

UNIVERSIDAD DON BOSCO  
FACULTAD DE INGENIERIA  
ESCUELA DE COMPUTACION



**PROTOTIPO DE SISTEMA DE INFORMACIÓN GEOGRÁFICA  
PARA LA UBICACIÓN DE PROYECTOS DE INVERSIÓN PÚBLICA,  
CON APOYO A LA TOMA DE DECISIONES EN EL MUNICIPIO DE SAN  
SALVADOR, EN AMBIENTE WEB.**

INTEGRANTES:

JENNIFFER ELIZABETH MARTINEZ ORDOÑEZ  
ENRIQUE ALEXANDER SANCHEZ ZALDAÑA  
CARMEN ELENA SUAREZ VELASQUEZ

Ciudadela Don Bosco, Febrero del 2004

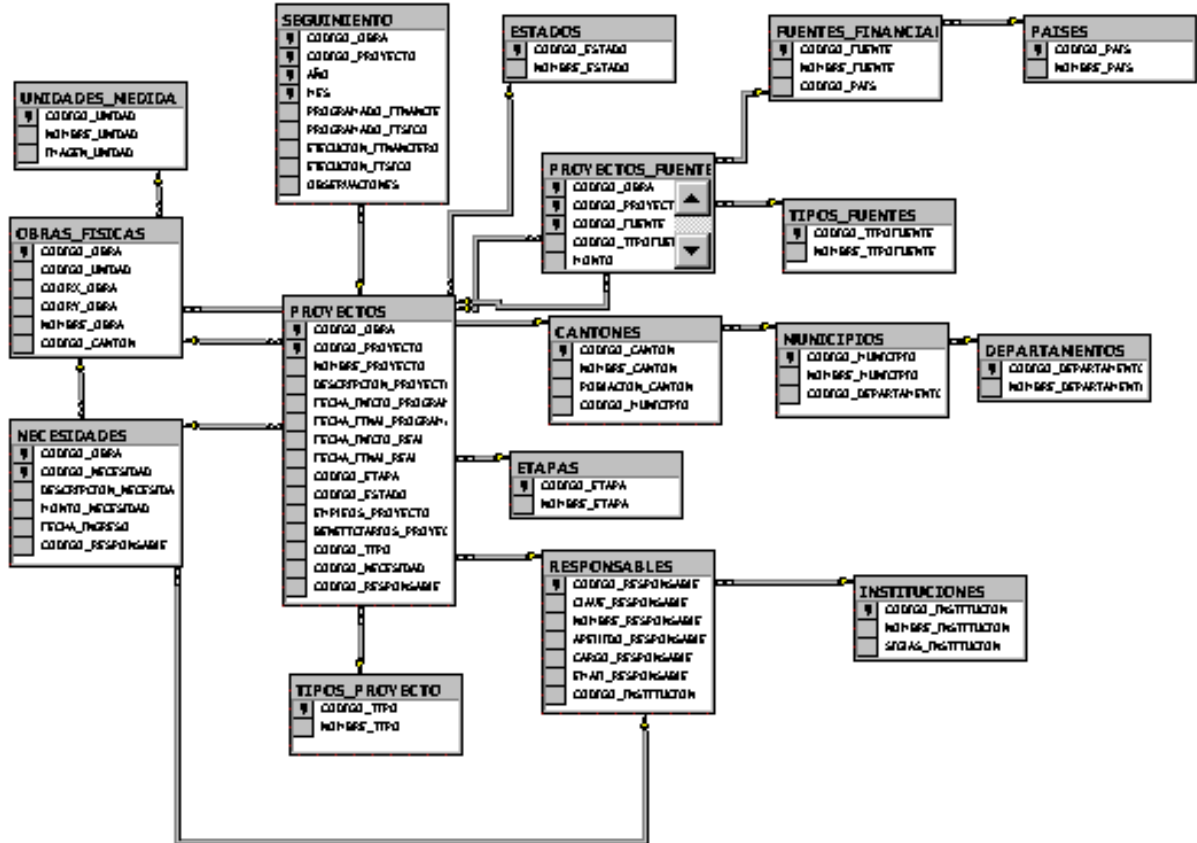
## INDICE

Modelo de datos	4
Diccionario de datos	5
Tablas para el esquema de administración	20
Servlets	22
- Estructura General.	22
- Ingreso al sistema.	26
- Página de inicio.	29
- Generación del menú.	30
- Generación de página de mapa.	43
- Generación de mapa para captura de obras.	47
- Generación de mapa para reportes.	52
- Captura de obras.	59
- Captura Unidades de Medida	61
o Captura Características unidades.	64
o Adjuntar Archivo.	70
- Registro de necesidades.	74
o Listado de necesidades.	74
o Captura de necesidades.	76
o Actualización de necesidades.	81
- Registro de proyectos.	84
o Listado de proyectos.	84
o Captura de proyectos.	86
o Actualización de proyectos.	95
- Registro de fuentes de financiamiento.	100
o Listado de fuentes.	100
o Captura de fuentes.	103
o Actualización de fuentes.	107
- Registro seguimiento de proyectos.	110
o Selección de periodo.	111
o Captura de seguimiento.	114
o Actualización de seguimiento.	119
- Información general de obras.	122
- Administración.	126
o Administración.	126
o Administración Modificación.	131
o Administración Actualización.	140
- Resultados en mapa	144
- Reportes de avance mostrados en el mapa.	154
o Reportes de avance en tablas.	154
o Reportes consolidados.	159
o Servlet para generar estado de reportes.	165
Scripts	170
- Control del mapa SVG para captura de obras.	170
- Control del mapa SVG para reportes.	179
- Interactividad y validaciones en formularios de captura	190

de datos.	
Hoja de estilos	206
SVG	213
- Organización básica de los archivos SVG.	213
Procedimientos almacenados y Triggers	219

## Modelo de datos

El modelo de datos definidos es el siguiente:



## Diccionario de datos

Tabla Cantones.

Almacena los cantones y sus atributos.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_CANTON	enteros (int)	Almacena el código de identificación de cada cantón.
NOMBRE_CANTON	alfanuméricos (varchar)	Almacena el nombre del cantón
POBLACION_CANTON	enteros (int)	Almacena el número de habitantes en cada cantón.
CODIGO_MUNICIPIO	enteros (int)	Almacena el código del municipio al cual pertenece el cantón.

Tabla Departamentos.

Almacena los departamentos y sus atributos.

Campos:

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_DEPARTAMENTO	enteros (int)	Almacena el código de identificación de cada departamento.
NOMBRE_DEPARTAMENTO	alfanuméricos (varchar)	Almacena el nombre del departamento.

Tabla Estados

Almacena los estados en los que puede encontrarse un proyecto, estos pueden ser: Atrasado, A Tiempo, Adelantado, Suspendido, No corresponde Iniciar.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_ESTADO	enteros (int)	Almacena el código que identifica a cada estado.
NOMBRE_ESTADO	alfanuméricos (varchar)	Almacena el nombre del estado.

#### Tabla Etapas

Almacena las etapas por las que puede pasar un proyecto. Por ejemplo, No Iniciado, En Ejecución Física, Terminado Físicamente, etc.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_ETAPA	enteros (int)	Almacena el código que identifica a cada etapa.
NOMBRE_ETAPA	alfanuméricos (varchar)	Almacena el nombre de cada etapa.

#### **Tabla fuentes-financiamiento**

Almacena las fuentes de financiamiento que pueden estar participando en un proyecto.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_FUENTE	alfanuméricos (varchar)	Este campo es la llave primaria de la tabla, almacena el código que identifica a cada fuente de financiamiento.
NOMBRE_FUENTE	alfanuméricos	Almacena los nombres de las

	(varchar)	fuentes de financiamiento.
--	-----------	----------------------------

#### Tabla Instituciones

Almacena las características de las instituciones que estén involucradas con el Sistema.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_INSTITUCION	enteros (int)	Este campo es la llave primaria de la tabla, almacena el código que identifica a cada institución.
NOMBRE_INSTITUCION	alfanuméricos (varchar)	Almacena el nombre de la institución.

#### Tabla Municipios.

Almacena las características de cada municipio.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_MUNICIPIO	enteros (int)	Este campo es la llave primaria de la tabla, almacena el código que identifica a cada municipio.
NOMBRE_MUNICIPIO	Alfanuméricos (varchar)	Almacena el nombre que describe a cada municipio.

#### Tabla Necesidades.

Almacena las necesidades identificadas para cada obra en particular.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_OBRA	enteros (int)	Este campo junto con el campo

		CODIGO_NECESIDAD forman la llave primaria de la tabla, almacena el código de la obra a la cual esta siendo asignada una necesidad identificada.
CODIGO_NECESIDAD	enteros (int)	Almacena el código con el que se identifica la necesidad registrada.
DESCRIPCION_NECESIDAD	Alfanuméricos (varchar)	Almacena la descripción de la necesidad para una comprensión de la situación en la obra a la cual esta siendo asignada. Tipo de datos permitidos.
MONTO_NECESIDAD	Numéricos, con dos decimales permitidos (decimal).	Almacena el monto estimado para cubrir la necesidad descrita.
FECHA_INGRESO	fecha (datetime)	Almacena la fecha de ingreso al sistema de la necesidad que se esta registrando.
CODIGO_RESPONSABLE	Alfanuméricos (varchar)	Almacena el código de identificación de cada usuario en el Sistema.

Tabla Obras\_Fisicas

Almacena las características de las obras físicas ubicadas en el mapa.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_OBRA	enteros (int)	Este campo es la llave primaria de la tabla, almacena el código de identificación de cada obra
CODIGO_UNIDAD	alfanuméricos	Este campo almacena el código

	(varchar)	del tipo de obra, por ejemplo, si es escuela, unidad de salud, etc.
COORX_OBRA	numéricos con X decimales (decimal).	Almacena el valor de la coordenada X en el mapa.
COORY_OBRA	numéricos con X decimales (decimal).	Almacena el valor de la coordenada Y en el mapa.
NOMBRE_OBRA	alfanuméricos (varchar)	Almacena el nombre de la obra.
CODIGO_CANTON	enteros (int)	Almacena el código del cantón. sobre el cual esta ubicado la obra.

#### Tabla Países.

Almacena los países a los cuales pueden pertenecer las Fuentes de financiamiento.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_PAIS	alfanuméricos (varchar).	Este campo es la llave primaria de la tabla, almacena el código que identifica a cada país.
NOMBRE_PAIS	alfanuméricos (varchar)	Almacena el nombre del país.

#### Tabla Proyectos.

Almacena los proyectos que tiene asignada cada obra que se encuentra ubicada en el mapa.

Nombre de Campo	Tipo de Dato	Descripción
-----------------	--------------	-------------

CODIGO_OBRA	enteros (int)	Este campo junto con el campo CODIGO_PROYECTO, representan la llave primaria de esta tabla. Almacena el código de la obra a la cual se le esta asignando el proyecto a registrar.
CODIGO_PROYECTO	enteros (int).	Almacena el código del proyecto que identifica al mismo.
NOMBRE_PROYECTO	alfanuméricos (varchar)	Almacena el nombre del proyecto.
DESCRIPCION_PROYECTO	alfanumérico (varchar)	Almacena la descripción del proyecto.
FECHA_INICIO_PROGRAMADA	fecha (datetime)	Almacena la fecha de inicio programada que se ha establecido para dar inicio al proyecto.
FECHA_FINAL_PROGRAMADA	fecha (datetime)	Almacena la fecha final programada que ha sido estimada para la finalización del proyecto.
FECHA_INICIO_REAL	fecha (datetime).	Almacena la fecha de inicio real del proyecto, una vez el proyecto ha sido iniciado.
FECHA_FINAL_REAL	fecha (datetime)	Almacena la fecha final real del proyecto, una vez el proyecto ha sido finalizado.
CODIGO_ETAPA	enteros (int)	Almacena el código de la etapa en la cual se encuentra el proyecto. Este campo mantiene relación con la tabla

		ETAPAS.
CODIGO_ESTADO	enteros (int)	Almacena el código del estado en el cual se encuentra el proyecto. Este campo esta relacionado con la tabla ESTADOS.
EMPLEOS_PROYECTO	enteros (int)	Almacena el número de empleos que están siendo generados por este proyecto.
BENEFICIARIOS_PROYECTO	enteros (int)	Almacena el número de personas que serán beneficiadas a través de este proyecto.
CODIGO_TIPO	enteros (int)	Almacena el código del tipo de proyecto. Este campo mantiene relación con la tabla TIPOS-PROYECTO.
CODIGO_NECESIDAD	enteros (int)	Almacena el código de la necesidad que esta siendo cubierta por este proyecto. Este campo mantiene relación con la tabla NECESIDADES.
CODIGO_RESPONSABLE	alfanuméricos (varchar)	Almacena el código del responsable de registro del proyecto en el Sistema. Este campo esta relacionado con la tabla RESPONSABLES.

Tabla Proyectos-Fuentes.

Un proyecto puede estar siendo financiado por muchas fuentes, y una fuente puede estar financiando muchos proyectos, con lo que nos encontramos con

una relación de muchos a muchos. Esta tabla normaliza la relación entre las tablas PROYECTOS y FUENTES\_FINANCIAMIENTO, almacena las fuentes de financiamiento que tienen participación en un proyecto.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_OBRA	enteros (int).	Este campo, junto con el campo CODIGO_PROYECTO y CODIGO_FUENTE representan la llave primaria de la tabla. Almacena el código de la obra a la que pertenece el proyecto.
CODIGO_PROYECTO	enteros (int).	Este campo almacena el código del proyecto al cual se le están especificando las diferentes fuentes que participan para llevar a cabo la ejecución.
CODIGO_FUENTE	alfanuméricos (varchar)	Almacena el código de la fuente que tiene participación en el proyecto especificado.
CODIGO_TIPOFUENTE	enteros (int)	Almacena el código que representa el tipo de financiamiento que se esta dando.

Tabla Responsables.

Almacena todas las características de los usuarios del Sistema.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_RESPONSABLE	alfanuméricos (varchar).	Almacena el código del responsable, este código será con el que ingrese al sistema.
CLAVE_RESPONSABLE	alfanuméricos	Almacena la contraseña con la

	(varchar)	cual será admitido el usuario en el sistema.
NOMBRE_RESPONSABLE	alfanuméricos (varchar)	Almacena el nombre del responsable.
APELLIDO_RESPONSABLE	alfanuméricos (varchar)	Almacena el o los apellidos de la persona responsable.
CARGO_RESPONSABLE	alfanumérico (varchar)	Cargo que desempeña la persona en su institución.
EMAIL_RESPONSABLE	alfanuméricos (varchar)	Almacena la dirección de correo electrónico en la cual se puede contactar a este responsable.

#### Tabla Seguimiento.

Almacena los países a los cuales pueden pertenecer las Fuentes de financiamiento.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_OBRA	enteros (int).	Este campo junto con los campos CODIGO_PROYECTO, AÑO y MES, constituyen la llave primaria de esta tabla. Almacena el código que identifica a la obra.
CODIGO_PROYECTO	enteros (int).	Almacena el código del proyecto al cual se le esta asignando registro de seguimiento. Tipo de datos permitidos: enteros (int). AÑO, almacena el año para el

		cual se esta registrando el seguimiento.
MES	enteros (int)	Almacena el mes para el cual se esta registrando el seguimiento.
PROGRAMADO_FINANCIERO	numéricos con dos decimales (decimal)	Almacena el monto programado.
EJECUCION_FINANCIERO	numéricos con dos decimales (decimal).	Almacena el monto de ejecución financiera.
PROGRAMADO_FISICO	numéricos con dos decimales (numeric)	Almacena la programación del porcentaje de avance físico.
EJECUCION_FISICO	numéricos con dos decimales (numeric).	Almacena el porcentaje de ejecución física.
OBSERVACIONES	alfanuméricos (varchar).	Almacena las observaciones referentes al seguimiento del proyecto en el período especificado.

Tabla Tipos\_fuentes.

Almacena los tipos de financiamiento que una fuente puede estar proporcionando, por ejemplo préstamo, donación, fondos propios, etc.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_TIPOFUENTE	enteros (int)	Almacena el código que identifica a cada tipo de financiamiento. Este campo representa la llave primaria de esta tabla.

NOMBRE_TIPOFUENTE	alfanuméricos (varchar)	Almacena el nombre que describe el tipo de financiamiento.
-------------------	----------------------------	--

Tabla Tipos\_Proyecto.

Almacena los tipos de proyecto que pueden estar siendo ejecutados, por ejemplo, si son ampliación, reconstrucción, construcción de nueva infraestructura, etc.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_TIPO	enteros (int).	Almacena el código que identifica el tipo de proyecto. Este campo representa la llave primaria de la tabla.
NOMBRE_TIPO	alfanuméricos (varchar)	Almacena el nombre que describe cada tipo de proyecto.

Tabla Unidades\_Medida.

Almacena los tipos de obras que pueden ser ubicados en el mapa, por ejemplo: escuelas, unidades de salud, vivienda, hospitales, etc.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_UNIDAD	alfanuméricos (varchar).	Almacena el código que identifica a cada unidad de medida. Este campo representa la llave primaria de esta tabla.
NOMBRE_UNIDAD	alfanuméricos (varchar)	Almacena el nombre que describe cada unidad de medida.

La organización de datos en tabla para este esquema es la siguiente:

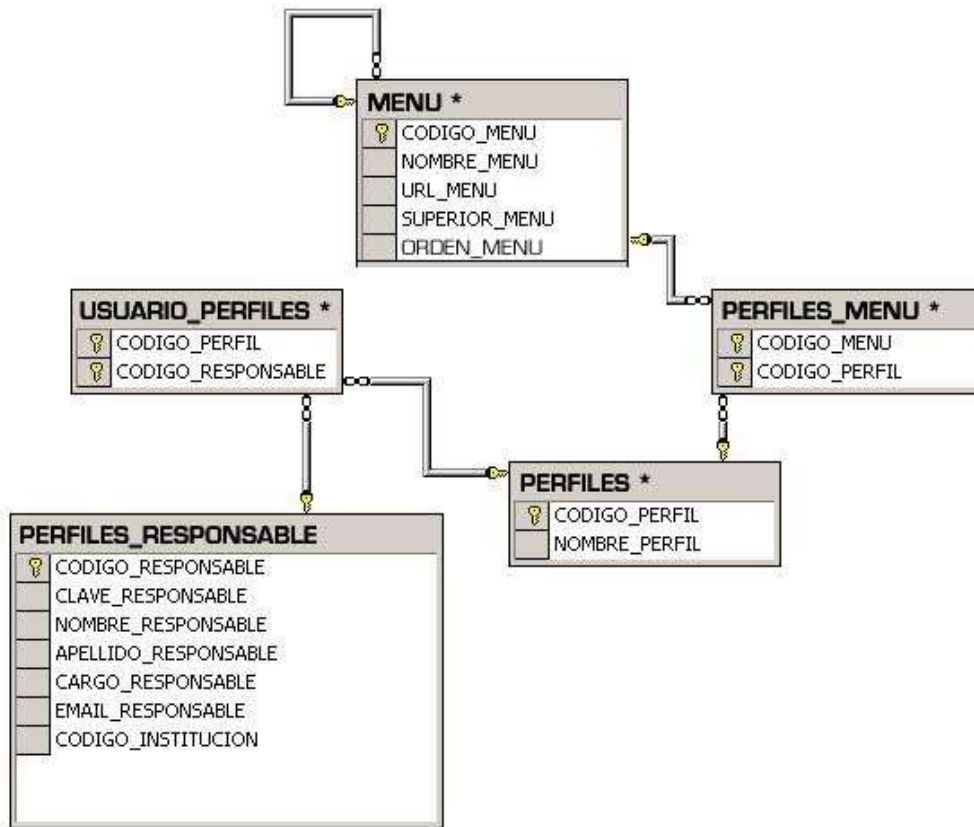


Tabla Menu.

De esta tabla se obtienen las características de las opciones del menú.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_MENU	alfanuméricos (char).	Almacena el código que identifica a cada menú. Este campo

		representa la llave primaria de esta tabla.
NOMBRE_MENU	alfanuméricos (varchar)	Almacena el nombre que describe cada elemento del menú.
URL_MENU	Alfanuméricos (varchar)	Almacena las especificaciones del archivo a ejecutar en caso de activar el menú.
SUPERIOR_MENU	Alfanumérico (char)	Almacena el código del menú padre del cual depende.
ORDEN_MENU	Entero (int)	Almacena el número que rige el orden de la opción de menú en el listado.

Tabla PERFILES.

Almacena la descripción de cada uno de los perfiles definidos para los usuarios del sistema.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_PERFIL	alfanuméricos (varchar).	Almacena el código que identifica a cada perfil. Este campo representa la llave primaria de esta tabla.
NOMBRE_PERFIL	alfanuméricos (varchar)	Almacena el nombre que describe a cada perfil.

Tabla PERFILES\_MENU.

Para poder asignar los menús que corresponden a cada perfil, es necesario tener una tabla intermedia entre estas dos, ya que, un perfil puede tener asignados muchos menús, y un mismo menú puede estar asignado a diferentes perfiles.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_MENU	alfanuméricos (char).	Almacena el código que identifica a cada menú. Este campo junto con el campo CODIGO_PERFIL representa la llave primaria de esta tabla. Este campo esta relacionado con la tabla MENU.
CODIGO_PERFIL	alfanuméricos (char)	Almacena el código del perfil. Este campo esta relacionado con el campo CODIGO_PERFIL de la tabla PERFILES.

#### Tabla PERFILES\_RESPONSABLE.

Un perfil puede estar asignado a diferentes usuarios, y un usuario puede tener asignados más de un perfil, por tanto, se vuelve necesaria una tabla intermedia que normalice la relación muchos a muchos entre ambas tablas.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_PERFIL	alfanuméricos (char).	Almacena el código que identifica a cada perfil. Este campo junto con el campo CODIGO_RESPONSABLE representa la llave primaria de esta tabla. Este campo esta relacionado con la tabla PERFILES.
CODIGO_RESPONSABLE	alfanuméricos (char)	Almacena el código del responsable. Este campo esta relacionado con el campo

		CODIGO_RESPONSABLE de la tabla RESPONSABLES.
--	--	--

### Tablas para el esquema de administración

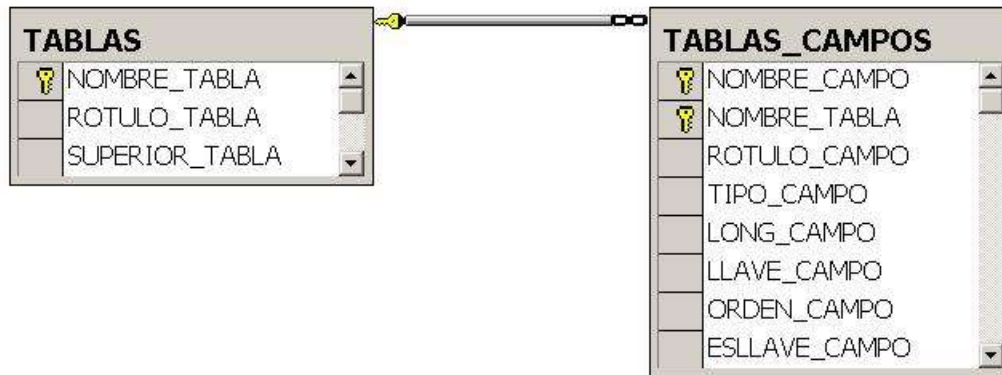


Tabla TABLAS

Almacena el nombre de la tabla en la base de datos, así como el nombre descriptivo, y su relación con tablas padres.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_TABLA	alfanuméricos (varchar).	Este campo es la llave primaria de la tabla, almacena el código que identifica a cada tabla, que es el nombre propio de la tabla en la base de datos.
NOMBRE_TABLA	alfanuméricos (varchar)	Almacena el nombre descriptivo de la tabla.
SUPERIOR_TABLA	Alfanuméricos	Almacena el nombre de la tabla padre, si así fuera el caso.

Tabla TABLAS\_CAMPOS

Almacena la descripción y características de todos los campos pertenecientes a una tabla en particular.

Nombre de Campo	Tipo de Dato	Descripción
CODIGO_CAMPO	alfanuméricos (varchar).	Este campo, junto con el campo CODIGO_TABLA forman la llave primaria de la tabla. Almacena el código que identifica al campo, es decir, el nombre del campo en la tabla.
CODIGO_TABLA	alfanuméricos (varchar)	Hace referencia a la tabla TABLAS, y almacena el nombre propio de la tabla a la que pertenece el campo.
ROTULO_CAMPO	Alfanuméricos	Almacena el nombre descriptivo (que se mostrará en pantalla) del campo.
TIPO_CAMPO	Alfanumérico	Define que tipo de datos acepta el campo (numérico, fecha, etc.)
LONG_CAMPO	Numérico	Define el tamaño del campo, es decir, el número de caracteres permitidos.
LLAVE_CAMPO	Alfanumérico	Si los datos permitidos en el campo de la base de datos obedecen a un listado previamente definido, es en este campo donde se especifica el nombre de la tabla que guarda dichos valores.
ORDEN_CAMPO	Numérico	El orden del campo en la tabla a

		la que pertenece.
ESLLAVE_CAMPO	Alfanumérico	Indica si el campo forma parte de la llave primaria de la tabla o no.

## Servlets

### Estructura General

Los servlets utilizados en esta aplicación poseen una estructura casi general, descrita a continuación.

Sección de librerías: es la primera parte que componen los servlets y es donde se importan los paquetes o librerías que contendrán las funciones a utilizarse durante la codificación, como lo son funciones de cadenas, números, formatos de salida, etc.

```
import Usuario;
import Menu;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;
```

Un punto muy importante a tomar en cuenta es que el nombre del archivo con extensión .java debe ser igual al nombre de la clase que se esta generando por que de esa forma lo identifica el compilador que genera las clases a ser utilizadas.

En la siguiente porción del servlet se declara el nombre de la clase y del tipo de respuesta que proporcionará, se le adiciona el método (DoGet) por el cual será invocado a recibir los parámetros y las variables a utilizar para solicitar los parámetros (Requerimiento) y par a mostrar los resultados (Respuesta).

Luego se especifica el tipo de contenido a ser escrito para este caso que genera las páginas html a ser visualizadas por el usuario, el contenido es de tipo text/html y en otro svt/text (para generar el archivo del mapa).

Se declara la variable a utilizar para la impresión del contenido durante toda la clase: `PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());`

Y a partir de este punto comienza la generación de la página html a mostrar, se inicia con el encabezado de la página, se establece la hoja que le dará estilo a los objetos dentro del html y el archivo .js para la manipulación de funciones JavaScript.

```
Salida.println("<html>");
Salida.println("<head>");
Salida.println("<title>Fuentes de Financiamiento</title>");
Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
Salida.println("<SCRIPT language=JavaScript src=\"/HTML/include/ourScript.js\"
type=text/javascript></SCRIPT>");
Salida.println("</head>");
```

```
Salida.println("<body>");
```

Este punto se deja abierto donde se llama una función que realizará el proceso de generación de contenido dinámico, las conexiones a la base de datos, solicitud de parámetros recibidos, etc.

```
String variable1=Requerimiento.getParameter("parametro1");  
String variable2=Requerimiento.getParameter("parametro2");  
String variable3=Requerimiento.getParameter("parametro3");
```

```
.  
. .  
. .
```

```
// Ejecutando Proceso Principal  
Proceso(variable1, variable2, variable3, ...);
```

La solicitud de los parámetros puede hacerse dentro de la función principal a ejecutarse, solo debe pasarse por parámetro el objeto que los contiene.

Posteriormente se cierra la estructura del doc html y se libera la memoria de la clase.

```
// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida  
Salida.println("</body>");  
Salida.println("</html>");  
Salida.flush();  
Salida.close();  
}
```

De manera completa se presenta la estructura del método de solicitud.

```
public void doGet(HttpServletRequest Requerimiento, HttpServletResponse Respuesta)  
throws ServletException, IOException  
{  
// Abriendo Archivo de Salida  
Respuesta.setContentType("text/html");  
PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());  
  
// Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro  
Salida.println("<html>");  
Salida.println("<head>");  
Salida.println("<title>Fuentes de Financiamiento</title>");  
Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");  
Salida.println("<SCRIPT language=JavaScript src=\"/HTML/include/ourScript.js\"  
type=text/javascript></SCRIPT>");  
Salida.println("</head>");  
Salida.println("<body>");
```

```
String variable1=Requerimiento.getParameter("parametro1");  
String variable2=Requerimiento.getParameter("parametro2");  
String variable3=Requerimiento.getParameter("parametro3");
```

```

// Ejecutando Proceso Principal
Proceso(PrintWriter Salida, variable1, variable2, variable3, ...);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

```

Se incluyen los archivos de hoja de estilos y JavaScript para automáticamente hacer uso de las funciones que ambos poseen.

**Función Proceso Principal:** Esta sección es la que lleva a cabo toda la ejecución de lo que se desea realice la clase que se está creando. Si dentro de esta clase se hará uso de variables que establecen conexión a la base de datos llevan la siguiente estructura:

```

private void Proceso(PrintWriter , tipo variable1, tipo variable2, tipo variable3, ...);
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();

```

Para realizar la conexión se utiliza la clase **Usuario** donde se definen el usuario, password, servidor y base de datos a la que el usuario tiene acceso.

Esto nos da la ventaja de que al querer modificar el usuario, password, nombre de la base de datos, servidor, etc. No debe modificarse uno por uno los servlet sino únicamente se modifica la clase **Usuario**.

Para la generación del menú en cada pantalla, se hace la llamada al servlet que lo genera. Las tres sentencias que se describen a continuación son las encargadas de generar el menú de acuerdo al usuario activo.

```
Menu DoMenu = new Menu();
DoMenu.CREARMENU(Usuario,Salida,Driver,Usr,DB,Pass);
DoMenu.Encabezado(Salida);
```

Ejecución de consultas, actualizaciones, inserciones, borrados de datos, etc, son realizados en esta parte, salida de datos haciendo uso de la variable Salida:

```
Salida.println("Texto o contenido html a mostrar")
Salida.println("Texto o contenido html a mostrar")
Salida.println("Texto o contenido html a mostrar")
Salida.println("Texto o contenido html a mostrar")
.
.
.
```

Al finalizar las instrucciones se procede al cierre de la función Proceso Principal, liberar memoria, destruir objetos de conexión y establecer procesos de capturas de errores, ya sean estos de codificación o de retorno de datos de las instrucciones SQL.

```
}

}
catch (Exception Error)
{
    Salida.println("Situación Anormal!");
    Error.printStackTrace(Salida);
}
finally
{
    try
    {
        if (ResultadoSQL!=null) ResultadoSQL.close();
        if (InstruccionSQL!=null) InstruccionSQL.close();
        if (ConexionSQL!=null) ConexionSQL.close();
    }
    catch (Exception Error) {}
}
}

}
```

Debe tenerse en cuenta la ayuda de libros que apoyen el uso de la sintaxis de Java, dado que no es el objetivo enseñar el lenguaje de programación en este manual ni la estructura y funcionamiento del mismo.

## Ingreso al Sistema

En este servlet se realiza proceso de verificación de usuarios registrados.

```
import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class Entrada extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Sistema de Información de Proyectos</title>");
        Salida.println("</head>");

        /* CODIGO_RESPONSABLE: es el usuario
        CLAVE_RESPONSABLE: es la contraseña
        Ambos han sido ingresado en la pantalla de entrada.
        */

        String RESPONSABLE=Requerimiento.getParameter("CODIGO_RESPONSABLE");
        String CONTRASEÑA=Requerimiento.getParameter("CLAVE_RESPONSABLE");

        // Ejecutando Proceso Principal
        Proceso(Salida,RESPONSABLE,CONTRASEÑA);

        // Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
        Salida.println("</html>");
        Salida.flush();
        Salida.close();
    }

    private void Proceso(PrintWriter Salida,String RESPONSABLE, String CONTRASEÑA)
    {
```

```

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;

    Class.forName(Driver);
    String myDB = DB;
    ConexionSQL = DriverManager.getConnection(myDB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();

//Verificando el usuario ingresado, que exista en la base de datos

    String VERIFICAR = "select count(*) from RESPONSABLES "+
        "where CODIGO_RESPONSABLE='"+RESPONSABLE+"'";

    ResultadoSQL=InstruccionSQL.executeQuery(VERIFICAR);
    ResultadoSQL.next();
    int CONTADOR=ResultadoSQL.getInt(1);

    if (CONTADOR == 1)
    {

        String CLAVE= "select
CLAVE_RESPONSABLE,NOMBRE_RESPONSABLE,APELLIDO_RESPONSABLE from
RESPONSABLES "+
        "where CODIGO_RESPONSABLE = '"+RESPONSABLE+"'";

        ResultadoSQL=InstruccionSQL.executeQuery(CLAVE);
        ResultadoSQL.next();
        String CONT=ResultadoSQL.getString(1).trim();
        String NOMBRES=ResultadoSQL.getString(2).trim()+" "+ ResultadoSQL.getString(3).trim();

        if (CONT.equals(CONTRASEÑA))
        {

/*      Creando interface de entrada si el usuario es correcto      */

            Salida.println("<frameset rows=*,1 leftmargin=0 framespacing=0>");
            Salida.println("<frame name=Principal frameborder=0 border=0 scrolling=auto>");
            Salida.println(" src=/servlet/Presentacion?USUARIO='"+RESPONSABLE+"'>");
            Salida.print ("<frame name=Nada scrolling=no frameborder=1 border=0
target=Principal");

```

```

        Salida.println(" src=">");
        Salida.println("</frameset>");
        Salida.println("<noframes>");
        Salida.println("<body bgcolor=#808000>");
        Salida.println(" <p>Esta página utiliza frames, pero su navegador no soporta esta
característica.</p>");
        Salida.println("</font>");
        Salida.println("</body>");
        Salida.println("</noframes>");
    }
    else
    {

/*      Si el usuario no ingreso de forma correcta su usuario o clave de acceso      */

        Salida.println("<br><br>");
        Salida.println("<center>");
        Salida.println("<font size=4 face=verdana> CLAVE NO ES CORRECTA </font>");
        Salida.println("<br><br>Usted debe utilizar el nombre y clave de usuario entregado
por el adminsitrador del sistema (en letras MAYÚSCULAS)<br><br>");
        Salida.println("<font size=2 face=verdana> <a href='/HTML/Ingreso.html'> Regresar
a Pantalla de Acceso </a></font></center>");
    }
    }
    else
    {

        Salida.println("<br><br>");
        Salida.println("<center>");
        Salida.println("<font size=4 face=verdana> USUARIO NO EXISTE </font>");
        Salida.println("<br><br>Usted debe utilizar el nombre y clave de usuario entregado por
el administrador del sistema (en letras MAYÚSCULAS)<br><br>");
        Salida.println("<font size=2 face=verdana> <a href='/HTML/Ingreso.html'> Regresar a
Pantalla de Acceso </a></font></center>");
    }
}
catch (Exception Error)
{
    Salida.println("Situación Anormal!");
    Error.printStackTrace(Salida);
}
finally
{
    try
    {
/*      Cerrando variables de conexión      */
        if (ResultadoSQL!=null) ResultadoSQL.close();
        if (InstruccionSQL!=null) InstruccionSQL.close();
        if (ConexionSQL!=null) ConexionSQL.close();
    }
    catch (Exception Error) {}
}
}

/*      Función para capturar los datos por medio del DoPost      */
public void doPost(HttpServletRequest SOLICITUD,
        HttpServletResponse RESPUESTA)
        throws ServletException, java.io.IOException

```

```

    {
        doGet(SOLICITUD,RESPUESTA);
    }
}

```

## **Página de inicio.**

Este servlet es el que recibe al usuario luego de haber sido autenticado. Se genera el menú personalizado con las opciones a las cuales el usuario tiene acceso.

```

import Usuario;
import Menu;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class Presentacion extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Sistema de Información de Proyectos</title>");
        Salida.println("<LINK href='\"/HTML/include/ourstyle.css\"' type=text/css rel=stylesheet>");
        Salida.println("<SCRIPT language=JavaScript src='\"/HTML/include/ourScript.js'
type=text/javascript></SCRIPT>");
        Salida.println("</head>");
        Salida.println("<body background='\"/HTML/images/presentacion.jpg\"' style='\"background-position:
center;background-repeat:no-repeat;background-color:black;\" topmargin=0 leftmargin=0>");

        String RESPONSABLE=Requerimiento.getParameter("USUARIO");

        // Ejecutando Proceso Principal
        Proceso(Salida,RESPONSABLE);

        // Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
        Salida.println("</body>");
        Salida.println("</html>");
        Salida.flush();
    }
}

```

```

        Salida.close();
    }

private void Proceso(PrintWriter Salida,String RESPONSABLE)
{

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    Menu DoMenu = new Menu();

    DoMenu.CREARMENU(RESPONSABLE,Salida,Driver,Usr,DB,Pass);

    Salida.println("<p>&nbsp;</p><br><br><br><table border=0 width=100%>");
    Salida.println("<tr><td background='/HTML/images/franja_verde.jpg' height='18px'><img
src='/HTML/images/franja_verde.jpg'></td></tr>");
    Salida.println("</table>");

}

public void doPost(HttpServletRequest SOLICITUD,
    HttpServletResponse RESPUESTA)
    throws ServletException, java.io.IOException
{
    doGet(SOLICITUD,RESPUESTA);
}

}

```

## Generación del menú.

Se ha creado una clase para ser llamada en cada servlet, la cual verifica el usuario activo y los permisos que este tiene en el sistema, en base a lo cual genera las opciones de menú.

```

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

public class Menu
{

    public void Encabezado(java.io.PrintWriter SALIDA)
    {
        SALIDA.println("<table width=100% align=center>");
    }
}

```

```

        SALIDA.println("        <tr>");
//        SALIDA.println("        <td align=center><font style=\"font-
size:20px;\">Sistema&nbsp;de&nbsp;Información&nbsp;de&nbsp;Proyectos</font></td>");
        SALIDA.println("        <td align=center><img
src=/HTML/banner.jpg></td>");
        SALIDA.println("        </tr>");
        SALIDA.println("</table>");
        SALIDA.println("<p align=right class=LaFecha>");
        SALIDA.println("<script>");
        SALIDA.println(" document.write(customDateString(new Date()))");
        SALIDA.println("</script>");
        SALIDA.println("</p>");

    }

    public void CREAMENU(String RESPONSABLE,java.io.PrintWriter SALIDA,String Driver, String
    Usr, String DB, String Pass)
    {

        int LONGICODMEN = 0;
        int NDEPEND = 0;
        String RUTA = "";
        String SUPERIOR = "";
        String URLVAL = "";
        String NOMBRE = "";
        String CODIGO = "";
        String URLAYU = "";

        int nav=1;
        int espace=0;
        int pos_x=0;
        int pos_y=80;
        int m=0;
        int d=0;
        String cod_menu="";
        String nom_menu="";
        String url="";
        String SubMenuBase="";
        String MenuBase="";
        int x=0;
        int y=0;
        int vacio=0;

        Connection ConexionSQL=null;
        Statement InstruccionSQL=null;
        ResultSet ResultadoSQL=null;

        Connection ConexionSQL2=null;
        Statement InstruccionSQL2=null;
        ResultSet ResultadoSQL2=null;

    try{

```

```

Class.forName(Driver);
ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
ConexionSQL2 = DriverManager.getConnection(DB,Usr,Pass);

InstruccionSQL=ConexionSQL.createStatement();
InstruccionSQL2=ConexionSQL2.createStatement();

if ( RESPONSABLE!="")
{

    MenuBase="Select Count(*) FROM " +
        " Perfiles_Responsables as SUP, Perfiles_Menu as SPM, Menu as SM, Perfiles
as SP " +
        " WHERE SUP.codigo_responsable='"+RESPONSABLE+"' and " +
        " SUP.codigo_perfil=SP.codigo_perfiles and
SP.codigo_perfil=SPM.codigo_perfil and SM.codigo_menu=SPM.codigo_menu " +
        " and SM.codigo_menu=SM.superior_menu";

    ResultadoSQL=InstruccionSQL.executeQuery(MenuBase);
    ResultadoSQL.next();

    vacio=ResultadoSQL.getInt(1);
    if (vacio!=0)
    {
        if (nav==2 || nav==3)
        {
            x=5;
            y=90;
        }
    }
    /* Creando menus escondidos, son menú hijos a ser visualizados mediante la inetracción del
usuario */
        Menu_Hijos("0",RESPONSABLE,x,0,nav,SALIDA,Driver,Usr,DB,Pass);
    }

    /* Inicio de la generación del menú simple visible para el usuario */
    MenuBase="Select Count(*) FROM " +
        " Perfiles_Responsables as SUP, Perfiles_Menu as SPM, Menu as SM, Perfiles
as SP " +
        " WHERE SUP.codigo_responsable='"+RESPONSABLE+"' and " +
        " SUP.codigo_perfil=SP.codigo_perfil and SP.codigo_perfil=SPM.codigo_perfil
and SM.codigo_menu=SPM.codigo_menu " +
        " and SM.codigo_menu=SM.superior_menu";

    ResultadoSQL=InstruccionSQL.executeQuery(MenuBase);
    ResultadoSQL.next();

    vacio=ResultadoSQL.getInt(1);
    if (vacio!=0)
    {
        int largo=0;

        MenuBase="Select SM.* FROM " +
            " Perfiles_Responsables as SUP, Perfiles_Menu as SPM, Menu as SM,
Perfiles as SP " +
            " WHERE SUP.codigo_responsable='"+RESPONSABLE+"' and " +

