

## MANUAL TECNICO

### Objetivo.

El objetivo del manual técnico es contemplar los aspectos que se detallan a continuación:

- Creación de la Base de Datos.
- Definir requerimientos obligatorios para el manejo de los sistemas y la generación de informes.
- Para saber cuál es la logística que se ha llevado a cabo para la realización del Aplicativo.
- Es un Asistente para el mantenimiento del mismo.

### Objetos de la base de datos.

La creación de las Bases de Datos con sus respectivas tablas que van a estar relacionadas, las mismas que se detallan a continuación:

GN_TIPOCLIENTE	GN_PROCESO	GN_VARIABLES
GN_MENSAJE	GN_ESTADO	
GN_TRANSACCIONESENVIOSGENERAL		
GN_TRANSACCIONESENVIOSESPECIF		

## **Diseño de Tablas.**

Las tablas son utilizadas en base de datos para representar la información de forma más compacta y de fácil acceso. Estas tablas están compuestas por filas y columnas. Las filas de una tabla equivalen a los registros los cuales contienen los valores de los objetos, y las columnas equivalen a los campos los cuales contienen los atributos de los objetos.

La estructura de las tablas muestra la forma como se almacenan los datos y en el caso de estar relacionadas con otras tablas cuales serian los campos que se relacionan, de dicha relación nacen los conceptos de campos foráneos y primarios.

### **Clave Primaria**

La clave primario es la de menor tamaño en cuanto a número de campos que la componen y es el identificador único para una tabla. Con la clave primaria nunca existen dos filas de una tabla con el mismo valor.

### **Clave Foránea**

Una clave foránea es un atributo de una tabla o adición de atributos ya existentes a una tabla; ésta inclusive puede ser una clave primaria de otra tabla. Sin embargo los atributos

de una clave foránea no necesariamente tienen que formar parte de la clave primaria de la tabla a la que pertenecen.

### **Definición Estructura de Datos.**

La estructura de las tablas muestra detalladamente los campos almacenados y la relación que existe entre cada una de las tablas para comprender el flujo de los datos.

**TABLA GN\_TIPOCLIENTE**

GN_TIPOCLIENTE	
<b>PK</b>	<b>ID</b>
	DESCRIPCION
	USUARIOACTUALIZA
	FECHAACTUALIZA

Esta tabla almacenara los tipos de clientes a los que se envía notificaciones.

### TABLA GN\_PROCESO

GN_PROCESO	
<b>PK</b>	<b>ID</b>
	NOMBRE_PROCESO
	DESCRIPCION
	USUARIOACTUALIZA
	FECHAACTUALIZA
	MASIVO

Aquí se almacenaran los diferentes procesos que se utilizaran para los envíos, los dos principales objetos del proyecto son los envíos de notificaciones por vencimientos de pólizas y vencimientos de cartera; también existirán procesos adicionales para promociones

### TABLA GN\_VARIABLES

GN_VARIABLES	
<b>PK</b>	<b>ID</b>
	NOMBRE_VARIABLE
	DESCRIPCION
	USUARIOACTUALIZA
	FECHAACTUALIZA
	REFERENCIA

Esta tabla almacenara todas las variables que formaran parte del cuerpo de los mensajes que serán enviados.

### TABLA GN\_ESTADO

GN_ESTADO	
<b>PK</b>	<b>ID</b>
	NOMBRE
	NOMBRECLASE
	USUARIOACTUALIZA
	FECHAACTUALIZA

Esta tabla almacenara los diferentes estados que puedan tener las transacciones, y los diferentes mensajes que se creen.

### TABLA GN\_MENSAJE

GN_MENSAJE	
<b>PK</b>	<b>ID</b>
<b>FK</b>	GN_PROCESOID
	INTERVALO_EJEC
	DIAS_RESTANTES
	CANT_ENVIOS
<b>FK</b>	GN_TIPOCLIENTEID
	MENSAJE

	ENVIA_SMS
	ENVIA_MAIL
	FECHAINICIO
	FECHAFIN
<b>FK</b>	GN_ESTADOID
	ADJUNTO_MAIL
	ADJUNTO_SMS
	USUARIOACTUALIZA
	FECHAACTUALIZA
	RUTA_ADJUNTOMAIL

Aquí se almacenaran todos los mensajes que se configuren con las variables necesarias, y estos serán los que posteriormente se envíen.

### **TABLA GN\_TRANSACCIONESENVIOSESPECIF**

<b>GN_TRANSACCIONESENVIOSESPECIF</b>	
<b>PK</b>	<b>ID</b>
	POLIZAID
	ENDOSOID
	AGENTE
	RAMO
	NUMPOLIZA

ORDEN  
CLIENTE  
TIPOENDOSO  
NUMENDOSO  
VALORASEGURADO  
VALORPRIMANETA  
ESTADOPOLIZA  
ESTADOENDOSO  
FECHAFACTURACION  
FECHACONTABILIZACION  
VIGENCIADESDE  
VIGENCIAHASTA  
FACTURANUM  
ESTADOFACTURA  
TIPOSEGURO  
MESPRODUCCION  
ANIOPRODUCCION  
EJECCOMERCIAL  
USUARIOEMITE  
CIUDADPRODUCCION  
DOCFACTID  
DOCCUOTAID  
DOCLETRAID  
NUMCUOTALETRA  
DESCRIPCIONCUOTALETRA  
VALORCUOTALETRA  
FEHAVENCIMIENODOC

	ESTADOCUOTALETRA
	SALDOCUOTALETRA
<b>FK</b>	GN_MENSAJEID
	FECHAENVIO
	RESULTADOSMS
	DESCRIPCIONRESULTSMS
	RESULTADOMAIL
	DESCRIPCIONRESULTMAIL
	MENSAJE
	CELULAR
	MAIL
	ADJUNTO

Esta tabla contendrá todas las transacciones enviadas por los procesos de cartera y de vencimientos de pólizas.

### **TABLA GN\_TRANSACCIONESENVIOSGENERAL**

<b>GN_TRANSACCIONESENVIOSGENERAL</b>	
<b>PK</b>	<b>ID</b>
<b>FK</b>	GN_MENSAJEID
	FECHAENVIO
	RESULTADOSMS
	DESCRIPCIONRESULTSMS
	RESULTADOMAIL
	DESCRIPCIONRESULTMAIL
	MENSAJE



CELULAR
MAIL
ADJUNTO
CLIENTE

Esta tabla contendrá todas las transacciones enviadas por los procesos de promociones o cualquier otro tipo de evento.

## Script de las Tablas Principales de la base de Datos

### Tabla GN\_MENSAJE

```
-- Create table
create table GN_MENSAJE
(
  ID          VARCHAR2(12) not null,
  GN_PROCESOID  VARCHAR2(20),
  INTERVALO_EJEC  NUMBER(10,2),
  DIAS_RESTANTES  NUMBER(10,2),
  CANT_ENVIOS    NUMBER(10,2),
  GN_TIPOCLIENTEID VARCHAR2(20),
  MENSAJE       VARCHAR2(1000),
  ENVIA_SMS     CHAR(1),
  ENVIA_MAIL    CHAR(1),
  FECHAINICIO   DATE,
  FECHAFIN     DATE,
  GN_ESTADOID   VARCHAR2(12),
  ADJUNTO_MAIL  BLOB,
  ADJUNTO_SMS   BLOB,
  USUARIOACTUALIZA VARCHAR2(12),
  FECHAACTUALIZA DATE,
  RUTA_ADJUNTOMAIL VARCHAR2(500)
)
tablespace USERS
pctfree 10
intrans 1
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table GN_MENSAJE
add constraint PK_GNMENSAJE primary key (ID)
using index
tablespace USERS
pctfree 10
intrans 2
maxtrans 255
storage
(
  initial 384K
  minextents 1
```

```

        maxextents unlimited
    );
alter table GN_MENSAJE
    add constraint FK_GNMENSAJE_GNESTADO foreign key (GN_ESTADOID)
    references GN_ESTADO (ID);
alter table GN_MENSAJE
    add constraint FK_GNMENSAJE_GNPROCESO foreign key (GN_PROCESOID)
    references GN_PROCESO (ID);
alter table GN_MENSAJE
    add constraint FK_GNMENSAJE_GNTIPOCLIENTE foreign key (GN_TIPOCLIENTEID)
    references GN_TIPOCLIENTE (ID);

```

## **Tabla GN\_TRANSACCIONESENVIOSPECIF**

```

-- Create table
create table GN_TRANSACCIONESENVIOSPECIF
(
    ID                VARCHAR2(12),
    POLIZAID          VARCHAR2(12),
    ENDOSOID           VARCHAR2(12),
    AGENTE             VARCHAR2(100),
    RAMO               VARCHAR2(40),
    NUMPOLIZA          NUMBER(10),
    ORDEN              NUMBER(10),
    CLIENTE            VARCHAR2(100),
    TIPOENDOSO         VARCHAR2(50),
    NUMENDOSO          NUMBER(10),
    VALORASEGURADO     NUMBER(20,2),
    VALORPRIMANETA     NUMBER(20,2),
    ESTADOPOLIZA       VARCHAR2(10),
    ESTADOENDOSO       VARCHAR2(10),
    FECHAFACTURACION  DATE,
    FECHACONTABILIZACION DATE,
    VIGENCIADESDE     DATE,
    VIGENCIAHASTA     DATE,
    FACTURANUM         VARCHAR2(20),
    ESTADOFACTURA     VARCHAR2(10),
    TIPOSEGURO         VARCHAR2(20),
    MESPRODUCCION     VARCHAR2(10),
    ANIOPRODUCCION    VARCHAR2(20),
    EJECOMERCIAL       VARCHAR2(100),
    USUARIOEMITE       VARCHAR2(100),
    CIUDADPRODUCCION  VARCHAR2(30),
    DOCFACID           VARCHAR2(12),
    DOCCUOTAID         VARCHAR2(12),
    DOCLETRAID         VARCHAR2(12),
    NUMCUOTALETRA      NUMBER(10),
    DESCRIPCIONCUOTALETRA VARCHAR2(500),
    VALORCUOTALETRA    NUMBER(20,2),

```

```

FECHAVENCIMIENODOC DATE,
ESTADOCUOTALETRA VARCHAR2(10),
SALDOCUOTALETRA NUMBER(20,2),
GN_MENSAJEID VARCHAR2(20),
FECHAENVIO DATE,
RESULTADOSMS VARCHAR2(10),
DESCRIPCIONRESULTSMS VARCHAR2(500),
RESULTADOMAIL VARCHAR2(10),
DESCRIPCIONRESULTMAIL VARCHAR2(500),
MENSAJE VARCHAR2(3000),
CELULAR VARCHAR2(15),
MAIL VARCHAR2(30),
ADJUNTO VARCHAR2(800)
)
tablespace USERS
pctfree 10
initrans 1
maxtrans 255
storage
(
initial 64K
minextents 1
maxextents unlimited
);
-- Create/Recreate primary, unique and foreign key constraints
alter table GN_TRANSACCIONESENVIOESPECIF
add constraint PK_GNTRANSACCIONESPECIF primary key (ID)
disable;
alter table GN_TRANSACCIONESENVIOESPECIF
add constraint FK_GNTRANSENVESP_GNMENSAJE foreign key (GN_MENSAJEID)
references GN_MENSAJE (ID)
disable;

```

## **Estándares de la Lógica del Negocio.**

Se va a detallar las clases principales que intervienen en la aplicación:

### **Clase Bean**

Esta clase permite la conexión con la base de datos, y es invocada cada vez que se realizan inserciones, actualizaciones, borrado, consultas. Etc.

```

package GestorNotificaciones;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.PageContext;

public class Bean {

    public JspWriter m_out;
    public PageContext m_pageContext;
    public HttpServletResponse m_response;
    public Connection m_conn;
    protected JspLib m_jl;
    public HttpServletRequest m_request;

    public Bean(PageContext p_pageContext,String usuario,String clave)throws Exception, SQLException {

        Class.forName ("oracle.jdbc.driver.OracleDriver");

        m_conn=DriverManager.getConnection("jdbc:oracle:thin:@(description=(address=(host=localhost)(protocol=
=tc)(port=1521))(connect_data=(sid=ORCL)))",usuario,clave);
        //Class.forName ("com.mysql.jdbc.Driver");

//m_conn=DriverManager.getConnection("jdbc:mysql://" +bd+"/FIRETECH?user=firewall&password="+password);
        m_pageContext=p_pageContext;
        m_out=m_pageContext.getOut();
        m_response=(HttpServletResponse)p_pageContext.getResponse();
        m_request = (HttpServletRequest)m_pageContext.getRequest();
        m_jl=new JspLib(p_pageContext);
    }
}

```

## Clase SendAuthentication2

Clase de Autenticación para el envío de mail sin archivos adjuntos

```

package GestorNotificaciones;

import java.sql.SQLException;

```

```

import java.util.Properties;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import javax.servlet.jsp.PageContext;

public class SendAuthentication2 extends Bean {

    public SendAuthentication2 (PageContext p_pageContext,String usuario,String password)throws
        Exception, SQLException {
        super(p_pageContext,usuario,password);
    }

    public String Send(String mail, String mensaje)
    {
        String host ="69.10.53.202";//Suponiendo que el servidor SMTP sea la propia máquina
        String from ="webmaster@customersoft.net";
        String to = mail;

        System.out.println ("Prueba para enviar un mail..." + new java.util.Date());

        Properties prop = new Properties();
        prop.put("mail.smtp.host", host);
        prop.put("mail.smtp.port", "26");
        /*Esta línea es la que indica al API que debe autenticarse*/
        prop.put("mail.smtp.auth", "true");
        /*Añadir esta línea si queremos ver una salida detallada del programa*/
        //prop.put("mail.debug", "true");

        try{
            SMTPAuthentication auth = new SMTPAuthentication();
            Session session = Session.getInstance(prop , auth );
            Message msg = getMessage(session, from, to, mensaje);
            System.out.println ("Enviando ..." );
            Transport.send(msg);
            System.out.println ("Mensaje enviado!");
            return ("Mensaje enviado!");
        }
        catch (Exception e)
        {
            ExceptionManager.ManageException(e);
            return ("Mensaje no se pudo enviar!" +e);
        }
    }

    private static MimeMessage getMessage(Session session, String from, String to, String mensaje)
    {
        try{
            MimeMessage msg = new MimeMessage(session);
            msg.setText(mensaje);
            msg.addRecipient(Message.RecipientType.TO, new InternetAddress(to));

