
Sale Master

Diseño de Clases

Version 1.0

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

Historia del documento

Fecha	Versión	Descripción	Autor
18/Jun/17	1.0	Elaboración del documento	José Luis Quievedo Velásquez

Contenido

1. Diseño de clase: Boleto

- 1.1 Breve descripción
- 1.2 Responsabilidades
- 1.3 Relaciones
- 1.4 Operaciones
- 1.5 Atributos
- 1.6 Requerimientos especiales
- 1.7 Diagramas

2. Diseño de clase: Venta

- 2.1 Breve descripción
- 2.2 Responsabilidades
- 2.3 Relaciones
- 2.4 Operaciones
- 2.5 Atributos
- 2.6 Requerimientos especiales
- 2.7 Diagramas

3. Diseño de clase: ConfirmarVenta

- 3.1 Breve descripción
- 3.2 Responsabilidades
- 3.3 Relaciones
- 3.4 Operaciones
- 3.5 Atributos
- 3.6 Requerimientos especiales
- 3.7 Diagramas

4. Diseño de clase: VentaCompleta

- 4.1 Breve descripción
- 4.2 Responsabilidades
- 4.3 Relaciones
- 4.4 Operaciones
- 4.5 Atributos
- 4.6 Requerimientos especiales
- 4.7 Diagramas

5. Diseño de clase: RegistroBoletos

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

- 5.1 Breve descripción
- 5.2 Responsabilidades
- 5.3 Relaciones
- 5.4 Operaciones
- 5.5 Atributos
- 5.6 Requerimientos especiales
- 5.7 Diagramas

6. Diseño de clase: Entregados

- 6.1 Breve descripción
- 6.2 Responsabilidades
- 6.3 Relaciones
- 6.4 Operaciones
- 6.5 Atributos
- 6.6 Requerimientos especiales
- 6.7 Diagramas

7. Diseño de clase: EntregarPaquete

- 7.1 Breve descripción
- 7.2 Responsabilidades
- 7.3 Relaciones
- 7.4 Operaciones
- 7.5 Atributos
- 7.6 Requerimientos especiales
- 7.7 Diagramas

8. Diseño de clase: BDentregas

- 8.1 Breve descripción
- 8.2 Responsabilidades
- 8.3 Relaciones
- 8.4 Operaciones
- 8.5 Atributos
- 8.6 Requerimientos especiales
- 8.7 Diagramas

9. Diseño de clase: RegistroEmpleado

- 9.1 Breve descripción
- 9.2 Responsabilidades
- 9.3 Relaciones
- 9.4 Operaciones
- 9.5 Atributos
- 9.6 Requerimientos especiales
- 9.7 Diagramas

10. Diseño de clase: Control-RegistroEmpleado

- 10.1 Breve descripción
- 10.2 Responsabilidades
- 10.3 Relaciones
- 10.4 Operaciones
- 10.5 Atributos
- 10.6 Requerimientos especiales

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

10.7 Diagramas

11. Diseño de clase: DB

- 11.1 Breve descripción
- 11.2 Responsabilidades
- 11.3 Relaciones
- 11.4 Operaciones
- 11.5 Atributos
- 11.6 Requerimientos especiales
- 11.7 Diagramas

12. Diseño de clase: Empleado-Ventas

- 12.1 Breve descripción
- 12.2 Responsabilidades
- 12.3 Relaciones
- 12.4 Operaciones
- 12.5 Atributos
- 12.6 Requerimientos especiales
- 12.7 Diagramas

13. Diseño de clase: Empleado-Conductor

- 13.1 Breve descripción
- 13.2 Responsabilidades
- 13.3 Relaciones
- 13.4 Operaciones
- 13.5 Atributos
- 13.6 Requerimientos especiales
- 13.7 Diagramas

14. Diseño de clase: Consultar_Empleado

- 14.1 Breve descripción
- 14.2 Responsabilidades
- 14.3 Relaciones
- 14.4 Operaciones
- 14.5 Atributos
- 14.6 Requerimientos especiales
- 14.7 Diagramas

15. Diseño de clase: Manejador_Consulta

- 15.1 Breve descripción
- 15.2 Responsabilidades
- 15.3 Relaciones
- 15.4 Operaciones
- 15.5 Atributos
- 15.6 Requerimientos especiales
- 15.7 Diagramas

16. Diseño de clase: RegistroPaq

- 16.1 Breve descripción
- 16.2 Responsabilidades
- 16.3 Relaciones

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

- 16.4 Operaciones
- 16.5 Atributos
- 16.6 Requerimientos especiales
- 16.7 Diagramas

17. Diseño de clase: CtrlRegistro

- 17.1 Breve descripción
- 17.2 Responsabilidades
- 17.3 Relaciones
- 17.4 Operaciones
- 17.5 Atributos
- 17.6 Requerimientos especiales
- 17.7 Diagramas

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

Diseño de Clases

1. Diseño de clase: Boleto

1.1 Breve descripción

Esta es una de las clases esenciales de nuestro sistema, ya que en ella se almacenan los datos del cliente.

1.2 Responsabilidades

Esta clase es responsable de almacenar los datos del cliente siendo estos: Nombre, Origen, Destino, Asiento, Paquetes y Costo. Cada venta debe contener a lo menos un boleto.

1.3 Relaciones

Venta(Composición): Cada objeto de la clase venta debe contener a lo menos un objeto de la clase Boleto. Su multiplicidad se da de manera uno a muchos.

1.4 Operaciones

Nuevo(Nombre, Destino, Origen, Asiento, Fecha, Hora): Esta operación funciona como constructor y crea un nuevo objeto de la clase Boleto. Recibe como argumentos los Strings necesarios para la creación del objeto.

CalcularCosto(Destino, Origen): Esta operación calcula el costo del boleto. Recibe como argumentos el lugar de origen y el lugar de destino para hacer el cálculo correspondiente.

