

9. PLIEGO DE CONDICIONES

9.1. Clase Conexion.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Conexion.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/
package ardb;

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

/**
 * Clase utilizada por el Servidor de Registro de Acceso para establecer y cerrar
 * la conexión con la base de datos "Registro de Acceso" ubicada en PostgreSQL
 * Database Server 8.0.
 * Para que exista comunicación se hacen necesarias las extensiones JDBC.
 */

public class Conexion {

    /**
     * Establece la conexión con la base de datos "Registro de Acceso".
     * Cargamos en primer lugar el driver JDBC, que nos permite comunicarnos con
     * el PostgreSQL Database Server 8.0 e intentamos realizar la conexión con la
     * base de datos "Registro de Acceso".
     * @return Retorna un objeto Connection con la conexión establecida entre el
     * servidor y la base de datos.
     * @throws Exception al cargar el driver JDBC o al realizar la
     * conexión con la base de datos "Registro de Acceso".
     */
    public static Connection EstablecerConexion()
    throws Exception
    {
        // Database URL de la base de datos "Registro de Acceso".
        String url = "jdbc:postgresql://localhost:5432/Registro de Acceso";
        // Nombre del usuario propietario de la base de datos "Registro de Acceso".
        String usuario = "postgres";
        // Contraseña del usuario propietario de la base de datos "Registro de Acceso".*/
        String password = "sevilla";
        // Obtiene la conexión con la base de datos.
        Connection conexion = null;

        // Intento de cargar el driver para conectar con la base de datos diseñada con
        PostgreSQL.
        // Intento de establecer la conexión con la base de datos "Registro de Acceso".
        try { Class.forName("org.postgresql.Driver");
            conexion = DriverManager.getConnection(url, usuario, password);
        }

        catch (ClassNotFoundException Ex)
        { System.out.println("Conexion.java: Error al cargar el driver");
          System.out.println(Ex.getMessage());
        }
        catch (SQLException sqlEx)
        { System.out.println("Conexion.java: Se ha producido un error al establecer la
          conexion con:" + url );
          System.out.println(sqlEx.getMessage());
        }

        System.out.println("Conexion.java: Conectando con la base de datos 'Registro de
        Accesos'.");

        return conexion;
    }
}
```

```
}

/**
 * Cierra la conexión con la base de datos referenciada en el argumento
 * objeto tipo Connection.
 * @param conexion : Connection con la conexión que quiere cerrarse.
 * @throws java.sql.SQLException al cerrar la conexión con la base de datos
 */
public static void CerrarConexion(Connection conexion)
throws SQLException
{
    //Intento de cerrar la conexión con la base de datos.
    try { conexion.close(); }
    catch (SQLException sqlEx)
    { System.out.println("
      + "Conexion.java: Se ha producido un error al cerrar "
      + "la conexión con la base de datos."+ sqlEx.getMessage());
    }
}

}
```

9.2. Clase RegistroDeAcceso.java.-

```

/**
*****
                Proyecto final de carrera.
        Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: RegistroDeAcceso.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ardb;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.ResultSet;
import java.sql.Statement;

/**
 * Crea, comprueba, actualiza y consulta la base de datos "Registro de Acceso"
 * ubicada en PostgreSQL Database Server 8.0
 * Las tablas y campos de la base de datos "Registro de Acceso" han sido diseñadas
 * con el fin de proporcionar una solución práctica a las exigencias, en cuanto
 * a materia de seguridad, que deben reunir los ficheros automatizados que
 * contengan datos de carácter personal relativos a la salud.
 * Motivado por el Real Decreto 994/1999, la Ley 41/2002, y la Ley Orgánica 15/1999.
 */
public class RegistroDeAcceso {

    /** Objeto de tipo Statement necesario para ejecutar las sentencias SQL con las
     * que nos comunicamos con la base de datos. */
    private static Statement st;
    /** Objeto de tipo Connection necesario para obtener la conexión con la base de
    datos.*/
    private static Connection conexion;
    /** String con la sentencia SQL a ejecutar */
    private static String sql;
    /** Objeto de tipo ResultSet con los resultados devueltos por la base de datos al
    consultarla. */
    private static ResultSet rs;

    /**
     * Diseña la base de datos "Registro de Acceso".
     * Una vez creada en PostgreSQL Database Server 8.0 un objeto base de datos con
     * el nombre "Registro de Acceso" se procederá a la ejecución de este método para
     * concretar el diseño de las tablas que formaran la solución práctica a las
     * exigencias en materia de seguridad aparecidas en la legislación vigente.
     * CrearTablasBaseDeDatos() establece la conexión con la base de datos, crea las
     * tablas, indica los index y señala los comentarios de cada columna actualizando
     * la base de datos con el metodo ActualizarBaseDeDatos().
     * @throws java.lang.Exception al crear las tablas en la base de datos.
     */
    public static void CrearTablasBaseDeDatos ()
    throws Exception
    {
        // Referencia a un objeto de la interface java.sql.Connection
        conexion = null;

        try {
            conexion = Conexion.EstablecerConexion();

            // Creamos las tablas y definimos los campos.
            sql = ""
            + "CREATE TABLE COMUNICACION (Codigo_Comunicacion char(5) PRIMARY KEY, "
            + "Descripcion varchar(200) NOT NULL) WITHOUT OIDS;"

            + "CREATE TABLE FORMA_DE_ENVIO (Codigo_Envio char(5) PRIMARY KEY, Descripcion
            varchar(200) NOT NULL)"
            + "WITHOUT OIDS;"

```

```
+ "CREATE TABLE INCIDENCIAS (Codigo_Incidencia char(5) PRIMARY KEY, Descripcion
varchar(200) NOT NULL, Efectos varchar(200) NOT NULL)"
+ "WITHOUT OIDS;"

+ "CREATE TABLE MOTIVO_DE_DENEGACION (Codigo_Denegacion char(5) PRIMARY KEY,
Descripcion varchar(200) NOT NULL)"
+ "WITHOUT OIDS;"

+ "CREATE TABLE TIPO_DE_SOPORTE (Codigo_Soporte char(5) PRIMARY KEY, Descripcion
varchar(200) NOT NULL)"
+ "WITHOUT OIDS;"

+ "CREATE TABLE SITUACION_ACCESO (Codigo_Acceso char(5) PRIMARY KEY, "
+ "Descripcion varchar(200) NOT NULL) WITHOUT OIDS;"

+ "CREATE TABLE CREDENCIALES (Codigo_Credenciales char(6) PRIMARY KEY, "
+ "Descripcion varchar(200) NOT NULL, Recursos varchar(2000) NOT NULL) WITHOUT
OIDS;"

+ "CREATE TABLE RECURSOS (Recurso decimal PRIMARY KEY, "
+ "Descripcion varchar(200) NOT NULL) WITHOUT OIDS;"

+ "CREATE TABLE REGISTRO_DE_INCIDENCIAS (Numero_Incidencia SERIAL PRIMARY KEY,
Codigo_Incidencia char(5) REFERENCES INCIDENCIAS (Codigo_Incidencia),"
+ "Fecha date, Hora time, Notificante varchar(50), Codigo_Comunicacion char(5)
REFERENCES COMUNICACION (Codigo_Comunicacion), "
+ "Responsable_Seguridad varchar(50))"
+ "WITHOUT OIDS;"

+ "CREATE TABLE REGISTRO_DE_RECUPERACION_DE_DATOS (Numero_Incidencia SERIAL
REFERENCES REGISTRO_DE_INCIDENCIAS (Numero_Incidencia), Id_Usuario varchar(50),"
+ "Datos text, Datos_Manuales text, Autorizada bool)"
+ "WITHOUT OIDS;"

+ "CREATE TABLE PERSONAL_AUTORIZADO (Login varchar(20) PRIMARY KEY, Password char(9)
NOT NULL, Id_Usuario varchar(50) NOT NULL, "
+ "Codigo_Credenciales char(6) REFERENCES CREDENCIALES (Codigo_Credenciales))
WITHOUT OIDS;"

+ "CREATE TABLE REGISTRO_DE_ACCESOS (Numero_Acceso SERIAL PRIMARY KEY, Id_Usuario
varchar(50), Id_Host varchar(50),"
+ "Credenciales varchar(50), Responsable_Seguridad varchar(50), Fecha date, Hora
time, Autorizado bool, "
+ "Fichero_Accedido varchar(50), Propiedades_Fichero_Accedido varchar(50),
Codigo_Denegacion char(5) REFERENCES MOTIVO_DE_DENEGACION (Codigo_Denegacion),"
+ "Id_Recurso varchar(50), URL varchar(500), Copia_Respaldo bool, Tipo_Acceso
varchar(50), Numero_Acceso_Fichero decimal,"
+ "Codigo_Acceso char(5) REFERENCES SITUACION_ACCESO (Codigo_Acceso),
Despersonalizacion bool)"
+ "WITHOUT OIDS;"

+ "CREATE TABLE REGISTRO_DE_RESPALDO (Numero_Acceso SERIAL REFERENCES
REGISTRO_DE_ACCESOS (Numero_Acceso), "
+ "Fecha_Copia_Respaldo date) WITHOUT OIDS;"

+ "CREATE TABLE REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS (Numero_Entrada SERIAL
PRIMARY KEY, Codigo_Soporte char(5) REFERENCES TIPO_DE_SOPORTE (Codigo_Soporte), "
+ "Responsable_Seguridad varchar(50), Emisor varchar(50), Receptor varchar(50),
Autorizacion_Escrita bool, Fecha date, Hora time,"
+ "Numero_Soportes decimal, Codigo_Envio char(5) REFERENCES FORMA_DE_ENVIO
(Codigo_Envio))"
+ "WITHOUT OIDS;"

+ "CREATE TABLE REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS (Numero_Salida SERIAL
PRIMARY KEY, Codigo_Soporte char(5) REFERENCES TIPO_DE_SOPORTE (Codigo_Soporte), "
+ "Responsable_Seguridad varchar(50), Receptor varchar(50), Emisor varchar(50),
Autorizacion_Escrita bool, Fecha date, Hora time,"
+ "Numero_Soportes decimal, Codigo_Envio char(5) REFERENCES FORMA_DE_ENVIO
(Codigo_Envio))"
+ "WITHOUT OIDS;"
;
ActualizarBaseDeDatos(conexion, sql);
```

```

// Creamos los campos indexados;
sql = ""
+ "CREATE INDEX Fecha_RA ON REGISTRO_DE_ACCESOS (Fecha);"
+ "CREATE INDEX Fichero_Accedido_RA ON REGISTRO_DE_ACCESOS (Fichero_Accedido);"
+ "CREATE INDEX Id_Usuario_RA ON REGISTRO_DE_ACCESOS (Id_Usuario);"
+ "CREATE INDEX Codigo_Soporte_RESI ON REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS
(Codigo_Soporte);"
+ "CREATE INDEX Emisor_RESI ON REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS
(Emisor);"
+ "CREATE INDEX Fecha_RESI ON REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS (Fecha);"
+ "CREATE INDEX Codigo_Comunicacion_RI ON REGISTRO_DE_INCIDENCIAS
(Codigo_Comunicacion);"
+ "CREATE INDEX Fecha_RI ON REGISTRO_DE_INCIDENCIAS (Fecha);"
+ "CREATE INDEX Codigo_Incidencia_RI ON REGISTRO_DE_INCIDENCIAS
(Codigo_Incidencia);"
+ "CREATE INDEX Fecha_RSSI ON REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS(Fecha);"
+ "CREATE INDEX Codigo_Soporte_RSSI ON REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS
(Codigo_Soporte);"
+ "CREATE INDEX Receptor_RSSI ON REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS
(Receptor);"
;
if (ActualizarBaseDeDatos(conexion, sql)) {}

// Añadimos los comentarios relativos a cada campo.
sql = ""
+ "COMMENT ON COLUMN COMUNICACION.Codigo_Comunicacion IS 'Valor Añadido. Recogerá el
código de las distintas posibilidades de comunicación de la incidencia.';"
+ "COMMENT ON COLUMN COMUNICACION.Descripcion IS 'Real Decreto 994/1999. Artículo
10. Recogerá a quien se le comunica la incidencia.';"

+ "COMMENT ON COLUMN FORMA_DE_ENVIO.Codigo_Envio IS 'Valor añadido. Recogerá el
código de envío que se relaciona con las distintas formas de envío de los ficheros.';"
+ "COMMENT ON COLUMN FORMA_DE_ENVIO.Descripcion IS 'Real Decreto 994/1999. Artículo
20. Describe la forma de envío de la información.';"

+ "COMMENT ON COLUMN INCIDENCIAS.Codigo_Incidencia IS 'Valor añadido. Recogerá el
código del tipo de incidencia que se ha producido.';"
+ "COMMENT ON COLUMN INCIDENCIAS.Descripcion IS 'Real Decreto 994/1999. Artículo 10.
Describe el tipo de incidencia que se ha producido.';"
+ "COMMENT ON COLUMN INCIDENCIAS.Efectos IS 'Real Decreto 994/1999. Artículo 10.
Recogerá los efectos que deriven de la incidencia que se ha producido.';"

+ "COMMENT ON COLUMN MOTIVO_DE_DENEGACION.Codigo_Denegacion IS 'Valor añadido.
Recogerá el código del tipo de denegación de acceso al fichero.';"
+ "COMMENT ON COLUMN MOTIVO_DE_DENEGACION.Descripcion IS 'Valor añadido. El motivo
por el cual no se ha autorizado el acceso.';"

+ "COMMENT ON COLUMN SITUACION_ACCESO.Codigo_Acceso IS 'Valor añadido. Recogerá el
código que será relacionado con una situación de acceso al recurso.';"
+ "COMMENT ON COLUMN SITUACION_ACCESO.Descripcion IS 'Ley 41/2002. Describe en que
contexto se accedio al recurso (judicial, profesional, investigación, docencia, el
paciente, ...).';"

+ "COMMENT ON COLUMN TIPO_DE_SOPORTE.Codigo_Soporte IS 'Valor añadido. Recogerá el
código del tipo de soporte.';"
+ "COMMENT ON COLUMN TIPO_DE_SOPORTE.Descripcion IS 'Real Decreto 994/1999. Artículo
20. Describe el tipo de soporte.';"

+ "COMMENT ON COLUMN CREDENCIALES.Codigo_Credenciales IS 'Valor añadido. Recogerá el
código que representa los permisos del personal autorizado en el Servidor de Registro de
Acceso';"
+ "COMMENT ON COLUMN CREDENCIALES.Descripcion IS 'Real Decreto 994/1999. Artículo
12. Recogerá la descripción de los permisos en el Servidor de Registro de Acceso .';"
+ "COMMENT ON COLUMN CREDENCIALES.Recursos IS 'Real Decreto 994/1999. Artículo 12.
Recogerá los valores numéricos que identifican los recursos en el Servidor de Registro
de Acceso.';"

+ "COMMENT ON COLUMN RECURSOS.Recurso IS 'Valor añadido. Recogerá el número con el
que se representa el recurso del Servidor de Registro de Acceso';"
+ "COMMENT ON COLUMN RECURSOS.Descripcion IS 'Valor añadido. Descripción del
recurso';"

+ "COMMENT ON COLUMN PERSONAL_AUTORIZADO.Login IS 'Valor añadido derivado del Real
Decreto 994/1999. Artículo 12. Alias del usuario autorizado en el Servidor de Registro
de Acceso.';"

```

```
+ "COMMENT ON COLUMN PERSONAL_AUTORIZADO.Password IS 'Valor añadido derivado del
Real Decreto 994/1999. Artículo 12. Contraseña del usuario autorizado en el Servidor de
Registro de Acceso.';"
+ "COMMENT ON COLUMN PERSONAL_AUTORIZADO.Id_Usuario IS 'Valor añadido derivado del
Real Decreto 994/1999. Artículo 12. Identificación del usuario autorizado en el Servidor
de Registro de Acceso.';"
+ "COMMENT ON COLUMN PERSONAL_AUTORIZADO.Codigo_Credenciales IS 'Valor añadido.
Recogerá un código que representa los permisos del personal autorizado en el Servidor de
Registro de Acceso.';"

+ "COMMENT ON COLUMN REGISTRO_DE_RESPALDO.Numero_Acceso IS 'Valor añadido. Enumera
los accesos que se producen.';"
+ "COMMENT ON COLUMN REGISTRO_DE_RESPALDO.Fecha_Copia_Respaldo IS 'Valor añadido.
Recogerá la fecha de la copia de respaldo.';"

+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Numero_Acceso IS 'Valor añadido. Enumera
los accesos que se producen.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Id_Usuario IS 'Real Decreto 994/1999.
Artículo 24. Recogerá la identificación del usuario.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Id_Host IS 'Valor añadido. Recogerá el Host
del cual se accede al fichero.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Credenciales IS 'Valor añadido. Recogerá
los permisos que posee el usuario en el momento de acceso al fichero.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Responsable_Seguridad IS 'Valor añadido.
Persona que por ley se responsabiliza de la seguridad de los ficheros.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Fecha IS 'Real Decreto 994/1999. Artículo
24. Recibirá la fecha en el momento de producirse el acceso.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Hora IS 'Real Decreto 994/1999. Artículo
24. Recibirá la hora en el momento de producirse el acceso.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Autorizado IS 'Real Decreto 994/1999.
Artículo 24. Recogerá si el acceso al fichero ha sido autorizado.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Fichero_Accedido IS 'Real Decreto 994/1999.
Artículo 24. Recogerá el nombre del fichero accedido.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Propiedades_Fichero_Accedido IS 'Real
Decreto 994/1999. Artículo 24. Recogerá información sobre el fichero accedido.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Codigo_Denegacion IS 'Valor añadido.
Recogerá el código del tipo de denegación de acceso al fichero.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Id_Recurso IS 'Valor añadido motivado por
el Real Decreto 994/1999. Artículo 24. Recogerá el nombre del recurso al que se
accede.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.URL IS 'Valor añadido motivado por el Real
Decreto 994/1999. Artículo 24. Recogerá el URL del fichero al que se accede.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Copia_Respaldo IS 'Valor añadido. Indica si
en el momento de acceso del fichero existe una copia de respaldo del mismo.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Tipo_Acceso IS 'Real Decreto 994/1999.
Artículo 24. Recogerá el tipo de acceso al fichero (lectura, escritura,...).'"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Numero_Acceso_Fichero IS 'Valor añadido.
Enumera los accesos al fichero.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Codigo_Acceso IS 'Valor añadido. Recogerá
el código de la situación en que se accede al recurso.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ACCESOS.Despersonalizacion IS 'Valor añadido.
Señala si se despersonalizó el fichero accedido.';"

+ "COMMENT ON COLUMN REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.Numero_Entrada IS
'Valor añadido. Enumera las entradas de soportes que se producen.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.Codigo_Soporte IS
'Valor añadido. Recogerá el código del tipo de soporte informático recibido.';"
+ "COMMENT ON COLUMN
REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.Responsable_Seguridad IS 'Valor añadido.
Persona que por ley se responsabiliza de la seguridad de los ficheros.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.Emisor IS 'Real
Decreto 994/1999. Artículo 20. Identifica al emisor de la información.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.Receptor IS 'Real
Decreto 994/1999. Artículo 20. Identifica al receptor de la información.';"
+ "COMMENT ON COLUMN
REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.Autorizacion_Escrita IS 'Real Decreto
994/1999. Artículo 20. Recoge si el receptor posee la autorización.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.Fecha IS 'Real
Decreto 994/1999. Artículo 20. Recibirá la fecha en el momento de producirse la entrada
del soporte.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.Hora IS 'Real
Decreto 994/1999. Artículo 20. Recibirá la hora en el momento de producirse la entrada
del soporte.';"
+ "COMMENT ON COLUMN REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.Numero_Soportes IS
'Real Decreto 994/1999. Artículo 20. Recoge el número de soportes recibidos.';"
```

```

+ "COMMENT ON COLUMN REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.Codigo_Envio IS
'Valor añadido. Recogerá el código de envío que se relaciona con las distintas formas de
envío de los ficheros.';"

+ "COMMENT ON COLUMN REGISTRO_DE_INCIDENCIAS.Numero_Incidencia IS 'Valor añadido.
Enumera las incidencias de seguridad que se producen.';"
+ "COMMENT ON COLUMN REGISTRO_DE_INCIDENCIAS.Codigo_Incidencia IS 'Valor añadido.
Recogerá el código del tipo de incidencia que se ha producido.';"
+ "COMMENT ON COLUMN REGISTRO_DE_INCIDENCIAS.Fecha IS 'Real Decreto 994/1999.
Artículo 10. Recibirá la fecha en el momento de producirse una incidencia.';"
+ "COMMENT ON COLUMN REGISTRO_DE_INCIDENCIAS.Hora IS 'Real Decreto 994/1999.
Artículo 10. Recibirá la hora en el momento de producirse una incidencia.';"
+ "COMMENT ON COLUMN REGISTRO_DE_INCIDENCIAS.Notificante IS 'Real Decreto 994/1999.
Artículo 10. Recogerá quien que notifica la incidencia.';"
+ "COMMENT ON COLUMN REGISTRO_DE_INCIDENCIAS.Codigo_Comunicacion IS 'Valor Añadido.
Recogerá el código de las distintas posibilidades de comunicación de la incidencia.';"
+ "COMMENT ON COLUMN REGISTRO_DE_INCIDENCIAS.Responsable_Seguridad IS 'Valor
añadido. Persona que por ley se responsabiliza de la seguridad de los ficheros.';"

+ "COMMENT ON COLUMN REGISTRO_DE_RECUPERACION_DE_DATOS.Numero_Incidencia IS 'Valor
añadido. Enumera las incidencias de seguridad que se producen.';"
+ "COMMENT ON COLUMN REGISTRO_DE_RECUPERACION_DE_DATOS.Id_Usuario IS 'Real Decreto
994/1999. Artículo 21. Persona que ejecuto el proceso de recuperación de datos.';"
+ "COMMENT ON COLUMN REGISTRO_DE_RECUPERACION_DE_DATOS.Datos IS 'Real Decreto
994/1999. Artículo 21. Datos restaurados.';"
+ "COMMENT ON COLUMN REGISTRO_DE_RECUPERACION_DE_DATOS.Datos_Manuales IS 'Real
Decreto 994/1999. Artículo 21. Datos grabados manualmente en el proceso de
recuperación.';"
+ "COMMENT ON COLUMN REGISTRO_DE_RECUPERACION_DE_DATOS.Autorizada IS 'Real Decreto
994/1999. Artículo 21. Para la ejecución de los procedimientos de recuperación de datos
es necesaria la autorización por escrito del responsable del fichero.';"

+ "COMMENT ON COLUMN REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.Numero_Salida IS
'Valor añadido. Enumera las salidas de soportes que se producen.';"
+ "COMMENT ON COLUMN REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.Codigo_Soporte IS
'Valor añadido. Recogerá el código del tipo de soporte informático enviado.';"
+ "COMMENT ON COLUMN
REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.Responsable_Seguridad IS 'Valor añadido.
Persona que por ley se responsabiliza de la seguridad de los ficheros.';"
+ "COMMENT ON COLUMN REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.Receptor IS 'Real
Decreto 994/1999. Artículo 20. Identifica al receptor de la información.';"
+ "COMMENT ON COLUMN REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.Emisor IS 'Real
Decreto 994/1999. Artículo 20. Identifica al emisor de la información.';"
+ "COMMENT ON COLUMN
REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.Autorizacion_Escrita IS 'Real Decreto
994/1999. Artículo 20. Recoge sie el emisor posee la autorización.';"
+ "COMMENT ON COLUMN REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.Fecha IS 'Real
Decreto 994/1999. Artículo 20. Recibirá la fecha en el momento de producirse la salida
del soporte.';"
+ "COMMENT ON COLUMN REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.Hora IS 'Real
Decreto 994/1999. Artículo 20. Recibirá la hora en el momento de producirse la salida
del soporte.';"
+ "COMMENT ON COLUMN REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.Numero_Soportes IS
'Real Decreto 994/1999. Artículo 20. Recoge el número de soportes enviados.';"
+ "COMMENT ON COLUMN REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.Codigo_Envio IS
'Valor añadido. Recogerá el código de envío que se relaciona con las distintas formas de
envío de los ficheros.';"
;
if (ActualizarBaseDeDatos(conexion, sql)) {}

Conexion.CerrarConexion(conexion);
System.out.println("RegistroDeAcceso.java: La base de datos Registro de Acceso ha
sido creada");
}

catch(SQLException sqlEx)
{ System.out.println("RegistroDeAcceso.java: Excepcion SQL al crear las tablas
de la base de datos: " + sqlEx.getMessage()); }
}

/**
 * Compueba si la base de datos "Registro de Acceso" ha sido creada.
 * Para ello consultamos los campos de la última tabla creada por el método
 * CrearBaseDeDatos(). En función de los resultados obtenidos en la consulta se
 * devolverá un boolean, true = la tabla existe y por tanto la base de datos ha
 * sido creada, false = la tabla no existe, no se ha creado la base de datos.

```

```
* @return Boolean según se haya creado o no la base de datos "Registro de Acceso"
*/
public static boolean ComprobarBaseDeDatos ()
{
    boolean resultado;
    conexion = null;
    sql = "SELECT * FROM TIPO_DE_SOPORTE";

    try {
        conexion = Conexion.EstablecerConexion();
        rs = ConsultarBaseDeDatos(conexion, sql);
        if (rs==null) { resultado = false; }
        else { resultado = true; }
    }

    catch (Exception Ex)
    { resultado = false;
      System.out.println("RegistroDeAcceso.java: La base de datos no ha sido creada");
    }

    return resultado;
}

/**
 * Consulta la base de datos referenciada en conexion con la sentencia SQL
 * pasada como argumento, devolviendo los resultados en el objeto ResultSet rs.
 * @return Retorna un ResultSet con el resultado de la consulta.
 * @param conexion Connection de la base de datos a consultar.
 * @param sql sentencia SQL a ejecutar.
 * @throws java.sql.SQLException al consultar la base de datos.
 */
public static ResultSet ConsultarBaseDeDatos (Connection conexion, String sql)
throws SQLException
{
    try {
        Statement st = conexion.createStatement();
        rs = st.executeQuery(sql);
    }

    catch(SQLException sqlEx)
    { rs = null;
      System.out.println("RegistroDeAcceso.java: Excepción SQL a consultar la base de
datos: " + sqlEx.getMessage());
    }

    return rs;
}

/**
 * Actualiza la base de datos referenciada en conexion, con la secuencia SQL
 * pasada como argumento. Se retorna un boolean indicando si la actualización
 * fue o no exitosa.
 * @return boolean true = actualización , false = no actualización.
 * @param conexion Connection de la base de datos a consultar.
 * @param sql Secuencia SQL que se ejecuta para actualizar la base de datos.
 * @throws java.sql.SQLException al actualizar la base de datos.
 */
public static boolean ActualizarBaseDeDatos(Connection conexion, String sql)
throws SQLException
{
    boolean Respuesta = false;
    try { Statement st = conexion.createStatement();
        st.executeUpdate(sql);
        Respuesta = true;
    }

    catch(SQLException sqlEx)
    { System.out.println("RegistroDeAcceso.java: Excepcion SQL al actualizar la base
de datos: " + sqlEx.getMessage()); }

    return Respuesta;
}
}
```

9.3. Clase Calendario.java.-

```

/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Calendario.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

import java.util.GregorianCalendar;
import java.util.Calendar;

/**
 * Clase utilizada por el Servidor de Registro de Acceso para obtener
 * la fecha y la hora en el instante que se precise.
 * Con funciones de las clases java.util.Calendar y
 * java.util.GregorianCalendar creamos el formato
 * de fecha y hora establecido por PostgreSQL Database Server 8.0 para los
 * campos tipo date y time.
 */
public class Calendario {

    /** GregorianCalendar que recogerá la información temporal actual*/
    GregorianCalendar calendario = new GregorianCalendar();

    /**
     * Método que devuelve la fecha actual en formato mm/dd/aaaa.
     * Donde: mm = mes del 01 al 12. Ej. Enero = 01, Diciembre = 12.
     *        dd = día del 01 al 31.
     *        aaaa = año con cuatro dígitos. Ej. 1999, 2005
     * Ejemplo de fecha: 03/06/2004 = 6 de Marzo de 2004.
     * Formato requerido por PostgreSQL Database Server 8.0 para los campos tipo date.
     * @return String Fecha en formato mm/dd/aaaa.
     */
    public String ObtenerFecha ()
    {
        String Fecha = Integer.toString(calendario.get(Calendar.MONTH)+1) + "/"
            + Integer.toString(calendario.get(Calendar.DATE)) + "/"
            + Integer.toString(calendario.get(Calendar.YEAR));

        return Fecha;
    }

    /**
     * Método que devuelve la hora actual en formato hh:mm:ss.
     * Donde: hh = hora de 00 a 23. Ej. 00 = doce de la noche, 12 = doce del mediodía.
     *        mm = minutos de 00 a 59.
     *        ss = segundos de 00 a 59.
     * Ejemplo de hora: 14:09:06.
     * Formato requerido por PostgreSQL Database Server 8.0 para los campos tipo time.
     * @return String Hora en formato hh/mm/ss.
     */
    public String ObtenerHora ()
    {
        String Hora = Integer.toString(calendario.get(Calendar.HOUR_OF_DAY)) + ":"
            + Integer.toString(calendario.get(Calendar.MINUTE)) + ":"
            + Integer.toString(calendario.get(Calendar.SECOND));

        return Hora;
    }
}

```

9.4. Clase ConsultasEstablecidas.java.-

```

/**
*****
                Proyecto final de carrera.
        Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ConsultasEstablecidas.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

import ardb.*;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.ResultSetMetaData;
import java.util.Vector;

/**
 * Clase utilizada por el servidor de Registro de Acceso para obtener
 * información de la base de datos "Registro de Acceso" con consultas establecidas
 * por el diseñador.
 * Toda consulta establecida utilizada para informar a los usuarios del servidor,
 * así como aquellas diseñadas para servicios de gestión y mantenimiento de la
 * base de datos, se encuentran en los métodos de esta clase.
 * Las consultas a la base de datos "Registro de Acceso" están ordenadas
 * numéricamente, de modo que la forma de elegir una u otra dependerá del valor
 * del parámetro tipo int Elección, que se pasa como argumento.
 * Una vez realizada la consulta, se prepara el formato del resultado para facilitar
 * su gestión por el cliente.
 */
public class ConsultasEstablecidas {

    /** Connection para conectarse con la base de datos. */
    private static Connection conexion = null;
    /** String con la sentencia SQL a ejecutar */
    private static String Sql;
    /** ResultSet con el resultado de la consulta */
    protected static ResultSet rs = null;

    /**
     * Método que ejecuta una u otra consulta en función del valor de Eleccion que se pasa
     * como argumento.
     * Cuando se llama al método se designa la sentencia SQL a comunicar en función
     * de Eleccion, se guardan los parámetros necesarios para generar un informe
     * PDF, se consulta la base de datos y se convierte el resultado a un Vector,
     * cuyos elementos son objetos de la clase ParametrosDeConsulta.java,
     * que es devuelto al convocante.
     *
     * Resumen de consultas.-
     * Se señalan las posibles consultas a realizar del siguiente modo:
     * Eleccion. Nombre de la consulta. Número de parámetros por elemento del Vector.
     *
     * Consultas generales de todos los campos. 1-15.
     * 1. Consulta de todos los campos de la tabla comunicacion. 2 parámetros/elemento.
     * 2. Consulta de todos los campos de la tabla credenciales. 3 parámetros/elemento
     * 3. Consulta de todos los campos de la tabla forma_de_envio. 2 parámetros/elemento
     * 4. Consulta de todos los campos de la tabla incidencias. 2 parámetros/elemento
     * 5. Consulta de todos los campos de la tabla motivo_de_denegacion. 2
    parámetros/elemento
     * 6. Consulta de todos los campos de la tabla personal_autorizado. 4
    parámetros/elemento
     * 7. Consulta de todos los campos de la tabla situacion_acceso. 2 parámetros/elemento
     * 8. Consulta de todos los campos de la tabla tipo_de_soporte. 2 parámetros/elemento
     * 9. Consulta de todos los campos de la tabla registro_de_acceso. 18
    parámetros/elemento
    
```

```

* 10. Consulta de todos los campos de la tabla
registro_de_entrada_de_soportes_informaticos. 10 parámetros/elemento
* 11. Consulta de todos los campos de la tabla registro_de_incidencias. 7
parámetros/elemento
* 12. Consulta de todos los campos de la tabla registro_de_recuperacion_de_datos. 5
parámetros/elemento
* 13. Consulta de todos los campos de la tabla
registro_de_salida_de_soportes_informaticos. 10 parámetros/elemento
* 14. Consulta de todos los campos de la tabla registro_de_respaldo. 2
parámetros/elemento
* 15. Consulta de todos los campos de la tabla recursos. 2 parámetros/elemento
*
* Consultas específicas diseñadas por el programador. 16-23
* 16. Consulta resumen de las incidencias registradas. 7 parámetros/elemento.
* 17. Consulta resumen de las incidencias recuperación de datos registradas. 7
parámetros/elemento.
* 18. Consulta resumen de las entradas de soportes al sistema. 8 parámetros/elemento.
* 19. Consulta resumen de las salidas de soportes del sistema. 8 parámetros/elemento.
* 20. Consulta resumen de los accesos al sistema según situación. 8
parámetros/elemento.
* 21. Consulta resumen de los accesos a los ficheros. 9 parámetros/elemento.
* 22. Consulta resumen de los ficheros accedidos. 12 parámetros/elemento.
* 23. Consulta resumen de los accesos a los ficheros con copia de respaldo. 6
parámetros/elemento.
*
* @param Eleccion Entero que recoge la elección de la consulta de 1 a 23.
* @return Retorna Vector con el resultado de la consulta. Cada elemento es un
* objeto de la clase ParametrosDeConsulta.java.
*/
public static Vector Estaticas (int Eleccion)
{ switch (Eleccion)
{
case 1: Sql = "SELECT * FROM COMUNICACION ORDER BY codigo_comunicacion"; break;
case 2: Sql = "SELECT * FROM CREDENCIALES ORDER BY codigo_credenciales"; break;
case 3: Sql = "SELECT * FROM FORMA_DE_ENVIO ORDER BY codigo_envio"; break;
case 4: Sql = "SELECT * FROM INCIDENCIAS ORDER BY codigo_incidencia"; break;
case 5: Sql = "SELECT * FROM MOTIVO_DE_DENEGACION ORDER BY codigo_denegacion";
break;
case 6: Sql = "SELECT * FROM PERSONAL_AUTORIZADO ORDER BY id_usuario"; break;
case 7: Sql = "SELECT * FROM SITUACION_ACCESO ORDER BY codigo_acceso"; break;
case 8: Sql = "SELECT * FROM TIPO_DE_SOPORTE ORDER BY codigo_soporte"; break;
case 9: Sql = "SELECT * FROM REGISTRO_DE_ACCESOS"; break;
case 10: Sql = "SELECT * FROM REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS"; break;
case 11: Sql = "SELECT * FROM REGISTRO_DE_INCIDENCIAS"; break;
case 12: Sql = "SELECT * FROM REGISTRO_DE_RECUPERACION_DE_DATOS"; break;
case 13: Sql = "SELECT * FROM REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS"; break;
case 14: Sql = "SELECT * FROM REGISTRO_DE_RESPALDO"; break;
case 15: Sql = "SELECT * FROM RECURSOS ORDER BY recurso"; break;

case 16: Sql = "SELECT "
+ "REGISTRO_DE_INCIDENCIAS.numero_incidencia,"
+ "INCIDENCIAS.descripcion,"
+ "INCIDENCIAS.efectos,"
+ "COMUNICACION.descripcion,"
+ "REGISTRO_DE_INCIDENCIAS.fecha,"
+ "REGISTRO_DE_INCIDENCIAS.hora,"
+ "REGISTRO_DE_INCIDENCIAS.notificante "
+ "FROM INCIDENCIAS, COMUNICACION, REGISTRO_DE_INCIDENCIAS WHERE "
+ "REGISTRO_DE_INCIDENCIAS.codigo_incidencia = INCIDENCIAS.codigo_incidencia "
+ "AND REGISTRO_DE_INCIDENCIAS.codigo_comunicacion = "
COMUNICACION.codigo_comunicacion ORDER BY "
+ "REGISTRO_DE_INCIDENCIAS.numero_incidencia;"; break;

case 17: Sql = "SELECT "
+ "REGISTRO_DE_INCIDENCIAS.fecha,"
+ "REGISTRO_DE_INCIDENCIAS.hora,"
+ "REGISTRO_DE_RECUPERACION_DE_DATOS.datos,"
+ "REGISTRO_DE_RECUPERACION_DE_DATOS.datos_manuales,"
+ "REGISTRO_DE_RECUPERACION_DE_DATOS.id_usuario,"
+ "REGISTRO_DE_RECUPERACION_DE_DATOS.autorizada,"
+ "REGISTRO_DE_INCIDENCIAS.notificante "
+ "FROM REGISTRO_DE_INCIDENCIAS, REGISTRO_DE_RECUPERACION_DE_DATOS WHERE "
+ "REGISTRO_DE_INCIDENCIAS.numero_incidencia = "
REGISTRO_DE_RECUPERACION_DE_DATOS.numero_incidencia "
+ "ORDER BY REGISTRO_DE_INCIDENCIAS.fecha, REGISTRO_DE_INCIDENCIAS.hora;"; break;

```

```

case 18: Sql = "SELECT "
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.fecha,"
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.hora,"
+ "TIPO_DE_SOPORTE.descripcion,"
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.numero_soportes,"
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.emisor,"
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.receptor,"
+ "FORMA_DE_ENVIO.descripcion,"
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.autorizacion_escrita "
+ "FROM REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS, FORMA_DE_ENVIO,
TIPO_DE_SOPORTE WHERE "
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.codigo_envio =
FORMA_DE_ENVIO.codigo_envio "
+ "AND REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.codigo_soporte =
TIPO_DE_SOPORTE.codigo_soporte "
+ "ORDER BY REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.fecha,
REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.hora;"; break;

case 19: Sql = "SELECT "
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.fecha,"
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.hora,"
+ "TIPO_DE_SOPORTE.descripcion,"
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.numero_soportes,"
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.receptor,"
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.emisor,"
+ "FORMA_DE_ENVIO.descripcion,"
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.autorizacion_escrita "
+ "FROM REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS, FORMA_DE_ENVIO, TIPO_DE_SOPORTE
WHERE "
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.codigo_envio =
FORMA_DE_ENVIO.codigo_envio "
+ "AND REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.codigo_soporte =
TIPO_DE_SOPORTE.codigo_soporte "
+ "ORDER BY REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.fecha,
REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.hora;"; break;

case 20: Sql = "SELECT "
+ "REGISTRO_DE_ACCESOS.fecha,"
+ "REGISTRO_DE_ACCESOS.hora,"
+ "REGISTRO_DE_ACCESOS.id_usuario,"
+ "REGISTRO_DE_ACCESOS.id_recurso,"
+ "REGISTRO_DE_ACCESOS.fichero_accedido,"
+ "REGISTRO_DE_ACCESOS.numero_acceso_fichero,"
+ "SITUACION_ACCESO.descripcion,"
+ "REGISTRO_DE_ACCESOS.despersonalizacion "
+ "FROM REGISTRO_DE_ACCESOS, SITUACION_ACCESO WHERE "
+ "REGISTRO_DE_ACCESOS.codigo_acceso = SITUACION_ACCESO.codigo_acceso "
+ "ORDER BY REGISTRO_DE_ACCESOS.fecha, REGISTRO_DE_ACCESOS.hora;"; break;

case 21: Sql = "SELECT "
+ "REGISTRO_DE_ACCESOS.fichero_accedido,"
+ "REGISTRO_DE_ACCESOS.autorizado,"
+ "MOTIVO_DE_DENEGACION.descripcion,"
+ "REGISTRO_DE_ACCESOS.tipo_acceso,"
+ "REGISTRO_DE_ACCESOS.fecha,"
+ "REGISTRO_DE_ACCESOS.hora,"
+ "REGISTRO_DE_ACCESOS.id_usuario,"
+ "REGISTRO_DE_ACCESOS.credenciales,"
+ "SITUACION_ACCESO.descripcion "
+ "FROM REGISTRO_DE_ACCESOS, MOTIVO_DE_DENEGACION, SITUACION_ACCESO WHERE "
+ "REGISTRO_DE_ACCESOS.codigo_denegacion = MOTIVO_DE_DENEGACION.codigo_denegacion "
+ "AND REGISTRO_DE_ACCESOS.codigo_acceso = SITUACION_ACCESO.codigo_acceso "
+ "ORDER BY REGISTRO_DE_ACCESOS.fichero_accedido, REGISTRO_DE_ACCESOS.fecha,
REGISTRO_DE_ACCESOS.hora;"; break;

case 22: Sql = "SELECT "
+ "REGISTRO_DE_ACCESOS.fecha,"
+ "REGISTRO_DE_ACCESOS.hora,"
+ "REGISTRO_DE_ACCESOS.fichero_accedido,"
+ "REGISTRO_DE_ACCESOS.propiedades_fichero_accedido,"
+ "MOTIVO_DE_DENEGACION.descripcion,"
+ "REGISTRO_DE_ACCESOS.tipo_acceso,"
+ "REGISTRO_DE_ACCESOS.numero_acceso_fichero,"
+ "REGISTRO_DE_ACCESOS.id_recurso,"
+ "REGISTRO_DE_ACCESOS.url,"
+ "REGISTRO_DE_ACCESOS.id_host,"

```

```

+ "REGISTRO_DE_ACCESOS.id_usuario,"
+ "REGISTRO_DE_ACCESOS.copia_respaldo "
+ "FROM REGISTRO_DE_ACCESOS, MOTIVO_DE_DENEGACION WHERE "
+ "REGISTRO_DE_ACCESOS.codigo_denegacion = MOTIVO_DE_DENEGACION.codigo_denegacion "
+ "ORDER BY REGISTRO_DE_ACCESOS.fecha, REGISTRO_DE_ACCESOS.hora,
REGISTRO_DE_ACCESOS.fichero_accedido;"; break;

case 23: Sql = "SELECT "
+ "REGISTRO_DE_ACCESOS.fecha,"
+ "REGISTRO_DE_ACCESOS.hora,"
+ "REGISTRO_DE_ACCESOS.fichero_accedido,"
+ "REGISTRO_DE_ACCESOS.id_usuario,"
+ "REGISTRO_DE_ACCESOS.id_recurso,"
+ "REGISTRO_DE_RESPALDO.fecha_copia_respaldo "
+ "FROM REGISTRO_DE_ACCESOS, REGISTRO_DE_RESPALDO WHERE "
+ "REGISTRO_DE_ACCESOS.numero_acceso = REGISTRO_DE_RESPALDO.numero_acceso "
+ "ORDER BY REGISTRO_DE_ACCESOS.fecha, REGISTRO_DE_ACCESOS.hora,
REGISTRO_DE_ACCESOS.fichero_accedido;"; break;
}