```

```

msg.setFrom(new InternetAddress(from,"Seguros Balboa"));
msg.setSubject("Notificación - Seguros Balboa");
return msg;
}
catch (java.io.UnsupportedEncodingException ex)
{
ExceptionManager.ManageException(ex);
return null;
}
catch (MessagingException ex)
{
ExceptionManager.ManageException(ex);
return null;
}
}
}
}

```

## Clase SendAuthentication

Clase de Autenticación para el envío de mail con archivos adjuntos

```

package GestorNotificaciones;

import java.sql.SQLException;
import java.util.Properties;

import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import javax.servlet.jsp.PageContext;
import javax.activation.DataHandler;
import javax.activation.FileDataSource;
import javax.mail.BodyPart;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMultipart;

public class SendAuthentication extends Bean {

    public SendAuthentication (PageContext p_pageContext,String usuario,String password)throws Exception,
SQLException {
        super(p_pageContext,usuario,password);
    }
}

```

```

public String Send(String mail, String mensaje, String ruta)
{

String host ="69.10.53.202";//Suponiendo que el servidor SMTPsea la propia máquina
String from ="webmaster@customersoft.net";
String to = mail;

System.out.println ("Prueba para enviar un mail..." + new java.util.Date());

Properties prop = new Properties();
prop.put("mail.smtp.host", host);
prop.put("mail.smtp.port", "26");
/*Esta línea es la que indica al API que debe autenticarse*/
prop.put("mail.smtp.auth", "true");
/*Añadir esta linea si queremos ver una salida detallada del programa*/
//prop.put("mail.debug", "true");

try{

        SMTPAuthentication auth = new SMTPAuthentication();
        Session session = Session.getInstance(prop , auth );

BodyPart texto = new MimeBodyPart();
texto.setText(mensaje);
// Se compone el adjunto con la imagen
BodyPart adjunto = new MimeBodyPart();
adjunto.setDataHandler(
        new DataHandler(new FileDataSource(ruta)));
adjunto.setFileName("adjunto");
// Una MultiParte para agrupar texto e imagen.
MimeMultipart multiParte = new MimeMultipart();
multiParte.addBodyPart(texto);
multiParte.addBodyPart(adjunto);

Message msg = getMessage(session, from, to, multiParte);
System.out.println ("Enviando ..." );

Transport.send(msg);

System.out.println ("Mensaje enviado!");
return ("Mensaje enviado!");

}
catch (Exception e)
{

ExceptionManager.ManageException(e);
return ("Mensaje no se pudo enviar!" +e);
}
}

private static MimeMessage getMessage(Session session, String from, String to, MimeMultipart mensaje)
{

```



```

try{

MimeMessage msg = new MimeMessage(session);
//msg.setText(mensaje);
msg.addRecipient(Message.RecipientType.TO, new InternetAddress(to));
msg.setFrom(new InternetAddress(from,"Seguros Balboa"));
msg.setSubject("Notificación - Seguros Balboa");
msg.setContent(mensaje);
return msg;
}
catch (java.io.UnsupportedEncodingException ex)
{
ExceptionManager.ManageException(ex);
return null;
}
catch (MessagingException ex)
{
ExceptionManager.ManageException(ex);
return null;
}
}
}
}
private static MimeMessage getMessage(Session session, String from, String to, String mensaje)
{

try{

MimeMessage msg = new MimeMessage(session);
msg.setText(mensaje);
msg.addRecipient(Message.RecipientType.TO, new InternetAddress(to));
msg.setFrom(new InternetAddress(from,"Seguros Balboa"));
msg.setSubject("Notificación - Seguros Balboa");
return msg;

}
catch (java.io.UnsupportedEncodingException ex)
{

ExceptionManager.ManageException(ex);
return null;

}
catch (MessagingException ex)
{

ExceptionManager.ManageException(ex);
return null;
}
}
}
}

```

## Clase SMS

Clase de encargada de preparar el envío y realizar la gestión de envío del sms, por medio de una conexión http.

```
package GestorNotificaciones;

import org.apache.commons.httpclient.*;
import org.apache.commons.httpclient.methods.*;
import javax.servlet.jsp.*;
import javax.servlet.http.*;

import java.sql.CallableStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.*;

public class sms extends Bean {

    public sms (PageContext p_pageContext,String usuario,String password)throws Exception,
    SQLException {
        super(p_pageContext,usuario,password);
    }

    @SuppressWarnings("deprecation")
    public String testSetCookieGet(String numero, String mensaje) throws Exception {

        //Se inicia el objeto HTTP
        System.out.println("Inicio");
        HttpClient client = new HttpClient();
        client.setStrictMode(true);
        //Se fija el tiempo máximo de espera de la respuesta del servidor
        client.setTimeout(60000);
        //Se fija el tiempo máximo de espera para conectar con el servidor
        client.setConnectionTimeout(5000);
        PostMethod post = null;
        //Se fija la URL sobre la que enviar la petición POST

        //Como ejemplo se supone www.pasarela_push_altiria.com/post/sms
        System.out.println("http");
        post = new PostMethod("http://extern.nrs-
group.com/send_multisims.php?user=042687185&pass=luisao&msg="+mensaje+"&phone="+numero+"&re
mit=59395681326&country=ec");

        //Se fija la codificación de caracteres en la cabecera de la petición
        post.setRequestHeader("Content-type", "application/x-www-form-urlencoded; charset=UTF-8");

        //Se crea la lista de parámetros a enviar en la petición POST
        System.out.println("095681326");
        NameValuePair[] parametersList = new NameValuePair[6];
        parametersList[0] = new NameValuePair("msg", "Mensaje de prueba");
```

```

parametersList[1] = new NameValuePair("phone", "59395681326");
parametersList[2] = new NameValuePair("remit", "59395681326");
parametersList[3] = new NameValuePair("user", "042687185");
parametersList[4] = new NameValuePair("pass", "luisao");
parametersList[5] = new NameValuePair("country", "ec");

//Se rellena el cuerpo de la petición POST con los parámetros
//post.setRequestBody(parametersList);
int httpstatus = 0;
String response = null;

try {
//Se envía la petición
System.out.println(post.getURI());
httpstatus = client.executeMethod(post);
//Se consigue la respuesta
response = post.getResponseBodyAsString();
}
catch (Exception e) {
//Habrà que prever la captura de excepciones
String error = "Error al enviar sms";
return error;
}
finally {
//En cualquier caso se cierra la conexión
post.releaseConnection();
}

//Habrà que prever posibles errores en la respuesta del servidor
if (httpstatus!=200){
String error1 = "Error al enviar sms";
return error1;
}
else {
System.out.println("Respuesta "+response);
return(response);
//Se procesa la respuesta capturada en la cadena "response"
}
}
}

```

## Clase Proceso

Clase que realiza los insert, select, update, delete, de los mensajes que se envían.

```

package GestorNotificaciones;
import java.sql.*;

import javax.servlet.jsp.*;
import javax.servlet.http.*;

```

```

import java.util.*;

public class Procesos extends Bean
{
    //Constructor
    public Procesos(PageContext p_pageContext,String usuario,String password)throws Exception,
SQLException {
        super(p_pageContext,usuario,password);
    }

    public String SecuenciaActual()throws Exception, SQLException
    {
        ResultSet lrs=m_conn.prepareStatement("select SEC_PROCESOS.currval + 1 from
dual").executeQuery();
        lrs.next();
        String sec = Long.toString(lrs.getLong(1));
        lrs.close();
        return sec;
    }

    public String SecuenciaSiguiente()throws Exception, SQLException
    {
        ResultSet lrs=m_conn.prepareStatement("select SEC_PROCESOS.nextval from
dual").executeQuery();

        lrs.next();
        String sec = Long.toString(lrs.getLong(1));
        lrs.close();
        return sec;
    }

    public Vector consulta()throws Exception, SQLException
    {
        Vector v = new Vector();
        ResultSet lrs=m_conn.prepareStatement(" select id, "
            +" nombre_proceso, "
            +" descripcion "
            +" from gn_proceso "
            +" where
masivo = 'N' "
            +" order by
to_number(id)").executeQuery();
        while (lrs.next())
        {
            v.addElement(new elementoVector("1",lrs.getString(1),
lrs.getString(2),
lrs.getString(3)));
        }
        return v;
    }

    public Vector consulta2()throws Exception, SQLException
    {
        Vector v = new Vector();
        ResultSet lrs=m_conn.prepareStatement(" select id, "
            +" nombre_proceso, "
            +" descripcion "

```

```

        +" from gn_proceso "
        +" where masivo = 'S' "
+" order by to_number(id)").executeQuery();
        while (lrs.next())
        {
                v.addElement(new elementoVector("1",lrs.getString(1),
lrs.getString(2),
lrs.getString(3)));
        }
        return v;
}

public Vector consulta3()throws Exception, SQLException
{
        Vector v = new Vector();
        ResultSet lrs=m_conn.prepareStatement(" select id, "
                +" nombre_proceso, "
                +" descripcion "
                +" from gn_proceso "
                +" order by to_number(id)").executeQuery();
        while (lrs.next())
        {
                v.addElement(new elementoVector("1",lrs.getString(1),
                lrs.getString(2),
                lrs.getString(3)));
        }
        return v;
}

public Vector consulta4()throws Exception, SQLException
{
        Vector v = new Vector();
        ResultSet lrs=m_conn.prepareStatement(" select id, "
                +" nombre "
                +" from gn_estado "
                +" order by to_number(id)").executeQuery();
        while (lrs.next())
        {
                v.addElement(new elementoVector("1",lrs.getString(1),
                lrs.getString(2)));
        }
        return v;
}

public void resultEnvioEspecif(Vector envios)throws Exception, SQLException
{
        String id = "";
        String result_sms = "";
        String result_mail = "";

```

```

        for (int i=0;i<envios.size();i++)
        {
            id = ((elementoVector)envios.elementAt(i)).elemento1;
            result_sms = ((elementoVector)envios.elementAt(i)).elemento3;
            result_mail = ((elementoVector)envios.elementAt(i)).elemento4;

            m_conn.prepareStatement("update gn_transaccionesenviosespecific "
                +" set descripcionresultsms = '"+result_sms+"', "
                +" descripcionresultmail = '"+result_mail+"' "
                +" where id = id).executeUpdate();m_conn.commit();
        }
    }

    public void resultEnvioGeneral(Vector envios)throws Exception, SQLException
    {
        String id = "";
        String result_sms = "";
        String result_mail = "";

        for (int i=0;i<envios.size();i++)
        {
            id = ((elementoVector)envios.elementAt(i)).elemento1;
            result_sms = ((elementoVector)envios.elementAt(i)).elemento3;
            result_mail = ((elementoVector)envios.elementAt(i)).elemento4;

            m_conn.prepareStatement("update gn_transaccionesenviosgeneral "
                +" set descripcionresultsms = '"+result_sms+"', "
                +" descripcionresultmail = '"+result_mail+"' "
                +" where id = "+id).executeUpdate();
            m_conn.commit();
        }
    }
}

```

## Clase SMTPAuthentication

Clase que gestiona la autenticación del mail con usuario y contraseña de la cuenta de correo que será el destinatario

```

package GestorNotificaciones;

import javax.mail.PasswordAuthentication;

public class SMTPAuthentication extends javax.mail.Authenticator
{

```

```
public PasswordAuthentication getPasswordAuthentication()
{

String username = "webmaster@customersoft.net";
String password = "admin123";
System.out.println("antes PasswordAuthentication");
return new PasswordAuthentication(username, password);

}

}
```

## Clase tipo\_proceso

Clase que gestiona el ingreso, consulta, modificación y eliminación de los tipos de procesos.