1.5 Atributos

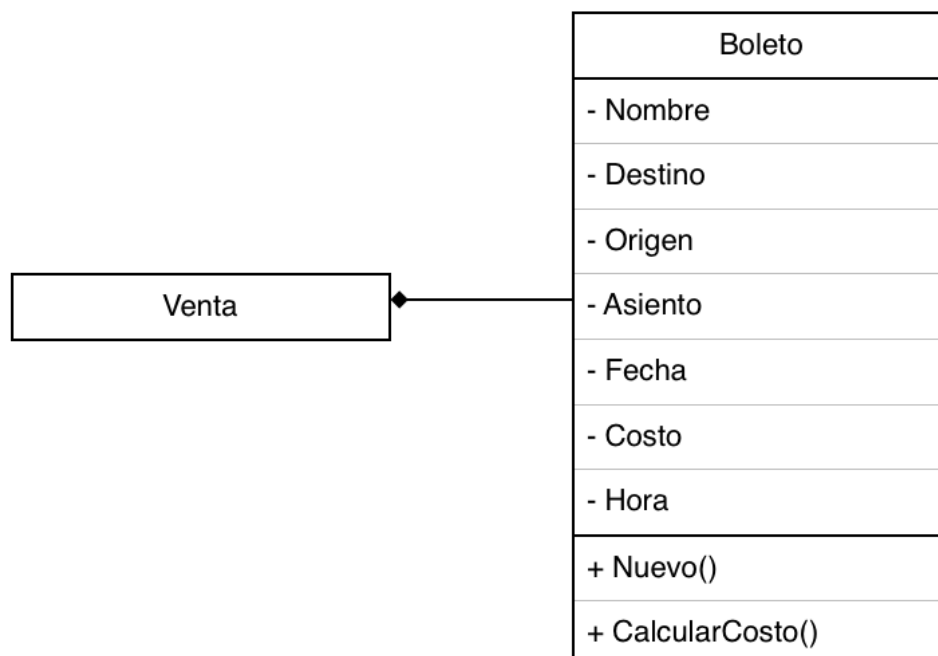
Nombre: String, cadena vacía. Este atributo guardará el nombre del cliente.

Destino: String, cadena vacía. Este atributo guardará el nombre de la ciudad destino del cliente.

Origen: String, cadena vacía. Este atributo guardará el nombre de la ciudad de origen del cliente.

Asiento: Int, 0. Este atributo guardará el número de asiento del cliente.

Fecha: String, cadena vacía. Este atributo guardará la fecha de partida del cliente.



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

2. Diseño de clase: Venta

2.1 Breve descripción

Esta clase funciona como puente Usuario – Sistema. En ella se crean las ventas de boleto.

2.2 Responsabilidades

Es la encargada de conectar al usuario con el sistema cuando se requiera hacer una venta de boleto(s).

2.3 Relaciones

Boleto(Composición): La clase Venta tiene como atributo un objeto de la clase Boleto

ConfirmarVenta(Composición): La clase Venta envía los datos a la clase ConfirmarVenta para que esta verifique los datos.

2.4 Operaciones

Nuevo(): Esta operación crea la venta y manda a verificar los datos obtenidos del boleto.

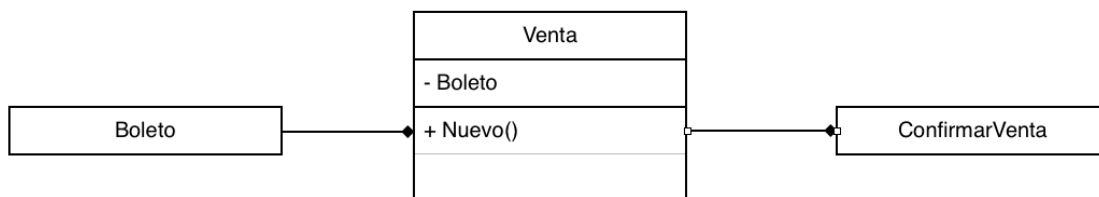
2.5 Atributos

Boleto: Objeto de la clase Boleto este atributo será enviado a la clase ConfirmarVenta

2.6 Requerimientos especiales

Ningún requerimiento especial.

2.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

3. Diseño de clase: ConfirmarVenta

3.1 Breve descripción

Esta clase es usada antes de completar la venta y da el permiso necesario para proseguir con la venta.

3.2 Responsabilidades

Esta clase es la responsable de confirmar todos los datos contenidos en el objeto de la clase venta, así también de verificar que el pago se haya realizado de manera exitosa.

3.3 Relaciones

Venta(Composición): La clase venta envía datos a esta clase para su verificación.

VentaCompleta(Dependencia): La clase ConfirmarVenta le da permisos a la clase Venta completa para la actualización de la clase RegistroBoletos.

3.4 Operaciones

Confirmar(): Esta operación revisa cada campo de Información, y verifica que estos sean correctos.

Crear(): Esta operación funciona como constructor para crear un objeto de la clase CofirmarVenta y mandarlo así a la clase VentaCompleta

Pago(): Esta operación recibe la cantidad del pago y lo almacena en el atributo Pago.

PagoRealizado(): Esta operación confirma que el pago se haya realizado de manera exitosa y que haya sido cubierto de manera satisfactoria.

3.5 Atributos

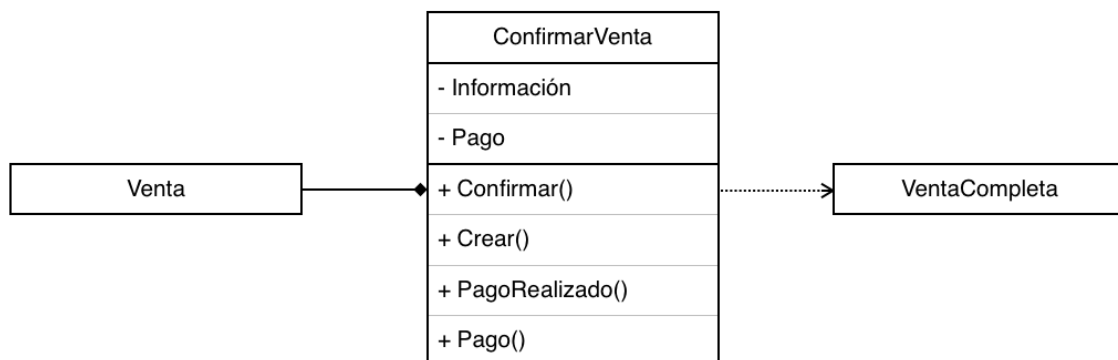
Información: Objeto de la clase Venta.

Pago: Int, 0. Este atributo almacenará el pago realizado por el cliente.

3.6 Requerimientos especiales

Ningún requerimiento especial.

3.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

4. Diseño de clase: VentaCompleta

4.1 Breve descripción

Una vez confirmada la venta como exitosa, entonces es almacenada como una venta confirmada, cumpliendo así con todos los campos.

