

**Sistema Gestión de Pagos Electrónicos (SGPE).**  
Conexión Web Services para consultas y pagos de facturas.  
Definición Técnica Interfaz WS.

## Contenido

- 1) Introducción.
- 2) Requerimientos Funcionales.
- 3) Diagrama UML de los mensajes a compartir.
- 4) Descripción de las operaciones o métodos.
- 5) Glosario de mensajes y errores.
- 6) Requerimientos de velocidad.
- 7) Requerimientos de seguridad.

# 1. Introducción

## 1.1- Propósito:

El objetivo de este documento es el de proveer el estándar utilizado por el Sistema Gestión de Pagos Electrónicos (SGPE) para que entidades externas puedan realizar las transacciones y movimientos necesarios para realizar pagos, consultas, conciliación de lote, cierre de lote y todos los procesos que conllevan realizar pagos. Este documento está dirigido a personas del área de Sistemas de Información debido al alto contenido técnico involucrado en el mismo.

## 1.2- Alcance:

Proveer, a las entidades interesadas en esta facilidad, con la información necesaria para poder poner en marcha la comunicación entre ambas aplicaciones.

## 1.3- Definiciones y Acrónimos

### XML:

Es el acrónimo del inglés eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C). XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.

(<http://es.wikipedia.org/wiki/XML>)

### WebService:

Un servicio Web (en inglés Web service) es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

([http://es.wikipedia.org/wiki/Servicio\\_Web](http://es.wikipedia.org/wiki/Servicio_Web))

### SOAP:

Simple Object Access Protocol es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Es uno de los protocolos utilizados en los servicios Web.

([http://es.wikipedia.org/wiki/Simple\\_Object\\_Access\\_Protocol](http://es.wikipedia.org/wiki/Simple_Object_Access_Protocol))

### Lote:

Cantidad de transacciones realizada por un cajero (caja) en una tanda o turno de trabajo.

**Conciliación:**

Proceso en el cual se realiza un cuadro de todas las transacciones realizadas en un lote por una entidad externa vs las transacciones registradas en nuestros sistemas comerciales.

**Transacción:**

Registro de una acción o movimiento entre el sistema gestión de pagos electrónicos (SGPE) y entidades externas.

**NIC:**

Número de Contrato NIC Edenorte.

## 2. Requerimientos Funcionales

### 2.1- Formato para el intercambio de información:

La interfaz para realizar las transacciones y movimientos necesarios que conllevan realizar pagos, está desarrollada en forma de un servicio web, el cual puede accederse utilizando protocolo SOAP.

### 2.2- Descripción:

Las entidades interesadas podrán realizar las siguientes operaciones:

- Consultas por referencias (NIC, cédula).
- Información de Pago Aplicado a una factura previamente consultada.
- Conciliación de Pagos realizados.