Para el proyecto son dos principalmente:

El proceso que hace el barrido de las pólizas que están por vencer, y envía el mensaje a los clientes **PR\_POLIZA**

El proceso que hace el barrido de los pagos de cuotas y letras que están por vencer, y envía el mensaje a los clientes **PR\_CARTERA**

```
package GestorNotificaciones;

import java.sql.*;
import javax.servlet.jsp.*;
import javax.servlet.http.*;
```

```

import java.util.*;

public class tipo_proceso extends Bean
{
    //Constructor
    public tipo_proceso(PageContext p_pageContext,String usuario,String
        password)throws Exception, SQLException {
        super(p_pageContext,usuario,password);
    }

    public String SecuenciaActual()throws Exception, SQLException
    {
        ResultSet lrs=m_conn.prepareStatement("select
            SEC_TIPO_PROCESO.currval + 1 from dual").executeQuery();
        lrs.next();
        String sec = Long.toString(lrs.getLong(1));
        lrs.close();
        return sec;
    }

    public String SecuenciaSiguiente()throws Exception, SQLException
    {
        ResultSet lrs=m_conn.prepareStatement("select
            SEC_TIPO_PROCESO.nextval from dual").executeQuery();
        lrs.next();
        String sec = Long.toString(lrs.getLong(1));
        lrs.close();
        return sec;
    }

    public Vector consulta()throws Exception, SQLException
    {
        Vector v = new Vector();
        ResultSet lrs=m_conn.prepareStatement(" select id, "
            + " nombre_proceso, "
            + " descripcion "
            + " from gn_proceso "
            + " order by to_number(id)").executeQuery();
        while (lrs.next())
        {
            v.addElement(new elementoVector("1",lrs.getString(1),
                lrs.getString(2),
                lrs.getString(3)));
        }
        return v;
    }

    //Grabar -- Opción Guardar
    public void graba(Vector rtipos)throws Exception, SQLException
    {
        String band = " ";
    }
}

```



```

        String codigo_proceso;
        String nombre_proceso;
        String descripcion_proceso;

        for (int i=0;i<rtipos.size();i++)
        {
            band = ((elementoVector)rtipos.elementAt(i)).elemento1;
            codigo_proceso = ((elementoVector)rtipos.elementAt(i)).elemento2;
            nombre_proceso = ((elementoVector)rtipos.elementAt(i)).elemento3;
            descripcion_proceso = ((elementoVector)rtipos.elementAt(i)).elemento4;

            if (band.equals("0"))
            {
                m_conn.prepareStatement("insert into gn_proceso (id, "
                    + " nombre_proceso, "
                    + " descripcion) "
                    + " values (" +codigo_proceso+", "
                    + " "+nombre_proceso+", "
                    + " "+descripcion_proceso+")
                    ").executeUpdate();
            }
            if (band.equals("1"))
            {
                m_conn.prepareStatement("update gn_proceso "
                    + " set nombre_proceso = "+nombre_proceso+", "
                    + " descripcion = "+descripcion_proceso+" "
                    + " where id = "+codigo_proceso+"").executeUpdate();
            }
            if (band.equals("2"))
            {
                m_conn.prepareStatement("delete from gn_proceso "
                    + " where id = "+codigo_proceso+"").executeUpdate();
            }
        }

        m_conn.commit();
    }
}

```

### **Clase tipo\_mensaje**

Clase que gestiona el ingreso, consulta, modificación y eliminación de los mensajes. Esta clase permite parametrizar que mensajes se van a enviar, con la respectiva programación de envió es decir: intervalos de tiempo, cantidad de envíos, fecha de inicio de envió del mensaje, fecha final, en texto del mensaje, etc.

```

package GestorNotificaciones;

import java.sql.*;
import javax.servlet.jsp.*;
import javax.servlet.http.*;
import java.util.*;

public class tipo_mensaje extends Bean
{
    //Constructor//
    public tipo_mensaje(PageContext p_pageContext,String usuario,String password)throws
Exception, SQLException {
        super(p_pageContext,usuario,password);
    }

    public String SecuenciaActual()throws Exception, SQLException
    {
        ResultSet lrs=m_conn.prepareStatement("select
SEC_TIPO_MENSAJES.currval + 1 from dual").executeQuery();
        lrs.next();
        String sec = Long.toString(lrs.getLong(1));
        lrs.close();
        return sec;
    }

    public String SecuenciaSiguiente()throws Exception, SQLException
    {
        ResultSet lrs=m_conn.prepareStatement("select
SEC_TIPO_MENSAJES.nextval from dual").executeQuery();
        lrs.next();
        String sec = Long.toString(lrs.getLong(1));
        lrs.close();
        return sec;
    }

    public Vector consulta()throws Exception, SQLException
    {
        Vector v = new Vector();
        ResultSet lrs=m_conn.prepareStatement(" select m.id, "
            +"          m.gn_procesoid, "
            +"          p.nombre_proceso , "
            +"          m.intervalo_ejec, "
            +"          m.dias_restantes, "
            +"          m.cant_envios, "
            +"          m.gn_tipoclienteid, "
            +"          tc.descripcion, "
            +"          m.mensaje, "
            +"          m.envia_sms, "
            +"          m.envia_mail, "
            +"          to_char(m.fechainicio,'DD/MM/YY HH12:MI:SS'), "

```

```

+"      to_char(m.fechafin,'DD/MM/YY HH12:MI:SS') , "
+"          m.gn_estadoid, "
+"          e.nombre, "
+"          m.adjunto_mail "
+"      from gn_mensaje m, gn_proceso p, "
+"          gn_tipocliente tc, gn_estado e "
+"  where m.gn_procesoid=p.id "
+"        and m.gn_procesoid=p.id "
+"        and m.gn_tipoclienteid=tc.id "
+"        and m.gn_estadoid=e.id "
+"  order by to_number(m.id)".executeQuery();
      while (lrs.next())
      {
          v.addElement(new elementoVector("1",lrs.getString(1),
                                          lrs.getString(2),
                                          lrs.getString(3),
                                          lrs.getString(4),
                                          lrs.getString(5),
                                          lrs.getString(6),
                                          lrs.getString(7),
                                          lrs.getString(8),
                                          lrs.getString(9),
                                          lrs.getString(10),
                                          lrs.getString(11),
                                          lrs.getString(12),
                                          lrs.getString(13),
                                          lrs.getString(14),
                                          lrs.getString(15),
                                          lrs.getString(16)));
      }
      return v;
  }

```

//Grabar -- Opción Guardar

```
public void graba(Vector rtipos)throws Exception, SQLException
```

```
{
```

```

    String band = " ";
    String codigo_mensaje;
    String procesoid;
    String intervalo;
    String dias_restantes;
    String cant_envios;
    String tipoclienteid;
    String mensaje;
    String envia_sms;
    String envia_mail;
    String fechainicio;
    String fechafin;
    String estadoid;
    String adjunto_mail;
    //String adjunto_sms;

```

```
    for (int i=0;i<rtipos.size();i++)
```

```
{
```

```

band = ((elementoVector)rtipos.elementAt(i)).elemento1;
codigo_mensaje =
((elementoVector)rtipos.elementAt(i)).elemento2;
procesoid = ((elementoVector)rtipos.elementAt(i)).elemento3;
intervalo = ((elementoVector)rtipos.elementAt(i)).elemento4;
dias_restantes = ((elementoVector)rtipos.elementAt(i)).elemento5;
cant_envios = ((elementoVector)rtipos.elementAt(i)).elemento6;
tipoclienteid = ((elementoVector)rtipos.elementAt(i)).elemento7;
mensaje = ((elementoVector)rtipos.elementAt(i)).elemento8;
envia_sms = ((elementoVector)rtipos.elementAt(i)).elemento9;
envia_mail = ((elementoVector)rtipos.elementAt(i)).elemento10;
fechainicio = ((elementoVector)rtipos.elementAt(i)).elemento11;
fechafin = ((elementoVector)rtipos.elementAt(i)).elemento12;
estadoid = ((elementoVector)rtipos.elementAt(i)).elemento13;
adjunto_mail = ((elementoVector)rtipos.elementAt(i)).elemento14;
//adjunto_sms = ((elementoVector)rtipos.elementAt(i)).elemento15;

```

```

if (band.equals("0"))
{
    m_conn.prepareStatement("insert into gn_mensaje (id, "
        + "                                gn_procesoid, "
        + "                                intervalo_ejec, "
        + "                                dias_restantes, "
        + "                                cant_envios, "
        + "                                gn_tipoclienteid, "
        + "                                mensaje, "
        + "                                envia_sms, "
        + "                                envia_mail, "
        + "                                fechainicio, "
        + "                                fechafin, "
        + "                                gn_estadoid, "
        + "                                adjunto_mail) "
        + " values (" +codigo_mensaje+", "
        + "          "+procesoid+", "
        + "          "+intervalo+", "
        + "          "+dias_restantes+", "
        + "          "+cant_envios+", "
        + "          "+tipoclienteid+", "
        + "          "+mensaje+", "
        + "          "+envia_sms+", "
        + "          "+envia_mail+", "
        + "          "+fechainicio+", "
        + "          "+fechafin+", "
        + "          "+estadoid+", "
        + "          "+adjunto_mail+" ") .executeUpdate();
}
if (band.equals("1"))
{
    m_conn.prepareStatement("update gn_mensaje "
        + " set gn_procesoid = "+procesoid+", "
        + "     intervalo_ejec = "+intervalo+", "
        + "     dias_restantes = "+dias_restantes+", "

```

```

        +" cant_envios = "+cant_envios+", "
        +" gn_tipoclienteid = "+tipoclienteid+", "
        +" mensaje = "+mensaje+", "
        +" envia_sms = "+envia_sms+", "
        +" envia_mail = "+envia_mail+", "
        +" fechainicio = "+fechainicio+", "
        +" fechafin = "+fechafin+", "
        +" gn_estadoid = "+estadoid+", "
        +" adjunto_sms = "+adjunto_mail+" "
+" where id = "+codigo_mensaje+"").executeUpdate();
    }
    if (band.equals("2"))
    {
        m_conn.prepareStatement("delete from gn_mensaje "
        +" where id = "+codigo_mensaje+"").executeUpdate();
    }
    }
    m_conn.commit();
}
}

```

### Clase tipo\_variable

Clase que gestiona el ingreso, consulta, modificación y eliminación de las variables. Estas variables son utilizadas en la construcción del mensaje que se utilizará para enviar vía mail o sms. El mensaje contendrá textos fijos y las variables que se adicionen en el mismo, dependiendo de la necesidad por ejemplo:

#### Estructura de un mensaje

<p>POLIZA - Estimados Empresa <b>#cliente#</b> su póliza de <b>#ramo#</b> No <b>#poliza#</b> esta por vencer el <b>#vigenciahasta#</b>. Nuestra compañía estará gustoso en gestionar su renovación.</p>
---

Dentro de la estructura del mensaje encontramos 4 variables: **#cliente#**, **#ramo#**, **#poliza#**, **#vigenciahasta#**; estas variables son reemplazadas por los valores correspondientes, al momento del envío.

```
package GestorNotificaciones;

import java.sql.*;
import javax.servlet.jsp.*;
import javax.servlet.http.*;
import java.util.*;

public class tipo_variable extends Bean
{
    //Constructor
    public tipo_variable(PageContext p_pageContext,String usuario,String password)throws
        Exception, SQLException {
        super(p_pageContext,usuario,password);
    }

    public String SecuenciaActual()throws Exception, SQLException
    {
        ResultSet lrs=m_conn.prepareStatement("select SEC_TIPO_VARIABLE.currval +
            1 from dual").executeQuery();
        lrs.next();
        String sec = Long.toString(lrs.getLong(1));
        lrs.close();
        return sec;
    }

    public String SecuenciaSiguiente()throws Exception, SQLException
    {
        ResultSet lrs=m_conn.prepareStatement("select SEC_TIPO_VARIABLE.nextval
            from dual").executeQuery();
        lrs.next();
        String sec = Long.toString(lrs.getLong(1));
        lrs.close();
        return sec;
    }

    public Vector consulta()throws Exception, SQLException
    {
        Vector v = new Vector();
        ResultSet lrs=m_conn.prepareStatement(" select id, "
            +" nombre_variable, "
            +" descripcion, "
            +" referencia "
            +" from gn_variables "
            +" order by to_number(id)").executeQuery();
```

```

        while (lrs.next())
        {
            v.addElement(new elementoVector("1",lrs.getString(1),
                lrs.getString(2),
                lrs.getString(3),
                lrs.getString(4)));
        }
        return v;
    }

//Grabar -- Opción Guardar
public void graba(Vector rtipos)throws Exception, SQLException
{
    String band = " ";
    String codigo_variable;
    String nombre_variable;
    String descripcion;
    String referencia;

    for (int i=0;i<rtipos.size();i++)
    {
        band = ((elementoVector)rtipos.elementAt(i)).elemento1;
        codigo_variable = ((elementoVector)rtipos.elementAt(i)).elemento2;
        nombre_variable = ((elementoVector)rtipos.elementAt(i)).elemento3;
        descripcion = ((elementoVector)rtipos.elementAt(i)).elemento4;
        referencia = ((elementoVector)rtipos.elementAt(i)).elemento5;

        if (band.equals("0"))
        {
            m_conn.prepareStatement("insert into gn_variables (id, "
                +"          nombre_variable, "
                +"          descripcion, "
                +"          referencia) "
                +" values (""+codigo_variable+"", "
                +"          ""+nombre_variable+"", "
                +"          ""+descripcion+"", "
                +"          ""+referencia+"") ").executeUpdate();
        }

        if (band.equals("1"))
        {
            m_conn.prepareStatement("update gn_variables "
                +" set nombre_variable = ""+nombre_variable+"", "
                +" descripcion = ""+descripcion+"", "
                +" referencia = ""+referencia+" "
                +" where id = ""+codigo_variable+""").executeUpdate();
        }

        if (band.equals("2"))
        {
            m_conn.prepareStatement("delete from gn_variables "
                +" where id = ""+codigo_variable+""").executeUpdate();
        }
    }

    m_conn.commit();
}

```

```
}  
}
```

A continuación se describirán las páginas jsp más importantes del proyecto

### **Página p\_login.jsp**

Este jsp muestra la página de inicio de la aplicación, en la cual se ingresa usuario y contraseña

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">  
  
<% @ page import="java.util.*" %>  
<% @ page import="java.io.*" %>  
<% @ page import="GestorNotificaciones.*" %>  
  
<%  
    String usuario = (String)request.getParameter("usuario");  
    String clave = (String)request.getParameter("clave");  
    String cont = (String)request.getParameter("cont");  
  
    int c = Integer.parseInt(cont);  
    session.setAttribute("usuario",usuario);  
    session.setAttribute("clave",clave);  
%>  
  
<head>  
<TITLE> Notifications Manager </TITLE>  
  
<script>  
    function Aceptar(cont)  
    {  
        var usuario = document.login.usuario.value;  
        var clave = document.login.clave.value;  
        document.login.action="p_login.jsp?usuario="+usuario+"&clave="+clave+"&cont="+cont;  
        document.login.target="MAIN";  
        document.login.submit();  
    }  
    function Cancelar()
```



```

    {
        window.close();
    }
</script>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"><style type="text/css">
<!--
body {
    background-color: #FFFFFF;
}
.adde {
    left: 0px;
    top: 0px;
    right: 0px;
    bottom: 0px;
    position: absolute;
}
.adde2 {
    position: absolute;
    left: 0px;
    top: 118px;
    right: 0px;
    bottom: 0px;
}
-->
</style>
</head>

