
Sale Master

Diseño de Clases

Version 3.0

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

Historia del documento

Fecha	Versión	Descripción	Autor
18/Jun/17	1.0	Elaboración del documento	José Luis Quievedo Velásquez
01/Julio/17	2.0	Corrección de diagramas y de operaiones.	José Luis Quevedo Velásquez
03/Julio/17	3.0	Integración de diagramas y corrección de clases.	José Luis Quevedo V y Reyes Hernández Nabí.

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: Ventas

- 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: Paquetería

- 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: Empleado

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

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

5. Diseño de clase: Listar_Empleados

- 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: Empleado_Servicio

- 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: PaqueteríaDatos

- 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: Suburban

- 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: Acceder

- 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: Login

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

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

- 10.6 Requerimientos especiales
- 10.7 Diagramas

11. Diseño de clase: Menu

- 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: Ticket

- 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: SaleMaster

- 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: Conexion

- 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: Boleto_servicios

- 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: Empleado_servicios

- 16.1 Breve descripción
- 16.2 Responsabilidades

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

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

17. Diseño de clase: Suburban_servicios

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

18. Diseño de clase: Paquetes_servicios

- 18.1 Breve descripción
- 18.2 Responsabilidades
- 18.3 Relaciones
- 18.4 Operaciones
- 18.5 Atributos
- 18.6 Requerimientos especiales
- 18.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/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.

1.3 Relaciones

No tiene.

1.4 Operaciones

GetDestino(): Método que devuelve el Destino.

GetFecha(): Método que devuelve la Fecha.

GetHora(): Método que devuelve la Hora

getID_boleto(): Método que devuelve el ID_boleto.

getID_suburban(): Método que devuelve ID_suburban.

getNo_Asiento(): Método que devuelve No_Asiento.

GetPasajero(): Método que devuelve Pasajero.

setDestino(java.lang.String destino): Método que asigna el destino al atributo Destino.

setFecha(java.lang.String fecha): Método que asigna la fecha al atributo Fecha

setHora(java.lang.String hora): Método que asigna la hora al atributo Hora.

setID_suburban(java.lang.Integer id_suburban): Método que asigna el numero de identificación al atributo ID_suburban.

setNo_Asiento(java.lang.Integer asiento): Método que asigna el numero de asiento al atributo No_Asiento.

setPasajero(java.lang.String pas): Método que asigna el nombre del pasajero al atributo Pasajero.

1.5 Atributos

id_boleto: El atributo id_boleto, es un entero que representa el numero con el que se identificara cada boleto.

Asiento: El atributo asiento, es un entero que representa el numero del asiento al que esta asociado el boleto

destino: El atributo destino, es un String que representa el nombre del lugar al que se tiene como destino.

Fecha: El atributo fecha, es un String con la fecha de salida de la suburban en la que se el pasajero viajara.

Hora: El atributo hora, es un String con la hora en la que la suburban partirá.

id_suburban: El atributo id_suburban, es un entero que representa el numero con el que se identifica cada suburban

nombre: El atributo nombre, es un String que representa el nombre del pasajero.

1.6 Requerimientos especiales

Ningún requerimiento especial.

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

1.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

2. Diseño de clase: Ventas

2.1 Breve descripción

Esta clase se encarga de almacenar los datos obtenidos en la venta de un boleto. Así también de su procesamiento.

2.2 Responsabilidades

Almacenar y procesar los boletos vendidos.

2.3 Relaciones

Menú (relación): Las ventas se efectúan a través de la clase Menú.

Ticket (relación): Cada venta realizada genera un elemento de la clase Ticket.

2.4 Operaciones

CargarSub(Suburban subs): Método que carga los datos de cada suburban(El estado actual de los asientos).

Limpiar(): Método que coloca cadenas vacías en los campos de Name y Seat.

2.5 Atributos

Fecha: El atributo fecha, es un String con la fecha de salida de la suburban en la que se el pasajero viajara.