```

```

        " SUP.codigo_perfil=SP.codigo_perfil and
SP.codigo_perfil=SPM.codigo_perfil and SM.codigo_menu=SPM.codigo_menu " +
        " and SM.codigo_menu=SM.superior_menu";

ResultadoSQL=InstruccionSQL.executeQuery(MenuBase);
while (ResultadoSQL.next())
{
    nom_menu=ResultadoSQL.getString(2).trim();
    largo=nom_menu.length();

    if (largo>espace)
    {
        espace=largo;
    }
}
espace=espace*6;

if (nav== 2 || nav==3)
{
    pos_x=5;
}
else
{
    pos_x=0;
}

ResultadoSQL=InstruccionSQL.executeQuery(MenuBase);
while (ResultadoSQL.next())
{
    m=1;

    cod_menu=ResultadoSQL.getString(1).trim();
    nom_menu=ResultadoSQL.getString(2).trim();

    url=ResultadoSQL.getString(3).trim()+"&USUARIO="+RESPONSABLE;

    SubMenuBase="Select Count(*) FROM " +
        " Perfiles_Responsables as SUP, Perfiles_Menu as SPM, Menu
as SM, Perfiles as SP " +
        " WHERE SUP.codigo_responsable="+RESPONSABLE+"
and " +
        " SUP.codigo_perfil=SP.codigo_perfil and
SP.codigo_perfil=SPM.codigo_perfil and SM.codigo_menu=SPM.codigo_menu " +
        " and SM.superior_menu="+cod_menu+" and
SM.codigo_menu<>SM.superior_menu";

    ResultadoSQL2=InstruccionSQL2.executeQuery(SubMenuBase);
    ResultadoSQL2.next();

    /*      Creando objetos Htm que conforman los menus padres      */
    vacio=ResultadoSQL2.getInt(1);
    if (vacio!=0)
    {
        SALIDA.println("<div id='men_'+cod_menu+"
class=menuBoxPP
Style="\position:absolute;left:"+pos_x+";top:"+pos_y+";width:"+((nom_menu.length()*10)+"px;height:16

```

```

px;" onMouseOver="\show('menu"+cod_menu+"','men_"+cod_menu+"',0)"
onMouseOut="\hide('menu"+cod_menu+"','0')\"><A class=menu href="+url+"
target='Principal'>"+nom_menu+"</A></div>");
    }
    else
    {
        SALIDA.println("<div id='men_"+cod_menu+"
class=menuBoxPP
Style="\position:absolute;left:"+pos_x+";top:"+pos_y+";width:"+((nom_menu.length()*10)+"px;height:16
px;"><A class=menu href="+url+" target='Principal'>"+nom_menu+"</A></div>");
    }
    m=(nom_menu.length()*6;
    pos_x=pos_x+m+24;
}
}

}

} //Fin del try

catch (Exception ex) {
    // Envia el error de regreso al cliente
    SALIDA.println("Contactar al adminisitrador del sistema !");
    ex.printStackTrace(SALIDA);
}
finally
{
    try{
        // Siempre cierra apropiadamente
        if (ResultadoSQL != null) {
            ResultadoSQL.close();
        }
        if (ResultadoSQL2 != null) {
            ResultadoSQL2.close();
        }
        if (InstruccionSQL != null) {
            InstruccionSQL.close();
        }
        if (InstruccionSQL2 != null) {
            InstruccionSQL2.close();
        }
        if (ConexionSQL != null) {
            ConexionSQL.close();
        }
        if (ConexionSQL2 != null) {
            ConexionSQL2.close();
        }
    }
}
catch (Exception ex)
{
    // Envia el error de regreso al cliente
    SALIDA.println("Contactar al administrador del sistema !!!");
    ex.printStackTrace(SALIDA);
}
}
}

```

```

}

/*****Fin Funcion CREARMENU*****/

void Menu_Hijos(String id, String RESPONSABLE, int x, int niv, int nav, java.io.PrintWriter SALIDA,
String Driver, String Usr, String DB, String Pass)
{

/* Función que genera los elementos de las cajitas que contienen los ítems de los submenus a los
cuales se hace referencia desde las funciones javascript si en dado caso tienen submenus */
//nav variable que identifica que tipo de navegador es
//nav=1 MSIE
//nav=2 NetScape
//nav=0 Cualquier otro explorador

int nivel=0;
int new_x=0;
int prof=0;
int f=0;
int y=0;
int m=0;
int vacio=0;
String SubMenuBaseInt="";
String MenuBaseInt="";
String st="";
String cod_menu="";

nivel=niv+1;
new_x=x;
prof=0;

if (id=="0")
{
    f=0;
}
else
{
    f=1;
}

Connection ConexionSQL3=null;
Statement InstruccionSQL3=null;
ResultSet ResultadoSQL3=null;

Connection ConexionSQL4=null;
Statement InstruccionSQL4=null;
ResultSet ResultadoSQL4=null;

Connection ConexionSQL5=null;
Statement InstruccionSQL5=null;
ResultSet ResultadoSQL5=null;

try{

```

```

Class.forName(Driver);
ConexionSQL3 = DriverManager.getConnection(DB,Usr,Pass);
ConexionSQL4 = DriverManager.getConnection(DB,Usr,Pass);
ConexionSQL5 = DriverManager.getConnection(DB,Usr,Pass);

InstruccionSQL3=ConexionSQL3.createStatement();
InstruccionSQL4=ConexionSQL4.createStatement();
InstruccionSQL5=ConexionSQL5.createStatement();

SubMenuBaseInt="Select Count(*) FROM " +
                " Perfiles_Responsable as SUP, Perfiles_Menu as SPM, Menu as SM, SISPER
as SP " +
                " WHERE SUP.codigo_responsable='"+RESPONSABLE+"' and " +
                " SUP.codigo_perfiles=SP.codigo_perfiles and
SP.codigo_perfiles=SPM.codigo_perfiles and SM.codigo_menu=SPM.codigo_menu ";

    if (f==0)
    {
        SubMenuBaseInt=SubMenuBaseInt+" and
SM.codigo_menu=SM.superior_menu";
    }
    else
    {
        SubMenuBaseInt=SubMenuBaseInt+" and SM.superior_menu="+id+" and
SM.codigo_menu<>SM.superior_menu";
    }

ResultadoSQL5=InstruccionSQL5.executeQuery(SubMenuBaseInt);
ResultadoSQL5.next();
vacio=ResultadoSQL5.getInt(1);

    if (vacio!=0)
    {
        SubMenuBaseInt="Select SM.* FROM " +
                " Perfiles_Responsable as SUP, Perfiles_Menu as SPM, Menu as SM, SISPER
as SP " +
                " WHERE SUP.codigo_responsable='"+RESPONSABLE+"' and " +
                " SUP.codigo_perfil=SP.codigo_perfil and SP.codigo_perfil=SPM.coigo_perfil
and SM.codigo_menu=SPM.codigo_menu ";

        if (f==0)
        {
            SubMenuBaseInt=SubMenuBaseInt+" and
SM.codigo_menu=SM.superior_menu";
        }
        else
        {
            SubMenuBaseInt=SubMenuBaseInt+" and SM.superior_menu="+id+" and
SM.codigo_men<>SM.superior_menu";
        }

ResultadoSQL3=InstruccionSQL3.executeQuery(SubMenuBaseInt);
while (ResultadoSQL3.next())

```

```

    {

        if (f==0)
        {
            id=ResultadoSQL3.getString(1).trim();
            cod_menu=id;
        }
        else
        {
            cod_menu=ResultadoSQL3.getString(1).trim();
        }

        prof+=1;

        SubMenuBaseInt="Select Count(*) FROM " +
            " Perfiles_Responsable as SUP, Perfiles_Menu as SPM, Menu as SM,
Perfiles as SP " +
            " WHERE SUP.codigo_responsable='"+RESPONSABLE+"' and " +
            " SUP.codigo_perfiles=SP.codigo_perfiles and
SP.codigo_perfiles=SPM.codigo_perfiles and SM.codigo_menu=SPM.codigo_menu " +
            " and SM.superior_menu='"+cod_menu+"' and
SM.codigo_menu<>SM.superior_menu";

        ResultadoSQL4=InstruccionSQL4.executeQuery(SubMenuBaseInt);
        ResultadoSQL4.next();

        vacio=ResultadoSQL4.getInt(1);
        if (vacio!=0)
        {

            if (nivel>1)
            {
                if (nav==2 || nav==3)
                {

                    st="style=\"visibility:hidden;position:absolute;left:"+(new_x-18)+";top:"+(98+(prof*13))+\"";
                }
                else
                {
                    st="";
                }

                SALIDA.println("<div id=menu"+cod_menu+"
class=menuBox "+st+"
onMouseOver='show('menu"+id+"','men_"+id+"','"+nivel+"');show2('menu"+cod_menu+"','menu"+id+"','"+
nivel+"','"+prof+"' onMouseOut=hide('menu"+cod_menu+"','menu"+id+"')>");

                /*      De forma recursiva se llama la función Menu, creando cada uno de los items de este
submenu*/
                Menu(cod_menu,RESPONSABLE,0,nivel,SALIDA,Driver,Usr,DB,Pass);
                SALIDA.println("</div>");

            }
            else

```

```

        {
            if (nav==2 || nav==3)
            {
                st="style=\"visibility:hidden;position:absolute;left:"+new_x+";top:98;\"";
            }
            else
            {
                st="";
            }
            SALIDA.println("<div id=menu"+cod_menu+"
class=menuBox "+st+" onMouseOver=show('menu"+cod_menu+"','men_"+id+"','+nivel+')
onMouseOut=hide('menu"+cod_menu+"','0')>");

/*      De forma recursiva se llama la función Menu, creando cada uno de los items de este
submenu*/
                                Menu
(cod_menu,RESPONSABLE,x,nivel,SALIDA,Driver,Usr,DB,Pass);
                                SALIDA.println("</div>");

                                m=((ResultadoSQL3.getString(2).trim()).length()*6;
                                new_x=new_x+m+24;
        }

        id=cod_menu;

/*      De forma recursiva se llama la función Menu_Hijos nuevamente, Creando una nueva cajita
de referencia para los items del nuevo submenu*/

        Menu_Hijos(id,RESPONSABLE,new_x,nivel,nav,SALIDA,Driver,Usr,DB,Pass);
        }
    }

} //Fin del try

catch (Exception ex) {
    // Envía el error de regreso al cliente
    SALIDA.println("Contactar al administrador del sistema !");
    ex.printStackTrace(SALIDA);
}
finally
{
    try{
        // Siempre cierra apropiadamente
        if (ResultadoSQL3 != null) {
            ResultadoSQL3.close();
        }
        if (ResultadoSQL4 != null) {
            ResultadoSQL4.close();
        }
        if (ResultadoSQL5 != null) {
            ResultadoSQL5.close();
        }
    }
}

```

```

        if (InstruccionSQL3 != null) {
            InstruccionSQL3.close();
        }
        if (InstruccionSQL4 != null) {
            InstruccionSQL4.close();
        }
        if (InstruccionSQL5 != null) {
            InstruccionSQL5.close();
        }

        if (ConexionSQL3 != null) {
            ConexionSQL3.close();
        }
        if (ConexionSQL4 != null) {
            ConexionSQL4.close();
        }
        if (ConexionSQL5 != null) {
            ConexionSQL5.close();
        }

    }
    catch (Exception ex)
    {
        // Envia el error de regreso al cliente
        SALIDA.println("Contactar al administrador del sistema !!!");
        ex.printStackTrace(SALIDA);
    }
}

}

/*****Fin Funcion Menu_hijos*****/

void Menu(String id, String RESPONSABLE, int x, int niv, java.io.PrintWriter SALIDA,String Driver,
String Usr, String DB, String Pass)
{

/* Función que llena los objetos de menus para ser mostrados, estos contienen los ítems de los menus
Esta funcionalidad es recursiva*/

    //nav variable que identifica que tipo de navegador es
    //nav=1 MSIE
    //nav=2 NetScape
    //nav=0 Cualquier otro explorador

    int nivel=0;
    int new_x=0;
    int prof=0;
    int f=0;
    int y=0;
    int m=0;

```

```

int vacio=0;
String SubMenuBaseInt="";
String MenuBaseInt="";
String st="";
String cod_menu="";
String nom_menu="";
String url="";
String ord="";

nivel=niv+1;
new_x=x;
prof=0;

if (id=="0")
{
    f=0;
}
else
{
    f=1;
}

Connection ConexionSQL6=null;
Statement InstruccionSQL6=null;
ResultSet ResultadoSQL6=null;

Connection ConexionSQL7=null;
Statement InstruccionSQL7=null;
ResultSet ResultadoSQL7=null;

Connection ConexionSQL8=null;
Statement InstruccionSQL8=null;
ResultSet ResultadoSQL8=null;

try{

Class.forName(Driver);
ConexionSQL6 = DriverManager.getConnection(DB,Usr,Pass);
ConexionSQL7 = DriverManager.getConnection(DB,Usr,Pass);
ConexionSQL8 = DriverManager.getConnection(DB,Usr,Pass);

InstruccionSQL6=ConexionSQL6.createStatement();

InstruccionSQL7=ConexionSQL7.createStatement();

InstruccionSQL8=ConexionSQL8.createStatement();

SubMenuBaseInt="Select Count(*) FROM " +
                " Perfiles_Responsable as SUP, Perfiles_Menu as SPM, Menu as SM, Perfielsas
SP " +
                " WHERE SUP.codigo_responsable='"+RESPONSABLE+"' and " +

```

```

        " SUP.codigo_perfil=SP.codifog_perfil and
SP.codigo_perfil=SPM.codigo_perfil and SM.codigo_menu=SPM.codigo_menu "+
        " and SM.superior_menu="+id+" and SM.codigo_menu<>SM.superior_menu";

ResultadoSQL8=InstruccionSQL8.executeQuery(SubMenuBaseInt);
ResultadoSQL8.next();
vacio=ResultadoSQL8.getInt(1);

if (vacio!=0)
{

    SALIDA.println("<table class=tableMen
onMouseOver=show('menu"+id+"','men_"+id+"','"+niv+"')>");
    SALIDA.println("        <tr><td>");
    SALIDA.println("                <table cellpadding=2 cellspacing=0>");

    SubMenuBaseInt="Select SM.* FROM " +
        " Perfiles_Responsable as SUP, Perfiles_Menu as SPM, Menu as SM, Perfiles as
SP " +
        " WHERE SUP.codigo_responsable='"+RESPONSABLE+"' and " +
        " SUP.codigo_perfil=SP.codigo_perfil and SP.codigo_perfil=SPM.codigo_per
fil and SM.codigo_menu=SPM.codigo_menu "+
        " and SM.superior_menu="+id+" and SM.codigo_menu<>SM.superior_menu
ORDER BY Convert(int,SM.orden_menu)";

    ResultadoSQL6=InstruccionSQL6.executeQuery(SubMenuBaseInt);
    while (ResultadoSQL6.next())
    {

        prof+=1;
        cod_menu=ResultadoSQL6.getString(1).trim();
        nom_menu=ResultadoSQL6.getString(2).trim();
        url=ResultadoSQL6.getString(3).trim()+"&USUARIO="+RESPONSABLE;
        ord=ResultadoSQL6.getString(6).trim();

        SubMenuBaseInt="Select Count(*) FROM " +
            " Perfiles_responsables as SUP, Perfiles_menu as SPM, Menu as SM,
Perfiles as SP " +
            " WHERE SUP.codigo_responsable='"+RESPONSABLE+"' and " +
            " SUP.codigo_perfil=SP.codigo_perfil and
SP.codigo_perfil=SPM.codigo_perfil and SM.codigo_menu=SPM.codigo_menu " +
            " and SM.superior_menu="+cod_menu+" and
SM.codigo_menu<>SM.superior_menu";

        ResultadoSQL7=InstruccionSQL7.executeQuery(SubMenuBaseInt);
        ResultadoSQL7.next();

        vacio=ResultadoSQL7.getInt(1);
        if (vacio!=0)
        {
            SALIDA.println("        <tr
onMouseOver='show('menu"+cod_menu+"','men_"+id+"','"+niv+"');show2('menu"+cod_menu+"','menu"+i
d+"','"+(niv+1)+"','"+prof+"') onMouseOut=hide('menu"+cod_menu+"','0')>");
            SALIDA.println("                <td>");

```

```

                SALIDA.println("                <div
id=men_ "+cod_menu+" class=submenu
onMouseOver='show('menu"+cod_menu+"','men_ "+id+"','"+niv+"');show2('menu"+cod_menu+"','menu"+i
d+"','"+(niv+1)+"','"+prof+')' onMouseOut=hide('menu"+cod_menu+"','0')>");
                SALIDA.println("                <A class=submenu
href='"+url+"' target='Principal'>" +ord+"." +nom_menu+"</A><br></div>");
                SALIDA.println("                </td></tr>");
            }
            else
            {
                SALIDA.println("                <tr
onMouseOver=show('menu"+id+"','men_ "+id+"','"+niv+"')><td >");
                SALIDA.println("                <A class=submenu href='"+url+"'
onMouseOver=show('menu"+id+"','men_ "+id+"','"+niv+"')
target='Principal'>" +ord+"." +nom_menu+"</A><br>");
                SALIDA.println("                </td></tr>");
            }
        }
        SALIDA.println("                </table>");
        SALIDA.println("                </td></tr>");
        SALIDA.println("</table>");
    }

} //Fin del try

catch (Exception ex) {
    // Envia el error de regreso al cliente
    SALIDA.println("Contactar al adminsitador del sistema !");
    ex.printStackTrace(SALIDA);
}
finally
{
    try{
        // Siempre cierra apropiadamente
        if (ResultadoSQL6 != null) {
            ResultadoSQL6.close();
        }
        if (ResultadoSQL7 != null) {
            ResultadoSQL7.close();
        }
        if (ResultadoSQL8 != null) {
            ResultadoSQL8.close();
        }

        if (InstruccionSQL6 != null) {
            InstruccionSQL6.close();
        }
        if (InstruccionSQL7 != null) {
            InstruccionSQL7.close();
        }
        if (InstruccionSQL8 != null) {
            InstruccionSQL8.close();
        }

        if (ConexionSQL6 != null) {

```

```

        ConexionSQL6.close();
    }
    if (ConexionSQL7 != null) {
        ConexionSQL7.close();
    }
    if (ConexionSQL8 != null) {
        ConexionSQL8.close();
    }

}
catch (Exception ex)
{
    // Envía el error de regreso al cliente
    SALIDA.println("Contactar al administrador del sistema !!!");
    ex.printStackTrace(SALIDA);
}
}

}

} //Fin de Menu

```

## Generación de página de mapa (mapa.java)

```

import Usuario;
import Menu;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class mapa extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        Usuario MU = new Usuario();
        String dirhtml= MU.DirHTML;

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Mapa</title>");
        Salida.println("<LINK href='"+dirhtml+"include/ourstyle.css\" type=text/css rel=stylesheet>");
    }
}

```

```

    Salida.println("<SCRIPT language=JavaScript src='"+dirhtml+"include/ourScript.js'
type=text/javascript></SCRIPT>");
    Salida.println("<SCRIPT language=JavaScript src='"+dirhtml+"mapita_svg_captura.js'
type=text/javascript></SCRIPT>");
    Salida.println("</head>");
    Salida.println("<body topmargin=0 onload=iniciarMapa(>");

String Usuario=Requerimiento.getParameter("USUARIO");

// Ejecutando Proceso Principal
Proceso(Salida,Usuario);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida

Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

private void Proceso(PrintWriter Salida,String Usuario)
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

/*      Inicializando Variables para Conexión con ODBC      */
Connection ConexionSQL2=null;
Statement InstruccionSQL5=null;
ResultSet ResultadoSQL5=null;
String cod_canton;
String nombre_canton;
String cod_depto;
String nombre_depto;
String cod_mun;
String nombre_mun;

try
{

/*      reacción de la conexión      */

Usuario MIUSUARIO = new Usuario();
String Driver = MIUSUARIO.driver;
String DB = MIUSUARIO.myBD;
String Usr = MIUSUARIO.usuario;
String Pass= MIUSUARIO.password;
String dirhtml= MIUSUARIO.DirHTML;

Class.forName(Driver);

```

```

String myDB = DB;
ConexionSQL2 = DriverManager.getConnection(myDB,Usr,Pass);
InstruccionSQL5=ConexionSQL2.createStatement();

Menu DoMenu = new Menu();

DoMenu.CREARMENU(Usuario,Salida,Driver,Usr,DB,Pass);

DoMenu.Encabezado(Salida);

Salida.println("<table border=0 cellpadding=0 cellspacing=0 width=760 align=center>");
Salida.println("<tr>");
Salida.println("<td align=center>");
Salida.println("    <HEADER><h1>CAPTURA DE OBRAS</h1></HEADER>");
Salida.println("</td>");
Salida.println("</tr>");

Salida.println("<tr>");
Salida.println("<td align=center>");
Salida.println("    <FORM>");
Salida.println("        <SELECT name=selec_depto class=Combos>");
Salida.println("            <OPTION SELECTED>Departamento...</OPTION>");

        String consulta1="SELECT codigo_departamento, nombre_departamento FROM
departamentos ORDER BY nombre_departamento";
        ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);
        while (ResultadoSQL5.next()){
            cod_depto=ResultadoSQL5.getString(1);
            nombre_depto=ResultadoSQL5.getString(2);
            Salida.println("            <OPTION value="+cod_depto+">"+nombre_depto+" ");
        }

Salida.println("        </OPTION>");
Salida.println("    </SELECT>");
Salida.println("    <SELECT name=selec_municip class=Combos>");
Salida.println("        <OPTION SELECTED>Municipio...</OPTION>");

        consulta1="SELECT codigo_municipio, nombre_municipio FROM municipios ORDER
BY nombre_municipio";
        ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);
        while (ResultadoSQL5.next()){
            cod_mun=ResultadoSQL5.getString(1);
            nombre_mun=ResultadoSQL5.getString(2);
            Salida.println("        <OPTION value="+cod_mun+">"+nombre_mun+" ");
        }

Salida.println("        </OPTION>");
Salida.println("    </SELECT>");
Salida.println("    <SELECT name=selec_canton onchange=ircombo(selec_canton.value)
class=Combos>");
Salida.println("        <OPTION SELECTED>Cantón...</OPTION>");

        consulta1="SELECT codigo_canton, nombre_canton FROM cantones ORDER BY
nombre_canton";
        ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);

```

```

        while (ResultadoSQL5.next()){
            cod_canton=ResultadoSQL5.getString(1);
            nombre_canton=ResultadoSQL5.getString(2);
Salida.println("        <OPTION value="+cod_canton+">"+nombre_canton+" ");
        }

Salida.println("                </OPTION>");
Salida.println("            </SELECT>");
Salida.println("        </FORM>");
Salida.println("    </td>");
Salida.println("</tr>");

Salida.println("<tr>");
Salida.println("    <td align=center>");

/*      Se genera el objeto que contendrá el mapa a ser utilizado, el objeto Embed tendrá como src
el servlet que generará sel mapa dinámicamente */

    Salida.println("<EMBED SRC='/servlet/GeneraMapaSVG?' HEIGHT='570' WIDTH='900'
TYPE='image/svg+xml' name=all_map>");
    Salida.println("</td>");
    Salida.println("</tr>");
    Salida.println("<tr>");
    Salida.println("    <td align=center valign=middle>");
    Salida.println("        <FORM name=proyectos>");
    Salida.println("            <table border=0 cellpadding=0 cellspacing=1 align=left width=100%
bgcolor='#BBBBBB'>");
        Salida.println("                <tr>");
            Salida.println("<td align=center><input type=image name='project'
src='/HTML/images/Botones/Boton_proyectos_off.gif'
OnMouseOver='\"BtnOver(this,'Boton_proyectos_off.gif')\"
OnMouseOut='\"BtnOut(this,'Boton_proyectos.gif')\" onclick='\"call_funciones('proyectos');return
false;'></td>");
            Salida.println("<td align=center><input type=image name='needs'
src='/HTML/images/Botones/Boton_needs_off.gif'
OnMouseOver='\"BtnOver(this,'Boton_needs_off.gif')\" OnMouseOut='\"BtnOut(this,'Boton_needs.gif')\"
onclick='\"call_funciones('needs');return false;'></td>");
            Salida.println("<td align=center><input type=image name='geninfo'
src='/HTML/images/Botones/Boton_infogral_off.gif'
OnMouseOver='\"BtnOver(this,'Boton_infogral_off.gif')\"
OnMouseOut='\"BtnOut(this,'Boton_infogral.gif')\" onclick='\"call_funciones('info');return false;'></td>");
// Salida.println("<td align=center><input type=image name='newobra'
src='/HTML/images/Botones/Boton_agregar_obra.gif'
OnMouseOver='\"BtnOver(this,'Boton_agregar_obra_off.gif')\"
OnMouseOut='\"BtnOut(this,'Boton_agregar_obra.gif')\" onclick='mostrar_icons();return false;'></td>");
            Salida.println("<td align=center><input type=image name='save'
src='/HTML/images/Botones/Boton_guardar_obra_off.gif'
OnMouseOver='\"BtnOver(this,'Boton_guardar_obra_off.gif')\"
OnMouseOut='\"BtnOut(this,'Boton_guardar_obra.gif')\" onClick='guardar_icons();return false;'></td>");

        Salida.println("                </tr>");
    Salida.println("            </tr></td>");
    Salida.println("        <input type=hidden name='id_tipo_obra' value='>");
    Salida.println("        <input type=hidden name='id_obra' value='>");
    Salida.println("        <input type=hidden name='pos_obra_x' value='>");
    Salida.println("        <input type=hidden name='pos_obra_y' value='>");

```

```

Salida.println("      <input type=hidden name='nom_obra' value='>");
Salida.println("      <input type=hidden name='id_canton' value='>");
Salida.println("      <input type=hidden name='cod_responsable' value='"+Usuario+">");
Salida.println("      </form>");
Salida.println("</td></tr>");
Salida.println("</table>");
Salida.println("</td>");
Salida.println("</tr>");
Salida.println("</table>");

}
catch (Exception Error)
{
Salida.println("Situación Anormal!");
Error.printStackTrace(Salida);
}
finally
{
try
{
if (ResultadoSQL!=null) ResultadoSQL.close();
if (InstruccionSQL!=null) InstruccionSQL.close();
if (ConexionSQL!=null) ConexionSQL.close();

if (ResultadoSQL5!=null) ResultadoSQL5.close();
if (InstruccionSQL5!=null) InstruccionSQL5.close();
if (ConexionSQL2!=null) ConexionSQL2.close();

}
catch (Exception Error) {}
}
}
}
}

```

## Generación de mapa para captura de obras

(GeneracionMapaSVG.java)

Servlet encargado de la generación del mapa para la captura de obras. Fundamenta su acción en la lectura del archivo SVG base y de la extracción de información de la base de datos y otros elementos.

```

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

public class GeneraMapaSVG extends HttpServlet
{
public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
throws ServletException, IOException

```

```

{

// Abriendo Archivo de Salida
Respuesta.setContentType("text/svg");
PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());
Salida.println("<svg enableZoomAndPanControls=\"false\">");

// Ejecutando Proceso Principal
Proceso(Salida,Requerimiento,Respuesta);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida

Salida.println("</svg>");
Salida.flush();
Salida.close();
}

public void init(ServletConfig cfg) throws ServletException { super.init(cfg); }

public void destroy() { super.destroy(); }

// Detalle del proceso principal
private void Proceso(PrintWriter Salida,HttpServletRequest Requerimiento,HttpServletResponse
Respuesta)
{
    String estilo_grupo;
    String paths_check;
    String estilo_cuadro;
    String estilo_rayita;

    String consulta1="SELECT codigo_unidad, nombre_unidad, imagen_unidad FROM
unidades_medida";
    String consulta2;
    String unidad="";
    String unidad_nombre="";
    int coorx;
    int coory;
    String obra;
    int yposcheck;
    int yposchtext;
    String unidad_imagen="";

////////////////////////////////// Inicializando Variables para Conexión con ODBC//////////////////////////////////
    Connection ConexionSQL2=null;
    Connection ConexionSQL3=null;
    Statement InstruccionSQL5=null;
    ResultSet ResultadoSQL5=null;
    Statement InstruccionSQL6=null;
    ResultSet ResultadoSQL6=null;

    try
    {

```

//**Creación de la conexión**//

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
String myDB = "jdbc:odbc:Driver={SQL Server};Server=LOBOS-
D8K1Y6P5F;Database=PROYECTOS";
ConexionSQL2 = DriverManager.getConnection(myDB,"sa","");
ConexionSQL3 = DriverManager.getConnection(myDB,"sa","");
```

//**Declaración/Inicialización de variables para escritura del svg**//

```
InstruccionSQL5=ConexionSQL2.createStatement();
InstruccionSQL6=ConexionSQL3.createStatement();

File datain = new File("C:\\HTML\\mapa_15Feb_captura.svg");
String loquebusco = " id=\"obras\"";
String loquebusc1 = " id=\"check\"";
String loquebusc2 = " id=\"simbl\"";
int sombrax;
int sombray;
String prueba;
int fin=0;
int inicio=0;
yposcheck=50;
yposchtext=10;

estilo_grupo=" style=\"fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-
miterlimit:4;\"> ";
paths_check= "<path style=\"fill:none;stroke:#663300;\" d=\"M0.5,9.9510-9.44h9.47\"/>
<path style=\"fill:none;stroke:#996600;\" d=\"M10.48,01-0.02,10.451-10.44,0\"/> <path
style=\"fill:none;stroke:#AA9900;\" d=\"M1.48,8.9510.01-7.47L8.98,1.5\"/> <path
style=\"fill:none;stroke:#CC9933;\" d=\"M9.47,110,8.46L0.98,9.45\"/>";
estilo_cuadro=" style=\"fill:#FFFFCC;stroke:none;\"
d=\"M8.96,8.95H1.98V1.98h6.97v6.96z\"/>";
estilo_rayita=" style=\"fill:none;stroke:#006600;stroke-width:2;\"
d=\"M2.91,4.3611.55,2.6313.73-4.24\"/>";
```

//**Verificación de la existencia del archivo svg Base**//

```
if (datain.exists()){
    BufferedReader in = new BufferedReader(new FileReader(datain));
    String line = in.readLine();
```

//**Lectura línea por línea del archivo svg base**//

```
while(line != null){

    Salida.println(line);
    line.trim();
    for (inicio=0;inicio<line.length()-loquebusco.length();inicio++){
        fin=inicio+loquebusco.length();
        prueba=line.substring(inicio,fin);
```

// **Inserción de checks para las obras**//

```
yposchtext=10;
```

```

if ( loquebusc1.equalsIgnoreCase(prueba) ){
    ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);
    Salida.println("<text><tspan x=\"650\" y=\"30\"
style=\"stroke:none\">OBRAS:</tspan></text>");
    while (ResultadoSQL5.next()){
        unidad=ResultadoSQL5.getString(1);
        unidad_nombre=ResultadoSQL5.getString(2);
        unidad_imagen=ResultadoSQL5.getString(3);
        Salida.println("<g id=\"check_\"+unidad+\"\"
transform=\"translate(645,\"+yposcheck+\" )\" ");
        Salida.println(estilo_grupo);
        Salida.println(paths_check);
        Salida.println("<text><tspan x=\"17\" y=\"\"+yposchtext+\"\"
style=\"stroke:none;font-size:10\">\"+unidad_nombre+\"</tspan></text>");
        Salida.println("<image id=\"obra_\"+unidad+\"\" x=\"-18\"
y=\"\"+(yposchtext-10)+\"\" height=\"15\" width=\"15\"
xlink:href=\"/HTML/images/iconos/\"+unidad_imagen+\"\"/>");
        Salida.println("<path onclick=\"mostrar('check'+unidad+\"
'capa_\"+unidad+\" )\" ");
        Salida.println(estilo_cuadro);
        Salida.println("<path id=\"check\"+unidad+\"\"
onclick=\"ocultar(evt,'capa_\"+unidad+\" )\" ");
        Salida.println(estilo_rayita);
        Salida.println("</g>");
        yposcheck+=20;
        yposchtext+=0.5;
    }// Fin while
}/// fin If

```

////////////////////////////////// **Inserción de símbolos para las obras**//////////////////////////////////

```

if ( loquebusc2.equalsIgnoreCase(prueba) ){

    yposchtext=30;
    int xposchtext=150;

    Salida.println("<g id=\"simbolos\" onload=\"SVGIcons(evt)\"
onclick=\"mostrar_simbolos(evt)\">");
    ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);
    Salida.println("<text><tspan x=\"650\" y=\"30\"
style=\"stroke:none\">ICONOS:</tspan></text>");
    while (ResultadoSQL5.next()){
        unidad=ResultadoSQL5.getString(1);
        unidad_nombre=ResultadoSQL5.getString(2);
        unidad_imagen=ResultadoSQL5.getString(3);
        Salida.println("<a><g id=\"dibobra_\"+unidad+\"\"
transform=\"translate(640,\"+yposchtext+\" )\" onclick=\"sombra_agrega(\"+unidad+\" )\"> ");
        Salida.println("<rect id=\"borde_agrega_\"+unidad+\"\" x=\"-
12\" y=\"\"+(yposchtext-12)+\"\" width=\"19\" height=\"19\" style=\"fill:#dddddd;visibility:hidden;\"/> ");
        Salida.println("<image id=\"\"+unidad+\"\" x=\"-10\"
y=\"\"+(yposchtext-10)+\"\" height=\"15\" width=\"15\"
xlink:href=\"/HTML/images/iconos/\"+unidad_imagen+\"\"/>");
        Salida.println("<text><tspan x=\"13\" y=\"\"+(yposchtext)+\"\"
style=\"stroke:none\">\"+unidad_nombre+\"</tspan></text>");
        Salida.println("</g></a>");
    }
}

```

```

        yposchtext+=10;
    }
    Salida.println("</g>");

}/// fin If

////////////////////////////////// Inserción de obras en el mapa//////////////////////////////////

    if ( loquebusco.equalsIgnoreCase(prueba) ){
        ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);
        while (ResultadoSQL5.next()){
            unidad=ResultadoSQL5.getString(1);
            unidad_imagen=ResultadoSQL5.getString(3);
            Salida.println("<g id=\"+capa_+unidad+\">");
            consulta2="SELECT codigo_obra,coorx_obra,coory_obra
FROM OBRAS_FISICAS WHERE codigo_unidad=\"+unidad+\"";
            ResultadoSQL6=InstruccionSQL6.executeQuery(consulta2);
            while (ResultadoSQL6.next()){
                obra=ResultadoSQL6.getString(1);
                coorx=ResultadoSQL6.getInt(2);
                coory=ResultadoSQL6.getInt(3);
                sombrax=coorx-2;
                sombray=coory-2;
                Salida.println("<a><g id=\""+obra+\"
transform=\"translate(\"+coorx+\",\"+coory+)\
onmousedown=\"parametros(evt,\"+obra+)\
onclick=\"sombra(\"+obra+)\>");
                Salida.println("<rect id=\"borde_\"+obra+\" x=\"-2\"
y=\"-2\" width=\"15\" height=\"14\" style=\"fill:#dddddd;visibility:hidden;\
");
                Salida.println("<image id=\""+unidad+\"
height=\"10\" width=\"11\" xlink:href=\"/HTML/images/iconos/\"+unidad_imagen+\">");
                Salida.println("</g></a>");
            }/// fin While
            Salida.println("</g>");
        }/// fin While
        ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);
        while (ResultadoSQL5.next()){
            unidad=ResultadoSQL5.getString(1);
            unidad_imagen=ResultadoSQL5.getString(3);
            Salida.println("<g id=\""+unidad+\"_obra\" clone=\"true\"
transform=\"translate(0,0)\
style=\"visibility:hidden;\>");
            Salida.println("<image height=\"10\" width=\"11\"
xlink:href=\"/HTML/images/iconos/\"+unidad_imagen+\">");
            Salida.println("</g>");
        }/// fin While
    }/// fin If
}/// fin For
line = in.readLine();
}/// Fin While

////////////////////////////////// Fin lectura svg base... Fin de insercion de elementos//////////////////////////////////

    in.close();
    //out.close();
    //System.out.println("Seacabo");
}/// Fin If

```

```

        else
            System.out.println("File not found - example.txt");
    }/// fin try

//////////////////////////////////////Cierre de variables-conexiones//////////////////////////////////////

catch (Exception Error)
{
    Salida.println("Situación Anormal!");
    Error.printStackTrace(Salida);
}
finally
{
    try
    {
        if (ResultadoSQL5!=null) ResultadoSQL5.close();
        if (InstruccionSQL5!=null) InstruccionSQL5.close();
        if (ResultadoSQL6!=null) ResultadoSQL6.close();
        if (InstruccionSQL6!=null) InstruccionSQL6.close();
        if (ConexionSQL2!=null) ConexionSQL2.close();
        if (ConexionSQL3!=null) ConexionSQL3.close();
    }
    catch (Exception Error) {}
}
}
}
}

```

## Generación de Mapa de Reportes (GeneracionMapaSVGReportes.java)

Servlet encargado de la generación del mapa de reportes. Fundamenta su acción en la lectura del archivo SVG base y de la extracción de información de la base de datos a través de consultas y procesos almacenados.