<HTML>

<BODY background=" ../imagenes/Fondo3.JPG">

    <% if (c > 0) {
        try{
            Bean b = new Bean(pageContext,usuario,clave);
        %>

        <script>
            window.open("f_bienvenida.jsp",target="MAIN", "toolbar=YES,resizable=NO, left=0px
top=0px, width=800px, height=100px");
        </script>

    <%     } catch(Exception e) { %>

        <div id="Layer4" style="position:absolute; left:360px; top:300px; width:150px; height:30px">
        <font color="blue" face="Arial" size="3" >
        <b>Usuario o Clave incorrecta</b>
        </font>
        </div>

    <%         c = c + 1;
        }

```

```

    }
    else {
        c = c + 1;
    }

    if (c > 3) { %>
<script>
        window.close();
</script>

    <% } %>

    <div align="center" class="ade"></div>
    <form name="login" method="post">
    <div id="Layer4" style="position:absolute; left:320px; top:150px; width:50px; height:30px">
    <font color="blue" face="Arial" size="2" >
        <b>Usuario</b>
    </font>
    </div>

    <div id="Layer4" style="position:absolute; left:400px; top:150px; width:50px; height:30px">
    <input type="text" name="usuario" >
    </div>
    <script language="JavaScript">
        login.usuario.focus();
    </script>

    <div id="Layer4" style="position:absolute; left:320px; top:185px; width:50px; height:30px">
    <font color="blue" face="Arial" size="2" >
        <b>Clave</b>
    </font>
    </div>

    <div id="Layer4" style="position:absolute; left:400px; top:185px; width:50px; height:30px">
    <input type="password" name="clave">
    </div>

    <div id="Layer4" style="position:absolute; left:320px; top:260px; width:50px; height:30px">
    <a href="javascript: Aceptar(<%=c%>)"></a>
    </div>

    <div id="Layer4" style="position:absolute; left:445px; top:260px; width:50px; height:30px">
    <!-- <input type="button" value="Cancelar" name="Cancelar" onclick="Cancelar()"/>-->
    <a href="javascript: Cancelar()"></a>
    </div>

    </form>
    </BODY>
    </HTML>

```

## Página p\_cab\_avisos.jsp

Esta página muestra la cabecera donde se escoge en un combo el tipo de proceso (envió de mensajes por vencimiento de cartera o por vencimiento de póliza), el proceso internamente busca los registros que cumplan con la condición de vencimiento y muestra el detalle de los registros en otra página que se muestra en la pagina inferior llamada p\_det\_avisos.jsp.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<% @ page import="java.io.*" %>
<% @ page import="java.util.*" %>
<% @ page import="GestorNotificaciones.*" %>

<%
    String usuario = (String)session.getAttribute("usuario");
    String clave = (String)session.getAttribute("clave");

    try{
        Bean b = new Bean(pageContext,usuario,clave);

        Procesos p = new Procesos(pageContext,usuario,clave);
        Vector procesos = new Vector();
        procesos = p.consulta();

    %>

<HTML>
<HEAD>
<TITLE> Cabecera Envío Avisos </TITLE>

<style>
table.helpT
{ text-align: center;
font-family: Verdana;
font-weight: normal;
font-size: 11px;
```

```
color: #404040;
width: 500px;
background-color: #fafafa;
border: 1px #6699CC solid;
border-collapse: collapse;
border-spacing: 0px; }
```

```
td.helpHed
{ border-bottom: 2px solid #F0F3F5;
border-left: 1px solid #F0F3F5;
background-color: #627685;
text-align: left;
text-indent: 5px;
font-family: Verdana;
font-weight: bold;
font-size: 11px;
color: #F0F3F5; }
```

```
td.helpBod
{ border-bottom: 1px solid #9CF;
border-top: 0px;
border-left: 1px solid #9CF;
border-right: 0px;
text-align: left;
text-indent: 10px;
font-family: Verdana, sans-serif, Arial;
font-weight: normal;
font-size: 11px;
color: #404040;
background-color: #fafafa; }
```

```
table.soft
{ text-align: center;
font-family: Verdana;
font-weight: normal;
font-size: 11px;
color: #404040;
width: 280px;
background-color: #BAC7D0;
border: 1px #6699CC solid;
border-collapse: collapse;
border-spacing: 0px; }
</style>
```

```
<script>
    function Aceptar()
    {
        var indice = document.forms.cab_avisos.proceso.options.selectedIndex;
        var cod_proceso = document.forms.cab_avisos.proceso.options[indice].value;

        window.open("p_det_avisos.jsp?opcion=A&cod_proceso="+cod_proceso,target="p_det_avisos");
    }
</script>
```

```

</HEAD>

<BODY background="../imagenes/admin_r6_c1.jpg">

<form name="cab_avisos">

<table border="1" name="procesos" class="soft" cellspacing="0">
  <tr>
    <TH COLSPAN="2"><FONT FACE="Arial" SIZE="4"><b>ENVIOS DE AVISOS - PROCESOS
    ESPECIFICOS</b></FONT> </TH>
  </tr>
  <tr>
    <td class="helpHed">
      <b>Proceso</b>
      <font>
    </td>
    <td class="sup">
      <select name="proceso">
        <% for (int i = 0; i < procesos.size(); i++) { %>
          <option
value=<%=((elementoVector)procesos.elementAt(i)).elemento2%>><%=((elementoVector)procesos.elementAt(i)).ele
mento4%>
          <% } %>
        </select>
      </td>
  </tr>
  <tr>
    <td>
      <a href="javascript: Aceptar()"></a>
    </td>
  </tr>
</table>

</form>

<% } catch(Exception e) { %>

<BODY>

<div id="Layer4" style="position:absolute; left:100px; top:50px; width:300px; height:30px">
<font color="red" face="Arial" size="3">
<b>No se ha conectado correctamente. Acceso Denegado.</b>
<b><%=e%></b>
</font>
</div>

<% } %>

</BODY>
</HTML>

```

## Página p\_det\_avisos.jsp

En esta página se muestra el detalle de los registros encontrados de acuerdo al proceso seleccionado en la página anterior y al criterio de búsqueda y fechas de vencimientos establecidos en la parametrización de mensajes y programado en los procesos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<% @ page import="java.io.*" %>
<% @ page import="java.util.*" %>
<% @ page import="GestorNotificaciones.*" %>

<%
String usuario = (String)session.getAttribute("usuario");
String clave = (String)session.getAttribute("clave");

try{
    Bean b = new Bean(pageContext,usuario,clave);

    String cod_proceso = request.getParameter("cod_proceso");
    String band = "0";
    String band2 = "0";
    if (cod_proceso == null)
        cod_proceso = (String)session.getAttribute("cod_proceso");
    String opcion = request.getParameter("opcion");
    if (opcion == null){
        opcion = (String)session.getAttribute("opcion");
        band = (String)session.getAttribute("band");
        band2 = (String)session.getAttribute("band2");
    }

    session.setAttribute("cod_proceso",cod_proceso);
    session.setAttribute("opcion",opcion);

    transEnvios t = new transEnvios(pageContext,usuario,clave);
    Vector envios = new Vector();
    envios = t.consulta(cod_proceso);
    session.setAttribute("envios",envios);

    if (opcion.equals("A"))
    {
        t.creaJob(cod_proceso);
        band = "1";
        session.setAttribute("band",band);
    }
}
```

```

        session.setAttribute("band2",band2);
        opcion = "B";
        session.setAttribute("opcion",opcion);
    }

    String id_job = t.consultaJob();

%>

<HTML>
<HEAD>
<TITLE> Detalle Envío Avisos </TITLE>

<style>
table.helpT
{ text-align: center;
font-family: Verdana;
font-weight: normal;
font-size: 11px;
color: #404040;
width: 500px;
background-color: #fafafa;
border: 1px #6699CC solid;
border-collapse: collapse;
border-spacing: 0px; }

td.helpHed
{ border-bottom: 2px solid #F0F3F5;
border-left: 1px solid #F0F3F5;
background-color: #627685;
text-align: left;
text-indent: 5px;
font-family: Verdana;
font-weight: bold;
font-size: 11px;
color: #F0F3F5; }

td.helpBod
{ border-bottom: 1px solid #9CF;
border-top: 0px;
border-left: 1px solid #9CF;
border-right: 0px;
text-align: left;
text-indent: 10px;
font-family: Verdana, sans-serif, Arial;
font-weight: normal;
font-size: 11px;
color: #404040;
background-color: #fafafa; }

table.sofT
{ text-align: center;
font-family: Verdana;

```

```
font-weight: normal;
font-size: 11px;
color: #404040;
width: 280px;
background-color: #BAC7D0;
border: 1px #6699CC solid;
border-collapse: collapse;
border-spacing: 0px; }
</style>
```

```
<SCRIPT>
function Enviar()
{
    window.open("p_envio.jsp?tipo=E",target="cuerpo");
}
</SCRIPT>
```

```
<%if (band.equals("1")||band2.equals("2")){ %>
    <META HTTP-EQUIV="REFRESH" CONTENT="5; URL=p_det_avisos.jsp">
    <%if (!(id_job.equals("90"))){
        band = "0";
        if (band2.equals("2"))
        {
            band2 = "3";
        }
        else
        {
            band2 = "2";
        }
        session.setAttribute("band",band);
    }
}%>
```

```
</HEAD>
```

```
<BODY background="../imagenes/admin_r6_c1.jpg">
```

```
<form name="det_avisos">
```

```
<table border="1" name="tipo" class="sofT" cellspacing="0">
```

```
<%
```

```
    for (int i = 0; i < envios.size(); i++){
```

```
%>
```

```
<tr>
```

```
<td class="sup">
```

```
<input type="text" name="id" value="<%=((elementoVector)envios.elementAt(i)).elemento1%>"
size="3" style="font-family:Verdana; border:0px blue solid; background-color:#BAC7D0;" readonly>
```

```
</td>
```

```
<td class="sup">
```

```
<input type="text" name="cliente"
value="<%=((elementoVector)envios.elementAt(i)).elemento3%>" size="30" style="font-family:Verdana; border:0px
blue solid; background-color:#BAC7D0;" readonly>
```

```
</td>
```



```

        <td class="sup">
            <input type="text" name="poliza"
value="<%=((elementoVector)envios.elementAt(i)).elemento2%>" size="10" style="font-family:Verdana; border:0px
blue solid; background-color:#BAC7D0;" readonly>
        </td>
        <td class="sup">
            <input type="text" name="ramo"
value="<%=((elementoVector)envios.elementAt(i)).elemento4%>" size="25" style="font-family:Verdana; border:0px
blue solid; background-color:#BAC7D0;" readonly>
        </td>
        <td class="sup">
            <input type="text" name="vigencia"
value="<%=((elementoVector)envios.elementAt(i)).elemento5%>" size="10" style="font-family:Verdana; border:0px
blue solid; background-color:#BAC7D0;" readonly>
        </td>
        <td class="sup">
            <input type="text" name="celular"
value="<%=((elementoVector)envios.elementAt(i)).elemento6%>" size="10" style="font-family:Verdana; border:0px
blue solid; background-color:#BAC7D0;" readonly>
        </td>
        <td class="sup">
            <input type="text" name="mail"
value="<%=((elementoVector)envios.elementAt(i)).elemento7%>" size="25" style="font-family:Verdana; border:0px
blue solid; background-color:#BAC7D0;" readonly>
        </td>
        <td class="sup">
            <textarea name="mensaje" rows="2"
cols="20"><%=((elementoVector)envios.elementAt(i)).elemento8%>
            </textarea>
        </td>
        <td class="sup">
            <input type="checkbox" name="escoger_sms" checked>
        </td>
        <td class="sup">
            <input type="checkbox" name="escoger_mail" checked>
        </td>
    </tr>
<%
}
%>

<tr>
<td></td>
<td class="sup">
    <a href="javascript: Enviar()"></a>
</td>
</tr>
</table>

<% }catch(Exception e) { %>

<u><BODY>

```

```
<div id="Layer4" style="position:absolute; left:100px; top:50px; width:300px; height:30px">
<font color="red" face="Arial" size="3">
<b>No se ha conectado correctamente. Acceso Denegado.</b>
<b><%=e%></b>
</font>
</div>

<% }%>

</BODY>
</HTML>
```

## **Página p\_envio.jsp**

Este jsp se encarga de realizar la invocación a las clases que realizan los envíos por mail y sms, realiza validaciones de tamaño del mensaje para el caso de sms, y muestra el resultado del envío si ha sido satisfactorio o no.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<% @ page import="java.io.*" %>
<% @ page import="java.util.*" %>
<% @ page import="GestorNotificaciones.*" %>

<%
String usuario = (String)session.getAttribute("usuario");
String clave = (String)session.getAttribute("clave");

try{
    Bean b = new Bean(pageContext,usuario,clave);

    String tipo = request.getParameter("tipo");
```

```

sms s = new sms(pageContext,usuario,clave);
SendAuthentication a = new SendAuthentication(pageContext,usuario,clave);
SendAuthentication2 z = new SendAuthentication2(pageContext,usuario,clave);
Procesos p = new Procesos(pageContext,usuario,clave);

```

```

Vector envios = new Vector();
Vector envios_mail_sms = new Vector();

```

```

envios = (Vector)session.getAttribute("envios");
String resultadoSMS = "";
String resultadoMAIL = "";
String sCadenaSinBlancos = "";
String cadena = "";
String band = "0";

```

```

for (int i = 0; i < envios.size(); i++){
    band = "0";
    sCadenaSinBlancos = "";
    for (int x=0; x < (((elementoVector)envios.elementAt(i)).elemento8).length(); x++) {
        if (((elementoVector)envios.elementAt(i)).elemento8.charAt(x) != ' ')
            sCadenaSinBlancos += (((elementoVector)envios.elementAt(i)).elemento8).charAt(x);
        else
            sCadenaSinBlancos += "%20";
    }
    do
    {
        if (!(sCadenaSinBlancos.equals(""))){
            if (sCadenaSinBlancos.length() > 155){
                cadena = sCadenaSinBlancos.substring(1,155);
                sCadenaSinBlancos = sCadenaSinBlancos.substring(156);
            }else{
                band = "1";
                cadena = sCadenaSinBlancos;
                sCadenaSinBlancos = "";
            }
            resultadoSMS = s.testSetCookieGet(