Vector vectorConsulta = new Vector();

// Introducimos en ParametrosPDF los datos necesarios.
ParametrosPDF.setSql(Sql);
ParametrosPDF.setNombresCampos(RetornarNombreCampos(Sql));

try { conexion = Conexion.EstablecerConexion();
    rs = RegistroDeAcceso.ConsultarBaseDeDatos(conexion, Sql);
    vectorConsulta = CrearVectorConsulta(rs);
}

catch (Exception Ex)
{ System.out.println("ConsultasEstablecidas.java: No se ha podido establecer la
conexión con la base de datos");
  System.out.println(Ex.getMessage());
}

return vectorConsulta;
}

/**
 * Clase que obtiene información de los registros principales de la base de datos
 * para unas fechas concretas.
 * Este método ejecuta una u otra consulta en función de los valores de Eleccion,
 * Fecha_Inferior y Fecha_Superior, que se pasan como argumento.
 * Cuando se llama al método se complementa la sentencia SQL a comunicar,
 * se guardan los parámetros necesarios para generar un informe PDF, se
 * consulta la base de datos y se convierte el resultado a un Vector, cuyos
 * elementos pertenecen a la clase ParametrosDeConsulta.java, que es devuelto
 * al convocante.
 *
 * Resumen de consultas.-
 * Se señalan las posibles consultas a realizar del siguiente modo:
 * Eleccion. Nombre de la consulta. Número de parámetros por elemento del Vector.
 *
 * Consultas dinámicas 30-33.
 * 30. Consulta de incidencias por fechas. 5 parámetros/elemento.
 * 31. Consulta de accesos por fechas. 5 parámetros/elemento
 * 32. Consulta de entrada de soportes por fecha. 5 parámetros/elemento
 * 33. Consulta de salida de soportes por fecha. 5 parámetros/elemento
 *
 * @param Fecha_Inferior Fecha en formato mm/dd/aaaa desde la cual se quiere consultar.
 * @param Fecha_Superior Fecha en formato mm/dd/aaaa hasta la cual se quiere consultar.
 * @param Eleccion Entero que recoge la elección de la consulta de 30 a 33.
 * @return Retorna Vector con el resultado de la consulta. Cada elemento es un
 * objeto de la clase ParametrosDeConsulta.java.
 */
public static Vector Dinamicas (String Fecha_Inferior, String Fecha_Superior, int
Eleccion)
{ switch (Eleccion)
{
case 30: Sql = ""
+ "SELECT "
+ "REGISTRO_DE_INCIDENCIAS.fecha,"
+ "REGISTRO_DE_INCIDENCIAS.hora,"
+ "REGISTRO_DE_INCIDENCIAS.codigo_incidencia, "

```

```

+ "INCIDENCIAS.descripcion,"
+ "INCIDENCIAS.efectos "
+ "FROM INCIDENCIAS, REGISTRO_DE_INCIDENCIAS WHERE "
+ "REGISTRO_DE_INCIDENCIAS.codigo_incidencia = INCIDENCIAS.codigo_incidencia "
+ "AND fecha > '"+Fecha_Inferior+"' AND fecha < '"+Fecha_Superior+"' ORDER BY "
+ "REGISTRO_DE_INCIDENCIAS.fecha, REGISTRO_DE_INCIDENCIAS.hora;"; break;

case 31: Sql = ""
+ "SELECT "
+ "REGISTRO_DE_ACCESOS.fecha,"
+ "REGISTRO_DE_ACCESOS.hora,"
+ "REGISTRO_DE_ACCESOS.id_usuario,"
+ "REGISTRO_DE_ACCESOS.id_recurso,"
+ "REGISTRO_DE_ACCESOS.fichero_accedido "
+ "FROM REGISTRO_DE_ACCESOS WHERE "
+ "fecha > '"+Fecha_Inferior+"' AND fecha < '"+Fecha_Superior+"' ORDER BY "
+ "REGISTRO_DE_ACCESOS.fecha, REGISTRO_DE_ACCESOS.hora;"; break;

case 32: Sql = ""
+ "SELECT "
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.fecha,"
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.hora,"
+ "TIPO_DE_SOPORTE.descripcion,"
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.emisor,"
+ "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.receptor "
+ "FROM REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS WHERE "
+ "fecha > '"+Fecha_Inferior+"' AND fecha < '"+Fecha_Superior+"' "
+ "AND REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.codigo_soporte = "
TIPO_DE_SOPORTE.codigo_soporte "
+ "ORDER BY REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.fecha,
REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS.hora;"; break;

case 33: Sql = ""
+ "SELECT "
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.fecha,"
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.hora,"
+ "TIPO_DE_SOPORTE.descripcion,"
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.receptor,"
+ "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.emisor "
+ "FROM REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS WHERE "
+ "fecha > '"+Fecha_Inferior+"' AND fecha < '"+Fecha_Superior+"' "
+ "AND REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.codigo_soporte = "
TIPO_DE_SOPORTE.codigo_soporte "
+ "ORDER BY REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.fecha,
REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS.hora;"; break;
}

Vector vectorConsulta = new Vector();

// Introducimos en ParametrosPDF los datos necesarios.
ParametrosPDF.setSql(Sql);
ParametrosPDF.setNombresCampos(RetornarNombreCampos(Sql));

try { conexion = Conexion.EstablecerConexion();
    rs = RegistroDeAcceso.ConsultarBaseDeDatos(conexion, Sql);
    vectorConsulta = CrearVectorConsulta(rs);
}

catch (Exception Ex)
{ System.out.println("No se ha podido establecer la conexión con la base de
datos");
    System.out.println(Ex.getMessage());
}

return vectorConsulta;
}

/**
 * Método que devuelve el texto "Consulta número: ?", donde "?" es el número con
 * el que se conoce a la consulta realizada por el usuario.
 * Este texto será almacenado en el campo "fichero_accedido" de la tabla
 * "registro_de_accesos" cuando el usuario haga algún tipo de consulta.
 * Las consultas a la base de datos "Registro de Acceso" son recursos del servidor
 * y por tanto una acceso a ellas debe ser registrado.
 * @param Eleccion numero de consulta realizada.

```

```

    * @return String texto "Consulta número: ? ".
    */
    public static String TextoConsulta (int Eleccion)
    {
        String Consulta = "Consulta número : "+Eleccion;
        return Consulta;
    }

    /**
     * Método que devuelve el campo "descripcion" de la tabla "incidencias" relacionado
     * con el "codigo_incidencia" pasado internamente al método por la clase
     * ControlIncidencia.java. Es necesario por tanto haber introducido el código
     * de incidencia en la clase ControlIncidencia.java.
     * Cuando se llama a este método se complementa la sentencia SQL a ejecutar,
     * se consulta la base de datos y se devuelve el resultado.
     * @return String con el valor del campo descripcion de la tabla incidencia.
     */
    public static String TextoCodigoIncidencia()
    {
        String Sql = "SELECT INCIDENCIAS.descripcion FROM INCIDENCIAS WHERE codigo_incidencia =
        '"+ControlIncidencia.getCodigoIncidencia()+"'";
        String Mensaje = "Ha habido un error al consultar la descripcion de la incidencia";

        try { conexion = Conexion.EstablecerConexion();
            rs = RegistroDeAcceso.ConsultarBaseDeDatos(conexion, Sql);
            while (rs.next())
                { Mensaje = rs.getString(1); }
        }
        catch (Exception Ex)
        { System.out.println("ConsultasEstablecidas.java: No se ha podido establecer la
        conexión con la base de datos");
            System.out.println(Ex.getMessage());
        }

        return Mensaje;
    }

    /**
     * Método que devuelve el número de registro próximo para las tablas
     * principales de la base de datos.
     * En función de la opción elegida se seleccionará una u otra tabla.
     *
     * Resumen de opciones.-
     * Opcion. Nombre de la tabla.
     * 9. Tabla registro_de_accesos.
     * 10. Tabla registro_de_entrada_de_soportes_informaticos.
     * 11. Tabla registro_de_incidencias.
     * 13. Tabla registro_de_salida_de_soportes_informaticos.
     *
     * @param Opcion int cuyo valor selecciona la tabla.
     * @return int con el número proximo de registro.
     */
    public static int NumeroDeRegistro (int Opcion)
    {
        int NumeroDeRegistro = 0;

        switch (Opcion)
        { case 9: Sql = "SELECT max(numero_acceso) FROM REGISTRO_DE_ACCESOS"; break;
          case 10: Sql = "SELECT max(numero_entrada) FROM
          REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS"; break;
          case 11: Sql = "SELECT max(numero_incidencia) FROM REGISTRO_DE_INCIDENCIAS";
          break;
          case 13: Sql = "SELECT max(numero_salida) FROM
          REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS"; break;
        }

        try { conexion = Conexion.EstablecerConexion();
            rs = RegistroDeAcceso.ConsultarBaseDeDatos(conexion, Sql);
            while (rs.next()) { NumeroDeRegistro = rs.getInt("max"); }
        }

        catch (Exception Ex)
        { System.out.println("ConsultasEstablecidas.java: Error al obtener número de
        secuencia en la base de datos");
            System.out.println(Ex.getMessage());
        }
    }

```

```
        return NumeroDeRegistro;
    }

    /**
     * Método que consulta el campo "fichero_accedido" de la tabla "registro_de_accesos"
     * para obtener el número de veces que se ha consultado un mismo fichero.
     * @param Nombre_Fichero String con el nombre del fichero.
     * @return int el número de veces que el fichero ha sido hallado en la tabla.
     */
    public static int NumeroDeFichero (String Nombre_Fichero)
    {
        int NumeroDeFichero = 0;
        Sql = "SELECT count(fichero_accedido) FROM REGISTRO_DE_ACCESOS WHERE fichero_accedido
        like '"+Nombre_Fichero+"'";

        try { conexion = Conexion.EstablecerConexion();
            rs = RegistroDeAcceso.ConsultarBaseDeDatos(conexion, Sql);
            while (rs.next()) { NumeroDeFichero = rs.getInt("count"); }

        }

        catch (Exception Ex)
        { System.out.println("ConsultasEstablecidas.java: Error al obtener número de
        secuencia en la base de datos");
            System.out.println(Ex.getMessage());
        }

        return NumeroDeFichero;
    }

    /**
     * Método que consulta las tablas "personal_authorized" y "credenciales" para
     * comprobación de usuarios y permisos en el Servidor de Registro de Acceso.
     *
     * Resumen de consultas.-
     * Eleccion. Tabla y Campos.
     *
     * 1. personal_authorized.id_usuario.
     * 2. personal_authorized.login, personal_authorized.password.
     * 3. personal_authorized.id_usuario, personal_authorized.codigo_credenciales.
     * 4. credenciales.codigo_credenciales, credenciales.recursos.
     *
     * @param Eleccion int con el el número de consulta a realizar.
     * @return ResultSet con la consulta realizada.
     */
    public static ResultSet UsuariosDelServidor (int Eleccion)
    {
        switch(Eleccion)
        {
            case 1: Sql = "SELECT PERSONAL_AUTORIZADO.id_usuario "
                + "FROM PERSONAL_AUTORIZADO;"; break;

            case 2: Sql = "SELECT PERSONAL_AUTORIZADO.login, PERSONAL_AUTORIZADO.password "
                + "FROM PERSONAL_AUTORIZADO;"; break;

            case 3: Sql = "SELECT PERSONAL_AUTORIZADO.id_usuario,
            PERSONAL_AUTORIZADO.codigo_credenciales "
                + "FROM PERSONAL_AUTORIZADO;"; break;

            case 4: Sql = "SELECT CREDENCIALES.codigo_credenciales, CREDENCIALES.Recursos "
                + "FROM CREDENCIALES;"; break;
        }

        try { conexion = Conexion.EstablecerConexion();
            rs = RegistroDeAcceso.ConsultarBaseDeDatos(conexion, Sql);
        }

        catch (Exception Ex)
        { System.out.println("ConsultasEstablecidas.java: Error al consultar la base de
        datos");
            System.out.println(Ex.getMessage());
        }

        return rs;
    }
}
```

```
/**
 * Método que devuelve un Vector con los nombres de los campos
 * consultados por la secuencia SQL pasada como parámetro.
 * Cada elemento del Vector se corresponde con el nombre de uno de los campos.
 * @param Sql String con la sentencia SQL de consulta realizada a la base de datos.
 * @return Vector con los nombres de los campos.
 */
public static Vector RetornarNombreCampos (String Sql)
{
    Vector NombresCampos = new Vector();

    try { conexion = Conexion.EstablecerConexion();
        rs = RegistroDeAcceso.ConsultarBaseDeDatos(conexion,Sql);
        if (rs==null) {}
        else { ResultSetMetaData rsmd = rs.getMetaData();
            for (int i=1; i<rsmd.getColumnCount()+1; i++)
            { String respuesta = rsmd.getColumnName(i);
                NombresCampos.addElement(respuesta);
            }
        }

    }

    catch (Exception Ex)
    { System.out.println("ConsultasEstablecidas.java: No se ha podido retornar los
nombres de los campos consultados");
        System.out.println(Ex.getMessage());
    }

    return NombresCampos;
}

/**
 * Método que convierte el resultado tipo ResultSet de las consultas a un Vector
 * cuyos elementos son parámetros de la clase ParametrosDeConsulta.java
 * @return Vector de ParametrosDeConsulta.java
 * @param rs ResultSet con el resultado de la consulta.
 */
public static Vector CrearVectorConsulta (ResultSet rs)
{
    Vector vectorConsulta = new Vector();

    try { ResultSetMetaData rsmd = rs.getMetaData();
        int Contador = rsmd.getColumnCount();

        if (Contador<2) { while (rs.next()) // Consultas de 1 Parámetro
            { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
                temporal.setParametro1(rs.getString(1));
                vectorConsulta.addElement(temporal);
            }
        }
        else if (Contador<3) { while (rs.next()) // Consultas de 2 Parámetros
            { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
                temporal.setParametro1(rs.getString(1));
                temporal.setParametro2(rs.getString(2));
                vectorConsulta.addElement(temporal);
            }
        }
        else if (Contador<4) { while (rs.next()) // Consultas de 3 Parámetros
            { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
                temporal.setParametro1(rs.getString(1));
                temporal.setParametro2(rs.getString(2));
                temporal.setParametro3(rs.getString(3));
                vectorConsulta.addElement(temporal);
            }
        }
        else if (Contador<5) { while (rs.next()) // Consultas de 4 Parámetros
            { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
                temporal.setParametro1(rs.getString(1));
                temporal.setParametro2(rs.getString(2));
                temporal.setParametro3(rs.getString(3));
                temporal.setParametro4(rs.getString(4));
            }
        }
    }
}
```

```
        vectorConsulta.addElement(temporal);
    }
    else if (Contador<6) { while (rs.next()) // Consultas de 5 Parámetros
    { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
        temporal.setParametro1(rs.getString(1));
        temporal.setParametro2(rs.getString(2));
        temporal.setParametro3(rs.getString(3));
        temporal.setParametro4(rs.getString(4));
        temporal.setParametro5(rs.getString(5));
        vectorConsulta.addElement(temporal);
    }
    }
    else if (Contador<7)
    { while (rs.next()) // Consultas de 6 Parámetros
    { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
        temporal.setParametro1(rs.getString(1));
        temporal.setParametro2(rs.getString(2));
        temporal.setParametro3(rs.getString(3));
        temporal.setParametro4(rs.getString(4));
        temporal.setParametro5(rs.getString(5));
        temporal.setParametro6(rs.getString(6));
        vectorConsulta.addElement(temporal);
    }
    }
    else if (Contador<8)
    { while (rs.next()) // Consultas de 7 Parámetros
    { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
        temporal.setParametro1(rs.getString(1));
        temporal.setParametro2(rs.getString(2));
        temporal.setParametro3(rs.getString(3));
        temporal.setParametro4(rs.getString(4));
        temporal.setParametro5(rs.getString(5));
        temporal.setParametro6(rs.getString(6));
        temporal.setParametro7(rs.getString(7));
        vectorConsulta.addElement(temporal);
    }
    }
    else if (Contador<9)
    { while (rs.next()) // Consultas de 8 Parámetros
    { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
        temporal.setParametro1(rs.getString(1));
        temporal.setParametro2(rs.getString(2));
        temporal.setParametro3(rs.getString(3));
        temporal.setParametro4(rs.getString(4));
        temporal.setParametro5(rs.getString(5));
        temporal.setParametro6(rs.getString(6));
        temporal.setParametro7(rs.getString(7));
        temporal.setParametro8(rs.getString(8));
        vectorConsulta.addElement(temporal);
    }
    }
    else if (Contador<10)
    { while (rs.next()) // Consultas de 9 Parámetros
    { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
        temporal.setParametro1(rs.getString(1));
        temporal.setParametro2(rs.getString(2));
        temporal.setParametro3(rs.getString(3));
        temporal.setParametro4(rs.getString(4));
        temporal.setParametro5(rs.getString(5));
        temporal.setParametro6(rs.getString(6));
        temporal.setParametro7(rs.getString(7));
        temporal.setParametro8(rs.getString(8));
        temporal.setParametro9(rs.getString(9));
        vectorConsulta.addElement(temporal);
    }
    }
```

```
        else if (Contador<11)
        { while (rs.next()) // Consultas de 10 Parámetros
          { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
              temporal.setParametro1(rs.getString(1));
              temporal.setParametro2(rs.getString(2));
              temporal.setParametro3(rs.getString(3));
              temporal.setParametro4(rs.getString(4));
              temporal.setParametro5(rs.getString(5));
              temporal.setParametro6(rs.getString(6));
              temporal.setParametro7(rs.getString(7));
              temporal.setParametro8(rs.getString(8));
              temporal.setParametro9(rs.getString(9));
              temporal.setParametro10(rs.getString(10));
              vectorConsulta.addElement(temporal);
          }
        }

        else if (Contador<12)
        { while (rs.next()) // Consultas de 11 Parámetros
          { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
              temporal.setParametro1(rs.getString(1));
              temporal.setParametro2(rs.getString(2));
              temporal.setParametro3(rs.getString(3));
              temporal.setParametro4(rs.getString(4));
              temporal.setParametro5(rs.getString(5));
              temporal.setParametro6(rs.getString(6));
              temporal.setParametro7(rs.getString(7));
              temporal.setParametro8(rs.getString(8));
              temporal.setParametro9(rs.getString(9));
              temporal.setParametro10(rs.getString(10));
              temporal.setParametro11(rs.getString(11));
              vectorConsulta.addElement(temporal);
          }
        }

        else if (Contador<13)
        { while (rs.next()) // Consultas de 12 Parámetros
          { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
              temporal.setParametro1(rs.getString(1));
              temporal.setParametro2(rs.getString(2));
              temporal.setParametro3(rs.getString(3));
              temporal.setParametro4(rs.getString(4));
              temporal.setParametro5(rs.getString(5));
              temporal.setParametro6(rs.getString(6));
              temporal.setParametro7(rs.getString(7));
              temporal.setParametro8(rs.getString(8));
              temporal.setParametro9(rs.getString(9));
              temporal.setParametro10(rs.getString(10));
              temporal.setParametro11(rs.getString(11));
              temporal.setParametro12(rs.getString(12));
              vectorConsulta.addElement(temporal);
          }
        }

        else if (Contador<14)
        { while (rs.next()) // Consultas de 13 Parámetros
          { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
              temporal.setParametro1(rs.getString(1));
              temporal.setParametro2(rs.getString(2));
              temporal.setParametro3(rs.getString(3));
              temporal.setParametro4(rs.getString(4));
              temporal.setParametro5(rs.getString(5));
              temporal.setParametro6(rs.getString(6));
              temporal.setParametro7(rs.getString(7));
              temporal.setParametro8(rs.getString(8));
              temporal.setParametro9(rs.getString(9));
              temporal.setParametro10(rs.getString(10));
              temporal.setParametro11(rs.getString(11));
              temporal.setParametro12(rs.getString(12));
              temporal.setParametro13(rs.getString(13));
              temporal.setParametro14(rs.getString(14));
              temporal.setParametro15(rs.getString(15));
              temporal.setParametro16(rs.getString(16));
```

```
        temporal.setParametro17(rs.getString(17));
        temporal.setParametro18(rs.getString(18));
        vectorConsulta.addElement(temporal);
    }
}
else if (Contador<15)
{ while (rs.next()) // Consultas de 14 Parámetros
{ ParametrosDeConsulta temporal = new
ParametrosDeConsulta();

        temporal.setParametro1(rs.getString(1));
        temporal.setParametro2(rs.getString(2));
        temporal.setParametro3(rs.getString(3));
        temporal.setParametro4(rs.getString(4));
        temporal.setParametro5(rs.getString(5));
        temporal.setParametro6(rs.getString(6));
        temporal.setParametro7(rs.getString(7));
        temporal.setParametro8(rs.getString(8));
        temporal.setParametro9(rs.getString(9));
        temporal.setParametro10(rs.getString(10));
        temporal.setParametro11(rs.getString(11));
        temporal.setParametro12(rs.getString(12));
        temporal.setParametro13(rs.getString(13));
        temporal.setParametro14(rs.getString(14));
        vectorConsulta.addElement(temporal);
    }
}
else if (Contador<16)
{ while (rs.next()) // Consultas de 15 Parámetros
{ ParametrosDeConsulta temporal = new
ParametrosDeConsulta();

        temporal.setParametro1(rs.getString(1));
        temporal.setParametro2(rs.getString(2));
        temporal.setParametro3(rs.getString(3));
        temporal.setParametro4(rs.getString(4));
        temporal.setParametro5(rs.getString(5));
        temporal.setParametro6(rs.getString(6));
        temporal.setParametro7(rs.getString(7));
        temporal.setParametro8(rs.getString(8));
        temporal.setParametro9(rs.getString(9));
        temporal.setParametro10(rs.getString(10));
        temporal.setParametro11(rs.getString(11));
        temporal.setParametro12(rs.getString(12));
        temporal.setParametro13(rs.getString(13));
        temporal.setParametro14(rs.getString(14));
        temporal.setParametro15(rs.getString(15));
        vectorConsulta.addElement(temporal);
    }
}
else if (Contador<17)
{ while (rs.next()) // Consultas de 16 Parámetros
{ ParametrosDeConsulta temporal = new
ParametrosDeConsulta();

        temporal.setParametro1(rs.getString(1));
        temporal.setParametro2(rs.getString(2));
        temporal.setParametro3(rs.getString(3));
        temporal.setParametro4(rs.getString(4));
        temporal.setParametro5(rs.getString(5));
        temporal.setParametro6(rs.getString(6));
        temporal.setParametro7(rs.getString(7));
        temporal.setParametro8(rs.getString(8));
        temporal.setParametro9(rs.getString(9));
        temporal.setParametro10(rs.getString(10));
        temporal.setParametro11(rs.getString(11));
        temporal.setParametro12(rs.getString(12));
        temporal.setParametro13(rs.getString(13));
        temporal.setParametro14(rs.getString(14));
        temporal.setParametro15(rs.getString(15));
        temporal.setParametro16(rs.getString(16));
        vectorConsulta.addElement(temporal);
    }
}
else if (Contador<18)
{ while (rs.next()) // Consultas de 17 Parámetros
{ ParametrosDeConsulta temporal = new
ParametrosDeConsulta();

        temporal.setParametro1(rs.getString(1));
```

```
        temporal.setParametro2(rs.getString(2));
        temporal.setParametro3(rs.getString(3));
        temporal.setParametro4(rs.getString(4));
        temporal.setParametro5(rs.getString(5));
        temporal.setParametro6(rs.getString(6));
        temporal.setParametro7(rs.getString(7));
        temporal.setParametro8(rs.getString(8));
        temporal.setParametro9(rs.getString(9));
        temporal.setParametro10(rs.getString(10));
        temporal.setParametro11(rs.getString(11));
        temporal.setParametro12(rs.getString(12));
        temporal.setParametro13(rs.getString(13));
        temporal.setParametro14(rs.getString(14));
        temporal.setParametro15(rs.getString(15));
        temporal.setParametro16(rs.getString(16));
        temporal.setParametro17(rs.getString(17));
        vectorConsulta.addElement(temporal);
    }
    else if (Contador<19)
    { while (rs.next()) // Consultas de 18 Parámetros
      { ParametrosDeConsulta temporal = new
ParametrosDeConsulta();
        temporal.setParametro1(rs.getString(1));
        temporal.setParametro2(rs.getString(2));
        temporal.setParametro3(rs.getString(3));
        temporal.setParametro4(rs.getString(4));
        temporal.setParametro5(rs.getString(5));
        temporal.setParametro6(rs.getString(6));
        temporal.setParametro7(rs.getString(7));
        temporal.setParametro8(rs.getString(8));
        temporal.setParametro9(rs.getString(9));
        temporal.setParametro10(rs.getString(10));
        temporal.setParametro11(rs.getString(11));
        temporal.setParametro12(rs.getString(12));
        temporal.setParametro13(rs.getString(13));
        temporal.setParametro14(rs.getString(14));
        temporal.setParametro15(rs.getString(15));
        temporal.setParametro16(rs.getString(16));
        temporal.setParametro17(rs.getString(17));
        temporal.setParametro18(rs.getString(18));
        vectorConsulta.addElement(temporal);
      }
    }

    catch (Exception Ex)
    { System.out.println("ConsultasEstablecidas.java: No se creado un vector a
partir del ResultSet");
      System.out.println(Ex.getMessage());
    }

    return vectorConsulta;
  }
}
```

9.5. Clase ConsultasPersonalizadas.java.-

```

/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ConsultasPersonalizadas.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

import ardb.*;

import java.util.*;
import java.sql.*;

/**
 * Clase utilizada por el Servidor de Registro de Acceso para obtener
 * información de la base de datos "Registro de Acceso" con consultas personalizadas
 * por el programador.
 * En esta clase se recogen métodos que permiten personalizar las consultas
 * a la base de datos "Registro de Acceso". La forma de crear una nueva consulta se
 * basa en la formación dinámica de sentencias SQL. El formato SQL de consulta
 * permitido responde al código SQL:
 * "SELECT (tablas.campos) FROM (tablas) WHERE (condiciones SQL) ORDER BY (campos);".
 */
public class ConsultasPersonalizadas {

    /**
     * Objeto tipo Connection con la conexión a la base de datos "Registro de Acceso".
     */
    private static Connection conexion = null;

    /**
     * Objeto tipo ResultSet con el resultado de la consulta a la base de datos.
     */
    private static ResultSet rs = null;

    /**
     * Objeto tipo DatabaseMetaData que obtiene las características de la base de datos.
     */
    private static DatabaseMetaData dmd = null;

    /**
     * Objeto tipo ResultSetMetaData que obtiene las características del "rs".
     */
    private static ResultSetMetaData rsmd = null;

    /**
     * Objeto tipo String con la sentencia SQL a ejecutar.
     */
    private static String Sql = null;

    /**
     * Objeto tipo String con el nombre de la tabla a utilizar.
     */
    private static String Tabla = null;

    /**
     * Objeto tipo String con el nombre del campo de la tabla a utilizar.
     */
    private static String Campo = null;

    /**
     * Objeto tipo String con la condición SQL que debe cumplir la consulta.
     */
    private static String Condicion = null;

    /**
     * Objeto tipo String con la descripción del orden que debe cumplir la consulta.
     */
    private static String Orden = null;

    /**
     * Objeto tipo Vector con los campos seleccionados para realizar la consulta.

```

```

    * El formato de cada elemento del Vector es TABLA.campo.
    */
    public static Vector CamposSeleccionados = new Vector();
    /**
    * Objeto tipo Vector con las tablas seleccionadas para realizar la consulta.
    */
    public static Vector TablasSeleccionadas = new Vector();
    /**
    * Objeto tipo Vector con las condiciones impuestas por el usuario a la consulta.
    * Esta condición debe estar escrita en formato SQL.
    */
    public static Vector CondicionesSeleccionadas = new Vector();
    /**
    * Objeto tipo Vector con los campos que marcan el orden.
    */
    public static Vector OrdenSeleccionadas = new Vector();
    /**
    * Objeto tipo Vector que almacena el nombre de las nuevas consultas.
    */
    public static Vector NombreConsultasAlmacenadas = new Vector();
    /**
    * Objeto tipo Vector que almacena las sentencias SQL pertenecientes a las nuevas
    consultas.
    */
    public static Vector ConsultasAlmacenadas = new Vector();
    /**
    * Objeto tipo Vector que almacena el número correspondiente a la consulta
    almacenada.
    */
    public static Vector NumeroEleccionConsultasAlmacenadas = new Vector();
    /**
    * Campo que indica el número de consulta a registrar. Para seguir con
    * la numeración aplicada a los recursos del servidor, las consultas
    * personalizadas recibirán los números desde el 40 al 199
    */
    static int EleccionConsulta = 40;

    /**
    * Método que devuelve un Vector con los nombres de los campos de la tabla
    * seleccionada de la base de datos "Registro de Acceso".
    * Cada una de las 15 tablas que componen la base de datos "Registro de Acceso"
    * está referenciada por un entero.
    *
    * Resumen de opciones.-
    * Se señalan las posibles opciones del siguiente modo:
    * Eleccion. Nombre de la tabla.
    *
    * 1. Tabla comunicacion.
    * 2. Tabla credenciales.
    * 3. Tabla forma_de_envio.
    * 4. Tabla incidencias.
    * 5. Tabla motivo_de_denegacion.
    * 6. Tabla personal_autorizado.
    * 7. Tabla situacion_acceso.
    * 8. Tabla tipo_de_soporte.
    * 9. Tabla registro_de_acceso.
    * 10. Tabla registro_de_entrada_de_soportes_informaticos.
    * 11. Tabla registro_de_incidencias.
    * 12. Tabla registro_de_recuperacion_de_datos.
    * 13. Tabla registro_de_salida_de_soportes_informaticos.
    * 14. Tabla registro_de_respaldo.
    * 15. Tabla recursos.
    *
    * @param Eleccion int que selecciona una de las 15 tablas de la base de datos.
    * @return Vector con el nombre de los campos de la tabla seleccionada.
    */
    public static Vector MostrarNombresCampos (int Eleccion)
    {
        switch (Eleccion)
        {
            case 1: Sql = "SELECT * FROM COMUNICACION ORDER BY codigo_comunicacion"; break;
            case 2: Sql = "SELECT * FROM CREDENCIALES ORDER BY codigo_credenciales"; break;
            case 3: Sql = "SELECT * FROM FORMA_DE_ENVIO ORDER BY codigo_envio"; break;
            case 4: Sql = "SELECT * FROM INCIDENCIAS ORDER BY codigo_incidencia"; break;
            case 5: Sql = "SELECT * FROM MOTIVO_DE_DENEGACION ORDER BY codigo_denegacion";
        }
        break;
    }

```

```

        case 6: Sql = "SELECT * FROM PERSONAL_AUTORIZADO ORDER BY id_usuario"; break;
        case 7: Sql = "SELECT * FROM SITUACION_ACCESO ORDER BY codigo_acceso"; break;
        case 8: Sql = "SELECT * FROM TIPO_DE_SOPORTE ORDER BY codigo_soporte"; break;
        case 9: Sql = "SELECT * FROM REGISTRO_DE_ACCESOS"; break;
        case 10: Sql = "SELECT * FROM REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS";
break;
        case 11: Sql = "SELECT * FROM REGISTRO_DE_INCIDENCIAS"; break;
        case 12: Sql = "SELECT * FROM REGISTRO_DE_RECUPERACION_DE_DATOS"; break;
        case 13: Sql = "SELECT * FROM REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS";
break;
        case 14: Sql = "SELECT * FROM REGISTRO_DE_RESPALDO";break;
        case 15: Sql = "SELECT * FROM RECURSOS";break;
    }

    Vector vectorConsulta = new Vector();

    try { conexion = Conexion.EstablecerConexion();
        rs = RegistroDeAcceso.ConsultarBaseDeDatos(conexion,Sql);
        rsmd = rs.getMetaData();

        for (int i=1; i<rsmd.getColumnCount()+1; i++)
        { String respuesta = rsmd.getColumnName(i);
            vectorConsulta.addElement(respuesta);
        }
    }

    catch (Exception Ex)
    { System.out.println("No se ha podido establecer la conexión con la base de
datos");
        System.out.println(Ex.getMessage());
    }

    return vectorConsulta;
}

/**
 * Método que devuelve un objeto tipo String con el nombre de la tabla
 * de la base de datos "Registro de Acceso" seleccionada con Eleccion.
 * @param Eleccion int que selecciona una de las 15 tablas de la base de
 * datos "Registro de Acceso".
 * @return String con el nombre de la tabla seleccionada.
 */
public static String MostrarNombreTabla (int Eleccion)
{
    switch (Eleccion)
    {
        case 1: Tabla = "COMUNICACION"; break;
        case 2: Tabla = "CREDENCIALES"; break;
        case 3: Tabla = "FORMA_DE_ENVIO"; break;
        case 4: Tabla = "INCIDENCIAS"; break;
        case 5: Tabla = "MOTIVO_DE_DENEGACION"; break;
        case 6: Tabla = "PERSONAL_AUTORIZADO"; break;
        case 7: Tabla = "SITUACION_ACCESO"; break;
        case 8: Tabla = "TIPO_DE_SOPORTE"; break;
        case 9: Tabla = "REGISTRO_DE_ACCESOS"; break;
        case 10: Tabla = "REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS"; break;
        case 11: Tabla = "REGISTRO_DE_INCIDENCIAS"; break;
        case 12: Tabla = "REGISTRO_DE_RECUPERACION_DE_DATOS"; break;
        case 13: Tabla = "REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS"; break;
        case 14: Tabla = "REGISTRO_DE_RESPALDO";break;
        case 15: Tabla = "RECURSOS";break;
    }

    return Tabla;
}

/**
 * Método que inserta en los Vectores CamposSeleccionados y
 * TablasSeleccionadas el campo seleccionado.
 */
public static void InsertarCampo ()
{
    String Tabla_Campo = getTabla()+"."+getCampo();
    CamposSeleccionados.addElement(Tabla_Campo);
    TablasSeleccionadas.addElement(getTabla());
}

```

```

}

/**
 * Método que inserta en el Vector CondicionesSeleccionadas la condicion SQL
 * configurada por el usuario.
 */
public static void InsertarCondicion ()
{
    Condicion = getCondicion();
    CondicionesSeleccionadas.addElement(Condicion);
}

/**
 * Método que inserta en el vector OrdenSeleccionadas los campos por los que
 * se quiere ordenar la consulta.
 */
public static void InsertarOrden ()
{
    String Orden = getOrden();
    OrdenSeleccionadas.addElement(Orden);
}

/**
 * Método que borra los Vectores CamposSeleccionados y TablasSeleccionadas.
 */
public static void BorrarCampos ()
{
    CamposSeleccionados.removeAllElements();
    TablasSeleccionadas.removeAllElements();
}

/**
 * Método que borra el último campo introducido.
 */
public static void BorrarUltimoCampo ()
{
    if (CamposSeleccionados.size()==0) {}
    else { CamposSeleccionados.removeElementAt(CamposSeleccionados.size()-1);
          TablasSeleccionadas.removeElementAt(TablasSeleccionadas.size()-1);
    }
}

/**
 * Método que borra el Vector CondicionesSeleccionadas.
 */
public static void BorrarCondiciones ()
{
    CondicionesSeleccionadas.removeAllElements(); }

/**
 * Método que borra la última condición introducida.
 */
public static void BorrarUltimaCondicion ()
{
    if (CondicionesSeleccionadas.size()==0) {}
    else { CondicionesSeleccionadas.removeElementAt(CondicionesSeleccionadas.size()-
1);
    }
}

/**
 * Método que borra el Vector OrdenSeleccionadas.
 */
public static void BorrarOrden ()
{
    OrdenSeleccionadas.removeAllElements(); }

/**
 * Método que borra el último campo seleccionado para ordenar la consulta.
 */
public static void BorrarUltimoOrden ()
{
    if (OrdenSeleccionadas.size()==0) {}
    else { OrdenSeleccionadas.removeElementAt(OrdenSeleccionadas.size()-1);
    }
}

/**
 * Método que borra todos los Vectores relativos a la creación de una nueva

```

```

        * consulta.
        */
public static void BorrarCamposCondicionesOrden()
{
    BorrarCampos();
    BorrarCondiciones();
    BorrarOrden();
}

/**
 * Método que guarda el nombre de la tabla seleccionada.
 * @param AuxTabla String con el nombre de la tabla seleccionada.
 */
public static void setTabla (String AuxTabla) { Tabla = AuxTabla; }
/**
 * Método que guarda el nombre del campo seleccionado.
 * @param AuxCampo String con el nombre del campo seleccionado.
 */
public static void setCampo (String AuxCampo) { Campo = AuxCampo; }
/**
 * Método que almacena la condición SQL seleccionada.
 * @param AuxCondicion String con la condición SQL seleccionada.
 */
public static void setCondicion (String AuxCondicion) { Condicion = AuxCondicion; }
/**
 * Método que guarda el nombre del campo por el cual se pretende ordenar la
 * consulta.
 * @param AuxOrden String con el nombre del campo por el cual se va a ordenar la
 * consulta.
 */
public static void setOrden (String AuxOrden) { Orden = AuxOrden; }
/**
 * Devuelve Tabla.
 * @return String con el nombre de la tabla.
 */
public static String getTabla () { return Tabla; }
/**
 * Devuelve Campo
 * @return String con el nombre del campo.
 */
public static String getCampo () { return Campo; }
/**
 * Devuelve Condición.
 * @return String con la condición SQL.
 */
public static String getCondicion () { return Condicion; }
/**
 * Devuelve Orden.
 * @return String con el nombre del campo.
 */
public static String getOrden () { return Orden; }
/** Borra las variables Campo y Tabla. */
public static void ClearCampo() { Tabla = null; Campo = null; }
/** Borra la variable Condicion. */
public static void ClearCondicion() { Condicion = null; }
/** Borra la variable Orden. */
public static void ClearOrden() { Orden = null; }

/**
 * Método que crea la sentencia SQL de consulta con información introducida en
 * CamposSeleccionados, TablasSeleccionadas, CondicionesSeleccionadas y
 * OrdenSeleccionadas. Recibe el nombre por el cual se va a conocer la consulta y
 * se almacena en los vectores NombreConsultasAlmacenadas, ConsultasAlmacenadas y
 * NumeroEleccionConsultasAlmacenadas el nombre de la consulta, su sentencia SQL
 * correspondiente y el número de consulta que posee.
 * Retorna un mensaje indicando si se ha creado la consulta o si ha habido
 * alguna incidencia.
 * @param NombreConsulta String con el nombre asignado a la consulta.
 * @return String con mensaje de confirmación o de negación.
 */
public static String CrearConsultaSQL (String NombreConsulta)
{
    Vector vectorConsulta = new Vector();
    String Respuesta = "No se creado la consulta";

```

```

        if (CamposSeleccionados.size()==0)
        { Respuesta = "No se han seleccionado campos para realizar la consulta"; }
    else {
        String Campos = "";
        Campos = Campos+(String)CamposSeleccionados.get(0);
        for (int i=1; i<CamposSeleccionados.size();i++)
            { Campos = Campos+", "+(String)CamposSeleccionados.get(i); }

        String Tablas = "";
        String Tabla = (String)TablasSeleccionadas.get(0);
        if (Tablas.contains(Tabla)) {}
        else { Tablas = Tablas+Tabla; }
        for (int i=1; i<TablasSeleccionadas.size();i++)
            { Tabla = (String)TablasSeleccionadas.get(i);
              if (Tablas.contains(Tabla)) {}
              else { Tablas =Tablas+", "+Tabla; }
            }

        Sql = "SELECT "+Campos+" FROM "+Tablas+"";

        if (CondicionesSeleccionadas.size()==0) {}
        else {
            String Donde = "";
            String Condicion = (String)CondicionesSeleccionadas.get(0);
            Donde = Donde+Condicion;
            for (int i=1; i<CondicionesSeleccionadas.size();i++)
                { Condicion = (String)CondicionesSeleccionadas.get(i);
                  Donde = Donde+" AND "+Condicion;
                }
            Sql = Sql+" WHERE "+Donde+"";
        }

        if (OrdenSeleccionadas.size()==0) {}
        else {
            String Orden = "";
            String Ordenado = (String)OrdenSeleccionadas.get(0);
            Orden = Orden + Ordenado;
            for (int i=1; i<OrdenSeleccionadas.size();i++)
                { Ordenado = (String)OrdenSeleccionadas.get(i);
                  Orden = Orden+", "+Ordenado;
                }

            Sql =Sql+" ORDER BY "+Orden+"";
        }

        Sql=Sql+"";

        InsertarConsultaAlmacenada(NombreConsulta,Sql);
        Respuesta = "La consulta '"+NombreConsulta+"' ha sido almacenada";
    }

    return Respuesta;
}

/**
 * Método que recibe como argumento una sentencia SQL de consulta, guarda los
 * datos necesarios para el informe PDF y retorna el resultado asociado,
 * en un Vector cuyos elementos son parámetros de la clase
 * ParametrosDeConsulta.java.
 * @param SQL String con la sentencia SQL.
 * @return Vector de parámetros de la clase ParametrosDeConsulta.java
 */
public static Vector MostrarConsultaSQL (String SQL)
{
    Vector vectorConsulta = new Vector();
    Sql = SQL;

    // Preparamos los parámetros para generar el informe PDF.
    ParametrosPDF.setSql(Sql);
    ParametrosPDF.setNombresCampos(ConsultasEstablecidas.RetornarNombreCampos(Sql));

    try { conexion = Conexion.EstablecerConexion();
        rs = RegistroDeAcceso.ConsultarBaseDeDatos(conexion, Sql);
        if (rs==null) { vectorConsulta = null; }
        else { vectorConsulta = ConsultasEstablecidas.CrearVectorConsulta(rs); }
    }

```

```
    }
    catch (Exception Ex)
    { System.out.println("No se ha podido establecer la conexión con la base de
datos");
      System.out.println(Ex.getMessage());
    }

    return vectorConsulta;
  }

/**
 * Inserta en los vectores NombreConsultasAlmacenadas, ConsultasAlmacenadas y
 * NumeroEleccionConsultasAlmacenadas el nombre, la sentencia SQL y el número
 * asociado a la nueva consulta.
 * @param NombreConsulta String con el nombre de la nueva consulta.
 * @param Sql String con la sentencia SQL de la nueva consulta.
 */
public static void InsertarConsultaAlmacenada (String NombreConsulta, String Sql)
{
    NombreConsultasAlmacenadas.addElement(NombreConsulta);
    ConsultasAlmacenadas.addElement(Sql);
    NumeroEleccionConsultasAlmacenadas.addElement(EleccionConsulta);
    EleccionConsulta++;
}

