

Programación 4

Informe del Modelo de Diseño - Diagramas de Clase de Diseño -

Grupo 3 Integrantes

Juan Ferrand
Bruno Bonfrisco
Manuel Freire
Franco Seveso

CI: 4.819.108-1
CI: 4.773.923-0
CI: 4.897.217-6
CI: 5.553.911-3

Docente: Ignacio Rafael Ferreira Urrutia

1 Análisis del impacto de los nuevos requerimientos

1.1 Modelo de dominio

En el modelo de dominio se agregó la interfaz IObserver entre Cliente y venta a domicilio al aplicar el patrón observer para permitir el envío de notificaciones cada vez que hay un cambio de estado.

También se agregaron objetos para cada uno de los estados en los que pueden estar los pedidos que a su vez serán singleton y solamente serán apuntados por cada venta a domicilio.

DtFactura pasó a ser instanciable para los casos en los que el cliente retira su pedido en el local, además se agrega una nueva función facturarRetiroLocal() que retorna un DtFactura. Esta se usará dentro del caso de uso “venta a domicilio” cuando el cliente no desee que se le asigne un repartidor

1.2

Se agregan cinco nuevos casos de uso: tres en los que el actor es el cliente, uno que es el repartidor y otro que es el admin. Se listan a continuación.

Nombre	Actualizar estado del pedido
Actores	Repartidor
Sinopsis	El caso de uso comienza cuando el repartidor desea actualizar el estado de un pedido que está pedido o en camino. Para ello el repartidor ingresa el N° de venta en el sistema del pedido que desea actualizar, y elije si quiere cancelarlo o avanzar al siguiente estado. de la cancelación al administrador y al cliente si este está suscripto.

Nombre	Ver últimas actualizaciones
Actores	Administrador
Sinopsis	El caso de uso comienza cuando el administrador desea ver las últimas actualizaciones de los pedidos. El sistema lista la última actualización de cada pedido.

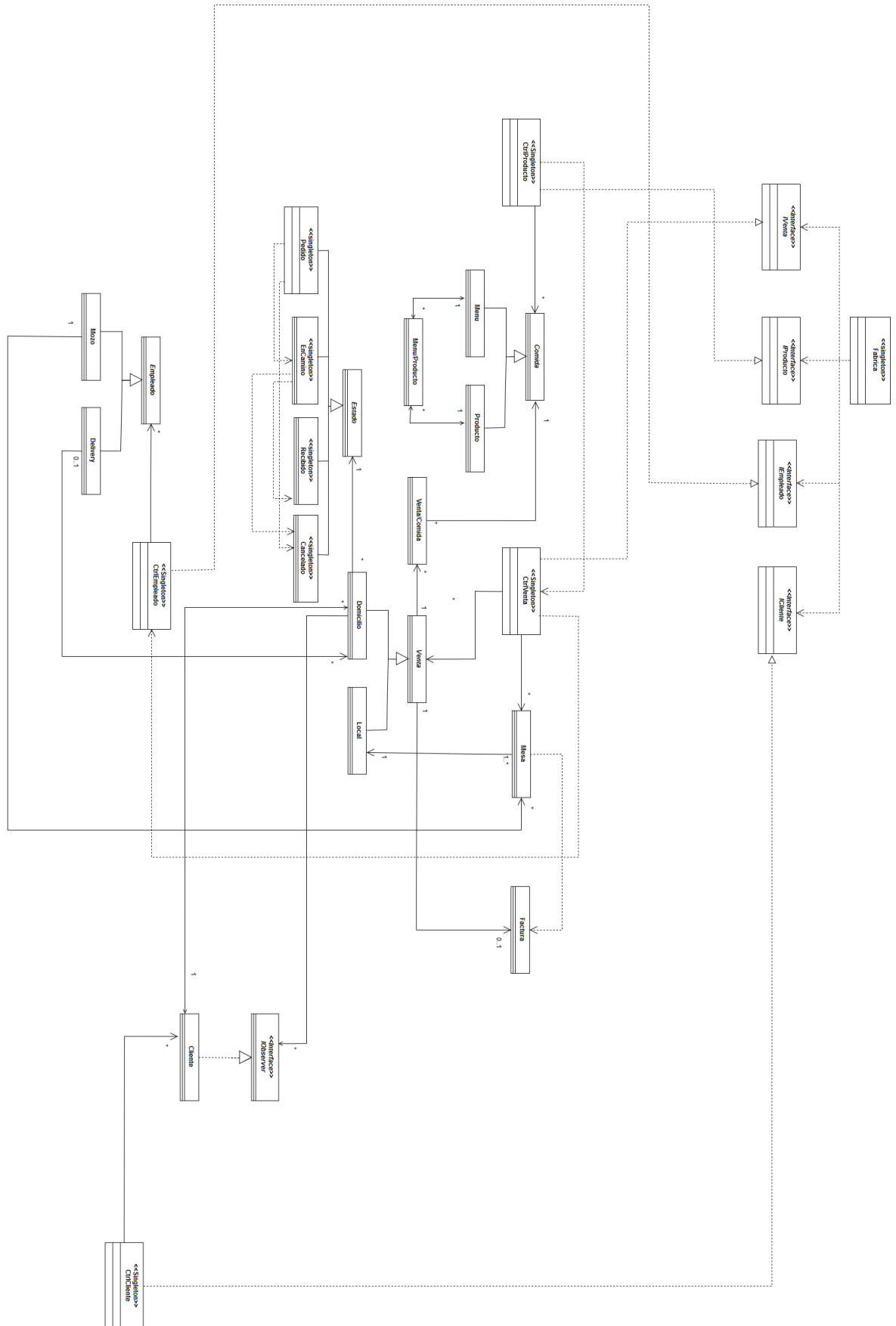
Nombre	Ver actualizaciones de los pedidos
Actores	Cliente
Sinopsis	El caso de uso comienza cuando el cliente quiere ver el historial de actualizaciones de sus pedidos. Para esto el cliente debe ingresar su mail y clave, luego el sistema devuelve todas las actualizaciones de sus pedidos.