Hora: El atributo hora, es un String con la hora en la que la suburban partirá.

Name: El atributo name, es un String con con el nombre del cliente.

Nboletos: El atributo Nboletos, es un entero con el número de boletos comprados.

Seat: El atributo seat, es un entero con el número de asiento comprado.

2.6 Requerimientos especiales

Ningún requerimiento especial.

2.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

3. Diseño de clase: Paquetería

3.1 Breve descripción

Esta clase funciona como puente Usuario – Sistema permitiéndole realizar las acciones pertinentes a la sección de “Paquetería”.

3.2 Responsabilidades

Mostrar en pantalla la ventana para realizar las actividades correspondientes a paquetería.

3.3 Relaciones

Menú (relación): Las acciones correspondientes a paquetería se efectúan a través de la clase Menú.

3.4 Operaciones

Bloquear(): Método para bloquear las entradas de texto en los campos de Importe y Bodigo.

Limpiar(): Método que coloca en las cajas de texto la cadena vacía.

3.5 Atributos

No aplica.

3.6 Requerimientos especiales

Ningún requerimiento especial.

3.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

4. Diseño de clase: Empleado

4.1 Breve descripción

Esta clase es ocupada para la administración de los datos referidos a los empleados de la empresa.

4.2 Responsabilidades

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

4.3 Relaciones

No aplica.

4.4 Operaciones

GetCP(): Método que devuelve CP.

GetDomicilio(): Método que devuelve domicilio.

getEstado_civil(): Método que devuelve estado_civil.

getFecha_nacimiento(): Método que devuelve Fecha_nacimiento.

getId_empleado(): Método que devuelve id_empleado.

GetLocalidad(): Método que devuelve localidad.

GetNacionalidad(): Método que devuelve nacionalidad.

GetNombre(): Método que devuelve nombre.

GetSexo(): Método que devuelve sexo.

GetTelefono(): Método que devuelve teléfono.

GetTipo(): Método que devuelve tipo.

setCP(java.lang.String CP): Método que asigna código postal al empleado.

setDomicilio(java.lang.String domicilio): Método que asigna domicilio al empleado.

setEstado_civil(java.lang.String estado_civil): Método que asigna el estado civil al empleado.

setFecha_nacimiento(java.lang.String Fecha_nacimiento): Método que asigna la fecha de nacimiento al empleado.

setLocalidad(java.lang.String localidad): Método que asigna la localidad al empleado.

setNacionalidad(java.lang.String nacionalidad): Método que asigna la nacionalidad al empleado.

setNombre(java.lang.String nombre): Método que asigna nombre al atributo nombre.

setSexo(java.lang.String sexo) :Método que asigna el sexo al empleado.

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

setTelefono(java.lang.String telefono): Método que asigna numero telefonico al empleado.

setTipo(java.lang.String tipo): Método que asigna tipo al empleado.

4.5 Atributos

id_empleado: El atributo id_empleado, es un entero con el numero de identificación del empleado.

Nombre: El atributo nombre, es un String con el nombre del empleado.

Tipo: El atributo tipo, es un String con el tipo de empleado(Vendedor, chófer/conductor).

Domicilio: El atributo domicilio, es un String con la dirección del empleado.

Localidad: El atributo localidad, es un String con la localidad en la que reside el empleado.

Telefono: El atributo teléfono, es un String con el numero telefónico del empleado.

CP: El atributo CP, es un String con el Código postal del empleado.

Sexo: El atributo sexo, es un String con el sexo del empleado(Masculino/Femenino).

estado_civil: El atributo estado_civil, es un String con el estado civil(Casado(a)/Soltero(a)) del empleado.

Nacionalidad: El atributo nacionalidad, es un String con la nacionalidad del empleado.

Fecha_nacimiento: El atributo Fecha_nacimiento, es un String con la fecha de nacimiento del empleado.