4.2 Responsabilidades

Esta clase se encarga de otorgar el permiso para la actualización de la base de datos.

4.3 Relaciones

ConfirmarVenta(Dependencia): La clase ConfirmarVenta, manda su objeto Información a esta clase.

RegistroBoletos(Composición): Esta clase manda a la clase RegistroBoletos su objeto Información para que este sea almacenado en la base de datos.

4.4 Operaciones

Actualizar(Informacion): Por medio de esta operación manda el objeto Información a la clase RegistroBoletos

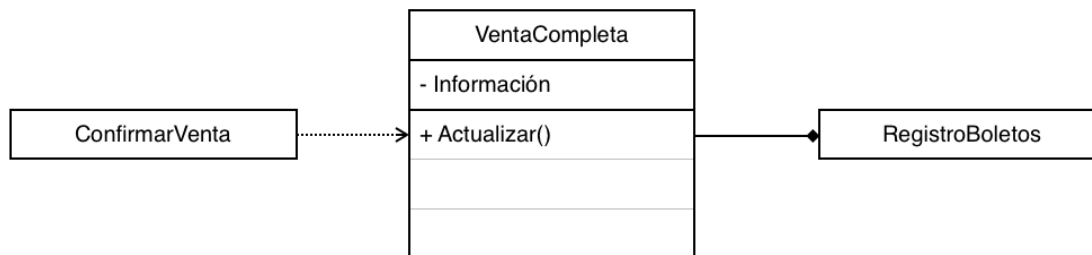
4.5 Atributos

Información: Objeto de la clase ConfirmarVenta

4.6 Requerimientos especiales

Ningún requerimiento especial.

4.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

5. Diseño de clase: RegistroBoletos

5.1 Breve descripción

Esta clase funciona como puente Sistema – Base de datos.

5.2 Responsabilidades

Esta clase se encarga actualizar la tabla en la base de datos correspondiente a los boletos.

5.3 Relaciones

VentaCompleta(Composición): La clase VentaCompleta manda los datos a actualizar y da la confirmación necesaria para que la actualización sea efectuada.

5.4 Operaciones

Nuevo(Información): Con esta función la base de datos es actualizada.

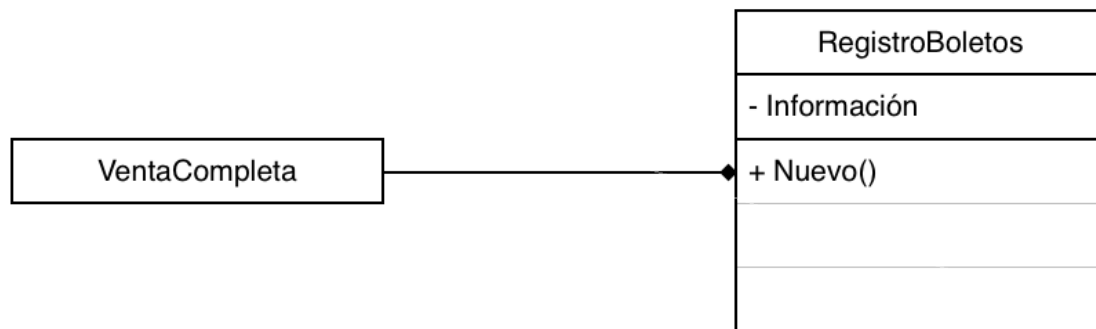
5.5 Atributos

Información: Objeto de la clase VentaCompleta.

5.6 Requerimientos especiales

Ningún requerimiento especial.

5.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

6. Diseño de clase: Entregados

6.1 Breve descripción

Esta clase contiene la información de un paquete, define sus atributos y su estado.

6.2 Responsabilidades

Es responsable de la administración de los paquetes y de su entrega.

6.3 Relaciones

EntregarPaquete(Composición): La clase Entregados manda el folio necesario a la clase EntregarPaquete para la actualización de su estado.

6.4 Operaciones

Buscar(Folio): Esta función manda a la clase EntregarPaquete el folio necesario para su actualización.

6.5 Atributos

Remitente: String, cadena vacía. Este atributo guardará el nombre del remitente.

Destinatario: String, cadena vacía. Este atributo guardará el nombre del destinatario.

Paquetes: Int, 0. Este atributo guardará el número de paquetes.

Importe: Int, 0. Este atributo guardará el precio de los paquetes.

Origen: String, cadena vacía. Este atributo guardará el nombre de la ciudad de origen del paquete

Destino: String, cadena vacía. Este atributo guardará el nombre de la ciudad de destino del paquete

Unidad: String, cadena vacía. Este atributo guardará el nombre del destinatario.

Conductor: Objeto de la clase "Empleado-Conductor".

Peso: Int, 0. Este atributo guardará el peso total del/los paquete(s)

Folio: String, cadena vacía. Este atributo guardará el folio o número de identificación del paquete.

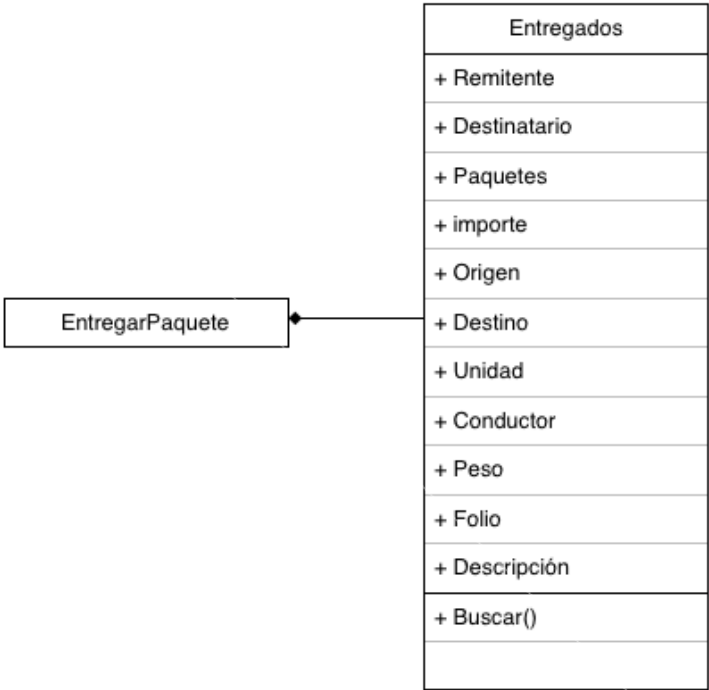
Descripción: String, cadena vacía. Este atributo guardará la descripción del paquete.