```

```

((elementoVector)envios.elementAt(i)).elemento6,

```

```

                cadena);

```

```

        }
    }while (band.equals("0"));

```

```

if (((elementoVector)envios.elementAt(i)).elemento9) == null){
    resultadoMAIL = z.Send(
        ((elementoVector)envios.elementAt(i)).elemento7,
        ((elementoVector)envios.elementAt(i)).elemento8);
}
else{
    resultadoMAIL = a.Send(
        ((elementoVector)envios.elementAt(i)).elemento7,
        ((elementoVector)envios.elementAt(i)).elemento8,
        ((elementoVector)envios.elementAt(i)).elemento9);
}

```

```

        envios_mail_sms.addElement((new elementoVector(
            ((elementoVector)envios.elementAt(i)).elemento1,
            ((elementoVector)envios.elementAt(i)).elemento6,
            resultadoSMS,
            resultadoMAIL)));
    }
    if (tipo.equals("E")){
        p.resultEnvioEspecif(envios_mail_sms);
    }
    else {
        if (tipo.equals("M")){
            p.resultEnvioGeneral(envios_mail_sms);
        }
    }
}

%>
<HTML>
<HEAD>
<TITLE> Enviar Mensajes</TITLE>
</HEAD>
<BODY background=" ../imagenes/admin_r6_c1.jpg">
<form name="envios">

<table border="1" name="envios_sms" class="soft" cellspacing="0">
<%
    for (int i = 0; i < envios_mail_sms.size(); i++){
%>
<tr>
<td class="sup">
    <input type="text" name="id"
value="<%=((elementoVector)envios_mail_sms.elementAt(i)).elemento1%>" size="3" style="font-family:Verdana;
border:0px blue solid; background-color:#BAC7D0;" readonly>
</td>
<td class="sup">
    <input type="text" name="celular"
value="<%=((elementoVector)envios_mail_sms.elementAt(i)).elemento2%>" size="15" style="font-family:Verdana;
border:0px blue solid; background-color:#BAC7D0;" readonly>
</td>
<td class="sup">
    <input type="text" name="resultado_sms"
value="<%=((elementoVector)envios_mail_sms.elementAt(i)).elemento3%>" size="40" style="font-family:Verdana;
border:0px blue solid; background-color:#BAC7D0;" readonly>
</td>
<td class="sup">
    <input type="text" name="resultado_mail"
value="<%=((elementoVector)envios_mail_sms.elementAt(i)).elemento4%>" size="40" style="font-family:Verdana;
border:0px blue solid; background-color:#BAC7D0;" readonly>
</td>
</tr>
<%
    }
%>
</table>

```

```
</form>

<% }catch(Exception e) { %>

<BODY>

<div id="Layer4" style="position:absolute; left:100px; top:50px; width:300px; height:30px">
<font color="red" face="Arial" size="3">
<b>No se ha conectado correctamente. Acceso Denegado.</b>
<b><%=e%></b>
</font>
</div>

<% }%>

</BODY>
</HTML>
```

## **Procedimientos Almacenados de la Base de Datos.**

### **Procedimiento PR\_POLIZA**

Este procedimiento es el encargado de acceder a la base de datos de la compañía de seguros y realizar un barrido de las tablas involucradas con el vencimiento de las pólizas; hace una copia en la base del sistema de notificaciones, de los registros extraídos y a su vez se encarga de realizar el envío via sms y mail, por medio de un job programado que se dispara de acuerdo a los parámetros de la tabla mensaje.

```
create or replace procedure pr_poliza (pv_proceso varchar2, pv_band varchar2) is
```

```

cursor c_proceso is
select ID,
       GN_PROCESOID,
       INTERVALO_EJEC,
       DIAS_RESTANTES,
       CANT_ENVIOS,
       GN_TIPOCLIENTEID,
       MENSAJE,
       ENVIA_SMS,
       ENVIA_MAIL,
       FECHAINICIO,
       FECHAFIN,
       GN_ESTADOID
from GN_MENSAJE
where GN_PROCESOID = PV_PROCESO;

cursor c_clientes (DIAS_RESTANTES NUMBER,
                   TIPO_CLIENTE VARCHAR2,
                   CANT_ENVIOS NUMBER)is
SELECT ENT2.NOMBRECOMPLETO AS AGENTE,
       R.NOMBRE AS RAMO,
       P.ID AS POLIZOID,
       P.NUMERO AS POLIZA,
       P.ORDEN,
       ENT.NOMBRECOMPLETO AS CLIENTE,
       TE.NOMBRE AS TIPOENDOSO,
       E.NUMERO AS NUMENDOSO,
       E.VALORASEGURADO,
       E.VALORPRIMANETA,
       D.FECHA AS FECHAFACTURACION,
       D.FECHACONTABILIZACION,
       E.VIGENCIADESDE,
       E.VIGENCIAHASTA,
       D.NUMERO AS FACTURA,
       ES.NOMBRE AS ESTADO_DOC,
       TS.NOMBRE AS TIPOSEGURO,
       ES2.NOMBRE AS EST_ACTUAL_POLIZA,
       E.ID AS ENDOSOID,
       ENT.EMAILPRINCIPAL EMAIL,
       ENT.TELEFONOCELULAR1 CELULAR
FROM SA.POLIZA P,
     SA.ENDOSO E,
     SA.RAMO R,
     SA.TIPOENDOSO TE,
     SA.ENDOSOESTADO EE,
     SA.CLIENTE C,
     SA.ENTIDAD ENT,
     SA.DOCUMENTO D,
     SA.ENDOSOAGENTE EA,
     SA.AGENTE AG,
     SA.ENTIDAD ENT2,
     SA.ESTADO ES,
     SA.TIPOSEGURO TS,
     SA.ESTADO ES2

```

```

WHERE P.ID = E.POLIZAID
AND P.RAMOID = R.ID
AND E.TIPOENDOSOID = TE.ID
AND E.ID = EE.ENDOSOID
AND P.CLIENTEID = C.ID
AND C.ENTIDADID = ENT.ID
AND EE.ESACTUAL = 1
AND EE.ESTADOID IN ('15', '13', '77')
AND E.ID = EA.ENDOSOID
AND P.ESTADOID = '6'
AND EA.AGENTEID = AG.ID
AND AG.ENTIDADID = ENT2.ID
AND EE.FACTURAID = D.ID
AND P.TIPOACEPTADOID IS NULL
AND EE.ESTADOID = ES.ID
AND E.TIPOSEGUROID = TS.ID
AND EA.TIPOAGENTE = 'L'
AND P.ESTADOID = ES2.ID
AND EE.ESACTUAL = '1'
/*AND TRUNC(P.VIGENCIAHASTA) BETWEEN TRUNC(SYSDATE)
AND TRUNC(SYSDATE) + dias_restantes*/
AND (TRUNC(P.VIGENCIAHASTA) = TRUNC(SYSDATE) + DIAS_RESTANTES
OR TRUNC(P.VIGENCIAHASTA) = TRUNC(SYSDATE) + CANT_ENVIOS)

AND TE.NEMOTECNICO = 'POL'
AND ENT.TIPOENTIDADID = TIPO_CLIENTE
ORDER BY 1, 2, 3, 4, 5, 7;

```

```

mens      varchar2(5000);
mens2     varchar2(5000);
mens3     varchar2(5000);
mens4     varchar2(5000);
var       varchar2(10000);
posic     number(10) := 0;
posic2    number(10) := 0;
sql_str   varchar2(5000);
var2      varchar2(1000);
valcampo  varchar2(5000);
secuencia number;
confirm_sms varchar2(5000);

```

```

begin
for r_proceso in c_proceso loop
  for r_clientes in c_clientes(r_proceso.dias_restantes,
    r_proceso.gn_tipoclienteid,
    r_proceso.cant_envios) loop

    insert into gn_transaccionesenviosespecific (id,
      polizaid,
      endosoid,
      agente,
      ramo,
      numpoliza,
      orden,

```

```

        cliente,
        tipoendoso,
        numendoso,
        valorprimaneta,
        vigenciadesde,
        vigenciahasta,
        tiposeguro,
        gn_mensajeid,
        celular,
        mail,
        fechaenvio)
values (sec_enviosespecif.nextval,
       r_clientes.polizaaid,
       r_clientes.endosoid,
       r_clientes.agente,
       r_clientes.ramo,
       r_clientes.poliza,
       r_clientes.orden,
       r_clientes.cliente,
       r_clientes.tipoendoso,
       r_clientes.numendoso,
       r_clientes.valorprimaneta,
       r_clientes.vigenciadesde,
       r_clientes.vigenciahasta,
       r_clientes.tiposeguro,
       r_proceso.id,
       r_clientes.celular,
       r_clientes.email,
       sysdate);
commit;

--construye mensaje-----
mens4 := null;
mens3 := r_proceso.mensaje;

select sec_enviosespecif.currval
into secuencia
from dual;

loop
mens := substr(mens3,0,instr(mens3,'#') - 1);
exit when mens is null;
posic2 := instr(mens3,'#')+1;
mens2 := substr(mens3,posic2);
var := '#'||substr(mens2,1,instr(mens2,'#'));
mens3 := substr(mens2,instr(mens2,'#') + 1);

select referencia
into var2
from gn_variables
where nombre_variable = var;

sql_str := 'SELECT '||var2|| ' FROM gn_transaccionesenviosespecif
           where id = '||secuencia;

```



```

EXECUTE IMMEDIATE sql_str INTO valcampo;

mens4 := mens4 || '||mens||' ||valcampo;

end loop;
mens := substr(mens3,0);

mens4 := mens4 || '||mens';

update gn_transaccionesenviosespecific
  set mensaje = mens4
  where id = secuencia;

commit;

if pv_band = '1' then

  loop
    mens := substr(mens4,1,155);
    mens4 := substr(mens4,156);
    exit when mens4 is null;
    envio_sms(r_clientes.celular,
              mens4,
              confirm_sms);
  end loop;

  update gn_transaccionesenviosespecific
    set descripcionresultsms = confirm_sms
    where id = secuencia;

  commit;
end if;
-----
end loop;
end loop;

end;

```

## **Procedimiento PR\_CARTERA**

Este procedimiento es el encargado de acceder a la base de datos de la compañía de seguros y realizar un barrido de las tablas involucradas con el vencimiento de las cuotas y letras, verificando los días de vencimiento; hace una copia en la base del sistema de notificaciones, de los registros extraídos y a su vez se encarga de realizar el envío via

sms y mail, por medio de un job programado que se dispara de acuerdo a los parámetros de la tabla mensaje

```
create or replace procedure pr_cartera (pv_proceso varchar2, pv_band varchar2) is
cursor c_proceso is
select ID,
       GN_PROCESOID,
       INTERVALO_EJEC,
       DIAS_RESTANTES,
       CANT_ENVIOS,
       GN_TIPOCLIENTEID,
       MENSAJE,
       ENVIA_SMS,
       ENVIA_MAIL,
       FECHAINICIO,
       FECHAFIN,
       GN_ESTADOID
from GN_MENSAJE
where GN_PROCESOID = PV_PROCESO;

cursor c_clientes (DIAS_RESTANTES NUMBER,
                  TIPO_CLIENTE VARCHAR2,
                  CANT_ENVIOS NUMBER)is
select to_number(DOC.NUMERO) as numero_fac, DCUO.NUMERO, DCUO.DESCRIPCION,
       DCUO.FECHA, DCUO.VALORLOCAL, DCUO.SALDOLOCAL, DCUO.ESTADOID,
DCUO.FECHAVENCIMIENTO,
ent.NOMBRECOMPLETO as nombre_cliente,
(select ra.NOMBRE
 from sa.endosoestado ee, sa.endoso en, sa.poliza pol, sa.ramo ra
 where ee.endosoid=en.id
       and en.polizaid= pol.ID
       and pol.RAMOID=ra.id
       and ee.ESACTUAL='1'
       and ee.facturaid=doc.id) as ramo,
(select pol.numero
 from sa.endosoestado ee, sa.endoso en, sa.poliza pol
 where ee.endosoid=en.id
       and en.polizaid= pol.ID
       and ee.ESACTUAL='1'
       and ee.facturaid=doc.id) as num_poliza,
(select pol.orden
 from sa.endosoestado ee, sa.endoso en, sa.poliza pol
 where ee.endosoid=en.id
       and en.polizaid= pol.ID
       and ee.ESACTUAL='1'
       and ee.facturaid=doc.id) as ORDEN,
(select en.NUMERO
 from sa.endosoestado ee, sa.endoso en, sa.poliza pol
 where ee.endosoid=en.id
       and en.polizaid= pol.ID
       and ee.ESACTUAL='1'
       and ee.facturaid=doc.id) as num_endoso,
```

```

ent.emailprincipal email,
ent.telefonocelular1 celular, c.orden as numcuota letra
from sa.documento doc, sa.cuota c, sa.documento dcuo, sa.entidad ent
where DOC.ID=C.FACTURACLIENTEID
and C.ID=DCUO.ID
and doc.ENTIDADID=ent.id
and doc.tipodocumentoid='1'
and doc.estadoid<>'161'
and doc.saldolocal>0
and DCUO.SALDOLOCAL>0
AND ENT.TIPOENTIDADID = TIPO_CLIENTE
AND (TRUNC(DCUO.FECHAVENCIMIENTO) = TRUNC(SYSDATE) + DIAS_RESTANTES
OR TRUNC(DCUO.FECHAVENCIMIENTO) = TRUNC(SYSDATE) + CANT_ENVIOS)
--and DCUO.FECHAVENCIMIENTO <= to_DATE ('11/08/2010','DD/MM/YYYY')
--order by to_number(doc.numero)-, dcuo.numero
union
select to_number(DOC.NUMERO) as numero_fac, dlet.NUMERO, dlet.DESCRIPCION,
dlet.FECHA, dlet.VALORLOCAL, dlet.SALDOLOCAL, dlet.ESTADOID, DLET.FECHAVENCIMIENTO,
ent.NOMBRECOMPLETO as nombre_cliente,
(select ra.NOMBRE
from sa.endosoestado ee, sa.endoso en, sa.poliza pol, sa.ramo ra
where ee.endosoid=en.id
and en.polizaid= pol.ID
and pol.RAMOID=ra.id
and ee.ESACTUAL='1'
and ee.facturaid=doc.id) as ramo,
(select pol.numero
from sa.endosoestado ee, sa.endoso en, sa.poliza pol
where ee.endosoid=en.id
and en.polizaid= pol.ID
and ee.ESACTUAL='1'
and ee.facturaid=doc.id) as num_poliza,
(select pol.orden
from sa.endosoestado ee, sa.endoso en, sa.poliza pol
where ee.endosoid=en.id
and en.polizaid= pol.ID
and ee.ESACTUAL='1'
and ee.facturaid=doc.id) as ORDEN,
(select en.NUMERO
from sa.endosoestado ee, sa.endoso en, sa.poliza pol
where ee.endosoid=en.id
and en.polizaid= pol.ID
and ee.ESACTUAL='1'
and ee.facturaid=doc.id) as num_endoso,
ent.emailprincipal email,
ent.telefonocelular1 celular, l.orden as numcuota letra
from sa.documento doc, sa.letra l, sa.documento dlet, sa.entidad ent
where DOC.ID=l.FACTURACLIENTEID
and l.ID=DLET.ID
and doc.ENTIDADID=ent.id
and doc.tipodocumentoid='1'
and doc.estadoid<>'161'
and doc.saldolocal>0
AND (TRUNC(DLET.FECHAVENCIMIENTO) = TRUNC(SYSDATE) + DIAS_RESTANTES