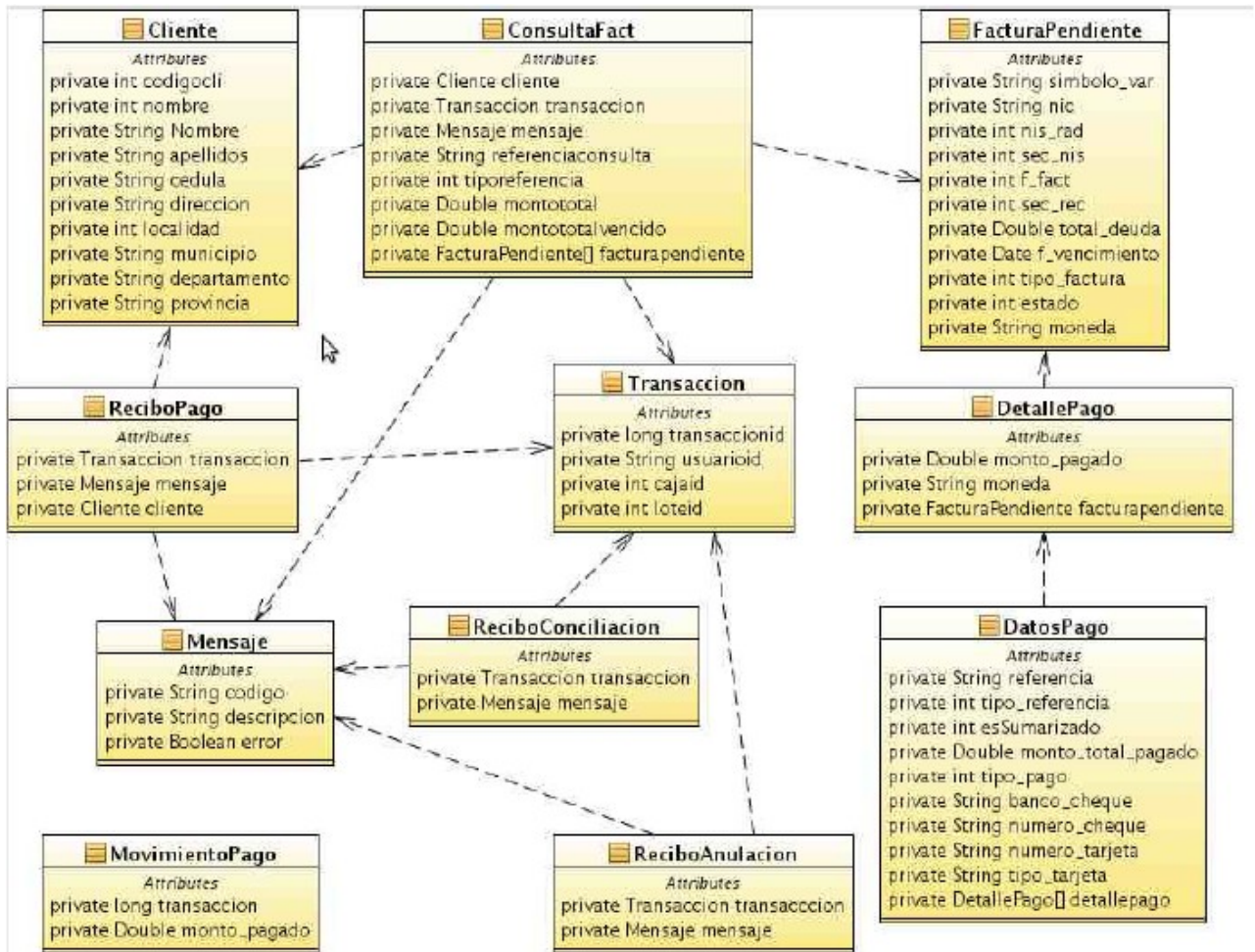
### 2.3- Ubicación del WSDL SGPE Ambiente Desarrollo

**Punto WS URL**

Wsd:

<http://prueba-pagoselectronicos.edenorte.com.do/WsEdenorteService/WsEdenorte?wsdl>

### 3.- Diagrama UML de los mensajes a compartir.



## 4.- Descripción de las operaciones o métodos:

### 4.1.1- Consulta Facturas Pendientes

**getConsultaFacturas( String referencia, int tipo\_referencia, String usuarioid, String clave, int cajaid ) : ConsultaFact**

### 4.1.2- Descripción

Permite obtener l(as) factur(as) que tiene pendiente un cliente.

### 4.1.3- Parámetros de entrada

**referencia** : El valor a enviar en este parámetro depende del parámetro tipo\_referencia. Por ejemplo: Si **tipo\_referencia** es 1, entonces referencia debe contener el NIC. Otros casos a especificar. (Cadena de caracteres alfanuméricos).

**tipo\_referencia** : Especificar por el tipo de referencia que desea consultar. El Parámetro tipo\_referencia es un entero y los valores a asignar son los siguientes:

#### Tipo Referencia:

1-NIC

Otros a especificar

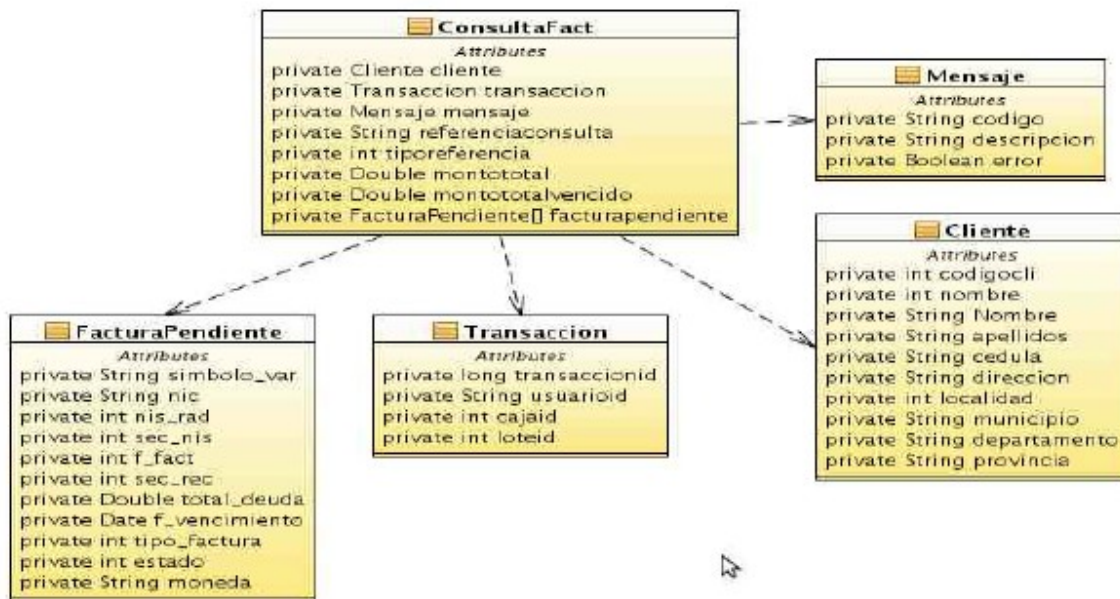
**usuarioid** : A designar, el usuario y password es único para cada empresa que realice la conexión al servicio (Cadena de caracteres alfanuméricos)