6.6 Requerimientos especiales

Ningún requerimiento especial.

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

6.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

7. Diseño de clase: EntregarPaquete

7.1 Breve descripción

Esta clase funciona como puente Sistema – Base de datos.

7.2 Responsabilidades

Enviar el folio del paquete a actualizar a la base de datos.

7.3 Relaciones

Entregados(Composición): Recibe de esta clase el folio del paquete a actualizar.

BDentregas(Composición): Envía a esta clase el paquete actualizado con el estado “Entregado”

7.4 Operaciones

ActualizarEstado(Folio): Envía a la clase BDentregas el folio del paquete a actualizar.

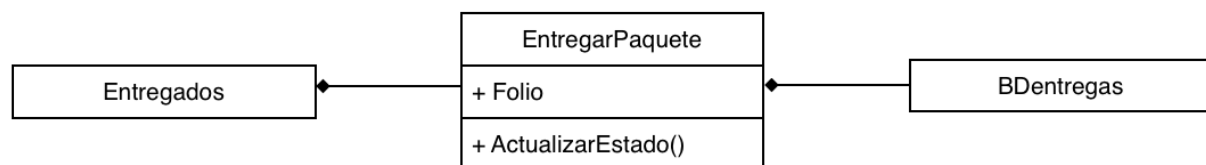
7.5 Atributos

Folio: String, cadena vacía. Este atributo guardará el folio o número de identificación del paquete a actualizar.

7.6 Requerimientos especiales

Ningún requerimiento especial.

7.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

8. Diseño de clase: BDentregas

8.1 Breve descripción

Esta clase funciona como puente Sistema – Base de datos.

8.2 Responsabilidades

Esta clase se conecta con la base de datos para actualizar estado del paquete.

8.3 Relaciones

EntregarPaquete(Composición): Recibe de esta clase el folio del paquete a actualizar.

CtrlRegistro(Composición): Recibe de la clase CtrlRegistro el paquete a registrar en la base de datos.

8.4 Operaciones

Buscar(Folio): Esta función busca dentro de la base de datos el folio del paquete.

Confirmar(): Una vez encontrado el folio del paquete cambiará Entregas a Confirmado.

8.5 Atributos

Remitente: String, cadena vacía. Este atributo guardará el nombre del remitente.

Destinatario: String, cadena vacía. Este atributo guardará el nombre del destinatario.

Paquetes: Int, 0. Este atributo guardará el número de paquetes.

Importe: Int, 0. Este atributo guardará el precio de los paquetes.

Origen: String, cadena vacía. Este atributo guardará el nombre de la ciudad de origen del paquete

Destino: String, cadena vacía. Este atributo guardará el nombre de la ciudad de destino del paquete

Unidad: String, cadena vacía. Este atributo guardará el nombre del destinatario.

Conductor: Objeto de la clase “Empleado-Conductor”

Peso: Int, 0. Este atributo guardará el peso total del/los paquete(s)

Folio: String, cadena vacía. Este atributo guardará el folio o número de identificación del paquete.

Estado: Boolean, False. Este atributo guardará el estado del paquete. (True para un estado “Entregado” y False para un estado “No Entregado”)

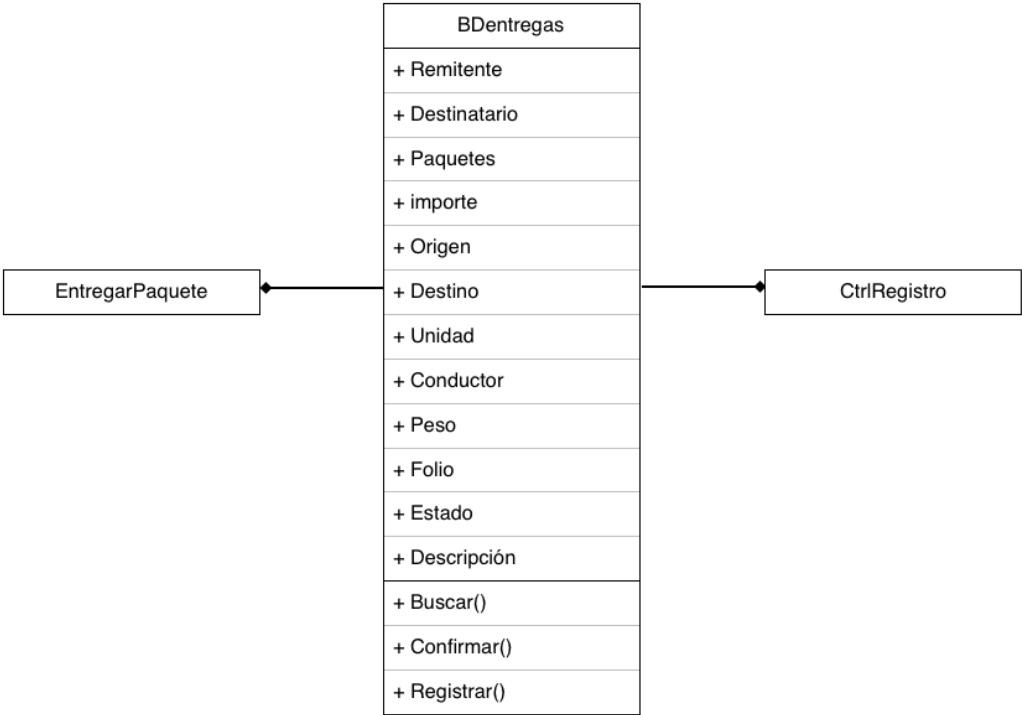
Descripción: String, cadena vacía. Este atributo guardará la descripción del paquete.

8.6 Requerimientos especiales

Ningún requerimiento especial.

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

8.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

9. Diseño de clase: RegistroEmpleado

9.1 Breve descripción

Esta clase funciona como puente Usuario – Sistema

9.2 Responsabilidades

Esta clase es responsable de la creación de los registros de empleados y de la captura de datos necesarios para los mismos.

9.3 Relaciones

Control-RegistroEmpleado(Relación): La clase RegistroEmpleado envía el registro a Control-RegistroEmpleado para su almacenamiento en la base de datos.

9.4 Operaciones

Nuevo(): Es el constructor de la clase RegistroEmpleado

ObtenerDatos(): En esta operación se obtienen los datos necesarios para llenar los atributos necesarios.