```

```

OR TRUNC(DLET.FECHAVENCIMIENTO) = TRUNC(SYSDATE) + CANT_ENVIOS)
AND ENT.TIPOENTIDADID = TIPO_CLIENTE
--and DLET.FECHAVENCIMIENTO <= to_DATE ('11/08/2010','DD/MM/YYYY')
and Dlet.SALDOLOCAL>0
order by nombre_cliente;

mens      varchar2(5000);
mens2     varchar2(5000);
mens3     varchar2(5000);
mens4     varchar2(5000);
var       varchar2(10000);
posic     number(10) := 0;
posic2    number(10) := 0;
sql_str   varchar2(5000);
var2      varchar2(1000);
valcampo  varchar2(5000);
secuencia number;
confirm_sms varchar2(5000);

begin
for r_proceso in c_proceso loop
for r_clientes in c_clientes(r_proceso.dias_restantes,
r_proceso.gn_tipoclienteid,
r_proceso.cant_envios) loop

insert into gn_transaccionesenviosespecific (id,
FACTURANUM,
NUMCUOTALETRA,
DESCRIPCIONCUOTALETRA,
FECHAFACTURACION,
VALORPRIMANETA,
SALDOCUOTALETRA,
ESTADOCUOTALETRA,
FECHAVENCIMIENODOC,
VIGENCIAHASTA,
CLIENTE,
RAMO,
NUMPOLIZA,
ORDEN,
NUMENDOSO,
gn_mensajeid,
celular,
mail,
fechaenvio) ----agregado 22-sept-2010
values (sec_enviosespecific.nextval,
r_clientes.numero_fac,
r_clientes.numcuotaetra,
r_clientes.descripcion,
r_clientes.fecha,
r_clientes.valorlocal,
r_clientes.saldolocal,
r_clientes.estadoid,
r_clientes.fechavencimiento,

```

```

        r_clientes.fechavencimiento,
        r_clientes.nombre_cliente,
        r_clientes.ramo,
        r_clientes.num_poliza,
        r_clientes.orden,
        r_clientes.num_endoso,
        r_proceso.id,
        --r_clientes.celular,
        --r_clientes.email
        '092641582',
        'luis.riofrio@hotmail.com',
        sysdate); ----agregado 22-sept-2010
commit;

--construye mensaje-----
mens4 := null;
mens3 := r_proceso.mensaje;

select sec_enviosespecific.currval
       into secuencia
       from dual;

loop
  mens := substr(mens3,0,instr(mens3,'#') - 1);
  exit when mens is null;
  posic2 := instr(mens3,'#')+1;
  mens2 := substr(mens3,posic2);
  var := '#'||substr(mens2,1,instr(mens2,'#'));
  mens3 := substr(mens2,instr(mens2,'#') + 1);

  select referencia
         into var2
         from gn_variables
         where nombre_variable = var;

  sql_str := 'SELECT '||var2|| ' FROM gn_transaccionesenviosespecific where id = '||secuencia;
  EXECUTE IMMEDIATE sql_str INTO valcampo;

  mens4 := mens4 ||' '||mens||' '||valcampo;

end loop;
mens := substr(mens3,0);

mens4 := mens4 ||' '||mens;

update gn_transaccionesenviosespecific
       set mensaje = mens4
       where id = secuencia;

commit;

if pv_band = '1' then

  loop

```

```

mens := substr(mens4,1,155);
mens4 := substr(mens4,156);
exit when mens4 is null;
envio_sms(r_clientes.celular,
mens4,
confirm_sms);
end loop;

update gn_transaccionesenviosespecific
set descripcionresultsms = confirm_sms
where id = secuencia;

commit;
end if;
-----
end loop;
end loop;

end;
```

## **Procedimiento PR\_PROMOCIONES**

Este procedimiento es el encargado de acceder a la base de datos de la compañía de seguros y realizar un barrido de las tablas de clientes y enviar cualquier tipo de mensaje ya sea de agradecimiento, navideños, promociones, etc; hace una copia en la base del sistema de notificaciones, de los registros extraídos y a su vez se encarga de realizar el envío via sms y mail, por medio de un job programado que se dispara de acuerdo a los parámetros de la tabla mensaje.

```

create or replace procedure pr_promociones (pv_proceso varchar2, pv_band varchar2) is
cursor c_proceso is
select ID,
GN_PROCESOID,
INTERVALO_EJEC,
DIAS_RESTANTES,
CANT_ENVIOS,
GN_TIPOCLIENTEID,
MENSAJE,
ENVIA_SMS,
```

```

        ENVIA_MAIL,
        FECHAINICIO,
        FECHAFIN,
        GN_ESTADOID
    from GN_MENSAJE
    where GN_PROCESOID = PV_PROCESO;

cursor c_clientes (TIPO_CLIENTE VARCHAR2)is
select ENT.ID,
       ENT.NOMBRECOMPLETO cliente,
       ENT.EMAILPRINCIPAL email,
       ENT.TELEFONOCELULAR1 celular
    from SA.ENTIDAD ENT
   where ENT.TIPOENTIDADID = TIPO_CLIENTE
        and ENT.id<>'1'
        and (ENT.EMAILPRINCIPAL <> 'NULL'
             OR ENT.TELEFONOCELULAR1 <> 'NULL')
        and rownum < 10
   order by ENT.NOMBRECOMPLETO;

mens      varchar2(5000);
mens2     varchar2(5000);
mens3     varchar2(5000);
mens4     varchar2(5000);
var       varchar2(10000);
posic     number(10) := 0;
posic2    number(10):= 0;
sql_str   varchar2(5000);
var2      varchar2(1000);
valcampo  varchar2(5000);
secuencia number;
confirm_sms varchar2(5000);

begin
for r_proceso in c_proceso loop
    for r_clientes in c_clientes(r_proceso.gn_tipoclienteid) loop

        insert into gn_transaccionesenviosgeneral (id,
                                                    gn_mensajeid,
                                                    CLIENTE,
                                                    celular,
                                                    mail,
                                                    fechaenvio)
        values (sec_enviosespecific.nextval,
                r_proceso.id,
                r_clientes.cliente,
                r_clientes.celular,
                r_clientes.email,
                sysdate);
        commit;

        --construye mensaje-----
        mens4 := null;
        mens3 := r_proceso.mensaje;

```

```

select sec_enviosespecific.currval
into secuencia
from dual;

loop
mens := substr(mens3,0,instr(mens3,'#') - 1);
exit when mens is null;
posic2 := instr(mens3,'#')+1;
mens2 := substr(mens3,posic2);
var := '#'||substr(mens2,1,instr(mens2,'#'));
mens3 := substr(mens2,instr(mens2,'#') + 1);

select referencia
into var2
from gn_variables
where nombre_variable = var;

sql_str := 'SELECT '||var2|| ' FROM gn_transaccionesenviosgeneral where id = '||secuencia;
EXECUTE IMMEDIATE sql_str INTO valcampo;

mens4 := mens4 || '||mens||' '||valcampo;

end loop;
mens := substr(mens3,0);

mens4 := mens4 || '||mens;

update gn_transaccionesenviosgeneral
set mensaje = mens4
where id = secuencia;

commit;

if pv_band = '1' then

loop
mens := substr(mens4,1,155);
mens4 := substr(mens4,156);
exit when mens4 is null;
envio_sms(r_clientes.celular,
mens4,
confirm_sms);
end loop;

update GN_TRANSACCIONESENVIOSGENERAL
set descripcionresultsms = confirm_sms
where id = secuencia;

commit;
end if;
-----
end loop;
end loop;

```



end;

## **Procedimiento ENVIO\_SMS**

Este procedimiento es el encargado de realizar el envío programado de sms, desde la base de datos y es llamado desde los procedimientos PR\_CARTERA, PR\_POLIZA, PR\_PROMOCIONES.

```
create or replace procedure envio_sms (celular varchar2, mensaje varchar2, confirmacion in out varchar2) is
  req utl_http.req;
  resp utl_http.resp;
  value VARCHAR2(1024);
BEGIN
  req := utl_http.begin_request('http://extern.nrs-
group.com/send_multisms.phpuser=042687185=luisao=||mensaje||'=||celular||'=59395681326=ec');
  utl_http.set_header(req, 'User-Agent', 'Mozilla/4.0');
  resp := utl_http.get_response(req);
  LOOP
    utl_http.read_line(resp, value, TRUE);
    dbms_output.put_line(value);
    confirmacion := value;
  END LOOP;
  utl_http.end_response(resp);
EXCEPTION
  WHEN utl_http.end_of_body THEN
    utl_http.end_response(resp);
END;
```

## **Procedimiento CREA\_JOB**

Este procedimiento se encarga de crear el job cuando se realiza el envío desde la aplicación, es decir cuando se fuerza el envío; ya que cuando este es programado se crea al job al crear un registro en la tabla gn\_mensaje.

```
create or replace procedure crea_job(proc varchar2, id varchar2) is
begin
  dbms_job.isubmit(90,proc||'('||id||','||'0'||')';sysdate);
  commit;
end;
```

、

## TRIGGER GN\_MENSAJE\_INSERT

Este trigger se dispara al momento de ingresar un mensaje (insert en la tabla GN\_MENSAJE ), y su función es crear un job programado en la base de datos, para que se ejecute los procesos de acuerdo a los datos ingresados.

```
create or replace trigger gn_mensaje_insert
before insert or update of gn_procesoid or delete on gn_mensaje
for each row
declare
    proceso varchar2(50);
begin

    if inserting then

        select nombre_proceso
        into proceso
        from gn_proceso
        where id = :new.gn_procesoid;

        begin
            dbms_job.isubmit(:new.id,
                proceso||(''||:new.gn_procesoid||','"1");',
                :new.fechainicio,
                'sysdate+1');
        end;
    elsif updating then

        select nombre_proceso
        into proceso
        from gn_proceso
        where id = :new.gn_procesoid;

        begin
            dbms_job.what(:new.id,
                proceso||(''||:new.gn_procesoid||','"1");');
        end;
    elsif deleting then

        select nombre_proceso
        into proceso
        from gn_proceso
        where id = :old.gn_procesoid;

        begin
            dbms_job.remove(:old.id);
        end;
    end if;

end;
```

# MANUAL DE USUARIO

## **Descripción**

El siguiente manual está orientado para el fácil y correcto manejo de la aplicación Gestor de Notificaciones, con la finalidad de que cualquier usuario final esté en la capacidad de poder utilizar el software implementado.

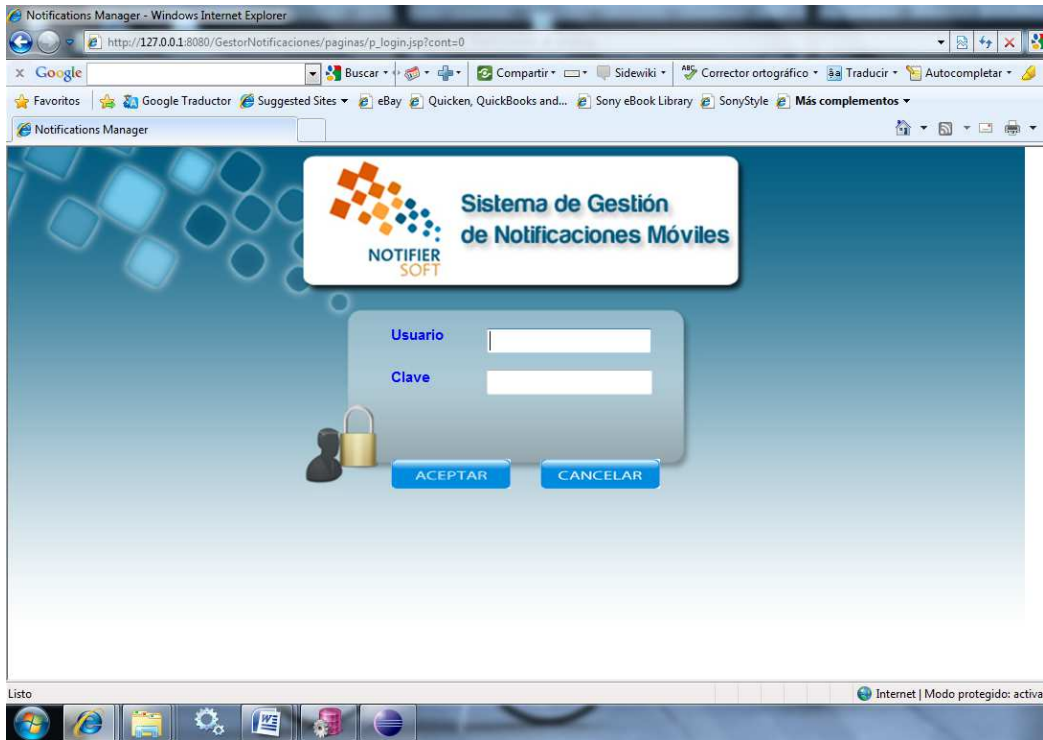
## **Objetivo**

El objetivo de éste manual es que mediante éste podemos especificar de forma detallada el funcionamiento del sistema, como parametrizar variables, mensajes, procesos; realizar envíos masivos vía mail y sms de acuerdo a procesos seleccionados.

