

---

---

# **Sale Master**

## **Diseño de Paquetes**

**Version 2.0**

Sale Master	Version: 2.0
Diseño de subsistema	Date: 02/07/17

## Historia del documento

Fecha	Versión	Descripción	Autor
20/Junio/2017	1.0	Se realiza la descripción de interfaces y diagramas	Nydia Beatriz Maciel Lopez.
02/07/17	2.0	Se realiza la descripción de interfaces y diagramas	Vásquez Cruz Yanet d ellos Ángeles.

## Contenido

1. Diseño del paquete: Modelos
  - 1.1 Breve descripción
  - 1.2 Interfaces
  - 1.3 Diagramas
  - 1.4 Elementos contenidos

Sale Master	Version: 2.0
Diseño de subsistema	Date: 02/07/17

# Diseño de paquetes

## 1. Diseño del paquete: Modelos

### 1.1 Breve descripción

Se encarga de recibir las peticiones del usuario y se envían las respuestas tras el proceso.

### 1.2 Interfaces

#### Nombre

Boleto

#### Descripción

Esta interfaz se encarga de recibir y administrar los datos recibidos por parte del usuario.

#### Operaciones

Boleto(): Se encarga de coordinar todas las llamadas de funciones para el correcto funcionamiento del caso de uso.

getID\_boleto(): Regresa el numero del boleto escogido.

getNo\_Asiento(): Regresa el numero de asiento escogido.

GetPasajero(): Regresa el nombre de la persona almacenada.

GetDestino(): Regresa el nombre del destino del viaje.

GetFecha(): Regresa la fecha de viaje seleccionado.

GetHora(): Regresa la hora de viaje seleccionado.

getID\_suburban(): Regresa el numero de la suburban seleccionado.

setNo\_Asiento(Integer asiento): Recibe y guarda el numero de asiento seleccionado.

setPasajero(String pas): Recibe y guarda el nombre del pasajero en turno.

setDestino(String destino): Recibe y guarda el nombre del destino de viaje seleccionado.

setFecha(String fecha): Recibe y guarda la fecha del viaje.

setHora(String hora): Recibe y guarda la hora del viaje seleccionada.

setID\_suburban(Integer id\_suburban): Recibe y guarda el numero de suburban seleccionado.

#### Atributos

ID\_boleto: De tipo Int, se encarga de asignar un numero de boleto a la venta futura.

No\_Asiento: De tipo Int, se encarga de almacenar el numero de asiento seleccionado.

Pasajero: De tipo string, se encarga de almacenar el nombre del cliente.

Destino: De tipo string, se encarga de almacenar el nombre del destino.

Fecha: De tipo string, se encarga de almacenar el nombre del cliente.

Hora: De tipo string, se encarga de almacenar la hora correspondiente del viaje disponible.

ID\_suburban: De tipo Int, se encarga de almacenar el numero de suburban escogido para el viaje.

#### Nombre

Empleado

#### Descripción

Esta interfaz se encarga de recibir y administrar los datos recibidos por parte del usuario correspondientes a los datos de un nuevo registro.

#### Operaciones

Empleado(): Se encarga de coordinar la funcionalidad.

getId\_empleado(): Regresa el id del empleado.

GetNombre(): Regresa el nombre del empleado.

GetTipo(): Regresa el tipo de empleado.

GetDomicilio(): Regresa el domicilio del nuevo empleado.

GetLocalidad(): Regresa la localidad del nuevo empleado.

Sale Master	Version: 2.0
Diseño de subsistema	Date: 02/07/17

GetTelefono(): Regresa el numero de telefono del empleado.

GetCP(): Regresa el codigo postal del empleado.

getEstado\_civil(): Regresa el estado civil del empleado.

GetNacionalidad(): Regresa la nacionalidad del nuevo empleado.

getFecha\_nacimiento(): Regresa la fecha de nacimiento de nuevo empleado.

setNombre(String nombre): Recibe y guarda el nombre del nuevo empleado.

setTipo(String tipo): Recibe y guarda el tipo del empleado del nuevo registro.

setDomicilio(String domicilio): Recibe y guarda el domicilio del nuevo registro.

setLocalidad(String localidad): Recibe y guarda la localidad del empleado del nuevo registro.

setTelefono(String telefono): Recibe y guarda el numero telefonico del empleado del nuevo registro.

setEstado\_civil(String estado\_civil): Recibe y guarda el estado civil del empleado del nuevo registro.

setNacionalidad(String nacionalidad): Recibe y guarda la nacionalidad del empleado del nuevo registro.

setFecha\_nacimiento(String Fecha\_nacimiento): Recibe y guarda la fecha de nacimiento del empleado del nuevo registro.