4.6 Requerimientos especiales

Ningún requerimiento especial.

4.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

5. Diseño de clase: Listar_Empleados

5.1 Breve descripción

La clase listar_empleados se encarga de mostrar todos los registros de los empleados desplegando una lista con toda la información en una tabla.

5.2 Responsabilidades

Esta clase ayuda para visualizar los registros de manera inmediata y facilita la eliminación y consulta la

5.3 Relaciones

5.4 Operaciones

Listar_empleados(): Creates new form Listar_empleados

limpiar(): Coloca los campos de texto en cadenas vacías.

CargarSub(Suburban subs): Carga los estados de los asientos de una suburban en específico.

Asiento_#MouseClicked(java.awt.event.MouseEvent evt): Metodo que llena el campo de asiento con el numero # tras dar click en el boton con el asiento numero #.

CancelarActionPerformed(java.awt.event.ActionEvent evt): Metodo que cancela la venta tras presionar el boton cancelar, Este solo aplica si aun no se vende el boleto.

NuevoActionPerformed(java.awt.event.ActionEvent evt): Metodo para hacer una nueva venta.

Completar_ventaActionPerformed(java.awt.event.ActionEvent evt): Metodo que manda la orden al sistema de guardar los datos de la venta del boleto.

Validar(): Metodo que verifica que los campos esten llenos.

Vender(): Metodo que guarda la informacion obtenida de los campos del boleto vendido en la base de datos.

marcarAsiento(Integer asiento): Metodo que cambia el estado del asiento que se vende y lo marca para que este ya no se permita vender otra vez.

SeatKeyTyped(java.awt.event.KeyEvent evt): Metodo que verifica que en el campo asiento solo se puedan ingresar numeros.

DestinoItemStateChanged(java.awt.event.ItemEvent evt): Metodo que carga el estado de una suburban segun el destino, fecha, hora en ese orden.

5.5 Atributos

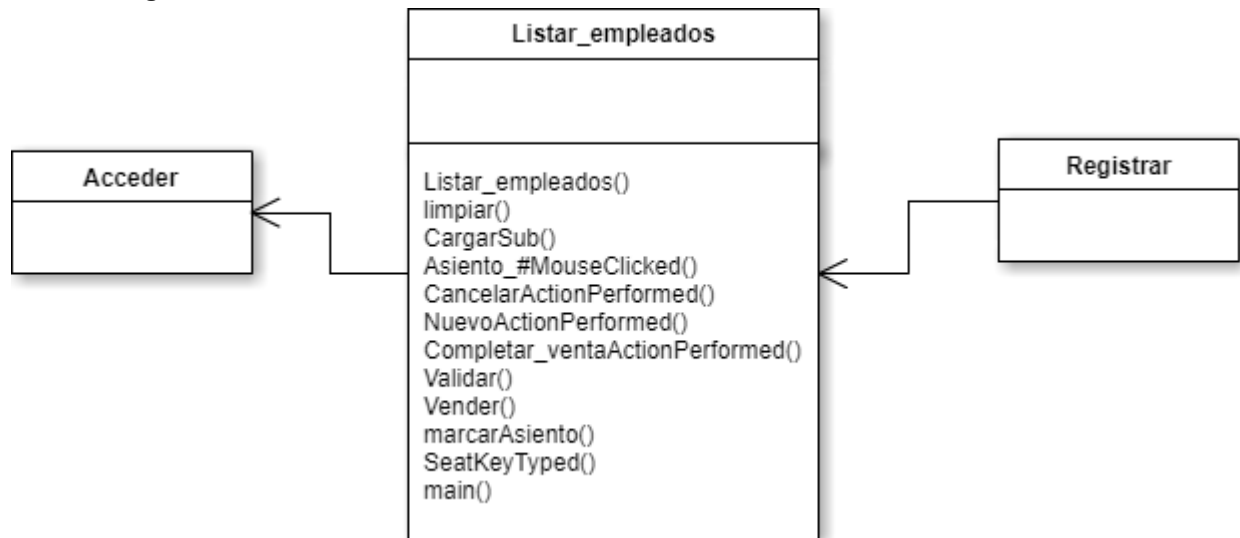
Sin atributos.