/**
 * Método que devuelve la sentencia SQL que corresponde al nombre de la consulta
 * pasada como argumento.
 * @param NombreConsulta String con el nombre de la consulta.
 * @return String con la sentencia SQL.
 */
public static String DevolverConsultaAlmacenada (String NombreConsulta)
{
    int index = 0;
    for (int i=0; i<NombreConsultasAlmacenadas.size();i++)
    { String Auxiliar = (String) NombreConsultasAlmacenadas.get(i);
      if (Auxiliar.contains(NombreConsulta)) { index =i; }
    }
    String SQL = (String) ConsultasAlmacenadas.get(index);

    return SQL;
}

/**
 * Método que devuelve el nombre de la consulta cuya sentencia SQL es pasada
 * como argumento.
 * @param SQL String con la sentencia SQL.
 * @return String con el nombre de la consulta.
 */
public static String DevolverNombreConsultaAlmacenada (String SQL)
{
    int index = 0;
    for (int i=0; i<ConsultasAlmacenadas.size();i++)
    { String Auxiliar = (String) ConsultasAlmacenadas.get(i);
      if (Auxiliar.contains(SQL)) { index =i; }
    }
    String NombreConsulta = (String) NombreConsultasAlmacenadas.get(index);

    return NombreConsulta;
}

/**
 * Método que devuelve el número asociado a la consulta cuya sentencia SQL
 * es pasada como argumento.
 * @param SQL String con la sentencia SQL de la consulta.
 * @return int con el número que describe el recurso(la consulta) en el servidor.
 */
public static int DevolverNumeroEleccion (String SQL)
{
    int index = 0;
    for (int i=0; i<ConsultasAlmacenadas.size();i++)
    { String Auxiliar = (String) ConsultasAlmacenadas.get(i);
      if (Auxiliar.contains(SQL))
      { index = i; }
    }
}
```

```
        int Eleccion =
Convertir.String_Int(NumeroEleccionConsultasAlmacenadas.get(index).toString());

        return Eleccion;
    }

    /**
     * Borra los Vectores NombreConsultasAlmacenadas, ConsultasAlmacenadas y
     * NumeroEleccionConsultasAlmacenadas.
     * Además pone de nuevo la variable EleccionConsulta a 40.
     */
    public static void BorrarConsultasAlmacenadas ()
    {
        NombreConsultasAlmacenadas.removeAllElements();
        ConsultasAlmacenadas.removeAllElements();
        NumeroEleccionConsultasAlmacenadas.removeAllElements();
        EleccionConsulta = 40;
    }

    /**
     * Borra la consulta cuya sentencia SQL es pasada como argumento.
     * @param SQL String con la sentencia SQL de la consulta.
     */
    public static void BorrarLaConsultaX (String SQL)
    {
        int index = -1;
        for (int i=0; i<ConsultasAlmacenadas.size();i++)
        { String Auxiliar = (String) ConsultasAlmacenadas.get(i);
          if (Auxiliar.contains(SQL))
          { index =i;
            NombreConsultasAlmacenadas.removeElementAt(index);
            ConsultasAlmacenadas.removeElementAt(index);
            NumeroEleccionConsultasAlmacenadas.removeElementAt(index);
          }
        }
    }
}

}
```

9.6. Clase ControlAcceso.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ControlAcceso.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

/**
 * Clase que almacena información sobre el acceso del usuario al Sistema de Información.
 */
public class ControlAcceso {

    /**
     * Identificación del usuario.
     */
    static String IdUsuario = null;
    /** Identificación del ordenador utilizado por el usuario para acceder al sistema. */
    static String IdHost = null;
    /** Credenciales del usuario al acceder al sistema. */
    static String Credenciales = null;
    /** Identificación del recurso al que accede el usuario. */
    static String IdRecurso = null;
    /**
     * Campo que recoge si el acceso del usuario al recurso fue autorizado.
     */
    static boolean Autorizado = false;
    /** Fichero accedido por el usuario. */
    static String FicheroAccedido = null;
    /** Propiedades del fichero accedido por el usuario. */
    static String PropiedadesFicheroAccedido = null;
    /** URL del fichero accedido por el usuario. */
    static String URL = null;
    /**
     * Campo que señala si existe una copia de respaldo para el fichero accedido.
     */
    static boolean CopiaRespaldo = false;
    /**
     * Campo que recoge el código por el cual no se pudo acceder al fichero.
     */
    static StringCodigoDenegado = null;
    /** Recoge el tipo de acceso del usuario al fichero. */
    static String TipoAcceso = null;
    /** El código que recoge en que situación accedió el usuario al fichero. */
    static StringCodigoAcceso = null;
    /**
     * Campo que recoge si se despersonalizó el fichero antes del acceso del usuario.
     */
    static boolean Despersonalizacion = false;

    /**
     * Almacena IdUsuario.
     * @param AuxIdUsuario String IdUsuario.
     */
    public static void setIdUsuario(String AuxIdUsuario)
    { IdUsuario = AuxIdUsuario; }
    /**
     * Devuelve IdUsuario.
     * @return String IdUsuario.
     */
    public static String getIdUsuario()
    { return IdUsuario; }
}
/**
```

```
*
* Almacena IdHost.
* @param AuxIdHost String IdHost.
*/
public static void setIdHost(String AuxIdHost)
{ IdHost = AuxIdHost; }
/**
* Devuelve IdHost.
* @return String IdHost. */
public static String getIdHost()
{ return IdHost; }

/**
*
* Almacena Credenciales.
* @param AuxCredenciales String Credenciales.
*/
public static void setCredenciales(String AuxCredenciales)
{ Credenciales = AuxCredenciales; }
/**
* Devuelve Credenciales.
* @return String Credenciales.
*/
public static String getCredenciales()
{ return Credenciales; }

/**
* Almacena IdRecurso.
* @param AuxIdRecurso String IdRecurso
*/
public static void setIdRecurso(String AuxIdRecurso)
{ IdRecurso = AuxIdRecurso; }
/**
* Devuelve IdRecurso,
* @return String IdRecurso.
*/
public static String getIdRecurso()
{ return IdRecurso; }

/**
* Almacena Autorizado.
* @param AuxAutorizado boolean Autorizado.
*/
public static void setAutorizado(boolean AuxAutorizado)
{ Autorizado = AuxAutorizado; }
/**
* Devuelve Autorizado
* @return boolean Autorizado.
*/
public static boolean getAutorizado()
{ return Autorizado; }

/**
* Almacena FicheroAccedido.
* @param AuxFicheroAccedido String FicheroAccedido.
*/
public static void setFicheroAccedido(String AuxFicheroAccedido)
{ FicheroAccedido = AuxFicheroAccedido; }
/**
* Devuelve FicheroAccedido.
* @return String FicheroAccedido.
*/
public static String getFicheroAccedido()
{ return FicheroAccedido; }

/**
* Almacena PropiedadesFicheroAccedido
* @param AuxPropiedadesFicheroAccedido String PropiedadesFicheroAccedido
*/
public static void setPropiedadesFicheroAccedido(String
AuxPropiedadesFicheroAccedido)
```

```
{ PropiedadesFicheroAccedido = AuxPropiedadesFicheroAccedido; }
/**
 * Devuelve PropiedadesFicheroAccedido
 * @return String PropiedadesFicheroAccedido
 */
public static String getPropiedadesFicheroAccedido()
{ return PropiedadesFicheroAccedido; }

/**
 * Almacena URL
 * @param AuxURL String URL
 */
public static void setURL(String AuxURL)
{ URL = AuxURL; }
/**
 * Devuelve
 * @return String URL
 */
public static String getURL()
{ return URL; }

/**
 * Almacena CopiaRespaldo.
 * @param AuxCopiaRespaldo boolean CopiaRespaldo.
 */
public static void setCopiaRespaldo(boolean AuxCopiaRespaldo)
{ CopiaRespaldo = AuxCopiaRespaldo; }
/**
 * Devuelve CopiaRespaldo.
 * @return boolean CopiaRespaldo.
 */
public static boolean getCopiaRespaldo()
{ return CopiaRespaldo; }

/**
 * Almacena CódigoDenegado.
 * @param AuxCodigoDenegado String CódigoDenegado.
 */
public static void setCodigoDenegado(String AuxCodigoDenegado)
{ CodigoDenegado = AuxCodigoDenegado; }
/**
 * Devuelve CódigoDenegado.
 * @return String CódigoDenegado.
 */
public static String getCodigoDenegado()
{ return CodigoDenegado; }

/**
 * Almacena TipoAcceso.
 * @param AuxTipoAcceso String TipoAcceso.
 */
public static void setTipoAcceso(String AuxTipoAcceso)
{ TipoAcceso = AuxTipoAcceso; }
/**
 * Devuelve TipoAcceso.
 * @return String TipoAcceso.
 */
public static String getTipoAcceso()
{ return TipoAcceso; }

/**
 * Almacena CódigoAcceso.
 * @param AuxCodigoAcceso String CódigoAcceso.
 */
public static void setCodigoAcceso(String AuxCodigoAcceso)
{ CodigoAcceso = AuxCodigoAcceso; }
/**
 * Devuelve CódigoAcceso.
 * @return String CódigoAcceso.
 */
public static String getCodigoAcceso()
```

```
{ returnCodigoAcceso; }

/**
 * Almacena Despersonalizacion.
 * @param AuxDespersonalizacion String Despersonalizacion.
 */
public static void setDespersonalizacion(boolean AuxDespersonalizacion)
{ Despersonalizacion = AuxDespersonalizacion; }
/**
 * Devuelve Despersonalizacion.
 * @return String Despersonalizacion.
 */
public static boolean getDespersonalizacion()
{ return Despersonalizacion; }

/**
 * Borra todos los parámetros del control de acceso.
 */
public static void Clear()
{ IdUsuario = null;
  Credenciales = null;
  IdRecurso = null;
  Autorizado = false;
  FicheroAccedido = null;
  PropiedadesFicheroAccedido = null;
  URL = null;
  CopiaRespaldo = false;
  CodigoDenegado = null;
  TipoAcceso = null;
  CodigoAcceso = null;
  Despersonalizacion = false;
}
}
```

9.7. Clase ControlCopiaRespaldo.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ControlCopiaRespaldo.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

/**
 * Clase que almacena información relacionada con la existencia de ficheros
 * con copia de respaldo en los recursos presentes en el Sistema de Información.
 */
public class ControlCopiaRespaldo {

    /**
     * Fecha de la copia de respaldo del fichero accedido en formato mm/dd/aaaa.
     */
    static String FechaCopiaRespaldo = null;

    /**
     * Almacena FechaCopiaRespaldo.
     * @param AuxFechaCopiaRespaldo String FechaCopiaRespaldo en formato mm/dd/aaaa.
     */
    public static void setFechaCopiaRespaldo(String AuxFechaCopiaRespaldo)
    { FechaCopiaRespaldo = AuxFechaCopiaRespaldo; }

    /**
     * Devuelve FechaCopiaRespaldo.
     * @return String FechaCopiaRespaldo en formato mm/dd/aaaa.
     */
    public static String getFechaCopiaRespaldo()
    { return FechaCopiaRespaldo; }

    /**
     * Borra todos los parámetros del control de copia de respaldo.
     */
    public static void Clear()
    { FechaCopiaRespaldo = null; }

}
```

9.8. Clase ControlIncidencia.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ControlIncidencia.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

/**
 * Clase que almacena información sobre las incidencias de seguridad que pueden
 * darse en el acceso del usuario al Sistema de Información.
 */
public class ControlIncidencia {

    /**
     * Código de la incidencia de seguridad detectada en el acceso.
     */
    static StringCodigoIncidencia = null;
    /**
     * Código de la comunicacion de la incidencia.
     */
    static StringCodigoComunicacion = null;
    /**
     * Persona o sistema que notifica la incidencia.
     */
    static StringNotificante = null;

    /**
     * Almacena CodigoIncidencia.
     * @param AuxCodigoIncidencia String CodigoIncidencia.
     */
    public static void setCodigoIncidencia(String AuxCodigoIncidencia)
    { CodigoIncidencia = AuxCodigoIncidencia; }
    /**
     * Devuelve CodigoIncidencia.
     * @return String CodigoIncidencia.
     */
    public static String getCodigoIncidencia()
    { return CodigoIncidencia; }

    /**
     * Almacena CodigoComunicacion.
     * @param AuxCodigoComunicacion String CodigoComunicacion.
     */
    public static void setCodigoComunicacion(String AuxCodigoComunicacion)
    { CodigoComunicacion = AuxCodigoComunicacion; }
    /**
     * Devuelve CodigoComunicacion.
     * @return String CodigoComunicacion.
     */
    public static String getCodigoComunicacion()
    { return CodigoComunicacion; }

    /**
     * Almacena Notificante.
     * @param AuxNotificante String Notificante.
     */
    public static void setNotificante(String AuxNotificante)
    { Notificante = AuxNotificante; }
    /**
     * Devuelve Notificante.
     * @return String Notificante.
     */
    public static String getNotificante()
}
```

```
{ return Notificante; }

/**
 * Borra todos los parámetros del control de incidencia.
 */
public static void Clear()
{
   CodigoIncidencia = null;
    CodigoComunicacion = null;
    Notificante = null;
}

}
```

9.9. Clase ControlRecuperacionDatos.java.-

```
/**
*****
Proyecto final de carrera.
Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ControlRecuperacionDatos.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

/**
 * Clase que almacena información sobre la recuperación de datos que pueda
 * darse en alguno de los recursos controlados en el Sistema de Información.
 */
public class ControlRecuperacionDatos {

    /**
     * Datos que han sido recuperados en el proceso.
     */
    static String Datos = null;

    /**
     * Datos que han sido recuperados manualmente en el proceso.
     */
    static String DatosManuales = null;

    /**
     * Campo que recoge la recuperación de datos ha sido autorizada.
     */
    static boolean RecuperacionAutorizada = false;

    /**
     * Almacena Datos.
     * @param AuxDatos String Datos.
     */
    public static void setDatos(String AuxDatos)
    { Datos = AuxDatos; }

    /**
     * Devuelve Datos.
     * @return String Datos.
     */
    public static String getDatos()
    { return Datos; }

    /**
     * Almacena DatosManuales.
     * @param AuxDatosManuales String DatosManuales.
     */
    public static void setDatosManuales(String AuxDatosManuales)
    { DatosManuales = AuxDatosManuales; }

    /**
     * Devuelve DatosManuales.
     * @return String DatosManuales.
     */
    public static String getDatosManuales()
    { return DatosManuales; }

    /**
     * Almacena RecuperacionAutorizada.
     * @param AuxRecuperacionAutorizada boolean RecuperacionAutorizada.
     */
    public static void setRecuperacionAutorizada(boolean AuxRecuperacionAutorizada)
    { RecuperacionAutorizada = AuxRecuperacionAutorizada; }
```

```
/**
 * Devuelve RecuperacionAutorizada.
 * @return boolean RecuperacionAutorizada.
 */
public static boolean getRecuperacionAutorizada()
{ return RecuperacionAutorizada; }

/**
 * Borra todos los parámetros del control de recuperación de datos.
 */
public static void Clear()
{ Datos = null;
  DatosManuales = null;
  RecuperacionAutorizada = false;
}
}
```

9.10. Clase ControlResponsableSeguridad.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ControlResponsableDeSeguridad.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

/**
 * Clase que almacena el nombre del Responsable de Seguridad definido en el
 * artículo 16 del Real Decreto 994/1999.
 */
public class ControlResponsableSeguridad {

    /** Nombre del Responsable de Seguridad. */
    static String ResponsableSeguridad = null;

    /**
     * Almacena ResponsableSeguridad.
     * @param AuxResponsableSeguridad String ResponsableSeguridad.
     */
    public static void setResponsableSeguridad(String AuxResponsableSeguridad)
    { ResponsableSeguridad = AuxResponsableSeguridad; }

    /**
     * Devuelve ResponsableSeguridad.
     * @return String ResponsableSeguridad.
     */
    public static String getResponsableSeguridad()
    { return ResponsableSeguridad; }

    /**
     * Borra todos los parámetros del control del Responsable de Seguridad.
     */
    public static void Clear()
    { ResponsableSeguridad = null; }
}
```

9.11. Clase ControlSoportesInformaticos.java.-

```

/**
*****
Proyecto final de carrera.
Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ControlSoportesInformaticos.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

/**
 * Clase que almacena información sobre la salida o entrada de soportes informáticos
 * en alguno de los recursos controlados en el Sistema de Información.
 */
public class ControlSoportesInformaticos
{
    /** Número de soportes que entran o salen del sistema. */
    static int NumeroSoportes = 0;
    /** Receptor de la información. */
    static String Receptor = null;
    /** Emisor de la información. */
    static String Emisor = null;
    /** Variable que recoge si existe una autorización escrita que autorice la entrada o
la
    * salida de los soportes seleccionados. */
    static boolean AutorizacionEscrita = false;
    /** Código de la forma de envío de la información. */
    static String CodigoEnvio = null;
    /** Código del tipo de soporte informático */
    static String CodigoSoporte = null;

    /**
     * Recoge el código del soporte informático.
     * @param AuxCodigoSoporte String CodigoSoporte.
     */
    public static void setCodigoSoporte(String AuxCodigoSoporte)
    { CodigoSoporte = AuxCodigoSoporte; }
    /**
     * Devuelve CodigoSoporte.
     * @return String CodigoSoporte.
     */
    public static String getCodigoSoporte()
    { if (CodigoSoporte==null) {CodigoSoporte="No especificado"; }
      return CodigoSoporte;
    }

    /**
     * Recoge CodigoSoporte.
     * @param AuxNumeroSoportes int NumeroSoportes.
     */
    public static void setNumeroSoportes(int AuxNumeroSoportes)
    { NumeroSoportes = AuxNumeroSoportes; }

    /**
     * Devuelve el número de soportes.
     * @return int con el número de soportes.
     */
    public static int getNumeroSoportes()
    { return NumeroSoportes; }

    /**
     * Recoge Receptor.
     * @param AuxReceptor String Receptor.
     */
    public static void setReceptor(String AuxReceptor)
    { Receptor = AuxReceptor; }
    /**

```

```
    * Devuelve Receptor.
    * @return String Receptor.
    */
    public static String getReceptor()
    { return Receptor; }

    /**
    * Recoge Emisor.
    * @param AuxEmisor String Emisor.
    */
    public static void setEmisor(String AuxEmisor)
    { Emisor = AuxEmisor; }
    /**
    * Devuelve Emisor.
    * @return String Emisor.
    */
    public static String getEmisor()
    { return Emisor; }

    /**
    * Recoge AutorizacionEscrita.
    * @param AuxAutorizacionEscrita boolean AutorizacionEscrita.
    */
    public static void setAutorizacionEscrita(boolean AuxAutorizacionEscrita)
    { AutorizacionEscrita = AuxAutorizacionEscrita; }
    /**
    * Devuelve AutorizacionEscrita.
    * @return boolean AutorizacionEscrita.
    */
    public static boolean getAutorizacionEscrita()
    { return AutorizacionEscrita; }

    /**
    * Recoge CódigoEnvío.
    * @param AuxCodigoEnvio String CódigoEnvío.
    */
    public static void setCodigoEnvio(String AuxCodigoEnvio)
    { CódigoEnvio = AuxCodigoEnvio; }
    /**
    * Devuelve CódigoEnvío.
    * @return String CódigoEnvío.
    */
    public static String getCodigoEnvio()
    { if (CodigoEnvio==null) {CodigoEnvio="No especificado";}
      return CódigoEnvio;
    }

    /**
    * Borra todos los parámetros del control de sopostes informáticos.
    */
    public static void Clear()
    {
        CódigoSoporte = null;
        NúmeroSoportes = 0;
        Receptor = null;
        Emisor = null;
        AutorizacionEscrita = false;
        CódigoEnvío = null;
    }
}
```

9.12. Clase Convertir.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Convertir.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

/**
 * Clase utilitaria que facilita la conversión de
 * tipos de datos simples. La clase Convertir.java presenta tres métodos para
 * transformar objetos; String que representen valores numéricos a variables tipo
 * int, variables char que representen valores numéricos a variables tipo int y String
 * que representen valores tipo boolean, a variables tipo boolean.
 */
public class Convertir {

    /**
     * Método que convierte a variable tipo int, el String pasado como argumento.
     * @param AuxString String a convertir.
     * @return int con el valor representado en el String.
     */
    public static int String_Int(String AuxString)
    {
        int Numero = Integer.valueOf(AuxString).intValue();
        return Numero;
    }

    /**
     * Método que convierte a variable tipo int, el parámetro tipo Char pasado como
     * argumento.
     * @param AuxChar Char a convertir.
     * @return int con el valor representado en la variable tipo Char.
     */
    public static int Char_Int(char AuxChar)
    {
        int Numero = Integer.valueOf(AuxChar).intValue();
        return Numero;
    }

    /**
     * Método que convierte a variable tipo boolean, el String pasado como argumento.
     * Para obtener un boolean true el String pasado puede presentar los siguientes
     * valores: TRUE, True o true. Cualquier otro String se asociará con false.
     * @param FALSE_TRUE String a convertir.
     * @return boolean con el valor representado en el String.
     */
    public static boolean String_Boolean (String FALSE_TRUE)
    {
        boolean false_true = false;

        if (FALSE_TRUE.contentEquals("TRUE") ||
            FALSE_TRUE.contentEquals("True") ||
            FALSE_TRUE.contentEquals("true"))
        { false_true = true; }

        return false_true;
    }
}
```

9.13. Clase GenerarPDF.java.-

```

/**
*****
                Proyecto final de carrera.
        Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: GenerarPDF.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

import java.io.FileOutputStream;
import java.io.IOException;

import com.lowagie.text Phrase;
import com.lowagie.text.FontFactory;
import com.lowagie.text.Cell;
import com.lowagie.text.Document;
import com.lowagie.text.DocumentException;
import com.lowagie.text.Table;
import com.lowagie.text.pdf.PdfWriter;
import com.lowagie.text.PageSize;
import com.lowagie.text.Image;
import com.lowagie.text.Paragraph;
import com.lowagie.text.Chunk;
import com.lowagie.text.*;
import java.sql.Connection;
import java.sql.ResultSet;
import java.awt.Color;
import ardb.*;

import javax.servlet.http.HttpSession;
import javax.servlet.ServletContext;

import java.util.Date;
import java.util.Vector;

/**
 * Clase que genera un informe en PDF con la consulta almacenada en los campos de
 * la clase ParametrosPDF.java.
 * El informe presenta un encabezado referente al proyecto, la fecha y la hora en
 * que se crea, el número de informe creado y las tablas con la consulta seleccionada.
 * El archivo será conocido como Informe.pdf y será guardado temporalmente
 * dentro del servidor hasta que se produzca otro informe.
 * La URL del archivo es ..(URL del servidor) /Temporales/Informe.pdf.
 */
public class GenerarPDF {

    static int cuenta = 1;

    /**
     * Método que recoge de la clase ParametrosPDF.java información sobre la consulta
     * seleccionada y crea un informe PDF con los valores asociados a la consulta.
     */
    public static void generarPDF()
    {
        Calendario Hoy = new Calendario();

        try {
            Document document = new Document(PageSize.A4.rotate());
            //Obtenemos el path real de la aplicación del servidor
            ServletContext contexto = ParametrosPDF.getSession().getServletContext();
            String path = contexto.getRealPath("/Temporales/Informe.pdf");
            String path2 = path.replace("\\build\\web", "\\web");
            PdfWriter.getInstance(document, new FileOutputStream(path2));

            String sql = ParametrosPDF.getSql();

```

```

        Connection conexion = Conexion.EstablecerConexion();
        ResultSet rs = RegistroDeAcceso.ConsultarBaseDeDatos(conexion, sql);

        //Abrimos el documento
        document.open();

        // Creamos el encabezado del informe.
        Image ImagenEncabezado =
Image.getInstance("http://localhost:8084/Servidor_de_Registro_de_Acceso/Imagenes/encabez
ado_pdf.bmp");
        ImagenEncabezado.scalePercent(80);
        ImagenEncabezado.setAlignment(Image.LEFT);

        Image ImagenPie =
Image.getInstance("http://localhost:8084/Servidor_de_Registro_de_Acceso/Imagenes/logo_es
i_texto.bmp");
        ImagenPie.scalePercent(80);
        ImagenPie.setAlignment(Image.ALIGN_RIGHT);

        document.add(ImagenEncabezado);

        Paragraph parrafo1 = new Paragraph();
        parrafo1.add(new Chunk("Informe: "+cuenta, new Font(Font.NORMAL, 10)));
        Paragraph parrafo2 = new Paragraph();
        parrafo2.add(new Chunk("Responsable de Seguridad:
"+ControlResponsableSeguridad.getResponsableSeguridad(), new Font(Font.NORMAL, 10)));
        Paragraph parrafo3 = new Paragraph();
        parrafo3.add(new Chunk("Creación: "+ new java.util.Date(), new Font(Font.NORMAL,
10)));

        cuenta++;

        document.add(parrafo1);
        document.add(parrafo2);
        document.add(parrafo3);

        // Creamos una tabla en el PDF con los nombres de la cabecera
        // y el resultado de la consulta.
        Vector cabecera = ParametrosPDF.getNombresCabecera();

        Table Resultado = new Table(cabecera.size());
        Resultado.setPadding(1);
        Resultado.setDefaultCellBackgroundColor(new Color(197,180,183));
        for (int c=0 ; c<cabecera.size() ; c++)
        {   Cell celda = new Cell (new
Phrase(cabecera.elementAt(c).toString(),FontFactory.getFont(FontFactory.HELVETICA,8)));
            celda.setHorizontalAlignment(Cell.ALIGN_CENTER);
            celda.setVerticalAlignment(Cell.ALIGN_CENTER);
            Resultado.addCell(celda);
        }
        Vector campos = ParametrosPDF.getNombresCampos();
        Resultado.setDefaultCellBackgroundColor(new Color(255,255,255));
        while (rs.next()){
            for (int p=1;p<campos.size()+1;p++)
            {
                Cell celda = new Cell (new
Phrase(CambiarFormato(rs.getString(p)),FontFactory.getFont(FontFactory.HELVETICA,8)));
                celda.setHorizontalAlignment(Cell.ALIGN_CENTER);
                celda.setVerticalAlignment(Cell.ALIGN_CENTER);
                Resultado.addCell(celda);
            }
        }
        document.add(Resultado);
        document.add(ImagenPie);

        // cerramos el documento
        document.close();

    }
    catch(DocumentException de) {
        System.err.println("Excepcion de documento: "+de.getMessage());
        System.out.println("Excepcion de documento: "+de.getMessage());
    }
    catch(IOException ioe) {
        System.err.println("Excepcion de escritura: "+ioe.getMessage());
        System.out.println("Excepcion de escritura: "+ioe.getMessage());
    }

```

```
    }
    catch(Exception e){
        System.err.println("Excepcion general: "+e.getMessage());
        System.out.println("Excepcion general: "+e.getMessage());
    }
}

/**
 * Método que transforma el formato de valores consultados a la base
 * de datos que no presentan un significado muy intuitivo.
 *
 * Transformación de parámetros.-
 * t -> "SI".
 * f -> "NO".
 * null -> "No especificado".
 *
 * @param AuxParametro posible String a transformar.
 * @return String con el parámetro.
 */
static public String CambiarFormato (String AuxParametro)
{ if (AuxParametro.contentEquals("t")) { AuxParametro = "SI"; }
  if (AuxParametro.contentEquals("f")) { AuxParametro = "NO"; }
  if (AuxParametro.contentEquals("null")) { AuxParametro = "No especificado"; }

  return AuxParametro;
}
}
```

9.14. Clase Informacion.java.-

```

/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Informacion.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;
import java.sql.Connection;
import java.sql.SQLException;
import ardb.*;

/**
 * Clase que actualiza los campos de las tablas, auxiliares y dedicadas a
 * usuarios del servidor, de la base de datos "Registro de Acceso".
 * Presenta métodos para modificar, ampliar o eliminar la información recogida
 * en estas tablas, así como el método que inicializa la información que contienen.
 */
public class Informacion {

    /**
     * Inserta información inicial en las tablas, auxiliares y dedicadas a usuarios
     * del servidor, necesaria para el funcionamiento originario del servidor.
     * Se establece la conexión con la base de datos y se actualiza con las
     * sentencias SQL. Posteriormente cerramos la conexión.
     * @throws java.lang.Exception al comunicarse con la base de datos.
     */
    public static void Inicial ()
    throws Exception {

        // Connection con la base de datos Registro de Acceso.
        Connection conexion = null;
        // String con la sentencia SQL a ejecutar.
        String Sql = null;

        try {
            conexion = Conexion.EstablecerConexion();
            Sql = ""
            + "INSERT INTO COMUNICACION VALUES ('C-100','Comunicado al usuario por HTML.');"
            + "INSERT INTO COMUNICACION VALUES ('C-101','Comunicado a la base de datos
Registro de Acceso.');"
            + "INSERT INTO COMUNICACION VALUES ('C-102','Comunicación al usuario por HTML y
registrado en la base de datos Registro de Acceso.');"
            + "INSERT INTO COMUNICACION VALUES ('C-103','Comunicación al usuario por HTML,
al Responsable de Seguridad por e-mail y registrado en la base de datos Registro de
Acceso.');"
            + "INSERT INTO COMUNICACION VALUES ('C-200','Comunicado por SMS al Responsable
de Seguridad.');"
            + "INSERT INTO COMUNICACION VALUES ('C-999','Anotar los posibles modos de
comunicación de incidencias de seguridad.');"

            + "INSERT INTO FORMA_DE_ENVIO VALUES ('E-100','Envio por HTTP 1.0.');"
            + "INSERT INTO FORMA_DE_ENVIO VALUES ('E-101','Envio por HTTP 1.1.');"
            + "INSERT INTO FORMA_DE_ENVIO VALUES ('E-102','Envio por SMTP.');"
            + "INSERT INTO FORMA_DE_ENVIO VALUES ('E-200','Envio por FTP.');"
            + "INSERT INTO FORMA_DE_ENVIO VALUES ('E-300','Envio por IPsec.');"
            + "INSERT INTO FORMA_DE_ENVIO VALUES ('E-999','Anotar las posibles formas de
envio.');"

            + "INSERT INTO INCIDENCIAS VALUES ('I-100','No identificado el
usuario.','Intranscendentes: El usuario no puede acceder a ningún recurso.');"
            + "INSERT INTO INCIDENCIAS VALUES ('I-101','Error en la contraseña del
usuario.','Intranscendentes: El usuario no puede acceder a ningún recurso.');"
            + "INSERT INTO INCIDENCIAS VALUES ('I-102','El usuario no presenta
credenciales.','Intranscendentes: El usuario no puede acceder a ningún recurso.');"

```

```
+ "INSERT INTO INCIDENCIAS VALUES ('I-103','Intento de acceder a los recursos
sin identificación.','Alerta: Intento de violación del dominio de seguridad.');"
+ "INSERT INTO INCIDENCIAS VALUES ('I-104','Las credenciales del usuario no le
permiten acceder al recurso solicitado.','Alerta: Intento de violación del dominio de
seguridad.');"
+ "INSERT INTO INCIDENCIAS VALUES ('I-105','Las credenciales del usuario en el
Servidor de Registro de Acceso no le permiten acceder al recurso.','Alerta: Intento de
violación del dominio de seguridad.');"
+ "INSERT INTO INCIDENCIAS VALUES ('I-106','El usuario del Servidor de Registro
de Acceso no presenta las credenciales necesarias.','No podrá utilizar el servicio
requerido del Servidor de Registro de Acceso.');"
+ "INSERT INTO INCIDENCIAS VALUES ('I-300','El recurso al que accede el cliente
no responde.','Alerta: Posible uso malicioso del recurso.');"
+ "INSERT INTO INCIDENCIAS VALUES ('I-400','Operación de escritura no
autorizada.','Alerta: Posible violación de los datos.');"
+ "INSERT INTO INCIDENCIAS VALUES ('I-500','Violación del tiempo de
respuesta.','Alerta: Posible información no encriptada.');"
+ "INSERT INTO INCIDENCIAS VALUES ('I-600','Recuperación de datos.','Atención:
Se pueden sobrescribir datos actualizados.');"
+ "INSERT INTO INCIDENCIAS VALUES ('I-999','Anotar las posibles incidencias que
afecten a la seguridad de los datos.','Intrascendentes.');"

+ "INSERT INTO MOTIVO_DE_DENEGACION VALUES ('D-000','Acceso permitido');"
+ "INSERT INTO MOTIVO_DE_DENEGACION VALUES ('D-100','El recurso no responde.');"
+ "INSERT INTO MOTIVO_DE_DENEGACION VALUES ('D-101','Las credenciales del
usuario impiden acceder al fichero.');"
+ "INSERT INTO MOTIVO_DE_DENEGACION VALUES ('D-102','El recurso no está
disponible.');"
+ "INSERT INTO MOTIVO_DE_DENEGACION VALUES ('D-200','El fichero está siendo
utilizado por otro usuario.');"
+ "INSERT INTO MOTIVO_DE_DENEGACION VALUES ('D-999','Anotar los posibles motivos
de denegación en el acceso al fichero.');"

+ "INSERT INTO SITUACION_ACCESO VALUES ('A-100','Acceso por profesionales
asistenciales');"
+ "INSERT INTO SITUACION_ACCESO VALUES ('A-101','Acceso con fines judiciales');"
+ "INSERT INTO SITUACION_ACCESO VALUES ('A-102','Acceso con fines
epidemiológicos');"
+ "INSERT INTO SITUACION_ACCESO VALUES ('A-103','Acceso con fines de salud
pública');"
+ "INSERT INTO SITUACION_ACCESO VALUES ('A-104','Acceso con fines de
investigación');"
+ "INSERT INTO SITUACION_ACCESO VALUES ('A-105','Acceso con fines de
docencia');"
+ "INSERT INTO SITUACION_ACCESO VALUES ('A-106','Acceso por personal sanitario
acreditado con funciones de inspección, evaluación, acreditación y planificación');"
+ "INSERT INTO SITUACION_ACCESO VALUES ('A-107','Acceso por el interesado');"
+ "INSERT INTO SITUACION_ACCESO VALUES ('A-108','Acceso del administrador.
Configurando el Sistema de Información.');"
+ "INSERT INTO SITUACION_ACCESO VALUES ('A-999','Anotar las posibles situaciones
de acceso.');"

+ "INSERT INTO CREDENCIALES VALUES ('CR-100','Acceso autorizado a todo el
servidor','1-999');"
+ "INSERT INTO CREDENCIALES VALUES ('CR-101','Acceso de sólo lectura a las
consultas expuestas en el servidor','16-199');"
+ "INSERT INTO CREDENCIALES VALUES ('CR-102','Acceso a consultas generales','16-
23');"
+ "INSERT INTO CREDENCIALES VALUES ('CR-103','Acceso a consultas estandar','16-
39');"
+ "INSERT INTO CREDENCIALES VALUES ('CR-104','Acceso a personalizar
consultas','300');"
+ "INSERT INTO CREDENCIALES VALUES ('CR-999','Insertar permisos dentro de mi
servidor','');"

+ "INSERT INTO RECURSOS VALUES ('1','Consulta de todos los campos de la tabla
COMUNICACION de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('2','Consulta de todos los campos de la tabla
CREDENCIALES de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('3','Consulta de todos los campos de la tabla
FORMA_DE_ENVIO de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('4','Consulta de todos los campos de la tabla
INCIDENCIAS de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('5','Consulta de todos los campos de la tabla
MOTIVO_DE_DENEGACION de la base de datos Registro de Acceso.');
```

```

+ "INSERT INTO RECURSOS VALUES ('6','Consulta de todos los campos de la tabla
PERSONAL_AUTORIZADO de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('7','Consulta de todos los campos de la tabla
SITUACION_ACCESO de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('8','Consulta de todos los campos de la tabla
TIPO_DE_SOPORTE de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('9','Consulta de todos los campos de la tabla
REGISTRO_DE_ACCESOS de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('10','Consulta de todos los campos de la tabla
REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('11','Consulta de todos los campos de la tabla
REGISTRO_DE_INCIDENCIAS de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('12','Consulta de todos los campos de la tabla
REGISTRO_DE_RECUPERACION_DE_DATOS de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('13','Consulta de todos los campos de la tabla
REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('14','Consulta de todos los campos de la tabla
REGISTRO_RESPALDO de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('15','Consulta de todos los campos de la tabla
RECURSOS de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('16','Consulta de incidencias.');"
+ "INSERT INTO RECURSOS VALUES ('17','Consulta sobre la recuperación de
datos.');"
+ "INSERT INTO RECURSOS VALUES ('18','Consulta sobre entrada de información.');"
+ "INSERT INTO RECURSOS VALUES ('19','Consulta sobre salida de información.');"
+ "INSERT INTO RECURSOS VALUES ('20','Consulta: Acceso según situación.');"
+ "INSERT INTO RECURSOS VALUES ('21','Consulta: ¿Quién ha accedido al
fichero?.');"
+ "INSERT INTO RECURSOS VALUES ('22','Consulta: ¿Qué fichero se accedio?.');"
+ "INSERT INTO RECURSOS VALUES ('23','Consulta: Ficheros con copia de
respaldo.');"
+ "INSERT INTO RECURSOS VALUES ('30','Consulta Dinámica: Registro de Incidencias
por fechas.');"
+ "INSERT INTO RECURSOS VALUES ('31','Consulta Dinámica: Registro de Accesos por
fechas.');"
+ "INSERT INTO RECURSOS VALUES ('32','Consulta Dinámica: Registro de Entrada de
Soportes Informáticos por fechas.');"
+ "INSERT INTO RECURSOS VALUES ('33','Consulta Dinámica: Registro de Salida de
Soportes Informáticos por fechas.');"
+ "INSERT INTO RECURSOS VALUES ('200','Configuración de la tabla COMUNICACION de
la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('201','Configuración de la tabla CREDENCIALES de
la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('202','Configuración de la tabla FORMA_DE_ENVIO
de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('203','Configuración de la tabla INCIDENCIAS de
la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('204','Configuración de la tabla
MOTIVO_DE_DENEGACION de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('205','Configuración de la tabla
PERSONAL_AUTORIZADO de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('206','Configuración de la tabla
SITUACION_ACCESO de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('207','Configuración de la tabla TIPO_DE_SOPORTE
de la base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('215','Configuración de la tabla RECURSOS de la
base de datos Registro de Acceso.');"
+ "INSERT INTO RECURSOS VALUES ('300','Personalizar consultas.');"

+ "INSERT INTO PERSONAL_AUTORIZADO VALUES ('jose maria','00000000A','José María
García Sánchez','CR-100');"
+ "INSERT INTO PERSONAL_AUTORIZADO VALUES ('maria','11111111B','María José Núñez
González','CR-102');"
+ "INSERT INTO PERSONAL_AUTORIZADO VALUES ('hector','22222222C','Héctor Moisés
García Malmierca','CR-102');"
+ "INSERT INTO PERSONAL_AUTORIZADO VALUES ('diana','33333333D','Diana Carretero
Malmierca','CR-102');"
+ "INSERT INTO PERSONAL_AUTORIZADO VALUES ('jose','44444444E','José Antonio
Núñez Santos','CR-102');"

+ "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-001','Archivo *.txt - Texto
normal.');"
+ "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-002','Archivo *.doc - Microsoft
Word');"
```

```

        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-003','Archivo *.rtf - Aplicación MFC
WORDPAD');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-004','Archivo *.pdf - Acrobat
Reader');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-100','Archivo *.jpg - Formato de
imágenes');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-101','Archivo *.gif - Formato de
imágenes');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-102','Archivo *.bmp - Imagen en mapa
de bits');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-103','Archivo *.tif - Formato de
imágenes');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-104','Archivo *.png - Formato de
imágenes');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-200','Archivo *.xls - Microsoft
Excel');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-201','Archivo *.html - Formato web');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-202','Archivo *.wav - Formato de
sonido');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-203','Archivo *.mp3 - Formato de
sonido');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-204','Archivo *.cda - Formato de
sonido');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-300','Archivo *.rar - Formato de
compresión de archivos');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-301','Archivo *.zip - Formato de
compresión de archivos');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-302','Archivo *.iso - Formato de
compresión de archivos');"
        + "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-999','Anotar los distintos tipos de
archivos intercambiables en este tipo de instituciones.');"
    ;

    if (RegistroDeAcceso.ActualizarBaseDeDatos(conexion, Sql))
    { Conexion.CerrarConexion(conexion); }
    catch (SQLException sqlEx)
    {
        System.out.println("Insertar Informacion.java: Excepcion SQL al actualizar
la base de datos: " + sqlEx.getMessage());
    }
}

/**
 * Método que actualiza los valores de los campos de las tablas: comunicacion,
 * forma_de_envio, motivo_de_denegacion, situacion_acceso, tipo_de_soporte y
 * recursos. Las tablas se actualizan con la información pasada como argumento.
 * El parámetro Eleccion determina la tabla a actualizar:
 * Eleccion = 1. Tabla comunicacion.
 * Eleccion = 3. Tabla forma_de_envio.
 * Eleccion = 5. Tabla motivo_de_denegacion.
 * Eleccion = 7. Tabla situacion_acceso.
 * Eleccion = 8. Tabla tipo_de_soporte.
 * Eleccion = 15. Tabla recursos.
 * Los parámetrosCodigo y Descripcion indican el código a actualizar y la
 * nueva descripción del mismo.
 * @return Mensaje con la confirmación o la negación de la actualización de la
 * base de datos.
 * @param Eleccion entero que selecciona la tabla a actualizar.
 * @param Codigo String con el código a actualizar.
 * @param Descripcion String con la nueva descripción.
 * @throws java.lang.Exception al comunicarse con la base de datos.
 */
public static String Actualizar (int Eleccion, String Codigo, String Descripcion)
throws Exception {

    // String con la sentencia SQL a ejecutar.
    String Sql = null;
    // String con el mensaje a devolver.
    String Mensaje = null;
    // String con el código insertado en la base de datos.
    String AuxCodigo = null;

    AuxCodigo = TransformarTresCifras(Codigo);