## **Detalle de Acciones a realizar y Respuestas del Sistema**

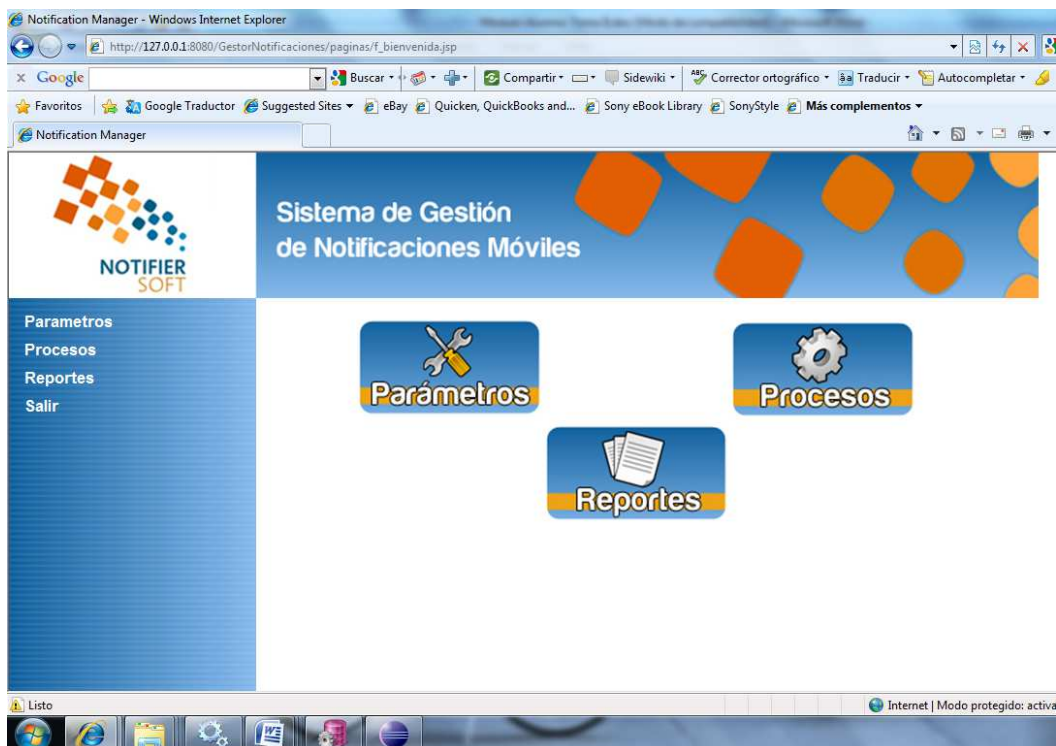
A continuación se explicará el uso correcto del sistema Notifier Soft:

**Pantalla de Ingreso al Sistema.-** Para acceder al Sistema de Notificaciones, la persona deberá ingresar con un usuario y una clave, tal como se muestra en la figura 1.



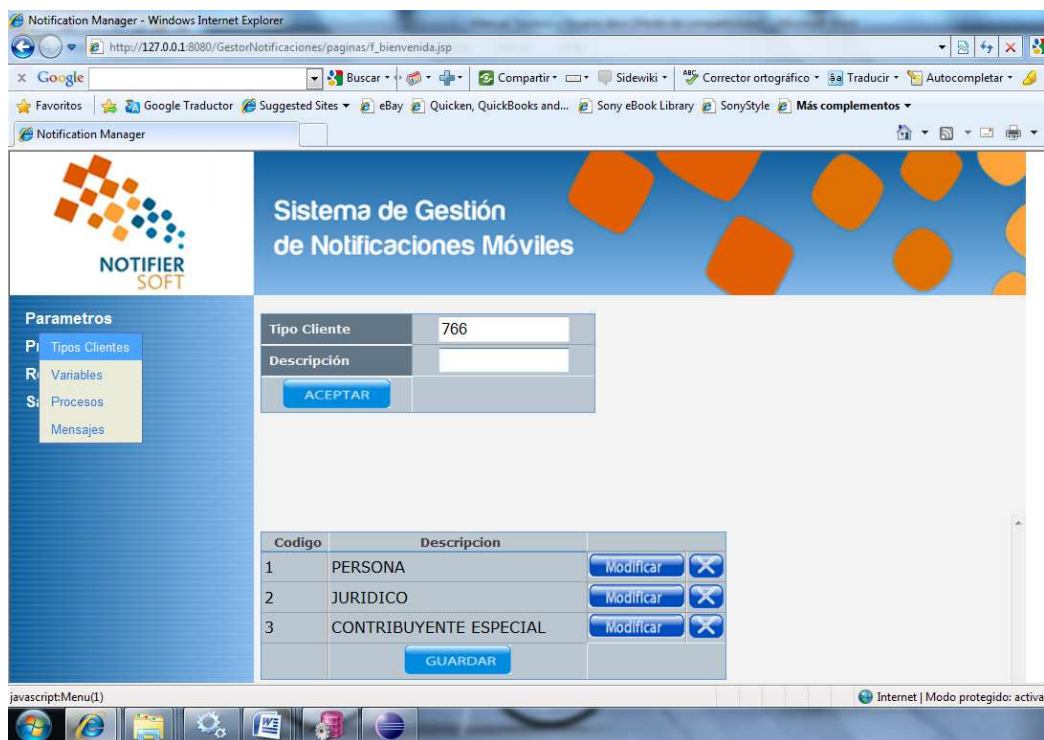
**Fig. 1. Pantalla de Ingreso al Sistema de Notificaciones – Notifier Soft.**

Posterior a esto, aparecerá una pantalla de inicio con las diferentes opciones de menú del sistema, como vemos en la figura 2



**Fig. 2. Pantalla de Inicio del Sistema**

**Pantalla de Parametrización de Tipos de Clientes.-** Para acceder a la opción de parametrización de Tipos Clientes se ingresa al menú Parámetros opción Tipo Clientes, tal como se muestra en la figura 3.



**Fig. 3. Pantalla de Parametrizacion de tipos de clientes**

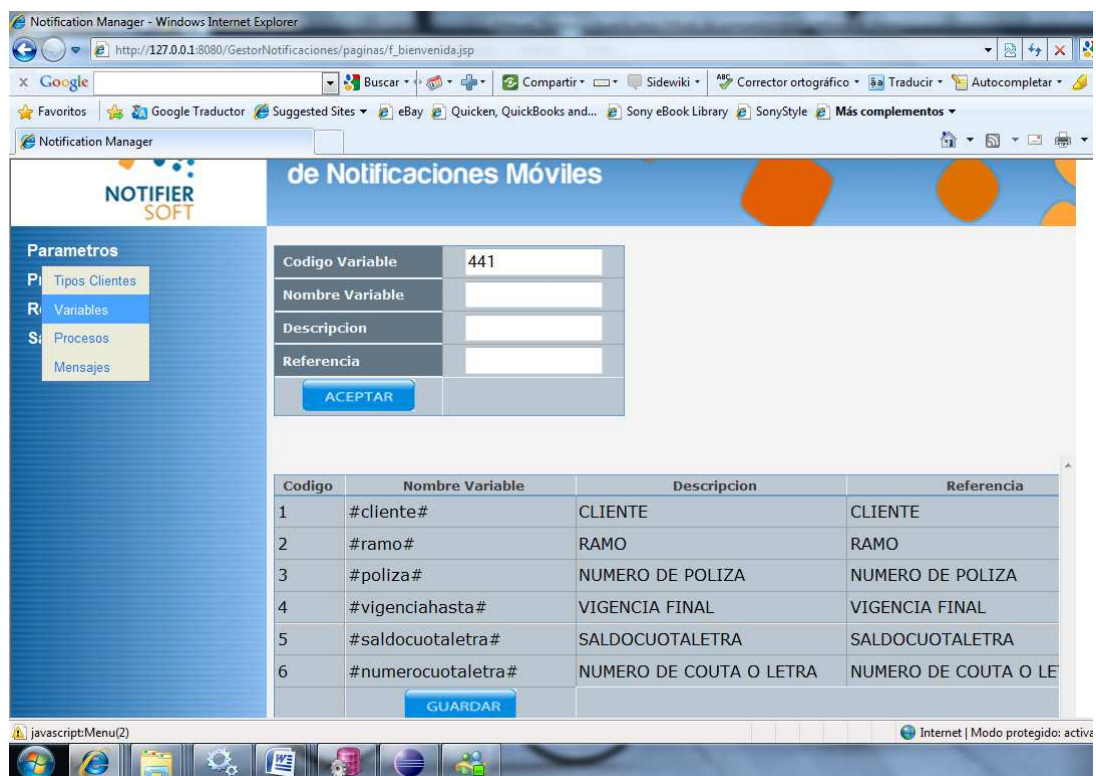
En la parte superior se tiene el campo Tipo Cliente, que es el id y se genera automáticamente, luego tenemos el campo de la descripción en el que se debe ingresar un texto obligatoriamente antes de dar click en el botón aceptar de lo contrario se dispara un mensaje de advertencia que indica que se debe ingresar algún dato. Al dar click en el

campo aceptar automáticamente se agrega un registro en la tabla de la parte inferior, al presionar el botón guardar se almacena el nuevo registro en la base de datos.

En la tabla que se muestra en la figura 2.3, se cuenta con dos botones, modificar y eliminar.

Los tipos de clientes se los parametriza debido a que los mensajes que se envían, se dirigen a distinto tipo de clientes, es decir podrían ser personas o empresas; esta opción también podría ser utilizada en otros tipos de empresas que se requiera parametrizar entidades como clientes, proveedores, empleados, etc; y enviarles mensajes via sms y mail.

**Pantalla de Parametrización de Variables.-** Para acceder a la opción de parametrización de variables, se ingresa al menú Parámetros opción Variables tal como se muestra en la figura 4.

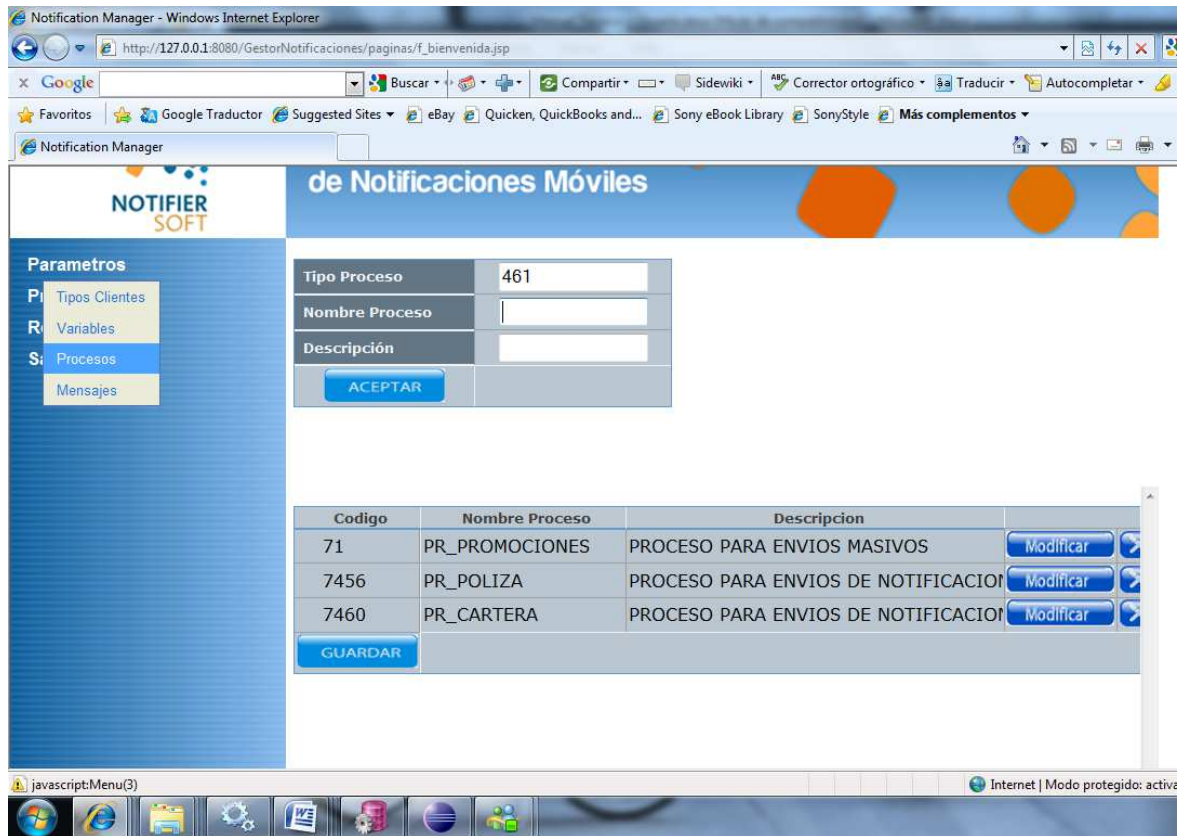


**Fig. 4. Pantalla de Parametrizacion de Variables**

La forma de ingreso, modificación y eliminación de las variables, se la realiza de la misma forma como los tipos de clientes.