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

5.6 Requerimientos especiales

Ningún requerimiento especial.

5.7 Diagramas



Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

6. Diseño de clase: Empleado_Servicio

6.1 Breve descripción

Esta clase se encarga de administrar los registros contenidos en l abase de datos de los empleados d ella empresa.

6.2 Responsabilidades

Es responsable de la administración de los registros, elimina y recupera para consulta.

6.3 Relaciones

Conexion (relacion): La clase Empleado_servicio depende de la clase conexión para establecer una conexión con l abase de datos.

6.4 Operaciones

eliminar(java.sql.Connection conexion, Empleado empleado): Método que elimina un registro de la base de datos.

guardar(java.sql.Connection conexion, Empleado empleado): Método que guarda un registro de un empleado metido en la base de datos.

recuperarPorId(java.sql.Connection conexion, int id_empleado): Método que recupera un registro de la base de datos a partir de un ID

recuperarTodas(java.sql.Connection conexión): Método que recupera todos los registros de empleados de la base de datos.

Empleado_servicios(): Constructor de la clase.

6.5 Atributos

ID: El taributo ID, es un entero con el número de identificación del empleado.

6.6 Requerimientos especiales

Ningún requerimiento especial.

6.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

7. Diseño de clase: PaqueteríaDatos

7.1 Breve descripción

Esta clase es responsable de la administración de los datos referentes a los paquetes de la empresa.

7.2 Responsabilidades

Se encarga de la administración, requisitación y almacenamiento de los registros de los paquetes a enviar y entregados por la empresa.

7.3 Relaciones

No aplica.

7.4 Operaciones

GetChofer(): Método que devuelve chofer.

GetCodigo(): Método que devuelve Codigo.

GetDescripcion(): Método que devuelve Descripcion.

GetDestinatario(): Método que devuelve Destinatario.

GetDestino(): Método que devuelve destino.

GetEntregado(): Método que devuelve Entregado.

GetFecha(): Método que devuelve fecha.

GetImporte(): Método que devuelve Importe.

GetNpaquetes(): Método que devuelve Npaquetes.

GetOrigen(): Método que devuelve origen.

GetPeso(): Método que devuelve Peso.

GetRemitente(): Método que devuelve Remitente.

GetUnidad(): Método que devuelve Unidad.

setChofer(java.lang.String Ch): Método para asignar el chofer de la suburban en la que se realizara el envío.

setDescrpccion(java.lang.String Des): Método que le asigna la descripcion al paquete.

setDestinatario(java.lang.String D): Método que asigna el destinatario al envío.

setDestino(java.lang.String d): Método que signa el destino al envío.

setEntregado(java.lang.String E):Método para asignar el estado del envío(si fue entregado o aun no).

setFecha(java.lang.String f): Método para asignar la fecha del envío.

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

setImporte(float I): Método que asigna el importe al envío.

setNpaquetes(java.lang.Integer NP): Método que asigna el Numero de paquetes al envío

setOrigen(java.lang.String o): Método que le asigna el lugar de origen al envío.

setPeso(float p): Método que le asigna el peso al envío.

setRemitente(java.lang.String R): Método que asigna el remitente al envío.

setUnidad(java.lang.String u): Método que asigna el numero de la suburban en la que se enviara el paquete.

7.5 Atributos

R: El paramero R, es un String con el nombre del remitente.

D: El atributo D, es un String con el nombre del destinatario.

Des: El atributo Des, es un String con la descripcion del paquete.

NP: El atributo NP, es un entero con el numero de paquetes.

P: El atributo p, es un floatante con el peso del paquete o paquetes.