### Atributos

id\_empleado: De tipo Int, se encarga de almacenar el numero clave del empleado.

Nombre: De tipo String , se encarga de almacenar el nombre del nuevo empleado.

Tipo: De tipo String , se encarga de almacenar el tipo de empleado del nuevo registro.

Domicilio: De tipo String , se encarga de almacenar los datos del domicilio.

Localidad: De tipo String , se encarga de almacenar el nombre d ella localidad.

Telefono: De tipo String , se encarga de almacenar el numero telefonico del nuevo registro.

CP: De tipo String , se encarga de almacenar el codigo postal del nuevo registro.

Sexo: De tipo String , se encarga de almacenar si es femenino o masculino según sea el caso.

estado\_civil: De tipo String , se encarga de almacenar el estado civil del nuevo registro.

Nacionalidad: De tipo String , se encarga de almacenar la nacionalidad del nuevo registro.

Fecha\_nacimiento: De tipo String , se encarga de almacenar la fecha de nacimiento insertada.

### Nombre

PaqueteriaDatos

### Descripción

Esta interfaz se encarga de recibir y administrar los datos recibidos por parte del usuario correspondientes a los datos de un nuevo registro.

### Operaciones

PaqueteriaDatos(String R, String D,String Des,Integer NP,float p,Integer C,float I,String o,String d,String f,String U,String Ch,String E ): Se encarga de coordinar la funcionalidad.

setRemitente(String R): Se encarga de recibir y guardar el nombre del remitente a quien se envia el paquete.

setDestinatario(String D): Se encarga de recibir y guardar el destino de viaje del paquete.

setDescrpccion(String Des): Se encarga de recibir y guardar la descripción del paquete.

setNpaquetes(Integer NP): Se encarga de recibir y guardar el numero de paquetes ha enviar.

setPeso(float p): Se encarga de recibir y guardar el peso del paquete total.

setImporte(float I): Se encarga de recibir guardar en total a pagar por el envio del paquete.

setOrigen(String o): Se encarga de recibir y guardar el nombre del lugar donde el paquete es enviado.

setDestino(String d): Se encarga de recibir y guardar el nombre del destino del paquete.

setUnidad(String u): Se encarga de recibir y guardar el numero de unidad de de la suburban donde se enviara el paquete.

setChofer(String Ch): Se encarga de recibir y guardar el nombre del chofer de la unidad donde se enviara el paquete.

GetRemitente(): Regresa el nombre del remitente.

GetFecha(): Regresa la fecha de envio del paquete.

GetOrigen(): Regresa el nombre del lugar dle paquete enviado.

Sale Master	Version: 2.0
Diseño de subsistema	Date: 02/07/17

GetPeso(): Regresa el peso total del paquete enviado.  
 GetDestino(): Regresa el nombre del lugar de destino del paquete enviado.  
 GetUnidad(): Regresa el numero de unidad donde se envio el paquete.  
 GetChofer(): Regresa el nombre del chofer de la unidad donde se envio el paquete.  
 GetNpaquetes(): Regresa el numero de paquete enviados.  
 GetCodigo(): Regresa el codigo del paquete enviado.  
 GetImporte(): Regresa el precio total del paquete enviado.

#### **Atributos**

Remitente: De tipo String  
 Destinatario: De tipo String  
 Descripcion: De tipo String  
 Npaquetes: De tipo Int  
 Peso: De tipo Float  
 Codigo: De tipo Int  
 Importe: De tipo Int  
 origen: De tipo String  
 destino: De tipo String  
 fecha: De tipo String  
 Unidad: De tipo String  
 chofer: De tipo String  
 Entregado: De tipo String

#### **Nombre**

Suburban

#### **Descripción**

Esta interfaz se encarga de recibir y administrar los datos recibidos por parte del usuario, asi como de la interacion con el mismo.

#### **Operaciones**

Suburban(): Se encarga de controlar la funcionalidad.  
 getID\_suburban(),getAsiento1(),getAsiento2(),getAsiento3(),getAsiento4(),getAsiento5(),getAsiento6(),getAsiento7(),getAsiento8(),getAsiento9(),getAsiento10(),getAsiento11(),getAsiento12(),getAsiento13(),getAsiento14(),getAsiento15(),getAsiento16(),getAsiento17(): Regresa el numero de asiento seleccionado.

setAsiento1(Boolean val),setAsiento2(Boolean val),setAsiento3(Boolean val),setAsiento4(Boolean val),setAsiento5(Boolean val),setAsiento6(Boolean val),setAsiento7(Boolean val),setAsiento8(Boolean val),setAsiento9(Boolean val),setAsiento10(Boolean val),setAsiento11(Boolean val),setAsiento12(Boolean val),setAsiento13(Boolean val),setAsiento14(Boolean val),setAsiento15(Boolean val),setAsiento16(Boolean val),setAsiento17(Boolean val): Recibe y guarda el numero de asiento seleccionado.

