedores, tales como agregar, modificar, eliminar u obtener un proveedor.

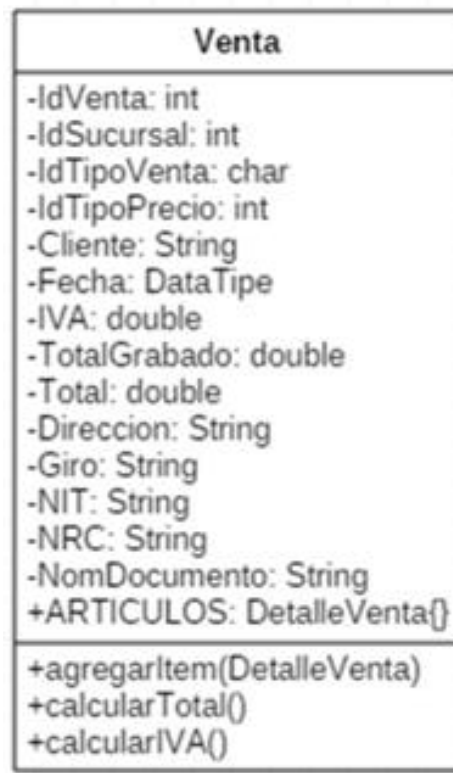


Métodos	
Nombre:	Agregar(:Proveedor)
Parámetros:	Un objeto Proveedor.
Devuelve:	Nada
Estático:	Sí
Descripción:	Este método realiza la tarea de registrar un nuevo Proveedor en la base de datos.
Nombre:	Modificar(:Proveedor)
Parámetros:	Un objeto Proveedor.
Devuelve:	Nada
Estático:	Sí
Descripción:	Modifica en la base de datos la información del Proveedor editado por el usuario.
Nombre:	Eliminar(:Proveedor)
Parámetros:	Un objeto Proveedor.
Devuelve:	Nada
Estático:	Sí
Descripción:	Elimina un Proveedor de la base de datos.

Nombre:	Buscar(:String):Proveedor[]
Parámetros:	Un String.
Devuelve:	Una lista de Proveedores
Estático:	Sí
Descripción:	El usuario busca a través de un criterio y el sistema utiliza este método para devolver al usuario una lista de los Proveedores encontrados.
Nombre:	Obtener(): Proveedor[]
Parámetros:	Ninguno.
Devuelve:	Una lista de todos los Proveedores
Estático:	Sí
Descripción:	Extrae la información de todos los proveedores registrados en la base de datos.
Nombre:	ObtenerProveedor(String): Proveedor
Parámetros:	Un string de búsqueda
Devuelve:	Un objeto proveedor
Estático:	Sí
Descripción:	Obtiene un objeto proveedor según el parámetro de búsqueda.

## VENTA

**Descripción:** En esta clase se tiene aquellos métodos que son necesarios para agregar productos a una lista de venta, obtener sus precios, obtener el impuesto y calcularlos en un total para registrar una venta con sus respectivos detalles.

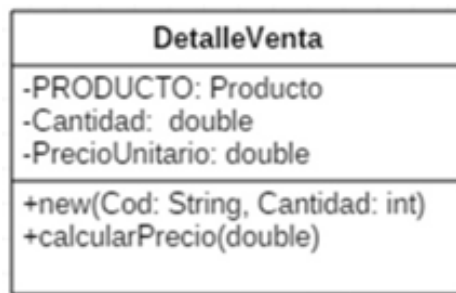


Métodos	
Nombre:	AgregarItem(:DetalleVenta)
Parámetros:	Ninguno.
Devuelve:	Nada.
Estático:	No
Descripción:	Agrega un elemento que es un objeto DetalleVenta, con todos los detalles de un producto, a la lista de la venta que se esté efectuando.

Nombre:	CalcularTotal()
Parámetros:	Ninguno
Devuelve:	Nada
Estático:	No
Descripción:	Calcula el monto total que el cliente pagará por una venta realizada.
Nombre:	CalcularIVA()
Parámetros:	Ninguno
Devuelve:	Nada
Estático:	No
Descripción:	Calcula el monto total de impuesto que el cliente pagará por una venta realizada.