C: El atributo C, es un entero con el numero de identificacion del paquete.

I: El atributo I, es un flotante con el importe por el envío.

O: El atributo o, es un String con el nombre del lugar de donde se envia el paquete.

D: El atributo d, es un String con el nombre del lugar de destino.

F: El atributo fecha, es un String con la fecha en la que se realiza el envío.

U: El atributo U, es un String con el numero de la unidad en la que se enviara el paquete.

Ch: El atributo Ch, es un String con el nombre del chofer de la suburban.

E: El atributo E es un String que servira para saber si el paquete ya se ha entregado o no.

7.6 Requerimientos especiales

Ningún requerimiento especial.

7.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

8. Diseño de clase: Suburban

8.1 Breve descripción

Esta clase es la implementación de una suburban real, con el fin de mantener un registro de los boletos vendidos por viaje en ella.

8.2 Responsabilidades

Es la responsable de almacenar la información por viaje de la suburban elegida.

8.3 Relaciones

No aplica.

8.4 Operaciones

GetAsiento1(): Metodo que devuelve Asiento1.

GetAsiento10(): Metodo que devuelve Asiento10.

GetAsiento11(): Metodo que devuelve Asiento11.

GetAsiento12(): Metodo que devuelve Asiento12.

GetAsiento13(): Metodo que devuelve Asiento13.

GetAsiento14(): Metodo que devuelve Asiento14.

GetAsiento15(): Metodo que devuelve Asiento15.

GetAsiento16(): Metodo que devuelve Asiento16.

GetAsiento17(): Metodo que devuelve Asiento17.

GetAsiento2(): Metodo que devuelve Asiento2.

GetAsiento3(): Metodo que devuelve Asiento3.

GetAsiento4(): Metodo que devuelve Asiento4.

GetAsiento5(): Metodo que devuelve Asiento5.

GetAsiento6(): Metodo que devuelve Asiento6.

GetAsiento7(): Metodo que devuelve Asiento7.

GetAsiento8(): Metodo que devuelve Asiento8.

GetAsiento9(): Metodo que devuelve Asiento9.

getID_suburban(): Metodo que devuelve ID_suburban.

setAsiento1(lean val):Metodo que asigna el estado del asiento 1.

setAsiento10(lean val): Metodo que asigna el estado del asiento 10.

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

setAsiento11(lean val): Metodo que asigna el estado del asiento 11.

setAsiento12(lean val): Metodo que asigna el estado del asiento 12.

setAsiento13(lean val): Metodo que asigna el estado del asiento 13.

setAsiento14(lean val): Metodo que asigna el estado del asiento 14.

setAsiento15(lean val): Metodo que asigna el estado del asiento 15.

setAsiento16(lean val): Metodo que asigna el estado del asiento 16.

setAsiento17(lean val): Metodo que asigna el estado del asiento 17.

setAsiento2(lean val): Metodo que asigna el estado del asiento 2.

setAsiento3(lean val): Metodo que asigna el estado del asiento 3.

setAsiento4(lean val): Metodo que asigna el estado del asiento 4.

setAsiento5(lean val): Metodo que asigna el estado del asiento 5.

setAsiento6(lean val): Metodo que asigna el estado del asiento 6.

setAsiento7(lean val): Metodo que asigna el estado del asiento 7.

setAsiento8(lean val): Metodo que asigna el estado del asiento 8.

setAsiento9(lean val): Metodo que asigna el estado del asiento 9.

8.5 Atributos

Id: El atributo id, es un entero con el numero de identificacion de la suburban.

A1: El atributo a1, es un booleano con el estado del asiento(comprado/libre).

A2: El atributo a2, es un booleano con el estado del asiento(comprado/libre).

A3: El atributo a3, es un booleano con el estado del asiento(comprado/libre).

A4: El atributo a4, es un booleano con el estado del asiento(comprado/libre).