Nombre	Suscribirse a las actualizaciones
Actores	Cliente
Sinopsis	Este caso de uso comienza cuando un cliente quiere recibir actualizaciones de sus pedidos, para esto el cliente ingresa su mail y clave en el sistema.

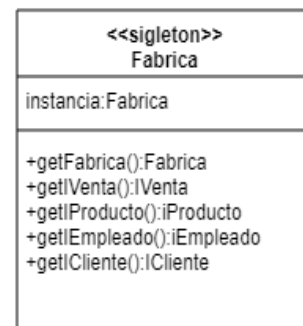
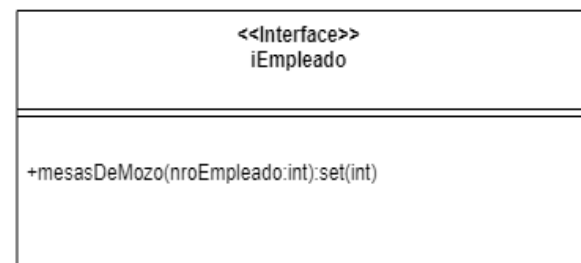
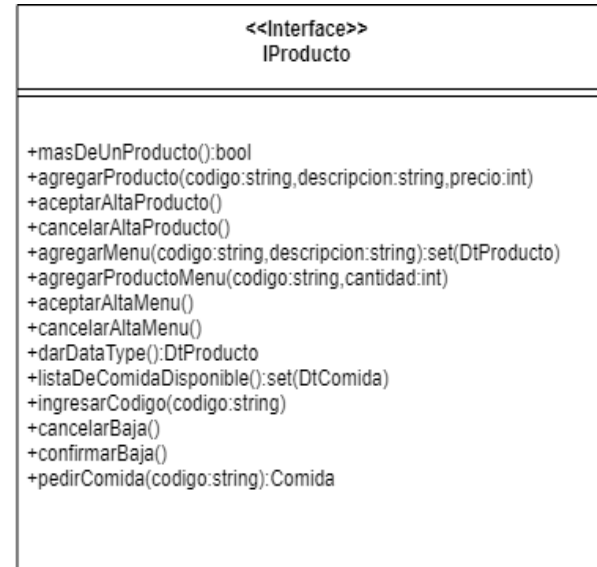
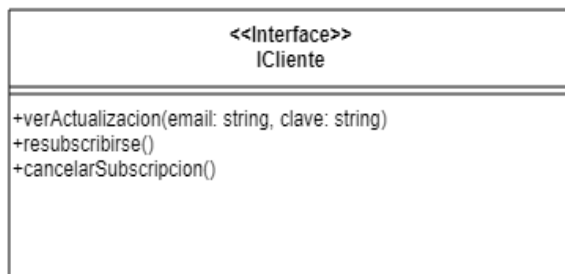
Nombre	Cancelar suscripción a las actualizaciones
Actores	Cliente
Sinopsis	Este caso de uso comienza cuando un cliente quiere dejar de recibir actualizaciones de sus pedidos, para esto el cliente ingresa su mail y clave en el sistema.