**clave** : A designar, el usuario y password es único para cada empresa que realice la conexión al servicio (Cadena de caracteres alfanuméricos).

**cajaid**: Número que identifica la caja desde donde se realiza la consulta [int,entero].

#### 4.1.4- Salida

Retorna un objeto tipo ConsultaFact.



#### 4.1.5- Definición de atributos de los objetos mostrados en la imagen anterior.

##### ConsultaFact :

Encapsula las factur(as) que tiene pendiente un cliente.

##### Propiedades

- **cliente**: Información del cliente dueño de la factura [Objeto Cliente].
- **transaccion**: Transacción en que se realizó la consulta [Objeto Transaccion].
- **mensaje** : contiene información sobre la consulta. Puede ser un mensaje de confirmación o error, según lo sucedido [Objeto Mensaje].
- **facturasPendiente**: Arreglo que contiene todas las facturas pendientes del cliente [ Arreglo de Objeto FacturaPendiente ].
- **referenciaConsulta**: El valor depende tipoReferencia . Por ejemplo: Si tipoReferencia es 1, entonces referencia debe contener el NIC. Otros casos a especificar. (Cadena de caracteres alfanuméricos).

- **tipoReferencia:** Indica las opciones de búsqueda. Es un entero y los valores a asignar son los siguientes:

**Tipo Referencia:**

1-NIC

Otros a especificar

- **montoTotal:** Saldo total pendiente de la factura [double].
- **montoTotalVencido:** Monto total de las facturas vencidas [double].
- **cantidadfacturaspendientes:** cantidad de facturas pendientes [int, entero].
- **nic:**Número de Contrato NIC Edenorte [int, entero].

**Cliente:**

Objeto que contiene la información o datos del cliente.

**Propiedades**

- codcli
- nombre
- apellidos
- cedula
- dirección
- localidad
- municipio
- departamento
- provincia
- puerta
- duplicador
- nombre\_finca

**Transacción:**

Objeto que encapsula una acción o movimiento entre el SGPE y una entidad externa.

**Propiedades**

- **transaccionid** : código o número de identificación de la transacción [long].
- **usuarioid** : A designar, el usuario y password es único para cada empresa que realice la conexión al servicio [String, cadena de caracteres alfanuméricos] .
- **cajaid** : Número que identifica la caja desde donde se realiza la consulta [int, entero].
- **loteid** : código o número de identificación del lote [long].



## Mensaje:

Encapsula el o los mensajes que puede intercambiar el SGPE con las entidades externas. Ver el glosario de mensajes 5.1

### • **Propiedades**

- **codigo:** código o número de identificación del mensaje [String, cadena de caracteres alfanuméricos].
- **descripcion:** En que consiste el mensaje [String, cadena de caracteres alfanuméricos].
- **error:** Establece si el mensaje es un error o no [boolean, recibe true o false].

## FacturaPendiente:

Datos sobre una factura pendiente

### **Propiedades**

- **SimboloVar:** Secuencial de la factura. Esta compuesto por el número de contrato más un secuencial de tres dígitos.
- **nic:** Número de Contrato NIC Edenorte [int, entero].
- **nisRad:** Número de identificación de suministro [int, entero].
- **secNis:** Secuencial de suministro [int, entero].
- **secRec:** Secuencial de recibo [int, entero].
- **fechaFactura:** Indica el período de la facturación [Date, fecha].
- **fechaVencimiento:** fecha de vencimiento de la factura [Date, fecha].
- **TipoFactura:** Información de descripción de la factura [String, cadena de caracteres alfanuméricos].
- **Estado:** Estado en que se encuentra la factura. [String, cadena de caracteres alfanuméricos].
- **moneda:** Moneda en que se realizó la factura. [String, cadena de caracteres alfanuméricos].
- **importeTotal:** Saldo pendiente de la factura [double].
- **TipoRecibo:** Indica tipo de recibo que será retornado. Información descriptiva. [String, cadena de caracteres alfanuméricos].