ValidarDatos(ID, Nombre, Tipo, NivelEstudio, Licencia, Edad, Domicilio, Localidad, Telefono, Sexo, FechaNacimiento, EstadoCivil, Nacionalidad, c.p): En esta operación se validan los datos obtenidos por la función ObtenerDatos(), para su almacenamiento en las variables correspondientes.

MandarDatos(): En esta operación se mandan los datos a la clase Control-RegistroEmpleado

9.5 Atributos

ID: Int, 0. Este atributo guardará el identificador del empleado.

Nombre: String, cadena vacía. Este atributo guardará el nombre del empleado.

Tipo: Boolean, False. Este atributo guardará el tipo de empleado. (False para uno de tipo Empleado-Ventas y True para uno de tipo Empleado-Conductor)

NivelEstudio: String, cadena vacía. Este atributo guardará nivel de estudio del empleado (Solo para empleados de “Ventas”).

Licencia: String, cadena vacía. Este atributo guardará el número de licencia (Solo para empleados de “Empleado-Conductor”).

Edad: Int, 0. Este atributo guardará la edad del empleado.

Domicilio: String, cadena vacía. Este atributo guardará el domicilio del empleado.

Localidad: String, cadena vacía. Este atributo guardará la localidad del empleado.

Telefono: Int, 0. Este atributo guardará el teléfono del empleado.

Sexo: Boolean, False. Este atributo guardará el sexo del empleado. (False para Hombre y True para mujer).

FechaNacimiento: String, cadena vacía. Este atributo guardará la fecha de nacimiento del empleado.

EstadoCivil: Boolean, False. Este atributo guardará el estado civil del empleado. (False para Soltero y True para Casado).

Nacionalidad: String, cadena vacía. Este atributo guardará la nacionalidad del empleado.

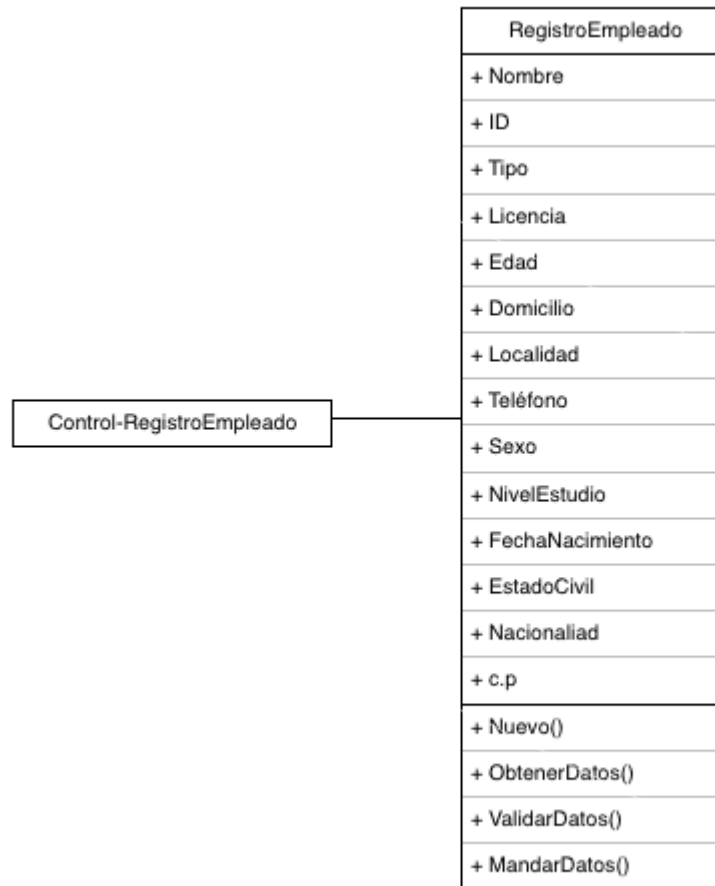
c.p: Int, 0. Este atributo guardará el código postal del empleado.

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

9.6 Requerimientos especiales

Ningún requerimiento especial.

9.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

10. Diseño de clase: Control-RegistroEmpleado

10.1 Breve descripción

Esta funciona como la clase de control de la clase RegistroEmpleado y de la clase DB

10.2 Responsabilidades

Coordina y controla los objetos de las clases RegistroEmpleado y DB

10.3 Relaciones

RegistroEmpleado(Relación): recibe de la clase RegistroEmpleado un registro para posteriormente ser enviado a la base de datos.

DB(Relación): Recibe de la clase Control-RegistroEmpleado el nuevo registro.

10.4 Operaciones

RegistroEmpleado(): Llama a la clase RegistroEmpleado.

DB(): Envía a la clase DB un objeto de la clase RegistroEmpleado

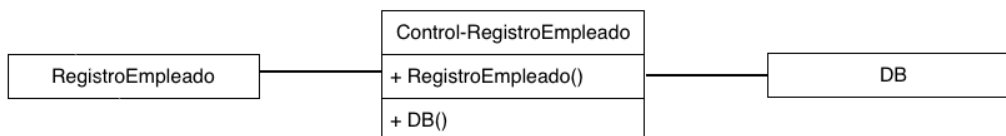
10.5 Atributos

Sin atributos.

10.6 Requerimientos especiales

Ningún requerimiento especial.

10.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

11. Diseño de clase: DB

11.1 Breve descripción

Esta clase funciona como puente Sistema – Base de Datos

11.2 Responsabilidades

Esta es la responsable de almacenar en la base de datos los registros entrantes.

11.3 Relaciones

Empleado-Ventas(Dependencia): Dependiendo del tipo de registro se mandará la información a esta clase.

Empleado-Conductor(Dependencia): Dependiendo del tipo de registro se mandará la información a esta clase.

Manejador_Consulta(Relación): Manejador_Consulta envía a la clase DB el ID referente a la consulta.

Control_RegistroEmpleado(Relación): Control_RegistroEmpleado manda a la clase DB la información necesaria.

11.4 Operaciones

RecibirDatos(Registro): Recibe un objeto de la clase RegistroEmpleado

Guarda(Registro): Envía a la clase correspondiente la información del registro.

Información(ID): Recibe una ID de un registro a buscar y regresa a la clase correspondiente la información del registro.