2 Diagrama de clases de diseño

2.1 Navegabilidades, dependencias, interfaces y fábricas



2.1 Atributos y funciones

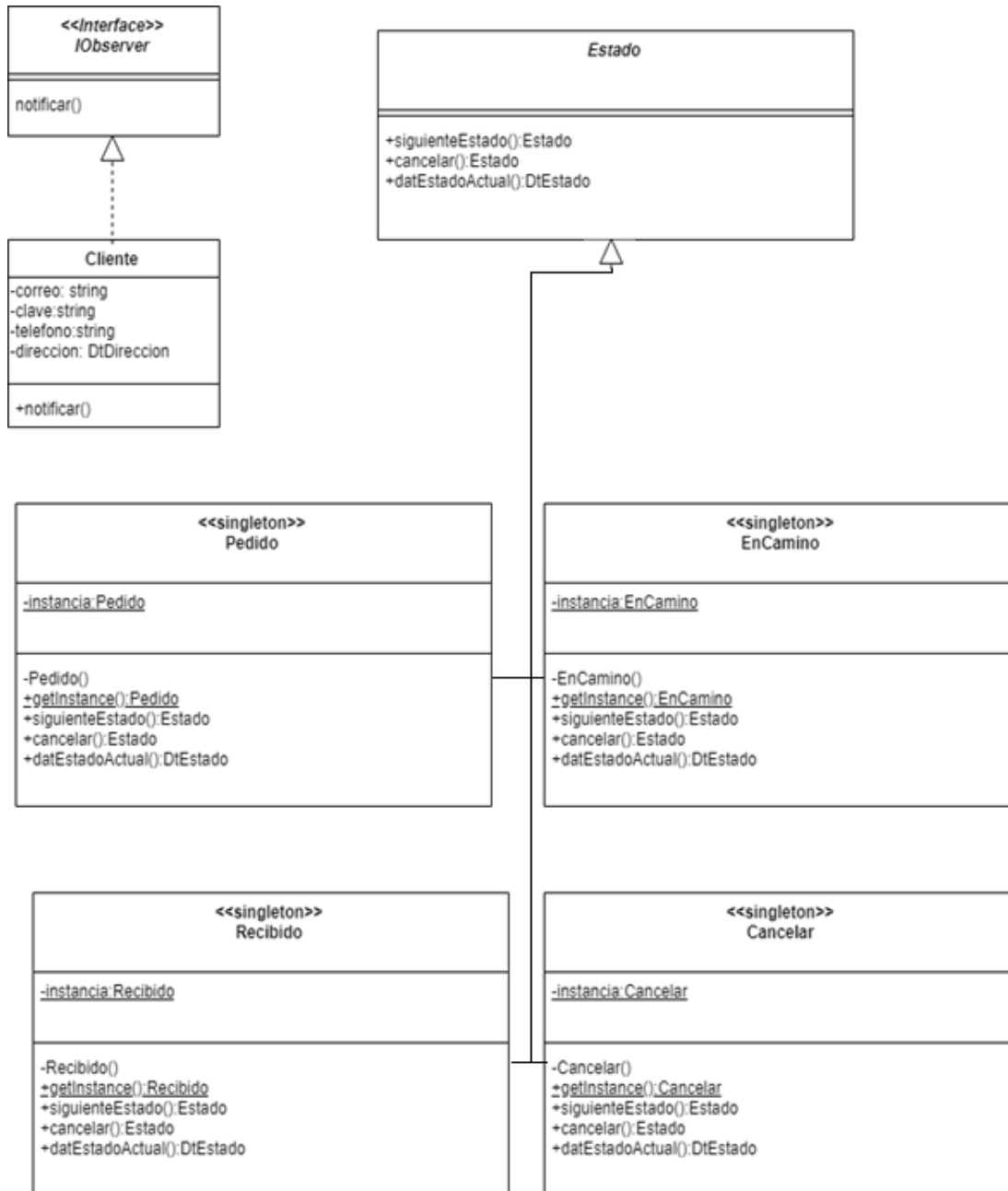


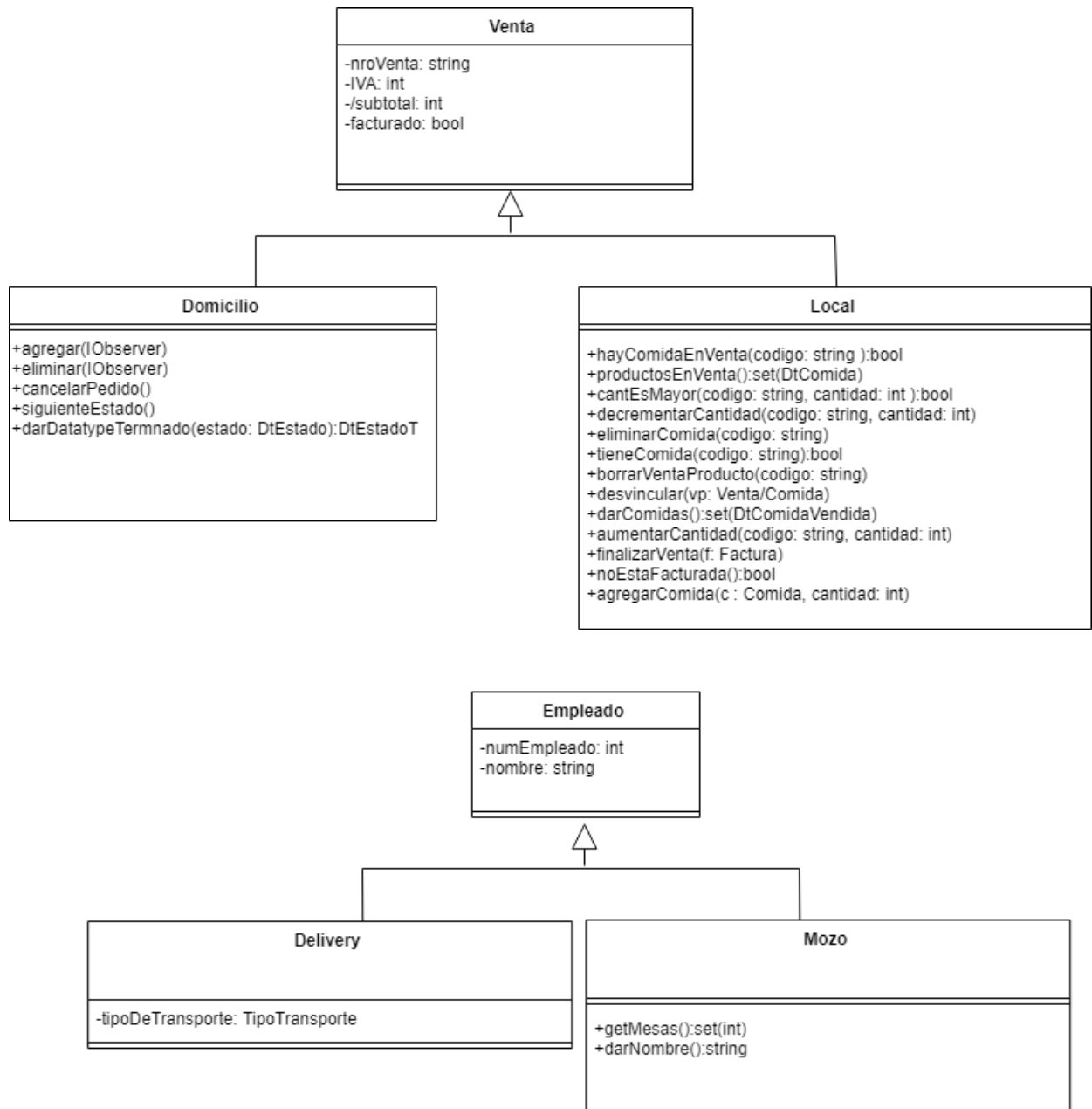
<<singleton>> CtrlCliente
-instancia: CtrlCliente
-CtrlCliente() <u>+getInstance(): CtrlCliente</u> +verActualizacion(email: string, clave: string) +rescribirse() +cancelarSubscripcion()