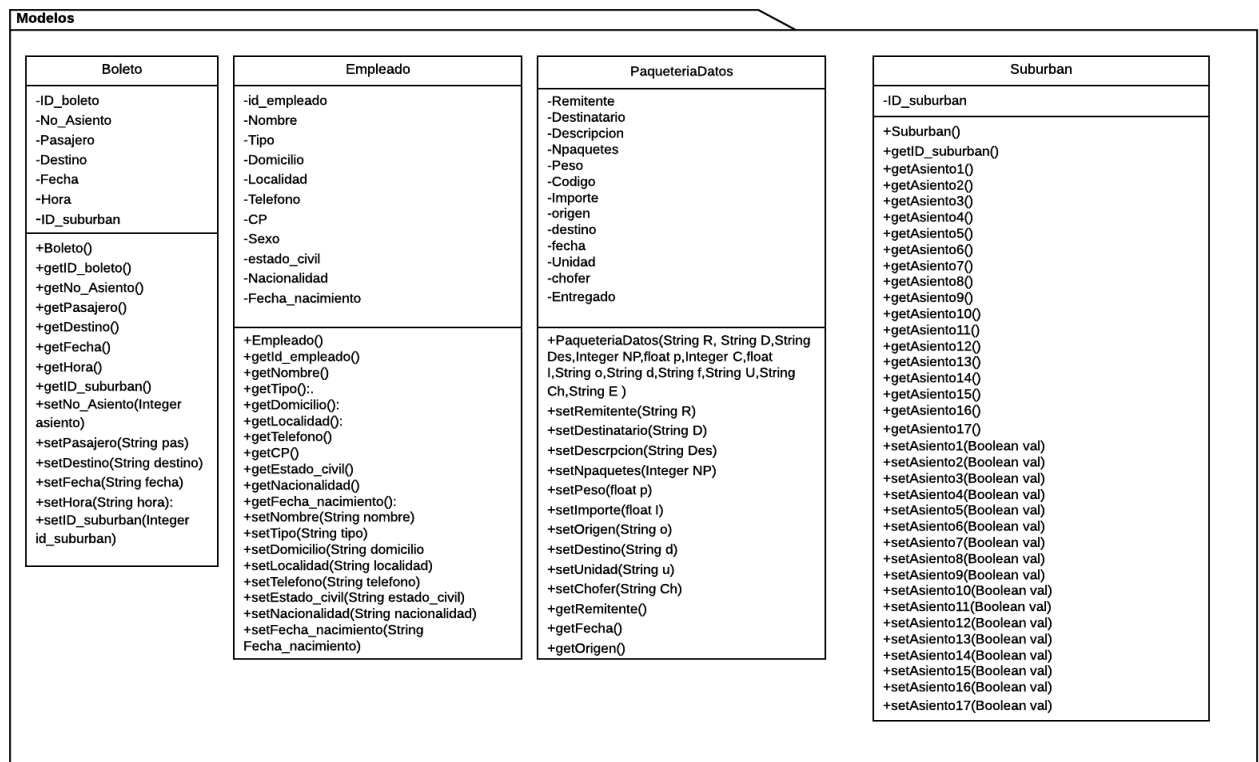
#### **Atributos**

ID\_suburban: Se encarga de almacenar el numero de suburban existentes.

Asiento1,Asiento2,Asiento3,Asiento4,Asiento5,Asiento6, Asiento7,  
 Asiento8,Asiento9,Asiento10,Asiento11,Asiento12, Asiento13,Asiento14,Asiento15,Asiento16,Asiento17:  
 correspondes a la cantidad de asientos disponibles.

Sale Master	Version: 2.0
Diseño de subsistema	Date: 02/07/17

### 1.3 Diagramas



### 1.4 Elementos contenidos

#### Nombre

Boleto

#### Descripción

Se encarga de guardar los datos que se reciben al utilizar la funcionalidad de ventas por parte del usuario.

#### Nombre

Empleado

#### Descripción

Se encarga de recibir y administrar los datos recibidos por parte del usuario correspondientes a los datos de un nuevo registro.

#### Nombre

Suburban

#### Descripción

Se encarga de recibir y administrar los datos recibidos por parte del usuario, así como de la interacción con el mismo.

Sale Master	Version: 2.0
Diseño de subsistema	Date: 02/07/17