A5: El atributo a5, es un booleano con el estado del asiento(comprado/libre).

A6: El atributo a6, es un booleano con el estado del asiento(comprado/libre).

A7: El atributo a7, es un booleano con el estado del asiento(comprado/libre).

A8: El atributo a8, es un booleano con el estado del asiento(comprado/libre).

A9: El atributo a9, es un booleano con el estado del asiento(comprado/libre).

a10: El atributo a10, es un booleano con el estado del asiento(comprado/libre).

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

A11: El atributo a11, es un booleano con el estado del asiento(comprado/libre).

A12: El atributo a12, es un booleano con el estado del asiento(comprado/libre).

A13: El atributo a13, es un booleano con el estado del asiento(comprado/libre).

A14: El atributo a14, es un booleano con el estado del asiento(comprado/libre).

A15: El atributo a15, es un booleano con el estado del asiento(comprado/libre).

A16: El atributo a16, es un booleano con el estado del asiento(comprado/libre).

A17: El atributo a17, es un booleano con el estado del asiento(comprado/libre).

8.6 Requerimientos especiales

Ningún requerimiento especial.

8.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

9. Diseño de clase: Acceder

9.1 Breve descripción

Esta es la clase por la que se inicia

9.2 Responsabilidades

Clase de acceso al portal.

9.3 Relaciones

Listar_Empleados (relacion):

Manu (relacion):

9.4 Operaciones

Acceder(): Creates new form Acceder

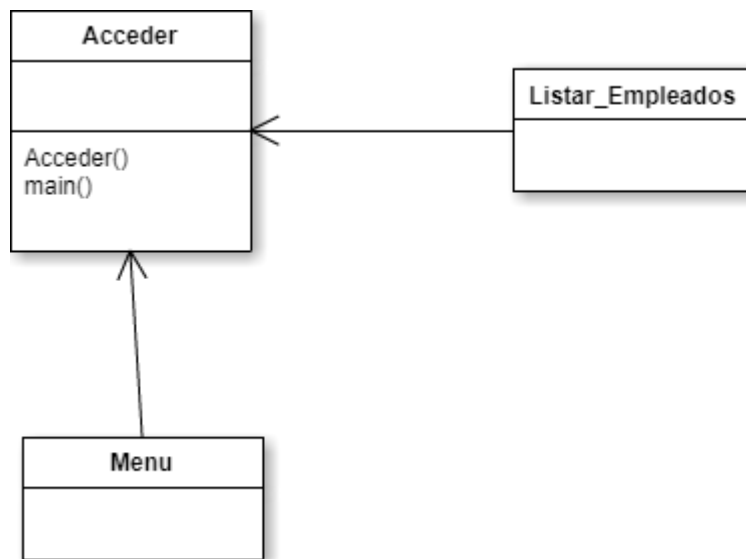
9.5 Atributos

No aplica

9.6 Requerimientos especiales

Ningún requerimiento especial.

9.7 Diagramas



Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

10. Diseño de clase: Login

10.1 Breve descripción

Esta clase funciona como puente Usuario - Sistema

10.2 Responsabilidades

10.3 Relaciones

10.4 Operaciones

10.5 Atributos

10.6 Requerimientos especiales

Ningún requerimiento especial.