## 4.2.1- Pago Sumarizado

```
aplicarPagoSumarizado( DatosPago pago, String usuarioid,String clave, int cajaid ) :
```

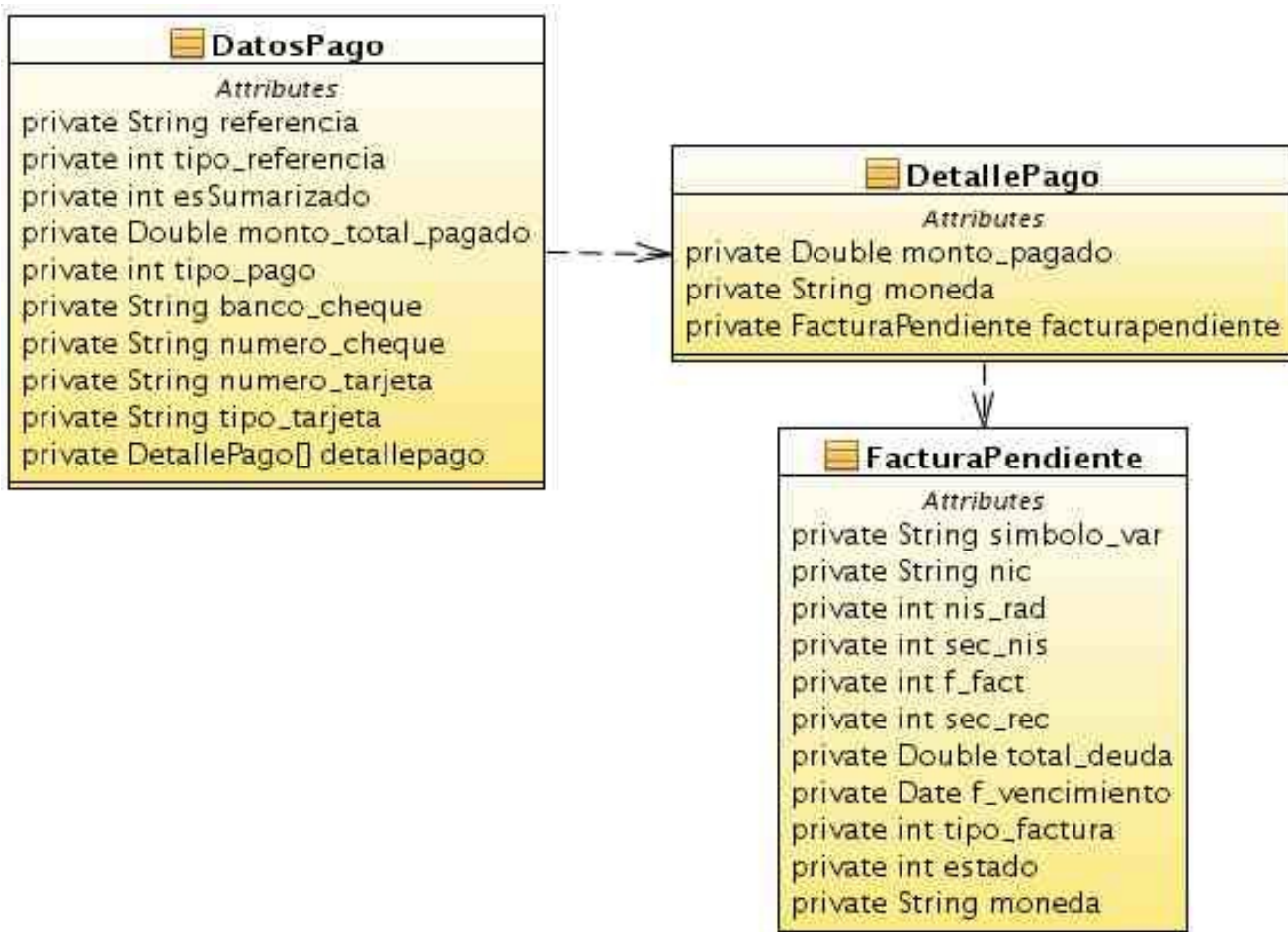
## ReciboPago

### 4.2.2- Descripción

Permite realizar un pago a una factura de forma sumarizada o agrupada, es decir, se aplica el monto total enviado al monto total adeudado a las posibles facturas pendientes que tenga el cliente.