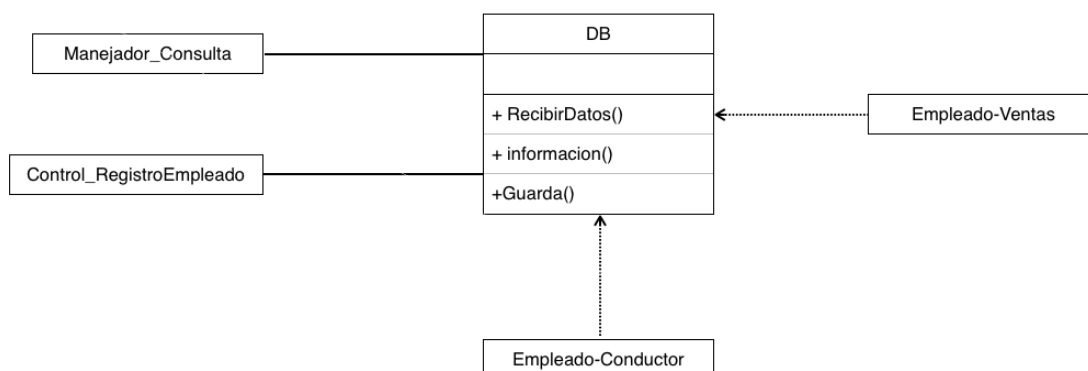
11.5 Atributos

Sin atributos.

11.6 Requerimientos especiales

Ningún requerimiento especial.

11.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

12. Diseño de clase: Empleado-Ventas

12.1 Breve descripción

Esta clase es una clase entidad que está en constante comunicación con la base de datos

12.2 Responsabilidades

Esta clase es la responsable de la creación y almacenamiento de nuestros registros de tipo Empleado de Ventas.

12.3 Relaciones

DB(Dependencia): Recibe de la clase DB la información necesaria para la requisitación de los atributos de la clase Empleado-Ventas.

12.4 Operaciones

Sin operaciones.

12.5 Atributos

ID: Int, 0. Este atributo guardará el identificador del empleado.

Nombre: String, cadena vacía. Este atributo guardará el nombre del empleado.

Tipo: Boolean, True. Este atributo guardará el tipo de empleado.

NivelEstudio: String, cadena vacía. Este atributo guardará nivel de estudio del empleado.

Edad: Int, 0. Este atributo guardará la edad del empleado.

Domicilio: String, cadena vacía. Este atributo guardará el domicilio del empleado.

Localidad: String, cadena vacía. Este atributo guardará la localidad del empleado.

Telefono: Int, 0. Este atributo guardará el teléfono del empleado.

Sexo: Boolean, False. Este atributo guardará el sexo del empleado. (False para Hombre y True para mujer).

FechaNacimiento: String, cadena vacía. Este atributo guardará la fecha de nacimiento del empleado.

EstadoCivil: Boolean, False. Este atributo guardará el estado civil del empleado. (False para Soltero y True para Casado).

Nacionalidad: String, cadena vacía. Este atributo guardará la nacionalidad del empleado.

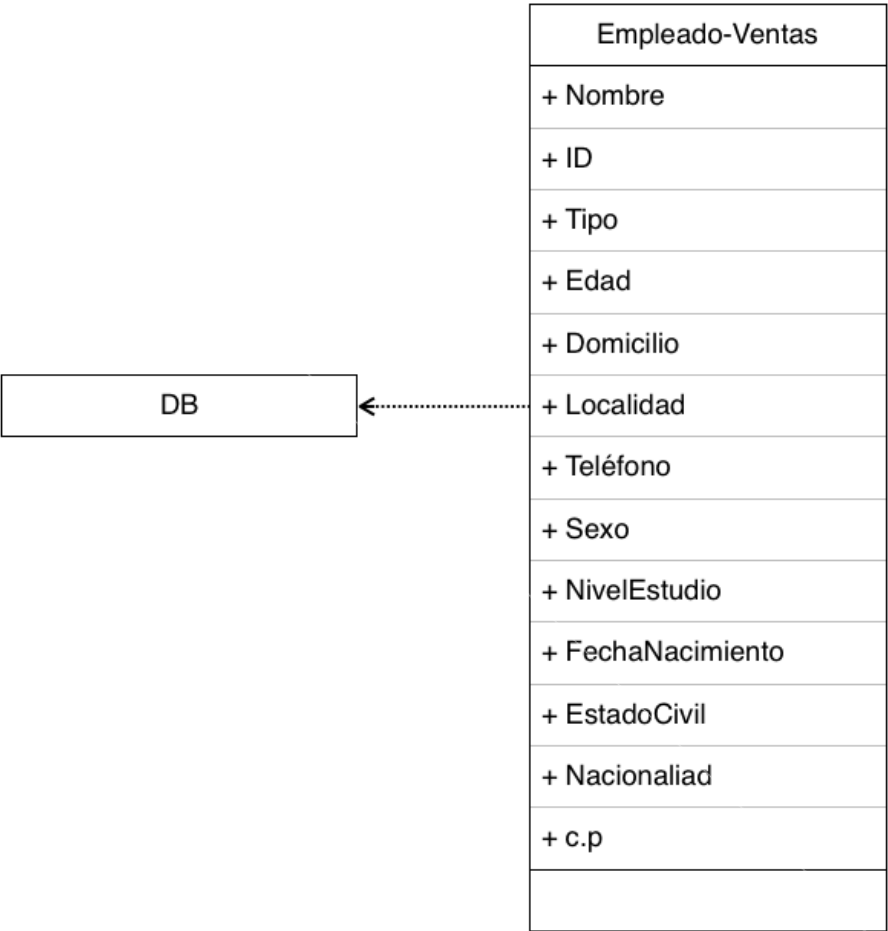
c.p: Int, 0. Este atributo guardará el código postal del empleado.

12.6 Requerimientos especiales

Ningún requerimiento especial.

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

12.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

13. Diseño de clase: Empleado-Conductor

13.1 Breve descripción

Esta clase es una clase entidad que está en constante comunicación con la base de datos

13.2 Responsabilidades

Esta clase es la responsable de la creación y almacenamiento de nuestros registros de tipo Empleado Conductor.

13.3 Relaciones

DB(Dependencia): Recibe de la clase DB la información necesaria para la requisitación de los atributos de la clase Empleado-Conductor.

13.4 Operaciones

Sin operaciones.

13.5 Atributos

ID: Int, 0. Este atributo guardará el identificador del empleado.

Nombre: String, cadena vacía. Este atributo guardará el nombre del empleado.

Tipo: Boolean, False. Este atributo guardará el tipo de empleado.