```

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

public class GeneraMapaSVGReportes extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {
        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/svg");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        Salida.println("<svg enableZoomAndPanControls=\"false\">");

        // Ejecutando Proceso Principal
        Proceso(Salida,Requerimiento,Respuesta);
    }
}

```

**// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida**

```
Salida.println("</svg>");
Salida.flush();
Salida.close();
}
```

```
public void init(ServletConfig cfg) throws ServletException { super.init(cfg); }
```

```
public void destroy() { super.destroy(); }
```

```
private void Proceso(PrintWriter Salida,HttpServletRequest Requerimiento,HttpServletResponse
Respuesta)
{
```

```
////////////////////Iniciación variables de reportes////////////////////
```

```
String estilo_grupo;
String paths_check;
String estilo_cuadro;
String estilo_rayita;
String institucion=Requerimiento.getParameter("Instituciones");
String anio=Requerimiento.getParameter("Anio");
String consulta1="SELECT codigo_unidad, nombre_unidad, imagen_unidad FROM
unidades_medida";
String consulta2;
String unidad="";
String unidad_nombre="";
int coorx;
int coory;
String obra;
int yposcheck;
int yposchtext;
String unidad_imagen="";
String cid;
String cporc;
String nom_obra="";
String monto_obra="";
String num_proys="";
String num_nec="";
String num_empleo="";
String porcentaje="";
String nom_obra_det="";
String monto_obra_det="";
String num_proys_det="";
String num_nec_det="";
String num_empleo_det="";
String porcentaje_det="";
```

```
//////////////////// Inicializando Variables para Conexión con ODBC////////////////////
```

```
Connection ConexionSQL2=null;
Connection ConexionSQL3=null;
Connection ConexionSQL7=null;
Statement InstruccionSQL5=null;
ResultSet ResultadoSQL5=null;
```

```

Statement InstruccionSQL6=null;
ResultSet ResultadoSQL6=null;
Statement InstruccionSQL7=null;
ResultSet ResultadoSQL7=null;

try
{

////////////////////////////////////Creación de la conexión////////////////////////////////////

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
String myDB = "jdbc:odbc:Driver={SQL Server};Server=LOBOS-
D8K1Y6P5F;Database=PROYECTOS";
ConexionSQL2 = DriverManager.getConnection(myDB,"sa","");
ConexionSQL3 = DriverManager.getConnection(myDB,"sa","");
ConexionSQL7 = DriverManager.getConnection(myDB,"sa","");

////////////////////////////////////Declaración/Inicialización de variables para escritura del svg //////////////////////////////////////

InstruccionSQL5=ConexionSQL2.createStatement();
InstruccionSQL6=ConexionSQL3.createStatement();
InstruccionSQL7=ConexionSQL7.createStatement();

File datain = new File("C:\\HTML\\mapa_15Feb_reportes.svg");

String loquebusco = " id=\"obras\"";
String loquebusc1 = " id=\"check\"";
String loquebusc2 = " id=\"simbl\"";
String loquebusc3 = " id=\"toolt\"";
String loquebusc4 = " id=\"repfn\"";
String loquebusc5 = " id=\"repfs\"";
int sombrax;
int sombray;
String prueba;
int fin=0;
int inicio=0;
yposcheck=50;
yposchtext=10;

estilo_grupo=" style=\"fill-rule:nonzero;clip-rule:nonzero;stroke:#000000;stroke-
miterlimit:4;\"> ";
paths_check= "<path style=\"fill:none;stroke:#663300;\" d=\"M0.5,9.9510-9.44h9.47\"/>
<path style=\"fill:none;stroke:#996600;\" d=\"M10.48,0l-0.02,10.45l-10.44,0\"/> <path
style=\"fill:none;stroke:#AA9900;\" d=\"M1.48,8.9510.01-7.47L8.98,1.5\"/> <path
style=\"fill:none;stroke:#CC9933;\" d=\"M9.47,110,8.46L0.98,9.45\"/>";
estilo_cuadro=" style=\"fill:#FFFFCC;stroke:none;\"
d=\"M8.96,8.95H1.98V1.98h6.97v6.96z\"/>";
estilo_rayita=" style=\"fill:none;stroke:#006600;stroke-width:2;\"
d=\"M2.91,4.36l1.55,2.63l3.73-4.24\"/>";

////////////////////////////////////Verificación de la existencia del archivo svg Base////////////////////////////////////

if (datain.exists()){
BufferedReader in = new BufferedReader(new FileReader(datain));

```

```

String line = in.readLine();

//////////////////////////////////Lectura linea por linea del archivo svg base//////////////////////////////////

while(line != null){

    Salida.println(line);
    line.trim();
    for (inicio=0;inicio<line.length()-loquebusco.length();inicio++){
        fin=inicio+loquebusco.length();
        prueba=line.substring(inicio,fin);

////////////////////////////////// Inserción de checkes para las obras//////////////////////////////////

        yposchtext=10;

        if ( loquebusc1.equalsIgnoreCase(prueba) ){
            ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);
            Salida.println("<text><tspan x=\"650\" y=\"30\"
style=\"stroke:none\">OBRAS:</tspan></text>");
            while (ResultadoSQL5.next()){
                unidad=ResultadoSQL5.getString(1);
                unidad_nombre=ResultadoSQL5.getString(2);
                unidad_imagen=ResultadoSQL5.getString(3);
                Salida.println("<g id=\"check_\"+unidad+\"\"
transform=\"translate(640,\"+yposcheck+\" )\" ");
                Salida.println(estilo_grupo);
                Salida.println(paths_check);
                Salida.println("<text><tspan x=\"13\" y=\"\"+yposchtext+\"\"
style=\"stroke:none\">\"+unidad_nombre+\"</tspan></text>");
                Salida.println("<image id=\"obra_\"+unidad+\"\" x=\"-15\"
y=\"\"+(yposchtext-10)+\"\" height=\"10\" width=\"11\"
xlink:href=\"/HTML/images/iconos/\"+unidad_imagen+\"\"/>");
                Salida.println("<path onclick=\"mostrar('check'+unidad+\"
'capa_\"+unidad+\" )\" ");
                Salida.println(estilo_cuadro);
                Salida.println("<path id=\"check\"+unidad+\"\"
onclick=\"ocultar(evt,'capa_\"+unidad+\" )\" ");
                Salida.println(estilo_rayita);
                Salida.println("</g>");
                yposcheck+=20;
                yposchtext+=0.5;
            }// Fin while
        }// fin If

////////////////////////////////// Inserción de simbolos para las obras//////////////////////////////////

        if ( loquebusc2.equalsIgnoreCase(prueba) ){

            yposchtext=30;
            int xposchtext=150;
            Salida.println("<g id=\"simbolos\" onload=\"SVGIcons(evt)\">");////////
onclick=\"mostrar_simbolos(evt)\">");
            ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);

```

```

        Salida.println("<text><tspan x=\"650\" y=\"30\"
style=\"stroke:none\">ICONOS:</tspan></text>");
        while (ResultadoSQL5.next()){
            unidad=ResultadoSQL5.getString(1);
            unidad_nombre=ResultadoSQL5.getString(2);
            unidad_imagen=ResultadoSQL5.getString(3);
            Salida.println("<a><g id=\"dibobra_"+unidad+"\"
transform=\"translate(640,\"+yposchtext+"")\"> ");
            Salida.println("<image id=\""+unidad+"\" x=\"-10\"
y=\""+(yposchtext-10)+"\" height=\"15\" width=\"15\"
xlink:href=\"/HTML/images/iconos/\"+unidad_imagen+"\"/>");
            Salida.println("<text><tspan x=\"13\" y=\""+(yposchtext)+"\"
style=\"stroke:none\">"+unidad_nombre+"</tspan></text>");
            Salida.println("</g></a>");
            yposchtext+=10;
        }
        Salida.println("</g>");
    }/// fin If

```

////////////////////////////////// **Inserción de obras en el mapa**//////////////////////////////////

```

    if ( loquebusco.equalsIgnoreCase(prueba) ){
        ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);
        while (ResultadoSQL5.next()){
            unidad=ResultadoSQL5.getString(1);
            unidad_imagen=ResultadoSQL5.getString(3);
            Salida.println("<g id=\"capa_"+unidad+"\">");
            consulta2="SELECT codigo_obra,coorx_obra,coory_obra
FROM OBRAS_FISICAS WHERE codigo_unidad=\""+unidad+"\"";
            ResultadoSQL6=InstruccionSQL6.executeQuery(consulta2);
            while (ResultadoSQL6.next()){
                obra=ResultadoSQL6.getString(1);
                coorx=ResultadoSQL6.getInt(2);
                coory=ResultadoSQL6.getInt(3);
                nom_obra="Nombre Obra: ";
                monto_obra="Monto invertido: ";
                num_proys="Proyectos asignados: ";
                num_nec="Necesidades: ";
                num_empleo="Empleos generados: ";
                porcentaje="% promedio global: ";

                sombrax=coorx-2;
                sombray=coory-2;

                Salida.println("<a><g id=\""+obra+"\"
transform=\"translate(\"+coorx+\",\"+coory+\")\" onmousedown=\"parametros(evt,\"+obra+\")\"
onmouseover=\"estetooltip(\"+obra+\")\" onmouseout=\"notooltip(\"+obra+\")\"
onclick=\"sombra(\"+obra+\")\">");
                Salida.println("<rect id=\"borde_"+obra+"\" x=\"-2\"
y=\"-2\" width=\"15\" height=\"14\" style=\"fill:#dddddd;visibility:hidden;\"/> ");
                Salida.println("<image id=\""+unidad+"\"
height=\"10\" width=\"11\" xlink:href=\"/HTML/images/iconos/\"+unidad_imagen+"\"/>");

                Salida.println("</g></a>");
            }/// fin While

```

```

        Salida.println("</g>");
    }/// fin While
    ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);
    while (ResultadoSQL5.next()){
        unidad=ResultadoSQL5.getString(1);
        unidad_imagen=ResultadoSQL5.getString(3);
        Salida.println("<g id=\""+unidad+"_obra\" clone=\"true\"
transform=\"translate(0,0)\" style=\"visibility:hidden;\">");
        Salida.println("<image height=\"10\" width=\"11\"
xlink:href=\"/HTML/images/iconos/\"+unidad_imagen+\"\"/>");
        Salida.println("</g>");
    }/// fin While
}/// fin If

```

////////////////////////////////// **Inserción de tooltip de obras en el mapa**//////////////////////////////////

```

if ( loquebusc3.equalsIgnoreCase(prueba) ){
    ResultadoSQL5=InstruccionSQL5.executeQuery(consulta1);
    while (ResultadoSQL5.next()){
        unidad=ResultadoSQL5.getString(1);
        unidad_imagen=ResultadoSQL5.getString(3);
        consulta2="SELECT codigo_obra,coorx_obra,coory_obra
FROM OBRAS_FISICAS WHERE codigo_unidad=\""+unidad+"\"";
        ResultadoSQL6=InstruccionSQL6.executeQuery(consulta2);
        while (ResultadoSQL6.next()){
            obra=ResultadoSQL6.getString(1);
            coorx=ResultadoSQL6.getInt(2);
            coory=ResultadoSQL6.getInt(3);
            nom_obra="Nombre Obra: ";
            monto_obra="Monto invertido: ";
            num_proys="Proyectos asignados: ";
            num_nec="Necesidades: ";
            num_empleo="Empleos generados: ";
            porcentaje="% promedio global: ";
            consulta2="EXEC ToolTipObra "+obra;

ResultadoSQL7=InstruccionSQL7.executeQuery(consulta2);
            while (ResultadoSQL7.next()){
                nom_obra_det=ResultadoSQL7.getString(1);

monto_obra_det=ResultadoSQL7.getString(2);

num_proys_det=ResultadoSQL7.getString(3);
                num_nec_det=ResultadoSQL7.getString(4);

num_empleo_det=ResultadoSQL7.getString(5);
                porcentaje_det=ResultadoSQL7.getString(6);

                Salida.println("<g id=\"tip_\"+obra+\"\"
style=\"visibility:hidden;\">");
                Salida.println("<rect x=\"5\" y=\"5\"
width=\"309\" height=\"100\" style=\"stroke:black;fill:#aacc99;stroke-width:1\" rx=\"10\" ry=\"10\"/>");
                Salida.println("<text style=\"font-
size:11;font-face:'Century Gothic\"><span x=\"20\" y=\"25\"
style=\"stroke:none\">"+nom_obra+\"</span><span
style=\"stroke:none;fill:#555555\">"+nom_obra_det+\"</span>");

```

```

Salida.println("<tspan x=\"20\" y=\"37\"
style=\"stroke:none\">"+monto_obra+"</tspan><tspan
style=\"stroke:none;fill:#555555\">"+monto_obra_det+"</tspan>");
Salida.println("<tspan x=\"20\" y=\"51\"
style=\"stroke:none\">"+num_proys+"</tspan><tspan
style=\"stroke:none;fill:#555555\">"+num_proys_det+"</tspan>");
Salida.println("<tspan x=\"20\" y=\"65\"
style=\"stroke:none\">"+num_nec+"</tspan><tspan
style=\"stroke:none;fill:#555555\">"+num_nec_det+"</tspan>");
Salida.println("<tspan x=\"20\" y=\"77\"
style=\"stroke:none\">"+num_empleo+"</tspan><tspan
style=\"stroke:none;fill:#555555\">"+num_empleo_det+"</tspan>");
Salida.println("<tspan x=\"20\" y=\"89\"
style=\"stroke:none\">"+porcentaje+"</tspan><tspan
style=\"stroke:none;fill:#555555\">"+porcentaje_det+"</tspan></text>");
Salida.println("</g>");
}///fin While
}/// fin While
}/// fin While
}/// fin If

////////////////////////////////// Inserción de datos de reporte financiero en el mapa//////////////////////////////////

if ( loquebusc4.equalsIgnoreCase(prueba) ){
consulta2="EXEC PorcentajeAvanceFinancieroCanton
"+institucion+", "+anio;

ResultadoSQL5=InstruccionSQL5.executeQuery(consulta2);
while (ResultadoSQL5.next()){
cid=ResultadoSQL5.getString(1);
cporc=ResultadoSQL5.getString(2);
Salida.println("<rect id=\"reportfin_cantones_"+cid+"\"
x=\"0\" y=\"0\" width=\"1\" height=\"1\" valor=\""+cporc+"\"/>");
}/// fin While
}/// fin If

////////////////////////////////// Inserción de datos de reporte fisico en el mapa//////////////////////////////////

if ( loquebusc5.equalsIgnoreCase(prueba) ){
consulta2="EXEC PorcentajeAvanceFisicoCanton
"+institucion+", "+anio;

ResultadoSQL5=InstruccionSQL5.executeQuery(consulta2);
while (ResultadoSQL5.next()){
cid=ResultadoSQL5.getString(1);
cporc=ResultadoSQL5.getString(2);
Salida.println("<rect id=\"reportfis_cantones_"+cid+"\" x=\"0\"
y=\"0\" width=\"1\" height=\"1\" valor=\""+cporc+"\"/>");
}/// fin While
}/// fin If
}/// fin For
line = in.readLine();
}/// Fin While

////////////////////////////////// Fin lectura svg base... Fin de insercion de elementos//////////////////////////////////

in.close();
}/// Fin If

```

```

        else
            System.out.println("File not found - example.txt");
    }/// fin try

catch (Exception Error)
{
    Salida.println("Situación Anormal!");
    Error.printStackTrace(Salida);
}
finally
{
    try
    {
        if (ResultadoSQL5!=null) ResultadoSQL5.close();
        if (InstruccionSQL5!=null) InstruccionSQL5.close();
        if (ResultadoSQL6!=null) ResultadoSQL6.close();
        if (InstruccionSQL6!=null) InstruccionSQL6.close();
        if (ResultadoSQL7!=null) ResultadoSQL7.close();
        if (InstruccionSQL7!=null) InstruccionSQL7.close();
        if (ConexionSQL2!=null) ConexionSQL2.close();
        if (ConexionSQL3!=null) ConexionSQL3.close();
        if (ConexionSQL7!=null) ConexionSQL7.close();
    }
    catch (Exception Error) {}
}
}
}
}

```

## Captura de obras

Este servlet es el encargado de llevar a cabo el registro de las obras ingresadas por el usuario desde la pantalla de mapa.

```

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class CapturaObras extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
    }
}

```

```
Salida.println("<title>Prueba Conexion BDD</title>");
Salida.println("</head>");
Salida.println("<body>");
```

```
String id_tipo_obra=Requerimiento.getParameter("id_tipo_obra");
String pos_obra_x=Requerimiento.getParameter("pos_obra_x");
String pos_obra_y=Requerimiento.getParameter("pos_obra_y");
String nombre_obra=Requerimiento.getParameter("nombre_obra");
String id_canton=Requerimiento.getParameter("id_canton");
String Usuario=Requerimiento.getParameter("Usuario");
```

```
// Ejecutando Proceso Principal
Proceso(Salida,id_tipo_obra,pos_obra_x,pos_obra_y,nombre_obra,id_canton,Usuario);
```

```
// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("<script language=JavaScript>");
Salida.println("alert('La obra ha sido almacenada satisfactoriamente')");
Salida.println("window.close()");
Salida.println("</script>");
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}
```

```
private void Proceso(PrintWriter Salida,String id_obra_fisica,String pos_obra_x,String pos_obra_y,String
nombre_obra,String id_canton, String Usuario)
{
```

```
// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;
```

```
try
{
```

```
// Iniciando Conexión
```

```
    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;
```

```
    Class.forName(Driver);
    String myDB = DB;
    ConexionSQL = DriverManager.getConnection(myDB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();
```

```

        InstruccionSQL.executeUpdate("insert into OBRAS_FISICAS values
("+id_obra_fisica+", "+pos_obra_x+", "+pos_obra_y+", "+nombre_obra+", "+id_canton+");

        String consulta= "select Max(codigo_obra) as id_obra from obras_fisicas";
        ResultadoSQL=InstruccionSQL.executeQuery(consulta);
        ResultadoSQL.next();
        String id_obra=ResultadoSQL.getString(1);

        Salida.println("<script language=javascript>");
        Salida.println("self.opener.habilitar_boton("+id_obra+", "+Usuario+"");");
        Salida.println("</script>");

    }
    catch (Exception Error)
    {
        Salida.println("Situación Anormal!");
        Error.printStackTrace(Salida);
    }
    finally
    {
        try
        {
            if (ResultadoSQL!=null) ResultadoSQL.close();
            if (InstruccionSQL!=null) InstruccionSQL.close();
            if (ConexionSQL!=null) ConexionSQL.close();
        }
        catch (Exception Error) {}
    }
}
}
}

```

## **Generación de Captura Unidades de Medida** (CapturaCaracteristicasUnidadesMedida.java)

Servlet que genera la pantalla de captura para las características de las unidades de medida nombre unidad de medida, nombre de archivo a adjuntar.

```

/*****/
// Servlet Utilizado para Adjuntar Archivos a un Documento
/*****/
import Usuario;
import Menu;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.io.*;

public class CapturaCaracteristicasUnidadesMedida extends HttpServlet
{
    /*****/
    /*          funcion doGet          */
}

```

```

/*****/
public void doGet(HttpServletRequest SOLICITUD,
                 HttpServletResponse RESPUESTA)
    throws ServletException, java.io.IOException
{

    //AQUI SE DECLARA QUE LA SALIDA SERA UN HTML
    RESPUESTA.setContentType("text/html");
    PrintWriter SALIDA = new PrintWriter(RESPUESTA.getOutputStream());

    /***/
    /* Impresion del Encabezado de la Pantalla */
    /***/
    SALIDA.println("<html>");
    SALIDA.println("<head>");
    SALIDA.println(" <title>Ingreso Unidades de Medida</title>");
    SALIDA.println("<link href='/HTML/include/ourstyle.css' rel='stylesheet'
type='text/css'>");
    SALIDA.println("<SCRIPT language=JavaScript src='/HTML/include/ourScript.js'
type=text/javascript></SCRIPT>");
    SALIDA.println("</head>");
    SALIDA.println("<body>");

    PRINCIPAL(SALIDA,SOLICITUD,RESPUESTA);

    /***/
    /***/ Cerrando la conexiona al HTML *****/
    /***/
    SALIDA.println("</body>");
    SALIDA.println("</html>");
    SALIDA.flush();
    SALIDA.close();

} /***/ FIN DE LA FUNCION doGet *****/

/*****/
/* Este metodo despliega una pantalla de captura */
/* adecuado a la tabla que se esta procesando */
/*****/
private void PRINCIPAL(java.io.PrintWriter SALIDA,
                      HttpServletRequest SOLICITUD,HttpServletResponse RESPUESTA)
{
    /***/
    /* Creando las propiedades necesarias para conectar con la Base de Datos */
    /***/

    /***/
    /* Inicializando a null elementos de conexion */
    /***/
    Connection CON1 = null;
    Statement STMT1 = null;
    ResultSet RS1 = null;

```

```

try
{
    /*******
    /*      Creando elementos necesarios para establecer y      */
    /*      utilizar la conexion      */
    /*******
    String Usuario = SOLICITUD.getParameter("USUARIO");

// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    Menu DoMenu = new Menu();

//      DoMenu.CREARMENU(Usuario,SALIDA,Driver,Usr,DB,Pass);

//      DoMenu.Encabezado(SALIDA);

        Class.forName(Driver);
        CON1 = DriverManager.getConnection(DB,Usr,Pass);
        STMT1=CON1.createStatement();

        SALIDA.println("<table border=1 class=colortable2 align=center>");

        SALIDA.print("<tr><td colspan=2><form action='/servlet/CapturaUnidadesMedida?'
method='Post'></td></tr>");
        SALIDA.println("<tr><td colspan=2 align=center class=trbotones><b>Paso
1:</b></td>");
        SALIDA.println("<tr><td colspan=2 align=center class=hcolor>Características de la
Unidad de Medida</td>");
        SALIDA.println("<tr><td class=tdfont>Nombre unidad medida:</td><td><input
type=text name=nombre_unidad size=50 maxlength=17></td>");
        SALIDA.println("<tr><td class=tdfont>Nombre del archivo:</td><td><input type=text
name=imagen_unidad size=50></td>");
        SALIDA.println("<tr><td colspan=2 align=center><input type='submit'
value='Aceptar'></td></tr>");
        SALIDA.println("<input type=hidden name=USUARIO value="+Usuario+">");
        SALIDA.println("<input type=hidden name=insertar value=T>");
        SALIDA.println("</form>");

} //fin del try

catch (Exception ex)
{
    SALIDA.println("Exception 1!");
    ex.printStackTrace(SALIDA);
}

```

```

    }
    finally
    {
        try
        {
            /***/
            /* Cerrando los elementos de la conexion */
            /***/
            if (RS1 != null)
            {
                RS1.close();
            }

            if (STMT1 != null)
            {
                STMT1.close();
            }

            if (CON1 != null)
            {
                CON1.close();
            }

        }
        catch (Exception ex)
        {
            SALIDA.println("ERROR DE CIERRE DE ELEMENTOS DE CONEXION !!!");
            ex.printStackTrace(SALIDA);
        }
    } //FIN DEL FINALLY
}

public void doPost(HttpServletRequest SOLICITUD,
    HttpServletResponse RESPUESTA)
    throws ServletException, java.io.IOException
{
    doGet(SOLICITUD,RESPUESTA);
}
}

```

## Genera Captura de Archivo a Cargar (CapturaUnidadesMedida.java)

Servlet responsable de generar la pantalla de captura para seleccionar la imagen, posteriormente se encarga de realizar el upload del archivo al directorio donde se encuentran los iconos de unidades de medida.

```

/***/
// Servlet Utilizado para Adjuntar Archivos a un Documento
/***/

import Usuario;

```

```

import Menu;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.io.*;

public class CapturaUnidadesMedida extends HttpServlet
{

    /**
     *
     */
    /**
     *
     */
    public void doGet(HttpServletRequest SOLICITUD,
        HttpServletResponse RESPUESTA)
        throws ServletException, java.io.IOException
    {

        //AQUI SE DECLARA QUE LA SALIDA SERA UN HTML
        RESPUESTA.setContentType("text/html");
        PrintWriter SALIDA = new PrintWriter(RESPUESTA.getOutputStream());

        /**
         *
         */
        /* Impresion del Encabezado de la Pantalla */
        /**
         *
         */
        SALIDA.println("<html>");
        SALIDA.println("<head>");
        SALIDA.println(" <title>Ingreso Unidades de Medida</title>");
        SALIDA.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\"
type=\"text/css\">");
        SALIDA.println("<SCRIPT language=JavaScript src=/HTML/include/ourScript.js'
type=text/javascript></SCRIPT>");

        SALIDA.println("<script>");

        SALIDA.println("function Cerrar(){");
        SALIDA.println("window.opener.location.reload()");
        SALIDA.println("window.close()");
        SALIDA.println("}");

        SALIDA.println("</script>");
        SALIDA.println("</head>");

        PRINCIPAL(SALIDA,SOLICITUD,RESPUESTA);

        /**
         *
         */
        /**
         *
         */
        /**
         *
         */
        SALIDA.println("</body>");
        SALIDA.println("</html>");
        SALIDA.flush();
        SALIDA.close();

    } /**
     *
     */
}

```

```

/*****
/*     Este metodo despliega una pantalla de captura     */
/*     adecuado a la tabla que se esta procesando     */
/*****
private void PRINCIPAL(java.io.PrintWriter SALIDA,
                      HttpServletRequest SOLICITUD,HttpServletResponse RESPUESTA)

{
/*****
/* Creando las propiedades necesarias para conectar con la Base de Datos */
/*****

/*****
/* LECTURA DE PARAMETROS */
/*****

String ADBOTON=SOLICITUD.getParameter("ADBOTON");
    String nombre_unidad=SOLICITUD.getParameter("nombre_unidad");
    String imagen_unidad=SOLICITUD.getParameter("imagen_unidad");
String insertar=SOLICITUD.getParameter("insertar");

/*****
/*     Inicializando a null elementos de conexion     */
/*****
Connection CON1 = null;
Statement STMT1 = null;
ResultSet RS1 = null;

try
{
/*****
/*     Creando elementos necesarios para establecer y     */
/*     utilizar la conexion     */
/*****

String Usuario = SOLICITUD.getParameter("USUARIO");

// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    Menu DoMenu = new Menu();

// DoMenu.CREARMENU(Usuario,SALIDA,Driver,Usr,DB,Pass);

```

```

//      DoMenu.Encabezado(SALIDA);

        Class.forName(Driver);
        CON1 = DriverManager.getConnection(DB,Usr,Pass);
        STMT1=CON1.createStatement();

        SALIDA.println("<body>");
        SALIDA.println("<Font face=Verdana size=2>");
        SALIDA.println("<table border=0>");

        if(ADBOTON==null)
        {
            SALIDA.print("<table ><tr><td colspan=2><form
action='/servlet/CapturaUnidadesMedida?ADBOTON=Adjuntar&nombre_unidad="+nombre_unidad+"&i
magen_unidad="+imagen_unidad+"&insertar="+insertar+" method='Post' OnSubmit='\return
Valido(this);' ">");
            SALIDA.println("<enctype='multipart/form-data' ></td></tr></table>");
            SALIDA.println("<table class=colortable2 border=1> <tr><td colspan=2 class=trbotones
width=100% align=center><b>Paso 2:</b></td></tr>");
            SALIDA.print("<tr><td colspan=2><font size=2 face=verdana>Haga clic en
<b>Browse</b> y seleccione un archivo. Luego, haga clic en Aceptar.<br><br></td></tr>");
            SALIDA.println("<tr><td class=hcolor>Seleccionar imagen (JPG):</td><td><input
type='file' name='Archivo' readonly>");
            SALIDA.println("<input type='submit' name='ADBOTON'
value='Aceptar'></td></tr>");
            SALIDA.println("<input type=hidden name=USUARIO value='"+Usuario+">");
            SALIDA.println("</form>");
        }
        else
        {
            if(insertar.equals("T"))
            {
                imagen_unidad=imagen_unidad+".jpg";
                String insert="insert into UNIDADES_MEDIDA
values('+nombre_unidad+', '"+imagen_unidad+')";
                STMT1.executeUpdate(insert);
                SALIDA.println("<table class=colortable2 border=1 align=center><tr><td colspan=3
align=center class=hcolor><font face=verdana size=3><b>El Archivo ha sido cargado
exitosamente.</td></tr>");
                SALIDA.println("<tr><td colspan=3 align=center ><input type=button value=Aceptar
OnClick='Cerrar()'></td></tr></table>");
            }
            else
            {
                SALIDA.println("<table class=colortable2 border=1 align=center><tr><td colspan=3
align=center class=hcolor><font face=verdana size=3><b>La imagen ha sido modificada
exitosamente</td></tr>");
                SALIDA.println("<tr><td colspan=3 align=center ><input type=button value=Aceptar
OnClick='Cerrar()'></td></tr></table>");
                File f=new File("C:\\HTML\\images\\iconos\\"+imagen_unidad);
                f.delete();
            }
        }
    }
}

```

```

    }

    /**
    // Lectura y Copia del Archivo
    /**
    // Get the content type
    String contentType = SOLICITUD.getContentType();

    // Set a default for the boundary string length
    // Find out where the boundary string starts
    int boundaryIndex = contentType.indexOf("boundary=");

    // Get the boundary string
    String boundary = contentType.substring(boundaryIndex+9);

    int boundaryStrLength = boundary.length();

    // Get the input stream to read the uploaded file
    ServletInputStream servIn = SOLICITUD.getInputStream();

    DataInputStream in = new DataInputStream(servIn);
    if(in==null)
        SALIDA.print("No hay datos que mostrar");
    // Lop off the headers from the content (read until you get a blank line)
    String line;

    while ((line = in.readLine()) != null)
    {
        getServletContext().log("Got line: "+line);
        if (line.trim().length() == 0) break;
    }

    ByteArrayOutputStream byteOut = new ByteArrayOutputStream(
    SOLICITUD.getContentLength());

    byte[] buffer = new byte[4096];
    int len;

    // Copy the uploaded file to a byte array
    while ((len = in.read(buffer)) > 0)
    {
        byteOut.write(buffer, 0, len);
    }

    byte[] outBytes = byteOut.toByteArray();

    FileOutputStream fileOut = new
    FileOutputStream("C:\\HTML\\images\\iconos\\"+imagen_unidad);

    // Write the byte array out the the file, trim off the boundary plus some
    // padding surrounding the boundary
    fileOut.write(outBytes, 0, outBytes.length - boundaryStrLength - 8);
    fileOut.close();

    String clientHost = SOLICITUD.getRemoteHost();

```

```

        }//fin del if

        SALIDA.println("</tr></table><br><br>");

    }//fin del try

    catch (Exception ex)
    {
        SALIDA.println("Exception 1!");
        ex.printStackTrace(SALIDA);
    }
    finally
    {
        try
        {
            /***/
            /* Cerrando los elementos de la conexion */
            /***/
            if (RS1 != null)
            {
                RS1.close();
            }

            if (STMT1 != null)
            {
                STMT1.close();
            }

            if (CON1 != null)
            {
                CON1.close();
            }

        }
        catch (Exception ex)
        {
            SALIDA.println("ERROR DE CIERRE DE ELEMENTOS DE CONEXION !!!");
            ex.printStackTrace(SALIDA);
        }
    }//FIN DEL FINALLY
}

public void doPost(HttpServletRequest SOLICITUD,
    HttpServletResponse RESPUESTA)
    throws ServletException, java.io.IOException
{
    doGet(SOLICITUD,RESPUESTA);
}
}

```

## Pantalla Listado Unidades de Medida (Servlet Adjuntar)

Servlet encargado de generar el listado de las unidades de medida registradas.

```

/*****/
// Servlet Utilizado para Adjuntar Archivos a un Documento
/*****/

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.io.*;

public class Adjuntar extends HttpServlet
{
    /*****/
    /*          funcion doGet          */
    /*****/
    public void doGet(HttpServletRequest SOLICITUD,
        HttpServletResponse RESPUESTA)
        throws ServletException, java.io.IOException
    {

        //AQUI SE DECLARA QUE LA SALIDA SERA UN HTML
        RESPUESTA.setContentType("text/html");
        PrintWriter SALIDA = new PrintWriter(RESPUESTA.getOutputStream());

        /*****/
        /* Impresion del Encabezado de la Pantalla */
        /*****/
        SALIDA.println("<html>");
        SALIDA.println("<head>");
        SALIDA.println(" <title>Adjuntar fotografías al proyecto</title>");

        SALIDA.println("<LINK href='\"/HTML/include/ourstyle.css\"' type=text/css rel=stylesheet>");
        SALIDA.println("<SCRIPT language=JavaScript src='\"/HTML/include/ourScript.js'
type=text/javascript></SCRIPT>");

        SALIDA.println("</head>");

        PRINCIPAL(SALIDA,SOLICITUD,RESPUESTA);

        /*****/
        /**** Cerrando la conexiona al HTML ****/
        /*****/
        SALIDA.println("</body>");
        SALIDA.println("</html>");
    }
}

```

```

SALIDA.flush();
SALIDA.close();

} //***** FIN DE LA FUNCION doGet *****

/*****
/* Este metodo despliega una pantalla de captura */
/* adecuado a la tabla que se esta procesando */
*****/
private void PRINCIPAL(java.io.PrintWriter SALIDA,
                      HttpServletRequest SOLICITUD,HttpServletResponse RESPUESTA)

{
/*****
/* Creando las propiedades necesarias para conectar con la Base de Datos */
*****/
    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

String RESPONSABLE= SOLICITUD.getParameter("USUARIO");

    Menu DoMenu = new Menu();

    DoMenu.CREARMENU(RESPONSABLE,SALIDA,Driver,Usr,DB,Pass);

    DoMenu.Encabezado(SALIDA);

/*****
/* Inicializando a null elementos de conexion */
*****/
    Connection CON1 = null;
    Statement STMT1 = null;
    ResultSet RS1 = null;

/*****
/* LECTURA DE PARAMETROS */
*****/

try
{
/*****
/* Creando elementos necesarios para establecer y */
/* utilizar la conexion */
*****/
    Class.forName(Driver).newInstance();
    CON1 = DriverManager.getConnection(DB,Usr,Pass);
    STMT1 = CON1.createStatement();

```





## Registro de necesidades

Para la captura de necesidades se han desarrollado tres servlets:

- ListadoNecesidades: Extrae de la base de datos, y de acuerdo al usuario, las necesidades registradas en la base de datos de la obra que ha sido seleccionada. Cada registro posee un enlace que al ser activado ejecutará el servlet CapturaNecesidades.
- CapturaNecesidades: Extrae de la base de datos todos los campos correspondientes al registro seleccionado, ubicandolos en un formulario para su modificación. En la parte inferior del formulario se encuentran los botones adicionar o modificar y eliminar, los cuales al ser activados ejecutan el servlet ActualizacionNecesidades.
- ActualizacionNecesidades: Según la acción seleccionada (adición, modificación o eliminación) se realizan los cambios a la base de datos.

### *Servlet ListadoNecesidades*

```
import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class ListadoNecesidades extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
        throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Necesidades</title>");
        Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
        Salida.println("<SCRIPT language=JavaScript src=\"/HTML/include/ourScript.js\"
type=text/javascript></SCRIPT>");
        Salida.println("</head>");
        Salida.println("<body>");

        String codigo_obra=Requerimiento.getParameter("codigo_obra");
        String usuario=Requerimiento.getParameter("USUARIO");
```

```

// Ejecutando Proceso Principal
Proceso(Salida,codigo_obra,usuario);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

private void Proceso(PrintWriter Salida,String codigo_obra,String usuario)
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();

//Seleccionando registros de necesidades de acuerdo al usuario y obra seleccionada en el mapa

    String Consulta="select codigo_necesidad,monto_necesidad from necesidades where
codigo_obra="+codigo_obra+" and codigo_responsable="+usuario+"";
    String codigo="";
    String monto="";

//Generando formulario para añadir nueva necesidad
Salida.println("<form name=form1 method=get
action=/servlet/CapturaNecesidades?codigo_obra="+codigo_obra+"&USUARIO="+usuario+"&adicionar
=T' onsubmit='return NoNulos(this)'>");
Salida.println("<table border=1 cellpadding=1 cellspacing=3 class=colortable align=center width=90%>");
Salida.println("<tr>");
Salida.println("<td class=hcolor>Código Necesidad:</td><td><input type=text name=codigo_necesidad
size=10 onkeypress='Numeros(this)' onblur='Enteros(this)'></td><td><input type=submit
value='Adicionar Necesidad'></td>");
Salida.println("<td><input type=hidden name='USUARIO' value='"+usuario+"' ></td>");
Salida.println("<td><input type=hidden name='adicionar' value='T' ></td>");
Salida.println("<td><input type=hidden name='codigo_obra' value='"+codigo_obra+"' ></td>");
Salida.println("</tr>");

```

```

Salida.println("</table>");

Salida.println("<br>");

//Generando listado de registros encontrados coincidentes con el usuario activo y la obra seleccionada.
Salida.println(" <table border=1 cellpadding=1 cellspacing=3 class=colortable align=center
width=90%>");
Salida.println("          <tr>");
Salida.println("                <td width=25% align=center class=hcolor>Necesidad</td>");
Salida.println("                <td width=70% align=center class=hcolor>Monto</td>");
Salida.println("          </tr>");

ResultadoSQL=InstruccionSQL.executeQuery(Consulta);
while(ResultadoSQL.next())
{
codigo=ResultadoSQL.getString(1);
monto=ResultadoSQL.getString(2);

Salida.println("          <tr>");
Salida.println("                <td class=tdfont><a
href=/servlet/CapturaNecesidades?codigo_obra="+codigo_obra+"&codigo_necesidad="+codigo+"&USU
ARIO="+usuario+"&adicionar=F'>"+codigo+"</a></td>");
Salida.println("                <td class=tdfontletra><font size=2>"+monto+"</td>");
Salida.println("          </tr>");
}

Salida.println(" </table>");

}
catch (Exception Error)
{
Salida.println("Situación Anormal!");
Error.printStackTrace(Salida);
}
finally
{
try
{
if (ResultadoSQL!=null) ResultadoSQL.close();
if (InstruccionSQL!=null) InstruccionSQL.close();
if (ConexionSQL!=null) ConexionSQL.close();
}
catch (Exception Error) {}
}
}
}

```

### *Servlet CapturaNecesidades*

```

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

```

```

public class CapturaNecesidades extends HttpServlet
{
public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
throws ServletException, IOException
{

// Abriendo Archivo de Salida
Respuesta.setContentType("text/html");
PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

// Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
Salida.println("<html>");
Salida.println("<head>");
Salida.println("<title>Necesidades</title>");
Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
Salida.println("<SCRIPT language=JavaScript src=\"/HTML/include/ourScript.js\"
type=text/javascript></SCRIPT>");
Salida.println("</head>");
Salida.println("<body>");

String codigo_obra=Requerimiento.getParameter("codigo_obra");
String codigo_necesidad=Requerimiento.getParameter("codigo_necesidad");
String usuario=Requerimiento.getParameter("USUARIO");
String adicionar=Requerimiento.getParameter("adicionar");

// Ejecutando Proceso Principal
Proceso(Salida,codigo_necesidad,codigo_obra,usuario,adicionar);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

private void Proceso(PrintWriter Salida,String codigo_necesidad, String codigo_obra,String usuario,String
adicionar)
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

// Iniciando Conexión

```

```

Usuario MIUSUARIO = new Usuario();
String Driver = MIUSUARIO.driver;
String DB = MIUSUARIO.myBD;
String Usr = MIUSUARIO.usuario;
String Pass= MIUSUARIO.password;
String dirhtml= MIUSUARIO.DirHTML;

ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
InstruccionSQL=ConexionSQL.createStatement();
String consulta=" ";

String nombre_necesidad;
String descripcion_necesidad;
String monto_necesidad;
String fecha_ingreso;

// Verificando que el usuario no desee adicionar una necesidad con un código ya existente o perteneciente a
//otro usuario.

ResultadoSQL=InstruccionSQL.executeQuery("select count(*) from necesidades where
codigo_necesidad="+codigo_necesidad+" and codigo_obra="+codigo_obra);
ResultadoSQL.next();
int Existe=ResultadoSQL.getInt(1);

//Si el código ya existe y pertenece al usuario se selecciona el registro y se muestra para poder ser
//modificado, si existe y no pertenece al usuario, se le asigna un código automáticamente.

if((Existe!=0)&&(adicionar.equals("T")))
{
ResultadoSQL=InstruccionSQL.executeQuery("select codigo_responsable from necesidades where
codigo_necesidad="+codigo_necesidad+" and codigo_obra="+codigo_obra);
ResultadoSQL.next();
String res=ResultadoSQL.getString(1);
    if(res.trim().equals(usuario.trim())) adicionar="F";
    else
    {
ResultadoSQL=InstruccionSQL.executeQuery("select max(codigo_necesidad)+1 from
necesidades where codigo_obra="+codigo_obra);
ResultadoSQL.next();
codigo_necesidad=ResultadoSQL.getString(1);
    }

}

//Si se esta adicionando registro, los campos aparecen vacíos.

if(adicionar.equals("T"))
{
    nombre_necesidad="";
    descripcion_necesidad="";
    monto_necesidad="";
ResultadoSQL=InstruccionSQL.executeQuery("select convert(char(10),getdate(),103)");
ResultadoSQL.next();
}

```

```

        fecha_ingreso=ResultadoSQL.getString(1);
    }
    //Si es un registro que se desea modificar, se extraen los valores de la base de datos.

    else
    {
        consulta= "select
nombre_necesidad,descripcion_necesidad,monto_necesidad,convert(char(10),fecha_ingreso,103)," +
                "codigo_responsable from NECESIDADES where
codigo_obra=" +codigo_obra+" and codigo_necesidad=" +codigo_necesidad;
        ResultadoSQL=InstruccionSQL.executeQuery(consulta);
        ResultadoSQL.next();
        nombre_necesidad=ResultadoSQL.getString(1);
        descripcion_necesidad=ResultadoSQL.getString(2);
        monto_necesidad=ResultadoSQL.getString(3);
        fecha_ingreso=ResultadoSQL.getString(4);
    }

    consulta="select nombre_obra from obras_fisicas where codigo_obra=" +codigo_obra+"";
    ResultadoSQL=InstruccionSQL.executeQuery(consulta);
    ResultadoSQL.next();
    String nombre_obra=ResultadoSQL.getString(1);

    Salida.println("<form name=form1 method=get action='/servlet/ActualizacionNecesidades?'
onsubmit='return NoNulos(this);'>");
    Salida.println(" <table border=1 cellpadding=1 cellspacing=3 class=colortable align=center>");
    Salida.println(" <tr><td colspan=2 class=hcolor align=center>CAPTURA DE
NECESIDADES</td></tr>");
    Salida.println(" <tr>");
    Salida.println(" <td class=tdfont>Obra:</td>");
    Salida.println(" <td>" +codigo_obra+" - " +nombre_obra+"</td>");
    Salida.println(" </tr>");
    Salida.println(" <tr>");
    Salida.println(" <td class=tdfont>Código:</td>");
    Salida.println(" <td><input type=text name=CODIGO_NECESIDAD size=5
value=" +codigo_necesidad+" readonly></td>");
    Salida.println(" </tr>");
    Salida.println(" <tr>");
    Salida.println(" <td class=tdfont>Nombre:</td>");
    Salida.println(" <td><input type=text size=20 name=NOMBRE_NECESIDAD
value=" +nombre_necesidad+"></td>");
    Salida.println(" </tr>");
    Salida.println(" <tr>");
    Salida.println(" <td class=tdfont>Descripción:</td>");
    Salida.println(" <td><textarea rows=5 cols=50
name=DESCRIPCION_NECESIDAD>" +descripcion_necesidad+" </textarea></td>");
    Salida.println(" </tr>");
    Salida.println(" <tr>");
    Salida.println(" <td class=tdfont>Monto:</td>");
    Salida.println(" <td><input type=text name=MONTO_NECESIDAD size=10
value=" +monto_necesidad+" onblur='Decimales(this)'></td>");
    Salida.println(" </tr>");
    Salida.println(" <tr>");
    Salida.println(" <td class=tdfont>Fecha Ingreso:</td>");

```

```

Salida.println("                <td><input type=text name=FECHA_INGRESO size=10
value="" +fecha_ingreso+"" readonly ></td>");
Salida.println("                </tr>");
Salida.println("                <tr>");
Salida.println("                <td class=tdfont>Responsable:</td>");

consulta="select nombre_responsable+' '+apellido_responsable from RESPONSABLES where
codigo_Responsable="" +usuario+"";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String Responsable=ResultadoSQL.getString(1);

Salida.println("                <td><input type=text name=Responsable size=30
value="" +Responsable+"" readonly></td>");
Salida.println("                </tr>");
Salida.println("                <tr>");

//Se proporcionan los botones Adicionar o Modificar.

if (adicionar.equals("T"))
Salida.println("                <td class=trbotones colspan=2 align=center><input type=submit name=guardar
value=Guardar></td>");
else
Salida.println("                <td class=trbotones colspan=2 align=center><input type=submit
name=actualizar value=Actualizar></td>");

Salida.println("                </tr>");
Salida.println("                <input type=hidden name=adicionar value="+adicionar+">");
Salida.println("                <input type=hidden name=USUARIO value="+usuario+">");
Salida.println("                <input type=hidden name=CODIGO_OBRA value="+codigo_obra+">");
Salida.println("        </table>");
Salida.println("    </form>");

}
catch (Exception Error)
{
    Salida.println("Situación Anormal!");
    Error.printStackTrace(Salida);
}
finally
{
    try
    {
        if (ResultadoSQL!=null) ResultadoSQL.close();
        if (InstruccionSQL!=null) InstruccionSQL.close();
        if (ConexionSQL!=null) ConexionSQL.close();
    }
    catch (Exception Error) {}
}
}
}

```

## *Servlet ActualizacionNecesidades*

```
import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class ActualizacionNecesidades extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Necesidades</title>");
        Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
        Salida.println("</head>");
        Salida.println("<body>");

        //Se verifica la acción que el usuario esta realizando
        String adicionar=Requerimiento.getParameter("adicionar");
        String codigo_obra=Requerimiento.getParameter("CODIGO_OBRA");
        String codigo_necesidad=Requerimiento.getParameter("CODIGO_NECESIDAD");
        String nombre_necesidad=Requerimiento.getParameter("NOMBRE_NECESIDAD");
        String descripcion_necesidad=Requerimiento.getParameter("DESCRIPCION_NECESIDAD");
        String monto_necesidad=Requerimiento.getParameter("MONTO_NECESIDAD");
        String fecha_ingreso=Requerimiento.getParameter("FECHA_INGRESO");
        String usuario=Requerimiento.getParameter("USUARIO");

        // Ejecutando Proceso Principal

        Proceso(Salida,adicionar,codigo_obra,codigo_necesidad,nombre_necesidad,descripcion_necesidad,monto_necesidad,fecha_ingreso,usuario);

        // Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
        Salida.println("</body>");
        Salida.println("</html>");
        Salida.flush();
        Salida.close();
    }
}
```

```

private void Proceso(PrintWriter Salida,String adicionar,String codigo_obra,String
codigo_necesidad,String nombre_necesidad,String descripcion_necesidad,String monto_necesidad,String
fecha_ingreso,String usuario)
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();

    String accion=" ";