```

```

        switch (Eleccion)
        { case 1: Sql = "UPDATE COMUNICACION SET descripcion = '"+Descripcion+"' "
            + "WHERE codigo_comunicacion = 'C-'+AuxCodigo+'"; break;

            case 3: Sql = "UPDATE FORMA_DE_ENVIO SET descripcion = '"+Descripcion+"' "
            + "WHERE codigo_envio = 'E-'+AuxCodigo+'"; break;

            case 5: Sql = "UPDATE MOTIVO_DE_DENEGACION SET descripcion = "
            + "WHERE codigo_denegacion = 'D-'+AuxCodigo+'"; break;

            case 7: Sql = "UPDATE SITUACION_ACCESO SET descripcion = "
            + "WHERE codigo_acceso = 'A-'+AuxCodigo+'"; break;

            case 8: Sql = "UPDATE TIPO_DE_SOPORTE SET descripcion = '"+Descripcion+"' "
            + "WHERE codigo_soporte = 'S-'+AuxCodigo+'"; break;

            case 15: Sql = "UPDATE RECURSOS SET descripcion = '"+Descripcion+"' "
            + "WHERE recurso = '"+AuxCodigo+'"; break;

        }

        Mensaje = Actualizacion(Sql);
        return Mensaje;
    }

    /**
     * Método que actualiza los valores de los campos de las tablas: credenciales
     * e incidencias. Las tablas se actualizan con la información pasada como
     * argumento.
     * El parámetro Eleccion determina la tabla a actualizar:
     * Eleccion = 2. Tabla credenciales.
     * Eleccion = 4. Tabla incidencias.
     * Los parámetrosCodigo, Descripcion y Recurso_Efectos indican el código a
     * actualizar, la nueva descripción del mismo y la elección de nuevos recursos
     * o la descripción de nuevos efectos.
     * @return Mensaje con la confirmación o la negación de la actualización de la
     * base de datos.
     * @param Eleccion entero que selecciona la tabla a actualizar.
     * @param Codigo String con el código a insertar.
     * @param Descripcion String con la nueva descripción.
     * @param Recurso_Efectos String con la nueva selección de recursos o la nueva
     * descripción de efectos.
     * @throws java.lang.Exception al comunicarse con la base de datos.
     */
    public static String Actualizar (int Eleccion, String Codigo, String Descripcion,
    String Recurso_Efectos)
    throws Exception {

        // String con la sentencia SQL a ejecutar.
        String Sql = null;
        // String con el mensaje a devolver.
        String Mensaje = null;
        // String con el código insertado en la base de datos.
        String AuxCodigo = null;

        AuxCodigo = TransformarTresCifras(Codigo);

        switch (Eleccion)
        { case 2: Sql = "UPDATE CREDENCIALES SET descripcion = '"+Descripcion+"', "
            + " recursos = '"+Recurso_Efectos+"' WHERE "
            + "codigo_credenciales = 'CR-'+AuxCodigo+'"; break;
            case 4: Sql = "UPDATE INCIDENCIAS SET descripcion = '"+Descripcion+"', "
            + " efectos = '"+Recurso_Efectos+"' WHERE codigo_incidencia = "
            + " 'I-'+AuxCodigo+'"; break;
        }

        Mensaje = Actualizacion(Sql);
        return Mensaje;
    }
}

```

```

/**
 * Método que actualiza los valores de los campos de la tabla personal_autorizado.
 * La tabla se actualiza con la información pasada como argumento.
 * Los parámetros Login, Password e IdUsuario indican el login a
 * actualizar, la nueva password y la nueva identificación del usuario.
 * @return Mensaje con la confirmación o la negación de la actualización de la
 * base de datos.
 * @param Login String con el login del usuario a actualizar.
 * @param Password String con la nueva Password a asignar al usuario.
 * @param IdUsuario String con la nueva identificación del usuario.
 * @throws java.lang.Exception al comunicarse con la base de datos.
 */
public static String Actualizar (String Login, String Password, String IdUsuario)
throws Exception {

    // String con la sentencia SQL a ejecutar.
    String Sql = null;
    // String con el mensaje a devolver.
    String Mensaje = null;

    Sql = "UPDATE PERSONAL_AUTORIZADO SET password = '"+Password+"', "
        + " id_usuario = '"+IdUsuario+"'"
        + " WHERE login = '"+Login+"';";

    Mensaje = Actualizacion(Sql);
    return Mensaje;
}

/**
 * Método que inserta nueva información en las tablas: comunicacion,
 * forma_de_envio, motivo_de_denegacion, situacion_acceso, tipo_de_soporte y
 * recursos. Las tablas se actualizan con la información pasada como argumento.
 * El parámetro Eleccion determina la tabla a actualizar:
 * Eleccion = 1. Tabla comunicacion.
 * Eleccion = 3. Tabla forma_de_envio.
 * Eleccion = 5. Tabla motivo_de_denegacion.
 * Eleccion = 7. Tabla situacion_acceso.
 * Eleccion = 8. Tabla tipo_de_soporte.
 * Eleccion = 15. Tabla recursos.
 * Los parámetros Codigo y Descripcion indican el nuevo código a insertar y la
 * nueva descripción del mismo.
 * @return Mensaje con la confirmación o la negación de la actualización de la
 * base de datos.
 * @param Eleccion entero que selecciona la tabla a actualizar.
 * @param Codigo String con el código a insertar.
 * @param Descripcion String con la nueva descripción.
 * @throws java.lang.Exception al comunicarse con la base de datos.
 */
public static String Insertar (int Eleccion, String Codigo, String Descripcion)
throws Exception {

    // String con la sentencia SQL a ejecutar.
    String Sql = null;
    // String con el AuxCodigo insertado en la base de datos.
    String AuxCodigo = null;
    // String con el mensaje a devolver.
    String Mensaje = null;

    AuxCodigo = TransformarTresCifras(Codigo);

    switch (Eleccion)
    { case 1: Sql = "INSERT INTO COMUNICACION VALUES ('C-"+AuxCodigo+"',"
        + "'"+Descripcion+"');"; break;

        case 3: Sql = "INSERT INTO FORMA_DE_ENVIO VALUES ('E-"+AuxCodigo+"',"
        + "'"+Descripcion+"');"; break;

        case 5: Sql = "INSERT INTO MOTIVO_DE_DENEGACION VALUES ('D-
"+AuxCodigo+"',"
        + "'"+Descripcion+"');"; break;

        case 7: Sql = "INSERT INTO SITUACION_ACCESO VALUES ('A-"+AuxCodigo+"',"
        + "'"+Descripcion+"');"; break;

        case 8: Sql = "INSERT INTO TIPO_DE_SOPORTE VALUES ('S-"+AuxCodigo+"',"
        + "'"+Descripcion+"');"; break;
    }
}

```

```

        case 15: Sql = "INSERT INTO RECURSOS VALUES ('"+AuxCodigo+"',"
            + "'"+Descripcion+"')"; break;
    }

    Mensaje = Actualizacion(Sql);
    return Mensaje;
}

/**
 * Método que inserta nueva información en las tablas credenciales e
 * incidencias. Las tablas se actualizan con la información pasada como
 * argumento.
 * El parámetro Eleccion determina la tabla a actualizar:
 * Eleccion = 2. Tabla credenciales.
 * Eleccion = 4. Tabla incidencias.
 * Los parámetrosCodigo, Descripcion y Recurso_Efectos indican el nuevo código
 * a insertar, la nueva descripción del mismo y la selección de nuevos recursos
 * o la descripción de nuevos efectos.
 * @return Mensaje con la confirmación o la negación de la actualización de la
 * base de datos.
 * @param Eleccion entero que selecciona la tabla a actualizar.
 * @param Codigo String con el código a insertar.
 * @param Descripcion String con la nueva descripción.
 * @param Recurso_Efectos String con la nueva selección de recursos o la nueva
 * descripción de efectos.
 * @throws java.lang.Exception al comunicarse con la base de datos.
 */
public static String Insertar (int Eleccion, String Codigo, String Descripcion, String
Recurso_Efectos)
    throws Exception {

    // String con la sentencia SQL a ejecutar.
    String Sql = null;
    // String con el Código insertado en la base de datos.
    String AuxCodigo = null;
    // String con el mensaje a devolver.
    String Mensaje = null;

    AuxCodigo = TransformarTresCifras(Codigo);

    switch (Eleccion)
    { case 2: Sql = "INSERT INTO CREDENCIALES VALUES ('CR-"+AuxCodigo+"',"
        + "'"+Descripcion+"','"+Recurso_Efectos+"')"; break;

        case 4: Sql = "INSERT INTO INCIDENCIAS VALUES ('I-"+AuxCodigo+"',"
        + "'"+Descripcion+"','"+Recurso_Efectos+"')"; break;

    }

    Mensaje = Actualizacion(Sql);
    return Mensaje;
}

/**
 * Método que inserta nueva información en la tabla personal autorizado.
 * La tabla se actualiza con la información pasada como argumento.
 * Los parámetros Login, Password, IdUsuario y Credenciales indican el nuevo
 * login a insertar, el password proporcionado, la identificación del usuario
 * y el código que especifica sus credenciales en el servidor.
 * @return Mensaje con la confirmación o la negación de la actualización de la
 * base de datos.
 * @param Login nuevo login a insertar.
 * @param Password nueva password asignada.
 * @param IdUsuario nueva identificación del usuario.
 * @param Credenciales nuevas credenciales asignadas.
 * @throws java.lang.Exception al comunicarse con la base de datos.
 */
public static String Insertar (String Login, String Password, String IdUsuario, String
Credenciales)
    throws Exception {

    // String con la sentencia SQL a ejecutar.
    String Sql = null;
    // String con el mensaje a devolver.

```

```

        String Mensaje = null;

        Sql = "INSERT INTO PERSONAL_AUTORIZADO VALUES ('"+Login+"',"
            + "'"+Password+"','"+IdUsuario+"',"
            + "'CR-"+Credenciales+"')";

        Mensaje = Actualizacion(Sql);
        return Mensaje;
    }

    /**
     * Método que elimina registros presentes en las tablas, auxiliares y dedicadas
     * a usuarios del servidor, de la base de datos "Registro de Acceso".
     * Las tablas se actualizan con la información pasada como argumento.
     * El parámetro Eleccion determina la tabla a actualizar:
     * Eleccion = 1. Tabla comunicacion.
     * Eleccion = 2. Tabla credenciales.
     * Eleccion = 3. Tabla forma_de_envio.
     * Eleccion = 4. Tabla incidencias.
     * Eleccion = 5. Tabla motivo_de_denegacion.
     * Eleccion = 6. Tabla personal_autorizado.
     * Eleccion = 7. Tabla situacion_acceso.
     * Eleccion = 8. Tabla tipo_de_soporte.
     * Eleccion = 15. Tabla recursos.
     * El parámetroCodigo_Login indica el código o el login a eliminar.
     * @return Mensaje con la confirmación o la negación de la eliminación del
     * registro.
     * @param Eleccion entero que selecciona la tabla a actualizar.
     * @param Codigo_Login String con el código o el login a eliminar.
     * @throws java.lang.Exception al comunicarse con la base de datos.
     */
    public static String Borrar (int Eleccion, String Codigo_Login)
        throws Exception {

        // String con la sentencia SQL a ejecutar.
        String Sql = null;
        // String con el mensaje a devolver.
        String Mensaje = null;

        switch (Eleccion)
        {
            case 1: Sql = "DELETE FROM COMUNICACION WHERE codigo_comunicacion = 'C-
"+Codigo_Login+"'"; break;

            case 2: Sql = "DELETE FROM CREDENCIALES WHERE codigo_credenciales = 'CR-
"+Codigo_Login+"'"; break;

            case 3: Sql = "DELETE FROM FORMA_DE_ENVIO WHERE codigo_envio = 'E-
"+Codigo_Login+"'"; break;

            case 4: Sql = "DELETE FROM INCIDENCIAS WHERE codigo_incidencia = 'I-
"+Codigo_Login+"'"; break;

            case 5: Sql = "DELETE FROM MOTIVO_DE_DENEGACION WHERE codigo_denegacion
= 'D-"+Codigo_Login+"'"; break;

            case 6: Sql = "DELETE FROM PERSONAL_AUTORIZADO WHERE login =
'"+Codigo_Login+"'"; break;

            case 7: Sql = "DELETE FROM SITUACION_ACCESO WHERE codigo_acceso = 'A-
"+Codigo_Login+"'"; break;

            case 8: Sql = "DELETE FROM TIPO_DE_SOPORTE WHERE codigo_soporte = 'S-
"+Codigo_Login+"'"; break;

            case 15: Sql = "DELETE FROM RECURSOS WHERE recurso =
'"+Codigo_Login+"'"; break;
        }

        Mensaje = Actualizacion(Sql);
        return Mensaje;
    }

    /**

```

```

* Método que ejecuta la actualización seleccionada en los métodos de esta
* clase. Recibe la sentencia SQL creada previamente y la comunica a la base
* de datos "Registro de Acceso", devolviendo un mensaje de confirmación o en
* su caso de negación de la actualización.
* @return Mensaje con la confirmación o la negación de la actualización de la
* base de datos.
* @param Sql sentencia SQL de actualización.
* @throws java.lang.Exception al comunicarse con la base de datos.
*/
public static String Actualizacion (String Sql)
throws Exception
{
    // String con el mensaje a devolver.
    String Mensaje = null;
    // Connection con la base de datos Registro de Acceso.
    Connection conexion = null;

    try {
        conexion = Conexion.EstablecerConexion();
        if (RegistroDeAcceso.ActualizarBaseDeDatos(conexion, Sql))
        {
            Conexion.CerrarConexion(conexion);
            Mensaje = "El mensaje de actualización se ha ejecutado.";
        }
        else {
            Conexion.CerrarConexion(conexion);
            Mensaje = "No se ha podido actualizar la base de datos 'Registro de
Acceso'.";
        }
    }
    catch (SQLException sqlEx)
    {
        System.out.println("Informacion.java: Excepcion SQL al actualizar la base de
datos: " + sqlEx.getMessage());
        Mensaje = "No se ha podido actualizar la base de datos 'Registro de
Acceso'.";
    }

    return Mensaje;
}

/**
* Método que convierte el código seleccionado por el usuario y en un código de 3
* cifras, a insertar dentro de la sentencia SQL de actualización.
* En el caso de que el código obtenido esté fuera del rango 1
* a 1000 se devolverá un valor null que provocará la excepción en la
* actualización.
* @return Mensaje con la confirmación o la negación de la actualización de la
* base de datos.
* @param AuxCodigo código a convertir.
* @throws java.lang.Exception al comunicarse con la base de datos.
*/
public static String TransformarTresCifras (String AuxCodigo)
throws Exception
{
    // String con el código a transformar.
    String Codigo = null;

    // Transformamos el código para que tenga tres cifras.
    try {
        if ((AuxCodigo.length()<4)&&(AuxCodigo.length()>0))
        {
            int numero = Convertir.String_Int(AuxCodigo);
            if ((numero>99)&&(numero<1000))
            {
                Codigo = AuxCodigo;
            }
            else if (numero>9)
            {
                Codigo="0"+AuxCodigo;
            }
            else if (numero>0)
            {
                Codigo="00"+AuxCodigo;
            }
            else {
                Codigo = null;
            }
        }
        else {
            Codigo = null;
        }
    }
    catch (Exception error) {
        Codigo = null;
    }

    return Codigo;
}
}

```

9.15. Clase ParametrosDeConsulta.java.-

```

/**
*****
                Proyecto final de carrera.
        Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ParametrosDeConsulta.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;
import java.util.*;

/**
 * Clase utilizada para almacenar la información, existente en los distintos campos
 * de las tablas de la base de datos, cuando se realiza una consulta.
 * Para simplificar la tarea de representación de datos al cliente se le devuelve
 * un Vector donde cada elemento es un objeto de tipo ParametrosDeConsulta.java.
 * En cada elemento del Vector se recoge una fila de la consulta realizada.
 * El número máximo de campos a consultar está limitado a 18 (número de campos en
 * la tabla "registro_de_accesos").
 */
public class ParametrosDeConsulta {

    /** Valor del primer campo consultado. */
    String Parametro1 = null;
    /** Valor del segundo campo consultado. */
    String Parametro2 = null;
    /** Valor del tercero campo consultado. */
    String Parametro3 = null;
    /** Valor del cuarto campo consultado. */
    String Parametro4 = null;
    /** Valor del quinto campo consultado. */
    String Parametro5 = null;
    /** Valor del sexto campo consultado. */
    String Parametro6 = null;
    /** Valor del séptimo campo consultado. */
    String Parametro7 = null;
    /** Valor del octavo campo consultado. */
    String Parametro8 = null;
    /** Valor del noveno campo consultado. */
    String Parametro9 = null;
    /** Valor del décimo campo consultado. */
    String Parametro10 = null;
    /** Valor del undécimo campo consultado. */
    String Parametro11 = null;
    /** Valor del duodécimo campo consultado. */
    String Parametro12 = null;
    /** Valor del decimotercero campo consultado. */
    String Parametro13 = null;
    /** Valor del decimocuarto campo consultado. */
    String Parametro14 = null;
    /** Valor del decimoquinto campo consultado. */
    String Parametro15 = null;
    /** Valor del decimosexto campo consultado. */
    String Parametro16 = null;
    /** Valor del decimoséptimo campo consultado. */
    String Parametro17 = null;
    /** Valor del decimooctavo campo consultado. */
    String Parametro18 = null;

    /**
     * Almacena Parametro1.
     * @param AuxParametro1 String Parametro1.
     */
    public void setParametro1 (String AuxParametro1) { Parametro1 =
        SioNo(AuxParametro1); }
}

```

```
* Devuelve Parametro1
* @return String Parametro1
*/
public String getParametro1() { return Parametro1; }

/**
 * Almacena Parametro2.
 * @param AuxParametro2 String Parametro2.
 */
public void setParametro2 (String AuxParametro2) { Parametro2 =
SioNo(AuxParametro2); }
/**
 * Devuelve Parametro2
 * @return String Parametro2
 */
public String getParametro2() { return Parametro2; }

/**
 * Almacena Parametro3.
 * @param AuxParametro3 String Parametro3.
 */
public void setParametro3 (String AuxParametro3) { Parametro3 =
SioNo(AuxParametro3); }
/**
 * Devuelve Parametro3
 * @return String Parametro3
 */
public String getParametro3() { return Parametro3; }

/**
 * Almacena Parametro4.
 * @param AuxParametro4 String Parametro4.
 */
public void setParametro4 (String AuxParametro4) { Parametro4 =
SioNo(AuxParametro4); }
/**
 * Devuelve Parametro4
 * @return String Parametro4
 */
public String getParametro4() { return Parametro4; }

/**
 * Almacena Parametro5.
 * @param AuxParametro5 String Parametro5.
 */
public void setParametro5 (String AuxParametro5) { Parametro5 =
SioNo(AuxParametro5); }
/**
 * Devuelve Parametro5
 * @return String Parametro5
 */
public String getParametro5() { return Parametro5; }

/**
 * Almacena Parametro6.
 * @param AuxParametro6 String Parametro6.
 */
public void setParametro6 (String AuxParametro6) { Parametro6 =
SioNo(AuxParametro6); }
/**
 * Devuelve Parametro6
 * @return String Parametro16
 */
public String getParametro6() { return Parametro6; }

/**
 * Almacena Parametro7.
 * @param AuxParametro7 String Parametro7.
 */
public void setParametro7 (String AuxParametro7) { Parametro7 =
SioNo(AuxParametro7); }
/**
 * Devuelve Parametro7
 * @return String Parametro7
 */
*/
```

```
public String getParametro7() { return Parametro7; }

/**
 * Almacena Parametro8.
 * @param AuxParametro8 String Parametro8.
 */
public void setParametro8 (String AuxParametro8) { Parametro8 =
SioNo(AuxParametro8); }
/**
 * Devuelve Parametro8
 * @return String Parametro8
 */
public String getParametro8() { return Parametro8; }

/**
 * Almacena Parametro9.
 * @param AuxParametro9 String Parametro9.
 */
public void setParametro9 (String AuxParametro9) { Parametro9 =
SioNo(AuxParametro9); }
/**
 * Devuelve Parametro9
 * @return String Parametro9
 */
public String getParametro9() { return Parametro9; }

/**
 * Almacena Parametro10.
 * @param AuxParametro10 String Parametro10.
 */
public void setParametro10 (String AuxParametro10) { Parametro10 =
SioNo(AuxParametro10); }
/**
 * Devuelve Parametro10
 * @return String Parametro10
 */
public String getParametro10() { return Parametro10; }

/**
 * Almacena Parametro11.
 * @param AuxParametro11 String Parametro11.
 */
public void setParametro11 (String AuxParametro11) { Parametro11 =
SioNo(AuxParametro11); }
/**
 * Devuelve Parametro11
 * @return String Parametro11
 */
public String getParametro11() { return Parametro11; }

/**
 * Almacena Parametro12.
 * @param AuxParametro12 String Parametro12.
 */
public void setParametro12 (String AuxParametro12) { Parametro12 =
SioNo(AuxParametro12); }
/**
 * Devuelve Parametro12
 * @return String Parametro12
 */
public String getParametro12() { return Parametro12; }

/**
 * Almacena Parametro13.
 * @param AuxParametro13 String Parametro13.
 */
public void setParametro13 (String AuxParametro13) { Parametro13 =
SioNo(AuxParametro13); }
/**
 * Devuelve Parametro13
 * @return String Parametro13
 */
public String getParametro13() { return Parametro13; }

/**
 * Almacena Parametro14.
```

```
* @param AuxParametro14 String Parametro14.
*/
public void setParametro14 (String AuxParametro14) { Parametro14 =
SioNo(AuxParametro14); }
/**
 * Devuelve Parametro14
 * @return String Parametro14
 */
public String getParametro14() { return Parametro14; }

/**
 * Almacena Parametro15.
 * @param AuxParametro15 String Parametro15.
 */
public void setParametro15 (String AuxParametro15) { Parametro15 =
SioNo(AuxParametro15); }
/**
 * Devuelve Parametro15
 * @return String Parametro15
 */
public String getParametro15() { return Parametro15; }

/**
 * Almacena Parametro16.
 * @param AuxParametro16 String Parametro16.
 */
public void setParametro16 (String AuxParametro16) { Parametro16 =
SioNo(AuxParametro16); }
/**
 * Devuelve Parametro16
 * @return String Parametro16
 */
public String getParametro16() { return Parametro16; }

/**
 * Almacena Parametro17.
 * @param AuxParametro17 String Parametro17.
 */
public void setParametro17 (String AuxParametro17) { Parametro17 =
SioNo(AuxParametro17); }
/**
 * Devuelve Parametro17
 * @return String Parametro17
 */
public String getParametro17() { return Parametro17; }

/**
 * Almacena Parametro18.
 * @param AuxParametro18 String Parametro18.
 */
public void setParametro18 (String AuxParametro18) { Parametro18 =
SioNo(AuxParametro18); }
/**
 * Devuelve Parametro18
 * @return String Parametro18
 */
public String getParametro18() { return Parametro18; }

/**
 * Método que transforma objetos tipo String devueltos por la base de datos
 * a un formato más inteligible.
 * @param AuxParametro String Parametro con posibilidades de transformación.
 * @return String con el Parametro transformado o no.
 */
public String SioNo (String AuxParametro)
{ if (AuxParametro.contentEquals("t")) { AuxParametro = "SI"; }
  if (AuxParametro.contentEquals("f")) { AuxParametro = "NO"; }
  if (AuxParametro.contentEquals("null")) { AuxParametro = "No especificado"; }

  return AuxParametro;
}

/**
 * Borra todos los parámetros de consulta.
 */
public void Clear()
```

```
{
    Parametro1 = null;
    Parametro2 = null;
    Parametro3 = null;
    Parametro4 = null;
    Parametro5 = null;
    Parametro6 = null;
    Parametro7 = null;
    Parametro8 = null;
    Parametro9 = null;
    Parametro10 = null;
    Parametro11 = null;
    Parametro12 = null;
    Parametro13 = null;
    Parametro14 = null;
    Parametro15 = null;
    Parametro16 = null;
    Parametro17 = null;
    Parametro18 = null;
}
```

9.16. Clase ParametrosPDF.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ParametrosPDF.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

import javax.servlet.http.HttpSession;
import java.util.Vector;

/**
 * Clase que almacena la información necesaria para generar el informe PDF.
 */
public class ParametrosPDF {

    /**
     * Objeto tipo Vector donde cada elemento se corresponde con el nombre del campo a
     * consultar.
     */
    static Vector NombresCampos = null;

    /**
     * Objeto tipo Vector donde cada elemento se corresponde con el nombre de cabecera
    que
     * se desea aplicar a cada campo para su representación.
     */
    static Vector NombresCabecera = null;

    /**
     * Objeto tipo String con la sentencia SQL a ejecutar.
     */
    static String Sql = null;

    /**
     * Objeto tipo HttpSession que recoge las características de la sesión del usuario.
     */
    static HttpSession Sesion = null;

    /**
     * Almacena NombresCampos.
     * @param AuxNombresCampos Vector NombresCampos.
     */
    public static void setNombresCampos (Vector AuxNombresCampos) { NombresCampos =
    AuxNombresCampos; }

    /**
     * Devuelve NombresCampos.
     * @return Vector NombresCampos.
     */
    public static Vector getNombresCampos() { return NombresCampos; }

    /**
     * Almacena NombresCabecera.
     * @param AuxNombresCabecera Vector NombresCabecera.
     */
    public static void setNombresCabecera (Vector AuxNombresCabecera) { NombresCabecera
    = AuxNombresCabecera; }

    /**
     * Devuelve NombresCabecera.
     * @return Vector NombresCabecera.
     */
    public static Vector getNombresCabecera() { return NombresCabecera; }

    /**
     * Almacena Sql.
     * @param AuxSql String Sql.
     */
}
```

```
    */
    public static void setSql (String AuxSql) { Sql = AuxSql; }
    /**
     * Devuelve Sql.
     * @return String Sql.
     */
    public static String getSql() { return Sql; }

    /**
     * Almacena Sesion.
     * @param AuxSesion HttpSession Sesion.
     */
    public static void setSesion (HttpSession AuxSesion) { Sesion = AuxSesion; }
    /**
     * Devuelve Sesion.
     * @return HttpSession Sesion.
     */
    public static HttpSession getSesion() { return Sesion; }

    /**
     * Borra todos los parámetros para formar el informe PDF
     */
    public static void Clear()
    {
        NombresCampos = null;
        NombresCabecera = null;
        Sql = null;
        Sesion = null;
    }
}
```

9.17. Clase RegistrarAcceso.java.-

```

/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: RegistrarAcceso.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

import ardb.*;
import java.io.*;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.*;

/**
 * Clase que ingresa en tabla "registro_de_accesos" información sobre el acceso
 * de los usuarios a los recursos del Sistema de Información.
 */
public class RegistrarAcceso extends Registro {

    /** Identificación del usuario que accede al recurso. */
    static String IdUsuario;
    /** Identificación del ordenador desde que se accede al recurso. */
    static String IdHost;
    /** Credenciales del usuario que accede al recurso. */
    static String Credenciales;
    /** Código de denegación de acceso a un fichero. */
    static String CódigoDenegado;
    /** Nombre del fichero accedido por el usuario. */
    static String FicheroAccedido;
    /** Propiedades del fichero accedido por el usuario. */
    static String PropiedadesFicheroAccedido;
    /** Identificación del recurso al que accede el usuario. */
    static String IdRecurso;
    /** URL del fichero al que accede el usuario. */
    static String URL;
    /** Fecha de la copia de respaldo del fichero al que accede el usuario. */
    static String FechaCopiaRespaldo;
    /** Tipo de acceso al fichero por medio del usuario. */
    static String TipoAcceso;
    /** Código de acceso con que se accede al fichero. */
    static String CódigoAcceso;
    /** Número de veces que se accede a un fichero. */
    static int NumeroAccesoFichero;
    /** Número de acceso. (Global) */
    static int NumeroAcceso;
    /** Variable que recoge si el acceso del usuario al recurso fue autorizado */
    static boolean Autorizado;
    /** Variable que recoge si el fichero accedido presenta copia de respaldo. */
    static boolean CopiaRespaldo;
    /** Variable que recoge si se despersonalizó el fichero accedido. */
    static boolean Despersonalizacion;

    /**
     * Método que accede a la tabla "registros_de_accesos" e inserta un nuevo
     registro
     * con los valores pasados a las clases ControlAcceso.java,
     * ControlResponsableSeguridad.java y ControlCopiaRespaldo.java.
     * @return String mensaje de confirmación o de negación del registro.
     */
    public static String Registrar()
    {

```

```

        Calendario Hoy = new Calendario();

        NumeroAcceso = ConsultasEstablecidas.NumeroDeRegistro(9)+1;
        IdUsuario = ControlAcceso.getIdUsuario();
        IdHost = ControlAcceso.getIdHost();
        Credenciales = ControlAcceso.getCredenciales();
        ResponsableSeguridad = ControlResponsableSeguridad.getResponsableSeguridad();
        Fecha = Hoy.ObtenerFecha();
        Hora = Hoy.ObtenerHora();
        Autorizado = ControlAcceso.getAutorizado();
       CodigoDenegado = ControlAcceso.getCodigoDenegado();
        FicheroAccedido = ControlAcceso.getFicheroAccedido();
        PropiedadesFicheroAccedido = ControlAcceso.getPropiedadesFicheroAccedido();
        IdRecurso = ControlAcceso.getIdRecurso();
        URL = ControlAcceso.getURL();
        CopiaRespaldo = ControlAcceso.getCopiaRespaldo();
        FechaCopiaRespaldo = ControlCopiaRespaldo.getFechaCopiaRespaldo();
        TipoAcceso = ControlAcceso.getTipoAcceso();
        NumeroAccesoFichero = ConsultasEstablecidas.NumeroDeFichero(FicheroAccedido);
        CogidoAcceso = ControlAcceso.getCodigoAcceso();
        Despersonalizacion = ControlAcceso.getDespersonalizacion();

        Sql = ""
            + " INSERT INTO REGISTRO_DE_ACCESOS VALUES
('"+NumeroAcceso+"', '"+IdUsuario+"', '"+IdHost+"', '"+Credenciales+"', "
            +
            ""+ResponsableSeguridad+"', '"+Fecha+"', '"+Hora+"', '"+Autorizado+"', '"+FicheroAccedido+"
            ', '"+PropiedadesFicheroAccedido+"', "
            + ""+ CodigoDenegado
            +"'', '"+IdRecurso+"', '"+URL+"', '"+CopiaRespaldo+"', '"+TipoAcceso+"', '"+NumeroAccesoFicher
            o+"', "
            + ""+CogidoAcceso+"', '"+Despersonalizacion+"');";

        if (CopiaRespaldo)
        { Sql = Sql + " INSERT INTO REGISTRO_DE_RESPALDO VALUES
('"+NumeroAcceso+"', '"+FechaCopiaRespaldo+"');"; }

        try { conexion = Conexion.EstablecerConexion();
            if (RegistroDeAcceso.ActualizarBaseDeDatos(conexion, Sql))
            { Mensaje = "La información sobre su acceso ha sido registrada en la
base de datos 'Registro de Acceso'"; }
            else { Mensaje = "No se ha podido actualizar la base de datos Registro de
Acceso"; }
        }
        catch (Exception Ex)
        { System.out.println("RegistrarAcceso.java. No se ha podido establecer la
conexión con la base de datos");
            System.out.println(Ex.getMessage());
            Mensaje = "No se ha podido establecer la conexión con la base de
datos"+Ex.getMessage();
        }

        return Mensaje;
    }

}

```

9.18. Clase RegistrarEntradaSoportesInformaticos.java.-

```

/**
*****
                Proyecto final de carrera.
        Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: RegistrarEntradaSoportesInformaticos.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

import ardb.*;
import java.io.*;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.*;
import ars.*;

/**
 * Clase que introduce en la tabla "registro_de_entrada_de_soportes_informaticos"
 * información sobre la entrada de soportes al Sistema de Información.
 */
public class RegistrarEntradaSoportesInformaticos extends Registro {

    /** Código del soporte informático. */
    static String CodigoSoporte;
    /** Emisor de la información. */
    static String Emisor;
    /** Receptor de la información. */
    static String Receptor;
    /** Código de la forma de envío de la información. */
    static String CodigoEnvio;
    /** Número de registro de entrada. */
    static int NumeroEntrada;
    /** Número de soportes recibidos. */
    static int NumeroSoportes;
    /** Variable que recoge si existe autorización escrita para la entrada de
     * información al dominio de seguridad. */
    static boolean AutorizacionEscrita;

    /**
     * Método que accede a la tabla "registro_de_entrada_de_soportes_informaticos"
     * e inserta un nuevo registro con los valores pasados al servidor en la clase
     * ControlSoportesInformaticos.java.
     * @return mensaje de confirmación o de negación del registro.
     */
    public static String Registrar ()
    {
        Calendario Hoy = new Calendario();

        CodigoSoporte = ControlSoportesInformaticos.getCodigoSoporte();
        ResponsableSeguridad = ControlResponsableSeguridad.getResponsableSeguridad();
        Emisor = ControlSoportesInformaticos.getEmisor();
        Receptor = ControlSoportesInformaticos.getReceptor();
        AutorizacionEscrita = ControlSoportesInformaticos.getAutorizacionEscrita();
        Fecha = Hoy.ObtenerFecha();
        Hora = Hoy.ObtenerHora();
        NumeroSoportes = ControlSoportesInformaticos.getNumeroSoportes();
        CodigoEnvio = ControlSoportesInformaticos.getCodigoEnvio();

        NumeroEntrada = ConsultasEstablecidas.NumeroDeRegistro(10)+1;
        Sql = ""
        + " INSERT INTO REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS VALUES
        ('"+NumeroEntrada+"', '"+CodigoSoporte+"', "

```

```
+
""+ResponsableSeguridad+"', '"+Emisor+"', '"+Receptor+"', '"+AutorizacionEscrita+"', '"+Fec
ha+"', "
    + ""'+Hora+"', '"+ NumeroSoportes +""', '"+CodigoEnvio+"')";
    Mensaje = "La información sobre la entrada de soportes ha sido registrada en la
base de datos 'Registro de Acceso'";

    try {    conexion = Conexion.EstablecerConexion();
        if ( RegistroDeAcceso.ActualizarBaseDeDatos(conexion, Sql)) {}
        else { Mensaje = "No se ha podido actualizar la base de datos 'Registro de
Acceso'"; }
    }
    catch (Exception Ex)
    { System.out.println("RegistrarSI.java. No se ha podido establecer la
conexión con la base de datos");
      System.out.println(Ex.getMessage());
      Mensaje ="No se ha podido establecer la conexión con la base de
datos"+Ex.getMessage();
    }
    return Mensaje;
}

}
```

9.19. Clase RegistrarIncidencia.java.-

```

/**
*****
                Proyecto final de carrera.
        Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: RegistrarIncidencia.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

import ardb.*;
import ars.*;
import java.io.*;
import java.util.*;
import java.sql.ResultSet;
import java.sql.SQLException;

/**
 * Clase que ingresa en tabla "registro_de_incidencias" información sobre las
 * incidencias de seguridad ocurridas durante el acceso del usuario al Sistema
 * de Información.
 */
public class RegistrarIncidencia extends Registro {

    /** Código de la incidencia ocurrida. */
    static StringCodigoIncidencia;
    /** Nombre del sistema o módulo notificante de la incidencia. */
    static StringNotificante;
    /** Código de la comunicación de la incidencia a quien corresponda. */
    static StringCodigoComunicacion;
    /** Identificación del usuario cuando se produce una incidencia de recuperación
de datos. */
    static StringIdUsuario;
    /** Datos recuperados automáticamente al producirse una incidencia de
recuperación. */
    static StringDatos;
    /** Datos recuperados manualmente al existir una incidencia de recuperación.
*/
    static StringDatosManuales;
    /** Número de incidencia registrada. (Global) */
    static intNumeroIncidencia;
    /** Variable que recoge si la recuperación de datos que provocó la incidencia
estaba autorizada. */
    static booleanRecuperacionAutorizada;

    /**
     * Método que accede a la tabla "registro_de_incidencias" e inserta un nuevo
     * registro con los valores pasados al servidor en las clases
     ControlIncidencia.java
     * y ControlRecuperacionDatos.java.
     * @return String mensaje de confirmación o de negación del registro.
     */
    public static String Registrar()
    {
        Calendario Hoy = new Calendario();

        NumeroIncidencia = ConsultasEstablecidas.NumeroDeRegistro(11)+1; //
Consultar(1) examina la tabla Registro de Incidencias.
CodigoIncidencia = ControlIncidencia.getCodigoIncidencia();
Fecha = Hoy.ObtenerFecha();
Hora = Hoy.ObtenerHora();
Notificante = ControlIncidencia.getNotificante();
CodigoComunicacion = ControlIncidencia.getCodigoComunicacion();
ResponsableSeguridad = ControlResponsableSeguridad.getResponsableSeguridad();

```

```
IdUsuario = ControlAcceso.getIdUsuario();
Datos = ControlRecuperacionDatos.getDatos();
DatosManuales = ControlRecuperacionDatos.getDatosManuales();
RecuperacionAutorizada = ControlRecuperacionDatos.getRecuperacionAutorizada();

if (CodigoIncidencia.contentEquals("I-600"))
{
    Sql = ""
    + " INSERT INTO REGISTRO_DE_INCIDENCIAS VALUES
    (" + NumeroIncidencia + "','" + CodigoIncidencia + "','"
    + " + Fecha + "','" + Hora + "','" + Notificante + "','" + CodigoComunicacion + "','" + ResponsableSegurid
    ad + "');
    + " INSERT INTO REGISTRO_DE_RECUPERACION_DE_DATOS VALUES
    (" + NumeroIncidencia + "','"
    + " + IdUsuario + "','" + Datos
    + "','" + DatosManuales + "','" + RecuperacionAutorizada + "');
}
else {
    Sql = ""
    + " INSERT INTO REGISTRO_DE_INCIDENCIAS VALUES
    (" + NumeroIncidencia + "','" + CodigoIncidencia + "','"
    + " + Fecha + "','" + Hora + "','" + Notificante + "','" + CodigoComunicacion + "','" + ResponsableSegurid
    ad + "');
}

try {
    conexion = Conexion.EstablecerConexion();
    if (RegistroDeAcceso.ActualizarBaseDeDatos(conexion, Sql))
    {
        Mensaje = ConsultasEstablecidas.TextoCodigoIncidencia();
    }
    else {
        Mensaje = "No se ha podido actualizar la base de datos Registro
de Acceso";
    }
}
catch (Exception Ex)
{
    System.out.println("RegistrarIncidencia.java: No se ha podido
establecer la conexión con la base de datos");
    System.out.println(Ex.getMessage());
    Mensaje = "No se ha podido establecer la conexión con la base de
datos" + Ex.getMessage();
}

return Mensaje;
}
}
```

9.20. Clase RegistrarSalidaSoportesInformaticos.java.-

```

/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: RegistrarSalidaSoportesInformaticos.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

import ardb.*;
import java.io.*;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.*;
import ars.*;

/**
 * Clase que introduce en la tabla "registro_de_salida_de_soportes_informaticos"
 * información sobre la salida de soportes del Sistema de Información.
 */
public class RegistrarSalidaSoportesInformaticos extends Registro {

    /** Código del soporte informático. */
    static String CodigoSoporte;
    /** Emisor de la información. */
    static String Emisor;
    /** Receptor de la información. */
    static String Receptor;
    /** Código de la forma de envío de la información. */
    static String CodigoEnvio;
    /** Número de registro de salida. */
    static int NumeroSalida;
    /** Número de soportes enviados. */
    static int NumeroSoportes;
    /** Variable que recoge si existe autorización escrita para la salida de
     * información del dominio de seguridad. */
    static boolean AutorizacionEscrita;

    /**
     * Método que accede a la tabla "registro_de_salida_de_soportes_informaticos"
     * e inserta un nuevo registro con los valores pasados al servidor en la clase
     * ControlSoportesInformaticos.java.
     * @return mensaje de confirmación o de negación del registro.
     */
    public static String Registrar ()
    {
        Calendario Hoy = new Calendario();

        CodigoSoporte = ControlSoportesInformaticos.getCodigoSoporte();
        ResponsableSeguridad = ControlResponsableSeguridad.getResponsableSeguridad();
        Emisor = ControlSoportesInformaticos.getEmisor();
        Receptor = ControlSoportesInformaticos.getReceptor();
        AutorizacionEscrita = ControlSoportesInformaticos.getAutorizacionEscrita();
        Fecha = Hoy.ObtenerFecha();
        Hora = Hoy.ObtenerHora();
        NumeroSoportes = ControlSoportesInformaticos.getNumeroSoportes();
        CodigoEnvio = ControlSoportesInformaticos.getCodigoEnvio();

        NumeroSalida = ConsultasEstablecidas.NumeroDeRegistro(13)+1;
        Sql = ""
        + " INSERT INTO REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS VALUES
('"+NumeroSalida+"', '"+CodigoSoporte+"', "

```

```
+
""+ResponsableSeguridad+"', '"+Receptor+"', '"+Emisor+"', '"+AutorizacionEscrita+"', '"+Fec
ha+"', "
+ ""'+Hora+"', '"+ NumeroSoportes +' , '"+CodigoEnvio+"')";
Mensaje = "La información sobre la salida de soportes ha sido registrada en la
base de datos 'Registro de Acceso'";

try {   conexion = Conexion.EstablecerConexion();
        if ( RegistroDeAcceso.ActualizarBaseDeDatos(conexion, Sql)) {}
        else { Mensaje = "No se ha podido actualizar la base de datos 'Registro de
Acceso'"; }
    }
    catch (Exception Ex)
    { System.out.println("RegistrarSI.java. No se ha podido establecer la
conexión con la base de datos");
      System.out.println(Ex.getMessage());
      Mensaje ="No se ha podido establecer la conexión con la base de
datos"+Ex.getMessage();
    }
    return Mensaje;
}

}
```

9.21. Clase Registro.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Registro.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;
import java.sql.Connection;

/**
 * Clase que recoge variables comunes para las clases RegistrarAcceso.java,
 * RegistrarIncidencia.java, RegistrarEntradaSoportesInformaticos.java y
 * RegistrarSalidaSoportesInformaticos.java que almacenan en la base de datos
 * "Registro de Acceso" los datos perteneciente al control de seguridad aplicado
 * a los usuarios que acceden al Sistema de Información.
 */
public class Registro
{
    /**
     * Objeto tipo String con la fecha actual en la que se produce un nuevo registro.
     * Formato mm/dd/aaaa.
     */
    static String Fecha;
    /**
     * Objeto tipo String con la hora actual en la que se produce un nuevo registro.
     * Formato hh:mm:ss.
     */
    static String Hora;
    /**
     * Objeto tipo String con el mensaje de confirmación o de negación del nuevo
registro.
     */
    static String Mensaje;
    /**
     * Objeto tipo String con la sentencia SQL a ejecutar.
     */
    static String Sql;
    /**
     * Objeto tipo String con el nombre del Responsable de Seguridad.
     */
    static String ResponsableSeguridad;
    /**
     * Objeto tipo Connection con el que se establece la conexión la base de datos
     * "Registro de Acceso".
     */
    static Connection conexion = null;
}
```