### 4.2.3- Parámetros de entrada

**pago:** Objeto del tipo DatosPago, el cual contiene los datos concernientes al pago que se desea realizar.



## 4.2.4- Definición de propiedades en los objetos mostrados de la imagen anterior.

### DatosPago

Datos o información del pago a aplicar.

### Propiedades

- **referenciaConsulta:** El valor depende tipoReferencia . Por ejemplo: Si tipoReferencia es 1, entonces referencia debe contener el NIC. Otros casos a especificar. (Cadena de caracteres alfanuméricos).
- **tipoReferencia:** Indica las opciones de búsqueda. Es un entero y los valores a asignar son los siguientes:  
Tipo Referencia:  
1-NIC  
Otros a especificar
- **nic:** Número de Contrato NIC Edenorte [int, entero].
- **esSumarizado:** establece si el pago es sumarizado o no[boolean,esSumarizado = true].
- **MontoTotalPagado:** monto total a pagar [double].
- **TipoPago:** Establece el tipo de pago. [String, cadena de caracteres alfanuméricos]. Los valores que podemos usar son los siguientes:  
*EFFECTIVO: FP001*  
*PAGO CHEQUE: FP002*  
*PAGO TARJETA: FP003*
- **bancoCheque:** banco emisor del cheque [String, cadena de caracteres alfanuméricos].
- **numeroCheque:** número del cheque [String, cadena de caracteres alfanuméricos].
- **numeroTarjeta:** Número de tarjeta [String, cadena de caracteres alfanuméricos].
- **tipoTarjeta:** El tipo de tarjeta, esto es visa, mastercard, etc. [String, cadena de caracteres alfanuméricos].
- **DetallePago[] listaDetallePago:** Arreglo de objetos del tipo DetallePago, que representa las facturas pendientes a las cuales desea aplicar pago.

### Consideraciones:

- 1- Si el atributo esSumarizado tiene un valor true, indica que es sumarizado y por tanto el atributo listaDetallePago no debe contener información alguna. En caso de poseerla el sistema enviará un mensaje de error.
- 2- El atributo **MontoTotalPagado** debe de poseer la misma cantidad que indica el total de las facturas, en caso de haber un descuadre el sistema enviará un mensaje de error.

3- Solo el atributo correspondiente al tipo de pago seleccionado debe de contener alguna información, en caso contrario el sistema enviará un mensaje de error. Por ejemplo, si **TipoPago** tiene un valor de FP001, indica que el pago se realizará en efectivo, entonces **bancoCheque**, **numeroCheque**, **numeroTarjeta** y **tipoTarjeta**, no pueden poseer ningún valor.

4- Si el pago se procesara con diferentes procesadoras de tarjeta, es necesario asignar una caja por cada procesadora.

5- Si la entidad el servicio se integrara con mas de un canal y dichos canales tienen comiciones diferentes, es necesario asignarle una caja por cada canal que trabajara el servicio y debe conciliarse de manera individual.

### DetallePago

Información sobre una factura pendiente y el monto a pagar.

#### Propiedades

- **facturaPendiente**: Datos sobre una factura pendiente [ Objeto FacturaPendiente ].
- **montoPagado**: Monto pagado [double].
- **moneda**: moneda en la cual se desea realizar el pago [String, cadena de caracteres alfanuméricos].

### FacturaPendiente

--ver descripción de propiedades en el método **getConsultaFacturas**.

**usuarioid** : A designar, el usuario y password es único para cada empresa que realice la conexión al servicio (Cadena de caracteres alfanuméricos).

**clave** : A designar, el usuario y password es único para cada empresa que realice la conexión al servicio (Cadena de caracteres alfanuméricos).

**cajaid**: Número que identifica la caja desde donde se realiza la consulta [int, entero].

### 4.2.5- Salida

Retorna un objeto tipo ReciboPago.

### 4.2.6- Definición de propiedades en los objetos mostrados de la imagen anterior.

### ReciboPago

Objeto de confirmación cuando se realiza un pago.

#### Propiedades

- **cliente**: información del cliente que realizó el pago [Objeto Cliente].
- **transaccion**: transacción en que se realizó el pago [Objeto Transaccion].

- **mensaje** : contiene información sobre el proceso de aplicar pago. Puede ser un mensaje de confirmación o error, según lo sucedido [Objeto Mensaje].