//Se construye la sentencia a ejecutar según sea la acción a realizar (insert, update)
if(adicionar.equals("T"))
accion="set dateformat dmy insert into NECESIDADES
values(""+codigo_obra+"",""+codigo_necesidad+"",""+nombre_necesidad+"",""+descripcion_necesidad+"",""+
monto_necesidad+"",""+fecha_ingreso+"",""+usuario+"");
else
accion="set dateformat dmy update NECESIDADES set nombre_necesidad="" +nombre_necesidad+",
descripcion_necesidad="" +descripcion_necesidad+", monto_necesidad="" +monto_necesidad+" where
codigo_obra="" +codigo_obra+" and codigo_necesidad="" +codigo_necesidad;

InstruccionSQL.executeUpdate(accion);

String consulta="select nombre_obra from obras_fisicas where codigo_obra="" +codigo_obra+""";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_obra=ResultadoSQL.getString(1);

//Se muestran los nuevos datos en la base de datos
Salida.println("<table border=1 class=colortable2 align=center width=90%>");
Salida.println("<tr>");
Salida.println("<td class=hcolor>Obra:</td>");
Salida.println("<td class=tdfontletra>"+codigo_obra+" - "+nombre_obra+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=hcolor>Necesidad:</td>");

```

```

Salida.println("<td class=tdfontletra>"+codigo_necesidad+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=hcolor>Descripción:</td>");
Salida.println("<td class=tdfontletra>"+descripcion_necesidad+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=hcolor>Monto:</td>");
Salida.println("<td class=tdfontletra>"+monto_necesidad+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=hcolor>Fecha Ingreso:</td>");
Salida.println("<td class=tdfontletra>"+fecha_ingreso+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");

//Se envia un mensaje de confirmación de acción

if (adicionar.equals("T"))
Salida.println("<td colspan=2 class=trbotones align=center>El Registro ha sido Agregado</td>");
else
Salida.println("<td colspan=2 class=trbotones align=center>El Registro ha sido Modificado</td>");
Salida.println("</tr>");
Salida.println("</table>");

Salida.println("<br><br>");

Salida.println("<a
href=/servlet/ListadoNecesidades?codigo_obra="+codigo_obra+"&USUARIO="+usuario+">Regresar a
Listado de Necesidades</a><br>");

}
catch (Exception Error)
{
Salida.println("Situación Anormal!");
Error.printStackTrace(Salida);
}
finally
{
try
{
if (ResultadoSQL!=null) ResultadoSQL.close();
if (InstruccionSQL!=null) InstruccionSQL.close();
if (ConexionSQL!=null) ConexionSQL.close();
}
catch (Exception Error) {}
}
}
}

```

## Registro de Proyectos

Para el registro de proyectos se han creado tres servlets:

- ListadoProyectos: Dependiendo del usuario y de la obra seleccionada realiza un filtro de los proyectos registrados en la base de datos y los muestra al usuario, con la opción de agregar un nuevo registro. Cada registro mostrado en el listado posee un enlace, el cual al ser activado ejecuta el servlet CapturaProyectos.
- CapturaProyectos: Según el proyecto seleccionado por el usuario, se extraen los valores de los campos del registro seleccionado, y se presentan al usuario en un formulario para su llenado o modificación (según el caso adición o modificación). Al final del formulario se presentan los botones que realizan la acción solicitada por el usuario, estos a su vez ejecutan el servlet ActualizacionProyectos.
- Actualización Proyectos: Modifica el registro en la base de datos o lo adiciona según sea el caso.

### *ListadoProyectos*

```
import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class ListadoProyectos extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Proyectos</title>");
        Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
        Salida.println("<SCRIPT language=JavaScript src=\"/HTML/include/ourScript.js\"
type=text/javascript></SCRIPT>");
        Salida.println("</head>");
        Salida.println("<body>");
```

```

String codigo_obra=Requerimiento.getParameter("codigo_obra");
String usuario=Requerimiento.getParameter("USUARIO");

// Ejecutando Proceso Principal
Proceso(Salida,codigo_obra,usuario);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

private void Proceso(PrintWriter Salida,String codigo_obra,String usuario)
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();

//Se consulta la base de datos en búsqueda de registros
    String Consulta="select codigo_proyecto,nombre_proyecto from proyectos where
codigo_obra="+codigo_obra+" and codigo_responsable="+usuario+"";
    String codigo="";
    String proyecto="";

//Formulario para solicitar ingreso de nuevo registro
Salida.println("<form name=form1 method=get
action=/servlet/CapturaProyectos?codigo_obra="+codigo_obra+"&USUARIO="+usuario+"&adicionar=T'
onsubmit='return NoNulos(this)'>");
Salida.println("<table border=1 cellpadding=1 cellspacing=3 class=colortable align=center width=90%>");
Salida.println("<tr>");
Salida.println("<td class=hcolor>Proyecto:</td><td><input type=text name=codigo_proyecto size=10
onblur='Enteros(this)' onkeypress='Numeros(this)'></td><td><input type=submit value='Adicionar
Proyecto' ></td>");
Salida.println("<td><input type=hidden name='USUARIO' value='"+usuario+"' ></td>");
Salida.println("<td><input type=hidden name='adicionar' value='T' ></td>");
Salida.println("<td><input type=hidden name='codigo_obra' value='"+codigo_obra+"' ></td>");
Salida.println("</form>");

```

```

Salida.println("</tr>");
Salida.println("</table>");

//Listado con registros existentes

Salida.println("<br>");
Salida.println(" <table border=1 cellpadding=1 cellspacing=3 class=colortable align=center
width=90%>");
Salida.println("          <tr>");
Salida.println("                <td width=25% align=center class=hcolor>Código</td>");
Salida.println("                <td width=70% align=center class=hcolor>Proyecto</td>");
Salida.println("          </tr>");

ResultadoSQL=InstruccionSQL.executeQuery(Consulta);
while(ResultadoSQL.next())
{
codigo=ResultadoSQL.getString(1);
proyecto=ResultadoSQL.getString(2);

Salida.println("          <tr>");
Salida.println("                <td class=tdfont><a
href='/servlet/CapturaProyectos?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo+"&USUARI
O="+usuario+"&adicionar=F'+>"+codigo+"</a></td>");
Salida.println("                <td><font size=2>"+proyecto+"</td>");
Salida.println("          </tr>");
}

Salida.println(" </table>");

}
catch (Exception Error)
{
Salida.println("Situación Anormal!");
Error.printStackTrace(Salida);
}
finally
{
try
{
if (ResultadoSQL!=null) ResultadoSQL.close();
if (InstruccionSQL!=null) InstruccionSQL.close();
if (ConexionSQL!=null) ConexionSQL.close();
}
catch (Exception Error) {}
}
}
}

```

### *CapturaProyectos*

```

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;

```

```

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

public class CapturaProyectos extends HttpServlet
{
public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
throws ServletException, IOException
{

// Abriendo Archivo de Salida
Respuesta.setContentType("text/html");
PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

// Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
Salida.println("<html>");
Salida.println("<head>");
Salida.println("<title>Proyectos</title>");
Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
Salida.println("<SCRIPT language=JavaScript src=\"/HTML/include/ourScript.js\"
type=text/javascript></SCRIPT>");

Salida.println("</head>");
Salida.println("<body>");

String codigo_obra=Requerimiento.getParameter("codigo_obra");
String codigo_proyecto=Requerimiento.getParameter("codigo_proyecto");
String USUARIO=Requerimiento.getParameter("USUARIO");
String adicionar=Requerimiento.getParameter("adicionar");

// Ejecutando Proceso Principal
Proceso(Salida,codigo_proyecto,codigo_obra,USUARIO,adicionar);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

private void Proceso(PrintWriter Salida,String codigo_proyecto, String codigo_obra,String usuario,String
adicionar)
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

```

```

try
{

// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();
    String nombre_proyecto;
    String descripcion_proyecto;
    String fecha_inicio_programada;
    String fecha_final_programada;
    String fecha_inicio_real;
    String fecha_final_real;
    String codigo_etapa;
    String codigo_estado;
    String empleos_proyecto;
    String beneficiarios_proyecto;
    String codigo_institucion;
    String codigo_tipo;
    String codigo_necesidad;
    String consulta;

//Verificación de existencia del registro
ResultadoSQL=InstruccionSQL.executeQuery("select count(*) from proyectos where
codigo_proyecto="+codigo_proyecto+" and codigo_obra="+codigo_obra);
ResultadoSQL.next();
int Existe=ResultadoSQL.getInt(1);

//Verificación de existencia del registro, si el registro existe y le pertenece al usuario se presenta en formato
//para su modificación, si existe y no pertenece al usuario, se le asigna un nuevo código de proyecto.
if((Existe!=0)&&(adicionar.equals("T")))
{
ResultadoSQL=InstruccionSQL.executeQuery("select codigo_responsable from proyectos where
codigo_proyecto="+codigo_proyecto+" and codigo_obra="+codigo_obra);
ResultadoSQL.next();
String res=ResultadoSQL.getString(1);
    if(res.trim().equals(usuario.trim())) adicionar="F";
    else
    {
        ResultadoSQL=InstruccionSQL.executeQuery("select max(codigo_proyecto)+1 from proyectos
where codigo_obra="+codigo_obra);
        ResultadoSQL.next();
        codigo_proyecto=ResultadoSQL.getString(1);
    }
}

if(adicionar.equals("T"))

```

```

{
    nombre_proyecto="";
    descripcion_proyecto="";
    fecha_inicio_programada="";
    fecha_final_programada="";
    fecha_inicio_real="";
    fecha_final_real="";
    codigo_etapa="";
    codigo_estado="";
    empleos_proyecto="";
    beneficiarios_proyecto="";
    codigo_institucion="";
    codigo_tipo="";
    codigo_necesidad="";
}
//se extraen los valores de los campos del registro seleccionado.
else
{
    consulta= "select
nombre_proyecto,descripcion_proyecto,convert(char(10),fecha_inicio_programada,103),convert(char(10),f
echa_final_programada,103),"
        "convert(char(10),fecha_inicio_real,103),convert(char(10),fecha_final_real,103),codigo_etapa,cod
igo_estado,empleos_proyecto,"
            "beneficiarios_proyecto,codigo_tipo,codigo_necesidad"+
            " from proyectos where codigo_obra="+codigo_obra+" and
codigo_proyecto="+codigo_proyecto;
    ResultadoSQL=InstruccionSQL.executeQuery(consulta);
    ResultadoSQL.next();
    nombre_proyecto=ResultadoSQL.getString(1);
    descripcion_proyecto=ResultadoSQL.getString(2);
    fecha_inicio_programada=ResultadoSQL.getString(3);
    fecha_final_programada=ResultadoSQL.getString(4);
    fecha_inicio_real=ResultadoSQL.getString(5);
    fecha_final_real=ResultadoSQL.getString(6);
    codigo_etapa=ResultadoSQL.getString(7);
    codigo_estado=ResultadoSQL.getString(8);
    empleos_proyecto=ResultadoSQL.getString(9);
    beneficiarios_proyecto=ResultadoSQL.getString(10);
    codigo_tipo=ResultadoSQL.getString(11);
    codigo_necesidad=ResultadoSQL.getString(12);
}

consulta="select nombre_obra from obras_fisicas where codigo_obra="+codigo_obra+"";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_obra=ResultadoSQL.getString(1);

Salida.println("<form name=form1 method=post action=/servlet/ActualizacionProyectos?
onsubmit=\<return NoNulos(this);\>");
Salida.println(" <table border=1 cellpadding=1 cellspacing=3 class=colortable align=center>");

```

```

Salida.println("          <tr><td colspan=2 align=center class=hcolor>CAPTURA DE
PROYECTOS</td></tr>");
Salida.println("          <tr>");
Salida.println("              <td class=tdfont>Obra:</td>");
Salida.println("              <td><font size=2>"+codigo_obra+" - "+nombre_obra+"</td>");
Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("              <td class=tdfont>Código:</td>");

Salida.println("<td><input type=text name=CODIGO_PROYECTO size=5 value="+codigo_proyecto+"
readonly></td>");

Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("              <td class=tdfont>Tipo de Proyecto:</td>");
Salida.println("              <td><select name=CODIGO_TIPO size=1>");

String codigo_tipo2;
String nombre_tipo;

ResultadoSQL=InstruccionSQL.executeQuery("select codigo_tipo,nombre_tipo from tipos_proyecto");
while(ResultadoSQL.next())
{
codigo_tipo2=ResultadoSQL.getString(1);
nombre_tipo=ResultadoSQL.getString(2);

if (codigo_tipo.equals(codigo_tipo2))
Salida.println("<option selected value="+codigo_tipo2+">"+nombre_tipo+"</option>");
else
Salida.println("<option value="+codigo_tipo2+">"+nombre_tipo+"</option>");
}

Salida.println("</select></td>");

Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("              <td class=tdfont>Nombre:</td>");
Salida.println("              <td><textarea rows=3 cols=40
name=NOMBRE_PROYECTO>"+nombre_proyecto+"</textarea></td>");
Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("              <td class=tdfont>Descripción:</td>");
Salida.println("              <td><textarea rows=3 cols=40
name=DESCRIPCION_PROYECTO>"+descripcion_proyecto+"</textarea></td>");
Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("              <td class=tdfont>Fecha Inicio Programada:</td>");
Salida.println("              <td><input type=text name=FECHA_INICIO_PROGRAMADA
size=10 value="+fecha_inicio_programada+" onblur=ValidFecha(this)'></td>");
Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("              <td class=tdfont>Fecha Final Programada:</td>");
Salida.println("              <td><input type=text name=FECHA_FINAL_PROGRAMADA
size=10 value="+fecha_final_programada+" onblur=ValidFecha(this)'></td>");
Salida.println("          </tr>");

```

```

Salida.println("          <tr>");
Salida.println("          <td class=tdfont>Fecha Inicio Real:</td>");
Salida.println("          <td><input type=text name=FECHA_INICIO_REAL size=10
value="+fecha_inicio_real+" onblur='ValidFecha(this)'></td>");
Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("          <td class=tdfont>Fecha Final Real:</td>");
Salida.println("          <td><input type=text name=FECHA_FINAL_REAL size=10
value="+fecha_final_real+" onblur='ValidFecha(this)'></td>");
Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("          <td class=tdfont>Etapa:</td>");
Salida.println("          <td><select name=CODIGO_ETAPA size=1>");

```

```

String codigo_etapa2;
String nombre_etapa;

```

```

ResultadoSQL=InstruccionSQL.executeQuery("select codigo_etapa,nombre_etapa from etapas");
while(ResultadoSQL.next())
{
codigo_etapa2=ResultadoSQL.getString(1);
nombre_etapa=ResultadoSQL.getString(2);

if (codigo_etapa.equals(codigo_etapa2))
Salida.println("<option selected value="+codigo_etapa2+">" + nombre_etapa + "</option>");
else
Salida.println("<option value="+codigo_etapa2+">" + nombre_etapa + "</option>");
}

Salida.println("</select></td>");

```

```

Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("          <td class=tdfont>Estado:</td>");
Salida.println("          <td><select name=CODIGO_ESTADO size=1>");

```

```

String codigo_estado2;
String nombre_estado;

```

```

ResultadoSQL=InstruccionSQL.executeQuery("select codigo_estado,nombre_estado from estados");
while(ResultadoSQL.next())
{
codigo_estado2=ResultadoSQL.getString(1);
nombre_estado=ResultadoSQL.getString(2);

if (codigo_estado.equals(codigo_estado2))
Salida.println("<option selected value="+codigo_estado2+">" + nombre_estado + "</option>");
else
Salida.println("<option value="+codigo_estado2+">" + nombre_estado + "</option>");
}

Salida.println("</select></td>");

```

```

Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("                <td class=tdfont>Empleos Generados:</td>");
Salida.println("                <td><input type=text name=EMPLEOS_PROYECTO size=5
value="" +empleos_proyecto+" onblur='Enteros(this)' onkeypress='Numeros(this)'></td>");
Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("                <td class=tdfont>Beneficiarios:</td>");
Salida.println("                <td><input type=text name=BENEFICIARIOS_PROYECTO size=5
value="" +beneficiarios_proyecto+" onblur='Enteros(this)' onkeypress='Numeros(this)'></td>");
Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("                <td class=tdfont>Necesidad:</td>");
Salida.println("                <td><select name=CODIGO_NECESIDAD size=1>");

```

```

String codigo_necesidad2;
String nombre_necesidad;

```

```

ResultadoSQL=InstruccionSQL.executeQuery("select codigo_necesidad,nombre_necesidad from
necesidades where codigo_obra="+codigo_obra);
while(ResultadoSQL.next())
{
codigo_necesidad2=ResultadoSQL.getString(1);
nombre_necesidad=ResultadoSQL.getString(2);

```

```

if (codigo_necesidad.equals(codigo_necesidad2))
Salida.println("<option selected value="+codigo_necesidad2+">" +nombre_necesidad+"</option>");
else
Salida.println("<option value="+codigo_necesidad2+">" +nombre_necesidad+"</option>");
}

```

```

Salida.println("</select></td>");

```

```

Salida.println("          </tr>");

```

```

Salida.println("          <tr>");
Salida.println("                <td class=tdfont>Responsable:</td>");

```

```

consulta="select nombre_responsable+' '+apellido_responsable from RESPONSABLES where
codigo_Responsable="" +usuario+""";

```

```

ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String Responsable=ResultadoSQL.getString(1);

```

```

Salida.println("                <td><input type=text name=Responsable size=30
value="" +Responsable+" readonly></td>");
Salida.println("          </tr>");

```

```

Salida.println("          <tr>");

```

```

if (adicionar.equals("T"))
//Salida.println("                <td class=trbotones colspan=2 align=center><input type=image name=guardar
src=/HTML/images/Botones/Boton_guardar.gif

```

```

OnMouseOver="\src=/HTML/images/Botones/Boton_guardar_off.gif\"
OnMouseOut="\src=/HTML/images/Botones/Boton_guardar.gif\"></td>");
Salida.println("      <td class=trbotones colspan=2 align=center><input type=submit name=guardar
value=Guardar></td>");
else
//Salida.println("      <td class=trbotones colspan=2 align=center><input type=image name=actualizar
src=/HTML/images/Botones/Boton_actualizar.gif
OnMouseOver="\src=/HTML/images/Botones/Boton_actualizar_off.gif\"
OnMouseOut="\src=/HTML/images/Botones/Boton_actualizar.gif\"></td>");
Salida.println("      <td class=trbotones colspan=2 align=center><input type=submit
name=actualizar value=Actualizar></td>");

Salida.println("      </tr>");
Salida.println("    </table>");
Salida.println("      <input type=hidden name=USUARIO value="+usuario+">");
Salida.println("      <input type=hidden name=CODIGO_OBRA value="+codigo_obra+">");
Salida.println("      <input type=hidden name=adicionar value="+adicionar+">");
Salida.println("</form>");

```

```
Salida.println("<br><br><br>");
```

```

Salida.println("<a
href=/servlet/ListadoProyectos?codigo_obra="+codigo_obra+"&USUARIO="+usuario+">Regresar a
Listado de Proyectos</a><br>");
Salida.println("<a
href=/servlet/ListadoFuentes?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&U
SUARIO="+usuario+">Ver Listado de Fuentes de Financiamiento</a><br>");
Salida.println("<a
href=/servlet/SeleccionPeriodo?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&
USUARIO="+usuario+">Selección de período para seguimiento</a><br>");

```

```
Salida.println("<br>");
```

```

if (!(adicionar.equals("T")))
{
//IR A SEGUIMIENTO
consulta="select count(*) from SEGUIMIENTO where codigo_obra="+codigo_obra+" and
codigo_proyecto="+codigo_proyecto;
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
int numero=ResultadoSQL.getInt(1);
    if(numero!=0)
    {
        String año;
        consulta="select distinct año from SEGUIMIENTO where codigo_obra="+codigo_obra+" and
codigo_proyecto="+codigo_proyecto;
        ResultadoSQL=InstruccionSQL.executeQuery(consulta);
        Salida.println("<form action=/servlet/CapturaSeguimiento? method=get>");
        Salida.println("<table border=1 class=colortable2>");
        Salida.println("<tr><td class=hcolor>");
        Salida.println("Ir a Seguimiento del año:</td>");
        Salida.println("<td><select name=codigo_ano size=1>");
        while (ResultadoSQL.next())

```

```

        {
        año=ResultadoSQL.getString(1);
        Salida.println("<option value="+año+">" + año + "</option>");
        }
    Salida.println("</select></td>");
    Salida.println("<td><input type=submit value=Aceptar></td></tr></table>");
    Salida.println("<input type=hidden name=codigo_obra value="+codigo_obra+">");
    Salida.println("<input type=hidden name=codigo_proyecto value="+codigo_proyecto+">");
    Salida.println("<input type=hidden name=mes1 value=1>");
    Salida.println("<input type=hidden name=mes2 value=12>");
    Salida.println("<input type=hidden name=adicionar value=F>");
    Salida.println("<input type=hidden name=USUARIO value="+usuario+">");
    Salida.println("</form>");
}

Salida.println("<br><br>");

//Mostrando enlaces para pantallas de fuentes de financiamiento y seguimiento de proyectos
consulta="select count(*) from PROYECTOS_FUENTES where codigo_obra="+codigo_obra+" and
codigo_proyecto="+codigo_proyecto;
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
numero=ResultadoSQL.getInt(1);
    if(numero!=0)
    {
        String codigo_fuente;
        consulta="select codigo_fuente from PROYECTOS_FUENTES where
codigo_obra="+codigo_obra+" and codigo_proyecto="+codigo_proyecto;
        ResultadoSQL=InstruccionSQL.executeQuery(consulta);
        Salida.println("<form action=/servlet/CapturaFuentes? method=get>");
        Salida.println("<table border=1 class=colortable2>");
        Salida.println("<tr><td class=hcolor>");
        Salida.println("Modificar informacion de la fuente:</td>");
        Salida.println("<td><select name=codigo_fuente size=1>");
        while (ResultadoSQL.next())
        {
            codigo_fuente=ResultadoSQL.getString(1);
            Salida.println("<option value="+codigo_fuente+">" + codigo_fuente + "</option>");
        }
        Salida.println("</select></td>");

        Salida.println("<td><input type=submit value=Aceptar></td></tr></table>");
        Salida.println("<input type=hidden name=codigo_obra value="+codigo_obra+">");
        Salida.println("<input type=hidden name=codigo_proyecto value="+codigo_proyecto+">");
        Salida.println("<input type=hidden name=adicionar value=F>");
        Salida.println("<input type=hidden name=USUARIO value="+usuario+">");
        Salida.println("</form>");
    }

    consulta="select codigo_proyecto from proyectos";
    ResultadoSQL=InstruccionSQL.executeQuery(consulta);
    String campo;
}

```

```

    }
    catch (Exception Error)
    {
        Salida.println("Situación Anormal!");
        Error.printStackTrace(Salida);
    }
    finally
    {
        try
        {
            if (ResultadoSQL!=null) ResultadoSQL.close();
            if (InstruccionSQL!=null) InstruccionSQL.close();
            if (ConexionSQL!=null) ConexionSQL.close();
        }
        catch (Exception Error) {}
    }
}
}
}

```

### *ActualizacionProyectos*

```

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class ActualizacionProyectos extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Proyectos</title>");
        Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
        Salida.println("</head>");
        Salida.println("<body>");

        // Ejecutando Proceso Principal
        Proceso(Salida,Requerimiento);

        // Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
    }
}

```

```

Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

```

```

private void Proceso(PrintWriter Salida,HttpServletRequest Requerimiento)
{

```

```

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

```

```

try
{

```

```

// Iniciando Conexión

```

```

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

```

```

    ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();

```

```

//Obteniendo datos del formulario y acción a ejecutar (inserción o modificación)

```

```

    String adicionar=Requerimiento.getParameter("adicionar");
    String usuario=Requerimiento.getParameter("USUARIO");
    String codigo_obra=Requerimiento.getParameter("CODIGO_OBRA");
    String codigo_proyecto=Requerimiento.getParameter("CODIGO_PROYECTO");
    String nombre_proyecto=Requerimiento.getParameter("NOMBRE_PROYECTO");
    String descripcion_proyecto=Requerimiento.getParameter("DESCRIPCION_PROYECTO");
    String
fecha_inicio_programada=Requerimiento.getParameter("FECHA_INICIO_PROGRAMADA");
    String
fecha_final_programada=Requerimiento.getParameter("FECHA_FINAL_PROGRAMADA");
    String fecha_inicio_real=Requerimiento.getParameter("FECHA_INICIO_REAL");
    String fecha_final_real=Requerimiento.getParameter("FECHA_FINAL_REAL");
    String codigo_etapa=Requerimiento.getParameter("CODIGO_ETAPA");
    String codigo_estado=Requerimiento.getParameter("CODIGO_ESTADO");
    String empleos_proyecto=Requerimiento.getParameter("EMPLEOS_PROYECTO");
    String beneficiarios_proyecto=Requerimiento.getParameter("BENEFICIARIOS_PROYECTO");
    String codigo_tipo=Requerimiento.getParameter("CODIGO_TIPO");
    String codigo_necesidad=Requerimiento.getParameter("CODIGO_NECESIDAD");

    String accion=" ";
    String consulta=" ";

```

```

//Construcción de sentencias según acción a realizar (insert o update)

```

```

if(adicionar.equals("T"))
accion="SET DATEFORMAT DMY insert into PROYECTOS
values('"+codigo_obra+"','"+codigo_proyecto+"','"+nombre_proyecto+"','"+
        ""+descripcion_proyecto+"','"+fecha_inicio_programada+"','"+fecha_final_programada+"','"+
        fecha_inicio_real+"','"+fecha_final_real+"','"+codigo_etapa+"','"+codigo_estado+"','"+
        empleos_proyecto+"','"+beneficiarios_proyecto+"','"+codigo_tipo+"','"+codigo_necesidad+"','"+usuar
io+"')";
else
accion="SET DATEFORMAT DMY update PROYECTOS set
nombre_proyecto='"+nombre_proyecto+"',descripcion_proyecto='"+descripcion_proyecto+
        "','fecha_inicio_programada='"+fecha_inicio_programada+"',fecha_final_programada='"+fecha_fin
al_programada+
        "','fecha_inicio_real='"+fecha_inicio_real+"',fecha_final_real='"+fecha_final_real+"',codigo_etapa
='"+codigo_etapa+
        "','codigo_estado='"+codigo_estado+"',empleos_proyecto='"+empleos_proyecto+"',beneficiarios_pro
yecto='"+beneficiarios_proyecto+
        "','codigo_tipo='"+codigo_tipo+"',codigo_necesidad='"+codigo_necesidad+"',codigo_responsable="
+usuario+
        "' where codigo_obra='"+codigo_obra+"' and codigo_proyecto='"+codigo_proyecto;

InstruccionSQL.executeUpdate(accion);

```

```

consulta="select nombre_obra from obras_fisicas where codigo_obra='"+codigo_obra+""";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_obra=ResultadoSQL.getString(1);

```

//Mostrando los datos almacenados en la base de datos

```

Salida.println("<table align=center border=1 class=colortable2 width=100%>");
Salida.println("<tr>");
Salida.println("<td class=tdfont>Obra:</td>");
Salida.println("<td class=tdfontletra>"+codigo_obra+" - "+nombre_obra+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=tdfont>Proyecto:</td>");
Salida.println("<td class=tdfontletra>"+codigo_proyecto+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=tdfont>Nombre:</td>");
Salida.println("<td class=tdfontletra>"+nombre_proyecto+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=tdfont valign=top>Descripcion:</td>");
Salida.println("<td class=tdfontletra>"+descripcion_proyecto+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=tdfont>Fecha Inicio Programada:</td>");
Salida.println("<td class=tdfontletra>"+fecha_inicio_programada+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=tdfont>Fecha Final Programada:</td>");

```

```

Salida.println("<td class=tdfontletra>"+fecha_final_programada+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=tdfont>Fecha Inicio Real:</td>");
Salida.println("<td class=tdfontletra>"+fecha_inicio_real+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=tdfont>Fecha Final Real:</td>");
Salida.println("<td class=tdfontletra>"+fecha_final_real+"</td>");
Salida.println("</tr>");

Salida.println("<tr>");
Salida.println("<td class=tdfont>Etapa:</td>");

consulta="select nombre_etapa from ETAPAS where codigo_etapa="+codigo_etapa;
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_etapa=ResultadoSQL.getString(1);

Salida.println("<td><font size=2>"+nombre_etapa+"</td>");
Salida.println("</tr>");

Salida.println("<tr>");
Salida.println("<td class=tdfont>Estado:</td>");

consulta="select nombre_estado from ESTADOS where codigo_estado="+codigo_estado;
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_estado=ResultadoSQL.getString(1);

Salida.println("<td>"+nombre_estado+"</td>");
Salida.println("</tr>");

Salida.println("<tr>");
Salida.println("<td class=tdfont>Empleos:</td>");
Salida.println("<td><font size=2>"+empleos_proyecto+"</td>");
Salida.println("</tr>");

Salida.println("<tr>");
Salida.println("<td class=tdfont>Beneficiarios:</td>");
Salida.println("<td><font size=2>"+beneficiarios_proyecto+"</td>");
Salida.println("</tr>");

Salida.println("<tr>");
Salida.println("<td class=tdfont>Tipo de Proyecto:</td>");

consulta="select nombre_tipo from TIPOS_PROYECTO where codigo_tipo="+codigo_tipo;
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_tipo=ResultadoSQL.getString(1);

Salida.println("<td><font size=2>"+nombre_tipo+"</td>");
Salida.println("</tr>");

Salida.println("<tr>");

```

```

//Enviando mensaje de confirmación de acción
if (adicionar.equals("T"))
Salida.println("<td colspan=2 align=center class=trbotones><b>El Registro ha sido Agregado</td>");
else
Salida.println("<td colspan=2 align=center class=trbotones><b>El Registro ha sido Modificado</td>");
Salida.println("</tr>");
Salida.println("</table>");

Salida.println("<br>");

Salida.println("<a
href=/servlet/ListadoProyectos?codigo_obra="+codigo_obra+"&USUARIO="+usuario+">Regresar a
Listado de Proyectos</a><br>");
Salida.println("<a
href=/servlet/ListadoFuentes?codigo_obra="+codigo_obra+"&USUARIO="+usuario+"&codigo_proyecto
="+codigo_proyecto+">Ver Listado de Fuentes de Financiamiento</a><br>");
Salida.println("<a
href=/servlet/SeleccionPeriodo?codigo_obra="+codigo_obra+"&USUARIO="+usuario+"&codigo_proyect
o="+codigo_proyecto+">Selección de período para seguimiento</a><br>");

}
catch (Exception Error)
{
Salida.println("Situación Anormal!");
Error.printStackTrace(Salida);
}
finally
{
try
{
if (ResultadoSQL!=null) ResultadoSQL.close();
if (InstruccionSQL!=null) InstruccionSQL.close();
if (ConexionSQL!=null) ConexionSQL.close();
}
catch (Exception Error) {}
}
}

public void doPost(HttpServletRequest Requerimiento,
HttpServletResponse Respuesta)
throws ServletException, java.io.IOException
{
doGet(Requerimiento,Respuesta);
}
}

```

## Registro fuentes de financiamiento para cada proyecto

Para el registro de fuentes de financiamiento por proyectos se han desarrollado tres servlets:

- ListadoFuentes: Selecciona los registros de fuentes de financiamiento registradas en la base de datos para un proyecto en particular, así como también proporcionar la interfaz para la inscripción de nuevas fuentes participantes en el proyecto. Cada registro mostrado posee un enlace que al ser activado ejecuta el servlet CapturaFuentes
- CapturaFuentes: Selecciona los valores de los campos del registro seleccionado, los cuales son presentados en un formulario.
- ActualizaciónFuentes: Actualiza o inserta un registro según haya seleccionado el usuario.

### ListadoFuentes

```
import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class ListadoFuentes extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
        throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Fuentes de Financiamiento</title>");
        Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
        Salida.println("<SCRIPT language=JavaScript src=/HTML/include/ourScript.js'
type=text/javascript></SCRIPT>");
        Salida.println("</head>");
        Salida.println("<body>");

        String codigo_obra=Requerimiento.getParameter("codigo_obra");
```

```

String codigo_proyecto=Requerimiento.getParameter("codigo_proyecto");
String usuario=Requerimiento.getParameter("USUARIO");

// Ejecutando Proceso Principal
Proceso(Salida,codigo_obra,codigo_proyecto,usuario);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

private void Proceso(PrintWriter Salida,String codigo_obra,String codigo_proyecto,String usuario)
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

// Iniciando Conexión
Usuario MIUSUARIO = new Usuario();
String Driver = MIUSUARIO.driver;
String DB = MIUSUARIO.myBD;
String Usr = MIUSUARIO.usuario;
String Pass= MIUSUARIO.password;
String dirhtml= MIUSUARIO.DirHTML;

ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
InstruccionSQL=ConexionSQL.createStatement();

//Seleccionando registros de fuentes de financiamiento para el proyecto seleccionado
String Consulta="select codigo_fuente,monto from Proyectos_fuentes where
codigo_obra="+codigo_obra+" and codigo_proyecto="+codigo_proyecto;
String codigo="";
String monto="";

//Interfaz para ingreso de nuevas fuentes
Salida.println("<form method=get
action=/servlet/CapturaFuentes?codigo_obra="+codigo_obra+"&USUARIO="+usuario+"&adicionar=T">");
);
//Listado de registros encontrados
Salida.println("<table border=0 align=center class=colortable>");
Salida.println("<tr>");
Salida.println("<td class=hcolor>Código Fuente:</td><td><select name=codigo_fuente>");
String fuentes="select codigo_fuente from fuentes_financiamiento";
String codigo_fuente=" ";
String nombre_fuente=" ";
ResultadoSQL=InstruccionSQL.executeQuery(fuentes);
while(ResultadoSQL.next())

```

```

{
codigo_fuente=ResultadoSQL.getString(1);
Salida.println("<option value="+codigo_fuente+">"+codigo_fuente+"</option>");
}
Salida.println("</select></td><td><input type=submit value='Adicionar'></td>");
Salida.println("<td><input type=hidden name='codigo_proyecto' value='"+codigo_proyecto+"' ></td>");
Salida.println("<td><input type=hidden name='adicionar' value='T' ></td>");
Salida.println("<td><input type=hidden name='codigo_obra' value='"+codigo_obra+"' ></td>");
Salida.println("<td><input type=hidden name='USUARIO' value='"+usuario+"' ></td>");
Salida.println("</tr>");
Salida.println("</table>");

Salida.println(" <table align=center class=colortable2>");
Salida.println(" <tr>");
Salida.println(" <td width=25% align=center class=hcolor>Fuente</td>");
Salida.println(" <td width=70% align=center class=hcolor>Monto</td>");
Salida.println(" </tr>");

ResultadoSQL=InstruccionSQL.executeQuery(Consulta);
while(ResultadoSQL.next())
{
codigo=ResultadoSQL.getString(1);
monto=ResultadoSQL.getString(2);

Salida.println("<br>");

Salida.println(" <tr>");
Salida.println(" <td class=tdfont><a
href=/servlet/CapturaFuentes?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&
codigo_fuente="+codigo+"&USUARIO="+usuario+"&adicionar=F">"+codigo+"</a></td>");
Salida.println(" <td><font size=2>"+monto+"</td>");
Salida.println(" </tr>");
}
Salida.println(" </table>");

}
catch (Exception Error)
{
Salida.println("Situación Anormal!");
Error.printStackTrace(Salida);
}
finally
{
try
{
if (ResultadoSQL!=null) ResultadoSQL.close();
if (InstruccionSQL!=null) InstruccionSQL.close();
if (ConexionSQL!=null) ConexionSQL.close();
}
catch (Exception Error) {}
}
}
}

```

## CapturaFuentes

```
import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class CapturaFuentes extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Fuentes de Financiamiento</title>");
        Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
        Salida.println("<SCRIPT language=JavaScript src=\"/HTML/include/ourScript.js\"
type=text/javascript></SCRIPT>");
        Salida.println("</head>");
        Salida.println("<body>");

        String codigo_obra=Requerimiento.getParameter("codigo_obra");
        String codigo_proyecto=Requerimiento.getParameter("codigo_proyecto");
        String codigo_fuente=Requerimiento.getParameter("codigo_fuente");
        String usuario=Requerimiento.getParameter("USUARIO");
        String adicionar=Requerimiento.getParameter("adicionar");

        // Ejecutando Proceso Principal
        Proceso(Salida,codigo_obra,codigo_proyecto,codigo_fuente,usuario,adicionar);

        // Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
        Salida.println("</body>");
        Salida.println("</html>");
        Salida.flush();
        Salida.close();
    }

    private void Proceso(PrintWriter Salida,String codigo_obra,String codigo_proyecto, String
codigo_fuente,String usuario,String adicionar)
    {

        // Inicializando Variables para Conexión
        Connection ConexionSQL=null;
```

```

Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();
    String consulta=" ";

    String codigo_tipofuente;
    String monto;

if(adicionar.equals("T"))
{
    codigo_tipofuente=" ";
    monto="0.00";
}
else
{
    consulta="select codigo_tipofuente,monto from PROYECTOS_FUENTES where
codigo_obra="+codigo_obra+" and codigo_proyecto="+codigo_proyecto+" and
codigo_fuente="+codigo_fuente+"";

    ResultadoSQL=InstruccionSQL.executeQuery(consulta);
    ResultadoSQL.next();
    codigo_tipofuente=ResultadoSQL.getString(1);
    monto=ResultadoSQL.getString(2);
}

consulta="select nombre_obra from obras_fisicas where codigo_obra="+codigo_obra+"";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_obra=ResultadoSQL.getString(1);

consulta="select nombre_proyecto from proyectos where codigo_proyecto="+codigo_proyecto;
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_proyecto=ResultadoSQL.getString(1);

consulta="select nombre_fuente from fuentes_financiamiento where codigo_fuente="+codigo_fuente+"";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_fuente=ResultadoSQL.getString(1);

```

```

Salida.println("<form name=form1 method=get action='/servlet/ActualizacionFuentes?' onsubmit='return
NoNulos(this)'>");
Salida.println(" <table align=center class=colortable2>");
Salida.println(" <tr><td colspan=2 align=center class=hcolor>CAPTURA FUENTES DE
FINANCIAMIENTO</td></tr>");
Salida.println(" <tr>");
Salida.println(" <td class=tdfont>Obra:</td>");
Salida.println(" <td>"+codigo_obra+" - "+nombre_obra+"</td>");
Salida.println(" </tr>");
Salida.println(" <tr>");
Salida.println(" <td class=tdfont>Proyecto:</td>");
Salida.println(" <td>"+codigo_proyecto+" - "+nombre_proyecto+"</td>");
Salida.println(" </tr>");
Salida.println(" <tr>");
Salida.println(" <td class=tdfont>Fuente:</td>");
Salida.println(" <td>"+codigo_fuente+" - "+nombre_fuente+"</td>");
Salida.println(" </tr>");
Salida.println(" <tr>");
Salida.println(" <td class=tdfont>Tipo de financiamiento:</td>");

Salida.println(" <td><select name=CODIGO_TIPOFUENTE size=1>");

String codigo_tipo2;
String nombre_tipo;

ResultadoSQL=InstruccionSQL.executeQuery("select codigo_tipofuente,nombre_tipofuente from
TIPOS_FUENTES");
while(ResultadoSQL.next())
{
codigo_tipo2=ResultadoSQL.getString(1);
nombre_tipo=ResultadoSQL.getString(2);

if (codigo_tipofuente.equals(codigo_tipo2))
Salida.println("<option selected value="+codigo_tipo2+">"+nombre_tipo+"</option>");
else
Salida.println("<option value="+codigo_tipo2+">"+nombre_tipo+"</option>");
}

Salida.println("</select></td>");

Salida.println(" </tr>");

Salida.println(" <tr>");
Salida.println(" <td class=tdfont>Monto:</td>");
Salida.println(" <td><input type=text name=MONTO size=10 value="+monto+"
onblur=\"Decimales(this)\" onkeypress=Numeros(this)></td>");
Salida.println(" </tr>");

if (adicionar.equals("T"))
Salida.println(" <td colspan=2 align=center class=trbotones><input type=submit
value=Adicionar></td>");
else
Salida.println(" <td colspan=2 align=center class=trbotones><input type=submit
value=Actualizar></td>");

```

```

Salida.println("          </tr>");
Salida.println("          <tr>");
Salida.println("          <td colspan=2><input type=hidden name=adicionar value="+adicionar+">");
Salida.println("          <td colspan=2><input type=hidden name=codigo_obra
value="+codigo_obra+">");
Salida.println("          <td colspan=2><input type=hidden name=codigo_proyecto
value="+codigo_proyecto+">");
Salida.println("          <td colspan=2><input type=hidden name=codigo_fuente
value="+codigo_fuente+">");
Salida.println("          <td colspan=2><input type=hidden name=USUARIO value="+usuario+">");
Salida.println("          </tr>");
Salida.println(" </table>");
Salida.println("</form>");

```

```
Salida.println("<br><br>");
```

```

Salida.println("<a
href=/servlet/ListadoFuentes?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&U
SUARIO="+usuario+">Regresar a Listado de Fuentes de Financiamiento</a><br>");
Salida.println("<a
href=/servlet/ListadoProyectos?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&
USUARIO="+usuario+">Regresar a Listado de Proyectos</a><br>");
Salida.println("<a
href=/servlet/SeleccionPeriodo?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&
USUARIO="+usuario+">Selección de período para seguimiento</a><br>");

```