## DETALLE VENTA

**Descripción:** Esta clase contiene los datos y métodos necesarios para crear un objeto que consta de un producto y sus detalles, la cantidad de dicho producto, calcular el precio de éste y poder realizar el cálculo de lo que el cliente pagará.

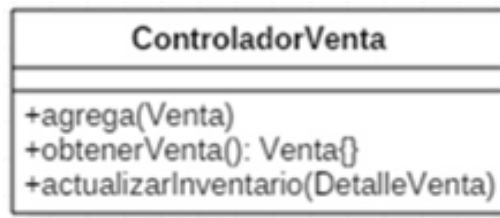


Métodos	
Nombre:	New(CodBarra: String, Canidad:Integer)
Parámetros:	Codigo de barras (String) y cantidad (Integer).
Devuelve:	Nada.
Estático:	No
Descripción:	Crea un nuevo detalle de venta con un producto y una cantidad.

Nombre:	CalcularPrecio(Double)
Parámetros:	Un cantidad de tipo Double
Devuelve:	Nada
Estático:	No
Descripción:	Calcula el Precio del detalle de compra.

## VENTA CONTROLADOR

**Descripción:** En esta clase se tienen los métodos necesarios para registrar las ventas realizadas.



Métodos	
Nombre:	Agregar (:Venta)
Parámetros:	Un objeto Venta.
Devuelve:	Nada
Estático:	Sí
Descripción:	Este método realiza la tarea de agregar una nueva venta a la base de datos.
Nombre:	ObtenerVenta():Venta
Parámetros:	Nada
Devuelve:	Un objeto Venta
Estático:	Sí
Descripción:	Obtiene desde la base de datos el objeto de la última venta efectuada.

Nombre:	ActualizarInventario(:DetalleVenta[])
Parámetros:	Una lista de objetos Detalleventa.
Devuelve:	Nada
Estático:	Sí
Descripción:	Actualiza la información de los productos en existencia en la sucursal de la venta a realizar en la base de datos..

## COMPRA

**Descripción:** En esta clase se detalla toda la información para la creación de un objeto compra y los métodos necesarios para el registro y control de una compra realizada.

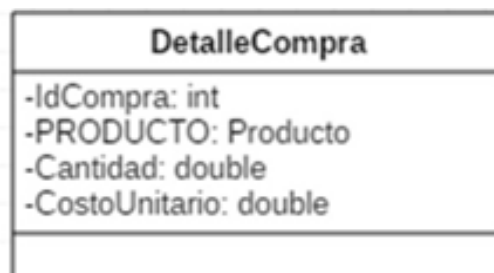


Métodos	
Nombre:	CalcularTotal()
Parámetros:	Ninguno.
Devuelve:	Nada.
Estático:	NO
Descripción:	Calcula el monto total a pagar por una

	compra realizada.
Nombre:	AgregarItem(:DetalleCompra)
Parámetros:	Un objeto de tipo DetalleCompra
Devuelve:	Nada
Estático:	NO
Descripción:	Agrega un artículo con sus detalles a través de un objeto DetalleCompra a la lista de objetos de tipo DetalleCompra de la Compra en proceso.
Nombre:	CalcularIVA()
Parámetros:	Ninguno
Devuelve:	Nada
Estático:	No
Descripción:	Calcula el monto total de impuesto IVA que se pagará por una compra realizada.
Nombre:	CalcularPercepcion ()
Parámetros:	Ninguno
Devuelve:	Nada
Estático:	No
Descripción:	Calcula el monto total de impuesto de percepción que se pagará por una compra realizada.

## DETALLE COMPRA

**Descripción:** La clase DetalleCompra contiene los datos necesarios para crear un objeto que consta de un producto, la cantidad de dicho producto y el costo de éste.