Ver Objetos Mensaje, Cliente y Transacción, en el método **getConsultaFacturas**.

### 4.3.1- Pago al Detalle

**aplicarPagoDetallado( DatosPago pago, String usuarioid, String clave, int cajaid ) :**  
**ReciboPago**

#### 4.3.2- Descripción

Permite realizar un pagos especificando el monto deseado a pagar a cada factura pendiente.

#### 4.3.3- Parámetros de entrada

**pago**: Objeto del tipo DatosPago, el cual contiene los datos concernientes al pago que se desea realizar. ( Ver diagrama UML y descripción de propiedades en el método aplicarPagoSumarizado)

**usuarioid** : A designar, el usuario y password es único para cada empresa que realice la conexión al servicio (Cadena de caracteres alfanuméricos)

**clave** : A designar, el usuario y password es único para cada empresa que realice la conexión al servicio (Cadena de caracteres alfanuméricos) .

**cajaid**: Número que identifica la caja desde donde se realiza la consulta [int,entero].

#### 4.3.4- Salida

Retorna un objeto tipo ReciboPago ( Ver diagrama UML y descripción de propiedades en el método aplicarPagoSumarizado)

### 4.4.1- Conciliar Pagos

**ConciliarPagos( MovimientoPago[] listaMovimientos, String usuarioid, String clave, int cajaid )**  
**: ReciboConciliacion**

#### 4.4.2- Descripción

Método que permite conciliar todas las transacciones en un periodo N entre conciliaciones.

#### 4.4.3- Consideraciones

- Para los servicios de pagos 24 horas, la interfaz no permite que un **lote** cruce de un día a otro, por lo cual, es necesario realizar una conciliación cercana a la media noche, de lo contrario el sistema pedirá al día siguiente (12:00:01 A.M.) un cierre de lote y no permitirá realizar ninguna otra transacción.
- Para la conciliación, el sistema espera el conjunto de transacciones que se realizaron entre la última conciliación y la próxima, las transacciones que no sean enviadas en la lista de MovimientosPagos[] serán anuladas automáticamente por el sistema.
- Se debe hacer una conciliación por cada caja asignada al usuario.

#### 4.4.3- Parámetros de entrada

**listaMovimientos:** Arreglo de objeto del tipo MovimientoPago, que representa todas las Transacciones realizadas en un lote por una entidad externa.



#### 4.4.4- Definición de propiedades en los objetos mostrados de la imagen anterior.

##### MovimientoPago

Encapsula una acción donde se realizó un pago.

##### • Propiedades

• **Transaccion:** Representa el registro de la transacción del tipo pago realizada [long].

• **montopagado:** Monto pagado [double].

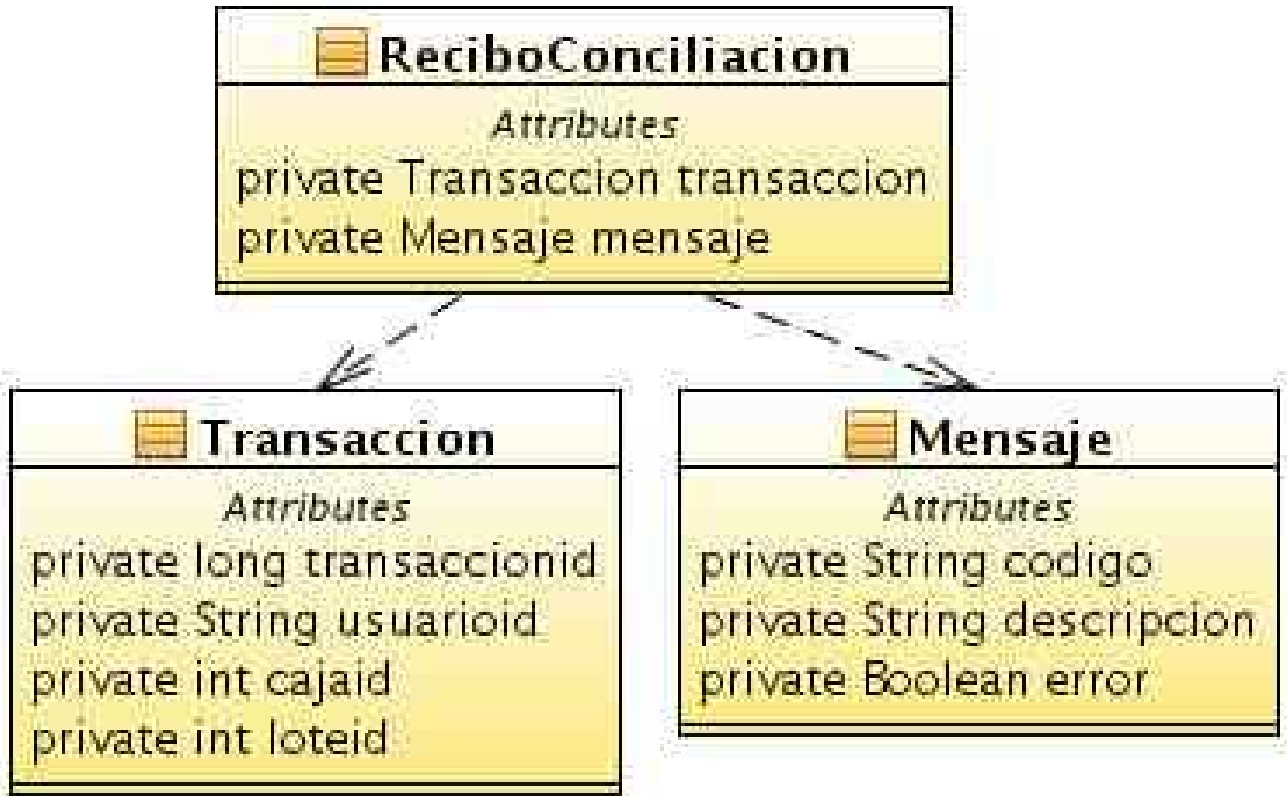
**usuarioid :** A designar, el usuario y password es único para cada empresa que realice la conexión al servicio (Cadena de caracteres alfanuméricos).