```

if (!(adicionar.equals("T")))
{
//IR A SEGUIMIENTO
consulta="select count(*) from SEGUIMIENTO where codigo_obra='"+codigo_obra+"' and
codigo_proyecto='"+codigo_proyecto;
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
int numero=ResultadoSQL.getInt(1);
    if(numero!=0)
    {
        String año;
        consulta="select distinct año from SEGUIMIENTO where codigo_obra='"+codigo_obra+"' and
codigo_proyecto='"+codigo_proyecto;
        ResultadoSQL=InstruccionSQL.executeQuery(consulta);
        Salida.println("<form action=/servlet/CapturaSeguimiento? method=get>");
        Salida.println("<table border=1 class=colortable2>");
        Salida.println("<tr><td class=hcolor>");
        Salida.println("Ir a Seguimiento del año:</td>");
        Salida.println("<td><select name=codigo_ano size=1>");
        while (ResultadoSQL.next())
        {
            año=ResultadoSQL.getString(1);
            Salida.println("<option value="+año+">" +año+"</option>");
        }
        Salida.println("</select></td>");
        Salida.println("<td><input type=submit value=Aceptar></td></tr></table>");
        Salida.println("<input type=hidden name=codigo_obra value="+codigo_obra+">");
    }
}

```

```

        Salida.println("<input type=hidden name=codigo_proyecto value="+codigo_proyecto+">");
        Salida.println("<input type=hidden name=USUARIO value="+usuario+">");
        Salida.println("<input type=hidden name=mes1 value=1>");
        Salida.println("<input type=hidden name=mes2 value=12>");
        Salida.println("<input type=hidden name=adicionar value=F>");
        Salida.println("</form>");
    }
}

}
catch (Exception Error)
{
    Salida.println("Situación Anormal!");
    Error.printStackTrace(Salida);
}
finally
{
    try
    {
        if (ResultadoSQL!=null) ResultadoSQL.close();
        if (InstruccionSQL!=null) InstruccionSQL.close();
        if (ConexionSQL!=null) ConexionSQL.close();
    }
    catch (Exception Error) {}
}
}
}
}
}

```

### *ActualizacionFuentes*

```

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class ActualizacionFuentes extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Fuentes de Financiamiento</title>");
    }
}

```

```

Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
Salida.println("</head>");
Salida.println("<body>");

// Ejecutando Proceso Principal
Proceso(Salida,Requerimiento);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

private void Proceso(PrintWriter Salida,HttpServletRequest Requerimiento)
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();
    String adicionar=Requerimiento.getParameter("adicionar");
    String usuario=Requerimiento.getParameter("USUARIO");
    String codigo_obra=Requerimiento.getParameter("codigo_obra");
    String codigo_proyecto=Requerimiento.getParameter("codigo_proyecto");
    String codigo_fuente=Requerimiento.getParameter("codigo_fuente");
    String codigo_tipofuente=Requerimiento.getParameter("CODIGO_TIPOFUENTE");
    String monto=Requerimiento.getParameter("MONTO");

    String accion=" ";

    if(adicionar.equals("T"))
    accion="insert into PROYECTOS_FUENTES
    values("+codigo_obra+","+codigo_proyecto+",""+codigo_fuente+",""+codigo_tipofuente+",""+monto+)";
    else

```

```
accion="update PROYECTOS_FUENTES set codigo_tipofuente="+codigo_tipofuente+",  
monto="+monto+" where codigo_obra="+codigo_obra+" and codigo_proyecto="+codigo_proyecto+" and  
codigo_fuente="+codigo_fuente+"";
```

```
InstruccionSQL.executeUpdate(accion);
```

```
String consulta="select nombre_obra from obras_fisicas where codigo_obra="+codigo_obra+"";  
ResultadoSQL=InstruccionSQL.executeQuery(consulta);  
ResultadoSQL.next();  
String nombre_obra=ResultadoSQL.getString(1);
```

```
consulta="select nombre_proyecto from proyectos where codigo_proyecto="+codigo_proyecto+" and  
codigo_obra="+codigo_obra+"";  
ResultadoSQL=InstruccionSQL.executeQuery(consulta);  
ResultadoSQL.next();  
String nombre_proyecto=ResultadoSQL.getString(1);
```

```
consulta="select nombre_fuente from FUENTES_FINANCIAMIENTO where  
codigo_fuente="+codigo_fuente+"";  
ResultadoSQL=InstruccionSQL.executeQuery(consulta);  
ResultadoSQL.next();  
String nombre_fuente=ResultadoSQL.getString(1);
```

```
Salida.println("<table align=center class=colortable2>");  
Salida.println("<tr>");  
Salida.println("<td class=tdfont>Obra:</td>");  
Salida.println("<td class=tdfontletra>"+codigo_obra+" - "+nombre_obra+"</td>");  
Salida.println("</tr>");  
Salida.println("<tr>");  
Salida.println("<td class=tdfont>Proyecto:</td>");  
Salida.println("<td class=tdfontletra>"+codigo_proyecto+" - "+nombre_proyecto+"</td>");  
Salida.println("</tr>");  
Salida.println("<tr>");  
Salida.println("<td class=tdfont>Fuente:</td>");  
Salida.println("<td class=tdfontletra>"+codigo_fuente+" - "+nombre_fuente+"</td>");  
Salida.println("</tr>");  
Salida.println("<tr>");  
Salida.println("<td class=tdfont>Tipo de Financiamiento:</td>");
```

```
consulta="select nombre_tipofuente from TIPOS_FUENTES where  
codigo_tipofuente="+codigo_tipofuente;  
ResultadoSQL=InstruccionSQL.executeQuery(consulta);  
ResultadoSQL.next();  
String nombre_tipofuente=ResultadoSQL.getString(1);
```

```
Salida.println("<td class=tdfontletra>"+nombre_tipofuente+"</td>");  
Salida.println("</tr>");  
Salida.println("<tr>");  
Salida.println("<td class=tdfont>Monto:</td>");  
Salida.println("<td class=tdfontletra>"+monto+"</td>");
```

```

Salida.println("</tr>");
Salida.println("<tr>");
if (adicionar.equals("T"))
Salida.println("<td colspan=2 align=center class=trbotones><b>El Registro ha sido Agregado</td>");
else
Salida.println("<td colspan=2 align=center class=trbotones><b>El Registro ha sido Modificado</td>");
Salida.println("</tr>");
Salida.println("</table>");

Salida.println("<br>");

Salida.println("<a
href=/servlet/ListadoProyectos?codigo_obra="+codigo_obra+"&USUARIO="+usuario+">Regresar a
Listado de Proyectos</a><br>");
Salida.println("<a
href=/servlet/ListadoFuentes?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&U
SUARIO="+usuario+">Regresar a Listado de Fuentes de Financiamiento</a><br>");
Salida.println("<a
href=/servlet/SeleccionPeriodo?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&
USUARIO="+usuario+">Selección de período para seguimiento</a><br>");

}
catch (Exception Error)
{
Salida.println("Situación Anormal!");
Error.printStackTrace(Salida);
}
finally
{
try
{
if (ResultadoSQL!=null) ResultadoSQL.close();
if (InstruccionSQL!=null) InstruccionSQL.close();
if (ConexionSQL!=null) ConexionSQL.close();
}
catch (Exception Error) {}
}
}

}

```

## Registro seguimiento de proyectos

Para realizar el registro de seguimiento de proyectos, se han desarrollado tres servlets:

- **SeleccionPeriodo:** Proporciona dos interfaces, la primera permite seleccionar un año, y un rango de meses a mostrar. La segunda permite adicionar un nuevo año. Con los botones aceptar se ejecuta el servlet CapturaSeguimiento.

- **CapturaSeguimiento:** Si la acción es adicionar, se inserta en la base de datos el año indicado con todos los meses y se presenta el formulario con todos los datos para ser llenados. Si la acción es modificar, se selecciona de la base de datos el rango de meses seleccionado por el usuario. Con el botón aceptar se ejecuta el servlet ActualizacionSeguimiento.
- **ActualizacionSeguimiento:** Obtiene los valores digitados en CapturaSeguimiento y procede a la actualización de la base de datos.

### *SeleccionPeriodo*

```

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class SeleccionPeriodo extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Seguimiento</title>");
        Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
        Salida.println("<SCRIPT language=JavaScript src=\"/HTML/include/ourScript.js\"
type=text/javascript></SCRIPT>");
        Salida.println("</head>");
        Salida.println("<body>");

        String codigo_obra=Requerimiento.getParameter("codigo_obra");
        String codigo_proyecto=Requerimiento.getParameter("codigo_proyecto");
        String usuario=Requerimiento.getParameter("USUARIO");

        // Ejecutando Proceso Principal
        Proceso(Salida,codigo_obra,codigo_proyecto,usuario);
    }
}

```



```

ResultadoSQL=InstruccionSQL.executeQuery(Consulta);
String codigo_ano;
while(ResultadoSQL.next())
{
codigo_ano=ResultadoSQL.getString(1);
Salida.println("<option value="+codigo_ano+">" +codigo_ano+"</option>");
}
Salida.println("</select></td></tr>");

Salida.println("<tr><td class=tdfont>Del mes:</td><td><select name=mес1 size=1>");
Salida.println("<option value=1>Enero</option>");
Salida.println("<option value=2>Febrero</option>");
Salida.println("<option value=3>Marzo</option>");
Salida.println("<option value=4>Abril</option>");
Salida.println("<option value=5>Mayo</option>");
Salida.println("<option value=6>Junio</option>");
Salida.println("<option value=7>Julio</option>");
Salida.println("<option value=8>Agosto</option>");
Salida.println("<option value=9>Septiembre</option>");
Salida.println("<option value=10>Octubre</option>");
Salida.println("<option value=11>Noviembre</option>");
Salida.println("<option value=12>Diciembre</option>");
Salida.println("</select></td>");

Salida.println("<td class=tdfont>Al mes:</td><td><select name=mес2 size=1>");
Salida.println("<option value=1>Enero</option>");
Salida.println("<option value=2>Febrero</option>");
Salida.println("<option value=3>Marzo</option>");
Salida.println("<option value=4>Abril</option>");
Salida.println("<option value=5>Mayo</option>");
Salida.println("<option value=6>Junio</option>");
Salida.println("<option value=7>Julio</option>");
Salida.println("<option value=8>Agosto</option>");
Salida.println("<option value=9>Septiembre</option>");
Salida.println("<option value=10>Octubre</option>");
Salida.println("<option value=11>Noviembre</option>");
Salida.println("<option value=12>Diciembre</option>");
Salida.println("</select></td>");

Salida.println("</tr>");

Salida.println("<tr><td colspan=4 align=center class=trbotones><input type=submit
value=Aceptar></td></tr>");
Salida.println("</table>");

Salida.println("<input type=hidden name=codigo_obra value="+codigo_obra+">");
Salida.println("<input type=hidden name=codigo_proyecto
value="+codigo_proyecto+">");
Salida.println("<input type=hidden name=USUARIO value="+usuario+">");
Salida.println("<input type=hidden name=adicionar value=F>");

Salida.println("</form>");
}
Salida.println("<br><br>");

```

```

        Salida.println("<hr>");
        Salida.println("<form name=form2 method=get action='/servlet/CapturaSeguimiento'
onsubmit='return NoNulos(this)' >");
        Salida.println("<table align=center class=colortable2>");
        Salida.println("<tr>");
        Salida.println("<td class=hcolor><b>Captura nuevo seguimiento para el año:</td>");
        Salida.println("<td><input type=text name=codigo_ano size=4
onblur='ValidAnio(this,1,\"+parafuncion+')' onkeypress='Numeros(this)'></td>");
        Salida.println("</tr>");
        Salida.println("<tr>");
        Salida.println("<td colspan=2 align=center class=trbotones><input type=submit
value=Aceptar></td>");
        Salida.println("</tr>");
        Salida.println("</table>");
        Salida.println("<input type=hidden name=codigo_obra value="+codigo_obra+">");
        Salida.println("<input type=hidden name=codigo_proyecto value="+codigo_proyecto+">");
        Salida.println("<input type=hidden name=USUARIO value="+usuario+">");
        Salida.println("<input type=hidden name=adicionar value=T>");
        Salida.println("</form>");

    }
    catch (Exception Error)
    {
        Salida.println("Situación Anormal!");
        Error.printStackTrace(Salida);
    }
    finally
    {
        try
        {
            if (ResultadoSQL!=null) ResultadoSQL.close();
            if (InstruccionSQL!=null) InstruccionSQL.close();
            if (ConexionSQL!=null) ConexionSQL.close();
        }
        catch (Exception Error) {}
    }
}
}
}

```

### *CapturaSeguimiento*

```

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class CapturaSeguimiento extends HttpServlet
{

```

```

public void doGet(HttpServletRequest Requerimiento, HttpServletResponse Respuesta)
throws ServletException, IOException
{

    // Abriendo Archivo de Salida
    Respuesta.setContentType("text/html");
    PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

    // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
    Salida.println("<html>");
    Salida.println("<head>");
    Salida.println("<title>Seguimiento</title>");
    Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
    Salida.println("<SCRIPT language=JavaScript src=\"/HTML/include/ourScript.js\"
type=text/javascript></SCRIPT>");
    Salida.println("</head>");
    Salida.println("<body>");

    String codigo_obra=Requerimiento.getParameter("codigo_obra");
    String codigo_proyecto=Requerimiento.getParameter("codigo_proyecto");
    String codigo_ano=Requerimiento.getParameter("codigo_ano");
    String mes1=Requerimiento.getParameter("mes1");
    String mes2=Requerimiento.getParameter("mes2");
    String adicionar=Requerimiento.getParameter("adicionar");
    String usuario=Requerimiento.getParameter("USUARIO");

    // Ejecutando Proceso Principal
    Proceso(Salida,codigo_obra,codigo_proyecto,codigo_ano,mes1,mes2,adicionar,usuario);

    // Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
    Salida.println("</body>");
    Salida.println("</html>");
    Salida.flush();
    Salida.close();
}

private void Proceso(PrintWriter Salida,String codigo_obra,String codigo_proyecto,String codigo_ano,
String mes1,String mes2,String adicionar,String usuario)
{

    // Inicializando Variables para Conexión
    Connection ConexionSQL=null;
    Statement InstruccionSQL=null;
    ResultSet ResultadoSQL=null;
    try
    {

        // Iniciando Conexión

        Usuario MIUSUARIO = new Usuario();
        String Driver = MIUSUARIO.driver;

```

```

String DB = MIUSUARIO.myBD;
String Usr = MIUSUARIO.usuario;
String Pass= MIUSUARIO.password;
String dirhtml= MIUSUARIO.DirHTML;

ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
InstruccionSQL=ConexionSQL.createStatement();

String consulta=" ";

consulta="select nombre_obra from obras_fisicas where codigo_obra="+codigo_obra+"";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_obra=ResultadoSQL.getString(1);

consulta="select nombre_proyecto from proyectos where codigo_obra="+codigo_obra+" and
codigo_proyecto="+codigo_proyecto;
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String nombre_proyecto=ResultadoSQL.getString(1);

Salida.println("<table align=center class=colortable>");
Salida.println(" <tr><td class=hcolor align=center colspan=2>CAPTURA DE
SEGUIMIENTO</td></tr>");
Salida.println("<tr>");
Salida.println("<td class=tdfont>Obra:</td><td class=tdfontletra>"+codigo_obra+" -
"+nombre_obra+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=tdfont>Proyecto:</td><td class=tdfontletra>"+codigo_proyecto+" -
"+nombre_proyecto+"</td>");
Salida.println("</tr>");
Salida.println("<tr>");
Salida.println("<td class=tdfont>Año:</td><td class=tdfontletra>"+codigo_ano+"</td>");
Salida.println("</table>");
Salida.println("</tr>");

String mes="";
String prog_finan="";
String prog_fis="";
String ejec_finan="";
String ejec_fis="";
String observaciones="";

Salida.println("<form name=form1 method=post action='/servlet/ActualizacionSeguimiento?'
onSubmit='return NoNulos(this);>");
Salida.println("<table border=1 align=center class=colortable2>");
Salida.println("<tr>");
Salida.println(" <td class=hcolor>Mes</td>");
Salida.println(" <td class=hcolor>Prog. Finan.</td>");
Salida.println(" <td class=hcolor>% Prog. Física</td>");
Salida.println(" <td class=hcolor>Ejec. Finan</td>");

```

```

Salida.println(" <td class=hcolor>% Ejec. Física</td>");
Salida.println(" <td class=hcolor>Observaciones</td>");
Salida.println(" </tr>");

if(adicionar.equals("T"))
{
    for(int i=1;i<=12;i++)
    {
        consulta="insert into SEGUIMIENTO values
("+codigo_obra+","+codigo_proyecto+","+codigo_ano+","+i+",0,0,0,0, ' )";
        InstruccionSQL.executeUpdate(consulta);
    }
    mes1="1";
    mes2="12";
}

consulta="select
programado_financiero,programado_fisico,ejecucion_financiero,ejecucion_fisico,observaciones"
" from SEGUIMIENTO where codigo_obra="+codigo_obra+" and
codigo_proyecto="+codigo_proyecto+
" and año="+codigo_ano+" and mes between "+mes1+" and "+mes2;

String Mes=" ";
int m1=Integer.parseInt(mes1);
int m2=Integer.parseInt(mes2);

ResultadoSQL=InstruccionSQL.executeQuery(consulta);
for (int i=m1;i<=m2;i++)
{
    ResultadoSQL.next();
    Salida.println("<tr>");

    if(i==1) Mes="Enero";
    if(i==2) Mes="Febrero";
    if(i==3) Mes="Marzo";
    if(i==4) Mes="Abril";
    if(i==5) Mes="Mayo";
    if(i==6) Mes="Junio";
    if(i==7) Mes="Julio";
    if(i==8) Mes="Agosto";
    if(i==9) Mes="Septiembre";
    if(i==10) Mes="Octubre";
    if(i==11) Mes="Noviembre";
    if(i==12) Mes="Diciembre";

    Salida.println(" <td><font size=2>"+Mes+"</td>");
    Salida.println(" <td align=center><input type=text name='prog_finan'+i+'
value="+ResultadoSQL.getString(1)+" size=10 onblur='Decimales(this)'
onkeypress='Numeros(this);'></td>");
    Salida.println(" <td align=center><input type=text name='prog_fisico'+i+'
value="+ResultadoSQL.getString(2)+" size=6 onblur='Decimales(this)'
onkeypress='Numeros(this);'></td>");
    Salida.println(" <td align=center><input type=text name='ejec_finan'+i+'
value="+ResultadoSQL.getString(3)+" size=10 onblur='Decimales(this)'
onkeypress='Numeros(this);'></td>");
}

```

```

                Salida.println(" <td align=center><input type=text name='ejec_fisico'+i+"
value="" +ResultadoSQL.getString(4)+" size=6 onblur='Decimales(this)'
onkeypress='Numeros(this);'></td>");
                Salida.println(" <td><textarea rows=3 cols=20
name=observaciones"+i+">" +ResultadoSQL.getString(5)+"</textarea></td>");
                Salida.println(" </tr>");
        }

        Salida.println("</table>");
        Salida.println("<center>");
        Salida.println("<br><input type=submit value=Actualizar></td>");
        Salida.println("</center>");

        Salida.println("<input type=hidden name=USUARIO value="+usuario+">");
        Salida.println("<input type=hidden name=codigo_obra value="+codigo_obra+">");
        Salida.println("<input type=hidden name=codigo_proyecto value="+codigo_proyecto+">");
        Salida.println("<input type=hidden name=codigo_ano value="+codigo_ano+">");
        Salida.println("<input type=hidden name=mes1 value="+mes1+">");
        Salida.println("<input type=hidden name=mes2 value="+mes2+">");
        Salida.println("</form>");

        Salida.println("<br><br>");
        Salida.println("<a
href=/servlet/ListadoProyectos?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&
USUARIO="+usuario+">Regresar a Listado de Proyectos</a><br>");
        Salida.println("<a
href=/servlet/ListadoFuentes?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&U
SUARIO="+usuario+">Ver Listado de Fuentes de Financiamiento</a><br>");

    }
    catch (Exception Error)
    {
        Salida.println("Situación Anormal!");
        Error.printStackTrace(Salida);
    }
    finally
    {
        try
        {
            if (ResultadoSQL!=null) ResultadoSQL.close();
            if (InstruccionSQL!=null) InstruccionSQL.close();
            if (ConexionSQL!=null) ConexionSQL.close();
        }
        catch (Exception Error) {}
    }
}

public void doPost(HttpServletRequest Requerimiento,
                HttpServletResponse Respuesta)
throws ServletException, java.io.IOException
{

```

```

        doGet(Requerimiento,Respuesta);
    }
}

```

### *ActualizacionSeguimiento*

```

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class ActualizacionSeguimiento extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Seguimiento</title>");
        Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
        Salida.println("</head>");
        Salida.println("<body>");

        // Ejecutando Proceso Principal
        Proceso(Salida,Requerimiento);

        // Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
        Salida.println("</body>");
        Salida.println("</html>");
        Salida.flush();
        Salida.close();
    }

    private void Proceso(PrintWriter Salida,HttpServletRequest Requerimiento)
    {

        // Inicializando Variables para Conexión
        Connection ConexionSQL=null;
        Statement InstruccionSQL=null;
        ResultSet ResultadoSQL=null;

        try

```

```

{
// Iniciando Conexión

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();

    String usuario=Requerimiento.getParameter("USUARIO");
    String codigo_obra=Requerimiento.getParameter("codigo_obra");
    String codigo_proyecto=Requerimiento.getParameter("codigo_proyecto");
    String codigo_ano=Requerimiento.getParameter("codigo_ano");
    String mes1S=Requerimiento.getParameter("mes1");
    String mes2S=Requerimiento.getParameter("mes2");

    int mes1=Integer.parseInt(mes1S);
    int mes2=Integer.parseInt(mes2S);

    String consulta=" ";
    String programado_financiero;
    String ejecucion_financiero;
    String programado_fisico;
    String ejecucion_fisico;
    String observaciones;
    String nombre_mes=" ";

    for(int i=mes1;i<=mes2;i++)
    {
        programado_financiero=Requerimiento.getParameter("prog_finan"+i);
        ejecucion_financiero=Requerimiento.getParameter("ejec_finan"+i);
        programado_fisico=Requerimiento.getParameter("prog_fisico"+i);
        ejecucion_fisico=Requerimiento.getParameter("ejec_fisico"+i);
        observaciones=Requerimiento.getParameter("observaciones"+i);
        consulta="update SEGUIMIENTO set programado_financiero="+programado_financiero+
",ejecucion_financiero="+ejecucion_financiero+",programado_fisico="+programado_fisico+
        ",ejecucion_fisico="+ejecucion_fisico+",observaciones="+observaciones+
        "' from seguimiento where codigo_obra='"+codigo_obra+"' and
codigo_proyecto='"+codigo_proyecto+
        "' and año='"+codigo_ano+"' and mes='"+i;

        InstruccionSQL.executeUpdate(consulta);
    }
}

```

```

        consulta="select
programado_financiero,ejecucion_financiero,programado_fisico,ejecucion_fisico,observaciones from
seguimiento where codigo_obra="+
                codigo_obra+" and codigo_proyecto="+codigo_proyecto+" and año="+codigo_año+"
and mes between "+mes1+" and "+mes2;

```

```

ResultadoSQL=InstruccionSQL.executeQuery(consulta);

```

```

Salida.println("<table align=Center border=1 class=colortable2>");
Salida.println("<tr>");
Salida.println("<td class=hcolor>Mes:</td>");
Salida.println("<td class=hcolor>Programación Financiera</td>");
Salida.println("<td class=hcolor>Programación Física</td>");
Salida.println("<td class=hcolor>Ejecución Financiera</td>");
Salida.println("<td class=hcolor>Ejecución Física</td>");
Salida.println("<td class=hcolor>Observaciones</td>");
Salida.println("</tr>");

```

```

for(int i=mes1;i<=mes2;i++)
{

```

```

        ResultadoSQL.next();
        programado_financiero=ResultadoSQL.getString(1);
        ejecucion_financiero=ResultadoSQL.getString(2);
        programado_fisico=ResultadoSQL.getString(3);
        ejecucion_fisico=ResultadoSQL.getString(4);
        observaciones=ResultadoSQL.getString(5);

```

```

        if (i==1) nombre_mes="Enero";
        if (i==2) nombre_mes="Febrero";
        if (i==3) nombre_mes="Marzo";
        if (i==4) nombre_mes="Abril";
        if (i==5) nombre_mes="Mayo";
        if (i==6) nombre_mes="Junio";
        if (i==7) nombre_mes="Julio";
        if (i==8) nombre_mes="Agosto";
        if (i==9) nombre_mes="Septiembre";
        if (i==10) nombre_mes="Octubre";
        if (i==11) nombre_mes="Noviembre";
        if (i==12) nombre_mes="Diciembre";

```

```

        Salida.println("<tr>");
        Salida.println("<td class=tdfont>"+nombre_mes+"</td>");
        Salida.println("<td align=right class=tdfontletra>"+programado_financiero+"</td>");
        Salida.println("<td align=right class=tdfontletra>"+programado_fisico+"</td>");
        Salida.println("<td align=right class=tdfontletra>"+ejecucion_financiero+"</td>");
        Salida.println("<td align=right class=tdfontletra>"+ejecucion_fisico+"</td>");
        Salida.println("<td class=tdfontletra>"+observaciones+"</td>");
        Salida.println("</tr>");

```

```

}

```

```

Salida.println("</table>");

```

```

        Salida.println("<br>");

        Salida.println("<center><font class=fontmensajes>Los Registros han sido
modificados</font></center><br><br>");

        Salida.println("<a
href='/servlet/ListadoProyectos?codigo_obra="+codigo_obra+"&USUARIO="+usuario+">Regresar a
Listado de Proyectos</a><br>");

        Salida.println("<a
href='/servlet/ListadoFuentes?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&U
SUARIO="+usuario+">Ver Listado de Fuentes de Financiamiento</a><br>");

        Salida.println("<a
href='/servlet/SeleccionPeriodo?codigo_obra="+codigo_obra+"&codigo_proyecto="+codigo_proyecto+"&
USUARIO="+usuario+">Selección de período para seguimiento</a><br>");

    }
    catch (Exception Error)
    {
        Salida.println("Situación Anormal!");
        Error.printStackTrace(Salida);
    }
    finally
    {
        try
        {
            if (ResultadoSQL!=null) ResultadoSQL.close();
            if (InstruccionSQL!=null) InstruccionSQL.close();
            if (ConexionSQL!=null) ConexionSQL.close();
        }
        catch (Exception Error) {}
    }
}

public void doPost(HttpServletRequest Requerimiento,
                    HttpServletResponse Respuesta)
    throws ServletException, java.io.IOException
{
    doGet(Requerimiento,Respuesta);
}
}

```

## Información General de Obras

El servlet InfoGralObra permite ver datos generales de las obras que están registradas, la información serán datos consolidados como: cantidad de proyectos asignados e introducidos por el usuario (estos corresponden únicamente al usuario que está activo en ese momento en el sistema.)

```

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class InfoGralObra extends HttpServlet
{
public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
throws ServletException, IOException
{

// Abriendo Archivo de Salida
Respuesta.setContentType("text/html");
PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

// Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
Salida.println("<html>");
Salida.println("<head>");
Salida.println("<title>Información General de la Obra</title>");
Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
Salida.println("</head>");
Salida.println("<body>");

String codigo_obra=Requerimiento.getParameter("codigo_obra");
String usuario=Requerimiento.getParameter("usuario");

// Ejecutando Proceso Principal
Proceso(Salida,codigo_obra,usuario);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

private void Proceso(PrintWriter Salida,String codigo_obra,String usuario)
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

```

```
// Iniciando Conexión
```

```
Usuario MIUSUARIO = new Usuario();  
String Driver = MIUSUARIO.driver;  
String DB = MIUSUARIO.myBD;  
String Usr = MIUSUARIO.usuario;  
String Pass= MIUSUARIO.password;  
String dirhtml= MIUSUARIO.DirHTML;
```

```
ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);  
InstruccionSQL=ConexionSQL.createStatement();
```

```
String consulta;
```

```
consulta="select nombre_obra from obras_fisicas where codigo_obra="+codigo_obra+"";  
ResultadoSQL=InstruccionSQL.executeQuery(consulta);  
ResultadoSQL.next();  
String nombre_obra=ResultadoSQL.getString(1);
```

```
Salida.println("<table border=1 cellpadding=1 cellspacing=3 class=colortable align=center  
width=90%>");  
Salida.println("<tr><td colspan=2 class=hcolor align=center>Información General</td></tr>");  
Salida.println("<tr>");  
Salida.println("<td class=tdfont width=5>Nombre&nbsp;Obra&nbsp;;</td><td  
class=tdfontletra>"+nombre_obra+"</td>");  
Salida.println("</tr>");
```

```
consulta="select count(*) from proyectos where codigo_obra="+codigo_obra+" and  
codigo_responsable="+usuario+" and codigo_etapa in (6,7)";  
ResultadoSQL=InstruccionSQL.executeQuery(consulta);  
ResultadoSQL.next();  
String proy_term=ResultadoSQL.getString(1);
```

```
Salida.println("<tr>");  
Salida.println("<td class=tdfont width=5>Proyectos&nbsp;Terminados&nbsp;;</td><td  
class=tdfontletra>"+proy_term+"</td>");  
Salida.println("</tr>");
```

```
consulta="select count(*) from proyectos where codigo_obra="+codigo_obra+" and  
codigo_responsable="+usuario+" and codigo_etapa in (2,3)";  
ResultadoSQL=InstruccionSQL.executeQuery(consulta);  
ResultadoSQL.next();  
String proy_noinic=ResultadoSQL.getString(1);
```

```
Salida.println("<tr>");  
Salida.println("<td class=tdfont width=5>Proyectos&nbsp;No&nbsp;Iniciados :</td><td  
class=tdfontletra>"+proy_noinic+"</td>");  
Salida.println("</tr>");
```

```

        consulta="select count(*) from seguimiento a, proyectos b where
a.codigo_obra="+codigo_obra+" and b.codigo_responsable="+usuario+" and
a.codigo_obra=b.codigo_obra and a.codigo_proyecto=b.codigo_proyecto";
        ResultadoSQL=InstruccionSQL.executeQuery(consulta);
        ResultadoSQL.next();
        int cuantos=ResultadoSQL.getInt(1);
        String inversion;
        if(cuantos!=0)
        {
            consulta="select sum(a.programado_financiero) from seguimiento a, proyectos b where
a.codigo_obra="+codigo_obra+" and b.codigo_responsable="+usuario+" and
a.codigo_obra=b.codigo_obra and a.codigo_proyecto=b.codigo_proyecto";
            ResultadoSQL=InstruccionSQL.executeQuery(consulta);
            ResultadoSQL.next();
            inversion=ResultadoSQL.getString(1);
        }
        else inversion="0";

        Salida.println("<tr>");
        Salida.println("<td class=tdfont width=5>Monto&nbsp;Invertido :</td><td class=tdfontletra>
$"+inversion+" </td>");
        Salida.println("</tr>");

        consulta="select count(*) from proyectos where codigo_obra="+codigo_obra+" and
codigo_responsable="+usuario+"";
        ResultadoSQL=InstruccionSQL.executeQuery(consulta);
        ResultadoSQL.next();
        cuantos=ResultadoSQL.getInt(1);
        String empleos;

        if(cuantos!=0)
        {
            consulta="select sum(empleos_proyecto) from proyectos where codigo_obra="+codigo_obra+"
and codigo_responsable="+usuario+"";
            ResultadoSQL=InstruccionSQL.executeQuery(consulta);
            ResultadoSQL.next();
            empleos=ResultadoSQL.getString(1);
        }
        else empleos="0";

        if(empleos.equals("null")) empleos="0";

        Salida.println("<tr>");
        Salida.println("<td class=tdfont width=5>Empleos&nbsp;Generados :</td><td class=tdfontletra>
"+empleos+" </td>");
        Salida.println("</tr>");

    }
    catch (Exception Error)
    {
        Salida.println("Situación Anormal!");
    }

```

```

    Error.printStackTrace(Salida);
}
finally
{
    try
    {
        if (ResultadoSQL!=null) ResultadoSQL.close();
        if (InstruccionSQL!=null) InstruccionSQL.close();
        if (ConexionSQL!=null) ConexionSQL.close();
    }
    catch (Exception Error) {}
}
}
}
}

```

## **Administración**

Para la administración de tablas con datos del sistema, que únicamente pueden ser manipulados por el administrador del sistema, se ha generado un esquema para la creación automática de las interfaces de acuerdo a los datos de cada tabla.

Haciendo uso de la estructura definida para este propósito y con tres servlets se logra el esquema.

Los tres servlets desarrollados son los siguientes:

- Administración: Verifica la tabla que se esta modificando, selecciona los registros de la base de datos y los muestra, proporcionando un enlace que al ser activado ejecuta el servlet AdministracionModificacion.
- AdministracionModificacion: Selecciona los campos del registro seleccionado y muestra los valores en un formulario. Se presentan las opciones adicionar, modificar, eliminar y cancelar. Al activar cualquiera de estos botones (a excepción de cancelar) se ejecuta el servlet AdministracionActualizacion.
- AdministracionActualizacion: Recibe todos los parámetros, construye la cadena que define la acción a realizar (inserción, eliminación, actualización).

### *Administración*

```

import Usuario;
import Menu;
import javax.servlet.*;

```

```

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

public class Administracion extends HttpServlet
{
public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
throws ServletException, IOException
{

// Abriendo Archivo de Salida
Respuesta.setContentType("text/html");
PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

// Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
Salida.println("<html>");
Salida.println("<head>");
Salida.println("<title>Administración</title>");
Salida.println("<link href='/HTML/include/ourstyle.css' rel='stylesheet' type='text/css'>");
Salida.println("<SCRIPT language=JavaScript src='/HTML/include/ourScript.js'
type=text/javascript></SCRIPT>");
Salida.println("</head>");
Salida.println("<body>");

//Recibiendo parámetro de tabla seleccionada
String tabla=Requerimiento.getParameter("tabla");
String nivel_s=Requerimiento.getParameter("nivel");
int nivel=Integer.parseInt(nivel_s);

// Ejecutando Proceso Principal
Proceso(Salida,tabla,nivel,Requerimiento);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

private void Proceso(PrintWriter Salida,String tabla,int nivel,HttpServletRequest Requerimiento)
{

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

```

```

String Usuario = Requerimiento.getParameter("USUARIO");

// Iniciando Conexión

Usuario MIUSUARIO = new Usuario();
String Driver = MIUSUARIO.driver;
String DB = MIUSUARIO.myBD;
String Usr = MIUSUARIO.usuario;
String Pass= MIUSUARIO.password;
String dirhtml= MIUSUARIO.DirHTML;

Menu DoMenu = new Menu();

DoMenu.CREARMENU(Usuario,Salida,Driver,Usr,DB,Pass);

DoMenu.Encabezado(Salida);

ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
InstruccionSQL=ConexionSQL.createStatement();

String consulta;
String cam1;
String cam2;
int num_llave;

Salida.println("<p><center><font size=2 face=verdana><b>Administración de la tabla
"+tabla+"</b></font></center></p>");

if (nivel==1)
{
    Salida.println("<p><center><a
href=/servlet/Administracion_modificacion?adicionar=T&tabla="+tabla+"&nivel="+nivel+"&USUARIO=
"+Usuario+">Adicionar Registro</a></center>");

//seleccionando los campos de la tabla seleccionada

    consulta="select nombre_campo,rotulo_campo from tablas_campos where
nombre_tabla='"+tabla+"' order by orden_campo";
    ResultadoSQL=InstruccionSQL.executeQuery(consulta);
    ResultadoSQL.next();
    String campo1=ResultadoSQL.getString(1);
    String rotulo1=ResultadoSQL.getString(2);
    ResultadoSQL.next();
    String campo2=ResultadoSQL.getString(1);
    String rotulo2=ResultadoSQL.getString(2);

    if(tabla.equals("OBRAS_FISICAS"))    consulta="select codigo_obra,nombre_obra from
"+tabla;
    else                                consulta="select "+campo1+", "+campo2+" from
"+tabla;
    ResultadoSQL=InstruccionSQL.executeQuery(consulta);

    Salida.println("<p><table border=1 class=colortable2 align=center>");

```

```

        if(tabla.equals("OBRAS_FISICAS")) Salida.println("<tr><td align=center
class=hcolor>Código</td><td align=center class=hcolor>Nombre de la obra</td></tr>");
        else Salida.println("<tr><td align=center
class=hcolor>"+rotulo1+"</td><td align=center class=hcolor>"+rotulo2+"</td></tr>");

        while(ResultadoSQL.next())
        {
            cam1=ResultadoSQL.getString(1);
            cam2=ResultadoSQL.getString(2);
            Salida.println("<tr>");
            Salida.println("<td class=tdfont align=center><a class=linkblanco
href=/servlet/Administracion_modificacion?adicionar=F&tabla="+tabla+"&nivel="+nivel+"&campo1="+c
ampo1+"&dato1="+cam1+"&USUARIO="+Usuario+">"+cam1+"</a></td>");
            Salida.println("<td>"+cam2+"</td>");
            Salida.println("</tr>");
        }

        Salida.println("</table>");
    }
    else
    {
        //Creando cadena de condiciones para la selección de registros
        String condicion="";
        for (int i=1;i<nivel;i++)
        {
            if (i==1)
            {
                condicion=Requerimiento.getParameter("campo"+i);
                condicion=condicion+"="+Requerimiento.getParameter("dato"+i)+"";
            }
            else
            {
                condicion=condicion+" and "+Requerimiento.getParameter("campo"+i);
                condicion=condicion+"="+Requerimiento.getParameter("dato"+i)+"";
            }
        }

        Salida.print("<p><center><a
href=/servlet/Administracion_modificacion?adicionar=T&tabla="+tabla+"&nivel="+nivel+"&USUARIO=
"+Usuario);

        //Creando vínculo con parámetros necesarios

        for (int i=1;i<nivel;i++)
        {
            Salida.print("&campo"+i+"="+Requerimiento.getParameter("campo"+i)+"&dato"+i+"="+Requeri
miento.getParameter("dato"+i));
        }

        Salida.println(">Adicionar Registro</a></center>");

        consulta="select nombre_campo,rotulo_campo from tablas_campos where
nombre_tabla="+tabla+" order by orden_campo";
        ResultadoSQL=InstruccionSQL.executeQuery(consulta);
        ResultadoSQL.next();
    }
}

```

```

String campo1=ResultadoSQL.getString(1);
String rotulo1=ResultadoSQL.getString(2);
ResultadoSQL.next();
String campo2=ResultadoSQL.getString(1);
String rotulo2=ResultadoSQL.getString(2);

consulta="select "+campo1+","+campo2+" from "+tabla+" where "+condicion;
ResultadoSQL=InstruccionSQL.executeQuery(consulta);

Salida.println("<p><table border=1 class=colortable2 align=center>");

Salida.println("<tr><td class=hcolor align=center>"+rotulo1+"</td><td
align=center>"+rotulo2+"</td></tr>");

while(ResultadoSQL.next())
{
cam1=ResultadoSQL.getString(1);
cam2=ResultadoSQL.getString(2);
Salida.println("<tr>");
Salida.print("<td class=tdfont><a class=linkblanco
href=/servlet/Administracion_modificacion?adicionar=F&tabla="+tabla+"&nivel="+nivel+"&USUARIO=
"+Usuario);

for (int i=1;i<nivel;i++)
{

Salida.print("&campo"+i+"="+Requerimiento.getParameter("campo"+i)+"&dato"+i+"="+Requeri
miento.getParameter("dato"+i));
}

Salida.print("&campo"+nivel+"="+campo1+"&dato"+nivel+"="+cam1+">"+cam1+"</a></td>");
Salida.println("<td>"+cam2+"</td>");
Salida.println("</tr>");
}

Salida.println("</table>");
}

}
catch (Exception Error)
{
Salida.println("Situación Anormal!");
Error.printStackTrace(Salida);
}
finally
{
try
{
if (ResultadoSQL!=null) ResultadoSQL.close();
if (InstruccionSQL!=null) InstruccionSQL.close();
if (ConexionSQL!=null) ConexionSQL.close();
}
}

```

```

    catch (Exception Error) {}
  }
}
}

```

### *AdministracionModificacion*

```

import Usuario;
import Menu;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class Administracion_modificacion extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Administración</title>");
        Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
        Salida.println("<SCRIPT language=JavaScript src=\"/HTML/include/ourScript.js\"
type=text/javascript></SCRIPT>");

        Salida.println("</head>");
        Salida.println("<body>");

        String tabla=Requerimiento.getParameter("tabla");
        String adicionar=Requerimiento.getParameter("adicionar");
        String campo=Requerimiento.getParameter("campo");
        String dato=Requerimiento.getParameter("dato");
        String nivel_s=Requerimiento.getParameter("nivel");

        // Ejecutando Proceso Principal
        Proceso(Salida,tabla,adicionar,Requerimiento,nivel_s,campo,dato);

        // Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
        Salida.println("</body>");
        Salida.println("</html>");
        Salida.flush();
    }
}

```

```

        Salida.close();
    }

private void Proceso(PrintWriter Salida,String tabla,String adicionar,HttpServletRequest
Requerimiento,String nivel_s,String campo,String dato)
{

        String Usuario = Requerimiento.getParameter("USUARIO");

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;
Connection ConexionSQL2=null;
Statement InstruccionSQL2=null;
ResultSet ResultadoSQL2=null;
Connection ConexionSQL3=null;
Statement InstruccionSQL3=null;
ResultSet ResultadoSQL3=null;

try
{

        Usuario MIUSUARIO = new Usuario();
        String Driver = MIUSUARIO.driver;
        String DB = MIUSUARIO.myBD;
        String Usr = MIUSUARIO.usuario;
        String Pass= MIUSUARIO.password;
        String dirhtml= MIUSUARIO.DirHTML;

        Menu DoMenu = new Menu();

        DoMenu.CREARMENU(Usuario,Salida,Driver,Usr,DB,Pass);

        DoMenu.Encabezado(Salida);

// Iniciando Conexión

        Class.forName(Driver);
        ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
        InstruccionSQL=ConexionSQL.createStatement();
        ConexionSQL2 = DriverManager.getConnection(DB,Usr,Pass);
        InstruccionSQL2=ConexionSQL2.createStatement();
        ConexionSQL3 = DriverManager.getConnection(DB,Usr,Pass);
        InstruccionSQL3=ConexionSQL3.createStatement();

String consulta;
String cadena="select ";
String con;
String val1;
String val2;

```