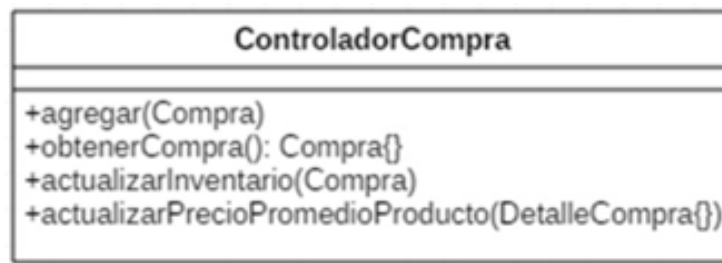


**Métodos:**

No contiene métodos.

**COMPRA CONTROLADOR**

**Descripción:** Esta clase se contiene los métodos necesarios para registrar nuevas compras en la base de datos, actualizar las existencias de productos y también la información de sus precios.



Métodos	
Nombre:	Agregar (:Compra)
Parámetros:	Un objeto Compra.
Devuelve:	Nada
Estático:	Sí
Descripción:	Este método agrega o registra una compra efectuada a la Base de datos, a la hora de agregar los detalles de los productos comprados, el sistema debe revisar a que sucursal se deben agregar estos productos al inventario en caso de que sean nuevos productos o actualizarlos en caso de que sean productos exsistentes

	.
Nombre:	ActualizarInventario(:Compra)
Parámetros:	Un objeto Compra
Devuelve:	Nada
Estático:	Sí
Descripción:	Actualiza las existencias de productos en la base de datos a partir de una compra nueva realizada y en la sucursal específica seleccionada.
Nombre:	ActualizarPrecioPromedioProducto(:DetalleCompra[])
Parámetros:	Una lista de objetos DetalleCompra[]
Devuelve:	Nada
Estático:	Sí
Descripción:	Actualiza los precios promedios de los productos detallados en la lista de compras y que este método recibe como parametro.
Nombre:	ObtenerCompra():Integer
Parámetros:	Ninguno.
Devuelve:	Una número entero
Estático:	Sí
Descripción:	Obtiene el objeto de la última compra registrada en la base de datos.

## PARAMETROS

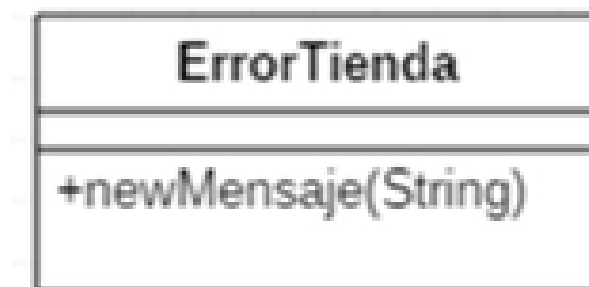
**Descripción:** La clase Parametros contiene los datos necesarios para consultar los parámetros de configuración del sistema y crear un objeto con estos.

Parametro
+IdParametro: int +Nombre: String +Valor: String
+obtener(String): Parametros{} +modificar(Parametro) +Obtener(): Parametro{}

Métodos	
Nombre:	Obtener(): Parametro[]
Parámetros:	Ninguno.
Devuelve:	Una lista de objetos de tipo Parametro
Estático:	Sí
Descripción:	Este método devuelve los parámetros almacenados en la base de datos.
Nombre:	Modificar(Parametro)
Parámetros:	Un objeto Parametro
Devuelve:	Nada
Estático:	Sí
Descripción:	Modifica el valor de un parametro ya establecido en el sistema.
Nombre:	Obtener (:String): Parametro[]
Parámetros:	Un String de búsqueda
Devuelve:	Una lista de parámetros relacionada a ese string
Estático:	Sí
Descripción:	Busca y devuelve una lista de parámetros que coincidan con el string de búsqueda.

## ERROR

**Descripción:** Esta clase se usa para el manejo de errores y excepciones.



Métodos	
Nombre:	New(Mensaje: String)
Parámetros:	Un String.
Devuelve:	Nada
Estático:	NO
Descripción:	En este método, cuando ocurre un error, se captura la información de éste.