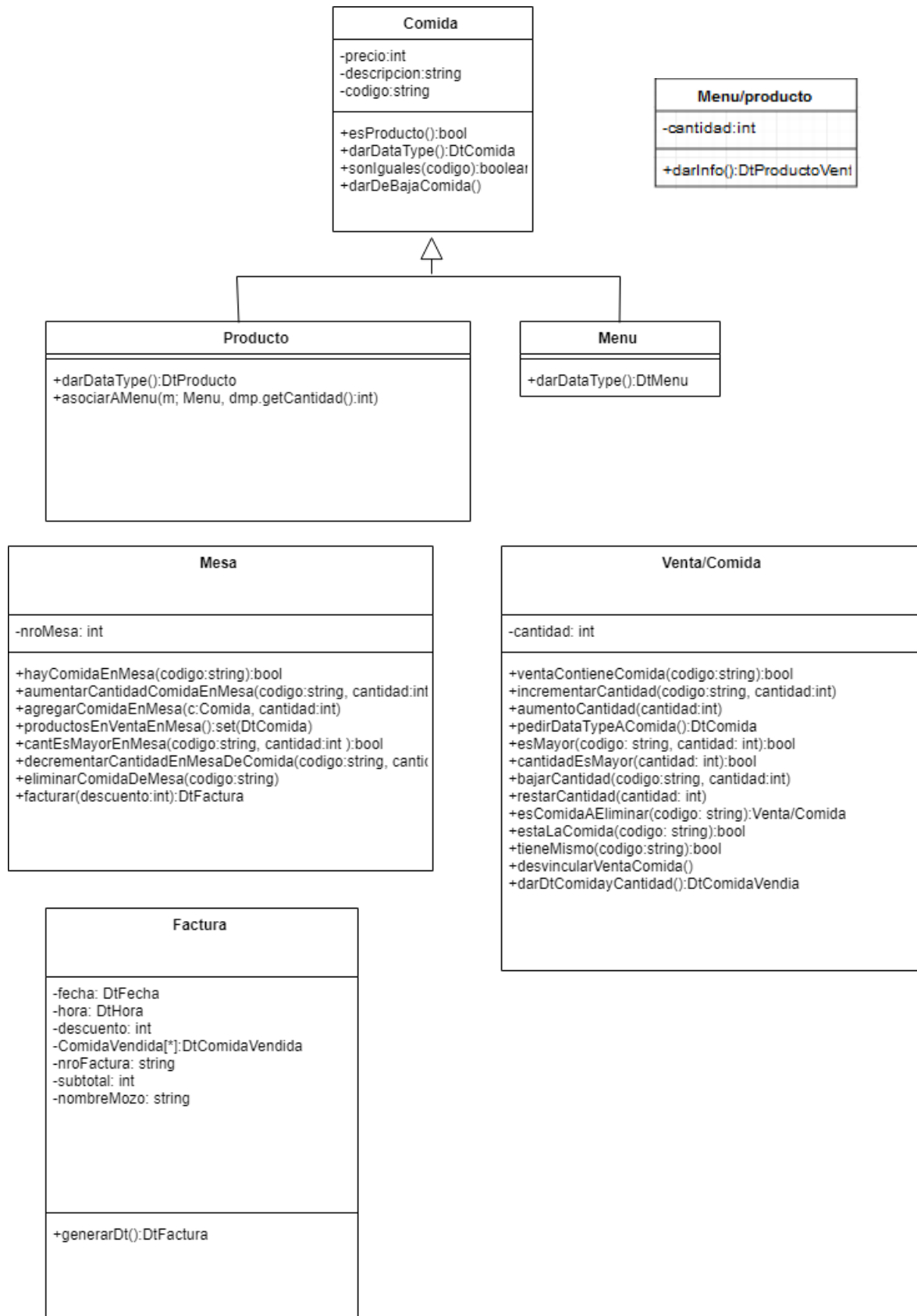
<<singleton>> CtrlEmpleado
-instancia: CtrlVenta
-CtrlEmpleado() <u>+getInstance(): CtrlEmpleado</u> +mesasDeMozo(nroEmpleado: int): set(int)