```

String nombre_campo;
String rotulo_campo;
String long_campo;
String llave_campo;
String orden_campo;
String valor_campo;
String tipo_campo;
String esllave_campo;
String esllave="";
String datollave="";
String otra_consulta;
String valor_seleccion;

```

```

Salida.println("<p><center><font size=2 face=verdana><b>Administración de registros de la tabla
"+tabla+"</b></font></center></p>");

```

```

Salida.println("<form name=form1 method=post action=/servlet/Administracion_actualizacion?
OnSubmit='return NoNulos(this);>");
String condicion="";
int nivel=Integer.parseInt(nivel_s);

for (int i=1;i<=nivel;i++)
{
if(i==1)
condicion=Requerimiento.getParameter("campo"+i)+"="+Requerimiento.getParameter("dato"+i)+"";
else condicion=condicion+" and
"+Requerimiento.getParameter("campo"+i)+"="+Requerimiento.getParameter("dato"+i)+"";
}

```

```

//Verificando la acción a realizar: T=Adicionar registro
if(!(adicionar.equals("T")))
{

```

```

    consulta="select count(*) from tablas_campos where nombre_tabla="+tabla+"";
    ResultadoSQL=InstruccionSQL.executeQuery(consulta);
    ResultadoSQL.next();
    int cuantos=ResultadoSQL.getInt(1);

```

```

    consulta="select nombre_campo from tablas_campos where nombre_tabla="+tabla+" order by
orden_campo";

```

```

    ResultadoSQL=InstruccionSQL.executeQuery(consulta);
    for (int i=1;i<=cuantos;i++)
    {
    ResultadoSQL.next();
    nombre_campo=ResultadoSQL.getString(1);
    if (i==cuantos) cadena=cadena+nombre_campo;
    else cadena=cadena+nombre_campo+",";
    }

```

```

    consulta="select
nombre_campo,rotulo_campo,tipo_campo,long_campo,llave_campo,orden_campo,esllave_campo from
tablas_campos where nombre_tabla="+tabla+" order by orden_campo";
    cadena=cadena+" from "+tabla+" where "+condicion;

```

```

ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL2=InstruccionSQL2.executeQuery(cadena);
ResultadoSQL2.next();
int i=0;

Salida.println("<table border=1 class=colortable2 align=center>");

while(ResultadoSQL.next())
{
Salida.println("<tr>");
i++;
valor_campo=ResultadoSQL2.getString(i);
nombre_campo=ResultadoSQL.getString(1);
rotulo_campo=ResultadoSQL.getString(2);
tipo_campo=ResultadoSQL.getString(3);
long_campo=ResultadoSQL.getString(4);
llave_campo=ResultadoSQL.getString(5);
orden_campo=ResultadoSQL.getString(6);
esllave_campo=ResultadoSQL.getString(7);

for (int k=1;k<=nivel;k++)
{
if(Requerimiento.getParameter("campo"+k).equals(nombre_campo))
esllave="T";
}

//Verificando que el campo es llave del registro y por tanto no se puede modificar

if(esllave.equals("T")||esllave_campo.equals("S"))
{
if (tipo_campo.equals("fecha"))
Salida.println("<td class=tdfont>"+rotulo_campo+"<td><input type=text size=10
readonly value='"+valor_campo+"' name='"+nombre_campo+"' onblur='ValidFecha(this)'>");

if (tipo_campo.equals("texto"))
Salida.println("<td class=tdfont>"+rotulo_campo+"<td><textarea rows=3 cols=30
readonly name='"+nombre_campo+"'>"+valor_campo+"</textarea>");

if (tipo_campo.equals("entero"))
Salida.println("<td class=tdfont>"+rotulo_campo+"<td><input type=text
size="+long_campo+" readonly value='"+valor_campo+"' name='"+nombre_campo+"'
onblur='Enteros(this);' onkeypress='Numeros(this);'>");

if (tipo_campo.equals("numerico"))
Salida.println("<td class=tdfont>"+rotulo_campo+"<td><input type=text
size="+long_campo+" readonly value='"+valor_campo+"' name='"+nombre_campo+"'
onblur='Decimales(this);' onkeypress='Numeros(this);'>");

if (tipo_campo.equals("caracter"))
Salida.println("<td class=tdfont>"+rotulo_campo+"<td><input type=text
size="+long_campo+" readonly value='"+valor_campo+"' name='"+nombre_campo+"'>");

if (tipo_campo.equals("seleccion"))
{

```

```

        otra_consulta="select * from "+llave_campo+" where
"+nombre_campo+"="+valor_campo+"";
        ResultadoSQL3=InstruccionSQL3.executeQuery(otra_consulta);
        ResultadoSQL3.next();
        valor_seleccion=ResultadoSQL3.getString(2);
        Salida.println("<td class=tdfont>"+rotulo_campo+"<td><input type=text
size="+long_campo+" readonly value="+valor_seleccion+" name=nombre_"+nombre_campo+">");
        Salida.println("<input type=hidden value="+valor_campo+"
name="+nombre_campo+">");
    }

    }
    else
    {

        if (tipo_campo.equals("fecha"))
        Salida.print("<td class=tdfont>"+rotulo_campo+"<td><input type=text size=10
value="+valor_campo+" name="+nombre_campo+" onblur='ValidFecha(this)'>");

        if (tipo_campo.equals("texto"))
        Salida.println("<td class=tdfont>"+rotulo_campo+"<td><textarea rows=3 cols=30
name="+nombre_campo+">"+valor_campo+"</textarea>");

        if (tipo_campo.equals("entero"))
        Salida.print("<td class=tdfont>"+rotulo_campo+"<td><input type=text
size="+long_campo+" value="+valor_campo+" name="+nombre_campo+" onblur='Enteros(this);'
onkeypress='Numeros(this);>");

        if (tipo_campo.equals("numerico"))
        Salida.print("<td class=tdfont>"+rotulo_campo+"<td><input type=text
size="+long_campo+" value="+valor_campo+" name="+nombre_campo+" onblur='Decimales(this);'
onkeypress='Numeros(this);>");

        if (tipo_campo.equals("caracter"))
        Salida.print("<td class=tdfont>"+rotulo_campo+"<td><input type=text
size="+long_campo+" value="+valor_campo+" name="+nombre_campo+">");

        if (tipo_campo.equals("seleccion"))
        {
        Salida.print("<td class=tdfont>"+rotulo_campo+"<td><select name="+nombre_campo+"
size=1>");

        con="select * from "+llave_campo;

        ResultadoSQL3=InstruccionSQL3.executeQuery(con);
        while (ResultadoSQL3.next())
        {
            val1=ResultadoSQL3.getString(1);
            val2=ResultadoSQL3.getString(2);
            if(valor_campo.equals(val1))
            {
                Salida.println("<option selected
value="+val1+">"+val2+"</option>");
            }
        }
    }

```

```

        else
        {
            Salida.println("<option value="+val1+">"+val2+"</option>");
        }
    }

    Salida.println("</select>");
}

} //FIN DEL ELSE

esllave="F";

} //FIN DEL WHILE
} //FIN DEL IF
else
{

    consulta="select
nombre_campo,rotulo_campo,tipo_campo,long_campo,llave_campo,orden_campo from tablas_campos
where nombre_tabla="+tabla+" order by orden_campo";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);

    Salida.println("<table border=1 class=colortable2 align=center>");

    while(ResultadoSQL.next())
    {
        Salida.println("<tr>");

        nombre_campo=ResultadoSQL.getString(1);
        rotulo_campo=ResultadoSQL.getString(2);
        tipo_campo=ResultadoSQL.getString(3);
        long_campo=ResultadoSQL.getString(4);
        llave_campo=ResultadoSQL.getString(5);
        orden_campo=ResultadoSQL.getString(6);

        for(int l=1;l<nivel;l++)
        {
            if(Requerimiento.getParameter("campo"+l).equals(nombre_campo))
            {
                esllave="T";
                datollave=Requerimiento.getParameter("dato"+l);
            }
        }

        if(esllave.equals("T"))
        {

            if (tipo_campo.equals("fecha"))
                Salida.println("<td class=tdfont>"+rotulo_campo+"<td><input
type=text size=10 readonly name="+nombre_campo+" value="+datollave+">");

            if (tipo_campo.equals("texto"))
                Salida.println("<td class=tdfont>"+rotulo_campo+"<td><textarea
rows=3 cols=30 readonly name="+nombre_campo+">"+datollave+"</textarea>");

```

```

        if (tipo_campo.equals("entero"))
            Salida.println("<td class=tdfont>" + rotulo_campo + "<td><input
type=text size="+long_campo+" readonly name="+nombre_campo+" value="+datollave+">");

        if (tipo_campo.equals("numerico"))
            Salida.println("<td class=tdfont>" + rotulo_campo + "<td><input
type=text size="+long_campo+" readonly name="+nombre_campo+" value="+datollave+">");

        if (tipo_campo.equals("caracter"))
            Salida.println("<td class=tdfont>" + rotulo_campo + "<td><input
type=text size="+long_campo+" readonly name="+nombre_campo+" value="+datollave+">");

        if (tipo_campo.equals("seleccion"))
        {
            con="select * from "+llave_campo+" where
"+nombre_campo+"="+datollave+"";

            ResultadoSQL3=InstruccionSQL3.executeQuery(con);
            ResultadoSQL3.next();
            valor_seleccion=ResultadoSQL3.getString(2);
            Salida.println("<td>" + rotulo_campo + "<td><input type=text
readonly name=nombre_" + nombre_campo + " value=" + valor_seleccion + " size="+long_campo+"></td>");
            Salida.println("<input type=hidden name="+nombre_campo+"
value="+datollave+">");

        }
    }
    else
    {

        if (tipo_campo.equals("fecha"))
            Salida.println("<td class=tdfont>" + rotulo_campo + "<td><input
type=text size=10 name="+nombre_campo+" onblur='ValidFecha(this)' >");

        if (tipo_campo.equals("texto"))
            Salida.println("<td class=tdfont>" + rotulo_campo + "<td><textarea
rows=3 cols=30 name="+nombre_campo+"></textarea>");

        if (tipo_campo.equals("entero"))
            Salida.println("<td class=tdfont>" + rotulo_campo + "<td><input
type=text size="+long_campo+" onblur='ValidNumero(this)' name="+nombre_campo+"
onblur='Enteros(this);' onkeypress='Numeros(this);'>");

        if (tipo_campo.equals("numerico"))
            Salida.println("<td class=tdfont>" + rotulo_campo + "<td><input
type=text size="+long_campo+" onblur='ValidNumero(this)' name="+nombre_campo+"
onblur='Decimales(this);' onkeypress='Numeros(this);'>");

        if (tipo_campo.equals("caracter"))
            Salida.println("<td class=tdfont>" + rotulo_campo + "<td><input
type=text size="+long_campo+" onblur='ValidCaracter(this)' name="+nombre_campo+">");

        if (tipo_campo.equals("seleccion"))
        {

```



```

    }
}

Salida.println("<input type=hidden name=USUARIO value="+Usuario+">");
Salida.println("</table>");

Salida.println("</form>");

if(!(adicionar.equals("T")))
{
    int nivel_i=nivel+1;
    String tabla_hija;

    consulta="select nombre_tabla from tablas where superior_tabla="+tabla+"";
    ResultadoSQL=InstruccionSQL.executeQuery(consulta);
    while(ResultadoSQL.next())
    {
        tabla_hija=ResultadoSQL.getString(1);
        if(!(tabla.equals(tabla_hija)))
        {
            Salida.println("<br>");
            Salida.print("<a
href=/servlet/Administracion?nivel="+nivel_i+"&tabla="+tabla_hija+"&USUARIO="+Usuario);
                for(int i=1;i<=nivel;i++)
                {
                    Salida.print("&campo"+i+"="+Requerimiento.getParameter("campo"+i)+"&dato"+i+"="+Requeri
miento.getParameter("dato"+i));
                }
            Salida.print(">Mantenimiento de la tabla "+tabla_hija+"</a>");
        }
    }
}

}

}

}

catch (Exception Error)
{
    Salida.println("Situación Anormal!");
    Error.printStackTrace(Salida);
}
finally
{
    try
    {
        if (ResultadoSQL!=null) ResultadoSQL.close();
        if (InstruccionSQL!=null) InstruccionSQL.close();
        if (ConexionSQL!=null) ConexionSQL.close();
        if (ResultadoSQL2!=null) ResultadoSQL2.close();
        if (InstruccionSQL2!=null) InstruccionSQL2.close();
        if (ConexionSQL2!=null) ConexionSQL2.close();
        if (ResultadoSQL3!=null) ResultadoSQL3.close();
    }
}

```

```

    if (InstruccionSQL3!=null) InstruccionSQL3.close();
    if (ConexionSQL3!=null) ConexionSQL3.close();
    }
    catch (Exception Error) {}
    }
}
}

```

### *AdministracionActualizacion*

```

import Usuario;
import Menu;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class Administracion_actualizacion extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Administración</title>");
        Salida.println("<link href=\"/HTML/include/ourstyle.css\" rel=\"stylesheet\" type=\"text/css\">");
        Salida.println("<SCRIPT language=JavaScript src=/HTML/include/ourScript.js'
type=text/javascript></SCRIPT>");
        Salida.println("</head>");
        Salida.println("<body>");

        String tabla=Requerimiento.getParameter("tabla");

        String adicionar=Requerimiento.getParameter("adicionar");

        // Ejecutando Proceso Principal
        Proceso(Salida,tabla,adicionar,Requerimiento);

        // Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
        Salida.println("</body>");
        Salida.println("</html>");
        Salida.flush();
    }
}

```

```

        Salida.close();
    }

private void Proceso(PrintWriter Salida,String tabla,String adicionar,HttpServletRequest Requerimiento)
{
    String Usuario= Requerimiento.getParameter("USUARIO");

    // Inicializando Variables para Conexión
    Connection ConexionSQL=null;
    Statement InstruccionSQL=null;
    ResultSet ResultadoSQL=null;
    Connection ConexionSQL2=null;
    Statement InstruccionSQL2=null;
    ResultSet ResultadoSQL2=null;
    Connection ConexionSQL3=null;
    Statement InstruccionSQL3=null;
    ResultSet ResultadoSQL3=null;

    try
    {
        // Iniciando Conexión

        Usuario MIUSUARIO = new Usuario();
        String Driver = MIUSUARIO.driver;
        String DB = MIUSUARIO.myBD;
        String Usr = MIUSUARIO.usuario;
        String Pass= MIUSUARIO.password;
        String dirhtml= MIUSUARIO.DirHTML;

        Menu DoMenu = new Menu();

        DoMenu.CREARMENU(Usuario,Salida,Driver,Usr,DB,Pass);

        DoMenu.Encabezado(Salida);

        Class.forName(Driver);
        ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
        InstruccionSQL=ConexionSQL.createStatement();
        ConexionSQL2 = DriverManager.getConnection(DB,Usr,Pass);
        InstruccionSQL2=ConexionSQL2.createStatement();
        ConexionSQL3 = DriverManager.getConnection(DB,Usr,Pass);
        InstruccionSQL3=ConexionSQL3.createStatement();

        String consulta;
        int cuantos;
        String parametro;
        String nombre_campo;
        String tipo_campo;
        String insertar="insert into "+tabla+" values(";
        String actualizar="update "+tabla+" set ";
        String eliminar="delete "+tabla;
        String accion=Requerimiento.getParameter("accion");
        String condicion="";
    }
}

```

```
String nivel_s=Requerimiento.getParameter("nivel");
int nivel=Integer.parseInt(nivel_s);
```

```
Salida.println("<p><center><font size=2 face=verdana><b>Administración de la tabla
"+tabla+"</b></font></center></p>");
```

```
consulta="select count(*) from tablas_campos where nombre_tabla='"+tabla+"'";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
cuantos=ResultadoSQL.getInt(1);
```

```
consulta="select nombre_campo,tipo_campo from tablas_campos where nombre_tabla='"+tabla+"' order
by orden_campo";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
```

```
for (int i=1;i<(cuantos+1);i++)
{
ResultadoSQL.next();
nombre_campo=ResultadoSQL.getString(1);
tipo_campo=ResultadoSQL.getString(2);
parametro=Requerimiento.getParameter(nombre_campo);
```

```
    if(adicionar.equals("T"))
    {
        if(tipo_campo.equals("numerico"))
        {
            if(i==cuantos)
                insertar=insertar+parametro+");";
            else
                insertar=insertar+parametro+",";";
        }
        else
        {
            if(i==cuantos)
                insertar=insertar+""+parametro+");";
            else
                insertar=insertar+""+parametro+",";";
        }
    }
    else
    {
        if(accion.equals("Modificar"))
        {
            if(tipo_campo.equals("numerico"))
            {
                if(i==cuantos)
                    actualizar=actualizar+nombre_campo+"="+parametro;
                else
                    actualizar=actualizar+nombre_campo+"="+parametro+",";";
            }
            else
            {
```

```

        {
            if(i==cuantos)
                actualizar=actualizar+nombre_campo+"="+parametro+"";
            else
                actualizar=actualizar+nombre_campo+"="+parametro+"";
        }

    }

    for (int j=1;j<=nivel;j++)
    {
        if(j==1)
            condicion=Requerimiento.getParameter("campo"+j)+"="+Requerimiento.getParameter("dato"+j)+"";
        else
            condicion=condicion + "and
"+Requerimiento.getParameter("campo"+j)+"="+Requerimiento.getParameter("dato"+j)+"";
    }

}

}

if(adicionar.equals("T"))
{

    InstruccionSQL.executeUpdate(insertar);
    Salida.println("El Registro ha sido agregado");
}
else
{
    if(accion.equals("Modificar"))
    {
        actualizar=actualizar+" where "+condicion;
        InstruccionSQL.executeUpdate(actualizar);
        Salida.println("<center>El Registro ha sido modificado</center>");
    }
    else
    {
        eliminar=eliminar+" where "+condicion;
        Salida.println("<br>eliminar"+eliminar);
        InstruccionSQL.executeUpdate(eliminar);
        Salida.println("<center>El Registro ha sido eliminado</center>");
    }
}
}
catch (Exception Error)
{
    Salida.println("Situación Anormal!");
    Error.printStackTrace(Salida);
}
finally
{
    try

```

```

    {
    if (ResultadoSQL!=null) ResultadoSQL.close();
    if (InstruccionSQL!=null) InstruccionSQL.close();
    if (ConexionSQL!=null) ConexionSQL.close();
    if (ResultadoSQL2!=null) ResultadoSQL2.close();
    if (InstruccionSQL2!=null) InstruccionSQL2.close();
    if (ConexionSQL2!=null) ConexionSQL2.close();
    if (ResultadoSQL3!=null) ResultadoSQL3.close();
    if (InstruccionSQL3!=null) InstruccionSQL3.close();
    if (ConexionSQL3!=null) ConexionSQL3.close();
    }
    catch (Exception Error) {}
    }
}

    public void doPost(HttpServletRequest Requerimiento,
        HttpServletResponse Respuesta)
        throws ServletException, java.io.IOException
    {
        doGet(Requerimiento,Respuesta);
    }
}

```

## **Resultados en Mapa** (mapaReportes.java)

### *Filtrado de datos y generación de mapa*

Este servlet genera inicialmente un formulario para realizar selección de criterios de filtrado, posteriormente en base a parametros al dar click en el botón de mostrar genera el resultado en mapa.

```

import Usuario;
import Menu;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class mapaReportes extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());
    }
}

```

```

        Usuario MU = new Usuario();
        String dirhtml= MU.DirHTML;

// Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
Salida.println("<html>");
Salida.println("<head>");
Salida.println("<title>Mapa</title>");
Salida.println("<LINK href='\"/HTML/include/ourstyle.css\"' type=text/css rel=stylesheet>");
Salida.println("<SCRIPT          language=JavaScript          src='/HTML/include/ourScript.js'
type=text/javascript></SCRIPT>");
    if(Requerimiento.getParameter("Filtrar").equals("2"))
    {
        Salida.println("<SCRIPT          language=JavaScript          src='\"+dirhtml+\"mapita_svg_reportes.js'
type=text/javascript></SCRIPT>");
        Salida.println("</head>");
        Salida.println("<BODY topmargin=0 onload=iniciarMapa()>");
    }
    else
    {
        Salida.println("</head>");
        Salida.println("<BODY topmargin=0>");
    }

String Usuario=Requerimiento.getParameter("USUARIO");

// Ejecutando Proceso Principal
Proceso(Salida,Usuario,Requerimiento);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida

Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

private void Proceso(PrintWriter Salida,String Usuario, HttpServletRequest Requerimiento)
{

String Filtrar=Requerimiento.getParameter("Filtrar");
String Instituciones=Requerimiento.getParameter("Instituciones");
String Anio=Requerimiento.getParameter("Anios");
String Consulta="";
String Consulta_Ins="";
String Consulta_Tip="";
String Consulta_Dep="";
String Consulta_Mun="";
String Consulta_Ani="";

String llenar="";
String codigo="";
String codigo1="";
String codigo2="";
String codigo_dpt="";

```

```

String tipo_usr="0";
String nombre_institucion="";
String temp="";

// Inicializando Variables para Conexión
Connection ConexionSQL=null;
Statement InstruccionSQL=null;
ResultSet ResultadoSQL=null;

try
{

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    ConexionSQL=DriverManager.getConnection(DB,Usr,Pass);

    Menu DoMenu = new Menu();

    DoMenu.CREARMENU(Usuario,Salida,Driver,Usr,DB,Pass);

    DoMenu.Encabezado(Salida);

    InstruccionSQL=ConexionSQL.createStatement();

    if (Filtrar.equals("1"))
    {
        codigo=Requerimiento.getParameter("Departamentos");
        codigo1=Requerimiento.getParameter("Instituciones");
        codigo2=Requerimiento.getParameter("TipoObra");

        if (!(codigo.equals("-1")))
        {
            Consulta_Ins="exec RetornarInstituciones d,"+codigo;
            Consulta_Tip="exec RetornarTipoObra d,"+codigo;
            Consulta_Dep="exec RetornarDepartamentos N,0";
            Consulta_Mun="exec RetornarMunicipios d,"+codigo;
            Consulta_Ani="exec RetornarAnios d,"+codigo;
        }
        else
        {
            if (!(codigo1.equals("-1")))
            {
                Consulta_Dep="exec RetornarDepartamentos i,"+codigo1;
                Consulta_Tip="exec RetornarTipoObra i,"+codigo1;
                Consulta_Ins="exec RetornarInstituciones N,0";
                Consulta_Mun="exec RetornarMunicipios i,"+codigo1;
                Consulta_Ani="exec RetornarAnios i,"+codigo1;
            }
            else

```

```

        {
            Consulta_Ins="exec RetornarInstituciones t,"+codigo2;
            Consulta_Dep="exec RetornarDepartamentos t,"+codigo2;
            Consulta_Tip="exec RetornarTipoObra N,0";
            Consulta_Mun="exec RetornarMunicipios t,"+codigo2;
            Consulta_Ani="exec RetornarAnios t,"+codigo2;
        }
    }
}
else
{
    Consulta_Ins="exec RetornarInstituciones N,0";
    Consulta_Tip="exec RetornarTipoObra N,0";
    Consulta_Dep="exec RetornarDepartamentos N,0";
    Consulta_Mun="exec RetornarMunicipios N,0";
    Consulta_Ani="exec RetornarAnios N,0";
}

Consulta_Ani="exec RetornarAnios N,0";

Consulta="Select * From sisusuper Where Codigo_responsable=""+Usuario+""";
ResultadoSQL=InstruccionSQL.executeQuery(Consulta);

if(ResultadoSQL.next())
{
    temp=ResultadoSQL.getString(1).trim();

    if (!(temp.equals("ADMIN")) && !(temp.equals("CONSULSETE")))
    {
        tipo_usr="1";
    }
}

if (tipo_usr.equals("1"))
{
    Consulta="select i.codigo_institucion, i.nombre_institucion from Responsables as r,
instituciones as i where r.codigo_institucion=i.codigo_institucion and
r.codigo_responsable=""+Usuario+""";
    ResultadoSQL=InstruccionSQL.executeQuery(Consulta);

    if(ResultadoSQL.next())
    {
        codigo1=ResultadoSQL.getString(1);
        nombre_institucion=ResultadoSQL.getString(2);
        Consulta_Ins="exec RetornarInstituciones N,"+codigo1+",2";
        Consulta_Tip="exec RetornarTipoObra i,"+codigo1;
        Consulta_Dep="exec RetornarDepartamentos i,"+codigo1;
    }
}

if (!(Filtrar.equals("2")))
{

```

```

        Salida.println("<form name=consulta method=post action='/servlet/mapaReportes?'\>");
        Salida.println("<table class=colortable align=center>");
        Salida.println(" <tr><td colspan=2 align=center class=hcolor>Parámetros de
Consulta</tr>");
        Salida.println(" <tr>");
        Salida.println(" <td class=tdfont>Instituci&oacute;n&nbsp;&nbsp;&nbsp;</td><td>");

        if(tipo_usr.equals("1"))
        {
            Salida.println("<select name=Instituciones>");
            Salida.println("<option value="+codigo1+"> "+nombre_institucion+"</option>");
        }
        else
        {
            if (!(Filtrar.equals("1")))
            {
                Salida.println("<select name=Instituciones
onChange='\&quot;cambiar("&quot;+Usuario+"&quot;)\&quot;>");
            }
            else
            {
                if(!(codigo1.equals("-1")))
                {
                    Salida.println("<select name=Instituciones
onChange='\&quot;cambiar("&quot;+Usuario+"&quot;)\&quot;>");
                }
                else
                {
                    Salida.println("<select name=Instituciones>");
                }
            }
            Salida.println("<option value=999> - Seleccione - </option>");

//          Consulta="Select Count(*) From Instituciones";
//          ResultadoSQL=InstruccionSQL.executeQuery(Consulta_Ins+",0");
//          ResultadoSQL.next();
//          if (ResultadoSQL.getInt(1)!=0)
//          {
                Consulta="Select * From Instituciones";
                ResultadoSQL=InstruccionSQL.executeQuery(Consulta_Ins+",1");

                while(ResultadoSQL.next())
                {
                    llenar=ResultadoSQL.getString(1);

                    if(llenar.equals(codigo1))
                    {
                        Salida.println("<option value="+llenar+"
Selected>"+ResultadoSQL.getString(2)+"</option>");
                    }
                    else
                    {
                        Salida.println("<option
value="+llenar+">"+ResultadoSQL.getString(2)+"</option>");
                    }
                }
            }
        }
    }
}

```

```

        }
    }

    Salida.println("</select>");
}

    Salida.println("        </td>");
    Salida.println("    </tr>");
    Salida.println(" <tr>");
    Salida.println("        <td class=tdfont>Tipos de Obra&nbsp;</td><td>");

    if (!Filtrar.equals("1"))
    {
        Salida.println("<select                                name=TipoObra
onChange=\"cambiar(\"+Usuario+\")\">");
    }
    else
    {
        Salida.println("<select name=TipoObra>");
    }
    Salida.println("<option value=-1> - Seleccione - </option>");

//        Consulta="Select Count(*) From unidades_medida";

        ResultadoSQL=InstruccionSQL.executeQuery(Consulta_Tip+",0");
        ResultadoSQL.next();
        if (ResultadoSQL.getInt(1)!=0)
        {
//            Consulta="Select * From unidades_medida";

            ResultadoSQL=InstruccionSQL.executeQuery(Consulta_Tip+",1");

            while(ResultadoSQL.next())
            {
                llenar=ResultadoSQL.getString(1);

                if(llenar.equals(codigo2))
                {
                    Salida.println("<option                                value="+llenar+"
Selected>"+ResultadoSQL.getString(2)+"</option>");
                }
                else
                {
                    Salida.println("<option
value="+llenar+">"+ResultadoSQL.getString(2)+"</option>");
                }
            }
        }

        Salida.println("</select>");
        Salida.println("        </td>");
        Salida.println("    </tr>");
        Salida.println(" <tr>");
        Salida.println("        <td class=tdfont>Etapas&nbsp;</td><td>");
        Salida.println("<select name=Etapas>");
        Salida.println("<option value=-1> - Seleccione - </option>");

```

```

Consulta="Select Count(*) From Etapas";
ResultadoSQL=InstruccionSQL.executeQuery(Consulta);
ResultadoSQL.next();
if (ResultadoSQL.getInt(1)!=0)
{
    Consulta="Select * From Etapas";
    ResultadoSQL=InstruccionSQL.executeQuery(Consulta);

    while(ResultadoSQL.next())
    {
        Salida.println("<option
value="+ResultadoSQL.getString(1)+">" +ResultadoSQL.getString(2)+"</option>");
    }
}
Salida.println("</select>");
Salida.println("        </td>");
Salida.println("    </tr>");
Salida.println("    <tr>");
Salida.println("        <td class=tdfont>Departamentos&nbsp;  </td><td>");

if (!(Filtrar.equals("1")))
{
    Salida.println("<select                                name=Departamentos
onChange=\cambiar(\""+Usuario+"\")\">");
}
else
{
    Salida.println("<select                                name=Departamentos
onChange=\cambiar(\""+Usuario+"\")\">");
}

Salida.println("<option value=-1> - Seleccione - </option>");

//
Consulta="Select Count(*) From DEPARTAMENTOS";
ResultadoSQL=InstruccionSQL.executeQuery(Consulta_Dep+"0");
ResultadoSQL.next();
if (ResultadoSQL.getInt(1)!=0)
{
//
    Consulta="Select * From DEPARTAMENTOS";
    ResultadoSQL=InstruccionSQL.executeQuery(Consulta_Dep+"1");

    while(ResultadoSQL.next())
    {

        llenar=ResultadoSQL.getString(1);

        if(llenar.equals(codigo))
        {
            Salida.println("<option                                value="+llenar+"
Selected>" +ResultadoSQL.getString(2)+"</option>");
        }
        else
        {
            Salida.println("<option
value="+llenar+">" +ResultadoSQL.getString(2)+"</option>");
        }
    }
}
}

```



```

value=Cancelar
onClick="\document.location=\/servlet/mapaReportes?USUARIO="+Usuario+"&Filtrar=0\">");
    Salida.println("<input type=hidden name=USUARIO value="+Usuario+">");
    Salida.println("<input type=hidden name=Filtrar value=2>");
    Salida.println("</td></tr>");
    Salida.println("</table>");
    Salida.println("</form>");
}
else
{
    if(Filtrar.equals("2"))
    {
        if ((Requerimiento.getParameter("Instituciones")).equals("999"))
        {
            nombre_institucion=" Reporte de Obras a Nivel Nacional ";
        }
        else
        {
            Consulta="select nombre_institucion from Instituciones Where
codigo_institucion="+Requerimiento.getParameter("Instituciones");
ResultadoSQL=InstruccionSQL.executeQuery(Consulta);
if (ResultadoSQL.next())
{
    nombre_institucion=" <center>Reporte de Obras <br>
"+ResultadoSQL.getString(1)+"</center>";
}
}

            Consulta="Instituciones="
Requerimiento.getParameter("Instituciones");
            Consulta=Consulta
"&TipoObra="+Requerimiento.getParameter("TipoObra");
            Consulta=Consulta
"&Etapas="+Requerimiento.getParameter("Etapas");
            Consulta=Consulta
"&Departamentos="+Requerimiento.getParameter("Departamentos");
            Consulta=Consulta
"&Municipios="+Requerimiento.getParameter("Municipios");
            Consulta=Consulta
"&Anios="+Requerimiento.getParameter("Anios");
//Salida.println("exec                                RetornarObras
"+Requerimiento.getParameter("Instituciones")+",""+Requerimiento.getParameter("TipoObra").trim()+","
+Requerimiento.getParameter("Departamentos")+",""+Requerimiento.getParameter("Etapas")+",""+Requeri
miento.getParameter("Municipios")+",""+Requerimiento.getParameter("Anios")+","0");

    Salida.println("<table border=0 cellpadding=0 cellspacing=0 width=760 align=center>");
    Salida.println("<tr>");
    Salida.println("<td align=center>");
    Salida.println("                <HEADER><h1>"+nombre_institucion+"</h1></HEADER>");
    Salida.println("</td>");
    Salida.println("</tr>");
    Salida.println("<tr>");
    Salida.println("<td align=center>");

```

```

        Salida.println("<EMBED
SRC='/servlet/GeneraMapaSVGReportes?USUARIO="+Usuario+"&"+Consulta+"          HEIGHT=\\"570\\"
WIDTH=\\"900\\" TYPE=\"image/svg+xml\" name=all_map>");
        Salida.println(" </td>");
        Salida.println("</tr>");
        Salida.println("<tr>");
        Salida.println(" <td align=center valign=middle>");
        Salida.println("          <FORM name=proyectos>");
        Salida.println("          <table border=0 cellpadding=0 cellspacing=1 align=left width=100%
bgcolor=\\"#BBBBBB\">");
        Salida.println(" <tr></td>");
        Salida.println("          <input type=hidden name='id_tipo_obra' value=\">");
        Salida.println("          <input type=hidden name='id_obra' value=\">");
        Salida.println("          <input type=hidden name='pos_obra_x' value=\">");
        Salida.println("          <input type=hidden name='pos_obra_y' value=\">");
        Salida.println("          <input type=hidden name='nom_obra' value=\">");
        Salida.println("          <input type=hidden name='id_canton' value=\">");
        Salida.println("          <input type=hidden name='cod_responsable' value=\""+Usuario+">");
        Salida.println("          </form>");
        Salida.println(" </td></tr>");
        Salida.println("        </table>");
        Salida.println(" </td>");
        Salida.println("</tr>");
        Salida.println("</table>");
        Salida.println("<script language=javascript>");
        Salida.println("institucion=\""+Requerimiento.getParameter("Instituciones")+";");
        Salida.println("anio=\""+Requerimiento.getParameter("Anios")+";");
        Salida.println("parametro='1';");
        Salida.println("usuario=\""+Usuario+";");
        Salida.println("</script>");
    }
}
}
catch (Exception Error)
{
    Salida.println("Situación Anormal!");
    Error.printStackTrace(Salida);
}
finally
{
    try
    {
        if (ResultadoSQL!=null) ResultadoSQL.close();
        if (InstruccionSQL!=null) InstruccionSQL.close();
        if (ConexionSQL!=null) ConexionSQL.close();
    }
    catch (Exception Error) {}
}
}

public void doPost(HttpServletRequest SOLICITUD,
    HttpServletResponse RESPUESTA)
    throws ServletException, java.io.IOException
{

```

```

        doGet(SOLICITUD,RESPUESTA);
    }

}

```

## Reportes de avance mostrados en el mapa

### *Reportes de avance en tablas*

Los reportes en formato de tabla que acompañan los reportes de avance en el mapa, son generados haciendo uso del servlet Reportes, el cual haciendo uso de la tabla Reportes, recibe los parámetros necesarios y ejecuta el procedimiento almacenado correspondiente al reporte que se quiere mostrar, el cual a su vez recibe los parámetros necesarios para realizar la consulta.

Los 6 reportes de tabla que acompañan al mapa son generados con el siguiente servlet:

```

import Usuario;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class Reportes extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento,HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Leyendo Parámetros
        String Codigo=Requerimiento.getParameter("Codigo");
        String USUARIO=Requerimiento.getParameter("USUARIO");
        String parametro=Requerimiento.getParameter("parametro");
        String institucion=Requerimiento.getParameter("institucion");
        String anio=Requerimiento.getParameter("anio");

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Reportes</title>");
        Salida.println("<LINK href=\\'/HTML/include/ourstyle.css\\' type=text/css rel=stylesheet>");
        Salida.println("</head>");
        Salida.println("<body style='font-family: Verdana; font-size: 6pt'>");

        // Ejecutando Proceso Principal
        Proceso(Salida,
            Codigo,
            USUARIO,

```

```

        parametro,
        institucion,
        anio);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

public void init(ServletConfig cfg) throws ServletException { super.init(cfg); }

public void destroy() { super.destroy(); }

private void Proceso(PrintWriter Salida,
        String Codigo,
        String USUARIO,
        String parametro,
        String institucion,
        String anio)
{

        // Inicializando Variables para Conexión con ODBC
        Connection ConexionSQL=null;
        Statement InstruccionSQL=null;
        ResultSet ResultadoSQL=null;
        Connection ConexionSQL2=null;
        Statement InstruccionSQL2=null;
        ResultSet ResultadoSQL2=null;

try
{

        Usuario MIUSUARIO = new Usuario();
        String Driver = MIUSUARIO.driver;
        String DB = MIUSUARIO.myBD;
        String Usr = MIUSUARIO.usuario;
        String Pass= MIUSUARIO.password;
        String dirhtml= MIUSUARIO.DirHTML;

        ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
        InstruccionSQL=ConexionSQL.createStatement();
        ConexionSQL2 = DriverManager.getConnection(DB,Usr,Pass);
        InstruccionSQL2=ConexionSQL2.createStatement();

        Menu DoMenu = new Menu();

        DoMenu.CREARMENU(USUARIO,Salida,Driver,Usr,DB,Pass);

```

```

DoMenu.Encabezado(Salida);

// Obteniendo parámetros desde la tabla SISREP
ResultadoSQL=InstruccionSQL.executeQuery("select
NOM_CUA,SEL_CUA,TIP_CUA,ENC_CUA,ESQ_CUA,DAT_CUA,CO1_CUA,CO2_CUA      from
REPORTES where COD_CUA='"+Codigo+"'");
ResultadoSQL.next();
String Nombre=ResultadoSQL.getString(1);
String Datos=ResultadoSQL.getString(2);
String Tipo=ResultadoSQL.getString(3);
String Titulos=ResultadoSQL.getString(4);
String Esquina=ResultadoSQL.getString(5);
String NumeroSubTitulos=ResultadoSQL.getString(6);
String SubTitulo1=ResultadoSQL.getString(7);
String SubTitulo2=ResultadoSQL.getString(8);

Datos=Datos+" "+institucion+" "," "+parametro+" "," +anio+" "," +USUARIO;

if (Tipo.equals("T"))
{

// TIPO TABLA

// Ejecutando Select Principal
ResultadoSQL = InstruccionSQL.executeQuery(Datos);

// Creando Encabezado de Tabla; Titulos y SUBTitulos
Salida.println("<h1 align=center>"+Nombre+"</h1><hr>");

ResultSetMetaData MetaDatosSQL=ResultadoSQL.getMetaData();
int NumeroTitulos=MetaDatosSQL.getColumnCount();
Salida.println("<table align=center border=1 class=colortable2>");
Salida.println("<tr>");
for (int i=0;i<NumeroTitulos;i++)
{
Salida.println("<td class=hcolor>"+MetaDatosSQL.getColumnLabel(i+1).trim());
}

// LLamando al Formateador de Tabla
Tabla(ResultadoSQL, Salida, NumeroTitulos);

// Cerrando Tabla
Salida.println("</table>");
}
else
{

// TIPO CUADRO

// Creando Encabezado de Tabla; Titulos y SUBTitulos
Salida.println("<h1 align=center>"+Nombre+"</h1><hr><br>");
ResultadoSQL=InstruccionSQL.executeQuery(Titulos);

```

```

int NumeroTitulos=0;
Salida.println("<table align=center border=1 cellpadding=0 cellspacing=0>");
Salida.println("<tr bgcolor=red><th rowspan="+NumeroSubTitulos+"><font color=FFFFFF
size=2>"+Esquina);
while (ResultadoSQL.next())
{
NumeroTitulos++;
Salida.println("<th colspan="+NumeroSubTitulos+"><font color=FFFFFF
size=2>"+ResultadoSQL.getString(1).trim());
}
if (NumeroSubTitulos.equals("2"))
{
Salida.println("<tr bgcolor=blue>");
for (int i=0;i<NumeroTitulos;i++)
{
Salida.println("<th><font color=FFFFFF size=1>"+SubTitulo1);
Salida.println("<th><font color=FFFFFF size=1>"+SubTitulo2);
}
}

// Ejecutando Select Principal
ResultadoSQL = InstruccionSQL.executeQuery(Datos);

// LLamando al Formateador de Cuadro
Cuadro(ResultadoSQL, Salida, NumeroTitulos, NumeroSubTitulos);

// Cerrando Tabla
Salida.println("</table>");

}
Salida.println("<p><center><a href=# class=linkblanco onClick='history.back();'>&lt;--
&nbsp;&nbsp;&nbsp;Regresar</a></center></p><br>");

}
catch (Exception Error)
{
Salida.println("Situación Anormal!");
Error.printStackTrace(Salida);
}
finally
{
try
{
if (ResultadoSQL!=null) ResultadoSQL.close();
if (InstruccionSQL!=null) InstruccionSQL.close();
if (ConexionSQL!=null) ConexionSQL.close();
if (ResultadoSQL2!=null) ResultadoSQL2.close();
if (InstruccionSQL2!=null) InstruccionSQL2.close();
if (ConexionSQL2!=null) ConexionSQL2.close(); }

catch (Exception Error) {}
}
}

```

```

private void Tabla(ResultSet ResultadoSQL,
                  PrintWriter Salida,
                  int NumeroTitulos)
throws Exception
{
    // Procesando Datos del Cuadro
    String Columna1="";
    while (ResultadoSQL.next())
    {
        Columna1=ResultadoSQL.getString(1);
        if (Columna1.startsWith("TOTAL"))
        {
            Salida.println("<tr align=right >");
            Salida.println("<td class=tdfont>" +Columna1);
            for (int i=2;i<=NumeroTitulos;i++)
            {
                Salida.println("<td>" + ResultadoSQL.getString(i));
            }
        }
        else
        {
            Salida.println("<tr align=left>");
            Salida.println("<td class=tdfont>" +Columna1);
            for (int i=2;i<=NumeroTitulos;i++)
            {
                Salida.println("<td><font size=2>" +ResultadoSQL.getString(i));
            }
        }
    }
}

private void Cuadro(ResultSet ResultadoSQL,
                    PrintWriter Salida,
                    int NumeroTitulos,
                    String NumeroSubTitulos)
throws Exception
{
    // Procesando Datos del Cuadro
    String Columna1 = "";
    while (ResultadoSQL.next())
    {
        Columna1=ResultadoSQL.getString(1).trim();
        if (Columna1.startsWith("TOTAL"))
        {
            Salida.println("<tr align=right bgcolor=black>");
            Salida.println("<th><font color=FFFFFF size=2>" +Columna1);