La finalidad de parametrizar variables es poder construir mensajes no solo con textos fijos, sino con datos que servirán para personalizar el mensaje de acuerdo a las necesidades de la empresa.

**Pantalla de Parametrización de Procesos.-** Para acceder a la opción de parametrización de procesos, se ingresa al menú Parámetros opción Procesos tal como se muestra en la figura 5.



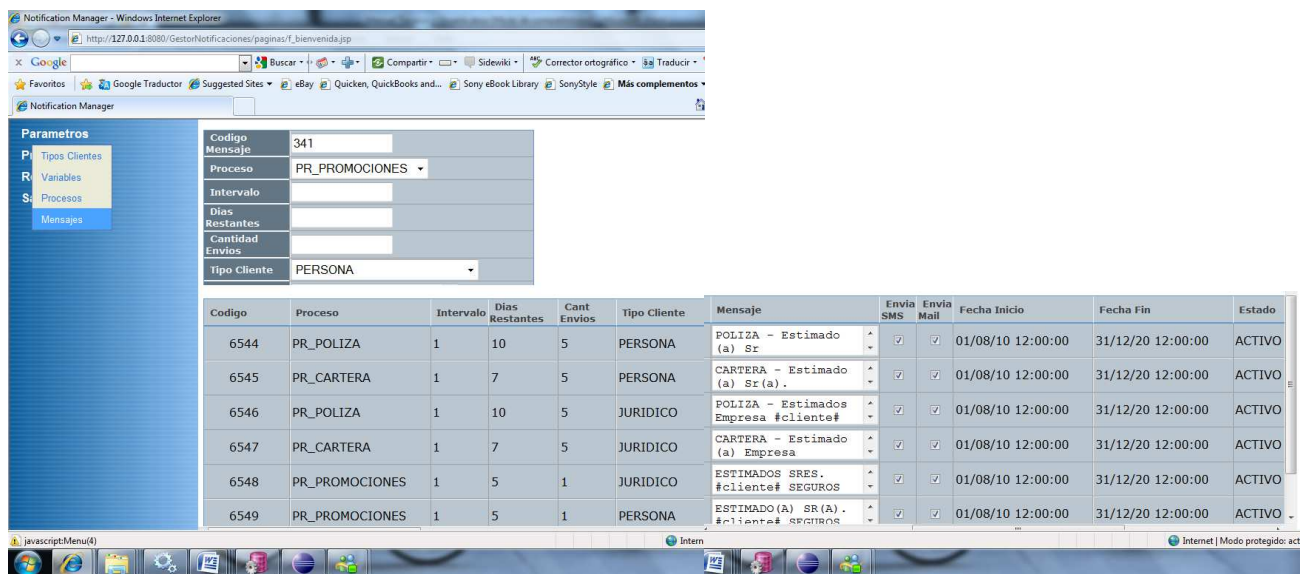
**Fig. 5. Pantalla de Parametrizacion de Procesos**

La forma de ingreso, modificación y eliminación de los procesos, se la realiza de la misma forma como los tipos de clientes.

Los procesos están ligados directamente al envío de notificaciones de vencimientos de pólizas y vencimientos de cartera, pero adicionalmente a la creación de nuevos procesos para ser utilizados en el envío de nuevas promociones.



**Pantalla de Parametrización de Mensajes.-** Para acceder a la opción de parametrización de mensajes, se ingresa al menú Parámetros opción Mensajes tal como se muestra en la figura 6.



**Fig. 6. Pantalla de Parametrización de Mensajes**

La forma de ingreso, modificación y eliminación de los mensajes, se la realiza de la misma forma como los tipos de clientes.

Esta es una de las pantallas más importantes del sistema, debido que aquí se realiza el ingreso de toda la información referente al mensaje en sí, y a la programación de su envío. Aquí se escoge el proceso que va a utilizar el mensaje, Intervalos de envíos,

cantidad de envíos, tipo cliente, se construye el mensajes con las variables ingresadas anteriormente (se escoge las que se necesite), si el envío es a sms, si el envío es a mail, fecha de inicio de envío, fecha final, estado del mensaje.

**Pantalla de Procesos – Envío Avisos Específicos .-** Para acceder a la opción de envíos de mensajes, se ingresa al menú Procesos opción Envío Avisos Específicos tal como se muestra en la figura 7.



**Fig. 7. Pantalla de Envío de procesos masivos**

Esta opción es utilizada para realizar una demostración de envíos de sms y mail, forzando a la aplicación a que realice el envío sin necesidad de esperar a que se dispare el proceso programado.

Primero se escoge el proceso entre los dos que son el objetivo de este proyecto (PROCESO VENCIMIENTO DE POLIZAS, Y PROCESO VENCIMIENTO DE CARTERA) de un combo box, y se da click en el botón aceptar; luego de esto se reflejara en la parte inferior de la pantalla un listado con los registros que cumplen con la condición del proceso seleccionado como podemos observar en la figura 8

Notification Manager - Windows Internet Explorer  
http://192.168.1.3:8080/GestorNotificaciones/paginas/f\_bienvenida.jsp

Archivo Edición Ver Favoritos Herramientas Ayuda

Windows Live Sitios sugeridos MSN Latinoamérica MSN Video Hotmail gratuito Galería de Web Slice Noticias en MSN

Notification Manager

**NOTIFIER SOFT**

**Sistema de Gestión de Notificaciones Móviles**

**ENVIOS DE AVISOS - PROCESOS ESPECIFICOS**

Proceso: PROCESO PARA ENVIOS DE NOTIFICACIONES DE VENCIMIENTOS DE POLIZAS

ACEPTAR

606	LOPEZ GARCIA MARTIN ERNESTO	1001189	TRANSPORTE	20/10/2010
607	SOCIEDAD ECUATORIANA DE ALIM	1002553	GARANTIAS ADUANERAS	20/10/2010
608	ELECTROSPEED S.A.	1001186	TRANSPORTE	15/10/2010
609	LOPEZ GARCIA MARTIN ERNESTO	1001189	TRANSPORTE	20/10/2010
610	SOCIEDAD ECUATORIANA DE ALIM	1002553	GARANTIAS ADUANERAS	20/10/2010
611	ELECTROSPEED S.A.	1001186	TRANSPORTE	15/10/2010

ENVIAR

Listo

Inicio 3 Explora... Diagrama d... 2 Microso... 3 Interne... Java EE - G... Vinculos

Fig. 8. Listado de registros generados por el proceso

Luego de esto se procede a dar click en el botón enviar, y automáticamente se realiza el envío a mail y celular y se muestra otra pantalla con el resumen de envío como se muestra en la figura 9.

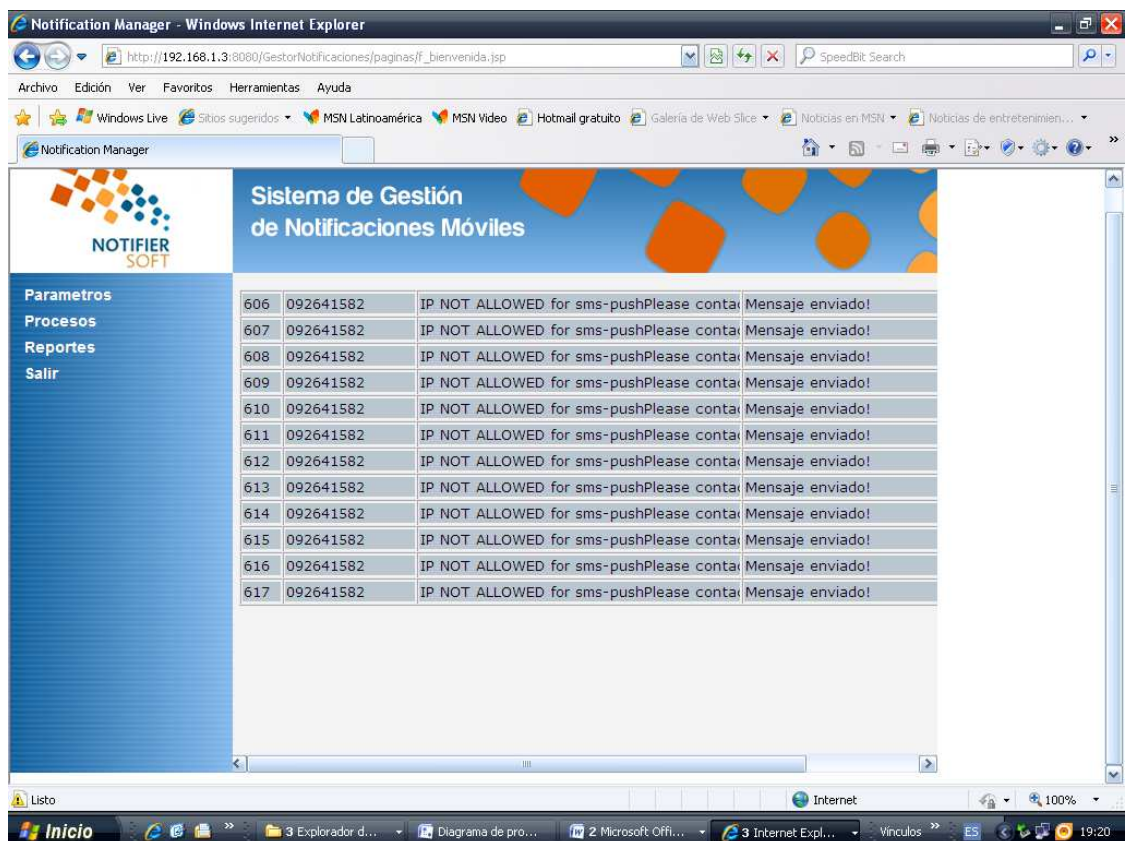
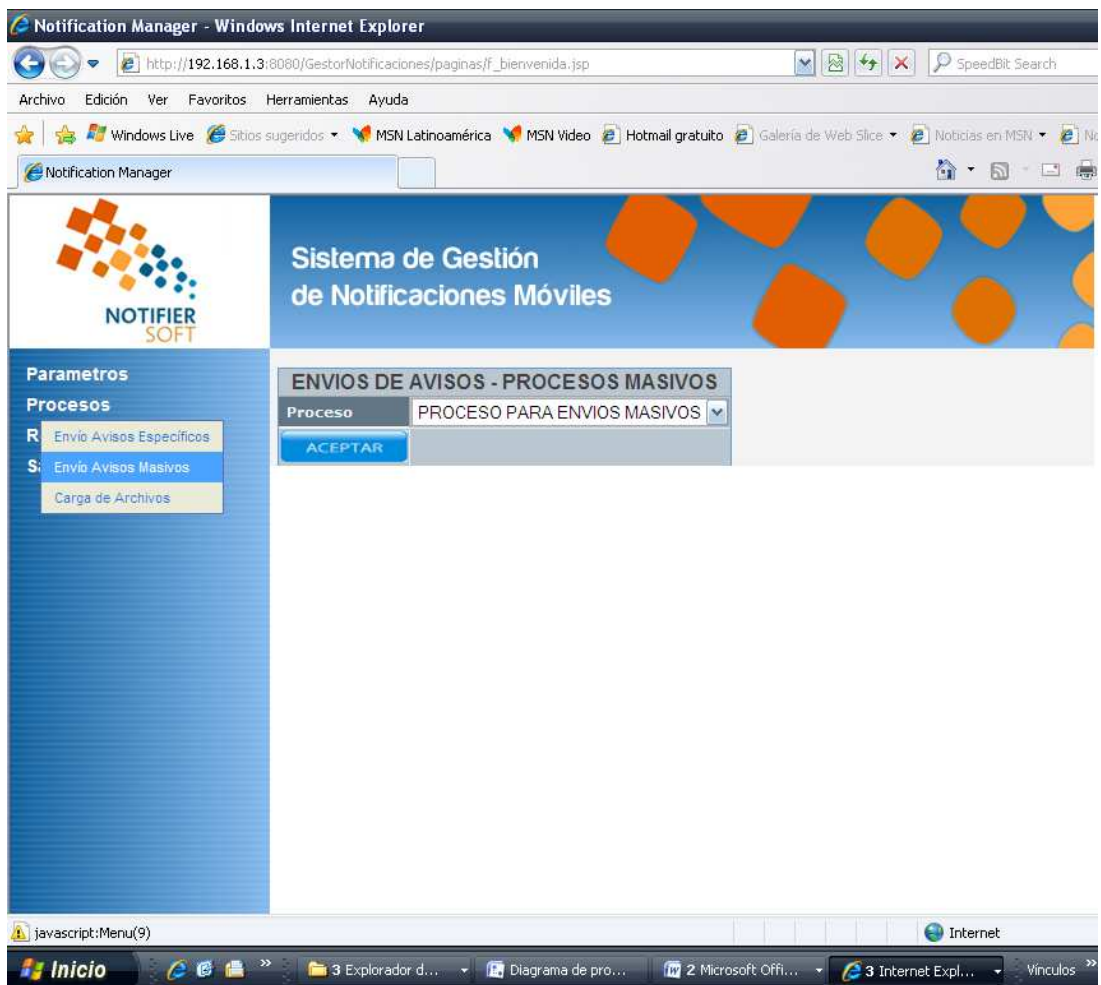


Fig. 9. Resultado presentado posterior al envío

**Pantalla de Procesos – Envío Avisos Masivos.-** La opción Parámetros → Envíos Avisos Masivos, tiene una función similar al Envío de Avisos Específicos, como se muestra en la figura 10.



**Fig. 10. Pantalla de Envío de procesos masivos generales**