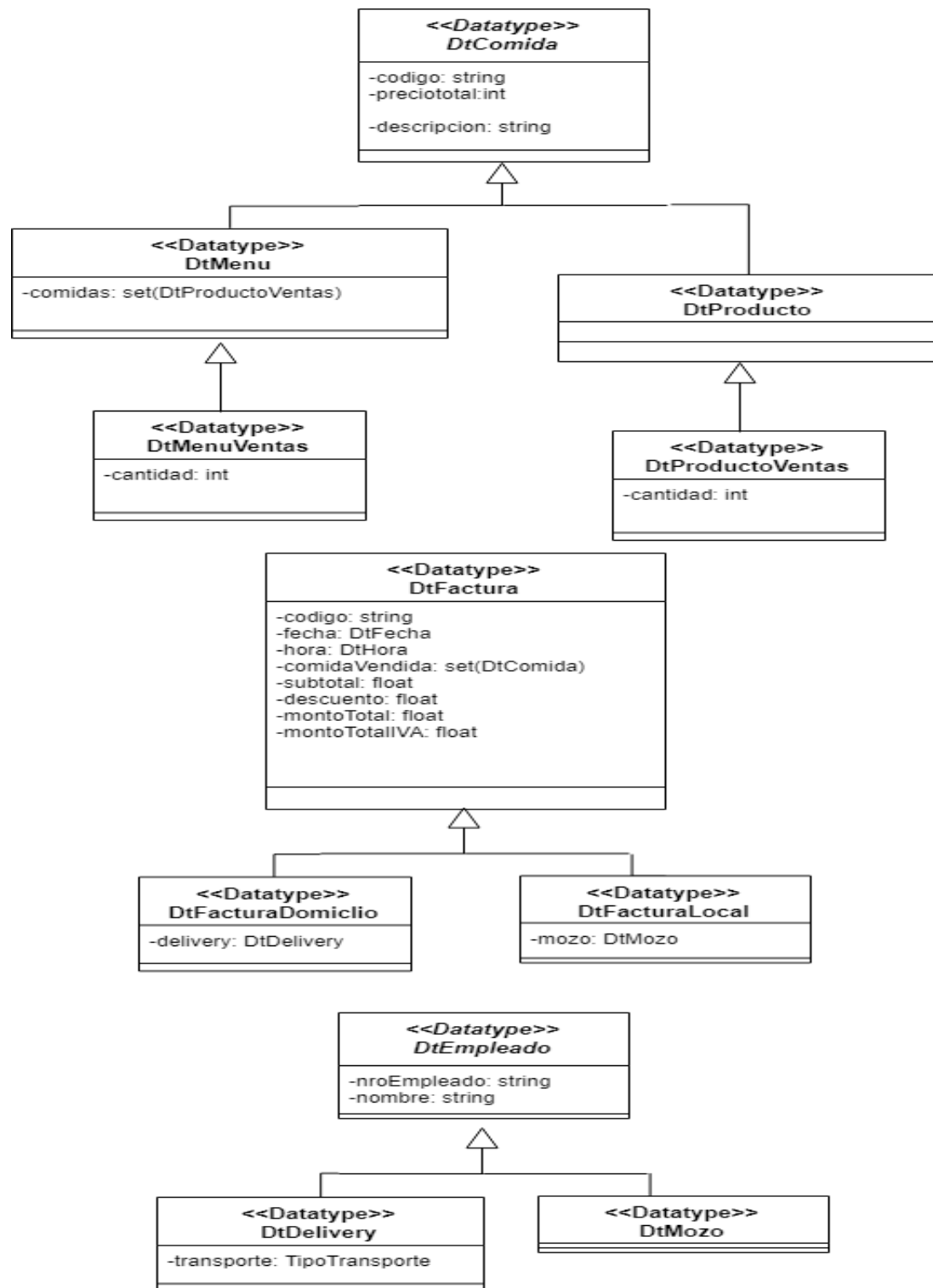
<<Singleton>> CtrlProducto
-instancia: CtrlProducto -infoProd: DtProducto -infoMenu: DtMenu -infoMP: set(DtProductoMenu) -codigo: string
-CtrlProducto() <u>+getInstance(): CtrlProducto</u> +masDeUnProducto(): bool +agregarProducto(codigo: string, descripcion: string, precio: int) +aceptarAltaProducto() +cancelarAltaProducto() +agregarMenu(codigo: string, descripcion: string): set(DtProducto) +agregarProductoMenu(codigo: string, cantidad: int) +aceptarAltaMenu() +cancelarAltaMenu() +darDataType(): DtProducto +listaDeComidaDisponible(): set(DtComida) +ingresarCodigo(codigo: string) +cancelarBaja() +confirmarBaja() +pedirComida(codigo: string): Comida