**clave :** A designar, el usuario y password es único para cada empresa que realice la conexión al servicio (Cadena de caracteres alfanuméricos).

**cajaid:** Número que identifica la caja desde donde se realiza la consulta [int, entero].

#### 4.4.5- Salida

Retorna un objeto tipo ReciboConciliacion



## 4.5- Anulaciones

### 4.5.1-Descripción:

Las anulaciones de pago es el proceso en el cual, luego de haber aplicado un cobro y este cobro este cobro se convierte en no deseado, se procede a realizar la anulación de pago.

### 4.5.2-Requisitos:

La anulación de cobro debe de ejecutarse el mismo día en el que se realizo el pago.

Se debe de solicitar al departamento de control de cobranza una clave de anulación para la transacción de pago que deseamos anular. **(Opcional)**

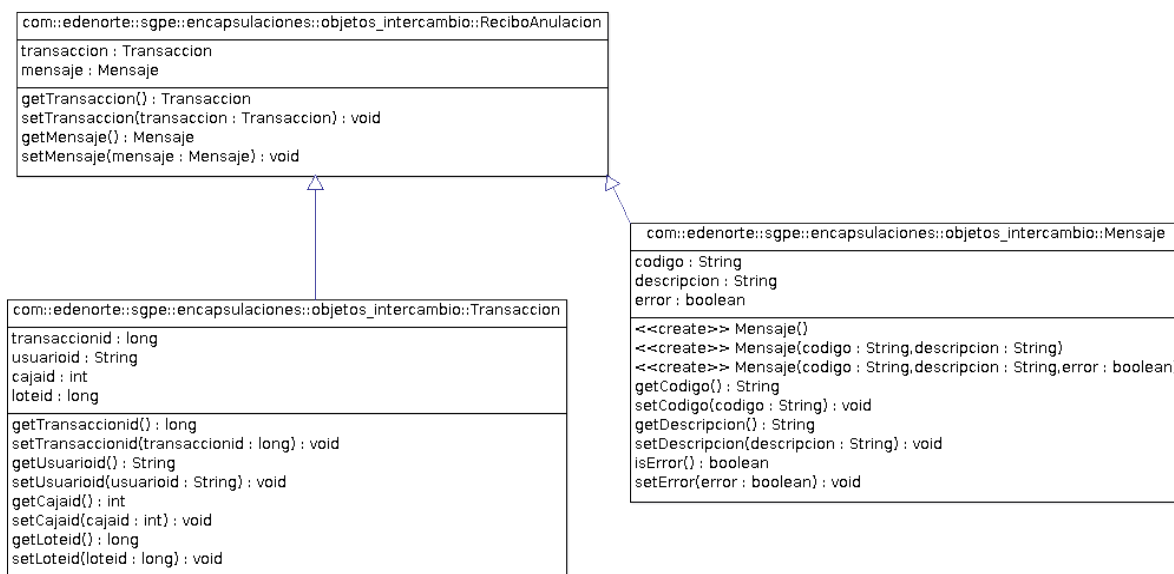
### 4.5.3-Método de Anulación:

`ReciboAnulacion anularTransaccionPago(long transaccionid,String claveAnulacion,  
String usuarioid, String clave, int cajaid)`

### 4.5.4-Parámetros

- **transaccionid** : transacción de pago a anular.
- **claveAnulacion**: Clave que entregara control de cobranza al momento de solicitar la anulación. (Opcional,Configurable) .
- **usuarioid**: usuario asignado para conectarse a los serviciosWeb.
- **clave**: clave asignada para conectarse a los serviciosWeb.
- **cajaid**: caja asignada al usuario.

#### 4.5.5-Retorno(ReciboAnulacion)



## 5- Guía de Mensajes

### 5.1 - Descripción

A continuación se presentara las tablas de mensajes de retorno que devuelve nuestra interfaz por cada uno de sus métodos.

### 5.2- Tabla de mensajes método **getConsultaFact**

Código	Mensaje
M01	Consulta realiza con éxito
M05	Debe realizar Cierre de lote, tiene lotes pendientes
E21	Formato de NIC no valido, debe ser numérico Este Cliente es Bonoluz Debe de pasar por la oficina de Edenorte para realizar el
E39	pago
E20	Cliente consultado sin deuda pendiente
E01	Referencia de búsqueda no valida
E99	Error Interno
E04	Usuario no registrado a caja



### 5.3- Tabla de mensajes método **aplicarPago**

Código	Mensaje
M05	Debe realizar Cierre de lote, tiene lotes pendientes
M02	Pago realizado con éxito
E70	Error en consulta obteniendo datos cliente
E09	Forma de pago Cheque no completada
E13	Forma de pago Cheque con datos de tarjeta completados
E10	Forma de pago Tarjeta de Crédito no completada
E14	Forma de pago Tarjeta con datos de cheques completados
E11	Forma de pago Efectivo con información en campos para cheque o tarjeta de crédito
E12	Forma de pago no valida
E07	Monto del pago factura NO puede ser cero o menor
E26	Pago sumariado no puede realizarse un pago anticipado.
E22	Pago sumariado no debe enviar detalle de documentos de pago.
E06	Monto del pago supera el valor de la deuda de la factura
E07	Monto del pago factura NO puede ser cero o menor
E08	Monto total y del detalle no corresponden
E05	Petición de pago por el detalle sin indicar los documentos de pagos
E80	Error Interno
E04	Usuario no registrado a caja

### 5.4- Tabla de mensajes método **anularTransaccionPago**

Código	Mensaje
M03	Anulación realizada con éxito
M05	Debe realizar Cierre de lote, tiene lotes pendientes
E81	Error Interno
E15	Lote del documento anular no corresponde con el anulado.
E17	Transacción especificada no tiene pagos registrados.
E18	Transacción ya anulada.
E16	Contraseña control anulación no coincide. Parámetro global no existe registrado, indicar al departamento Sistema
E23	Edenorte
E44	Debe de solicitar permiso de anulación a control de cobranza
E45	No hay permisos para anular cobros sin Autorización
E98	Error Aplicando la anulación
E45	El pago no fue anulado en el SGC
E97	Error Al anular el pago SGC.
E04	Usuario no registrado a caja
E48	Transacción de anulación invalida

## 5.5- Tabla de mensajes método **conciliarPagos**

Código	Mensaje
M04	Conciliación realizada con éxito
M05	Debe realizar Cierre de lote, tiene lotes pendientes
E95	Error Aplicando la conciliación
E82	<b>Error Interno</b>
E04	Usuario no registrado a caja
W01	Conciliación realizada con descuadres

### 6- Requerimientos de velocidad.

Edenorte dominicana garantiza de que el tiempo de respuesta de nuestro servicio web Oscilará entre los 8 y 60 segundos en la realización de una o más transacciones, en caso de que alguna entidad tenga un tiempo mayor al establecido debe de revisar la forma en que se conecta al servicio web.

Es importante recordar que nuestro objetivo debe ser siempre dejar en nuestros clientes una gran satisfacción, de tal forma que sirva de estímulo para futuras visitas.

### 7- Requerimientos de seguridad.

Es necesario establecer una Red Privada Virtual (VPN) para la salida en productivo.

Protocolo: IPSEC

### 8- Credenciales para el servicio

Usuarioid: pruebas

Claveid: pruebas123

Cajaid: 128,88,1,108,149,48,28,6,5