Licencia: String, cadena vacía. Este atributo guardará el número de licencia.

Edad: Int, 0. Este atributo guardará la edad del empleado.

Domicilio: String, cadena vacía. Este atributo guardará el domicilio del empleado.

Localidad: String, cadena vacía. Este atributo guardará la localidad del empleado.

Telefono: Int, 0. Este atributo guardará el teléfono del empleado.

Sexo: Boolean, False. Este atributo guardará el sexo del empleado. (False para Hombre y True para mujer).

FechaNacimiento: String, cadena vacía. Este atributo guardará la fecha de nacimiento del empleado.

EstadoCivil: Boolean, False. Este atributo guardará el estado civil del empleado. (False para Soltero y True para Casado).

Nacionalidad: String, cadena vacía. Este atributo guardará la nacionalidad del empleado.

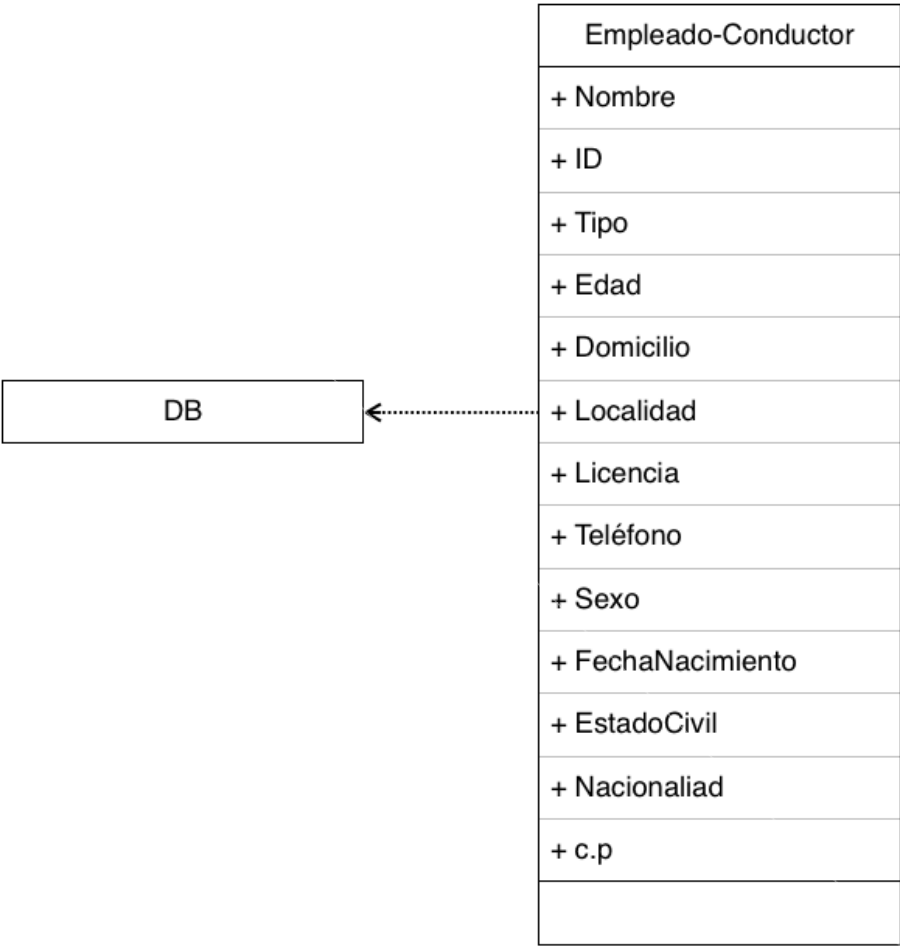
c.p: Int, 0. Este atributo guardará el código postal del empleado.

13.6 Requerimientos especiales

Ningún requerimiento especial.

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

13.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

14. Diseño de clase: Consultar_Empleado

14.1 Breve descripción

Esta clase funciona como puente Usuario – Sistema

14.2 Responsabilidades

Esta clase es la responsable de recibir la información a buscar y mostrarla en pantalla.

14.3 Relaciones

Manejador_Consulta(Relación): Manejador_consulta recibe la información a buscar.

14.4 Operaciones

MostarInfo(Empleado): Muestra en pantalla los datos de la consulta realizada.

ConsultarID(ID): Manda a la clase Manejador_Consulta el ID del registro a buscar.

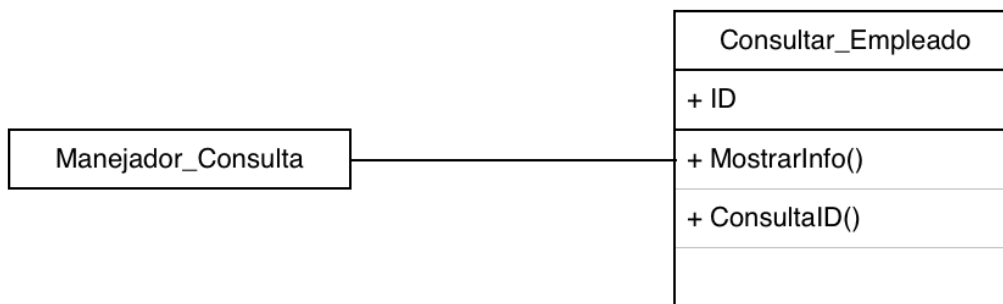
14.5 Atributos

ID: Int, 0. Este atributo guardará el identificador del empleado.

14.6 Requerimientos especiales

Ningún requerimiento especial.

14.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

15. Diseño de clase: Manejador_Con consulta

15.1 Breve descripción

Es la clase administradora de las consultas de los empleados.

15.2 Responsabilidades

Coordina y controla los objetos de las clases ConsultarEmpleado y DB

15.3 Relaciones

ConsultarEmpleado(Relación): Manejador_Con consulta recibe de la clase ConsultarEmpleado el ID a buscar

DB(Relación): Manejador_Con consulta envía a la clase DB el ID referente a la consulta.

15.4 Operaciones

RecibirDatos(ID): Esta función recibe el ID del empleado a consultar.

Consultar_Empleado(): Esta operación recibe los resultados de la búsqueda realizada.

MandarInfoP(): Esta operación manda los resultados de la búsqueda realizada y los envía a la clase Consultar_Empleado para su ejecución.