10.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

11. Diseño de clase: Menu

11.1 Breve descripción

11.2 Responsabilidades

11.3 Relaciones

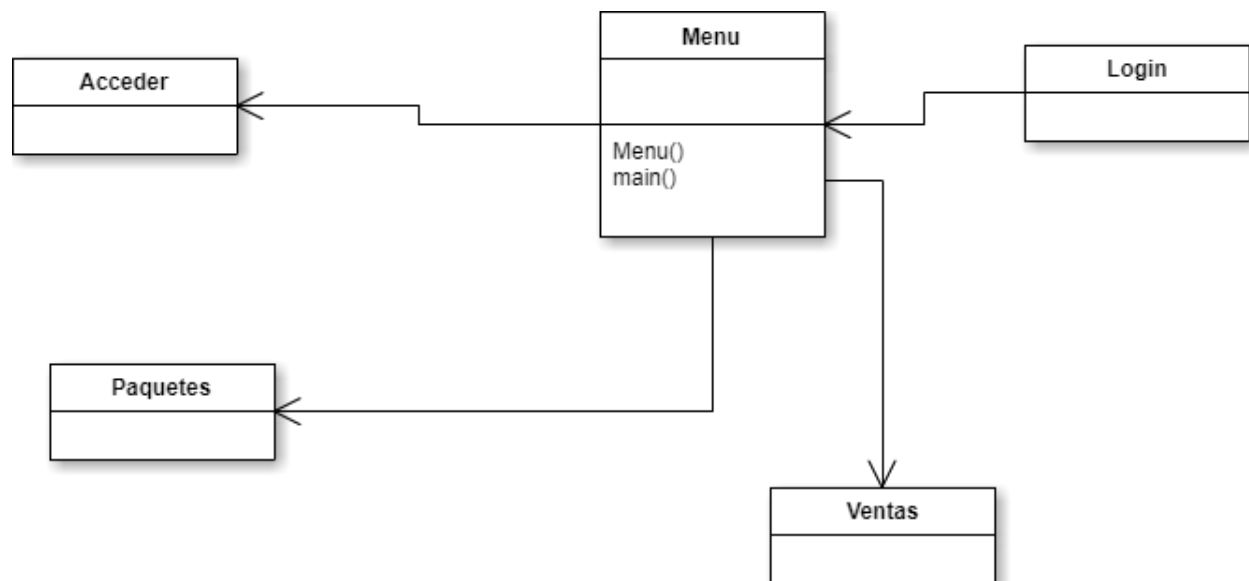
11.4 Operaciones

11.5 Atributos

11.6 Requerimientos especiales

Ningún requerimiento especial.

11.7 Diagramas



Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

12. Diseño de clase: Ticket

12.1 Breve descripción

12.2 Responsabilidades

12.3 Relaciones

12.4 Operaciones

12.5 Atributos

12.6 Requerimientos especiales

Ningún requerimiento especial.

12.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

13. Diseño de clase: SaleMaster

13.1 Breve descripción

13.2 Responsabilidades

13.3 Relaciones

13.4 Operaciones

13.5 Atributos

13.6 Requerimientos especiales

Ningún requerimiento especial.

13.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

14. Diseño de clase: Conexion

14.1 Breve descripción

14.2 Responsabilidades

14.3 Relaciones

14.4 Operaciones

14.5 Atributos

14.6 Requerimientos especiales

Ningún requerimiento especial.

14.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

15. Diseño de clase: Boleto_servicios

15.1 Breve descripción

15.2 Responsabilidades

15.3 Relaciones

15.4 Operaciones

15.5 Atributos

15.6 Requerimientos especiales

Ningún requerimiento especial.

15.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

16. Diseño de clase: Empleado_servicios

16.1 Breve descripción

16.2 Responsabilidades

16.3 Relaciones

16.4 Operaciones

16.5 Atributos

16.6 Requerimientos especiales

16.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

17. Diseño de clase: Suburban_servicios

17.1 Breve descripción

17.2 Responsabilidades

17.3 Relaciones

17.4 Operaciones

17.5 Atributos

17.6 Requerimientos especiales

Ningún requerimiento especial.

17.7 Diagramas

Sale Master	Version: 3.0
Diseño de Clases	Date: 03/Jul/17

18. Diseño de clase: Paquetes_servicios

18.1 Breve descripción

18.2 Responsabilidades

18.3 Relaciones

18.4 Operaciones

18.5 Atributos

18.6 Requerimientos especiales

Ningún requerimiento especial.

18.7 Diagramas