            Salida.println("<th><font color=FFFFFF size=2>" +ResultadoSQL.getString(3).trim());
            if (NumeroSubTitulos.equals("2"))
                Salida.println("<th><font color=FFFFFF size=2>" +ResultadoSQL.getString(4).trim());
            for (int i=1;i<NumeroTitulos;i++)
            {
                ResultadoSQL.next();
                Salida.println("<th><font color=FFFFFF size=1>" + ResultadoSQL.getString(3).trim());
            }
        }
    }
}

```

```

        if (NumeroSubTitulos.equals("2")) Salida.println("<th><font color=FFFFFF
size=2>" + ResultadoSQL.getString(4).trim());
    }
}
else
{
    Salida.println("<tr align=right>");
    Salida.println("<td><font size=1>" + Columna1);
    Salida.println("<td><font size=1>" + ResultadoSQL.getString(3).trim());
    if (NumeroSubTitulos.equals("2")) Salida.println("<td><font
size=2>" + ResultadoSQL.getString(4).trim());
    for (int i=1; i<NumeroTitulos; i++)
    {
        ResultadoSQL.next();
        Salida.println("<td><font size=1>" + ResultadoSQL.getString(3).trim());
        if (NumeroSubTitulos.equals("2")) Salida.println("<td><font
size=2>" + ResultadoSQL.getString(4).trim());
    }
}
}
}
}
}
}

```

### *Reportes consolidados*

Para la generación de los reportes consolidados, se crean dos servlets desarrollados especialmente para la generación de estos reportes debido a que cada uno de ellos tiene sus propios parámetros y no se pueden homogenizar. El servlet que realiza esta actividad es el siguiente:

```

import Usuario;
import Menu;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class ReportesConsolidados extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento, HttpServletResponse Respuesta)
    throws ServletException, IOException
    {

        // Leyendo Parámetros

        String USUARIO=Requerimiento.getParameter("USUARIO");
        String ANIO=Requerimiento.getParameter("ANIO");
        String INSTITUCION=Requerimiento.getParameter("INSTITUCION");

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
    }
}

```

```

PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

// Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
Salida.println("<html>");
Salida.println("<head>");
Salida.println("<title>Sistema de Información de Proyectos</title>");
Salida.println("<LINK href='\"/HTML/include/ourstyle.css\"' type=text/css rel=stylesheet>");
Salida.println("<SCRIPT          language=JavaScript          src='/HTML/include/ourScript.js'
type=text/javascript></SCRIPT>");
Salida.println("</head>");
Salida.println("<body>");

// Ejecutando Proceso Principal
Proceso(Salida,
        USUARIO,
        ANIO,
        INSTITUCION);

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

public void init(ServletConfig cfg) throws ServletException { super.init(cfg); }

public void destroy() { super.destroy(); }

private void Proceso(PrintWriter Salida,
        String USUARIO,
        String ANIO,
        String INSTITUCION)
{
        // Inicializando Variables para Conexión con ODBC
        Connection ConexionSQL=null;
        Statement InstruccionSQL=null;
        ResultSet ResultadoSQL=null;
        Connection ConexionSQL2=null;
        Statement InstruccionSQL2=null;
        ResultSet ResultadoSQL2=null;

try
{

        Usuario MIUSUARIO = new Usuario();
        String Driver = MIUSUARIO.driver;
        String DB = MIUSUARIO.myBD;
        String Usr = MIUSUARIO.usuario;
        String Pass= MIUSUARIO.password;
        String dirhtml= MIUSUARIO.DirHTML;

```

```

ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
InstruccionSQL=ConexionSQL.createStatement();
ConexionSQL2 = DriverManager.getConnection(DB,Usr,Pass);
InstruccionSQL2=ConexionSQL2.createStatement();

Menu DoMenu = new Menu();

DoMenu.CREARMENU(USUARIO,Salida,Driver,Usr,DB,Pass);

DoMenu.Encabezado(Salida);

String institucion="";
int NumeroTitulos=0;
ResultSetMetaData MetaDatosSQL;
String consulta="";

if(INSTITUCION==null)
{
consulta="select cod_per from sisusuper where codigo_responsable="+USUARIO+"";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
String perfil=ResultadoSQL.getString(1);

if ((perfil.equals("CONSULSETE"))||(perfil.equals("ADMIN"))) institucion="999";
else
{
consulta="select codigo_institucion from responsables where codigo_responsable="+USUARIO+"";
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
institucion=ResultadoSQL.getString(1);
}
}
else institucion=INSTITUCION;

Salida.println("<center><font size=2 face=verdana><b>Reportes Consolidados <br>");
if(institucion.equals("999")) Salida.println("A nivel nacional");
else
{
consulta="select nombre_institucion from instituciones where codigo_institucion="+institucion;
ResultadoSQL=InstruccionSQL.executeQuery(consulta);
ResultadoSQL.next();
Salida.println(ResultadoSQL.getString(1));
}
Salida.println("<br>Año:" + ANIO + "<p>&nbsp;</p>");
Salida.println("<table border=1 width=80% cellpadding=5 align=center bgcolor=#BBBBBB>");

Salida.println("<tr>");

//REPORTE DE PROGRAMACION ANUAL-MENSUAL

```

```
Salida.println("<td align=center><b>Programación y Ejecución Mensual</b><br><br>");
```

```
Salida.println("<table class=colortable>");
```

```
ResultadoSQL=InstruccionSQL.executeQuery("exec  
"+institucion+", "+ANIO);
```

ProgramacionEjecucionAnual

```
MetaDatosSQL=ResultadoSQL.getMetaData();
```

```
NumeroTitulos=MetaDatosSQL.getColumnCount();
```

```
Salida.println("<table align=center class=colortable2>");
```

```
Salida.println("<tr >");
```

```
for (int i=0;i<NumeroTitulos;i++)
```

```
{
```

```
Salida.println("<td class=hcolor>"+MetaDatosSQL.getColumnLabel(i+1).trim());
```

```
}
```

```
Tabla(ResultadoSQL, Salida, NumeroTitulos);
```

```
Salida.println("</table>");
```

```
Salida.println("</td>");
```

```
//REPORTE POR ETAPAS
```

```
Salida.println("<td align=center><b>Programación y Ejecución Por Etapa</b><br><br>");
```

```
Salida.println("<table class=colortable>");
```

```
ResultadoSQL=InstruccionSQL.executeQuery("exec  
"+USUARIO+", "+institucion+", "+ANIO);
```

ProgramacionEjecucionEtapa

```
MetaDatosSQL=ResultadoSQL.getMetaData();
```

```
NumeroTitulos=MetaDatosSQL.getColumnCount();
```

```
Salida.println("<table align=center class=colortable2>");
```

```
Salida.println("<tr>");
```

```
for (int i=0;i<NumeroTitulos;i++)
```

```
{
```

```
Salida.println("<td class=hcolor>"+MetaDatosSQL.getColumnLabel(i+1).trim());
```

```
}
```

```
Tabla(ResultadoSQL, Salida, NumeroTitulos);
```

```
Salida.println("</table>");
```

```
Salida.println("</td>");
```

```
Salida.println("<tr>");
```

```
//REPORTE POR TIPOS DE OBRA
```

```
Salida.println("<td align=center><b>Programación y Ejecución por Tipo de Obra</b><br><br>");
```

```
Salida.println("<table class=colortable>");
```

```
ResultadoSQL=InstruccionSQL.executeQuery("exec  
"+USUARIO+", "+institucion+", "+ANIO);
```

ProgramacionEjecucionUnidadesMedida

```
MetaDatosSQL=ResultadoSQL.getMetaData();
```

```
NumeroTitulos=MetaDatosSQL.getColumnCount();
```

```
Salida.println("<table align=center class=colortable2>");
```

```

Salida.println("<tr>");
    for (int i=0;i<NumeroTitulos;i++)
    {
        Salida.println("<td class=hcolor>"+MetaDatosSQL.getColumnLabel(i+1).trim());
    }

```

```

Tabla(ResultadoSQL, Salida, NumeroTitulos);
Salida.println("</table>");
Salida.println("</td>");

```

```
//REPORTE POR FUENTE DE FINANCIAMIENTO
```

```

Salida.println("<td align=center><b>Programación y Ejecución Por Fuente de
Financiamiento</b><br><br>");
Salida.println("<table class=colortable>");
ResultadoSQL=InstruccionSQL.executeQuery("exec ProgramacionEjecucionFuente
"+USUARIO+", "+institucion+", "+ANIO);

```

```

MetaDatosSQL=ResultadoSQL.getMetaData();
NumeroTitulos=MetaDatosSQL.getColumnCount();
Salida.println("<table align=center class=colortable2>");
Salida.println("<tr>");
    for (int i=0;i<NumeroTitulos;i++)
    {
        Salida.println("<td class=hcolor>"+MetaDatosSQL.getColumnLabel(i+1).trim());
    }

```

```

Tabla(ResultadoSQL, Salida, NumeroTitulos);
Salida.println("</table>");
Salida.println("</td>");
Salida.println("</tr>");

```

```

Salida.println("<tr>");
//REPORTE POR INSTITUCION
Salida.println("<td align=center colspan=2><b>Programación y Ejecución Por Institución</b><br><br>");
Salida.println("<table class=colortable>");
ResultadoSQL=InstruccionSQL.executeQuery("exec ProgramacionEjecucionInstitucion
"+USUARIO+", "+institucion+", "+ANIO);

```

```

MetaDatosSQL=ResultadoSQL.getMetaData();
NumeroTitulos=MetaDatosSQL.getColumnCount();
Salida.println("<table align=center class=colortable2>");
Salida.println("<tr>");
    for (int i=0;i<NumeroTitulos;i++)
    {
        Salida.println("<td class=hcolor>"+MetaDatosSQL.getColumnLabel(i+1).trim());
    }

```

```

Tabla(ResultadoSQL, Salida, NumeroTitulos);
Salida.println("</table>");

```

```
Salida.println("</td>");
Salida.println("</tr>");
```

```
    }
    catch (Exception Error)
    {
        Salida.println("Situación Anormal!");
        Error.printStackTrace(Salida);
    }
    finally
    {
        try
        {
            if (ResultadoSQL!=null) ResultadoSQL.close();
            if (InstruccionSQL!=null) InstruccionSQL.close();
            if (ConexionSQL!=null) ConexionSQL.close();
            if (ResultadoSQL2!=null) ResultadoSQL2.close();
            if (InstruccionSQL2!=null) InstruccionSQL2.close();
            if (ConexionSQL2!=null) ConexionSQL2.close(); }
        }
    }
```

```
    catch (Exception Error) {}
}
}
```

```
private void Tabla(ResultSet ResultadoSQL,
                  PrintWriter Salida,
                  int NumeroTitulos)
```

```
throws Exception
```

```
{
```

```
    // Procesando Datos del Cuadro
```

```
    String Columna1="";
```

```
    while (ResultadoSQL.next())
```

```
    {
```

```
        Columna1=ResultadoSQL.getString(1);
```

```
        if (Columna1.startsWith("TOTAL"))
```

```
        {
```

```
            Salida.println("<tr align=right bgcolor=black>");
```

```
            Salida.println("<th><font color=FFFFFF size=2>" +Columna1);
```

```
            for (int i=2;i<=NumeroTitulos;i++)
```

```
            {
```

```
                Salida.println("<td class=tdfont>" + ResultadoSQL.getString(i));
```

```
            }
```

```
        }
```

```
        else
```

```
        {
```

```
            Salida.println("<tr align=left>");
```

```
            Salida.println("<td><font size=2>" +Columna1);
```

```
            for (int i=2;i<=NumeroTitulos;i++)
```

```
            {
```

```

        Salida.println("<td class=tdfont>" + ResultadoSQL.getString(i);
    }
}
}
}
}
}

```

### *Servlet para generar estado de reportes*

Para lograr reportes con un mayor detalle se ha creado el siguiente servlet:

```

import Usuario;
import Menu;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.io.*;

public class ReportesListado extends HttpServlet
{
    public void doGet(HttpServletRequest Requerimiento, HttpServletResponse Respuesta)
    throws ServletException, IOException
    {
        // Leyendo Parámetros

        String PARAMETRO=Requerimiento.getParameter("PARAMETRO");
        String ANIO=Requerimiento.getParameter("ANIO");
        String CODIGO=Requerimiento.getParameter("CODIGO");
        String INSTITUCION=Requerimiento.getParameter("INSTITUCION");
        String USUARIO=Requerimiento.getParameter("USUARIO");

        // Abriendo Archivo de Salida
        Respuesta.setContentType("text/html");
        PrintWriter Salida=new PrintWriter(Respuesta.getOutputStream());

        // Iniciando Escritura de Documento HTML; Encabezado y Titulo de Cuadro
        Salida.println("<html>");
        Salida.println("<head>");
        Salida.println("<title>Sistema de Información de Proyectos</title>");
        Salida.println("<LINK href=\\\"/HTML/include/ourstyle.css\\\" type=text/css rel=stylesheet>");
        Salida.println("<SCRIPT          language=JavaScript          src=/HTML/include/ourScript.js'
type=text/javascript></SCRIPT>");
        Salida.println("</head>");
        Salida.println("<body>");

        // Ejecutando Proceso Principal
        Proceso(Salida,CODIGO,PARAMETRO,INSTITUCION,ANIO,USUARIO);
    }
}

```

```

// Finalizando Escritura de Documento HTML y Cerrando Archivo de Salida
Salida.println("</body>");
Salida.println("</html>");
Salida.flush();
Salida.close();
}

public void init(ServletConfig cfg) throws ServletException { super.init(cfg); }

public void destroy() { super.destroy(); }

private void Proceso(PrintWriter Salida,String CODIGO, String PARAMETRO,String
INSTITUCION,String ANIO,String USUARIO)
{

    // Inicializando Variables para Conexión con ODBC
    Connection ConexionSQL=null;
    Statement InstruccionSQL=null;
    ResultSet ResultadoSQL=null;
    Connection ConexionSQL2=null;
    Statement InstruccionSQL2=null;
    ResultSet ResultadoSQL2=null;

try
{

    Usuario MIUSUARIO = new Usuario();
    String Driver = MIUSUARIO.driver;
    String DB = MIUSUARIO.myBD;
    String Usr = MIUSUARIO.usuario;
    String Pass= MIUSUARIO.password;
    String dirhtml= MIUSUARIO.DirHTML;

    ConexionSQL = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL=ConexionSQL.createStatement();
    ConexionSQL2 = DriverManager.getConnection(DB,Usr,Pass);
    InstruccionSQL2=ConexionSQL2.createStatement();

    Menu DoMenu = new Menu();

    DoMenu.CREARMENU(USUARIO,Salida,Driver,Usr,DB,Pass);

    DoMenu.Encabezado(Salida);

    String Consulta="";
    int NumeroTitulos;
    ResultSetMetaData MetaDatosSQL;

    if(CODIGO.equals("01"))
    {

```

```

Consulta="select nombre_etapa from etapas where codigo_etapa="+PARAMETRO;
ResultadoSQL=InstruccionSQL.executeQuery(Consulta);
ResultadoSQL.next();
String etapa=ResultadoSQL.getString(1);

Salida.println("<center><h1>PROYECTOS EN ETAPA "+etapa+"</h1></center>");

// Salida.println("<table>");

ResultadoSQL=InstruccionSQL.executeQuery("exec                                ProyectosEtapa
"+PARAMETRO+", "+INSTITUCION+", "+ANIO);

MetaDatosSQL=ResultadoSQL.getMetaData();
NumeroTitulos=MetaDatosSQL.getColumnCount();
Salida.println("<table align=center class=colortable2>");
Salida.println("<tr >");
    for (int i=0;i<NumeroTitulos;i++)
    {
        Salida.println("<td class=hcolor>"+MetaDatosSQL.getColumnLabel(i+1).trim());
    }

Tabla(ResultadoSQL, Salida, NumeroTitulos);
Salida.println("</table>");
}

if(CODIGO.equals("02"))
{
    Consulta="select          nombre_fuente          from          fuentes_financiamiento          where
codigo_fuente="+PARAMETRO+"";
ResultadoSQL=InstruccionSQL.executeQuery(Consulta);
ResultadoSQL.next();
String fuente=ResultadoSQL.getString(1);

Salida.println("<center><h1>PROYECTOS FINANCIADOS POR "+fuente+"</h1></center>");

// Salida.println("<table>");
ResultadoSQL=InstruccionSQL.executeQuery("exec                                ProyectosFuente
"+PARAMETRO+", "+INSTITUCION+", "+ANIO);

MetaDatosSQL=ResultadoSQL.getMetaData();
NumeroTitulos=MetaDatosSQL.getColumnCount();
Salida.println("<table align=center class=colortable2>");
Salida.println("<tr >");
    for (int i=0;i<NumeroTitulos;i++)
    {
        Salida.println("<td class=hcolor>"+MetaDatosSQL.getColumnLabel(i+1).trim());
    }

Tabla(ResultadoSQL, Salida, NumeroTitulos);
Salida.println("</table>");

```

```

}

if(CODIGO.equals("03"))
{
    Consulta="select nombre_unidad from unidades_medida where codigo_unidad="+PARAMETRO;
    ResultadoSQL=InstruccionSQL.executeQuery(Consulta);
    ResultadoSQL.next();
    String unidad=ResultadoSQL.getString(1);

    Salida.println("<center><h1>OBRAS DE TIPO \"'+unidad+'\"</h1></center>");

//    Salida.println("<table>");

        ResultadoSQL=InstruccionSQL.executeQuery("exec                                TiposObra
"+PARAMETRO+", "+INSTITUCION+", "+ANIO+", "+USUARIO);

        MetaDatosSQL=ResultadoSQL.getMetaData();
        NumeroTitulos=MetaDatosSQL.getColumnCount();
        Salida.println("<table align=center class=colortable2>");
        Salida.println("<tr >");
            for (int i=0;i<NumeroTitulos;i++)
            {
                Salida.println("<td class=hcolor>"+MetaDatosSQL.getColumnLabel(i+1).trim());
            }

        Tabla(ResultadoSQL, Salida, NumeroTitulos);
        Salida.println("</table>");

}

if(CODIGO.equals("04"))
{
    Consulta="select nombre_obra from obras_fisicas where codigo_obra="+PARAMETRO;
    ResultadoSQL=InstruccionSQL.executeQuery(Consulta);
    ResultadoSQL.next();
    String obra=ResultadoSQL.getString(1);

    Salida.println("<center><h1>PROYECTOS DE "+obra+"</h1></center>");

//    Salida.println("<table class=colortable>");

        ResultadoSQL=InstruccionSQL.executeQuery("exec                                ProyectosObra
"+PARAMETRO+", "+INSTITUCION+", "+ANIO);

        MetaDatosSQL=ResultadoSQL.getMetaData();
        NumeroTitulos=MetaDatosSQL.getColumnCount();

```

```

        Salida.println("<table align=center class=colortable2>");
        Salida.println("<tr >");
        for (int i=0;i<NumeroTitulos;i++)
        {
            Salida.println("<td class=hcolor>"+MetaDatosSQL.getColumnLabel(i+1).trim());
        }

        Tabla(ResultadoSQL, Salida, NumeroTitulos);
        Salida.println("</table>");

    }

}

catch (Exception Error)
{
    Salida.println("Situación Anormal!");
    Error.printStackTrace(Salida);
}
finally
{
    try
    {
        if (ResultadoSQL!=null) ResultadoSQL.close();
        if (InstruccionSQL!=null) InstruccionSQL.close();
        if (ConexionSQL!=null) ConexionSQL.close();
        if (ResultadoSQL2!=null) ResultadoSQL2.close();
        if (InstruccionSQL2!=null) InstruccionSQL2.close();
        if (ConexionSQL2!=null) ConexionSQL2.close(); }

}

catch (Exception Error) {}
}

private void Tabla(ResultSet ResultadoSQL,
                  PrintWriter Salida,
                  int NumeroTitulos)
throws Exception
{
    // Procesando Datos del Cuadro
    String Columna1="";
    while (ResultadoSQL.next())
    {
        Columna1=ResultadoSQL.getString(1);
        if (Columna1.startsWith("TOTAL"))
        {
            Salida.println("<tr align=right bgcolor=black>");
            Salida.println("<th><font color=FFFFFF size=2>"+Columna1);
            for (int i=2;i<=NumeroTitulos;i++)
            {

```



```

var svgDocument;
var textoU1;
var textoU2;

////////////////////////////////////

function SVGIcons(evt){
    svgiconDoc = evt.getTarget();
}
////////////////////////////////////
////////////////////////////////////Iniciar valores del mapa////////////////////////////////////
////////////////////////////////////Viewport del mapa////////////////////////////////////
////////////////////////////////////

function iniciarMapa()
{
    svgdib = document.all_map.getSVGDocument();
    svgMapaViewport = svgdib.getElementById("mapasvg");
    svgPanelS = svgdib.getElementById("panelSelecObra");
    svgPanelU = svgdib.getElementById("panelUbicObra");
    svgMapitaView = svgdib.getElementById("sombraCuad");
}

////////////////////////////////////Otros valores inicializados////////////////////////////////////
////////////////////////////////////Alturas, anchos, centros////////////////////////////////////

function initMap(evt)
{
    var obj2;
    svgDocument=document.all_map.getSVGDocument();
    Mapa = evt.getTarget();
    mapah=625;
    mapaw=615;
    nuevah=625; /// el ctrx = 312.5
    nuevaw=615; /// el ctry = 307.5
    ctrx=mapaw/2;
    ctry=mapah/2;

    obj2=svgDocument.getElementById("UbicOcultar");
    obj2.setAttribute('visibility','hidden');

    textoU1=svgDocument.getElementById("txt_ubic1");
    textoU2=svgDocument.getElementById("txt_ubic2");
}

////////////////////////////////////Inicialización de mapa de orientación////////////////////////////////////

function initme(evt)
{
    mapita = evt.getTarget();
    //navega=evt.getTarget();
}

////////////////////////////////////Movimiento del mapa de orientación////////////////////////////////////

```

```

function initmapita(evt)
{
mover_mapita = evt.getTarget();
}

////////////////////////////////////
//Funcion para desplazar el mapa hacia un punto destx,desty//
////////////////////////////////////

function Ir(destx,desty)
{
transx_aux=transx_aux-transx_aux+destx-(nuevaw/2);
transy_aux=transy_aux-transy_aux+desty-(nuevah/2);
transx_aux1=transx_aux1-transx_aux1+destx-(nuevaw/2);
transy_aux1=transy_aux1-transy_aux1+desty-(nuevah/2);
//alert("transx_aux: "+transx_aux+"  transy_aux: "+transy_aux);

nuevoViewBox = transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
svgMapaViewport.setAttribute("viewBox",nuevoViewBox);
svgMapitaView.setAttribute("x",transx_aux);
svgMapitaView.setAttribute("y",transy_aux);
svgMapitaView.setAttribute("width",nuevaw);
svgMapitaView.setAttribute("height",nuevah);
}

////////////////////////////////////
//Función para cambiar el nivel de zoom del mapa//////////
////////////////////////////////////

function cambiarzoom(zoomlevel)
{
var nuevoViewBox;
var factor;

factor=1/zoomlevel;
nuevah=mapah*factor;
nuevaw=mapaw*factor;

//nuevoViewBox = transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
transx_aux=transx_aux-transx_aux+ctrx-(nuevaw/2);
transy_aux=transy_aux-transy_aux+ctry-(nuevah/2);
transx_aux1=transx_aux1-transx_aux1+ctrx-(nuevaw/2);
transy_aux1=transy_aux1-transy_aux1+ctry-(nuevah/2);

nuevoViewBox = transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
svgMapaViewport.setAttribute("viewBox",nuevoViewBox);
svgMapitaView.setAttribute("x",transx_aux);
svgMapitaView.setAttribute("y",transy_aux);
svgMapitaView.setAttribute("width",nuevaw);
svgMapitaView.setAttribute("height",nuevah);
if (zoomlevel<1){
svgMapitaView.setAttribute("x",0);
svgMapitaView.setAttribute("y",0);
svgMapitaView.setAttribute("width",mapaw);
svgMapitaView.setAttribute("height",mapah);
}
}

```

```

}
scale_aux=zoomlevel;
}

////////////////////////////////////
//Función para realizar desplazamientos por el mapa////////
////////////////////////////////////

function desplazar(sentido,nivel)
{
var nuevoViewBox;
var desp;
var factor;

factor=1/scale_aux;

switch (nivel) {
    case 1: desp=10;
        break;
    case 2: desp=30;
        break;
    case 3: desp=100;
        break;
    }

switch (sentido) {
    case "der":
        transx_aux+=desp;
        transx_aux1+=desp;
        ctrx+=desp;
        nuevoViewBox=transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
        svgMapViewport.setAttribute("viewBox",nuevoViewBox);
        svgMapitaView.setAttribute("x",transx_aux);
        break;
    case "izq":
        if (transx_aux>0){
            transx_aux-=desp;
            transx_aux1-=desp;
            ctrx-=desp;
            nuevoViewBox=transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
            svgMapViewport.setAttribute("viewBox",nuevoViewBox);
            svgMapitaView.setAttribute("x",transx_aux);
        }
        break;
    case "arriba":
        if (transy_aux>0){
            transy_aux-=desp;
            transy_aux1-=desp;
            //transy_auxp-=desp*factor;
            ctry-=desp;
            nuevoViewBox=transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
            svgMapViewport.setAttribute("viewBox",nuevoViewBox);
            svgMapitaView.setAttribute("y",transy_aux);
        }
        break;
    case "abajo":

```

```

        transy_aux+=desp;
        transy_aux1+=desp;
        //transy_auxp+=desp*factor;
        ctry+=desp;
        nuevoViewBox=transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
        svgMapViewport.setAttribute("viewBox",nuevoViewBox);
        svgMapitaView.setAttribute("y",transy_aux);
    break;
}

//alert(ctrx+" "+ctry);

}

////////////////////////////////////
//Función para Ocultar checkes de las cajas de selección/////
////////////////////////////////////

function ocultar(evt,quien)
{
var obj=evt.getTarget();
obj.setAttribute('visibility','hidden');
obj2=svgDocument.getElementById(quien);
obj2.setAttribute('visibility','hidden');
}

////////////////////////////////////
//Función para mostrar checkes de las cajas de selección/////
////////////////////////////////////

function mostrar(quien1,quien2)
{
obj=svgDocument.getElementById(quien1);
obj.setAttribute('visibility','visible');
obj2=svgDocument.getElementById(quien2);
obj2.setAttribute('visibility','visible');
}

////////////////////////////////////
//Función para la obtención inicial de coordenadas/////
////////////////////////////////////

function coord(evt)
{
directTarget = evt.getTarget();
svgDoc = directTarget.getOwnerDocument();
svgSVG OBJ = svgDoc.getDocumentElement();
scale = svgSVG OBJ.getCurrentScale();
trans = svgSVG OBJ.getCurrentTranslate();
transx = trans.getX();
transy = trans.getY();

scale1 = scale;
trans1 = trans;
transx1 = transx1;
transy1 = transy1;

```

```

}

////////////////////////////////////
///Obtención de coordenadas - mapa de orientacion////////////////////////////////
////////////////////////////////////

function coordPeq(evt)
{
var objetivo = evt.getTarget();
var svgDocPeq= objetivo.getOwnerDocument();
var svgSVGOBJPeq = svgDocPeq.getDocumentElement();
transPeq = svgSVGOBJPeq.getCurrentTranslate();
var XPosPeq = (evt.getClientX()-633)*5.5;
var YPosPeq = (evt.getClientY()-191)*5.5;
Ir(XPosPeq, YPosPeq);
}

////////////////////////////////////
//////////Cambiar color - control de desplazamiento////////////////////////////////
////////////////////////////////////

function changeCol(evt, color)
{
var targetshape = evt.getTarget();
targetshape.setAttribute("fill",color);
}

////////////////////////////////////
///Retornar color default - control desplazamiento////////////////////////////////
////////////////////////////////////

function regresaCol(evt)
{
var targetshape = evt.getTarget();
targetshape.setAttribute("fill", "#FDFCAF");
}

////////////////////////////////////
//////////Textos Barra de Zoom////////////////////////////////////////
////////////////////////////////////

function ver(evt)
{
var targetshape = evt.getTarget();
targetshape.setAttribute("fill", "#FF0000");
}

////////////////////////////////////
//////////Textos Barra de Zoom////////////////////////////////////////
////////////////////////////////////

function nover(evt)
{
var targetshape = evt.getTarget();
targetshape.setAttribute("fill", "#FFFFFF");
}

```

```

////////////////////////////////////
//////Cambiar texto de visualización de coordenadas////////////////////////////////
////////////////////////////////////

function changeText(evt)
{
    coord(evt);
    targetXtext=svgDocument.getElementById("XPos");
    targetYtext=svgDocument.getElementById("YPos");

    var XPos = evt.getClientX();
    var YPos = evt.getClientY();
    if (scale_aux != 1) scale=scale_aux;
    if (transx_aux != 0) transx=transx_aux;
    if (transy_aux != 0) transy=transy_aux;
    XPos=parseInt((XPos+(transx_aux*scale_aux))/scale_aux);
    YPos=parseInt((YPos+(transy_aux*scale_aux))/scale_aux);

    var newXPosText = svgDocument.createTextNode("Posicion X : " + XPos);
    var newYPosText = svgDocument.createTextNode("Posicion Y : " + YPos);

    targetXtext.replaceChild(newXPosText,targetXtext.getFirstChild());
    targetYtext.replaceChild(newYPosText,targetYtext.getFirstChild());
}

////////////////////////////////////
//////Valores del texto de visualización de coordenadas////////////////////////////////
//////cuando el cursor no está sobre el mapa////////////////////////////////
////////////////////////////////////

function changeTextNotOver(evt)
{
    targetXtext=svgDocument.getElementById("XPos");
    targetYtext=svgDocument.getElementById("YPos");

    //var newXPosText = svgDocument.createTextNode("Escala val : " + scale);
    //var newYPosText = svgDocument.createTextNode("X pan : "+transx+" Y pan:"+transy);

    //targetXtext.replaceChild(newXPosText,targetXtext.getFirstChild());
    //targetYtext.replaceChild(newYPosText,targetYtext.getFirstChild());

    cantonid="";
}

////////////////////////////////////
//////Función para la obtención de coordenadas////////////////////////////////
//////al hacer clic sobre el mapa////////////////////////////////
////////////////////////////////////

function recordClick(evt)
{
    coord(evt);
    targetClickText=svgDocument.getElementById("ClickPos");

    var XPos = evt.getClientX();

```

```

var YPos = evt.getClientY();

if (scale_aux != 1) scale=scale_aux;
if (transx_aux != 0) transx=transx_aux;
if (transy_aux != 0) transy=transy_aux;

XPos=parseInt((XPos+(transx_aux*scale_aux))/scale_aux);
YPos=parseInt((YPos+(transy_aux*scale_aux))/scale_aux);
XPos1=XPos;
YPos1=YPos;

var newClickText =
  svgDocument.createTextNode("L.C. X=" + XPos + " Y=" + YPos);

targetClickText.replaceChild(newClickText,
  targetClickText.getFirstChild());
}

////////////////////////////////////
/////Visualización del panel de agregar obras////////////////////////////////
////////////////////////////////////

function verPanelU()
{
var obj, obj2;
obj=svgDocument.getElementById("UbicMostrar");
obj2=svgDocument.getElementById("UbicOcultar");
switch (panelU) {
  case 0:
    svgPanelU.setAttribute('transform',"translate(140,0)");
    panelU=1;
    obj.setAttribute('visibility','hidden');
    obj2.setAttribute('visibility','visible');
    textoU1.setAttribute('visibility','hidden');
    textoU2.setAttribute('visibility','visible');
    mostrar_icons();
    break;
  case 1:
    svgPanelU.setAttribute('transform',"translate(0,0)");
    panelU=0;
    obj.setAttribute('visibility','visible');
    obj2.setAttribute('visibility','hidden');
    textoU1.setAttribute('visibility','visible');
    textoU2.setAttribute('visibility','hidden');
    break;
}
}

////////////////////////////////////
/////Funcionalidad cuadro resumen de informacion (tooltip)/////
////////////////////////////////////

function estetooltip(estaobra)
{
  cuadro = svgDocument.getElementById("tip_"+estaobra);
  cuadro.setAttribute('style','visibility:visible;');
}

```

```

}

////////////////////////////////////
//Funcionalidad cuadro resumen de informacion (tooltip)/////
////////////////////////////////////

function notooltip(estaobra)
{
    cuadro = svgDocument.getElementById("tip_"+estaobra);
    cuadro.setAttribute('style','visibility:hidden;');
}

////////////////////////////////////
//Desplazamiento por combos/////
////////////////////////////////////

function ircombo(lugar)
{
    var donde="cantones_"+lugar;
    svgdib = document.all_map.getSVGDocument();
    destino=svgdib.getElementById(donde);
    if (destino==null)
        alert("No hay datos para el destino seleccionado");
    else
    {
        //mipos = destino.getCurrentTranslate();
        miposx = destino.getAttribute('x');
        miposy = destino.getAttribute('y');
        Ir(miposx,miposy);
    }
}

////////////////////////////////////
//Borde de seleccion de obras (en el mapa)/////
////////////////////////////////////

function sombra(otraobra)
{
    sombrita = svgDocument.getElementById("borde_"+otraobra);
    switch (indica_sombra) {
        case 0:
            sombrita.setAttribute('style','fill:#CC00FF;visibility:visible');
            sombraobra=sombrita;
            indica_sombra=1;
            break;
        case 1:
            sombrita.setAttribute('style','fill:#CC00FF;visibility:visible');
            sombraobra.setAttribute('style','fill:#CC00FF;visibility:hidden');
            sombraobra=sombrita;
            break;
    }
}

////////////////////////////////////
//Borde de seleccion de obras para agregar (en el panel)///
////////////////////////////////////

```

```

function sombra_agrega(estaunidad)
{
    sombrita_agrega = svgDocument.getElementById("borde_agrega_"+estaunidad);
    //alert(cuadro+" "+estaobra);
    // svgdib = document.all_map.getSVGDocument();
    // cuadro = svgdib.getElementById("tip_"+estaobra);
    // cuadro.setAttribute('visibility','visible');
    // alert(sombraobra);

    switch (indica_sombra_agrega) {
        case 0:
            sombrita_agrega.setAttribute('style','fill:#CC00FF;visibility:visible');
            sombraobra_agrega=sombrita_agrega;
            indica_sombra_agrega=1;
            break;
        //case 1:
        // sombrita_agrega.setAttribute('style','fill:#CC00FF;visibility:visible');
        // sombraobra_agrega.setAttribute('style','fill:#CC00FF;visibility:hidden');
        // sombraobra_agrega=sombrita_agrega;
        //break;
    }
}

```

## Control del mapa SVG para reportes (mapita\_svg\_reportes.js)

```

////////////////////Declaración de variables////////////////////
var svgSVGOBJ;
var scale;
var scale_aux=1;
var svgDoc;
var directTarget;
var trans;
var transx;
var transy;
var transy_aux=0;
var transx_aux=0;
var Mapa;
var mapita;
var zoomlevel;
var mover_mapita;
var mapah;
var mapaw;
var nuevah=625;
var nuevaw=615;
var nuevahpeq=3125;
var nuevawpeq=3075;
var ctrx;
var ctry;
var panelR=0;
var panelS=0;
var svgdib;
var sombraobra=0;
var indica_sombra=0;

```

```

var svgDocument;
var val;
var delineado1;
var delineado2;

////////////////////////////////////

function SVGIcons(evt){
    svgiconDoc = evt.getTarget();
}

////////////////////////////////////
////////////////////////////////////Iniciar valores del mapa////////////////////////////////////
////////////////////////////////////Viewport del mapa////////////////////////////////////
////////////////////////////////////

function iniciarMapa()
{
    svgdib = document.all_map.getSVGDocument();
    svgMapaViewport = svgdib.getElementById("mapasvg");
    svgPanelS = svgdib.getElementById("panelSelecObra");
    svgPanelR = svgdib.getElementById("panelRepObra");
    svgMapitaView = svgdib.getElementById("sombraCuad");

}
////////////////////////////////////
////////////////////////////////////Otros valores inicializados////////////////////////////////////
////////////////////////////////////Alturas, anchos, centros////////////////////////////////////
////////////////////////////////////

function initMap(evt)
{
    var obj1,obj2;
    var numcanfin;
    val=0;
    svgDocument=document.all_map.getSVGDocument();
    Mapa = evt.getTarget();
    mapah=625;
    mapaw=615;
    nuevah=625; /// el ctrx = 312.5
    nuevaw=615; /// el ctry = 307.5
    ctrx=mapaw/2;
    ctry=mapah/2;
    arr_can= new Array();
    for(i=1; i<2500;i++){
        numcanfin=svgDocument.getElementById("cantones_"+i);
        if (numcanfin!=null){
            arr_can[val]=numcanfin;
            val++;
            //alert(numcanfin);
        }
    }

}

obj1=svgDocument.getElementById("reporOcultar");
obj2=svgDocument.getElementById("selecOcultar");
obj1.setAttribute('visibility','hidden');

```

```

obj2.setAttribute('visibility','hidden');
delineado1=svgDocument.getElementById("rep_finan");
delineado2=svgDocument.getElementById("rep_fisico");

}

//////////Inicialización de mapa de orientación//////////

function initme(evt)
{
mapita = evt.getTarget();
//navega=evt.getTarget();
}

//////////Movimiento del mapa de orientación//////////

function initmapita(evt)
{
mover_mapita = evt.getTarget();
}

//////////////////////////////////////
///Funcion para desplazar el mapa hacia un punto destx,desty///
//////////////////////////////////////

function Ir(destx,desty)
{
transx_aux=transx_aux-transx_aux+destx-(nuevaw/2);
transy_aux=transy_aux-transy_aux+desty-(nuevah/2);
transx_aux1=transx_aux1-transx_aux1+destx-(nuevaw/2);
transy_aux1=transy_aux1-transy_aux1+desty-(nuevah/2);
//alert("transx_aux: "+transx_aux+"  transy_aux: "+transy_aux);

nuevoViewBox = transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
svgMapViewport.setAttribute("viewBox",nuevoViewBox);
svgMapitaView.setAttribute("x",transx_aux);
svgMapitaView.setAttribute("y",transy_aux);
svgMapitaView.setAttribute("width",nuevaw);
svgMapitaView.setAttribute("height",nuevah);
}

//////////////////////////////////////
///Función para cambiar el nivel de zoom del mapa//////////
//////////////////////////////////////

function cambiarzoom(zoomlevel)
{
var nuevoViewBox;
var factor;

factor=1/zoomlevel;
nuevah=mapah*factor;
nuevaw=mapaw*factor;

transx_aux=transx_aux-transx_aux+ctrx-(nuevaw/2);

```

```

transy_aux=transy_aux-transy_aux+ctry-(nuevah/2);
transx_aux1=transx_aux1-transx_aux1+ctrx-(nuevaw/2);
transy_aux1=transy_aux1-transy_aux1+ctry-(nuevah/2);

nuevoViewBox = transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
svgMapViewport.setAttribute("viewBox",nuevoViewBox);
svgMapitaView.setAttribute("x",transx_aux);
svgMapitaView.setAttribute("y",transy_aux);
svgMapitaView.setAttribute("width",nuevaw);
svgMapitaView.setAttribute("height",nuevah);
if (zoomlevel<1){
svgMapitaView.setAttribute("x",0);
svgMapitaView.setAttribute("y",0);
svgMapitaView.setAttribute("width",mapaw);
svgMapitaView.setAttribute("height",mapah);
}
scale_aux=zoomlevel;

}

////////////////////////////////////
//Función para realizar desplazamientos por el mapa////////
////////////////////////////////////

function desplazar(sentido,nivel)
{
var nuevoViewBox;
var desp;
var factor;

factor=1/scale_aux;

switch (nivel) {
    case 1: desp=10;
        break;
    case 2: desp=30;
        break;
    case 3: desp=100;
        break;
}

switch (sentido) {
    case "der":
        transx_aux+=desp;
        transx_aux1+=desp;
        ctrx+=desp;
        nuevoViewBox=transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
        svgMapViewport.setAttribute("viewBox",nuevoViewBox);
        svgMapitaView.setAttribute("x",transx_aux);
        break;
    case "izq":
        if (transx_aux>0){
            transx_aux-=desp;
            transx_aux1-=desp;
            ctrx-=desp;
            nuevoViewBox=transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;

```

```

        svgMapaViewport.setAttribute("viewBox",nuevoViewBox);
        svgMapitaView.setAttribute("x",transx_aux);
    }
    break;
    case "arriba":
        if (transy_aux>0){
            transy_aux-=desp;
            transy_aux1-=desp;
            //transy_auxp-=desp*factor;
            ctry-=desp;
            nuevoViewBox=transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
            svgMapaViewport.setAttribute("viewBox",nuevoViewBox);
            svgMapitaView.setAttribute("y",transy_aux);
        }
    break;
    case "abajo":
        transy_aux+=desp;
        transy_aux1+=desp;
        //transy_auxp+=desp*factor;
        ctry+=desp;
        nuevoViewBox=transx_aux+" "+transy_aux+" "+nuevaw+" "+nuevah;
        svgMapaViewport.setAttribute("viewBox",nuevoViewBox);
        svgMapitaView.setAttribute("y",transy_aux);
    break;
}
}
}

```

```

////////////////////////////////////
//Función para Ocultar checkes de las cajas de selección/////
////////////////////////////////////

```

```

function ocultar(evt,quien)
{
    var obj=evt.getTarget();
    obj.setAttribute('visibility','hidden');
    obj2=svgDocument.getElementById(quien);
    obj2.setAttribute('visibility','hidden');
}

```

```

////////////////////////////////////
//Función para mostrar checkes de las cajas de selección/////
////////////////////////////////////

```