9.22. Clase UsuariosDelServidor.java.-

```

/**
*****
                Proyecto final de carrera.
        Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: UsuariosDelServidor.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package ars;

import java.sql.ResultSet;
import java.util.Vector;

/**
 * Clase utilizada por el Servidor de Registro de Acceso para comprobar los
 * datos y permisos de los usuarios que acceden a él.
 */
public class UsuariosDelServidor {

    /**
     * Método que comprueba si el usuario está registrado para usar el servidor.
     * @param IdUsuario String con la identificación del usuario.
     * @return boolean true = existe, false = desconocido.
     */
    public static boolean ComprobarUsuario (String IdUsuario)
    {
        boolean Respuesta = false;
        // Comprobamos que este registrado en la tabla de usuarios.
        try {
            ResultSet rs = ConsultasEstablecidas.UsuariosDelServidor(1);
            while (rs.next())
            { if (rs.getString(1).contentEquals(IdUsuario))
              { Respuesta = true; }
            }
        }
        catch (Exception Ex)
        { System.out.println("UsuariosDelServidor.java: Error al consultar la base
de datos");
          System.out.println(Ex.getMessage());
        }

        return Respuesta;
    }

    /**
     * Método que comprueba el Login y el Password del usuario del servidor.
     * @param Login String con el identificador del usuario del servidor.
     * @param Password String con la clave del usuario del servidor.
     * @return boolean true = existe, false = desconocido.
     */
    public static boolean ComprobarLoginPassword (String Login, String Password)
    {
        boolean Respuesta = false;
        try {
            ResultSet rs = ConsultasEstablecidas.UsuariosDelServidor(2);
            while (rs.next())
            { if
((rs.getString(1).contentEquals(Login))&(rs.getString(2).contentEquals>Password)))
              { Respuesta = true; }
            }
        }
        catch (Exception Ex)
        { System.out.println("UsuariosDelServidor.java: Error al consultar la base
de datos");
          System.out.println(Ex.getMessage());
        }
    }
}

```

```
        }

        return Respuesta;
    }

    /**
     * Método que comprueba si el usuario del servidor tiene permiso para acceder a
     * un determinado recurso.
     * @param IdUsuario String con la identificación del usuario.
     * @param Recurso int que señala el recurso a acceder.
     * @return boolean con el permiso o la denegación.
     */
    public static boolean ComprobarAcceso (String IdUsuario, int Recurso)
    {
        String Credenciales = null;
        String Recursos = null;
        boolean Respuesta = false;

        try { ResultSet rs = ConsultasEstablecidas.UsuariosDelServidor(3);
            while (rs.next())
            { if (rs.getString(1).contentEquals(IdUsuario)) { Credenciales =
rs.getString(2); } }

                rs = ConsultasEstablecidas.UsuariosDelServidor(4);
                while (rs.next())
                { if (rs.getString(1).contentEquals(Credenciales)) { Recursos =
rs.getString(2); } }
            }

            catch (Exception Ex)
            { System.out.println("UsuariosServidor.java: Error al consultar la base de
datos");
                System.out.println(Ex.getMessage());
            }

            // Comprobamos si las credenciales del usuario lengthpermiten acceder al recurso
solicitado
            Vector RecursosAccesibles = new Vector();
            String ConjuntoDeRecursos[] = Recursos.split(",");

            for (int i=0; i<ConjuntoDeRecursos.length; i++)
            { if (ConjuntoDeRecursos[i].contains("-"))
                { String Conjunto[] = ConjuntoDeRecursos[i].split("-");
                    int Primero = Convertir.String_Int(Conjunto[0]);
                    int Ultimo = Convertir.String_Int(Conjunto[1]);
                    while (Primero<Ultimo)
                    { RecursosAccesibles.addElement(Primero);
                        Primero++;
                    }
                    RecursosAccesibles.addElement(Ultimo);
                }
                else {
RecursosAccesibles.addElement(Convertir.String_Int(ConjuntoDeRecursos[i])); }
            }

            for (int i=0; i<RecursosAccesibles.size(); i++)
            { if (Recurso==Convertir.String_Int(RecursosAccesibles.elementAt(i).toString()))
                { Respuesta = true; }
            }

            return Respuesta;
        }
    }
}
```

9.23. Clase Autorizado.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Autorizado.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package cliente;

/**
 * Clase que almacena la información necesaria para el agente de decisión de acceso.
 */
public class Autorizado {

    /**
     * Campo que recoge la decisión del agente de decisión de acceso.
     */
    static boolean Autorizado = false;

    /**
     * Almacena Autorizado.
     * @param Autorizacion boolean Autorizado.
     */
    public static void setAutorizado(boolean Autorizacion)
    { Autorizado = Autorizacion; }

    /**
     * Devuelve Autorizado.
     * @return boolean Autorizado.
     */
    public static boolean getAutorizado()
    { return Autorizado; }

    /**
     * Borra Autorizado.
     */
    public static void Clear()
    { Autorizado = false; }
}
```

9.24. Clase Credenciales.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Credenciales.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package cliente;

/**
 * Clase que almacena la información necesaria para el agente de obtención de
 * credenciales.
 */
public class Credenciales

{ /**
 * Credenciales del usuario que accede al Sistema de Información.
 */
    static String Credenciales = null;

    /**
 * Almacena Credenciales.
 * @param AuxCredenciales String Credenciales.
 */
    public static void setCredenciales(String AuxCredenciales)
    { Credenciales = AuxCredenciales; }

    /**
 * Devuelve Credenciales.
 * @return String Credenciales.
 */
    public static String getCredenciales()
    { return Credenciales; }

    /**
 * Borra Credenciales.
 */
    public static void Clear()
    { Credenciales = null; }
}
```

9.25. Clase Decision.java.-

```

/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Decision.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package cliente;

/**
 * Clase que simula simplemente las funciones del agente de decisión de acceso.
 * Actualiza el campo "Autorizado" de la clase Autorizado.java si el usuario
 * posee las credenciales necesarias para acceder al recurso seleccionado.
 */
public class Decision {

    /**
     * Campo que almacena si el usuario posee permiso para acceder al recurso.
     */
    static boolean Autorizacion;

    /**
     * Método que comprueba las credenciales de un usuario para acceder a un recurso.
     * @param Credenciales String con las credenciales del usuario.
     * @param IdRecurso String con el nombre del recurso a acceder.
     */
    public static void Acceso (String Credenciales, String IdRecurso)

    {
        Autorizacion = false;

        if (Credenciales.contentEquals("Administrador"))    { Autorizacion = true; }

        if (Credenciales.contentEquals("Jefe de Servicio")
            && (IdRecurso.contentEquals("Base de datos Historial Médico")
            || IdRecurso.contentEquals("Administración del Sistema")) )    { Autorizacion =
true; }

        if (Credenciales.contentEquals("Residentes")
            && (IdRecurso.contentEquals("Base de datos Historial Médico")
            || IdRecurso.contentEquals("Administración del Sistema")) )    { Autorizacion =
true; }

        if (Credenciales.contentEquals("Personal de Enfermería")
            && IdRecurso.contentEquals("Base de datos Historial Médico")) { Autorizacion =
true; }

        if (Credenciales.contentEquals("Auxiliares")
            && IdRecurso.contentEquals("Base de datos Historial Médico")) { Autorizacion =
true; }

        if (Credenciales.contentEquals("Administrativos")
            && IdRecurso.contentEquals("Gestión del Centro"))          { Autorizacion =
true; }

        if (Credenciales.contentEquals("Adjuntos")
            && (IdRecurso.contentEquals("Base de datos Historial Médico")
            || IdRecurso.contentEquals("Gestión del Centro")) )    { Autorizacion = true; }

        if (Credenciales.contentEquals("Facultativos")
            && IdRecurso.contentEquals("Base de datos Historial Médico")) { Autorizacion =
true; }

        Autorizado.setAutorizado(Autorizacion);
    }
}

```

9.26. Clase Despersonalizacion.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Despersonalizacion.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package cliente;

import ars.ControlAcceso;

/**
 * Clase que asocia la situación, en la que se produce el acceso, con la
 * despersonalización de ficheros.
 */
public class Despersonalizacion {

    /**
     * Campo que recoge si se produce la despersonalización de los ficheros accedidos.
     */
    static boolean Despersonalizacion;

    /**
     * Método que recoge la situación, en la que se produce el acceso, y establece
     * si los ficheros consultados fueron despersonalizados.
     * @param Codigo_Acceso String con el código de acceso asociado a la situación de
     * acceso al fichero.
     */
    public static void Acceso (String Codigo_Acceso)
    {
        Despersonalizacion = false;

        if (Codigo_Acceso.contentEquals("A-104")) { Despersonalizacion = true; }
        if (Codigo_Acceso.contentEquals("A-105")) { Despersonalizacion = true; }

        ControlAcceso.setDespersonalizacion(Despersonalizacion);
    }
}
```

9.27. Clase Identificacion.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área:Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Identificacion.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package cliente;

/**
 * Clase que almacena información necesaria para el agente de identificación y
 * autenticación.
 */
public class Identificacion {

    /**
     * Objeto tipo String con la identificación del usuario.
     */
    static String IdUsuario = null;
    /**
     * Objeto tipo String con la clave del usuario.
     */
    static String Password = null;

    /**
     * Almacena IdUsuario.
     * @param AuxIdUsuario String IdUsuario.
     */
    public static void setIdUsuario(String AuxIdUsuario)
    { IdUsuario = AuxIdUsuario; }
    /**
     * Devuelve IdUsuario.
     * @return String IdUsuario.
     */
    public static String getIdUsuario()
    { return IdUsuario; }

    /**
     * Almacena Password.
     * @param AuxPassword String Password.
     */
    public static void setPassword(String AuxPassword)
    { Password = AuxPassword; }
    /**
     * Devuelve Password.
     * @return String Password.
     */
    public static String getPassword()
    { return Password; }

    /**
     * Borra los parámetros de identificación.
     */
    public static void Clear()
    { IdUsuario = null;
      Password = null;
    }
}
```

9.28. Clase Relacion.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Relacion.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package cliente;

import ars.ControlIncidencia;
/**
 * Clase que relaciona los códigos de incidencia con los códigos de comunicación.
 * Cuando se produce una incidencia de seguridad en el sistema, lo habitual es
 * que según el tipo de incidencia generada se informe de una u otra manera.
 */
public class Relacion {

    /**
     * Método que aplica un código de comunicación a un determinado código de
     incidencia.
     * @return String con el código de comunicación.
     */
    public static String Incidencia_Comunicacion ()
    {
        String Codigo_Incidencia = ControlIncidencia.getCodigoIncidencia();
        String Codigo_Comunicacion = "C-102"; // Comunicacion por defecto.
        if (Codigo_Incidencia.contentEquals("I-600")) { Codigo_Comunicacion = "C-103"; }

        return Codigo_Comunicacion;
    }
}
```

9.29. Clase Validar.java.-

```

/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Validar.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package cliente;

import java.io.*;
import java.util.*;

/**
 * Clase que busca en los archivos Usuarios_Registrados.dat y Credenciales.dat los
 * datos de identificación proporcionados por el usuario.
 */
public class Validar {

    /** Identificación del usuario. */
    static String IdUsuario;
    /** Password del usuario. */
    static String Password;

    /**
     * Método que busca en el archivo Usuarios_Registrados.dat el login y el
     * password del usuario que intenta acceder al Sistema de Información.
     * @param Login String con el identificador o login del usuario.
     * @param Password String con la clave o password del usuario.
     */
    public static void Identificacion(String Login, String Password)
    {
        String Auxiliar;
        String AuxNombreCompleto;
        String AuxLogin;
        String AuxPassword;

        System.out.println("Tu identificación es: Usuario=" + Login + " / Password=" +
        Password );

        // Buscamos en el archivo Usuarios_Registrados.dat los datos introducidos.
        try { BufferedReader AuxLeer = new BufferedReader
            (new FileReader("C:\\Servidor de Registro de
            Acceso\\web\\Modulo_Identificacion\\Usuarios_Registrados.dat"));
            while( ( Auxiliar = AuxLeer.readLine() ) != null )
            { StringTokenizer ClaveValor = new StringTokenizer( Auxiliar, "-" );
                // Rompe la cadena de caracteres buscando -
                AuxLogin = (String) ClaveValor.nextToken();
                AuxPassword = (String) ClaveValor.nextToken();
                AuxNombreCompleto = (String) ClaveValor.nextToken();
                if (Login.contentEquals(AuxLogin))
                { if (Password.contentEquals(AuxPassword))
                    { Identificacion.setIdUsuario(AuxNombreCompleto);
                      Identificacion.setPassword(AuxPassword);
                    }
                    else { Identificacion.setIdUsuario(AuxNombreCompleto); }
                }
            }
        } catch (IOException Ex){ System.out.println("No se pudo encontrar el fichero de
        usuarios");}
    }

    /**
     * Método que busca en el archivo Credenciales.dat las credenciales del usuario
     * identificado por el agente de identificación y autenticación.
     */

```

```

    * @param IdUsuario String con la Identificación del usuario.
    */
    public static void Credenciales (String IdUsuario)
    {
        String Auxiliari;
        String AuxCredenciales;
        String AuxIdUsuario;

        try {    BufferedReader AuxLeer = new BufferedReader(
new FileReader( "C:\\Servidor de Registro de
Acceso\\web\\Modulo_Credenciales\\Credenciales.dat" ));
            while( ( Auxiliari = AuxLeer.readLine() ) != null )
            { StringTokenizer ClaveValor = new StringTokenizer( Auxiliari, "-" );
              AuxIdUsuario = (String) ClaveValor.nextToken();
              AuxCredenciales = (String) ClaveValor.nextToken();
              if (IdUsuario.contentEquals(AuxIdUsuario))
              { Credenciales.setCredenciales(AuxCredenciales); }
            }
        }
        catch (IOException Ex){ System.out.println("No se pudo encontrar las credenciales
del usuario");}
    }
}
```

9.30. Clase Recurso_Administración_de_Sistema.java.-

```
/**
*****
                Proyecto final de carrera.
        Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Recurso_Administracion_del_Sistema.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package recurso;

/**
 * Clase que describe el Recurso Administración del Sistema.
 */
public class Recurso_Administracion_del_Sistema {

    /**
     * Método donde se describen los parámetros necesarios para identificar y
    utilizar
     * el recurso.
     * @return ResourceName con las propiedades del recurso.
     */
    public static ResourceName Descripcion()
    {
        ResourceName Recurso_Administracion_del_Sistema = new ResourceName();
        ResourceNameComponentsList ResourceNameComponent = new
        ResourceNameComponentsList ();

        String ResourceNamingAuthority = "Administración del Sistema.";
        String ResourceLink =
        "../Modulo_Decision_de_Acceso/DA_Administracion_del_Sistema.jsp";

        ResourceNameComponent.setTipoRecurso("Servidor de Registro de Acceso para
        Sistemas de Informacion Sanitarios.");
        ResourceNameComponent.setDescripcion("Será el encargado de registrar cualquier
        acceso, incidencia o intercambio de información que se produzca en el Sistema de
        Información Sanitario.");

        Recurso_Administracion_del_Sistema.setResourceName(ResourceNamingAuthority,
        ResourceLink, ResourceNameComponent);
        return Recurso_Administracion_del_Sistema;
    }
}
```

9.31. Clase Recurso_Base_de_Datos_Historial_Medico.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Recurso_Base_de_Datos_Historial_Medico.java, v 1.0, 116 de Mayo de 2005, 12:00 $
*****
*/

package recurso;

/**
 * Clase que describe el Recurso Base de Datos Historial Médico.
 */
public class Recurso_Base_de_Datos_Historial_Medico {

    /**
     * Método donde se describen los parámetros necesarios para identificar y
    utilizar
     * el recurso.
     * @return ResourceName con las propiedades del recurso.
     */
    public static ResourceName Descripcion()
    {
        ResourceName Recurso_Base_de_Datos_Historial_Medico = new ResourceName();
        ResourceNameComponentsList ResourceNameComponent = new
        ResourceNameComponentsList ();

        String ResourceNamingAuthority = "Base de Datos 'Historial Medico'.";
        String ResourceLink =
        "../Modulo_Decision_de_Acceso/DA_Base_de_Datos_Historial_Medico.jsp";

        ResourceNameComponent.setTipoRecurso("Base de Datos.");
        ResourceNameComponent.setDescripcion("Base de Datos donde se almacena todos los
        documentos del Historial Médico de los pacientes atendidos en el centro.");

        Recurso_Base_de_Datos_Historial_Medico.setResourceName(ResourceNamingAuthority,
        ResourceLink, ResourceNameComponent);

        return Recurso_Base_de_Datos_Historial_Medico;
    }
}
```

9.32. Clase Recurso_Gestion_del_Centro.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: Recurso_Gestion_del_Centro.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package recurso;

/**
 * Clase que describe el Recurso Gestión del Centro.
 */
public class Recurso_Gestion_del_Centro {

    /**
     * Método donde se describen los parámetros necesarios para identificar y
    utilizar
     * el recurso.
     * @return ResourceName con las propiedades del recurso.
     */
    public static ResourceName Descripcion()
    {
        ResourceName Recurso_Gestion_del_Centro = new ResourceName();
        ResourceNameComponentsList ResourceNameComponent = new
        ResourceNameComponentsList ();

        String ResourceNamingAuthority = "Gestión del Centro.";
        String ResourceLink = "../Modulo_Decision_de_Acceso/DA_Gestion_del_Centro.jsp";

        ResourceNameComponent.setTipoRecurso("Servidor de Gestion.");
        ResourceNameComponent.setDescripcion("Servidor encargado de gestionar la
    documentación del centro.");

        Recurso_Gestion_del_Centro.setResourceName(ResourceNamingAuthority,
        ResourceLink, ResourceNameComponent);

        return Recurso_Gestion_del_Centro;
    }
}
```

9.33. Clase RecursosRegistrados.java.-

```
/**
*****
                Proyecto final de carrera.
    Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: RecursosRegistrados.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package recurso;

/**
 * Clase que presenta una clasificación de los recursos del Sistema de Información.
 * Cada recurso es identificado por un entero. Para seleccionar la descripción de
 * alguno de los recursos se pasa como argumento el entero que lo identifica. La
 * asignación es la siguiente:
 * 1- Administración del Sistema.
 * 2- Base de Datos Historial Médico.
 * 3- Gestión del Centro.
 * El método devuelve un objeto de tipo "ResourceName" que caracteriza al recurso.
 */
public class RecursosRegistrados {

    /** Número de recursos registrados. */
    public static int Numero_de_Recurso = 3;

    /**
     * Método que devuelve las propiedades del recurso seleccionado por el argumento.
     * @param Opcion int que representa el recurso seleccionado.
     * @return ResourceName con la descripción del recurso.
     */
    public static ResourceName Eleccion (int Opcion)
    {
        ResourceName Recurso = new ResourceName();
        switch (Opcion)
        {
            case 1: Recurso = Recurso_Administracion_del_Sistema.Descripcion(); break;
            case 2: Recurso = Recurso_Base_de_Datos_Historial_Medico.Descripcion(); break;
            case 3: Recurso = Recurso_Gestion_del_Centro.Descripcion(); break;
        }

        return Recurso;
    }
}
```

9.34. Clase ResourceName.java.-

```

/**
*****
                Proyecto final de carrera.
        Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ResourceName.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package recurso;

/**
 * La norma Resource Access Decision (RAD) del proyecto CORBAMED señala que el
 * recurso se describe utilizando un ResourceName o "nombre de recurso" formado
 * por un identificador único de la autoridad de nombrado (ResourceNamingAuthority)
 * y una secuencia de componentes (ResourceNameComponent), donde cada componente
 * esta formado por una pareja nombre-valor.
 * Esta clase forma objetos tipo ResourceName que describen los recursos del
 * Sistema de Información.
 */
public class ResourceName {

    /** Identificador único de la autoridad de nombrado. */
    String ResourceNamingAuthority;
    /** Link, si existe, de la página principal del recurso. */
    String ResourceLink;
    /** Secuencia de componentes del recurso. */
    ResourceNameComponentsList ResourceNameComponent;

    /**
     * Constructor vacío.
     */
    public ResourceName () {}

    /**
     * Almacena ResourceNamingAuthority, ResourceLink y ResourceNameComponent.
     * @param AuxResourceNamingAuthority String ResourceNamingAuthority.
     * @param AuxResourceLink String ResourceLink.
     * @param AuxResourceNameComponent ResourceNameComponentsList ResourceNameComponent.
     */
    public void setResourceName ( String AuxResourceNamingAuthority,
                                String AuxResourceLink,
                                ResourceNameComponentsList AuxResourceNameComponent)
    {
        ResourceNamingAuthority = AuxResourceNamingAuthority;
        ResourceLink = AuxResourceLink;
        ResourceNameComponent = AuxResourceNameComponent;
    }

    /**
     * Devuelve ResourceNamingAuthority
     * @return String ResourceNamingAuthority.
     */
    public String getResourceNamingAuthority ()
    {
        return ResourceNamingAuthority;
    }

    /**
     * Devuelve ResourceLink.
     * @return String ResourceLink.
     */
    public String getResourceLink ()
    {
        return ResourceLink;
    }
}

```

```
    }  
  
    /**  
     * Devuelve ResourceNameComponent.  
     * @return ResourceNameComponentsList ResourceNameComponent.  
     */  
    public ResourceNameComponentsList getResourceNameComponent ()  
    {  
        return ResourceNameComponent;  
    }  
}
```

9.35. Clase ResourceNameComponentsList.java.-

```
/**
*****
                Proyecto final de carrera.
        Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
*****
Autor: Daniel David García Malmierca.
Área: Ingeniería Telemática.
Departamento: Ingeniería de Sistemas y Automática.
Tutora: Isabel Román Martínez.
Universidad de Sevilla.
*****
$Id: ResourceNameComponentsList.java, v 1.0, 16 de Mayo de 2005, 12:00 $
*****
*/

package recurso;

/**
 * Clase que almacena la secuencia de componentes que caracterizan al recurso.
 */
public class ResourceNameComponentsList
{
    /**
     * Tipo del recurso.
     */
    String TipoRecurso = null;
    /**
     * Breve descripción del recurso.
     */
    String Descripción = null;

    /**
     * Almacena TipoRecurso.
     * @param AuxTipoRecurso String TipoRecurso.
     */
    public void setTipoRecurso(String AuxTipoRecurso)
    { TipoRecurso = AuxTipoRecurso; }
    /**
     * Devuelve TipoRecurso.
     * @return String TipoRecurso.
     */
    public String getTipoRecurso()
    { return TipoRecurso; }

    /**
     * Almacena Descripción.
     * @param AuxDescripción String Descripción.
     */
    public void setDescription(String AuxDescripción)
    { Descripción = AuxDescripción; }
    /**
     * Devuelve Descripción.
     * @return String Descripción.
     */
    public String getDescripción()
    { return Descripción; }
}
```

9.36. index.html.-

```
<html>
<head>
  <title>Proyecto final de carrera</title>
</head>

<body>
  <center>
    <p></p>
    <p></p>
    <p></p>

    <font face="Batang" size="2" color="#666666">
      <p><b>Proyecto final de carrera</b></p>
      <p><b>SERVIDOR DE REGISTRO DE ACCESO PARA SISTEMAS DE INFORMACIÓN
SANITARIOS</b></p>
      <br>
      <br>
      <br>
    </font>

    <hr size="1" color="#000000">
    <b>| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
bienvenidos</a> |
    </b>
    <hr size="1" color="#000000">
  </center>
  <br>
  <br>
  <br>
  <font face="Batang" size="2" color="#666666">
    <b>Daniel David García Malmierca</b><br><br>
    <b>Junio, 2005</b>
  </font>
</body>
</html>
```

9.37. Bienvenido.jsp.-

```

/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Bienvenido.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* JSP que da la bienvenida al Servidor de Registro de Acceso. Controla que la
* base de datos "Registro de Acceso" haya sido inicializada, creando las
* tablas necesarias antes de cualquier uso. En el caso que no se haya creado
* la base de datos presentará un mensaje en pantalla.
* =====
*/

%>
<%@page import = "ardb.*, ars.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlResponsableSeguridad.setResponsableSeguridad("Universidad de Sevilla"); %>

<!-- Titulo página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" align=center>
<font class="Estilo_2">
<br>
<b><u> Proyecto final de carrera. </u></b><br><br>
Servidor de Registro de Acceso para Sistemas de Información Sanitarios.<br><br>
Autor: Daniel David García Malmierca.
</font>
</td>
</tr>
</table>
<!-- Mensajes ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" align=center>
<br>
<% if (RegistroDeAcceso.ComprobarBaseDeDatos())
{ %>
<font class="Estilo_7">
La base de datos "Registro de Acceso" ha sido detectada
correctamente.<br><br>
<a href=
"http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identificaci
on.jsp"
title="Bienvenido al Servidor de Registro de Acceso">Entrar
</a>
</font>
<% }
else {
try {
RegistroDeAcceso.CrearTablasBaseDeDatos();
Informacion.Inicial(); %>
<font class="Estilo_7">
Las tablas de la base de datos "Registro de Acceso" han sido
inicializadas.<br><br>
<a
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">Comienza a
usar el servidor</a>
</font>
<% }
catch (Exception Error)
{ %>
<font class="Estilo_7">
Ha habido un error al crear las tablas de la base de datos "Registro
de Acceso".<br>
Póngase en contacto con el administrador del sistema.
</font>

```

```
<% }
    } %>
</td>
</tr>
</table>
<!-- Opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">&nbsp;</td>
  </tr>
</table>
<!-- Descripcion ===== -->
<table width="100%">
  <tr>
    <td width="25%"></td>
    <td width="50%">
      <blockquote class="Estilo_8">
        Las páginas que se presentan a continuación simulan el acceso de usuarios
        a un Sistema de Información Sanitario. Se han simulado simplemente
        los agentes encargados de identificación y autenticación,
        de obtención de credenciales y de decisión de acceso.
      </blockquote>
    </td>
    <td width="25%"></td>
  </tr>
</table>
<br>
<!-- Camino ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <b>| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
      Bienvenida al cliente.</a> |
    </b>
    </td>
  </tr>
</table>
<% // Cada vez que entremos en esta página se inicializará el Control de Acceso.
ControlAcceso.Clear(); %>
<%@include file="/Comunes/Pie.jsp" %>
```

9.38. Identificacion.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Identificacion.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* JSP que forma parte del módulo que simula el agente de identificación y
* autenticación. Este JSP pide el login y la clave del usuario y la pasa al
* fichero Validar_Identificacion.jsp.
* =====
*/
%>

<%@include file="../Comunes/Encabezado.jsp" %>
<!-- Título página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Agente de Identificación y Autenticación</u></b>
      </p>
    </td>
  </tr>
</table>
<!-- Mensajes ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <font class="Estilo_7">
      <p align="center"><b>Identifíquese para acceder a los recursos.</b></p>
    </td>
  </tr>
</table>
<!-- Opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <center>
      <form method="post"
action="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Valid
ar_Identificacion.jsp">
        <table class="Estilo_6">
          <tr></tr>
          <tr>
            <td>Login: </td>
            <td>
              <input type="text" name="IdUsuario" maxlength="20" style="font-
family: Arial Narrow" size="15">
            </td>
          </tr>
          <tr>
            <td>Password: </td>
            <td>
              <input type="password" name="Password" maxlength="9" style="font-
family: Arial Narrow" size="15">
            </td>
          </tr>
          <tr>
            <td></td>
            <td align="right">
              <input type="submit" VALUE="Validar" style="font-family: Arial
Narrow">
            </td>
          </tr>
        </table>
      </center>
    </td>
  </tr>
</table>

```

```

        </tr>
      </table>
    </form>
  </center>
</td>
</tr>
</table>
<!-- Descripcion ===== -->
<table width="100%">
  <tr>
    <td width="25%"></td>
    <td width="50%">
      <blockquote class="Estilo_8">
        Agente de Identificación y Autenticación:<br>
        Simula al agente encargado de identificar y autenticar al usuario
        cuando accede al Sistema de Información.
      </blockquote>
    </td>
    <td width="25%"></td>
  </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%" class="Estilo_6">
      <b>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
        Bienvenida al cliente. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
        Identificación & Autenticación. </a> |
      </b>
    </td>
  </tr>
</table>

<% // Cada vez que entremos en esta página se inicializará el Control de Acceso.
ControlAcceso.Clear(); %>
<%@include file="/Comunes/Pie.jsp" %>

```

9.39. Usuarios_Registrados.dat.-

```
jose maria-00000000A-José María García Sánchez
maria-11111111B-María José Núñez González
hector-22222222C-Héctor Moisés García Malmierca
diana-33333333D-Diana Carretero Malmierca
jose-44444444E-José Antonio Núñez Santos
salvador-55555555F-Salvador Alcón Paniagua
marimar-66666666G-María del Mar Izquierdo Valerio
alicia-77777777H-Alicia Paredes Márquez
roberto-88888888I-Roberto Morato Toro
julia-99999999J-Julia Perera López
jesus-10101010K-Jesús González Pérez
carlos-02020202L-Carlos Escudero
```

9.40. Validar_Identificacion.jsp.-

```
<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Validar_Identificacion.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* JSP que forma parte del módulo que simula el agente de identificación y
* autenticación.
* Este JSP recibe un identificador de usuario y una clave y lo compara con las
* existentes en el archivo Usuarios_Registrados.dat informando al servidor del
* resultado de la búsqueda si no ha habido éxito. Si el usuario es
* identificado se mostrará por pantalla el nombre completo del usuario.
* =====
*/ %>

<%@page import="java.util.*, java.io.*, ars.*, cliente.*, java.sql.*" %>
<%@include file="../Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Agente de Identificación y Autenticación"); %>

<!-- Titulo página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Agente de Identificación y Autenticación</u></b>
</p>
</td>
</tr>
</table>
<!-- Mensajes ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% try { Validar.Identificacion(request.getParameter("IdUsuario"),
request.getParameter("Password"));
if (Identificacion.getIdUsuario()==null) // El usuario no ha sido
identificado.
{ ControlIncidencia.setCodigoIncidencia("I-103");
ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion()); %>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
```

```

        </p>
        <% }
        else if (Identificacion.getPassword()==null) // La clave introducida
es incorrecta.
            { ControlIncidencia.setCodigoIncidencia("I-101");
ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
            <br>
            <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
            </p>
            <% }

            else { ControlAcceso.setIdUsuario(Identificacion.getIdUsuario());
ControlAcceso.setIdHost(request.getRemoteHost().toString());%>
            <center>
            <br>
            <p><h3>Bienvenido/a</h3></p>
            <p><h3 class="Estilo_4"><%= ControlAcceso.getIdUsuario()
%></h3></p>
            <a class="Estilo_3"
HREF='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp'>
            Obtenga sus credenciales</a>
            </center>
            <% }
            }
        catch (Exception Ex)
        { ControlIncidencia.setCodigoIncidencia("I-103");
ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
            <br>
            <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
            </p>
            <% } %>
        </font>
    </td>
</tr>
</table>
<!-- Opciones ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%"></td>
    </tr>
</table>
<!-- Descripcion ===== -->
<table width="100%">
    <tr>
        <td width="25%"></td>
        <td width="50%">
            <blockquote class="Estilo_8">
                Agente de Identificación y Autentificación:<br>
                Simula al agente encargado de identificar y autentificar al usuario
                cuando accede al Sistema de Información.
            </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
            <b>

```

```
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
  Bienvenida al cliente. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
  Identificación & Autenticación. </a> |
</b>
</td>
</tr>
</table>

<% Identificacion.Clear(); // Borramos los parámetros de identificación para posteriores
accesos. %>
<%@include file="/Comunes/Pie.jsp" %>
```

9.41. Credenciales.dat.-

José María García Sánchez-Administrador
 María José Núñez González-Jefe de Servicio
 Héctor Moisés García Malmierca-Residentes
 Diana Carretero Malmierca-Residentes
 José Antonio Núñez Santos-Residentes
 Salvador Alcón Paniagua-Personal de Enfermería
 María del Mar Izquierdo Valerio-Auxiliares
 Alicia Paredes Márquez-Administrativos
 Roberto Morato Toro-Adjuntos
 Julia Perera López-Paciente
 Jesús González Pérez-Facultativos

9.42. Obtencion_Credenciales.jsp.-

```
<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Obtencion_Credenciales.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* JSP que simula el agente de obtención de credenciales.
* Según la identificación del usuario obtendremos unas credenciales que
* especifican las características del usuario en el Sistema.
* =====
*/
%>

<@page import="java.util.*, java.io.*, ars.*, cliente.*, java.sql.*" %>
<@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Agente de obtención de credenciales"); %>

<!-- Titulo página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Agente de obtención de credenciales</u></b>
</p>
</td>
</tr>
</table>
<!-- Mensajes ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if (ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align="center" class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<% }
else { Validar.Credenciales( ControlAcceso.getIdUsuario());
if (Credenciales.getCredenciales()== null)
{ ControlIncidencia.setCodigoIncidencia("I-102");
```

```

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
    <br>
    <p align=center class="Estilo_3">
    <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
    Se ha informado al Servidor de Registro de Acceso.<br><br>
    <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
    </p>
    <% }
    else { ControlAcceso.setCredenciales(Credenciales.getCredenciales());
%>
    <br>
    <center>
    <h3>Sus credenciales son de:<b class="Estilo_4"> <%=
ControlAcceso.getCredenciales() %></b></h3>
    <p>
    <a class="Estilo_3"
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp'>
    Acceso a Recursos</a>
    </p>
    </center>
    <% }
    } %>
    </font>
</td>
</tr>
</table>
<!-- Opciones ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%">
        </td>
    </tr>
</table>
<!-- Descripcion ===== -->
<table width="100%">
    <tr>
        <td width="25%"></td>
        <td width="50%">
        <blockquote class="Estilo_8">
        Agente de obtención de credenciales:<br>
        Simula al agente encargado de obtener las credenciales del usuario
        tras ser autenticado. Las credenciales son conjuntos de atributos
        que especifican las características del usuario en el Sistema.
        </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
        <b>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
        Bienvenida al cliente. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
        Identificación & Autenticación. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
        Obtención de Credenciales. </a> |
        </b>
        </td>
    </tr>
</table>

<% Credenciales.Clear(); // Borraremos las credenciales para posteriores accesos. %>
<%@include file="/Comunes/Pie.jsp" %>

```

9.43. Eleccion_de_Recursos.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Eleccion_de_Recursos.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP simula el agente de decisión de acceso a recursos.
* En este JSP se muestran los distintos recursos del Sistema de Información simulado.
* =====
*/
%>

<@page import="java.util.*, cliente.*, java.io.*, ars.*, java.sql.*, recurso.*" %>
<@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Agente de decisión de acceso a recursos"); %>

<!-- Titulo página ===== -->


|                                                                                                                |
|----------------------------------------------------------------------------------------------------------------|
| <br><p align="center" class="Estilo_2"> <b>&lt;u&gt;Agente de decisión de acceso a recursos&lt;/u&gt;</b> </p> |
|----------------------------------------------------------------------------------------------------------------|


</table>
<!-- Mensajes y opciones ===== -->


|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <font class="Estilo_7"> <% if (ControlAcceso.getIdUsuario()==null) { ControlIncidencia.setCodigoIncidencia("I-103");  ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%><br><p align="center" class="Estilo_3"> <b>ACCESO NO PERMITIDO: &lt;br&gt;&lt;br&gt;&lt;br&gt;&lt;%= RegistrarIncidencia.Registrar()</b> </p> %><br> <p align="center"> Se ha informado al Servidor de Registro de Acceso.&lt;br&gt;&lt;br&gt; <a &gt;volver&lt;="" <="" a&gt;="" href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identificacion.jsp" p=""> &lt;/p&gt; &lt;br&gt; &lt;%} else { %&gt; &lt;br&gt; &lt;center&gt;Elija una opción: &lt;br&gt; &lt;br&gt; &lt;br&gt; &lt;@include file="Identificacion_de_Recursos.jsp"%&gt; &lt;% } %&gt; &lt;/font&gt; </a></p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|


</table>
<br>
<!-- Descripción ===== -->


|                                                                                                                                                 |                                                                                                                     |                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| <td width="25%">&lt;/td&gt; <td width="50%"> &lt;blockquote class="Estilo_8"&gt; Agente de decisión de acceso a recursos:&lt;br&gt; </td> </td> | </td> <td width="50%"> &lt;blockquote class="Estilo_8"&gt; Agente de decisión de acceso a recursos:&lt;br&gt; </td> | <blockquote class="Estilo_8"> Agente de decisión de acceso a recursos:<br> |
|-------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|


```

```
        Simula al agente encargado de determinar si un usuario puede realizar o
        no determinado acceso a un recurso.
    </blockquote>
</td>
<td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" class="Estilo_6">
<b>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
Bienvenida al cliente. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
Identificación & Autenticación. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
Obtención de Credenciales. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
Decisión de Acceso. </a> |
</b>
</td>
</tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>
```

9.44. Identificacion_de_Recursos.jsp.-

```
<% /* JSP, incluido en Elección_de_Recursos.jsp, que recoge las propiedades de
 * los recursos del Sistema de Información simulado. */ %>

<% ResourceName AuxResourceName = new ResourceName();
// Creamos objetos ResourceName() y Recursos.%>
<table>
<ol>
<% for (int i=1; i<RecursosRegistrados.Numero_de_Recursos+1; i++)
    { AuxResourceName = RecursosRegistrados.Eleccion(i);    %>
      <tr>
        <td>
          <a class="Estilo_3" href='<%= AuxResourceName.getResourceLink()%>'>
            Recurso: <%= AuxResourceName.getResourceNamingAuthority()%>
          </a>
        </td>
      </tr>
    <% } %>
</ol>
</table>
```

9.45. DA_Base_de_Datos_Historial_Medico.jsp.-

```
<%
/* =====
 * Proyecto final de carrera.
 * Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
 * Autor: Daniel David García Malmierca.
 * Área: Ingeniería Telemática.
 * Departamento: Ingeniería de Sistemas y Automática.
 * Tutora: Isabel Román Martínez.
 * Ingeniería de Telecomunicación - Universidad de Sevilla.
 * =====
 * $Id: DA_Base_de_Datos_Historial_Medico.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
 * =====
 * Comentarios.-
 * Este JSP presenta el fallo del agente de decisión de acceso sobre el acceso al
 * recurso base de datos Historial Médico.
 * =====
 */
%>
<%@page import="java.util.*, java.io.*, ars.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Agente de decisión de acceso a recursos"); %>

<!-- Título página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Agente de decisión de acceso a recursos</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if ( ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
        <br>
        <p align="center" class="Estilo_3">
          <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
          Se ha informado al Servidor de Registro de Acceso.<br><br>
```

```

        <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
    </p>
    <br>
    <% }
    else { ControlAcceso.setIdRecurso("Base de datos Historial Médico");
    Decision.Acceso(ControlAcceso.getCredenciales(),
ControlAcceso.getIdRecurso());
    if (Autorizado.getAutorizado())
    { ControlAcceso.setAutorizado(true); %>
    <%@include file="Acceso_Autorizado.jsp"%>
    <p align="center">
    <a
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Control_de_Acceso/Recu
rso_Base_de_Datos_Historial_Medico.jsp">
    Acceso al recuso</a>
    </p>
    <% }
    else { ControlIncidencia.setCodigoIncidencia("I-104");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
    <br>
    <p align=center class="Estilo_3">
    <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
    Se ha informado al Servidor de Registro de Acceso.<br><br>
    </p>
    <br>
    <% }
    } %>
    </font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
<tr>
<td width="25%"></td>
<td width="50%">
<blockquote class="Estilo_8">
Agente de decisión de acceso a recursos:<br>
Simula al agente encargado de determinar si un usuario puede realizar o
no determininado acceso a un recurso.
</blockquote>
</td>
<td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" class="Estilo_6">
<b>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
Bienvenida al cliente. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
Identificación & Autentificación. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
Obtención de Credenciales. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
Decisión de Acceso. </a> |
</b>
</td>
</tr>
</table>
<% Autorizado.Clear(); %>
<%@include file="/Comunes/Pie.jsp" %>

```

9.46. DA_Administracion_del_Sistema.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: DA_Administracion_del_Sistema.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP presenta el fallo del agente de decisión de acceso sobre el acceso al
* recurso Administración del Sistema.
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Agente de decisión de acceso a recursos"); %>

<!-- Titulo página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Agente de decisión de acceso a recursos</u></b>
</p>
</td>
</tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if ( ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<br>
<% }

else { ControlAcceso.setIdRecurso("Administración del Sistema");
Decision.Acceso(ControlAcceso.getCredenciales(),
ControlAcceso.getIdRecurso());
if
((Autorizado.getAutorizado())&(UsuariosDelServidor.ComprobarUsuario(ControlAcceso.getIdU
suario()))))
{ ControlAcceso.setAutorizado(true); %>
<%@include file="Acceso_Autorizado.jsp"%>
<p align="center">
<a
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
Acceso al recuso</a>
</p>
<% }
else { if (Autorizado.getAutorizado())
{ ControlIncidencia.setCodigoIncidencia("I-106");

```

```

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());
    }
    else { ControlIncidencia.setCodigoIncidencia("I-104");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());
    }%>
    <br>
    <p align=center class="Estilo_3">
    <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
    Se ha informado al Servidor de Registro de Acceso.
    </p>
    <br>
    <% }
    } %>

    </font>
  </td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
  <tr>
    <td width="25%"></td>
    <td width="50%">
      <blockquote class="Estilo_8">
        Agente de decisión de acceso a recursos:<br>
        Simula al agente encargado de determinar si un usuario puede realizar o
        no determinado acceso a un recurso.
      </blockquote>
    </td>
    <td width="25%"></td>
  </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%" class="Estilo_6">
      <b>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
        Bienvenida al cliente. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
        Identificación & Autentificación. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
        Obtención de Credenciales. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
        Decisión de Acceso. </a> |
      </b>
    </td>
  </tr>
</table>
<% Autorizado.Clear(); %>
<%@include file="/Comunes/Pie.jsp" %>

```

9.47. DA_Gestion_de_Centro.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: DA_Gestion_del_Centro.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP presenta la decisión tomada por el módulo Decisión de Acceso sobre el
* acceso al recurso Gestión del Centro.
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Agente de decisión de acceso a recursos"); %>

<!-- Titulo página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Agente de decisión de acceso a recursos</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if ( ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
          <br>
          <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
          </p>
          <br>
        <% }

        else { ControlAcceso.setIdRecurso("Gestión del Centro");
          Decision.Acceso(ControlAcceso.getCredenciales(),
ControlAcceso.getIdRecurso());
          if (Autorizado.getAutorizado())
          { ControlAcceso.setAutorizado(true); %>
            <%@include file="Acceso_Autorizado.jsp"%>
            <p align="center">
              <a
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Control_de_Acceso/Recu
rso_Gestion_del_Centro.jsp">
                Acceso al recuso</a>
            </p>
          <% }
          else { ControlIncidencia.setCodigoIncidencia("I-104");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
            <br>
            <p align=center class="Estilo_3">

```

```

                <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
                Se ha informado al Servidor de Registro de Acceso.<br><br>
                </p>
                <br>
                <% }
                } %>
            </font>
        </td>
    </tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
    <tr>
        <td width="25%"></td>
        <td width="50%">
            <blockquote class="Estilo_8">
                Agente de decisión de acceso a recursos:<br>
                Simula al agente encargado de determinar si un usuario puede realizar o
                no determinado acceso a un recurso.
            </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
            <b>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
                Bienvenida al cliente. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
                Identificación & Autentificación. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
                Obtención de Credenciales. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
                Decisión de Acceso. </a> |
            </b>
        </td>
    </tr>
</table>
<% Autorizado.Clear(); %>
<%@include file="/Comunes/Pie.jsp" %>

```

9.48. Acceso_Autorizado.jsp.-