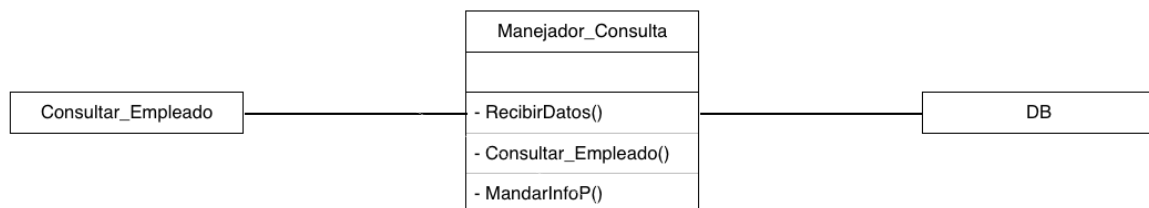
15.5 Atributos

No tiene atributos.

15.6 Requerimientos especiales

Ningún requerimiento especial.

15.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

16. Diseño de clase: RegistroPaq

16.1 Breve descripción

Esta clase funciona como puente Usuario - Sistema

16.2 Responsabilidades

Esta clase recibe datos del paquete, calcula el importe de los paquetes y verifica que todos los datos estén completos y debidamente requisitados.

16.3 Relaciones

CtrlRegistro(Composición): La clase RegistroPaquete envía a la clase CtrlRegistro un objeto Paquete a registrar en la base de datos.

16.4 Operaciones

Calcular(Paquetes, Peso): Esta operación calcula el precio total del envío del/los paquete(s).

Registrar(Remitente, Destinatario, Origen, Destino, Peso, Paquetes, Descripcion, Unidad, Importe, Estado, Folio): Esta operación recibe los resultados de la búsqueda realizada.

16.5 Atributos

Remitente: String, cadena vacía. Este atributo guardará el nombre del remitente.

Destinatario: String, cadena vacía. Este atributo guardará el nombre del destinatario.

Paquetes: Int, 0. Este atributo guardará el número de paquetes.

Importe: Int, 0. Este atributo guardará el precio de los paquetes.

Origen: String, cadena vacía. Este atributo guardará el nombre de la ciudad de origen del paquete

Destino: String, cadena vacía. Este atributo guardará el nombre de la ciudad de destino del paquete

Unidad: String, cadena vacía. Este atributo guardará el nombre del destinatario.

Conductor: Objeto de la clase "Empleado-Conductor".

Peso: Int, 0. Este atributo guardará el peso total del/los paquete(s)

Folio: String, cadena vacía. Este atributo guardará el folio o número de identificación del paquete.

Estado: Boolean, False. Este atributo guardará el estado del envío. (False para un estado "No entregado" y True para un estado "No Entregado").

Descripcion: String, cadena vacía. Este atributo guardará la descripción del paquete.

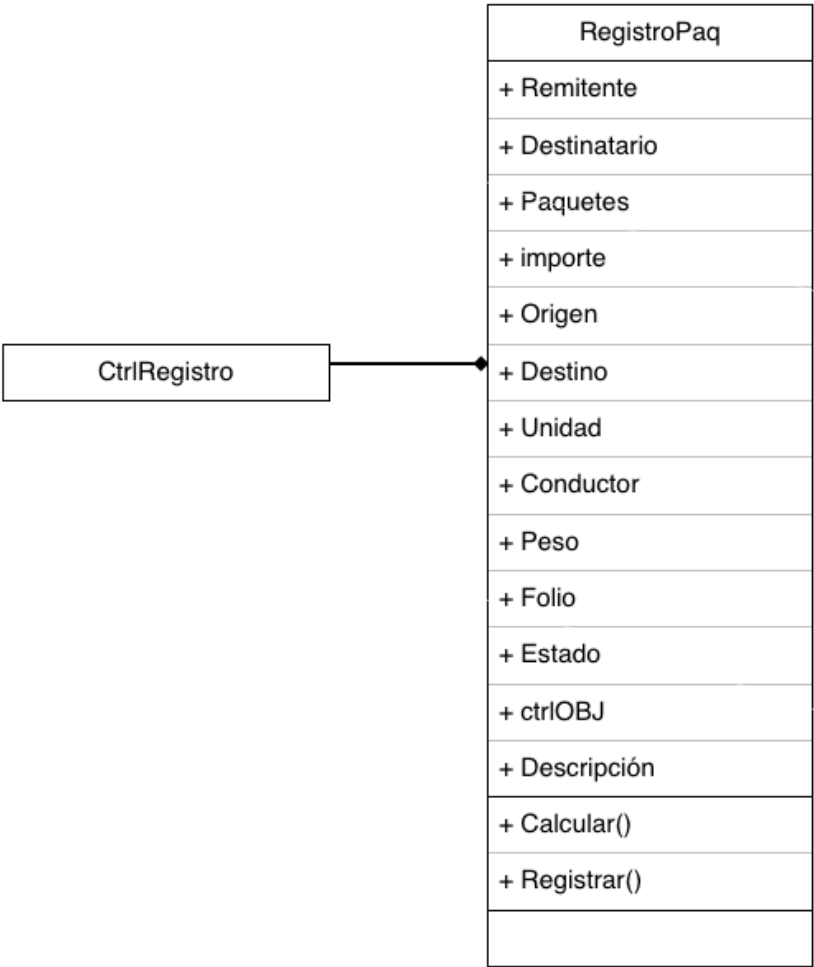
ctrlOBJ: Objeto de la clase CtrlRegistro.

16.6 Requerimientos especiales

Ningún requerimiento especial.

Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

16.7 Diagramas



Sale Master	Version: <1.0>
Diseño de Clases	Date: 18/Jun/17

17. Diseño de clase: CtrlRegistro

17.1 Breve descripción

Esta clase funciona como puente Sistema – Base de Datos.

17.2 Responsabilidades

Esta clase es la responsable de enviar el objeto de tipo RegistroPaq a la base de datos para su almacenamiento.

17.3 Relaciones

RegistroPaquete(Composición): La clase CtrlRegistro recibe de la clase RegistroPaquete el objeto que tendrá que ser enviado a la clase DBentregas.

DBentregas(Composición): La clase CtrlRegistro envía a la clase DBentregas el objeto a almacenar dentro de la base de datos.

17.4 Operaciones

RegistroPaq(paqueteOBJ)

17.5 Atributos

PaqueteOBJ: Objeto de la clase RegistroPaquete

17.6 Requerimientos especiales

Ningún requerimiento especial.

17.7 Diagramas