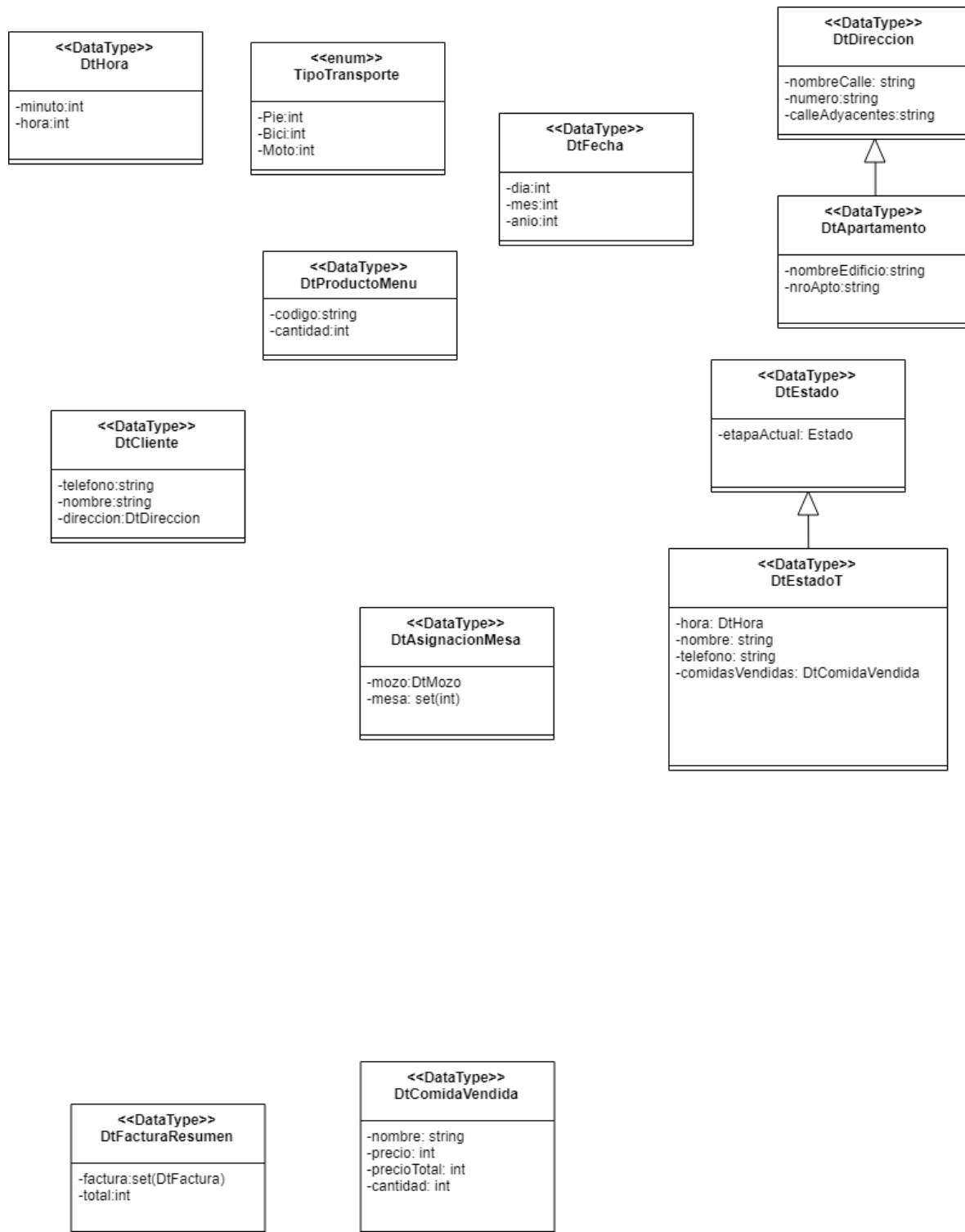
<<singleton>> CtrlVenta
-instancia: CtrlVenta -nroMesa: int -codigo: string -cantidad: int -set[*]: DtComida -mesas[*]: int -nroEmpleado: int
-CtrlVenta() <u>+getInstance(): CtrlVenta</u> +ingresarNroMesa(nroMesa: int) +seleccionarComida(codigo: string, cantidad: int) +existeComidaEnVenta(): bool +incrementarCantidad() +agregarPorPrimeraVez() +descartarAgregado() +liberarNroMesa() +productosEnVentaEnMesa(nroMesa: int): set(DtComida) +cantidadEsMayor(): bool +disminuirCantidad() +eliminarComidaDeVenta() +descartarEliminacion() +quitarComidaVenta(codigo: string) +agregarMesaAVenta(nroMesa: int) +mostrarMesasSeleccionadas(): set(int) +descartarVentasEnMesa() +mesasAsignadas(nroEmpleado: int) +confirmarVentaEnMesas() +generarFactura(nromesa: int, descuento: int): DtFactura +cancelarPedido(nroVenta: int) +ultimasActualizaciones()











3 Patrones de diseño

State

Rol	Clase
Contexto	Domicilio
Estado	Estado
Estado Concreto 1	Pedido
Estado Concreto 2	EnCamino
Estado Concreto 3	Recibido
Estado Concreto 4	Cancelado

Observer

Rol	Clase
Subject	Domicilio
Observer	IObserver
Observer Concreto 1	Cliente

Factory

Rol	Clase
Formulario	
IProveedor 1	IVenta
IProveedor 2	IProducto
IProveedor 3	IEmpleado
IProveedor 4	ICliente
Fabrica	Fabrica
IProveedor Concreto 1 de IProveedor 1	CtrlVenta
IProveedor Concreto 1 de IProveedor 2	CtrlProducto
IProveedor Concreto 1 de IProveedor 3	CtrlEmpleado
IProveedor Concreto 1 de IProveedor 4	CtrlCliente

Sigleton

Las clases son:

Pedido

EnCamino

Recibido

Cancelado

CtrlEmpleado

CtrlCliente

CtrlVenta

CtrlProducto

Fabrica