```

<center>
<table class="Estilo_4" border="0">
  <tr><td align="right">Usuario: </td><td><b><%= ControlAcceso.getIdUsuario()
%></b></td></tr>
  <tr><td align="right">Credenciales del usuario: </td><td><b> <%=
ControlAcceso.getCredenciales() %></b></td></tr>
  <tr><td align="right">Identificación del recurso: </td><td><b> <%=
ControlAcceso.getIdRecurso() %></b></td></tr>
</table>
<p class="Estilo_3"><b>ACCESO AUTORIZADO</b></p>
</center>

```

9.49. Recurso_Base_de_Datos_Historial_Medico.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Recurso_Base_de_Datos_Historial_Medico.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP presenta la página de inicio del recurso Base de Datos Historial Médico.
* =====
*/
%>

<@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Agente de control de acceso"); %>
<!-- Titulo página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Agente de control de acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
        <br>
        <p align="center" class="Estilo_3">
          <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
          Se ha informado al Servidor de Registro de Acceso.<br><br>
          <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
        </p>
        <br>
        <% }
        else { ControlAcceso.setFicheroAccedido("paciente.doc");
          ControlAcceso.setURL(request.getServletPath());

          // Aquí debe haber un sistema que detecte si existe una copia de
respaldo

```

```

// o no y devuelva true o false y la fecha de esta.
ControlAcceso.setCopiaRespaldo(true);
ControlCopiaRespaldo.setFechaCopiaRespaldo("03/28/1999");
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setTipoAcceso("Lectura");
// Aquí debe haber un sistema que detecte la situación del
// usuario cuando accede al sistema.
ControlAcceso.setCodigoAcceso("A-100");
ControlAcceso.setDespersonalizacion(false);%>
<center>
<h2>BASE DE DATOS "HISTORIAL MEDICO"</h2><br>
<b><%= new java.util.Date() %></b><br><br>
<p class="Estilo_3"><%= RegistrarAcceso.Registrar() %></p><br>
<p class="Estilo_4"><b> Fichero de propiedades del
recurso:</b><br><br>
<table class="Estilo_4">
<tr><td><b>Tipo de Recurso: </b><%=
Recurso_Base_de_Datos_Historial_Medico.Descripcion().getResourceNameComponent().getTipoR
ecurso() %><br></td></tr>
<tr><td><b>Descripcion: </b><%=
Recurso_Base_de_Datos_Historial_Medico.Descripcion().getResourceNameComponent().getDescr
ipcion()%></td></tr>
</table>
</center>
<% } %>
</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
<tr>
<td width="25%"></td>
<td width="50%">
<blockquote class="Estilo_8">
Agente de control de acceso:<br>
Simula al agente que controla el acceso al recurso, informando al
Servidor de Registro de Acceso de las decisiones tomadas por el usuario.
</blockquote>
</td>
<td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" class="Estilo_6">
<b>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
Bienvenida al cliente. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
Identificación & Autenticación. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
Obtención de Credenciales. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
Decisión de Acceso. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
Cerrar sesión. </a> |
</b>
</td>
</tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.50. Recurso_Gestion_de_Centro.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Recurso_Gestion_del_Centro.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP presenta la página de inicio del recurso Gestión del Centro.
* =====
*/
%>

<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Agente de control de acceso"); %>
<!-- Titulo página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Agente de control de acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
          <br>
          <p align="center" class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
          </p>
          <br>
          <% }

          else { ControlAcceso.setFicheroAccedido("cuentas.xml");
ControlAcceso.setURL(request.getServletPath());

          // Aquí debe haber un sistema que detecte si existe una copia de
respaldo

          // o no y devuelva true o false y la fecha de esta.
ControlAcceso.setCopiaRespaldo(false);
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setTipoAcceso("Escritura");
          // Aquí debe haber un sistema que detecte la situación del
          // usuario cuando accede al sistema.
ControlAcceso.setCodigoAcceso("A-100");
ControlAcceso.setDespersonalizacion(false);%>
          <center>
          <h2>GESTION DEL CENTRO</h2><br>
          <b><%= new java.util.Date() %></b><br><br>
          <p class="Estilo_3"><%= RegistrarAcceso.Registrar() %></p><br>
          <p class="Estilo_4"><b> Fichero de propiedades del
recurso:</b><br><br>

```

```

        <table class="Estilo_4">
            <tr><td><b>Tipo de Recurso: </b><%=
Recurso_Gestion_del_Centro.Descripcion().getResourceNameComponent().getTipoRecurso()
%><br></td></tr>
            <tr><td><b>Descripcion: </b><%=
Recurso_Gestion_del_Centro.Descripcion().getResourceNameComponent().getDescripcion()%></
td></tr>
        </table>
    </center>
    <% } %>
</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
    <tr>
        <td width="25%"></td>
        <td width="50%">
            <blockquote class="Estilo_8">
                Agente de control de acceso:<br>
                Simula al agente que controla el acceso al recurso, informando al
                Servidor de Registro de Acceso de las decisiones tomadas por el usuario.
            </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
            <b>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
                Bienvenida al cliente. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
                Identificación & Autentificación. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
                Obtención de Credenciales. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
                Decisión de Acceso. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
                Cerrar sesión. </a> |
            </b>
        </td>
    </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.51. Recurso_Administracion_del_Sistema.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Recurso_Administracion_del_Sistema.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP presenta la página de inicio del recurso Administración del Sistema.
* =====
*/
%>
<@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Servidor de Registro de Acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
        <br>
        <p align="center" class="Estilo_3">
          <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
          Se ha informado al Servidor de Registro de Acceso.<br><br>
          <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
        </p>
        <br>
        <% }
        else { %>
          <center>
            <h2>ADMINISTRACIÓN DEL SISTEMA</h2>
            <p><b><%= new java.util.Date() %></b></p>
            <p>Elija entre las siguientes opciones:</p>
            <table>
              <ol>
                <tr>
                  <td><a class="Estilo_3"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Menu_Consultas.
jsp">
                    Consultar la base de datos "Registro de Acceso".</a>
                </td>
              </tr>
              <tr>
                <td><a class="Estilo_3"
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Simulaciones/Menu_Simulac
iones.jsp'>
                    Simulaciones.</a>
                </td>
              </tr>
            </table>
          </center>
        <% }
      </font>
    </td>
  </tr>
</table>

```

```

        <tr>
            <td><a class="Estilo_3"
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp'>
                Configuracion del Sistema.</a>
            </td>
        </tr>
    </ol>
</table>
<p class="Estilo_4"><b> Fichero de propiedades del
recurso:</b><br><br>
        <table class="Estilo_4">
            <tr><td><b>Tipo de Recurso: </b><%=
Recurso_Administracion_del_Sistema.Descripcion().getResourceNameComponent().getTipoRecur
so() %><br></td></tr>
            <tr><td><b>Descripcion: </b><%=
Recurso_Administracion_del_Sistema.Descripcion().getResourceNameComponent().getDescripci
on() %></td></tr>
        </table>
    </center>
    <% } %>
</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
    <tr>
        <td width="25%"></td>
        <td width="50%">
            <blockquote class="Estilo_8">
                El Servidor de Registro de Acceso será el encargado de registrar
                en la base de datos "Registro de Acceso" todas las incidencias,
                accesos e intercambios de información que se produzcan
                en la arquitectura de seguridad local de un Sistema de Infomación
                Sanitario.
            </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
            <b>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
                Bienvenida al cliente. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
                Identificación & Autenticación. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
                Obtención de Credenciales. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
                Decisión de Acceso. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
                Administración del Sistema </a> |
            </b>
        </td>
    </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.52. Menu_Configuraciones.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Menu_Configuraciones.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP presenta las opciones de configuración de las tablas, auxiliares y
* dedicadas a usuarios del servidor, de la base de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Titulo página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Servidor de Registro de Acceso</u></b>
</p>
</td>
</tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if (ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align="center" class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<br>
<% }
else { %>
<center>
<h2>ADMINISTRACIÓN DEL SISTEMA</h2>
<h3>CONFIGURACIONES</h3>
<p><b><%= new java.util.Date() %></b></p>
<% if (request.getParameter("Login")==null)
{ %>
<center>
<p class="Estilo_4"> Por favor <b><%=
ControlAcceso.getIdUsuario() %></b> introduce el <br>
Login y el Password utilizados en el
Servidor de Registro de Acceso.
</p>
<form method="post">
<table class="Estilo_6">
<tr></tr>
<tr>
<td>Login: </td>
<td>

```

```

<input type="text" name="Login" maxLength="20"
style="font-family: Arial Narrow" size="15">
</td>
</tr>
<tr>
 Password: <input type="password" name="Password" maxLength="9" style="font-family: Arial Narrow" size="15"> </td> </tr> <tr>  </td>  <a class="Estilo_3" href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Configura cion_Comunicacion.jsp"> Códigos Comunicación.</a> </td> </tr> <tr>  <a class="Estilo_3" href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Configura cion_Forma_de_Envio.jsp"> Códigos Envíos.</a> </td> </tr> <tr>  <a class="Estilo_3" href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Configura cion_Motivos_Denegacion.jsp"> Códigos Denegación.</a> </td> </tr> <tr>  <a class="Estilo_3" href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Configura cion_Incidencias.jsp"> Códigos Incidencias.</a> </td> </tr> <tr>  <a class="Estilo_3" href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Configura cion_Situacion_Acceso.jsp"> Códigos Accesos.</a> </td> </tr> <tr>  <a class="Estilo_3" href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Configura cion_Tipo_de_Soporte.jsp"> Códigos Soportes.</a> </td> </tr> <tr>  <a class="Estilo_3" href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Configura cion_Personal_Autorizado.jsp"> Personal Autorizado.</a> | | | | | | | | |
```

```

        </td>
      </tr>
      <tr>
        <td><a class="Estilo_3"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Configura
cion_Credenciales.jsp">
          Códigos Credenciales.</a>
        </td>
      </tr>
      <tr>
        <td><a class="Estilo_3"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Configura
cion_Recursos.jsp">
          Recursos.</a>
        </td>
      </tr>
    </ol>
  </table>
</center>
<% }
else { ControlIncidencia.setCodigoIncidencia("I-106");
ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
  <br>
  <p align=center class="Estilo_3">
  <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
  Se ha informado al Servidor de Registro de
Acceso.<br><br>
  </p>
  <br>
  <% }
} %>
</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
  <tr>
    <td width="25%"></td>
    <td width="50%">
      <blockquote class="Estilo_8">
        El Servidor de Registro de Acceso será el encargado de registrar
        en la base de datos "Registro de Acceso" todas las incidencias,
        accesos e intercambios de información que se produzcan
        en la arquitectura de seguridad local de un Sistema de Infomación
        Sanitario.
      </blockquote>
    </td>
    <td width="25%"></td>
  </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%" class="Estilo_6">
      <b>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
        Bienvenida al cliente. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
        Identificación & Autenticación. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
        Obtención de Credenciales. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
        Decisión de Acceso. </a> |

```

```
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
    Administración del Sistema </a> |<br>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp">
    Configuraciones </a> |
</b>
</td>
</tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>
```

9.53. Configuracion_Comunicacion.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Configuracion_Comunicacion.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP permite modificar los campos de la tabla "comunicacion" de la base
* de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Servidor de Registro de Acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
          <br>
          <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
          </p>
          <br>
          <% }
          else {
            if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),201)))
              { ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
              <br>
              <p align=center class="Estilo_3">
                <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
                Se ha informado al Servidor de Registro de Acceso.<br><br>
                </p>
                <br>
                <% }

            else {
              if ( request.getParameter("NuevoCodigo")==null) {}
              else { %>
                <table align="center" class="Estilo_3">
                  <tr>

```

```

        <td><b><u>Mensaje:</u></b></td>
    </tr>
    <tr>
        <td><%=
Informacion.Insertar(1,request.getParameter("NuevoCodigo"),request.getParameter("NuevaDe
scripcion")) %></td>
    </tr>
</table>
<br>
<% }
if ( request.getParameter("ActualizarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Actualizar(1,request.getParameter("ActualizarCodigo"),request.getParameter("
ActualizarDescripcion")) %></td>
        </tr>
    </table>
    <br>
    <% }
if ( request.getParameter("BorrarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Borrar(1,request.getParameter("BorrarCodigo")) %></td>
        </tr>
    </table>
    <br>
    <% }
    Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(1);
    ParametrosDeConsulta mostrar = null; %>
    <center>
    <p><b>Códigos de Comunicación registrados en la base de
datos.</b></p>
    <table class="Estilo_4" border="1" width="70%">
        <tr>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Código</font></td>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Descripción</font></td>
        </tr>
        <% for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar =
(ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1()
%> </td>
            <td align="center"> <%= mostrar.getParametro2()
%> </td>
        </tr>
    <% } %>
    </table>
    <br>
    </center>
    <TABLE CLASS="Estilo_6" ALIGN=CENTER BGCOLOR="#FFFFCC"
BORDER="1">
        <TR>
            <TD WIDTH="39%">
                <form method="POST">
                    <table class="Estilo_6" align=center>
                        <tr></tr>
                        <tr>
                            <td>Nuevo Código: </td><td>C-<INPUT
TYPE="TEXT" MAXLENGTH="3" NAME="NuevoCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                        </tr>
                    </table>
                </form>
            </TD>
        </TR>
    </TABLE>

```

```

                                <td>Descripción: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="200" VALUE="[Este código no presenta descripción]"
NAME="NuevaDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Insertar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="42%">
                                <form method="POST">
                                <table class="Estilo_6" align="center">
                                <tr></tr>
                                <tr>
                                <td>Actualizar Código: </td><td>C-
<INPUT TYPE="TEXT" MAXLENGTH="3" NAME="ActualizarCodigo" style="font-family: Arial
Narrow" SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Nueva Descripción:
</td><td><INPUT TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe la nueva descripción]"
NAME="ActualizarDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Actualizar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="19%">
                                <form method="POST">
                                <table class="Estilo_6" align="center">
                                <tr></tr>
                                <tr>
                                <td>Borrar Código: </td><td>C-<INPUT
TYPE="TEXT" MAXLENGTH="3" NAME="BorrarCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                                </tr>
                                <tr></tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Borrar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                </TR>
                                </TABLE>
                                <%
ControlAcceso.setFicheroAccedido("Configuracion_Comunicacion.jsp");
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setURL(request.getServletPath());
ControlAcceso.setCopiaRespaldo(false);
ControlAcceso.setTipoAcceso("Escritura");
ControlAcceso.setCodigoAcceso("A-108");

Despersonalizacion.Acceso(ControlAcceso.getCodigoAcceso());%>
                                <p align="center" class="Estilo_3"><%=
RegistrarAcceso.Registrar() %></p>
                                <% }
                                } %>
                                </font>
                                </td>
                                </tr>
                                </table>
                                <br>
                                <!-- Descripcion ===== -->
                                <table width="100%">
                                <tr>
                                <td width="25%"></td>
                                <td width="50%">
                                <blockquote class="Estilo_8">
                                El Servidor de Registro de Acceso será el encargado de registrar

```

```

        en la base de datos "Registro de Acceso" todas las incidencias,
        accesos e intercambios de información que se produzcan
        en la arquitectura de seguridad local de un Sistema de Infomación
        Sanitario.
    </blockquote>
</td>
<td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" class="Estilo_6">
<b>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
Bienvenida al cliente. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
Identificación & Autentificación. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
Obtención de Credenciales. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
Decisión de Acceso. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
Administración del Sistema </a> |<br>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp">
Configuraciones </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
Cerrar sesión. </a> |
</b>
</td>
</tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.54. Configuracion_Credenciales.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Configuracion_Credenciales.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP permite modificar los campos de la tabla "credenciales" de la base
* de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Servidor de Registro de Acceso</u></b>
</p>
</td>
</tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if (ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<br>
<% }
else {
if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),202)))
{ ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
</p>
<br>
<% }

else {
if ( request.getParameter("NuevoCodigo")==null) {}
else { %>
<table align="center" class="Estilo_3">
<tr>

```

```

        <td><b><u>Mensaje:</u></b></td>
    </tr>
    <tr>
        <td><%= Informacion.Insertar(2,
request.getParameter("NuevoCodigo"),
request.getParameter("NuevaDescripcion"),
request.getParameter("NuevoRecurso"))%>
    </td>
    </tr>
</table>
<br>
<% }
if ( request.getParameter("ActualizarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Actualizar(2,request.getParameter("ActualizarCodigo"),
request.getParameter("ActualizarDescripcion"),
request.getParameter("ActualizarRecurso")) %>
        </td>
    </tr>
    </table>
    <br>
    <% }
    if ( request.getParameter("BorrarCodigo")==null) {}
    else { %>
        <table align="center" class="Estilo_3">
            <tr>
                <td><b><u>Mensaje:</u></b></td>
            </tr>
            <tr>
                <td><%=
Informacion.Borrar(2,request.getParameter("BorrarCodigo")) %></td>
            </tr>
        </table>
        <br>
    <% }
    Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(2);
    ParametrosDeConsulta mostrar = null; %>
    <center>
    <p><b>Códigos de Credenciales para el Servidor de
Registro de Acceso.</b></p>
    <table class="Estilo_4" border="1" width="70%">
        <tr>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Código</font></td>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Descripción</font></td>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Recursos</font></td>
        </tr>
        <% for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar =
        (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1()
%> </td>
            <td align="center"> <%= mostrar.getParametro2()
%> </td>
            <td align="center"> <%= mostrar.getParametro3()
%> </td>
        </tr>
    <% } %>
    </table>
    <br>
</center>

```

```

<TABLE CLASS="Estilo_6" ALIGN="center" BGCOLOR="#FFFFCC"
BORDER="1">
    <TR>
        <TD WIDTH="39%">
            <form method="POST">
                <table class="Estilo_6" align="center">
                    <tr></tr>
                    <tr>
                        <td>Nuevo Código: </td><td>CR-<INPUT
TYPE="TEXT" MAXLENGTH="3" NAME="NuevoCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                    </tr>
                    <tr>
                        <td>Descripción: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="200" VALUE="[Este codigo no presenta descripción]"
NAME="NuevaDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                    </tr>
                    <tr>
                        <td>Recursos:<br>(Ejemplo: 1,3,5-
9,...)</td><td><INPUT TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe los números de
recursos]" NAME="NuevoRecurso" style="font-family: Arial Narrow" SIZE="40"></td>
                    </tr>
                    <tr>
                        <td></td><td align="right"><input
type="submit" value="Insertar" style="font-family: Arial Narrow"></td>
                    </tr>
                </table>
            </form>
        </TD>
        <TD WIDTH="42%">
            <form method="POST">
                <table class="Estilo_6" align="center">
                    <tr></tr>
                    <tr>
                        <td>Actualizar Código: </td><td>CR-
<INPUT TYPE="TEXT" MAXLENGTH="3" NAME="ActualizarCodigo" style="font-family: Arial
Narrow" SIZE="5"></td>
                    </tr>
                    <tr>
                        <td>Nueva Descripción:
</td><td><INPUT TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe la nueva descripción]"
NAME="ActualizarDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                    </tr>
                    <tr>
                        <td>Nuevos Recursos:<br>(Ejemplo:
1,3,5-9,...)</td><td><INPUT TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe los nuevos
números de recursos]" NAME="ActualizarRecurso" style="font-family: Arial Narrow"
SIZE="40"></td>
                    </tr>
                    <tr>
                        <td></td><td align="right"><input
type="submit" value="Actualizar" style="font-family: Arial Narrow"></td>
                    </tr>
                </table>
            </form>
        </TD>
        <TD WIDTH="19%">
            <form method="POST">
                <table class="Estilo_6" align="center">
                    <tr></tr>
                    <tr>
                        <td>Borrar Código: </td><td>CR-
<INPUT TYPE="TEXT" MAXLENGTH="3" NAME="BorrarCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                    </tr>
                    <tr></tr>
                    <tr>
                        <td></td><td align="right"><input
type="submit" value="Borrar" style="font-family: Arial Narrow"></td>
                    </tr>
                </table>
            </form>
        </TD>
    </TR>
</TABLE>

```

```

        <%
ControlAcceso.setFicheroAccedido("Configuracion_Credenciales.jsp");
        ControlAcceso.setCodigoDenegado("D-000");
        ControlAcceso.setURL(request.getServletPath());
        ControlAcceso.setCopiaRespaldo(false);
        ControlAcceso.setTipoAcceso("Escritura");
        ControlAcceso.setCodigoAcceso("A-108");

Despersonalizacion.Acceso(ControlAcceso.getCodigoAcceso());%>
        <p align="center" class="Estilo_3"><%=
RegistrarAcceso.Registrar() %></p>
        <% }
    } %>
    </font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
    <tr>
        <td width="25%"></td>
        <td width="50%">
            <blockquote class="Estilo_8">
                El Servidor de Registro de Acceso será el encargado de registrar
                en la base de datos "Registro de Acceso" todas las incidencias,
                accesos e intercambios de información que se produzcan
                en la arquitectura de seguridad local de un Sistema de Infomación
                Sanitario.
            </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
            <b>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
                Bienvenida al cliente. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
                Identificación & Autenticación. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
                Obtención de Credenciales. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
                Decisión de Acceso. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
                Administración del Sistema </a> |<br>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp">
                Configuraciones </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
                Cerrar sesión. </a> |
            </b>
        </td>
    </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.55. Configuracion_Forma_de_Envio.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Configuracion_Forma_de_Envio.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP permite modificar los campos de la tabla "forma_de_envio" de la base
* de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Servidor de Registro de Acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
          <br>
          <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
          </p>
          <br>
          <% }
          else {
            if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),203)))
              { ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
              <br>
              <p align=center class="Estilo_3">
                <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
                Se ha informado al Servidor de Registro de Acceso.<br><br>
                </p>
                <br>
                <% }

            else {
              if ( request.getParameter("NuevoCodigo")==null) {}
              else { %>
                <table align="center" class="Estilo_3">
                  <tr>

```

```

        <td><b><u>Mensaje:</u></b></td>
    </tr>
    <tr>
        <td><%=
Informacion.Insertar(3,request.getParameter("NuevoCodigo"),
request.getParameter("NuevaDescripcion")) %>
    </td>
    </tr>
</table>
<br>
<% }
if ( request.getParameter("ActualizarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Actualizar(3,request.getParameter("ActualizarCodigo"),
request.getParameter("ActualizarDescripcion")) %>
        </td>
        </tr>
    </table>
    <br>
    <% }
if ( request.getParameter("BorrarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Borrar(3,request.getParameter("BorrarCodigo")) %></td>
        </tr>
    </table>
    <br>
    <% }
    Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(3);
    ParametrosDeConsulta mostrar = null; %>
    <center>
    <p><b>Códigos de Envío registrados en la base de
datos.</b></p>
    <table class="Estilo_4" border="1" width="70%">
        <tr>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Código</font></td>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Descripción</font></td>
        </tr>
        <% for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar =
        (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1()
            <td align="center"> <%= mostrar.getParametro2()

        </tr>
        <% } %>
    </table>
    <br>
    </center>
    <TABLE CLASS="Estilo_6" ALIGN=CENTER BGCOLOR="#FFFFCC"
    BORDER="1">
        <TR>
            <TD WIDTH="39%">
                <form method="POST">
                    <table class="Estilo_6" align=center>
                        <tr></tr>
                    <tr>

```

```

                                <td>Nuevo Código: </td><td>E-<INPUT
TYPE="TEXT" MAXLENGTH="3" NAME="NuevoCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Descripción: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="200" VALUE="[Este codigo no presenta descripción]"
NAME="NuevaDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Insertar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="42%">
                                <form method="POST">
                                <table class="Estilo_6" align="center">
                                <tr></tr>
                                <tr>
                                <td>Actualizar Código: </td><td>E-
<INPUT TYPE="TEXT" MAXLENGTH="3" NAME="ActualizarCodigo" style="font-family: Arial
Narrow" SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Nueva Descripción:
</td><td><INPUT TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe la nueva descripción]"
NAME="ActualizarDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Actualizar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="19%">
                                <form method="POST">
                                <table class="Estilo_6" align="center">
                                <tr></tr>
                                <tr>
                                <td>Borrar Código: </td><td>E-<INPUT
TYPE="TEXT" MAXLENGTH="3" NAME="BorrarCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                                </tr>
                                <tr></tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Borrar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                </TR>
                                </TABLE>
                                <%
ControlAcceso.setFicheroAccedido("Configuracion_Forma_de_Envio.jsp");
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setURL(request.getServletPath());
ControlAcceso.setCopiaRespaldo(false);
ControlAcceso.setTipoAcceso("Escritura");
ControlAcceso.setCodigoAcceso("A-108");

Despersonalizacion.Acceso(ControlAcceso.getCodigoAcceso());%>
                                <p align="center" class="Estilo_3"><%=
RegistrarAcceso.Registrar() %></p>
                                <% }
                                } %>
                                </font>
                                </td>
                                </tr>
                                </table>
                                <br>
                                <!-- Descripcion ===== -->
                                <table width="100%">

```

```

<tr>
  <td width="25%"></td>
  <td width="50%">
    <blockquote class="Estilo_8">
      El Servidor de Registro de Acceso será el encargado de registrar
      en la base de datos "Registro de Acceso" todas las incidencias,
      accesos e intercambios de información que se produzcan
      en la arquitectura de seguridad local de un Sistema de Infomación
      Sanitario.
    </blockquote>
  </td>
  <td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%" class="Estilo_6">
      <b>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
        Bienvenida al cliente. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
        Identificación & Autenticación. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
        Obtención de Credenciales. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
        Decisión de Acceso. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
        Administración del Sistema </a> |<br>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp">
        Configuraciones </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
        Cerrar sesión. </a> |
      </b>
    </td>
  </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.56. Configuracion_Incidencias.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Configuracion_Incidencias.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP permite modificar los campos de la tabla "incidencias" de la base
* de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Servidor de Registro de Acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
          <br>
          <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
          </p>
          <br>
          <% }
          else {
            if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),204)))
              { ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
              <br>
              <p align=center class="Estilo_3">
                <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
                Se ha informado al Servidor de Registro de Acceso.<br><br>
                </p>
                <br>
                <% }

            else {
              if ( request.getParameter("NuevoCodigo")==null) {}
              else { %>
                <table align="center" class="Estilo_3">
                  <tr>

```

```

        <td><b><u>Mensaje:</u></b></td>
    </tr>
    <tr>
        <td><%=
Informacion.Insertar(4,request.getParameter("NuevoCodigo"),
request.getParameter("NuevaDescripcion"),
request.getParameter("NuevosEfectos")) %>
    </td>
    </tr>
</table>
<br>
<% }
if ( request.getParameter("ActualizarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Actualizar(4,request.getParameter("ActualizarCodigo"),
request.getParameter("ActualizarDescripcion"),
request.getParameter("ActualizarEfectos")) %></td>
        </tr>
    </table>
    <br>
    <% }
if ( request.getParameter("BorrarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Borrar(4,request.getParameter("BorrarCodigo")) %></td>
        </tr>
    </table>
    <br>
    <% }
    Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(4);
    ParametrosDeConsulta mostrar = null; %>
    <center>
    <p><b>Códigos de Incidencias registrados en la base de
datos.</b></p>
    <table class="Estilo_4" border="1" width="70%">
        <tr>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Código</font></td>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Descripción</font></td>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Efectos</font></td>
        </tr>
        <% for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar =
(ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1()
%> </td>
            <td align="center"> <%= mostrar.getParametro2()
%> </td>
            <td align="center"> <%= mostrar.getParametro3()
%> </td>
        </tr>
    <% } %>
    </table>
    <br>
    </center>
<TABLE CLASS="Estilo_6" ALIGN=CENTER BGCOLOR="#FFFFCC"
BORDER="1">

```

```

        <TR>
            <TD WIDTH="39%">
                <form method="POST">
                    <table class="Estilo_6" align=center>
                        <tr></tr>
                        <tr>
                            <td>Nuevo Código: </td><td><input
TYPE="TEXT" MAXLENGTH="3" NAME="NuevoCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                        </tr>
                        <tr>
                            <td>Descripción: </td><td><input
TYPE="TEXT" MAXLENGTH="200" VALUE="[Este codigo no presenta descripción]"
NAME="NuevaDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                        </tr>
                        <tr>
                            <td>Efectos: </td><td><input
TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe los efectos derivados.]"
NAME="NuevosEfectos" style="font-family: Arial Narrow" SIZE="40"></td>
                        </tr>
                        <tr>
                            <td></td><td align="right"><input
type="submit" value="Insertar" style="font-family: Arial Narrow"></td>
                        </tr>
                    </table>
                </form>
            </TD>
            <TD WIDTH="42%">
                <form method="POST">
                    <table class="Estilo_6" align=center>
                        <tr></tr>
                        <tr>
                            <td>Actualizar Código: </td><td><input
<input TYPE="TEXT" MAXLENGTH="3" NAME="ActualizarCodigo" style="font-family: Arial
Narrow" SIZE="5"></td>
                        </tr>
                        <tr>
                            <td>Nueva Descripción:
</td><td><input TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe la nueva descripción]"
NAME="ActualizarDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                        </tr>
                        <tr>
                            <td>Nuevos Efectos: </td><td><input
TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe los nuevos efectos]"
NAME="ActualizarEfectos" style="font-family: Arial Narrow" SIZE="40"></td>
                        </tr>
                        <tr>
                            <td></td><td align="right"><input
type="submit" value="Actualizar" style="font-family: Arial Narrow"></td>
                        </tr>
                    </table>
                </form>
            </TD>
            <TD WIDTH="19%">
                <form method="POST">
                    <table class="Estilo_6" align=center>
                        <tr></tr>
                        <tr>
                            <td>Borrar Código: </td><td><input
TYPE="TEXT" MAXLENGTH="3" NAME="BorrarCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                        </tr>
                        <tr>
                            <td></td><td align="right"><input
type="submit" value="Borrar" style="font-family: Arial Narrow"></td>
                        </tr>
                    </table>
                </form>
            </TD>
        </TR>
    </TABLE>
<%
ControlAcceso.setFicheroAccedido("Configuracion_Incidencias.jsp");
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setURL(request.getServletPath());

```

```

        ControlAcceso.setCopiaRespaldo(false);
        ControlAcceso.setTipoAcceso("Escritura");
        ControlAcceso.setCodigoAcceso("A-108");

Despersonalizacion.Acceso(ControlAcceso.getCodigoAcceso());%>
        <p align="center" class="Estilo_3"><%=
RegistrarAcceso.Registrar() %></p>
        <% }
    } %>
    </font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
    <tr>
        <td width="25%"></td>
        <td width="50%">
            <blockquote class="Estilo_8">
                El Servidor de Registro de Acceso será el encargado de registrar
                en la base de datos "Registro de Acceso" todas las incidencias,
                accesos e intercambios de información que se produzcan
                en la arquitectura de seguridad local de un Sistema de Infomación
                Sanitario.
            </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
            <b>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
                Bienvenida al cliente. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
                Identificación & Autentificación. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
                Obtención de Credenciales. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
                Decisión de Acceso. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
                Administración del Sistema </a> |<br>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp">
                Configuraciones </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
                Cerrar sesión. </a> |
            </b>
        </td>
    </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.57. Configuracion_Motivos_Denegacion.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Configuracion_Motivos_Denegacion.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP permite modificar los campos de la tabla "motivos_de_denegacion" de la base
* de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Servidor de Registro de Acceso</u></b>
</p>
</td>
</tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if (ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<br>

<% }
else {
if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),205)))
{ ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
</p>
<br>

<% }

else {
if ( request.getParameter("NuevoCodigo")==null) {}
else { %>
<table align="center" class="Estilo_3">
<tr>

```

```

        <td><b><u>Mensaje:</u></b></td>
    </tr>
    <tr>
        <td><%=
Informacion.Insertar(5,request.getParameter("NuevoCodigo"),
request.getParameter("NuevaDescripcion")) %>
    </td>
    </tr>
</table>
<br>
<% }
if ( request.getParameter("ActualizarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Actualizar(5,request.getParameter("ActualizarCodigo"),
request.getParameter("ActualizarDescripcion")) %>
        </td>
        </tr>
    </table>
    <br>
    <% }
if ( request.getParameter("BorrarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Borrar(5,request.getParameter("BorrarCodigo")) %></td>
        </tr>
    </table>
    <br>
    <% }
    Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(5);
    ParametrosDeConsulta mostrar = null; %>
    <center>
    <p><b>Códigos de denegación registrados en la base de
datos.</b></p>
    <table class="Estilo_4" border="1" width="70%">
        <tr>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Código</font></td>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Descripción</font></td>
        </tr>
        <% for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar =
        (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1()
            <td align="center"> <%= mostrar.getParametro2()

        </tr>
    <% } %>
    </table>
    <br>
    </center>
    <TABLE CLASS="Estilo_6" ALIGN=CENTER BGCOLOR="#FFFFCC"
    BORDER="1">
        <TR>
            <TD WIDTH="39%">
                <form method="POST">
                    <table class="Estilo_6" align=center>
                        <tr></tr>
                    <tr>

```

```

                                <td>Nuevo Código: </td><td>D-<INPUT
TYPE="TEXT" MAXLENGTH="3" NAME="NuevoCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Descripción: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="200" VALUE="[Este codigo no presenta descripción]"
NAME="NuevaDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Insertar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="42%">
                                <form method="POST">
                                <table class="Estilo_6" align="center">
                                <tr></tr>
                                <tr>
                                <td>Actualizar Código: </td><td>D-
                                <INPUT TYPE="TEXT" MAXLENGTH="3" NAME="ActualizarCodigo" style="font-family: Arial
                                Narrow" SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Nueva Descripción:
                                </td><td><INPUT TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe la nueva descripción]"
                                NAME="ActualizarDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
                                type="submit" value="Actualizar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="19%">
                                <form method="POST">
                                <table class="Estilo_6" align="center">
                                <tr></tr>
                                <tr>
                                <td>Borrar Código: </td><td>D-<INPUT
                                TYPE="TEXT" MAXLENGTH="3" NAME="BorrarCodigo" style="font-family: Arial Narrow"
                                SIZE="5"></td>
                                </tr>
                                <tr></tr>
                                <tr>
                                <td></td><td align="right"><input
                                type="submit" value="Borrar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                </TR>
                                </TABLE>
                                <%
ControlAcceso.setFicheroAccedido("Configuracion_Motivos_Denegacion.jsp");
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setURL(request.getServletPath());
ControlAcceso.setCopiaRespaldo(false);
ControlAcceso.setTipoAcceso("Escritura");
ControlAcceso.setCodigoAcceso("A-108");

Despersonalizacion.Acceso(ControlAcceso.getCodigoAcceso());%>
                                <p align="center" class="Estilo_3"><%=
RegistrarAcceso.Registrar() %></p>
                                <% }
                                } %>
                                </font>
                                </td>
                                </tr>
                                </table>
                                <br>
                                <!-- Descripcion ===== -->
                                <table width="100%">

```

```

<tr>
  <td width="25%"></td>
  <td width="50%">
    <blockquote class="Estilo_8">
      El Servidor de Registro de Acceso será el encargado de registrar
      en la base de datos "Registro de Acceso" todas las incidencias,
      accesos e intercambios de información que se produzcan
      en la arquitectura de seguridad local de un Sistema de Infomación
      Sanitario.
    </blockquote>
  </td>
  <td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%" class="Estilo_6">
      <b>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
        Bienvenida al cliente. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
        Identificación & Autenticación. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
        Obtención de Credenciales. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
        Decisión de Acceso. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
        Administración del Sistema </a> |<br>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp">
        Configuraciones </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
        Cerrar sesión. </a> |
      </b>
    </td>
  </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.58. Configuracion_Personal_Autorizado.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Configuracion_Personal_Autorizado.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP permite modificar los campos de la tabla "personal_autorizado" de la base
* de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Servidor de Registro de Acceso</u></b>
</p>
</td>
</tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if (ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align="center" class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<br>
<% }
else {
if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),206)))
{ ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align="center" class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
</p>
<br>
<% }
else {
if ( request.getParameter("NuevoLogin")==null) {}
else { %>
<table align="center" class="Estilo_3">
<tr>
<td><b><u>Mensaje:</u></b></td>

```

```

        </tr>
        <tr>
            <td><%=
Informacion.Insertar(request.getParameter("NuevoLogin"),
request.getParameter("NuevaPassword"),
request.getParameter("NuevaIdentificacion"),
request.getParameter("NuevasCredenciales")) %>
            </td>
        </tr>
    </table>
    <br>
    <% }
    if ( request.getParameter("ActualizarLogin")==null) {}
    else { %>
        <table align="center" class="Estilo_3">
            <tr>
                <td><b><u>Mensaje:</u></b></td>
            </tr>
            <tr>
                <td><%=
Informacion.Actualizar(request.getParameter("ActualizarLogin"),
request.getParameter("ActualizarPassword"),
request.getParameter("ActualizarIdentificacion")) %>
            </td>
        </tr>
    </table>
    <br>
    <% }
    if ( request.getParameter("BorrarLogin")==null) {}
    else { %>
        <table align="center" class="Estilo_3">
            <tr>
                <td><b><u>Mensaje:</u></b></td>
            </tr>
            <tr>
                <td><%=
Informacion.Borrar(6,request.getParameter("BorrarLogin")) %></td>
            </tr>
        </table>
        <br>
    <% }
    Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(6);
    ParametrosDeConsulta mostrar = null; %>
    <center>
    <p><b>Personal Autorizado en el Servidor de Registro de
Acceso.</b></p>
        <table class="Estilo_4" border="1" width="70%">
            <tr>
                <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Login</font></td>
                <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Password</font></td>
                <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Identificación</font></td>
                <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Credenciales</font></td>
            </tr>
            <% for (int i=0; i<ConsultaEstatica.size();i++)
            { mostrar =
(ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
            <tr>
                <td align="center"> <%= mostrar.getParametro1()
%> </td>
                <td align="center"> <%= mostrar.getParametro2()
%> </td>
                <td align="center"> <%= mostrar.getParametro3()
%> </td>
                <td align="center"> <%= mostrar.getParametro4()
%> </td>
            </tr>
            <% } %>
        </table>
        <br>
    </center>
<TABLE CLASS="Estilo_6" ALIGN=CENTER BGCOLOR="#FFFFCC" BORDER="1">

```

```

<TR>
  <TD WIDTH="39%">
    <form method="POST">
      <table class="Estilo_6" align=center>
        <tr>Nuevo Usuario.</tr>
        <tr>
          <td>Login: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="15" NAME="NuevoLogin" style="font-family: Arial Narrow"
SIZE="15"></td>
        </tr>
        <tr>
          <td>Password: </td><td><INPUT
TYPE="Password" MAXLENGTH="9" NAME="NuevaPassword" style="font-family: Arial Narrow"
SIZE="15"></td>
        </tr>
        <tr>
          <td>Identificación: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="200" NAME="NuevaIdentificacion" style="font-family: Arial Narrow"
SIZE="20"></td>
        </tr>
        <tr>
          <td>Código Credenciales (1):
</td><td>CR-<INPUT TYPE="TEXT" MAXLENGTH="3" NAME="NuevasCredenciales" style="font-
family: Arial Narrow" SIZE="3"></td>
        </tr>
        <tr>
          <td></td><td align="right"><input
type="submit" value="Insertar" style="font-family: Arial Narrow"></td>
        </tr>
      </table>
    </form>
  </TD>
  <TD WIDTH="42%">
    <form method="POST">
      <table class="Estilo_6" align=center>
        <tr>Actualizar Usuario (2).</tr>
        <tr>
          <td>Login: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="15" NAME="ActualizarLogin" style="font-family: Arial Narrow"
SIZE="15"></td>
        </tr>
        <tr>
          <td>Nueva Password: </td><td><INPUT
TYPE="Password" MAXLENGTH="9" NAME="ActualizarPassword" style="font-family: Arial
Narrow" SIZE="15"></td>
        </tr>
        <tr>
          <td>Nueva Identificación:
</td><td><INPUT TYPE="TEXT" MAXLENGTH="200" NAME="ActualizarIdentificacion" style="font-
family: Arial Narrow" SIZE="20"></td>
        </tr>
        <tr>
          <td></td><td align="right"><input
type="submit" value="Actualizar" style="font-family: Arial Narrow"></td>
        </tr>
      </table>
    </form>
  </TD>
  <TD WIDTH="19%">
    <form method="POST">
      <table class="Estilo_6" align=center>
        <tr>Borrar usuario.</tr>
        <tr>
          <td>Login: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="15" NAME="BorrarLogin" style="font-family: Arial Narrow"
SIZE="15"></td>
        </tr>
        <tr></tr>
        <tr>
          <td></td><td align="right"><input
type="submit" value="Borrar" style="font-family: Arial Narrow"></td>
        </tr>
      </table>
    </form>
  </TD>
</TR>

```

```

        </TABLE>
        <p align=center class="Estilo_6">(1) Consulte las
credenciales en el apartado Códigos Credenciales.</p>
        <p align=center class="Estilo_6">(2) Para modificar las
credenciales del usuario es necesario eliminarlo y crear uno nuevo.</p>
        <%
ControlAcceso.setFicheroAccedido("Configuracion_Personal_Autorizado.jsp");
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setURL(request.getServletPath());
ControlAcceso.setCopiaRespaldo(false);
ControlAcceso.setTipoAcceso("Escritura");
ControlAcceso.setCodigoAcceso("A-108");

Despersonalizacion.Acceso(ControlAcceso.getCodigoAcceso());%>
        <p align="center" class="Estilo_3"><%=
RegistrarAcceso.Registrar() %></p>
        <% }
        } %>
    </font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
    <tr>
        <td width="25%"></td>
        <td width="50%">
            <blockquote class="Estilo_8">
                El Servidor de Registro de Acceso será el encargado de registrar
                en la base de datos "Registro de Acceso" todas las incidencias,
                accesos e intercambios de información que se produzcan
                en la arquitectura de seguridad local de un Sistema de Infomación
                Sanitario.
            </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
            <b>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
                Bienvenida al cliente. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
                Identificación & Autentificación. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
                Obtención de Credenciales. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
                Decisión de Acceso. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
                Administración del Sistema </a> |<br>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp">
                Configuraciones </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
                Cerrar sesión. </a> |
            </b>
        </td>
    </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.59. Configuracion_Recursos.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Configuracion_Recursos.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP permite modificar los campos de la tabla "recursos" de la base
* de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Servidor de Registro de Acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
          <br>
          <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
          </p>
          <br>
          <% }
          else {
            if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),215)))
              { ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
              <br>
              <p align=center class="Estilo_3">
                <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
                Se ha informado al Servidor de Registro de Acceso.<br><br>
                </p>
                <br>
                <% }

            else {
              if ( request.getParameter("NuevoCodigo")==null) {}
              else { %>
                <table align="center" class="Estilo_3">
                  <tr>

```

```

        <td><b><u>Mensaje:</u></b></td>
    </tr>
    <tr>
        <td><%=
Informacion.Insertar(15,request.getParameter("NuevoCodigo"),
request.getParameter("NuevaDescripcion")) %>
        </td>
    </tr>
</table>
<br>
<% }
if ( request.getParameter("ActualizarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Actualizar(15,request.getParameter("ActualizarCodigo"),
request.getParameter("ActualizarDescripcion")) %>
            </td>
        </tr>
    </table>
    <br>
    <% }
    if ( request.getParameter("BorrarCodigo")==null) {}
    else { %>
        <table align="center" class="Estilo_3">
            <tr>
                <td><b><u>Mensaje:</u></b></td>
            </tr>
            <tr>
                <td><%=
Informacion.Borrar(15,request.getParameter("BorrarCodigo")) %></td>
            </tr>
        </table>
        <br>
    <% }
    Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(15);
    ParametrosDeConsulta mostrar = null; %>
    <center>
    <p><b>Recursos del Servidor de Registro de
Acceso.</b></p>
    <table class="Estilo_4" border="1" width="70%">
        <tr>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Recurso</font></td>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Descripción</font></td>
        </tr>
        <% for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar =
(ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1()
            <td align="center"> <%= mostrar.getParametro2()

        </tr>
    <% } %>
</table>
<br>
</center>
<TABLE CLASS="Estilo_6" ALIGN=CENTER BGCOLOR="#FFFFCC"
BORDER="1">
    <TR>
        <TD WIDTH="39%">
            <form method="POST">
                <table class="Estilo_6" align=center>
                    <tr></tr>
                <tr>

```

```

                                <td>Nuevo Recurso: </td><td><INPUT
TYPE="TEXT" NAME="NuevoCodigo" style="font-family: Arial Narrow" SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Descripción: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="200" VALUE="[Este codigo no presenta descripción]"
NAME="NuevaDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Insertar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="42%">
                                <form method="POST">
                                <table class="Estilo_6" align="center">
                                <tr></tr>
                                <tr>
                                <td>Actualizar Recurso:
</td><td><INPUT TYPE="TEXT" NAME="ActualizarCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Nueva Descripción:
</td><td><INPUT TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe la nueva descripción]"
NAME="ActualizarDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Actualizar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="19%">
                                <form method="POST">
                                <table class="Estilo_6" align="center">
                                <tr></tr>
                                <tr>
                                <td>Borrar Recurso: </td><td><INPUT
TYPE="TEXT" NAME="BorrarCodigo" style="font-family: Arial Narrow" SIZE="5"></td>
                                </tr>
                                <tr></tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Borrar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                </TR>
                                </TABLE>
                                <%
ControlAcceso.setFicheroAccedido("Configuracion_Recursos.jsp");
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setURL(request.getServletPath());
ControlAcceso.setCopiaRespaldo(false);
ControlAcceso.setTipoAcceso("Escritura");
ControlAcceso.setCodigoAcceso("A-108");

Despersonalizacion.Acceso(ControlAcceso.getCodigoAcceso());%>
                                <p align="center" class="Estilo_3"><%=
RegistrarAcceso.Registrar() %></p>
                                <% }
                                } %>
                                </font>
                                </td>
                                </tr>
                                </table>
                                <br>
                                <!-- Descripcion ===== -->
                                <table width="100%">
                                <tr>
                                <td width="25%"></td>

```

```

        <td width="50%">
            <blockquote class="Estilo_8">
                El Servidor de Registro de Acceso será el encargado de registrar
                en la base de datos "Registro de Acceso" todas las incidencias,
                accesos e intercambios de información que se produzcan
                en la arquitectura de seguridad local de un Sistema de Infomación
                Sanitario.
            </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
            <b>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
                Bienvenida al cliente. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
                Identificación & Autentificación. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
                Obtención de Credenciales. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
                Decisión de Acceso. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
                Administración del Sistema </a> |<br>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp">
                Configuraciones </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
                Cerrar sesión. </a> |
            </b>
        </td>
    </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.60. Configuracion_Situacion_Acceso.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Configuracion_Situacion_Acceso.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP permite modificar los campos de la tabla "situacion_acceso" de la base
* de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Titulo página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Servidor de Registro de Acceso</u></b>
</p>
</td>
</tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if (ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<br>
<% }
else {
if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),207)))
{ ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
</p>
<br>
<% }

else {
if ( request.getParameter("NuevoCodigo")==null) {}
else { %>
<table align="center" class="Estilo_3">
<tr>

```

```

        <td><b><u>Mensaje:</u></b></td>
    </tr>
    <tr>
        <td><%=
Informacion.Insertar(7,request.getParameter("NuevoCodigo"),
request.getParameter("NuevaDescripcion")) %>
    </td>
    </tr>
</table>
<br>
<% }
if ( request.getParameter("ActualizarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Actualizar(7,request.getParameter("ActualizarCodigo"),
request.getParameter("ActualizarDescripcion")) %>
        </td>
        </tr>
    </table>
    <br>
    <% }
if ( request.getParameter("BorrarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Borrar(7,request.getParameter("BorrarCodigo")) %></td>
        </tr>
    </table>
    <br>
    <% }
    Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(7);
    ParametrosDeConsulta mostrar = null; %>
    <center>
    <p><b>Códigos de Situación de Acceso registrados en la
base de datos.</b></p>
    <table class="Estilo_4" border="1" width="70%">
        <tr>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Código</font></td>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Descripción</font></td>
        </tr>
        <% for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar =
        (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1()
            <td align="center"> <%= mostrar.getParametro2()

        </tr>
        <% } %>
    </table>
    <br>
    </center>
    <TABLE CLASS="Estilo_6" ALIGN=CENTER BGCOLOR="#FFFFCC"
    BORDER="1">
        <TR>
            <TD WIDTH="39%">
                <form method="POST">
                    <table class="Estilo_6" align=center>
                        <tr></tr>
                    <tr>

```

```

                                <td>Nuevo Código: </td><td>A-<INPUT
TYPE="TEXT" MAXLENGTH="3" NAME="NuevoCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Descripción: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="200" VALUE="[Este codigo no presenta descripción]"
NAME="NuevaDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Insertar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="42%">
                                <form method="POST">
                                <table class="Estilo_6" align=center>
                                <tr></tr>
                                <tr>
                                <td>Actualizar Código: </td><td>A-
<INPUT TYPE="TEXT" MAXLENGTH="3" NAME="ActualizarCodigo" style="font-family: Arial
Narrow" SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Nueva Descripción:
</td><td><INPUT TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe la nueva descripción]"
NAME="ActualizarDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Actualizar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="19%">
                                <form method="POST">
                                <table class="Estilo_6" align=center>
                                <tr></tr>
                                <tr>
                                <td>Borrar Código: </td><td>A-<INPUT
TYPE="TEXT" MAXLENGTH="3" NAME="BorrarCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                                </tr>
                                <tr></tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Borrar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                </TR>
                                </TABLE>
                                <%
ControlAcceso.setFicheroAccedido("Configuracion_Situacion_Acceso.jsp");
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setURL(request.getServletPath());
ControlAcceso.setCopiaRespaldo(false);
ControlAcceso.setTipoAcceso("Escritura");
ControlAcceso.setCodigoAcceso("A-108");

Despersonalizacion.Acceso(ControlAcceso.getCodigoAcceso());%>
                                <p align="center" class="Estilo_3"><%=
RegistrarAcceso.Registrar() %></p>
                                <% }
                                } %>
                                </font>
                                </td>
                                </tr>
                                </table>
                                <br>
                                <!-- Descripcion ===== -->
                                <table width="100%">

```

```

<tr>
  <td width="25%"></td>
  <td width="50%">
    <blockquote class="Estilo_8">
      El Servidor de Registro de Acceso será el encargado de registrar
      en la base de datos "Registro de Acceso" todas las incidencias,
      accesos e intercambios de información que se produzcan
      en la arquitectura de seguridad local de un Sistema de Infomación
      Sanitario.
    </blockquote>
  </td>
  <td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%" class="Estilo_6">
      <b>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
        Bienvenida al cliente. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
        Identificación & Autenticación. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
        Obtención de Credenciales. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
        Decisión de Acceso. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
        Administración del Sistema </a> |<br>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp">
        Configuraciones </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
        Cerrar sesión. </a> |
      </b>
    </td>
  </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.61. Configuracion_Tipo_de_Soporte.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Configuracion_Tipo_de_Soporte.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP permite modificar los campos de la tabla "tipo_de_soporte" de la base
* de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Titulo página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Servidor de Registro de Acceso</u></b>
</p>
</td>
</tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if (ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<br>
<% }
else {
if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),208)))
{ ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
</p>
<br>
<% }

else {
if ( request.getParameter("NuevoCodigo")==null) {}
else { %>
<table align="center" class="Estilo_3">
<tr>

```

```

        <td><b><u>Mensaje:</u></b></td>
    </tr>
    <tr>
        <td><%=
Informacion.Insertar(8,request.getParameter("NuevoCodigo"),
request.getParameter("NuevaDescripcion")) %>
        </td>
    </tr>
</table>
<br>
<% }
if ( request.getParameter("ActualizarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Actualizar(8,request.getParameter("ActualizarCodigo"),
request.getParameter("ActualizarDescripcion")) %>
            </td>
        </tr>
    </table>
    <br>
    <% }
if ( request.getParameter("BorrarCodigo")==null) {}
else { %>
    <table align="center" class="Estilo_3">
        <tr>
            <td><b><u>Mensaje:</u></b></td>
        </tr>
        <tr>
            <td><%=
Informacion.Borrar(8,request.getParameter("BorrarCodigo")) %></td>
        </tr>
    </table>
    <br>
    <% }
    Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(8);
    ParametrosDeConsulta mostrar = null; %>
    <center>
    <p><b>Códigos de Tipo de Soportes Informáticos
registrados en la base de datos.</b></p>
    <table class="Estilo_4" border="1" width="70%">
        <tr>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Código</font></td>
            <td bgcolor="#808080" align="center"><font
color="#FFFFFF">Descripción</font></td>
        </tr>
        <% for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar =
(ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1()
%> </td>
            <td align="center"> <%= mostrar.getParametro2()
%> </td>
        </tr>
    <% } %>
    </table>
    <br>
    </center>
    <TABLE CLASS="Estilo_6" ALIGN=CENTER BGCOLOR="#FFFFCC"
    BORDER="1">
        <TR>
            <TD WIDTH="39%">
                <form method="POST">
                    <table class="Estilo_6" align=center>
                        <tr></tr>
                    <tr>

```

```

                                <td>Nuevo Código: </td><td>S-<INPUT
TYPE="TEXT" MAXLENGTH="3" NAME="NuevoCodigo" style="font-family: Arial Narrow"
SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Descripción: </td><td><INPUT
TYPE="TEXT" MAXLENGTH="200" VALUE="[Este codigo no presenta descripción]"
NAME="NuevaDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
type="submit" value="Insertar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="42%">
                                <form method="POST">
                                <table class="Estilo_6" align="center">
                                <tr></tr>
                                <tr>
                                <td>Actualizar Código: </td><td>S-
                                <INPUT TYPE="TEXT" MAXLENGTH="3" NAME="ActualizarCodigo" style="font-family: Arial
                                Narrow" SIZE="5"></td>
                                </tr>
                                <tr>
                                <td>Nueva Descripción:
                                </td><td><INPUT TYPE="TEXT" MAXLENGTH="200" VALUE="[Escribe la nueva descripción]"
                                NAME="ActualizarDescripcion" style="font-family: Arial Narrow" SIZE="40"></td>
                                </tr>
                                <tr>
                                <td></td><td align="right"><input
                                type="submit" value="Actualizar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                <TD WIDTH="19%">
                                <form method="POST">
                                <table class="Estilo_6" align="center">
                                <tr></tr>
                                <tr>
                                <td>Borrar Código: </td><td>S-<INPUT
                                TYPE="TEXT" MAXLENGTH="3" NAME="BorrarCodigo" style="font-family: Arial Narrow"
                                SIZE="5"></td>
                                </tr>
                                <tr></tr>
                                <tr>
                                <td></td><td align="right"><input
                                type="submit" value="Borrar" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                </TD>
                                </TR>
                                </TABLE>
                                <%
ControlAcceso.setFicheroAccedido("Configuracion_Tipo_de_Soporte.jsp");
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setURL(request.getServletPath());
ControlAcceso.setCopiaRespaldo(false);
ControlAcceso.setTipoAcceso("Escritura");
ControlAcceso.setCodigoAcceso("A-108");

Despersonalizacion.Acceso(ControlAcceso.getCodigoAcceso());%>
                                <p align="center" class="Estilo_3"><%=
RegistrarAcceso.Registrar() %></p>
                                <% }
                                } %>
                                </font>
                                </td>
                                </tr>
                                </table>
                                <br>
                                <!-- Descripcion ===== -->
                                <table width="100%">

```

```

<tr>
  <td width="25%"></td>
  <td width="50%">
    <blockquote class="Estilo_8">
      El Servidor de Registro de Acceso será el encargado de registrar
      en la base de datos "Registro de Acceso" todas las incidencias,
      accesos e intercambios de información que se produzcan
      en la arquitectura de seguridad local de un Sistema de Infomación
      Sanitario.
    </blockquote>
  </td>
  <td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%" class="Estilo_6">
      <b>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
        Bienvenida al cliente. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
        Identificación & Autenticación. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
        Obtención de Credenciales. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
        Decisión de Acceso. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
        Administración del Sistema </a> |<br>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Configuraciones/Menu_Conf
iguraciones.jsp">
        Configuraciones </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
        Cerrar sesión. </a> |
      </b>
    </td>
  </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.62. Menu_Consultas.jsp.-

```

/* =====
 * Proyecto final de carrera.
 * Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
 * Autor: Daniel David García Malmierca.
 * Área: Ingeniería Telemática.
 * Departamento: Ingeniería de Sistemas y Automática.
 * Tutora: Isabel Román Martínez.
 * Ingeniería de Telecomunicación - Universidad de Sevilla.
 * =====
 * $Id: Menu_Consultas.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
 * =====
 * Comentarios.-
 * Este JSP presenta las opciones de consulta de la base de datos "Registro de Acceso".
 * =====
 */
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Titulo página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Servidor de Registro de Acceso</u></b>
</p>
</td>
</tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if (ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<br>
<% }
else { %>
<center>
<h2>ADMINISTRACIÓN DEL SISTEMA</h2>
<h3>CONSULTAS</h3>
<p><b><%= new java.util.Date() %></b></p>
<p>Elija entre las siguientes opciones:</p>
<table>
<ol>
<tr>
<td><a class="Estilo_3"
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Consultas_Estab
lecidas.jsp'>
Consultas establecidas.</a>
</td>
</tr>
<tr>
<td><a class="Estilo_3"
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Consultas_Perso
nalizadas.jsp'>
Consultas personalizadas.</a>
</td>

```

```

        </tr>
        <tr>
            <td><a class="Estilo_3"
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Configurar_Cons
ultas_Personalizadas.jsp'>
                Configurar consultas personalizadas.</a>
            </td>
        </tr>
    </ol>
</table>
</center>
<% } %>
</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
    <tr>
        <td width="25%"></td>
        <td width="50%">
            <blockquote class="Estilo_8">
                El Servidor de Registro de Acceso será el encargado de registrar
                en la base de datos "Registro de Acceso" todas las incidencias,
                accesos e intercambios de información que se produzcan
                en la arquitectura de seguridad local de un Sistema de Infomación
                Sanitario.
            </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
            <b>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
                Bienvenida al cliente.</a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
                Identificación & Autentificación.</a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
                Obtención de Credenciales.</a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
                Decisión de Acceso.</a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
                Administración del Sistema</a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Menu_Consultas.
jsp">
                Consultas</a> |
            </b>
        </td>
    </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.63. Configurar_Consultas_Personalizadas.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Configurar_Consultas_Personalizadas.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP muestra un asistente que permite crear consultas personalizadas a la
* base de datos "Registro de Acceso".
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Servidor de Registro de Acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
          <br>
          <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
          </p>
          <br>
          <% }
          else { if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),300)))
          { ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
          <br>
          <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
          </p>
          <br>
          <% }
          else { %>
            <center>
            <h2>ADMINISTRACIÓN DEL SISTEMA</h2>
            <h3>CONFIGURAR CONSULTAS PERSONALIZADAS</h3>
            <p><b><%= new java.util.Date() %></b></p>
            <% if (request.getParameter("Campo")==null) {}

```

```

        else {
ConsultasPersonalizadas.setCampo(request.getParameter("Campo"));
        ConsultasPersonalizadas.InsertarCampo();
        ConsultasPersonalizadas.ClearCampo();
        }
        if (request.getParameter("condicion")==null) {}
        else {
ConsultasPersonalizadas.setCondicion(request.getParameter("condicion"));
        ConsultasPersonalizadas.InsertarCondicion();
        ConsultasPersonalizadas.ClearCondicion();
        }
        if (request.getParameter("orden")==null) {}
        else {
ConsultasPersonalizadas.setOrden(request.getParameter("orden"));
        ConsultasPersonalizadas.InsertarOrden();
        ConsultasPersonalizadas.ClearOrden();
        }
        if (request.getParameter("borrar")==null) {}
        else { if
(request.getParameter("borrar").contentEquals("Campos"))
        { ConsultasPersonalizadas.BorrarCampos(); }
        if
(request.getParameter("borrar").contentEquals("UltimoCampo"))
        { ConsultasPersonalizadas.BorrarUltimoCampo(); }
        }
        if (request.getParameter("borrar")==null) {}
        else { if
(request.getParameter("borrar").contentEquals("Condiciones"))
        { ConsultasPersonalizadas.BorrarCondiciones(); }
        if
(request.getParameter("borrar").contentEquals("UltimaCondicion"))
        { ConsultasPersonalizadas.BorrarUltimaCondicion(); }
        }
        if (request.getParameter("borrar")==null) {}
        else { if
(request.getParameter("borrar").contentEquals("Orden"))
        { ConsultasPersonalizadas.BorrarOrden(); }
        if
(request.getParameter("borrar").contentEquals("UltimoOrden"))
        { ConsultasPersonalizadas.BorrarUltimoOrden(); }
        } %>
        <font class="Estilo_4">
        <b>Asistente de consultas personalizadas</b>
        </font>
        <hr>
        <% if (request.getParameter("NombreConsulta")==null){}
        else { %><p align=center class="Estilo_2">
        Mensaje: <%=
ConsultasPersonalizadas.CrearConsultaSQL(request.getParameter("NombreConsulta"))
%><br><br>
        <a
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Consultas_Perso
nalizadas.jsp">Ver Consultas Personalizadas </a>
        </p>
        <% } %>
        <p align=left>
        <b>Campos seleccionados:</b><br>
        <table>
        <tr>
        <% for (int j=0;
j<ConsultasPersonalizadas.CamposSeleccionados.size(); j++)
        { %><td
class="Estilo_6"><%= (String)ConsultasPersonalizadas.CamposSeleccionados.elementAt(j)%></
td><% } %>
        </tr>
        </table>
        <b>Condiciones seleccionadas:</b><br>
        <table>
        <tr>
        <% for (int h=0;
h<ConsultasPersonalizadas.CondicionesSeleccionadas.size(); h++)
        { %><td
class="Estilo_6"><%= (String)ConsultasPersonalizadas.CondicionesSeleccionadas.elementAt(h)
%></td><% } %>
        </tr>
        </table>

```

```

        <b>Orden seleccionado:</b><br>
        <table>
            <tr>
                <% for (int f=0;
f<ConsultasPersonalizadas.OrdenSeleccionadas.size(); f++)
                    { %><td
class="Estilo_6"><%= (String)ConsultasPersonalizadas.OrdenSeleccionadas.elementAt(f)%></t
d><% } %>
                </tr>
            </table>
        </p>
        <hr>
        <form method="POST">
            <table align="center">
                <tr>
                    <td><b>Paso 1. Selecciona una de las tablas.</b></td>
                </tr>
                <tr>
                    <td class="Estilo_4">
                        <select class="Estilo_4" name="Tabla">
                            <option value="1">Tabla: COMUNICACION</option>
                            <option value="2">Tabla: CREDENCIALES</option>
                            <option value="3">Tabla: FORMA DE ENVIO</option>
                            <option value="4">Tabla: INCIDENCIAS</option>
                            <option value="5">Tabla: MOTIVO DE
DENEGACION</option>
                            <option value="6">Tabla: PERSONAL
AUTORIZADO</option>
                            <option value="7">Tabla: SITUACION DE
ACCESO</option>
                            <option value="8">Tabla: TIPO DE SOPORTE</option>
                            <option value="9">Tabla: REGISTRO DE
ACCESOS</option>
                            <option value="10">Tabla: REGISTRO DE ENTRADA DE
SOPORTES INFORMATICOS</option>
                            <option value="11">Tabla: REGISTRO DE
INCIDENCIAS</option>
                            <option value="12">Tabla: REGISTRO DE
RECUPERACION DE DATOS</option>
                            <option value="13">Tabla: REGISTRO DE SALIDA DE
SOPORTES INFORMATICOS</option>
                            <option value="14">Tabla: REGISTRO
RESPALDO</option>
                            <option value="15">Tabla: RECURSOS</option>
                        </select>
                    </td>
                    <td align="right"><input type="submit"
value="Consultar campos" style="font-family: Arial Narrow"></td>
                </tr>
            </table>
        </form>
        <% if (request.getParameter("Tabla")==null) {}
        else {
ConsultasPersonalizadas.setTabla(ConsultasPersonalizadas.MostrarNombreTabla(Convertir.St
ring_Int(request.getParameter("Tabla"))));%>
            <b>Tabla: </b><font class="Estilo_2"><%=
ConsultasPersonalizadas.MostrarNombreTabla(Convertir.String_Int(request.getParameter("Ta
bla")))%></font>
            <form method="POST">
                <table align="center">
                    <tr>
                        <td><b>Paso 2. Selecciona uno de los
campos.</b>
                    </td>
                </tr>
                <tr>
                    <td class="Estilo_4">
                        <select class="Estilo_4" name="Campo">
                            <% Vector Consulta =
ConsultasPersonalizadas.MostrarNombresCampos(Convertir.String_Int(request.getParameter("
Tabla")));
                            for (int i=0; i<Consulta.size(); i++)
                                { %><option value="<%=
(String)Consulta.elementAt(i)%>"><%= (String)Consulta.elementAt(i)%></option><% } %>
                                </select>

```

```

        </td>
      </tr>
    <tr>
      <td align="right"><input type="submit"
value="Registrar campos" style="font-family: Arial Narrow">
      </td>
    </tr>
  </table>
</form>
<% } %>
<form method="POST">
  <table align="center">
    <tr>
      <td><b>Paso 3. Escribe la condición que debe cumplir
la consulta.</b></td>
      <td class="Estilo_4"><input type="text"
name="condicion" size=80></td>
      <td align="right"><input type="submit"
value="Aplicar condición" style="font-family: Arial Narrow"></td>
    </tr>
  </table>
</form>
<form method="POST">
  <table align="center">
    <tr>
      <td><b>Paso 4. Señala que campo ordena la
consulta.</b></td>
      <td class="Estilo_4">
        <% Vector Consulta =
ConsultasPersonalizadas.CamposSeleccionados;
        for (int i=0; i<Consulta.size(); i++)
        { %><input type="radio" cheked name="orden"
value="<%= (String)Consulta.elementAt(i)%>"><%= (String)Consulta.elementAt(i)%><br><% }
%>
        </td>
      <td align="right"><input type="submit"
value="Ordenar consulta" style="font-family: Arial Narrow"></td>
    </tr>
  </table>
</form>
<b>Paso 5. Comprueba los campos y condiciones
elegidas.</b><br><br>
<TABLE>
  <TR>
    <TD>Borra los campos que no te interesen</TD>
    <TD>Borra las condiciones que no te interesen</TD>
    <TD>Ordena la consulta como quieras</TD>
  </TR>
  <TR>
    <TD><font class="Estilo_2" size="10px">
      <FORM METHOD="POST">
        <INPUT TYPE="RADIO" CHECKED NAME="borrar"
VALUE="Campos"> Borrar - campos seleccionados.<br>
        <INPUT TYPE="RADIO" CHECKED NAME="borrar"
VALUE="UltimoCampo"> Borrar - último campo introducido<br>
        <INPUT TYPE="RADIO" CHECKED NAME="borrar"
VALUE="Ninguno"> Elige una opcion <br>
        <INPUT TYPE="SUBMIT" VALUE="Borrar">
      </FORM>
    </TD>
    <TD><font class="Estilo_2" size="10px">
      <FORM METHOD="POST">
        <INPUT TYPE="RADIO" CHECKED NAME="borrar"
VALUE="Condiciones"> Borrar - Condiciones seleccionadas.<br>
        <INPUT TYPE="RADIO" CHECKED NAME="borrar"
VALUE="UltimaCondicion"> Borrar - última condición introducida<br>

```

```

                                <INPUT TYPE="RADIO" CHECKED NAME="borrar"
VALUE="Ninguno"> Elige una opcion <br>
                                <INPUT TYPE="SUBMIT" VALUE="Borrar">
                                </FORM>
                                </font>
                                </TD>
                                <TD><font class="Estilo_2" size="10px">
                                <FORM METHOD="POST">
                                <INPUT TYPE="RADIO" CHECKED NAME="borrar"
VALUE="Orden"> Borrar - Ordenes seleccionados.<br>
                                <INPUT TYPE="RADIO" CHECKED NAME="borrar"
VALUE="UltimoOrden"> Borrar - último orden introducido<br>
                                <INPUT TYPE="RADIO" CHECKED NAME="borrar"
VALUE="Ninguno"> Elige una opcion <br>
                                <INPUT TYPE="SUBMIT" VALUE="Borrar">
                                </FORM>
                                </font>
                                </TD>
                                </TR>
                                <TR>
                                </TABLE>
                                <hr>
                                <form method="POST">
                                <table align="center">
                                <tr>
                                <td><b>Paso 6. Introduce un nombre para la consulta:
</b></td>
                                </tr>
                                <tr>
                                <td class="Estilo_4"><input type="text"
name="NombreConsulta" size=35></td>
                                </tr>
                                <tr>
                                <td align="right"><input type="submit" value="Crear
Consulta" style="font-family: Arial Narrow"></td>
                                </tr>
                                </table>
                                </form>
                                <% }
                                } %>
                                </font>
                                </td>
                                </tr>
                                </table>
                                <br>
                                <!-- Descripcion ===== -->
                                <table width="100%">
                                <tr>
                                <td width="25%"></td>
                                <td width="50%">
                                <blockquote class="Estilo_8">
                                El Servidor de Registro de Acceso será el encargado de registrar
                                en la base de datos "Registro de Acceso" todas las incidencias,
                                accesos e intercambios de información que se produzcan
                                en la arquitectura de seguridad local de un Sistema de Infomación
                                Sanitario.
                                </blockquote>
                                </td>
                                <td width="25%"></td>
                                </tr>
                                </table>
                                <!-- Camino ===== -->
                                <table border="0" width="100%">
                                <tr>
                                <td width="100%" class="Estilo_6">
                                <b>
                                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
                                Bienvenida al cliente.</a> |
                                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
                                Identificación & Autentificación.</a> |
                                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">

```

```
Obtención de Credenciales.</a> |  
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"  
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele  
ccion_de_Recursos.jsp">  
Decisión de Acceso.</a> |  
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"  
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de  
l_Sistema.jsp">  
Administración del Sistema</a> |  
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"  
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Menu_Consultas.  
jsp">  
Consultas</a> |<br>  
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"  
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">  
Cerrar sesión.</a> |  
</b>  
</td>  
</tr>  
</table>  
<%@include file="/Comunes/Pie.jsp" %>
```

9.64. Consultas_Establecidas.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Consultas_Establecidas.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP muestra las consultas diseñadas por el programador.
* =====
*/
%>
<@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Servidor de Registro de Acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
        <br>
        <p align="center" class="Estilo_3">
          <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
          Se ha informado al Servidor de Registro de Acceso.<br><br>
          <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
        </p>
        <br>
        <% }
        else { %>
          <center>
            <h2>ADMINISTRACIÓN DEL SISTEMA</h2>
            <h3>CONSULTAS ESTABLECIDAS</h3>
            <p><b><%= new java.util.Date() %></b></p>
            <TABLE>
              <TR>
                <TD WIDTH="40%">
                  <form method="POST">
                    <table>
                      <tr>
                        <td class="Estilo_4">Consulta el contenido
de las tablas de la base de datos "Registro de Acceso".</td>
                      </tr>
                      <tr>
                        <td>
                          <select class="Estilo_4"
name="Eleccion">
                            <option value="1">Consulta de la tabla:
COMUNICACION</option>

```

```

CREDENCIALES</option>
FORMA_DE_ENVIO</option>
INCIDENCIAS</option>
MOTIVO_DE_DENEGACION</option>
PERSONAL_AUTORIZADO</option>
SITUACION_DE_ACCESO</option>
TIPO_DE_SOPORTE</option>
REGISTRO_DE_ACCESOS</option>
REGISTRO_DE_ENTRADA_DE_SOPORTES_INFORMATICOS</option>
REGISTRO_DE_INCIDENCIAS</option>
REGISTRO_DE_RECUPERACION_DE_DATOS</option>
REGISTRO_DE_SALIDA_DE_SOPORTES_INFORMATICOS</option>
REGISTRO_DE_RESPALDO</option>
RECURSOS</option>
</select>
</td>
</tr>
</table>
<input type="submit" value="Consultar" style="font-
family: Arial Narrow">
</form>
</TD>
<TD WIDTH="25%" ALIGN=left>
<form method="POST">
<table>
<tr>
<td class="Estilo_4">Consultas
seleccionas.</td>
</tr>
<tr>
<td>
<select class="Estilo_4"
<option value="16">Las
<option value="17">Recuperacion de
<option value="18">Entrada de
<option value="19">Salida de
<option value="20">Acceso según
<option value="21">¿Quién ha accedido al
<option value="22">¿Qué fichero se
<option value="23">Ficheros con copia de
</select>
</td>
</tr>
</table>
<input type="submit" value="Consultar" style="font-
family: Arial Narrow">
</form>
</TD>
</TR>
<TR>
<TD WIDTH="40%">
<form method="POST">
<table>

```

```
<tr>
  <td class="Estilo_4">Consulta los registros
de la base de datos por fechas.</td>
  <td class="Estilo_4">Desde:</td>
  <td class="Estilo_4">Hasta:</td>
</tr>
<tr>
  <td>
    name="Eleccion">
    Incidencias</option>
    Accesos</option>
    de Soportes Informáticos</option>
    Soportes Informáticos</option>
  </td>
  <td>
    <select class="Estilo_4"
      <option value="30">Registro de
      <option value="31">Registro de
      <option value="32">Registro de Entrada
      <option value="33">Registro de Salida de
    </select>
  </td>
  <td>
    <select class="Estilo_4"
      <option value="06/30/2005">1 Julio
      <option value="07/1/2005">2 Julio
      <option value="07/2/2005">3 Julio
      <option value="07/3/2005">4 Julio
      <option value="07/4/2005">5 Julio
      <option value="07/5/2005">6 Julio
      <option value="07/6/2005">7 Julio
      <option value="07/7/2005">8 Julio
      <option value="07/8/2005">9 Julio
      <option value="07/9/2005">10 Julio
      <option value="07/10/2005">11 Julio
      <option value="07/11/2005">12 Julio
      <option value="07/12/2005">13 Julio
      <option value="07/13/2005">14 Julio
      <option value="07/14/2005">15 Julio
      <option value="07/15/2005">16 Julio
      <option value="07/16/2005">17 Julio
      <option value="07/17/2005">18 Julio
      <option value="07/18/2005">19 Julio
      <option value="07/19/2005">20 Julio
      <option value="07/20/2005">21 Julio
      <option value="07/21/2005">22 Julio
      <option value="07/22/2005">23 Julio
      <option value="07/23/2005">24 Julio
      <option value="07/24/2005">25 Julio
      <option value="07/25/2005">26 Julio
    </select>
  </td>
</tr>
```

```
2005</option>                <option value="07/26/2005">27 Julio
2005</option>                <option value="07/27/2005">28 Julio
2005</option>                <option value="07/28/2005">29 Julio
2005</option>                <option value="07/29/2005">30 Julio
2005</option>                <option value="07/30/2005">31 Julio
2005</option>                </select>
                                </td>
                                <td>
                                <select class="Estilo_4"
                                <option value="07/2/2005">1 Julio
                                <option value="07/3/2005">2 Julio
                                <option value="07/4/2005">3 Julio
                                <option value="07/5/2005">4 Julio
                                <option value="07/6/2005">5 Julio
                                <option value="07/7/2005">6 Julio
                                <option value="07/8/2005">7 Julio
                                <option value="07/9/2005">8 Julio
                                <option value="07/10/2005">9 Julio
                                <option value="07/11/2005">10 Julio
                                <option value="07/12/2005">11 Julio
                                <option value="07/13/2005">12 Julio
                                <option value="07/14/2005">13 Julio
                                <option value="07/15/2005">14 Julio
                                <option value="07/16/2005">15 Julio
                                <option value="07/17/2005">16 Julio
                                <option value="07/18/2005">17 Julio
                                <option value="07/19/2005">18 Julio
                                <option value="07/20/2005">19 Julio
                                <option value="07/21/2005">20 Julio
                                <option value="07/22/2005">21 Julio
                                <option value="07/23/2005">22 Julio
                                <option value="07/24/2005">23 Julio
                                <option value="07/25/2005">24 Julio
                                <option value="07/26/2005">25 Julio
                                <option value="07/27/2005">26 Julio
                                <option value="07/28/2005">27 Julio
                                <option value="07/29/2005">28 Julio
                                <option value="07/30/2005">29 Julio
                                <option value="07/31/2005">30 Julio
```

```

                <option value="08/01/2005">31 Julio
2005</option>
            </select>
        </td>
    </tr>
</table>
<input type="submit" value="Consultar" style="font-
family: Arial Narrow">
</form>
</TD>
<TD>
</TD>
</TR>
</TABLE>
<% if ( request.getParameter("Eleccion")==null) {}
else { if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),
Convertir.String_Int(request.getParameter("Eleccion"))))
{ ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align=center class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de
Acceso.<br><br>
</p>
<br>
<% }
else { %>
    <area class="Estilo_3" >
    <p><b>Estos son los resultados de la consulta
realizada a la base de datos "Registro de Acceso".</b></p>
    <table class="Estilo_4" border="1">
        <%@include
file="Mostrar_Consultas_Establecidas_Estaticas.jsp"%>
        <%@include
file="Mostrar_Consultas_Establecidas_Dinamicas.jsp"%>
    </table>
    <p class="Estilo_4"> Crear informe de la consulta </p>
    <a
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Mostrar_PDF.jsp
"
        title="Genera el informe de tu consulta"></a>
    </area>
    <%
ControlAcceso.setFicheroAccedido(ConsultasEstablecidas.TextoConsulta(Convertir.String_In
t(request.getParameter("Eleccion"))));
ControlAcceso.setCodigoDenegado("D-000");
ControlAcceso.setURL(request.getServletPath());
ControlAcceso.setCopiaRespaldo(false);
ControlAcceso.setTipoAcceso("Lectura");
ControlAcceso.setCodigoAcceso("A-100");

Despersonalizacion.Acceso(ControlAcceso.getCodigoAcceso()); %>
    <p class="Estilo_3"><%= RegistrarAcceso.Registrar()
%></p>
    <% }
    } %>
</center>
<% } %>
</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
<tr>
    <td width="25%"></td>
    <td width="50%">
        <blockquote class="Estilo_8">
            El Servidor de Registro de Acceso será el encargado de registrar

```

```

        en la base de datos "Registro de Acceso" todas las incidencias,
        accesos e intercambios de información que se produzcan
        en la arquitectura de seguridad local de un Sistema de Infomación
        Sanitario.
    </blockquote>
</td>
<td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" class="Estilo_6">
<b>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
Bienvenida al cliente.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
Identificación & Autenticación.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
Obtención de Credenciales.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
Decisión de Acceso.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
Administración del Sistema</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Menu_Consultas.
jsp">
Consultas</a> |<br>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
Cerrar sesión.</a> |
</b>
</td>
</tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.65. Consultas_Personalizadas.jsp.-

```

<%
/* =====
 * Proyecto final de carrera.
 * Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
 * Autor: Daniel David García Malmierca.
 * Área: Ingeniería Telemática.
 * Departamento: Ingeniería de Sistemas y Automática.
 * Tutora: Isabel Román Martínez.
 * Ingeniería de Telecomunicación - Universidad de Sevilla.
 * =====
 * $Id: Consultas_Personalizadas.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
 * =====
 * Comentarios.-
 * Este JSP muestra las consultas personalizadas creadas con el asistente
 * Configurar_Consultas_Personalizadas.jsp.
 * =====
 */
%>
<@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->


|                                                                                                       |
|-------------------------------------------------------------------------------------------------------|
| <br><p align="center" class="Estilo_2"> <b>&lt;u&gt;Servidor de Registro de Acceso&lt;/u&gt;</b> </p> |
|-------------------------------------------------------------------------------------------------------|

  

<!-- Mensajes y opciones ===== -->


|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <font class="Estilo_7"> <% if (ControlAcceso.getIdUsuario()==null) { ControlIncidencia.setCodigoIncidencia("I-103");  ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%><br><p align="center" class="Estilo_3"> <b>ACCESO NO PERMITIDO: &lt;/b&gt;&lt;br&gt;&lt;br&gt;&lt;%= RegistrarIncidencia.Registrar() %&gt;&lt;br&gt; Se ha informado al Servidor de Registro de Acceso.&lt;br&gt;&lt;br&gt; <a &gt;volver&lt;="" <="" a&gt;="" href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identificacion.jsp" p=""> </a></b></p> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

  

<% }
else { %>
<center>
<h2>ADMINISTRACIÓN DEL SISTEMA</h2>
<h3>CONSULTAS PERSONALIZADAS</h3>
<p><b><%= new java.util.Date() %></b></p>
<% if (request.getParameter("consulta")==null){
else { if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(), ConsultasPersonaliza
das.DevolverNumeroEleccion(request.getParameter("consulta"))))
{ ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
  


ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de
Acceso.<br><br>





<% }

```

```

        else { %>
            <area class="Estilo_3" >
                <%@include file="Mostrar_Consultas_Personalizadas.jsp"
%>
                <p class="Estilo_4"> Crear informe de la consulta </p>
                <a
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Mostrar_PDF.jsp"
                title="Genera el informe de tu consulta"></a>
                </area>
            <% }
        } %>
        <% if (request.getParameter("borrar")==null) {}
        else { if
(! (UsuariosDelServidor.ComprobarAcceso(ControlAcceso.getIdUsuario(),ConsultasPersonaliza
das.DevolverNumeroEleccion(request.getParameter("borrar")))))
            { ControlIncidencia.setCodigoIncidencia("I-105");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
            <br>
            <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de
Acceso.<br><br>
            </p>
            <br>
            <% }
        else { if (
request.getParameter("borrar").contentEquals("Todas"))
        {
ConsultasPersonalizadas.BorrarConsultasAlmacenadas(); }
        else if
(request.getParameter("borrar").contentEquals("Ninguno")) {}
        else {
ConsultasPersonalizadas.BorrarLaConsultaX(request.getParameter("borrar")); }
        } %>
        <form method="POST">
        <table align="center" class="Estilo_4">
            <tr>
                <td><b>Consultas Almacenadas: </b></td>
            </tr>
            <tr>
                <td class="Estilo_4">
                    <% Vector ConsultaNombres =
ConsultasPersonalizadas.NombreConsultasAlmacenadas;
                    Vector ConsultaSQL =
ConsultasPersonalizadas.ConsultasAlmacenadas;
                    for (int i=0; i<ConsultaNombres.size(); i++)
                    { %><input type="radio" name="consulta"
value="<%= (String)ConsultaSQL.elementAt(i)%>"><%=
(String)ConsultaNombres.elementAt(i)%><br><% } %>
                    </td>
                <td align="right"><input type="submit" value="Ver consulta"
style="font-family: Arial Narrow"></td>
            </tr>
        </table>
        </form>

        <br>
        <table align=center class="Estilo_4">
            <tr>
                <td><b>Borrar Consultas Almacenadas: </b></td>
            </tr>
            <tr>
                <td>
                    <font class="Estilo_2" size="10px">
                    <form method="post">
                        <input type="radio" checked name="borrar" value="Todas">
Borrar todas las consultas almacenadas.<br>
                        <% ConsultaNombres =
ConsultasPersonalizadas.NombreConsultasAlmacenadas;

```

```

ConsultasPersonalizadas.ConsultasAlmacenadas;
        ConsultasSQL =
        for (int i=0; i<ConsultaNombres.size(); i++)
        { %><input type="radio" name="borrar" value="<%=
        (String)ConsultaSQL.elementAt(i) %>">Borrar la consulta : <%=
        (String)ConsultaNombres.elementAt(i)%><br><%= } %>
        <input type="radio" checked name="borrar"
        value="Ninguno"> Elige una opcion <br>
        <input type="submit" value="Borrar">
        </form>
        </font>
        </td>
        </tr>
    </table>
    <% } %>
</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
<tr>
<td width="25%"></td>
<td width="50%">
<blockquote class="Estilo_8">
El Servidor de Registro de Acceso será el encargado de registrar
en la base de datos "Registro de Acceso" todas las incidencias,
accesos e intercambios de información que se produzcan
en la arquitectura de seguridad local de un Sistema de Infomación
Sanitario.
</blockquote>
</td>
<td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" class="Estilo_6">
<b>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
Bienvenida al cliente.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
Identificación & Autenticación.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
Obtención de Credenciales.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
Decisión de Acceso.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
Administración del Sistema</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Menu_Consultas.
jsp">
Consultas</a> |<br>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
Cerrar sesión.</a> |
</b>
</td>
</tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.66. Mostrar_Consultas_Establecidas_Estaticas.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Mostrar_Consultas_Establecidas_Estaticas.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP muestra el resultado de las consultas estáticas diseñadas por el
* programador.
* =====
*/
%>
<%
if (
(request.getParameter("Eleccion")==null)|(!(Convertir.String_Int(request.getParameter("E
leccion"))<30))) {}
else { Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(Convertir.String_Int(request.getParameter("Eleccion")));
    ParametrosDeConsulta mostrar = null;
    //En este caso utilizamos el nombre de los campos como nombres de cabecera.
    Podemos pasar un vector con
    //los nombres que queramos.
    ParametrosPDF.setNombresCabecera(ParametrosPDF.getNombresCampos());
    ParametrosPDF.setSesion(request.getSession()); %>
    <tr>
    <% // Mostramos la cabecera de la consulta.
        Vector cabecera = ParametrosPDF.getNombresCabecera();
        for (int c=0 ; c<cabecera.size(); c++)
            { %><td bgcolor="#808080" align="center"><font color="#FFFFFF"><%=
cabecera.elementAt(c).toString()%></font></td><% } %>
        </tr>
    <% // Mostramos el resultado de la consulta.
        if ( (request.getParameter("Eleccion").contentEquals("1")) |
(request.getParameter("Eleccion").contentEquals("3")) |
            (request.getParameter("Eleccion").contentEquals("5")) |
(request.getParameter("Eleccion").contentEquals("7")) |
            (request.getParameter("Eleccion").contentEquals("8")) |
(request.getParameter("Eleccion").contentEquals("14")) |
            (request.getParameter("Eleccion").contentEquals("15")))
            { for (int i=0; i<ConsultaEstatica.size();i++)
                { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i); %>
                <tr>
                    <td align="center"> <%= mostrar.getParametro1() %> </td>
                    <td align="center"> <%= mostrar.getParametro2() %> </td>
                </tr>
                <% } %>
            }
        else if ( (request.getParameter("Eleccion").contentEquals("2")) |
(request.getParameter("Eleccion").contentEquals("4")))
            { for (int i=0; i<ConsultaEstatica.size();i++)
                { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
                <tr>
                    <td align="center"> <%= mostrar.getParametro1() %> </td>
                    <td align="center"> <%= mostrar.getParametro2() %> </td>
                    <td align="center"> <%= mostrar.getParametro3() %> </td>
                </tr>
                <% } %>
            }
        else if ( request.getParameter("Eleccion").contentEquals("6") )
            { for (int i=0; i<ConsultaEstatica.size();i++)
                { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
                <tr>
                    <td align="center"> <%= mostrar.getParametro1() %> </td>
                    <td align="center"> <%= mostrar.getParametro2() %> </td>
                    <td align="center"> <%= mostrar.getParametro3() %> </td>
                    <td align="center"> <%= mostrar.getParametro4() %> </td>
                </tr>
            }
        }
    }
}

```

```

        <% } %>
    <% }
else if ( request.getParameter("Eleccion").contentEquals("12") )
    {for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
            <tr>
                <td align="center"> <%= mostrar.getParametro1() %> </td>
                <td align="center"> <%= mostrar.getParametro2() %> </td>
                <td align="center"> <%= mostrar.getParametro3() %> </td>
                <td align="center"> <%= mostrar.getParametro4() %> </td>
                <td align="center"> <%= mostrar.getParametro5() %> </td>
            </tr>
        <% } %>
    <% }
else if ( request.getParameter("Eleccion").contentEquals("23"))
    {for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
            <tr>
                <td align="center"> <%= mostrar.getParametro1() %> </td>
                <td align="center"> <%= mostrar.getParametro2() %> </td>
                <td align="center"> <%= mostrar.getParametro3() %> </td>
                <td align="center"> <%= mostrar.getParametro4() %> </td>
                <td align="center"> <%= mostrar.getParametro5() %> </td>
                <td align="center"> <%= mostrar.getParametro6() %> </td>
            </tr>
        <% } %>
    <% }
else if (
(request.getParameter("Eleccion").contentEquals("11"))|(request.getParameter("Eleccion")
.contentEquals("16"))|
(request.getParameter("Eleccion").contentEquals("17")))
    { for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
            <tr>
                <td align="center"> <%= mostrar.getParametro1() %> </td>
                <td align="center"> <%= mostrar.getParametro2() %> </td>
                <td align="center"> <%= mostrar.getParametro3() %> </td>
                <td align="center"> <%= mostrar.getParametro4() %> </td>
                <td align="center"> <%= mostrar.getParametro5() %> </td>
                <td align="center"> <%= mostrar.getParametro6() %> </td>
                <td align="center"> <%= mostrar.getParametro7() %> </td>
            </tr>
        <% } %>
    <% }
else if (
(request.getParameter("Eleccion").contentEquals("18"))|(request.getParameter("Eleccion")
.contentEquals("19"))|
(request.getParameter("Eleccion").contentEquals("20")))
    { for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
            <tr>
                <td align="center"> <%= mostrar.getParametro1() %> </td>
                <td align="center"> <%= mostrar.getParametro2() %> </td>
                <td align="center"> <%= mostrar.getParametro3() %> </td>
                <td align="center"> <%= mostrar.getParametro4() %> </td>
                <td align="center"> <%= mostrar.getParametro5() %> </td>
                <td align="center"> <%= mostrar.getParametro6() %> </td>
                <td align="center"> <%= mostrar.getParametro7() %> </td>
                <td align="center"> <%= mostrar.getParametro8() %> </td>
            </tr>
        <% } %>
    <% }
else if ( request.getParameter("Eleccion").contentEquals("21"))
    { for (int i=0; i<ConsultaEstatica.size();i++)
        { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
            <tr>
                <td align="center"> <%= mostrar.getParametro1() %> </td>
                <td align="center"> <%= mostrar.getParametro2() %> </td>
                <td align="center"> <%= mostrar.getParametro3() %> </td>
                <td align="center"> <%= mostrar.getParametro4() %> </td>
                <td align="center"> <%= mostrar.getParametro5() %> </td>
                <td align="center"> <%= mostrar.getParametro6() %> </td>
                <td align="center"> <%= mostrar.getParametro7() %> </td>
                <td align="center"> <%= mostrar.getParametro8() %> </td>
                <td align="center"> <%= mostrar.getParametro9() %> </td>
            </tr>
    }
}

```

```

        <% } %>
    <% }
    else if (
(request.getParameter("Eleccion").contentEquals("10"))|(request.getParameter("Eleccion")
.contentEquals("13"))) )
        { for (int i=0; i<ConsultaEstatica.size();i++)
            { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
                <tr>
                    <td align="center"> <%= mostrar.getParametro1() %> </td>
                    <td align="center"> <%= mostrar.getParametro2() %> </td>
                    <td align="center"> <%= mostrar.getParametro3() %> </td>
                    <td align="center"> <%= mostrar.getParametro4() %> </td>
                    <td align="center"> <%= mostrar.getParametro5() %> </td>
                    <td align="center"> <%= mostrar.getParametro6() %> </td>
                    <td align="center"> <%= mostrar.getParametro7() %> </td>
                    <td align="center"> <%= mostrar.getParametro8() %> </td>
                    <td align="center"> <%= mostrar.getParametro9() %> </td>
                    <td align="center"> <%= mostrar.getParametro10() %> </td>
                </tr>
            <% } %>
        <% }
    else if ( request.getParameter("Eleccion").contentEquals("22") )
        { for (int i=0; i<ConsultaEstatica.size();i++)
            { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
                <tr>
                    <td align="center"> <%= mostrar.getParametro1() %> </td>
                    <td align="center"> <%= mostrar.getParametro2() %> </td>
                    <td align="center"> <%= mostrar.getParametro3() %> </td>
                    <td align="center"> <%= mostrar.getParametro4() %> </td>
                    <td align="center"> <%= mostrar.getParametro5() %> </td>
                    <td align="center"> <%= mostrar.getParametro6() %> </td>
                    <td align="center"> <%= mostrar.getParametro7() %> </td>
                    <td align="center"> <%= mostrar.getParametro8() %> </td>
                    <td align="center"> <%= mostrar.getParametro9() %> </td>
                    <td align="center"> <%= mostrar.getParametro10() %> </td>
                    <td align="center"> <%= mostrar.getParametro11() %> </td>
                    <td align="center"> <%= mostrar.getParametro12() %> </td>
                </tr>
            <% } %>
        <% }
    else if ( request.getParameter("Eleccion").contentEquals("9") )
        { for (int i=0; i<ConsultaEstatica.size();i++)
            { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
                <tr>
                    <td align="center"> <%= mostrar.getParametro1() %> </td>
                    <td align="center"> <%= mostrar.getParametro2() %> </td>
                    <td align="center"> <%= mostrar.getParametro3() %> </td>
                    <td align="center"> <%= mostrar.getParametro4() %> </td>
                    <td align="center"> <%= mostrar.getParametro5() %> </td>
                    <td align="center"> <%= mostrar.getParametro6() %> </td>
                    <td align="center"> <%= mostrar.getParametro7() %> </td>
                    <td align="center"> <%= mostrar.getParametro8() %> </td>
                    <td align="center"> <%= mostrar.getParametro9() %> </td>
                    <td align="center"> <%= mostrar.getParametro10() %> </td>
                    <td align="center"> <%= mostrar.getParametro11() %> </td>
                    <td align="center"> <%= mostrar.getParametro12() %> </td>
                    <td align="center"> <%= mostrar.getParametro13() %> </td>
                    <td align="center"> <%= mostrar.getParametro14() %> </td>
                    <td align="center"> <%= mostrar.getParametro15() %> </td>
                    <td align="center"> <%= mostrar.getParametro16() %> </td>
                    <td align="center"> <%= mostrar.getParametro17() %> </td>
                    <td align="center"> <%= mostrar.getParametro18() %> </td>
                </tr>
            <% } %>
        <% }
    } %>

```

9.67. Mostrar_Consultas_Establecidas_Dinamicas.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Mostrar_Consultas_Establecidas_Dinamicas.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP muestra el resultado de las consultas dinámicas diseñadas por el
* programador.
* =====
*/
%>
<%
    if ( request.getParameter("Eleccion")==null) {}
    else if ( // Mostramos el resultado de la consulta.

(request.getParameter("Eleccion").contentEquals("30"))|(request.getParameter("Eleccion")
.contentEquals("31"))|

(request.getParameter("Eleccion").contentEquals("32"))|(request.getParameter("Eleccion")
.contentEquals("33")))
        { Vector ConsultaEstatica =
ConsultasEstablecidas.Dinamicas(request.getParameter("Fecha_Inferior"),
request.getParameter("Fecha_Superior"),

Convertir.String_Int(request.getParameter("Eleccion")));
        ParametrosDeConsulta mostrar = null;
        //En este caso utilizamos el nombre de los campos como nombres de cabecera.
Podemos pasar un vector con
        //los nombres que queramos.
        ParametrosPDF.setNombresCabecera(ParametrosPDF.getNombresCampos());
        ParametrosPDF.setSesion(request.getSession()); %>
        <tr>
        <% // Mostramos la cabecera de la consulta.
        Vector cabecera = ParametrosPDF.getNombresCabecera();
        for (int c=0 ; c<cabecera.size(); c++)
            { %><td bgcolor="#808080" align="center"><font color="#FFFFFF"><%=
cabecera.elementAt(c).toString()%></font></td><% } %>
        </tr>
        <% for (int i=0; i<ConsultaEstatica.size();i++)
            { mostrar = (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
                <tr>
                    <td align="center"> <%= mostrar.getParametro1() %> </td>
                    <td align="center"> <%= mostrar.getParametro2() %> </td>
                    <td align="center"> <%= mostrar.getParametro3() %> </td>
                    <td align="center"> <%= mostrar.getParametro4() %> </td>
                    <td align="center"> <%= mostrar.getParametro5() %> </td>
                </tr>
            } %>
        <% } %>
    } %>

```

9.68. Mostrar_Consultas_Personalizadas.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Mostrar_Consultas_Personalizadas.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP muestra el resultado de las consultas personalizadas.
* =====
*/
%>
<% Vector Respuesta =
ConsultasPersonalizadas.MostrarConsultaSQL(request.getParameter("consulta"));

    if (Respuesta==null)
    { %> <p class="Estilo_3">El diseño de la consulta es incorrecto, compruebe la
condición SQL introducida.; <% }

    else { ParametrosDeConsulta mostrar = null;
ParametrosPDF.setNombresCabecera(ParametrosPDF.getNombresCampos());
ParametrosPDF.setSesion(request.getSession()); %>
<p><b>Estos son los resultados de la consulta realizada a la base de datos
"Registro de Acceso".</b></p>
<p align=center class="Estilo_2">
Consulta: <b><%=
ConsultasPersonalizadas.DevolverNombreConsultaAlmacenada(request.getParameter("consulta"
)) %></b><br>
Recurso del servidor: número <%=
ConsultasPersonalizadas.DevolverNumeroEleccion(request.getParameter("consulta")) %>
</p>
<table class="Estilo_4" border="1">
<tr>
<% // Mostramos la cabecera de la consulta.
Vector cabecera = ParametrosPDF.getNombresCabecera();
for (int c=0 ; c<cabecera.size(); c++)
    { %><td bgcolor="#808080" align="center"><font color="#FFFFFF"><%=
cabecera.elementAt(c).toString()></font></td><% } %>
</tr>
<% if (cabecera.size())<2>
    {
        for (int i=0; i<Respuesta.size();i++)
        { mostrar = (ParametrosDeConsulta)Respuesta.elementAt(i);%>
<tr>
<td align="center"> <%= mostrar.getParametro1() %> </td>
</tr>

<% }
    }
    else if (cabecera.size())<3>
    {
        for (int i=0; i<Respuesta.size();i++)
        { mostrar = (ParametrosDeConsulta)Respuesta.elementAt(i);%>
<tr>
<td align="center"> <%= mostrar.getParametro1() %> </td>
<td align="center"> <%= mostrar.getParametro2() %> </td>
</tr>

<% }
    }
    else if (cabecera.size())<4>
    {
        for (int i=0; i<Respuesta.size();i++)
        { mostrar = (ParametrosDeConsulta)Respuesta.elementAt(i);%>
<tr>
<td align="center"> <%= mostrar.getParametro1() %> </td>
<td align="center"> <%= mostrar.getParametro2() %> </td>
<td align="center"> <%= mostrar.getParametro3() %> </td>
</tr>

<% }
    }
}
}

```

```

else if (cabecera.size()<5)
{
    for (int i=0; i<Respuesta.size();i++)
    {
        mostrar = (ParametrosDeConsulta)Respuesta.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1() %> </td>
            <td align="center"> <%= mostrar.getParametro2() %> </td>
            <td align="center"> <%= mostrar.getParametro3() %> </td>
            <td align="center"> <%= mostrar.getParametro4() %> </td>
        </tr>

        <% }
    }
}
else if (cabecera.size()<6)
{
    for (int i=0; i<Respuesta.size();i++)
    {
        mostrar = (ParametrosDeConsulta)Respuesta.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1() %> </td>
            <td align="center"> <%= mostrar.getParametro2() %> </td>
            <td align="center"> <%= mostrar.getParametro3() %> </td>
            <td align="center"> <%= mostrar.getParametro4() %> </td>
            <td align="center"> <%= mostrar.getParametro5() %> </td>
        </tr>

        <% }
    }
}
else if (cabecera.size()<7)
{
    for (int i=0; i<Respuesta.size();i++)
    {
        mostrar = (ParametrosDeConsulta)Respuesta.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1() %> </td>
            <td align="center"> <%= mostrar.getParametro2() %> </td>
            <td align="center"> <%= mostrar.getParametro3() %> </td>
            <td align="center"> <%= mostrar.getParametro4() %> </td>
            <td align="center"> <%= mostrar.getParametro5() %> </td>
            <td align="center"> <%= mostrar.getParametro6() %> </td>
        </tr>

        <% }
    }
}
else if (cabecera.size()<8)
{
    for (int i=0; i<Respuesta.size();i++)
    {
        mostrar = (ParametrosDeConsulta)Respuesta.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1() %> </td>
            <td align="center"> <%= mostrar.getParametro2() %> </td>
            <td align="center"> <%= mostrar.getParametro3() %> </td>
            <td align="center"> <%= mostrar.getParametro4() %> </td>
            <td align="center"> <%= mostrar.getParametro5() %> </td>
            <td align="center"> <%= mostrar.getParametro6() %> </td>
            <td align="center"> <%= mostrar.getParametro7() %> </td>
        </tr>

        <% }
    }
}
else if (cabecera.size()<9)
{
    for (int i=0; i<Respuesta.size();i++)
    {
        mostrar = (ParametrosDeConsulta)Respuesta.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1() %> </td>
            <td align="center"> <%= mostrar.getParametro2() %> </td>
            <td align="center"> <%= mostrar.getParametro3() %> </td>
            <td align="center"> <%= mostrar.getParametro4() %> </td>
            <td align="center"> <%= mostrar.getParametro5() %> </td>
            <td align="center"> <%= mostrar.getParametro6() %> </td>
            <td align="center"> <%= mostrar.getParametro7() %> </td>
            <td align="center"> <%= mostrar.getParametro8() %> </td>
        </tr>

        <% }
    }
}
else if (cabecera.size()<10)
{
    for (int i=0; i<Respuesta.size();i++)
    {
        mostrar = (ParametrosDeConsulta)Respuesta.elementAt(i);%>
        <tr>
            <td align="center"> <%= mostrar.getParametro1() %> </td>
            <td align="center"> <%= mostrar.getParametro2() %> </td>
            <td align="center"> <%= mostrar.getParametro3() %> </td>
            <td align="center"> <%= mostrar.getParametro4() %> </td>
            <td align="center"> <%= mostrar.getParametro5() %> </td>
            <td align="center"> <%= mostrar.getParametro6() %> </td>
        </tr>
    }
}

```

[illegible]

[illegible]

```

        <td align="center"> <%= mostrar.getParametro2() %> </td>
        <td align="center"> <%= mostrar.getParametro3() %> </td>
        <td align="center"> <%= mostrar.getParametro4() %> </td>
        <td align="center"> <%= mostrar.getParametro5() %> </td>
        <td align="center"> <%= mostrar.getParametro6() %> </td>
        <td align="center"> <%= mostrar.getParametro7() %> </td>
        <td align="center"> <%= mostrar.getParametro8() %> </td>
        <td align="center"> <%= mostrar.getParametro9() %> </td>
        <td align="center"> <%= mostrar.getParametro10() %> </td>
        <td align="center"> <%= mostrar.getParametro11() %> </td>
        <td align="center"> <%= mostrar.getParametro12() %> </td>
        <td align="center"> <%= mostrar.getParametro13() %> </td>
        <td align="center"> <%= mostrar.getParametro14() %> </td>
        <td align="center"> <%= mostrar.getParametro15() %> </td>
        <td align="center"> <%= mostrar.getParametro16() %> </td>
        <td align="center"> <%= mostrar.getParametro17() %> </td>
    </tr>
    <% }
    }
    else if (cabecera.size()<19)
    {    for (int i=0; i<Respuesta.size();i++)
        {    mostrar = (ParametrosDeConsulta)Respuesta.elementAt(i);%>
            <tr>
            <td align="center"> <%= mostrar.getParametro1() %> </td>
            <td align="center"> <%= mostrar.getParametro2() %> </td>
            <td align="center"> <%= mostrar.getParametro3() %> </td>
            <td align="center"> <%= mostrar.getParametro4() %> </td>
            <td align="center"> <%= mostrar.getParametro5() %> </td>
            <td align="center"> <%= mostrar.getParametro6() %> </td>
            <td align="center"> <%= mostrar.getParametro7() %> </td>
            <td align="center"> <%= mostrar.getParametro8() %> </td>
            <td align="center"> <%= mostrar.getParametro9() %> </td>
            <td align="center"> <%= mostrar.getParametro10() %> </td>
            <td align="center"> <%= mostrar.getParametro11() %> </td>
            <td align="center"> <%= mostrar.getParametro12() %> </td>
            <td align="center"> <%= mostrar.getParametro13() %> </td>
            <td align="center"> <%= mostrar.getParametro14() %> </td>
            <td align="center"> <%= mostrar.getParametro15() %> </td>
            <td align="center"> <%= mostrar.getParametro16() %> </td>
            <td align="center"> <%= mostrar.getParametro17() %> </td>
            <td align="center"> <%= mostrar.getParametro18() %> </td>
            </tr>
            <% }
            } %>
        } %>
    </table>
    <% } %>

```

9.69. Mostrar_PDF.jsp.-

```

/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Mostrar_PDF.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* Este JSP muestra en pantalla el informe PDF.
* =====
*/

%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Servidor de Registro de Acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
  <font class="Estilo_7">
    <% if (ControlAcceso.getIdUsuario()==null)
      { ControlIncidencia.setCodigoIncidencia("I-103");
ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
      <br>
      <p align=center class="Estilo_3">
        <b> ACCESO NO PERMITIDO: </b><br><br><br>%=
RegistrarIncidencia.Registrar() %><br>
        Se ha informado al Servidor de Registro de Acceso.<br><br>
        <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
        </p>
        <br>
        <% }
      else { %>
        <center>
          <h2>ADMINISTRACIÓN DEL SISTEMA</h2>
          <h3>INFORME</h3>
          <p><b><%= new java.util.Date() %></b></p>
          <% GenerarPDF.generarPDF();
            ParametrosPDF.Clear(); %>
          <iframe src="C:\Servidor de Registro de
Acceso\web\Temporales\Informe.pdf" width="80%" height="80%">
          </iframe>
          </center>
        <% } %>
      </font>
<br>
<!-- Descripcion ===== -->
<table width="100%">
  <tr>
    <td width="25%"></td>
    <td width="50%">
      <blockquote class="Estilo_8">
        El Servidor de Registro de Acceso será el encargado de registrar
        en la base de datos "Registro de Acceso" todas las incidencias,
        accesos e intercambios de información que se produzcan
        en la arquitectura de seguridad local de un Sistema de Infomación

```

```

        Sanitario.
    </blockquote>
</td>
<td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" class="Estilo_6">
<b>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
Bienvenida al cliente.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
Identificación & Autentificación.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
Obtención de Credenciales.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
Decisión de Acceso.</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
Administración del Sistema</a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Consultas/Menu_Consultas.
jsp">
Consultas</a> |<br>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
Cerrar sesión.</a> |
</b>
</td>
</tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.70. Menu_Simulaciones.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Menu_Simulaciones.jsp, v 1.0, 12 de Abril de 2005, 17:12 $
* =====
* Comentarios.-
* Este JSP presenta las opciones de simulación que verifican el funcionamiento
* del servidor.
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante("Servidor de Registro de Acceso"); %>
<!-- Titulo página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Servidor de Registro de Acceso</u></b>
</p>
</td>
</tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if (ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align="center" class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<br>
<% }
else { %>
<center>
<h2>ADMINISTRACIÓN DEL SISTEMA</h2>
<h3>SIMULACIONES</h3>
<p><b><%= new java.util.Date() %></b></p>
<p>Elija entre las siguientes opciones:</p>
<table>
<ol>
<tr>
<td><a class="Estilo_3"
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Simulaciones/Registrar_ES
I.jsp'>
Entrada de Soportes Informáticos.</a>
</td>
</tr>
<tr>
<td><a class="Estilo_3"
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Simulaciones/Registrar_SS
I.jsp'>
Salida de Soportes Informáticos.</a>

```

```

        </td>
      </tr>
      <tr>
        <td><a class="Estilo_3"
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Simulaciones/Recuperacion
_de_Datos.jsp'>
          Recuperación de Datos.</a>
        </td>
      </tr>
    </ol>
  </table>
</center>
<% } %>
</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
  <tr>
    <td width="25%"></td>
    <td width="50%">
      <blockquote class="Estilo_8">
        El Servidor de Registro de Acceso será el encargado de registrar
        en la base de datos "Registro de Acceso" todas las incidencias,
        accesos e intercambios de información que se produzcan
        en la arquitectura de seguridad local de un Sistema de Infomación
        Sanitario.
      </blockquote>
    </td>
    <td width="25%"></td>
  </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%" class="Estilo_6">
      <b>
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
        Bienvenida al cliente. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
        Identificación & Autenticación. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
        Obtención de Credenciales. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
        Decisión de Acceso. </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
        Administración del Sistema </a> |
        | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Simulaciones/Menu_Simulac
iones.jsp">
        Simulaciones </a> |
      </b>
    </td>
  </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.71. Recuperacion_de_Datos.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Recuperacion_de_Datos.jsp, v 1.0, 12 de Abril de 2005, 17:12 $
* =====
* Comentarios.-
* Este JSP simula la recuperación de datos almacenados.
* Es por tanto en si un recurso más. Realizado para mostrar la opción de
* recuperación de datos, es el único recurso en el que una misma acción debe ser
* indicada en dos registros de la base de datos. Según el Real Decreto 994/1999
* la recuperación de datos debe ser registrada como una incidencia de seguridad.
* =====
*/
%>

<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante ("Servidor de Registro de Acceso"); %>
<!-- Título página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Servidor de Registro de Acceso</u></b>
</p>
</td>
</tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<font class="Estilo_7">
<% if (ControlAcceso.getIdUsuario()==null)
{ ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
<br>
<p align="center" class="Estilo_3">
<b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
Se ha informado al Servidor de Registro de Acceso.<br><br>
<a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
</p>
<br>
<% }
else { %>
<center>
<h2>SIMULACIÓN DE RECUPERACION DE DATOS</h2>
<p><b><%= new java.util.Date() %></b></p>
<%%// PARA REGISTRAR LA INCIDENCIA DE LA RECUPERACIÓN DE DATOS
ControlRecuperacionDatos.setDatos("Copia_Seguridad.rar");
ControlRecuperacionDatos.setDatosManuales("Ninguno");
ControlRecuperacionDatos.setRecuperacionAutorizada(true);
// PARA REGISTRAR TAMBIEN EL ACCESO AL RECURSO.-
ControlAcceso.setCodigoDenegado("D-000");

ControlAcceso.setFicheroAccedido(ControlRecuperacionDatos.getDatos());
ControlAcceso.setURL(request.getServletPath());
ControlAcceso.setCopiaRespaldo(true);
ControlCopiaRespaldo.setFechaCopiaRespaldo("03/28/2005");
ControlAcceso.setTipoAcceso("Escritura");

```

```

ControlAcceso.setCodigoAcceso("A-100");
ControlAcceso.setDespersonalizacion(false);
ControlIncidencia.setCodigoIncidencia("I-600");
ControlIncidencia.setCodigoComunicacion("C-102"); %>
<p class="Estilo_3"><%= RegistrarIncidencia.Registrar() %><br><br>
<%= RegistrarAcceso.Registrar() %>

</p>
</center>

<% } %>

</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
<tr>
<td width="25%"></td>
<td width="50%">
<blockquote class="Estilo_8">
Simulación de Recuperación de Datos:<br>
Según el Real Decreto 994/1999 la recuperación de datos debe ser
registrada como una incidencia de seguridad.
</blockquote>
</td>
<td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" class="Estilo_6">
<b>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
Bienvenida al cliente. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
Identificación & Autenticación. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
Obtención de Credenciales. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
Decisión de Acceso. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
Administración del Sistema </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Simulaciones/Menu_Simulac
iones.jsp">
Simulaciones </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
Cerrar sesión. </a> |
</b>
</td>
</tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.72. Registrar_ESI.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Registrar_ESI.jsp, v 1.0, 12 de Abril de 2005, 19:36 $
* =====
* Comentarios.-
* Este JSP simula la entrada de soportes informáticos al Sistema de Información
* Sanitario.
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante ("Servidor de Registro de Acceso"); %>
<!-- Titulo página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Servidor de Registro de Acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
          <br>
          <p align=center class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
          </p>
          <br>
          <% }
          else { %>
            <center>
              <h2>SIMULACIÓN DE ENTRADA DE SOPORTES INFORMÁTICOS</h2>
              <p><b><%= new java.util.Date() %></b></p>
              <% if (request.getParameter("CodigoSoporte")==null) { %> <p
align="center"> Elija entre las distintas opciones. </p><% }
              else {
ControlSoportesInformaticos.setCodigoSoporte(request.getParameter("CodigoSoporte")); }
              if (request.getParameter("CodigoEnvio")==null) {}
              else {
ControlSoportesInformaticos.setCodigoEnvio(request.getParameter("CodigoEnvio")); }
              if (request.getParameter("NumeroSoportes")==null) {}
              else {
ControlSoportesInformaticos.setEmisor(request.getRemoteHost());

ControlSoportesInformaticos.setReceptor(request.getServerName());
ControlSoportesInformaticos.setAutorizacionEscrita(true);

ControlSoportesInformaticos.setNumeroSoportes(Convertir.String_Int(request.getParameter(
"NumeroSoportes"))));%>

```

```

        <p align=center class="Estilo_3"><%=
RegistrarEntradaSoportesInformaticos.Registrar()%></p>
        <% ControlSoportesInformaticos.Clear(); %>
    <% } %>
    <!-- Formulario1 ===== -->
    <center>
    <form method="POST">
    <table>
    <tr>
    <td><p class="Estilo_4">1.Elige un tipo de soporte informático:
</td>
    <td></td>
    </tr>
    <tr>
    <td>
        <select class="Estilo_4" name="CodigoSoporte">
        <% Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(8);
        ParametrosDeConsulta mostrar = null;
        for (int i=0;i<ConsultaEstatica.size();i++)
        { mostrar =
        (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <option value="<%= mostrar.getParametro1() %%"><%=
mostrar.getParametro2() %"></option>
        <% } %>
        </select>
    </td>
    <td><input type="submit" value="Selección de soporte"
style="font-family: Arial Narrow"></td>
    </tr>
    </table>
    <p class="Estilo_4"><b>Código de soporte informático seleccionado: </b>
<%= ControlSoportesInformaticos.getCodigoSoporte() %></p>
    </form>
    <!-- Formulario2 ===== -->
    <form method="POST">
    <table>
    <tr>
    <td><p class="Estilo_4">2.Elige una forma de envío: </td>
    <td></td>
    </tr>
    <tr>
    <td>
        <select class="Estilo_4" name="CodigoEnvio">
        <% ConsultaEstatica = ConsultasEstablecidas.Estaticas(3);
        mostrar = null;
        for (int i=0;i<ConsultaEstatica.size();i++)
        { mostrar =
        (ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <option value="<%= mostrar.getParametro1() %%"><%=
mostrar.getParametro2() %"></option>
        <% } %>
        </select>
    </td>
    <td><input type="submit" value="Selección de envío" style="font-
family: Arial Narrow"></td>
    </tr>
    </table>
    <p class="Estilo_4"><b>Código de envío seleccionado: </b> <%=
ControlSoportesInformaticos.getCodigoEnvio() %></p>
    </form>
    <!-- Formulario3 ===== -->
    <form method="POST">
    <table>
    <tr>
    <td><p class="Estilo_4">3.Selecciona el número de soportes
informáticos: </td>
    <td></td>
    </tr>
    <tr>
    <td>
        <select class="Estilo_4" name="NumeroSoportes">
        <% for (int i=1;i<1001;i++)
        { %><option value="<%= i %%"><%= i %"></option><% } %>
        </select>
    </td>

```

```

        <td><input type="submit" value="Registrar Entrada" style="font-
family: Arial Narrow"></td>
    </tr>
</table>
</form>
</center>
<% } %>
</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
<tr>
<td width="25%"></td>
<td width="50%">
<blockquote class="Estilo_8">
Simulación de Entrada de Soportes Informáticos:<br>
La entrada de información al dominio de seguridad deberá ser registrada.
</blockquote>
</td>
<td width="25%"></td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" class="Estilo_6">
<b>
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
Bienvenida al cliente. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
Identificación & Autentificación. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
Obtención de Credenciales. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
Decisión de Acceso. </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
Administración del Sistema </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Simulaciones/Menu_Simulac
iones.jsp">
Simulaciones </a> |
| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
Cerrar sesión. </a> |
</b>
</td>
</tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.73. Registrar_SSI.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Registrar_SSI.jsp, v 1.0, 12 de Abril de 2005, 19:36 $
* =====
* Comentarios.-
* Este JSP simula la salida de soportes informáticos del Sistema de Información
* Sanitario.
* =====
*/
%>
<%@page import="java.util.*, java.io.*, ars.*, recurso.*, cliente.*, java.sql.*" %>
<%@include file="/Comunes/Encabezado.jsp" %>
<% ControlIncidencia.setNotificante ("Servidor de Registro de Acceso"); %>
<!-- Titulo página ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <br>
      <p align="center" class="Estilo_2">
        <b><u>Servidor de Registro de Acceso</u></b>
      </p>
    </td>
  </tr>
</table>
<br>
<!-- Mensajes y opciones ===== -->
<table border="0" width="100%">
  <tr>
    <td width="100%">
      <font class="Estilo_7">
        <% if (ControlAcceso.getIdUsuario()==null)
          { ControlIncidencia.setCodigoIncidencia("I-103");

ControlIncidencia.setCodigoComunicacion(Relacion.Incidencia_Comunicacion());%>
          <br>
          <p align="center" class="Estilo_3">
            <b> ACCESO NO PERMITIDO: </b><br><br><%=
RegistrarIncidencia.Registrar() %><br>
            Se ha informado al Servidor de Registro de Acceso.<br><br>
            <a
href='http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp'>Volver</a>
          </p>
          <br>
          <% }
          else { %>
            <center>
              <h2>SIMULACIÓN DE SALIDA DE SOPORTES INFORMÁTICOS</h2>
              <p><b><%= new java.util.Date() %></b></p>
              <% if (request.getParameter("CodigoSoporte")==null) { %> <p
align="center"> Elija entre las distintas opciones. </p><% }
              else {
ControlSoportesInformaticos.setCodigoSoporte(request.getParameter("CodigoSoporte")); }
              if (request.getParameter("CodigoEnvio")==null) {}
              else {
ControlSoportesInformaticos.setCodigoEnvio(request.getParameter("CodigoEnvio")); }
              if (request.getParameter("NumeroSoportes")==null) {}
              else {
ControlSoportesInformaticos.setEmisor(request.getServerName());

ControlSoportesInformaticos.setReceptor(request.getRemoteHost());
ControlSoportesInformaticos.setAutorizacionEscrita(true);

ControlSoportesInformaticos.setNumeroSoportes(Convertir.String_Int(request.getParameter(
"NumeroSoportes"))));%>

```

```

        <p align=center class="Estilo_3"><%=
RegistrarSalidaSoportesInformaticos.Registrar() %></p>
        <% ControlSoportesInformaticos.Clear(); %>
    <% } %>
    <!-- Formulario1 ===== -->
    <center>
    <form method="POST">
    <table>
    <tr>
    <td><p class="Estilo_4">1.Elige un tipo de soporte informático:
</td>
    <td></td>
    </tr>
    <tr>
    <td>
        <select class="Estilo_4" name="CodigoSoporte">
        <% Vector ConsultaEstatica =
ConsultasEstablecidas.Estaticas(8);
        ParametrosDeConsulta mostrar = null;
        for (int i=0;i<ConsultaEstatica.size();i++)
        { mostrar =
(ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <option value="<%= mostrar.getParametro1() %>"><%=
mostrar.getParametro2() %></option>
        <% } %>
        </select>
    </td>
    <td><input type="submit" value="Selección de soporte"
style="font-family: Arial Narrow"></td>
    </tr>
    </table>
    <p class="Estilo_4"><b>Código de soporte informático seleccionado: </b>
<%= ControlSoportesInformaticos.getCodigoSoporte() %></p>
    </form>
    <!-- Formulario2 ===== -->
    <form method="POST">
    <table>
    <tr>
    <td><p class="Estilo_4">2.Elige una forma de envío: </td>
    <td></td>
    </tr>
    <tr>
    <td>
        <select class="Estilo_4" name="CodigoEnvio">
        <% ConsultaEstatica = ConsultasEstablecidas.Estaticas(3);
        mostrar = null;
        for (int i=0;i<ConsultaEstatica.size();i++)
        { mostrar =
(ParametrosDeConsulta)ConsultaEstatica.elementAt(i);%>
        <option value="<%= mostrar.getParametro1() %>"><%=
mostrar.getParametro2() %></option>
        <% } %>
        </select>
    </td>
    <td><input type="submit" value="Selección de envío" style="font-
family: Arial Narrow"></td>
    </tr>
    </table>
    <p class="Estilo_4"><b>Código de envío seleccionado: </b> <%=
ControlSoportesInformaticos.getCodigoEnvio() %></p>
    </form>
    <!-- Formulario3 ===== -->
    <form method="POST">
    <table>
    <tr>
    <td><p class="Estilo_4">3.Selecciona el número de soportes
informáticos: </td>
    <td></td>
    </tr>
    <tr>
    <td>
        <select class="Estilo_4" name="NumeroSoportes">
        <% for (int i=1;i<1001;i++)
        { %><option value="<%= i %>"><%= i %></option><% } %>
        </select>
    </td>

```

```

        <td><input type="submit" value="Registrar Salida" style="font-
family: Arial Narrow"></td>
    </tr>
</table>
</form>
</center>
<% } %>
</font>
</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table width="100%">
    <tr>
        <td width="25%"></td>
        <td width="50%">
            <blockquote class="Estilo_8">
                Simulación de Salida de Soportes Informáticos:<br>
                La salida de información del dominio de seguridad deberá ser registrada.
            </blockquote>
        </td>
        <td width="25%"></td>
    </tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
    <tr>
        <td width="100%" class="Estilo_6">
            <b>
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Bienvenido.jsp">
                Bienvenida al cliente. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Identificacion/Identif
icacion.jsp">
                Identificación & Autentificación. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Credenciales/Obtencion
_Credenciales.jsp">
                Obtención de Credenciales. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Modulo_Decision_de_Acceso/Ele
ccion_de_Recursos.jsp">
                Decisión de Acceso. </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Recurso_Administracion_de
l_Sistema.jsp">
                Administración del Sistema </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/ARS/Simulaciones/Menu_Simulac
iones.jsp">
                Simulaciones </a> |
                | <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
                Cerrar sesión. </a> |
            </b>
        </td>
    </tr>
</table>
<%@include file="/Comunes/Pie.jsp" %>

```

9.74. Encabezado.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Encabezado.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* JSP con el encabezado de las páginas que forman el cliente.
* =====
*/
%>

<@page import="ars.ControlAcceso" errorPage = "/Comunes/Error.jsp" %>

<html>
<head>
<title>Servidor de Registro de Acceso</title>
<STYLE>
.Estilo_1 { BACKGROUND-COLOR: #FCB12D; FONT-SIZE: 15px; COLOR: #FFFFFF;
FONT-FAMILY: Arial Narrow }
.Estilo_2 { FONT-SIZE: 15px; COLOR: #0000FF; FONT-FAMILY: Batang }
.Estilo_3 { FONT-SIZE: 15px; COLOR: #FF3300; FONT-FAMILY: Batang }
.Estilo_4 { FONT-SIZE: 12px; FONT-FAMILY: Batang }
.Estilo_5 { BACKGROUND-COLOR: #FFFF99; FONT-SIZE: 12px; FONT-FAMILY: Verdana }
.Estilo_6 { FONT-SIZE: 13px; FONT-FAMILY: Franklin Gothic Medium }
.Estilo_7 { FONT-SIZE: 18px; FONT-FAMILY: Monotype Corsiva }
.Estilo_8 { FONT-SIZE: 15px; COLOR: #333333; FONT-FAMILY: Arial Narrow }
.Estilo_9 { FONT-SIZE: 10px; COLOR: #FFFFFF; FONT-FAMILY: Comic Sans MS }
</STYLE>
</head>

<body>

<table border="0" width="100%">
<tr>
<td width="10%"></td>
<td width="66%"></td>
<td width="24%"></td>
</tr>
</table>

<table border="0" width="100%"
background="http://localhost:8084/Servidor_de_Registro_de_Acceso/Imagenes/linea_encabeza
do.jpg">
<tr>
<td width="100%">&nbsp;</td>
</tr>
</table>

```

9.75. Pie.jsp.-

```

<%
/* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Pie.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* JSP que recoge el pie de las páginas que forman la aplicación.
* =====
*/
%>

<table border="1" width="100%" height="26">
  <tr>
    <td align=center width="100%" height="20">

      <font color="#666666" face="Arial Narrow" size="2">
        © Junio 2005 - Proyecto final de carrera: Servidor de Registro de Accesos para
        Sistemas de Información Sanitarios.
      </font>

    </td>
  </tr>
</table>

<p align=right>
<a href="http://www.sun.es" target="_blank"></a>
<a href="http://www.sun.es" target="_blank"></a>
<a href="http://www.netbeans.org" target="_blank"></a>
<a href="http://www.postgresql.org" target="_blank"></a>
<a href="http://jakarta.apache.org" target="_blank"></a>
<a href="http://www.avantbrowser.com" target="_blank"></a>
<a href="http://www.esi.us.es" target="_blank"></a>
<a href="http://trajano.us.es" target="_blank"></a>
</p>

</body>
</html>

```

5.76. Error.jsp.-

```

<% /* =====
* Proyecto final de carrera.
* Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
* Autor: Daniel David García Malmierca.
* Área: Ingeniería Telemática.
* Departamento: Ingeniería de Sistemas y Automática.
* Tutora: Isabel Román Martínez.
* Ingeniería de Telecomunicación - Universidad de Sevilla.
* =====
* $Id: Error.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
* =====
* Comentarios.-
* JSP que se muestra en caso que exista alguna excepción en el
* transcurso de la ejecución del cliente.
* =====
*/ %>
<%@page isErrorPage="true" import="java.util.*" %>
<%@include file="/Comunes/Encabezado.jsp"%>
<!-- Titulo página ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<br>
<p align="center" class="Estilo_2">
<b><u>Página de Error</u></b>
</p>
</td>
</tr>
</table>
<!-- Mensajes ===== -->
<table border="0" width="100%">
<tr>
<td width="100%" align="center">
<font class="Estilo_7">
<br>
<p><font size="+2"> Ha ocurrido una excepción.</font></p>
<p class="Estilo_3"> La identificación de la excepción es: <b><%=
exception.toString() %></b></p>
<b>Intente de nuevo la acción que provocó la excepción.<br>
Si el problema persiste póngase en contacto con el administrador
del sistema.
</b><br>
</font>
</td>
</tr>
</table>
<!-- Opciones ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">&nbsp;</td>
</tr>
</table>
<br>
<!-- Descripcion ===== -->
<table border="0" width="100%">
<tr>
<td width="25%">&nbsp;</td>
<td width="50%">&nbsp;</td>
<td width="25%">&nbsp;</td>
</tr>
</table>
<!-- Camino ===== -->
<table border="0" width="100%">
<tr>
<td width="100%">
<b>| <a style="FONT-SIZE: 10px; COLOR: #666666; FONT-FAMILY: Batang"
href="http://localhost:8084/Servidor_de_Registro_de_Acceso/Comunes/Cerrar_Sesion.jsp">
Cerrar sesión.</a> |
</b>
</td>
</tr>
</table>
<%@include file="/Comunes/Pie.jsp"%>

```

9.77. Cerrar_Sesion.jsp.-

```
<%
/* =====
 * Proyecto final de carrera.
 * Servidor de Registros de Acceso para Sistemas de Información Sanitarios.
 * Autor: Daniel David García Malmierca.
 * Área: Ingeniería Telemática.
 * Departamento: Ingeniería de Sistemas y Automática.
 * Tutora: Isabel Román Martínez.
 * Ingeniería de Telecomunicación - Universidad de Sevilla.
 * =====
 * $Id: Cerrar_Sesion.jsp, v 1.0, 16 de Mayo de 2005, 12:00 $
 * =====
 * Comentarios.-
 * JSP que inicializa los parámetros de control de acceso y redirecciona al
 * usuario del cliente a la página de bienvenida.
 * =====
 */
%>
<%@page import="ars.ControlAcceso" %>
<% ControlAcceso.Clear(); %>
<jsp:forward page=".." />
```