```

function mostrar(quien1,quien2)
{
    obj=svgDocument.getElementById(quien1);
    obj.setAttribute('visibility','visible');
    obj2=svgDocument.getElementById(quien2);
    obj2.setAttribute('visibility','visible');
}

```

```

////////////////////////////////////
//Función para la obtención inicial de coordenadas//////////
////////////////////////////////////

```

```

function coord(evt)

```

```

{
directTarget = evt.getTarget();
svgDoc = directTarget.getOwnerDocument();
svgSVGObj = svgDoc.getDocumentElement();
scale = svgSVGObj.getCurrentScale();
trans = svgSVGObj.getCurrentTranslate();
transx = trans.getX();
transy = trans.getY();

scale1 = scale;
trans1 = trans;
transx1 = transx1;
transy1 = transy1;
}

////////////////////////////////////
////Obtención de coordenadas - mapa de orientacion////////////////////////////////
////////////////////////////////////

function coordPeq(evt)
{
var objetivo = evt.getTarget();
var svgDocPeq= objetivo.getOwnerDocument();
var svgSVGObjPeq = svgDocPeq.getDocumentElement();
transPeq = svgSVGObjPeq.getCurrentTranslate();
var XPosPeq = (evt.getClientX()-633)*5.5;
var YPosPeq = (evt.getClientY()-191)*5.5;
Ir(XPosPeq,YPosPeq);
}

////////////////////////////////////
///////Cambiar color - control de desplazamiento////////////////////////////////
////////////////////////////////////

function changeCol(evt, color)
{
var targetshape = evt.getTarget();
targetshape.setAttribute("fill",color);
}

////////////////////////////////////
////Retornar color default - control desplazamiento////////////////////////////////
////////////////////////////////////

function regresaCol(evt)
{
var targetshape = evt.getTarget();
targetshape.setAttribute("fill", "#FDFCAF");
}
////////////////////////////////////
////////////////////////////////////Textos Barra de Zoom////////////////////////////////
////////////////////////////////////

function ver(evt)
{
var targetshape = evt.getTarget();

```

```

    targetshape.setAttribute("fill", "#FF0000");
}

////////////////////////////////////
/////////Textos Barra de Zoom/////////
////////////////////////////////////

function nover(evt)
{
    var targetshape = evt.getTarget();
    targetshape.setAttribute("fill", "#FFFFFF");
}

////////////////////////////////////
/////////Cambiar texto de visualización de coordenadas/////////
////////////////////////////////////

function changeText(evt)
{
    coord(evt);
    targetXtext=svgDocument.getElementById("XPos");
    targetYtext=svgDocument.getElementById("YPos");

    var XPos = evt.getClientX();
    var YPos = evt.getClientY();
    if (scale_aux != 1) scale=scale_aux;
    if (transx_aux != 0) transx=transx_aux;
    if (transy_aux != 0) transy=transy_aux;
    XPos=parseInt((XPos+(transx_aux*scale_aux))/scale_aux);
    YPos=parseInt((YPos+(transy_aux*scale_aux))/scale_aux);

    var newXPosText = svgDocument.createTextNode("Posicion X : " + XPos);
    var newYPosText = svgDocument.createTextNode("Posicion Y : " + YPos);

    targetXtext.replaceChild(newXPosText,targetXtext.getFirstChild());
    targetYtext.replaceChild(newYPosText,targetYtext.getFirstChild());
}

////////////////////////////////////
/////////Valores del texto de visualización de coordenadas/////////
/////////cuando el cursor no está sobre el mapa/////////
////////////////////////////////////

function changeTextNotOver(evt)
{
    targetXtext=svgDocument.getElementById("XPos");
    targetYtext=svgDocument.getElementById("YPos");

    //var newXPosText = svgDocument.createTextNode("Escala val : " + scale);
    //var newYPosText = svgDocument.createTextNode("X pan : "+transx+" Y pan:"+transy);

    //targetXtext.replaceChild(newXPosText,targetXtext.getFirstChild());
    //targetYtext.replaceChild(newYPosText,targetYtext.getFirstChild());

    cantonid="";
}

```

```

////////////////////////////////////
////////Función para la obtención de coordenadas////////
////////al hacer clic sobre el mapa////////
////////////////////////////////////

function recordClick(evt)
{
    coord(evt);
    targetClickText=svgDocument.getElementById("ClickPos");
    var XPos = evt.getClientX();
    var YPos = evt.getClientY();

    if (scale_aux != 1) scale=scale_aux;
    if (transx_aux != 0) transx=transx_aux;
    if (transy_aux != 0) transy=transy_aux;

    XPos=parseInt((XPos+(transx_aux*scale_aux))/scale_aux);
    YPos=parseInt((YPos+(transy_aux*scale_aux))/scale_aux);
    XPos1=XPos;
    YPos1=YPos;

    var newClickText =
        svgDocument.createTextNode("L.C. X=" + XPos + " Y=" + YPos);

    targetClickText.replaceChild(newClickText,
        targetClickText.getFirstChild());
}

```

```

////////////////////////////////////
////////Visualización del panel de seleccionar obras////////
////////////////////////////////////

```

```

function verPanelS()
{
    var obj, obj2;
    obj=svgDocument.getElementById("selecMostrar");
    obj2=svgDocument.getElementById("selecOcultar");

    switch (panelS) {
        case 0:
            svgPanelS.setAttribute('transform',"translate(140,0)");
            panelS=1;
            obj.setAttribute('visibility','hidden');
            obj2.setAttribute('visibility','visible');
            break;
        case 1:
            svgPanelS.setAttribute('transform',"translate(0,0)");
            panelS=0;
            obj.setAttribute('visibility','visible');
            obj2.setAttribute('visibility','hidden');
            break;
    }
}

```

```

////////////////////////////////////

```

```
/////////Visualización del panel de reportes/////////
/////////
```

```
function verPanelR()
{
var obj, obj2;
obj=svgDocument.getElementById("reporMostrar");
obj2=svgDocument.getElementById("reporOcultar");
switch (panelR) {
    case 0:
        svgPanelR.setAttribute('transform',"translate(140,0)");
        panelR=1;
        obj.setAttribute('visibility','hidden');
        obj2.setAttribute('visibility','visible');
        mostrar_icons();
        break;
    case 1:
        svgPanelR.setAttribute('transform',"translate(0,0)");
        panelR=0;
        obj.setAttribute('visibility','visible');
        obj2.setAttribute('visibility','hidden');
        break;
    }
}
```

```
/////////
//////Funcionalidad cuadro resumen de informacion (tooltip)////
/////////
```

```
function estetooltip(estaobra)
{
    cuadro = svgDocument.getElementById("tip_"+estaobra);
    cuadro.setAttribute('style','visibility:visible;');
}
```

```
/////////
//////Funcionalidad cuadro resumen de informacion (tooltip)////
/////////
```

```
function notooltip(estaobra)
{
    cuadro = svgDocument.getElementById("tip_"+estaobra);
    cuadro.setAttribute('style','visibility:hidden;');
}
```

```
/////////
//////////Desplazamiento por combos/////////
/////////
```

```
function ircombo(lugar)
{
    var donde="cantones_"+lugar;

    svgdib = document.all_map.getSVGDocument();
    destino=svgdib.getElementById(donde);
```

```

if (destino==null)
    alert("No hay datos para el destino seleccionado");
else
{
    mipoxx = destino.getAttribute('x');
    mipoxy = destino.getAttribute('y');
    Ir(mipoxx,mipoxy);
}
}

```

```

////////////////////////////////////
//////////Borde de seleccion de obras (en el mapa)//////////
////////////////////////////////////

```

```

function sombra(otraobra)
{
    sombrita = svgDocument.getElementById("borde_"+otraobra);
    switch (indica_sombra) {
        case 0:
            sombrita.setAttribute('style','fill:#CC00FF;visibility:visible');
            sombraobra=sombrita;
            indica_sombra=1;
            break;
        case 1:
            sombrita.setAttribute('style','fill:#CC00FF;visibility:visible');
            sombraobra.setAttribute('style','fill:#CC00FF;visibility:hidden');
            sombraobra=sombrita;
            break;
    }
}

```

```

////////////////////////////////////
///Clasificación de cantones en base a reporte financiero///
////////////////////////////////////

```

```

function colorear_finan()
{
    var uncanton;
    var estecanton;
    svgdib = document.all_map.getSVGDocument();
    for(i=0; i<val;i++){
        mystyle = arr_can[i].getStyle();
        estecanton=svgdib.getElementById("reportfin_"+arr_can[i].getAttribute('id'));
        if (estecanton!=null){
            nivel=estecanton.getAttribute('valor');
            if (nivel<85 && nivel>=0)
            {
                color="#FF0000";
            }
            else
            {
                if (nivel>=85 && nivel<90)
                {
                    color="#FFFF66";
                }
            }
        }
    }
}

```

```

else
{
    if (nivel>=90)
    {
        color="#CCFF99";
    }
    else
    {
        color="#FFFFFF";
    }
}
}
mystyle.setProperty('fill',color);
mystyle.setProperty('fill-opacity','1');
}
}
cuadroestilo=svgdib.getElementById("rep_finan");
cuadroestilo2=svgdib.getElementById("rep_fisico");
suestilo =cuadroestilo.getStyle();
suestilo2 =cuadroestilo2.getStyle();
suestilo.setProperty('visibility','visible');
suestilo2.setProperty('visibility','hidden');
barra1=svgdib.getElementById("barra1");
barra1.setAttribute('style','visibility:visible;');
}

////////////////////////////////////
////Clasificación de cantones en base a reporte fisico/////
////////////////////////////////////

function colorear_fisico()
{
    var estecanton;
    var cuadroestilo;
    svgdib = document.all_map.getSVGDocument();
    for(i=0; i<val;i++){
        mystyle = arr_can[i].getStyle();
        estecanton=svgdib.getElementById("reportfis_"+arr_can[i].getAttribute('id'));
        if (estecanton!=null){
            nivel=estecanton.getAttribute('valor');
            if (nivel<85 && nivel>=0)
            {
                color="#FF3333";
            }
            else
            {
                if (nivel>=85 && nivel<90)
                {
                    color="#FFee77";
                }
                else
                {
                    if (nivel>=90)
                    {
                        color="#bbdd77";
                    }
                }
            }
        }
    }
}

```

```

        else
        {
            color="#FFFFFF";
        }
    }
}
mystyle.setProperty('fill',color);
mystyle.setProperty('fill-opacity','1');
//alert("+mystyle+");
}
}
cuadroestilo=svgdib.getElementById("rep_fisico");
cuadroestilo2=svgdib.getElementById("rep_finan");
suestilo =cuadroestilo.getStyle();
suestilo2 =cuadroestilo2.getStyle();
suestilo.setProperty('visibility','visible');
suestilo2.setProperty('visibility','hidden');
barra1=svgdib.getElementById("barra1");
barra1.setAttribute('style','visibility:visible;');
}

```

## **Interactividad y validaciones en formularios de captura de datos.** (ourScript.js)

```

    var ban_nueva_obra=0;
    var svgiconDoc;
    var svgmapDoc;
    var svgobraDoc;
    var myid="";
    var idobra=0;
    var myiddone="";
    var cantonid="";
    var tempcantonid="";
    var institucion;
    var parametro;
    var anio;
    var usuario;

var scale1;
var scale_aux1=1;
var trans1;
var transx1;
var transy1;
var transy_aux1=0;
var transx_aux1=0;
var transy_auxp1=0;
var transx_auxp1=0;
var XPos1=0;
var YPos1=0;

var punto=0;

```

```
/****** Muestra el canton sobre la que se esta ubicando la obra en el mapa******/
```

```
function cualcanton(evt,q)
```

```
{  
  
    if (q==1)  
    {  
        var onMyObj = get_target(evt);  
  
        var tempcantonid = onMyObj.getAttribute('id');  
        if(tempcantonid!=null && tempcantonid!="")  
        {  
            if (tempcantonid.indexOf('cantones_',0)!=-1)  
            {  
                var arraycan=tempcantonid.split('_');  
                cantonid="+arraycan[1]+";  
            }  
        }  
    }  
    if(q==2)  
    {  
        cantonid="";  
        document.proyectos.id_canton.value="";  
    }  
}
```

```
/****** Muestra los iconos para registro de obras******/
```

```
function mostrar_icons(evt)
```

```
{  
    svgiconDoc=document.all_map.getSVGDocument();  
    var element = svgiconDoc.getElementById('simbl');  
    var mystyle = element.getStyle();  
    mystyle.setProperty('visibility','visible');  
}
```

```
/****** Activa el icono que se seleccionó para ubicar obra en el mapa******/
```

```
function mostrar_simbolos(evt)
```

```
{  
    svgiconDoc=document.all_map.getSVGDocument();  
    if (ban_nueva_obra==0)  
    {  
        var onMyObj = get_target(evt);  
        myid=onMyObj.getAttribute('id');  
  
        if (myiddone!="")  
        {  
            var ids=myiddone.Split(',');  
        }  
  
        if (myid!='simbolos')
```

```

        {
            //alert('id= '+myid);
            xcuad=0;
            ycuad=0;

            var element = svgiconDoc.getElementById(myid);
            var mystyle = element.getStyle();
            mystyle.setProperty('fill','#000000');

            ban_nueva_obra=1;
            document.proyectos.id_tipo_obra.value=myid;
        }
        else
        {
            myid="";
            document.proyectos.id_tipo_obra.value="";
            cantonid="";
            document.proyectos.id_canton.value="";
        }
    }
}

/***** Captura los eventos que se generan dentro del SVG
*****/
function get_target (evt)
{
    var target = evt.getTarget();
    while (target && !target.getAttribute('id'))
        target = target.parentNode();
    return target;
}

/***** Mantiene las coordenadas capturadas del mapa
*****/

function Mycoord(evt)
{
    var directTarget = get_target(evt);
    svgmapDoc=document.all_map.getSVGDocument();
    var svgSVGObj = svgmapDoc.getElementById('mapa');
    scale1 = svgSVGObj.getCurrentScale();
    trans1 = svgSVGObj.getCurrentTranslate();
    transx1 = trans1.getX();
    transy1 = trans1.getY();
}

/***** Captura la ultima posición en la que se dió click, para el
ingreso de una nueva obra
cambia las propiedades del icono *****/

function MyrecordClick(evt)
{
    if (myid!=" && cantonid!=")

```

```

    {
        XPos1=XPos1;
        YPos1=YPos1;
        svgobraDoc=document.all_map.getSVGDocument();

        var element = svgobraDoc.getElementById(myid+'_obra');
        var mystyle = element.getStyle();
        mystyle.setProperty('visibility','visible');
        element.setAttribute('transform', 'translate('+XPos1+', '+YPos1+')');

        document.proyectos.pos_obra_x.value=XPos1;
        document.proyectos.pos_obra_y.value=YPos1;
        document.proyectos.id_canton.value=cantonid;
    }
    else
    {
        document.proyectos.pos_obra_x.value="";
        document.proyectos.pos_obra_y.value="";
        document.proyectos.id_canton.value="";
        cantonid="";
    }
}

/***** Función que realiza la validación de los datos necesarios para el registro
de obra *****/
function guardar_icons()
{
    var obra_nombre="";
    var good_obra=0;

    if (ban_nueva_obra==1)
    {
        if (document.proyectos.pos_obra_x.value==" ||
document.proyectos.pos_obra_y.value=="")
        {
            good_obra=1;
            alert('Debe ubicar la obra dando click sobre el mapa');
            return false;
        }

        obra_nombre=prompt('Introduzca nombre de Obra');

        if (obra_nombre==" || obra_nombre==null)
        {
            good_obra=1;
            alert('Debe introducir nombre de obra');
            return false;
        }
        if (cantonid==" && document.proyectos.id_canton.value=="")
        {
            good_obra=1;
            alert('Debe posicionar la obra dentro de los límites de algún canton');
            return false;
        }
    }
}

```

```

        if(good_obra==0)
        {
            document.proyectos.nom_obra.value=obra_nombre;

            window.open('/servlet/CapturaObras?id_tipo_obra='+document.proyectos.id_tipo_obra.value+'&pos_obra_x='+document.proyectos.pos_obra_x.value+'&pos_obra_y='+document.proyectos.pos_obra_y.value+'&nombre_obra='+document.proyectos.nom_obra.value+'&id_canton='+document.proyectos.id_canton.value+'&Usuario='+document.proyectos.cod_responsable.value,'Insertar_Obra','width=5 height=5');
        }
    }
    else
    {
        alert('No hay obra que guardar');
    }
}

```

```

function habilitar_boton(id_this,usuario)
{
    document.proyectos.id_obra.value=id_this;
    idobra=id_this;

    ban_nueva_obra=0;
    document.proyectos.cod_responsable.value=usuario;
    window.location='/servlet/mapa?Usuario='+usuario;
}

```

/\*\*\*\*\* Si existe una obra activa dentro del mapa,
 es decir que ha sido seleccionada para ver información esta función abre la
 ventana correspondiente al botón seleccionado \*\*\*\*\*/

```

function call_funciones(cual)
{
    if (idobra!=0)
    {
        if (cual=='proyectos')
        {
            window.open('/servlet/ListadoProyectos?codigo_obra='+idobra+'&USUARIO='+document.proyectos.cod_responsable.value,'Insertar_Obra','width=580,height=550,top=25,toolbars=no,scrollbars=yes,status=no,resizable=no,maximize=no');
        }
        if (cual=='needs')
        {
            window.open('/servlet/ListadoNecesidades?codigo_obra='+idobra+'&USUARIO='+document.proyectos.cod_responsable.value,'Insertar_Obra','width=510,height=400,top=25,toolbars=no,scrollbars=yes,status=no,resizable=no,maximize=no');
        }
        if (cual=='info')
        {
            window.open('/servlet/InfoGralObra?codigo_obra='+idobra+'&USUARIO='+document.proyectos.

```

```

cod_responsable.value,'Insertar_Obra','width=500,height=230,top=25,toolbars=no,scrollbars=yes,status=no
,resizable=no,maximize=no');
    }
}

```

```

function parametros(evt,id_obra_existente)
{
    document.proyectos.id_obra.value=id_obra_existente;
    idobra=id_obra_existente;
}

```

```

/***** Función que permite el efecto de encender los botones
al pasar el mouse sobre ellos, esto si hay alguna obra seleccionada (activa)
*****/

```

```

function BtnOver(imgobj,imagen)
{
    if (idobra!=0)
    {
        imgobj.src='/HTML/images/Botones/'+imagen;
    }
    else
    {
        if(ban_nueva_obra==0 && imgobj.name=='newobra')
        {
            imgobj.src='/HTML/images/Botones/'+imagen;
        }
        else if(ban_nueva_obra==1 && (imgobj.name=='newobra' || imgobj.name=='save'))
        {
            imgobj.src='/HTML/images/Botones/'+imagen;
        }
    }
}

```

```

/***** Función que permite el efecto de apagar los botones
al pasar el mouse sobre ellos, esto si hay alguna obra seleccionada (activa)
*****/

```

```

function BtnOut(imgobj,imagen)
{
    if (idobra!=0)
    {
        imgobj.src='/HTML/images/Botones/'+imagen;
    }
    else
    {
        if(ban_nueva_obra==0 && imgobj.name=='newobra')
        {

```

```

        imgobj.src='/HTML/images/Botones/'+imagen;
    }
    else if(ban_nueva_obra==1 && (imgobj.name=='newobra' || imgobj.name=='save'))
    {
        imgobj.src='/HTML/images/Botones/'+imagen;
    }
}
}
}

```

/\*\*\*\*\* Esta función permite el llenado de la lista de datos en la
 pagina de criterios de filtrado \*\*\*\*\*/

```

function cambiar(Usuario)
{
    var parametros="";
    var parametros2="";
    var repdpto='-1';
    var reptipo='-1';
    var repinst='-1';

    // if(document.consulta.Instituciones.options[0].selected==false)
    // {

        parametros+'&Instituciones='+document.consulta.Instituciones.options[document.consulta.Instituciones.options.selectedIndex].value;
        parametros=parametros+'&Departamentos=-1';
        parametros=parametros+'&TipoObra=-1';

        repinst=document.consulta.Instituciones.options[document.consulta.Instituciones.options.selectedIndex].value;;

    // }
    //else
    // {
        if(document.consulta.Departamentos.options[0].selected==false)
        {

            parametros+'&Departamentos='+document.consulta.Departamentos.options[document.consulta.Departamentos.options.selectedIndex].value;
            parametros=parametros+'&Instituciones=-1';
            parametros=parametros+'&TipoObra=-1';

            repdpto=document.consulta.Departamentos.options[document.consulta.Departamentos.options.selectedIndex].value;

        }
        //else
        // {
            if(document.consulta.TipoObra.options[0].selected==false)
            {

```

```

        parametros+'&TipoObra='+document.consulta.TipoObra.options[document.consulta.TipoObra.opt
ions.selectedIndex].value;
        parametros=parametros+'&Departamentos=-1';
        parametros=parametros+'&Instituciones=-1';

        reptipo=document.consulta.TipoObra.options[document.consulta.TipoObra.options.selectedIndex]
.value;

        }
    //}
//}

parametros2="&Instituciones="+replast+"&Departamentos="+repldpto+"&TipoObra="+repltipo;

//        if(parametros!="")
//        if(replast!='-1' && repldpto!='-1' && repltipo!='-1')
//        {

        window.location="/servlet/mapaReportes?USUARIO="+Usuario+"&Filtrar=1"+parametros2;
//        }

}

```

/\*\*\*\*\*\* Llena los combos de Departamentos, municipios y cantones al ser seleccionado en la pantalla de captura de obras \*\*\*\*\*/

```

function llenar_combos(Usuario,tipo)
{
    if (tipo=="d")
    {
        if
(document.Combos.selec_depto.options[document.Combos.selec_depto.selectedIndex].value!="-1")
        {

            window.location="/servlet/mapa?USUARIO="+Usuario+"&Departamento="+document.Combos.
selec_depto.options[document.Combos.selec_depto.selectedIndex].value+"&Municipio=-1";
        }
        else
        {
            if (tipo=="m")
            {
                if
(document.Combos.selec_municip.options[document.Combos.selec_municip.selectedIndex].value!="-1")
                {

                    window.location="/servlet/mapa?USUARIO="+Usuario+"&Departamento="+document.Combos.
selec_depto.options[document.Combos.selec_depto.selectedIndex].value+"&Municipio="+document.Com
bos.selec_municip.options[document.Combos.selec_municip.selectedIndex].value;
                }
            }
        }
    }
}

```

```

    }
}

/***** funciones para abrir ventanas independientes
*****/
function WinOpen(url)
{
    msg=open(url,'DisplayWindow','toolbar=no,directories=no,menubar=no,resizable=no,scrollbars=yes,WIDTH=500 HEIGHT=150, status=no,location=no');
}

function WinOpenReporte(codigo)
{
    var
url='/servlet/Reportes?Codigo='+codigo+'&institucion='+institucion+'&parametro='+parametro+'&anio='+
anio+'&USUARIO='+usuario;

    msg=open(url,'DisplayWindow','toolbar=no,directories=no,menubar=no,resizable=no,scrollbars=yes,WIDTH=400,HEIGHT=550,status=no,location=no,left=740,top=185');
}

/***** función para cargar la fecha del sistema y
poder mostrarlo en el encabezado de cada página
*****/

function MakeArray(n)
{
    this.length = n
    return this
}

monthNames = new MakeArray(12);
monthNames[1] = 'Enero';
monthNames[2] = 'Febrero';
monthNames[3] = 'Marzo';
monthNames[4] = 'Abril';
monthNames[5] = 'Mayo';
monthNames[6] = 'Junio';
monthNames[7] = 'Julio';
monthNames[8] = 'Agosto';
monthNames[9] = 'Septiembre';
monthNames[10] = 'Octubre';
monthNames[11] = 'Noviembre';
monthNames[12] = 'Diciembre';

function customDateString(oneDate)
{
    var theMonth = monthNames[oneDate.getMonth() + 1];
    var theYear = oneDate.getYear();
    theYear += (theYear < 100) ? 1900 : 0;
    return oneDate.getDate() + " de " + theMonth + " de " + theYear;
}

```

```
/****** Funciones que permiten la interactividad del usuario
con los menus del sistema *****/
```

```
menpos_x=45;
menpos_y=45;
menpix=13;
```

```
/****** Muestra Menus *****/

function show(object,padre,x,y,l)
{
    if (document.getElementById)
    {
        menpos_x=document.getElementById(padre).style.posLeft;
        document.getElementById(object).style.posLeft= menpos_x;

document.getElementById(object).style.posTop=document.getElementById(padre).style.posTop+
15;
        document.getElementById(object).style.visibility = 'visible';
    }
    else if (document.layers)
    {
        document.layers[object].left=document.layers[padre].left;
        document.layers[object].Top=document.layers[padre].Top+23;
        document.layers[object].visibility = 'visible';
    }
    else if (document.all)
    {
        document.all[object].style.posLeft = document.all[padre].style.posLeft;
        document.all[object].style.posTop = document.all[padre].style.posTop+23;
        document.all[object].style.visibility = 'visible';
    }
}

/****** Muestra Menus Hijos *****/
```

```
function show2(object,padre,x,y,l)
{
    if (document.getElementById)
    {
        menpos_x=document.getElementById(padre).style.posLeft + 121;
        document.getElementById(object).style.posLeft= menpos_x;

document.getElementById(object).style.posTop=document.getElementById(padre).style.posTop+
(y*menpix);
        document.getElementById(object).style.visibility = 'visible';
    }
    else if (document.layers)
    {
        document.layers[object].left=document.layers[padre].left;
        document.layers[object].Top=document.layers[padre].Top+23;
        document.layers[object].visibility = 'visible';
    }
}
```

```

    }
    else if (document.all)
    {
        document.all[object].style.posLeft = document.all[padre].style.posLef;
        document.all[object].style.posTop = document.all[padre].style.posTop+23;
        document.all[object].style.visibility = 'visible';
    }
}

```

/\*\*\*\*\*\* Esconde Menus \*\*\*\*\*\*/

```

function hide(object,padre)
{
    if (document.getElementById)
    {
        document.getElementById(object).style.visibility = 'hidden';
        if (padre!='0')
        {
            document.getElementById(padre).style.visibility = 'hidden';
        }
    }
    else if (document.layers && document.layers[object])
    {
        document.layers[object].visibility = 'hidden';
        if (padre!='0')
        {
            document.layers[padre].visibility = 'hidden';
        }
    }
    else if (document.all)
    {
        document.all[object].style.visibility = 'hidden';
        if (padre!='0')
        {
            document.all[padre].style.visibility = 'hidden';
        }
    }
}

```

/\*\*\*\*\*\* Validación de datos \*\*\*\*\*\*/

```

/* No permite campos vacios en captura o actualización de datos */
function NoNulos(myform)
{
    var mensaje="";
    var i=0;
    var n=0;
    for (i=0,n=myform.elements.length;i<n-1;i++)
    {
        if(myform.elements[i].type!='hidden')
        {
            if (myform.elements[i].value=="")

```

```

        {
            mensaje=mensaje+'El campo '+myform.elements[i].name+' no
puede ser vacio\n';
        }
    }

    if (mensaje!="")
    {
        alert(mensaje);
        return false;
    }
    else
    {
        return true;
    }
}

```

/\* Valida que los datos sean decimales en captura o actualización de datos \*/

```

function Decimales(obj)
{
    punto=0;
    var j=0;
    var i=0;
    var result="";
    var NumValue=obj.value;
    if (NumValue!="")
    {
        for (i=0;i<NumValue.length;i++)
        {
            if (NumValue.charAt(i)=='.')
            {
                if(i==0)
                {
                    j=i+1;
                }
                else
                {
                    j=i;
                }
            }
            if (j!=0 && NumValue.charAt(i)!='.')
            {
                result= result+ NumValue.charAt(i);
            }
        }

        if (j!=0)
        {
            if (result.length==0 || result.length>2)
            {
                alert('Ingrese por lo menos un decimal, máximo 2\n');
                obj.focus();
                return false;
            }
        }
    }
}

```

```

    }
    }
    else
    {
        alert("Debe ingresar Datos");
        obj.focus();
        return false;
    }
}

/***** No permite el ingreso de letras o cualquier otro
tipo de caracteres especiales si es un campo numérico
*****/

function Numeros(field)
{
    if (field.value!="")
    {
        if(field.value.indexOf('.')!=-1)
        {
            punto=1;
        }
    }
    if ((punto==0 && event.keyCode==46))
    {
        punto=1;
    }
    else
    {
        if ((event.keyCode > 32 && event.keyCode < 48) || (event.keyCode > 57 &&
event.keyCode < 65) || (event.keyCode > 90 && event.keyCode < 97)) event.returnValue = false;
        if (event.keyCode < 45 || event.keyCode > 57) event.returnValue = false; if
(event.keyCode==34 || event.keyCode==39) event.returnValue = false;
    }
}

/* Solo permite datos enteros en captura o actualización de datos */

function Enteros(campo)
{
    if (campo.value.indexOf('.')!=-1)
    {
        alert("El valor debe ser entero");
        campo.focus();
        return false;
    }
    else return true;
}

/* Necesita un año válido para el ingreso de Seguimiento, no permite repetido */

```

```

function ValidAnio(campo,exisanio,obj)
{
    var i=0;
    var sianio=0;
    if(Enteros(campo))
    {
        if(campo.value<1986 || campo.value>2010)
        {
            alert("Escriba un año válido");
            campo.focus();
            return(false);
        }
        else
        {
            if (exisanio==1 && obj!="")
            {
                for (i=0;i<obj.length;i++)
                {
                    obj.options[i].value==campo.value;
                    sianio=1;
                    break;
                }
            }
            if(sianio==1)
            {
                sianio=0;
                alert("El año ingresado ya existe");
                return false;
            }
        }
    }
}

/***** VALIDACION DE CARACTERES *****/
function ValidCaracter(CAMPO)
{
    var checkOK = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzŠŽšžŸÀÁÂÃÄÅÆÇÈÉÊËÏÎÏÑÒÓÔÕÖÙÚÛÜÝÞàáâãäåæçèéêëìíîïðñòóôõöùúûüýþ";
    var checkStr = CAMPO.value;
    var allValid = true;
    for (i = 0; i < checkStr.length; i++)
    {
        ch = checkStr.charAt(i);
        for (j = 0; j < checkOK.length; j++)
            if (ch == checkOK.charAt(j))
                break;
        if (j == checkOK.length)
        {
            allValid = false;
            break;
        }
    }
    if (!allValid)

```

```

    {
    alert('Por favor Introduzca solamente letras');
    CAMPO.focus();
    CAMPO.value = "";
    }
}

```

/\*\*\*\*\*\* VALIDACION DE NUMEROS \*\*\*\*\*/

```

function ValidNumero(CAMPO)
{
var checkOK = "0123456789.";
var checkStr = CAMPO.value;
var allValid = true;
var decPoints = 0;
var allNum = "";
for (i = 0; i < checkStr.length; i++)
{
ch = checkStr.charAt(i);
for (j = 0; j < checkOK.length; j++)
if (ch == checkOK.charAt(j))
break;
if (j == checkOK.length)
{
allValid = false;
break;
}
if (ch != ",")
allNum += ch;
}
if (!allValid)
{
alert('Por favor introduza solo cantidades numéricas');
CAMPO.focus();
CAMPO.value = "";
}
}

```

/\*\*\*\*\*\* VALIDACION DE CAMPOS FECHA \*\*\*\*\*/

```

function ValidFecha(campo)
{
var checkStr = campo.value;
var dia = checkStr.substring(0,2);
var mes = checkStr.substring(3,5);
var ano = checkStr.substring(6,10);
if (EsNumero(campo.value)==false)
{
alert('El campo solo admite digitos');
campo.value="";
campo.focus()
}
else
{
if (EsFecha(dia,mes,ano)>0)

```

```

    {
        alert('La fecha no es valida');
        campo.select()
    }
}
function EsNumero(checkStr)
{
    var checkOK = "-/0123456789";
    var allValid = true;
    for (i = 0; i < checkStr.length; i++)
    {
        ch = checkStr.charAt(i);
        for (j = 0; j < checkOK.length; j++)
            if (ch == checkOK.charAt(j))
                break;
        if (j == checkOK.length)
        {
            allValid = false;
            break;
        }
    }
    if (!allValid)
    {
        return (false);
    }
    return (true);
}

function EsFecha(Dia, Mes,Ano)
{
    var Bisiesto=Ano%4
    if (Dia==" && Mes==" && Ano==" )
    {
        return 0;
    }
    if (Dia!=" && (Mes==" || Ano==" ) )
    {
        return 1;
    }
    if (Ano.length<4)
    {
        return 1;
    }
    if (Mes==1 || Mes==3 || Mes==5 || Mes==7 || Mes==8 || Mes==10 || Mes==12)
    {
        if (Dia<1 || Dia>31)
        {
            return 2;
        }
    }
    else
    {
        if (Mes==4 || Mes==6 || Mes==9 || Mes==11)
        {
            if (Dia<1 || Dia>30)

```

```

    {
    return 2;
    }
    }
    else
    {
if (Mes==2)
{
if (Bisiesto==0)
{
if (Dia<1 || Dia>29)
{
return 2;
}
}
else
{
if (Dia<1 || Dia>28)
{
return 2;
}
}
}
}
else
{
return 3;
}
}
}
return 0;
}

```

## Hoja de estilo (ourStyle.css)

Hoja de estilo utilizada para darle formato y color a toda la aplicación:

```

BODY
{
    FONT-SIZE: 11px;
    COLOR: #000000;
    BACKGROUND-COLOR:#999999;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
    MARGIN-TOP:0px;
    MARGIN-LEFT:0px;
    MARGIN-RIGHT:0px;
}
A:link
{
    FONT-SIZE: 11px;
    COLOR: #FFFFFF;
    TEXT-DECORATION: underline;
}
A:visited
{

```

```

    FONT-SIZE: 11px;
    COLOR: #003366;
    TEXT-DECORATION: underline;
}
A:hover
{
    FONT-SIZE: 11px;
    COLOR: 003366;
    TEXT-DECORATION: underline;
}

P
{
    LINE-HEIGHT: 15px;
}

H1
{
    FONT-SIZE: 12pt;
    COLOR: #333333;
}

H2
{
    FONT-SIZE: 14pt;
    COLOR: #333333;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
}

TD
{
    FONT-SIZE: 11px;
    COLOR: #000000;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
}
INPUT
{
    FONT-SIZE: 10px;
    FONT-STYLE: normal;

    FONT-FAMILY: Arial, Helvetica, sans-serif;
    FONT-VARIANT: normal;
}
INPUT.sinborde
{
    BORDER-RIGHT: white 0px;
    BORDER-TOP: white 0px;
    FONT-SIZE: 10px;
    BACKGROUND: white;
    BORDER-LEFT: white 0px;
    BORDER-BOTTOM: white 0px;
    FONT-STYLE: normal;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
    FONT-VARIANT: normal;
}

```

```

SELECT
{
  FONT-SIZE: 10px;
  FONT-STYLE: normal;
  FONT-FAMILY: Arial, Helvetica, sans-serif;
  COLOR: #000000;
  BACKGROUND-COLOR: #FFFFFF;
  FONT-VARIANT: normal;
}
BUTTON
{
  FONT-SIZE: 10px;
  BACKGROUND-IMAGE: none;
  BACKGROUND-REPEAT: repeat;
  FONT-STYLE: normal;
  FONT-FAMILY: Georgia, "Times New Roman", Times, serif;
  FONT-VARIANT: normal;
}
TEXTAREA
{
  FONT-SIZE: 10px;
  FONT-STYLE: normal;
  FONT-FAMILY: Arial, Helvetica, sans-serif;
  FONT-VARIANT: normal;
}
TEXTAREA.sinborde
{
  BORDER-RIGHT: white 0px;
  BORDER-TOP: white 0px;
  FONT-SIZE: 10px;
  BACKGROUND: white;
  BORDER-LEFT: white 0px;
  COLOR: #003399;
  BORDER-BOTTOM: white 0px;
  FONT-STYLE: normal;
  FONT-FAMILY: Arial, Helvetica, sans-serif;
  FONT-VARIANT: normal;
}
OPTION
{
  FONT-SIZE: 10px;
  FONT-STYLE: normal;
  FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
  FONT-VARIANT: normal;
}
A.linkblanco:link
{
  FONT-SIZE: 11px;
  COLOR: #FFFFFF;
  FONT-FAMILY: Arial, Helvetica, sans-serif;
  TEXT-DECORATION: underline;
}
A.linkblanco:visited
{
  FONT-SIZE: 11px;
  COLOR: #FFFFFF;
}

```

```

    FONT-FAMILY: Arial, Helvetica, sans-serif;
    TEXT-DECORATION: none;
    FONT-WEIGHT:bold;
}
A.linkblanco:hover
{
    FONT-SIZE: 11px;
    COLOR: white;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
    TEXT-DECORATION: underline;
}
A.linkamarillo:link
{
    FONT-SIZE: 11px;
    COLOR: #ffcc00;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
    TEXT-DECORATION: underline;
}
A.linkamarillo:visited
{
    FONT-SIZE: 11px;
    COLOR: #ffcc00;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
    TEXT-DECORATION: underline;
}
A.linkamarillo:hover
{
    FONT-SIZE: 11px;
    COLOR: #ffcc00;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
    TEXT-DECORATION: underline;
}

INPUT.letragris
{
    FONT-SIZE: 10px;
    COLOR: gray;
    FONT-STYLE: normal;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
    FONT-VARIANT: normal
}

.title{
    font-family: Arial, Verdana, Helvetica, sans-serif;
    color:#000000;
    font-weight:bolder;
    font-size:15px;
    FONT-VARIANT: normal;
}
.title1{
    font-family: Arial, Verdana, Helvetica, sans-serif;
    color:#006699;
    font-weight:bolder;
    font-size:12px;
    FONT-VARIANT: normal;
}

```

```

.row1{
    background-color:#CCFFDD;
}

.row2{
    background-color:#DDFFCC;
}

.row3{
    background-color:#99CCFF;
}

hr{
    border-style : dotted;
    border : ridge;
}

A.est:link
{
    FONT-SIZE: 11px;
    COLOR: #000000;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
    TEXT-DECORATION: none
}
A.est:visited
{
    FONT-SIZE: 11px;
    COLOR: #000000;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
    TEXT-DECORATION: none
}
A.est:hover
{
    FONT-SIZE: 11px;
    COLOR: #555555;
    FONT-FAMILY: Arial, Helvetica, sans-serif;
    TEXT-DECORATION: none
}

    .tableMen{
        border-style: solid;
        background: #CCCCCC;
        border-color: #000000;
        border-width: thin;
        padding: 0px;
    }
    .menuBoxPP {
        visibility:visible;
        position:absolute;
        overflow:hidden;
    }

    .menuBox {
        position:absolute;

```

```

        visibility:hidden;
        font-family: verdana,arial,sans-serif;
        overflow:hidden;
        color:#000000;
        font-size:13;
    }
    a.submenu:link,a.submenu:visited{
        color:#000000;
        text-decoration:none;
        font-size:13;
        font-weight:bold;
    }

    a.submenu:hover{
        color:#FFFFFF;
        text-decoration:none;
        font-size:13;
        font-weight:bolder;
    }
    a.menu:link,a.menu:visited{
        color:#000000;
        text-decoration:none;
        font-size:13;
        font-weight:bold;
    }
    a.menu:hover{
        color:#000000;
        text-decoration:none;
        font-size:13;
        font-weight:bolder;
    }

```

```
#menuAnchor { position:relative; visibility:hidden;}
```

```
body.svg
{
    scrollbar-face-color: #104A7B;
    scrollbar-highlight-color: #FFFFFF;
    scrollbar-3dlight-color: #FFFFFF;
    scrollbar-darkshadow-color: #93BEE2;
    scrollbar-arrow-color: #93BEE2;
    scrollbar-track-color: #93BEE2;
}

```

```
.colortable
{
    background: #CCCCCC;
    border-style: solid;
    border-width: double;
    padding: 6px 6px 6px 6px ;
    spacing: 6px 6px 6px 6px ;
}

```

```
.colortable2
{

```

```

        background: #DDDDDD;
        border-style: solid;
        border-width: thin;
        padding: 6px 6px 6px 6px ;
        spacing: 6px 6px 6px 6px ;
    }

.LaFecha
{
    FONT-SIZE: 11px;
    COLOR: #000000;
}

.tdfont
{
    FONT-SIZE: 12px;
    COLOR: #FFFFFF;
    BACKGROUND-COLOR: #669966;
}

.tdfontletra
{
    FONT-SIZE: 12px;
    COLOR: #000000;
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    BACKGROUND-COLOR: #DDDDDD;
}

.fontmensajes
{
    FONT-SIZE: 12px;
    COLOR: #000000;
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    FONT-WEIGHT:bold;
}

.trbotones
{
    FONT-SIZE: 12px;
    COLOR: #FFFFFF;
    FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
    BACKGROUND: #999999;
}

.titulo{
    font-family: Arial, Verdana, Helvetica, sans-serif;
    color:#000000;
    font-weight:bolder;
    font-size:13px;
    FONT-VARIANT: normal;
}

.tcolor{
    background-color:#40AD65;

```

```

        border-color:#40AD65;
        border-style :double;
    }
    .tcolor2{
        border-color:#CCFFDD;
        border-style :double;
    }
    .hcolor{
        BACKGROUND-COLOR: #339966;
        font-weight:bold;
        color:#FFFFFF;
        text-align:center;
        padding: 3px;
        padding-left:5px;
        padding-right:5px;
    }
    .fcolor{
        font-weight:bold;
        color:#FFFFFF;
    }
}

```

## SVG

Se han diseñado y utilizado dos archivos SVG, que brindan la funcionalidad necesaria para las siguientes áreas:

- *Mapa SVG para captura*  
(mapa\_15Feb\_captura.svg)
- *Mapa SVG para captura*  
(mapa\_15Feb\_reportes.svg)

Ambos guardan similitudes esenciales. Son distintos en pequeñas, pero importantes secciones. Por ello, a continuación se detallan partes de su organización.

### *Organización básica de los archivos SVG.*

#### 1) Sección de Paneles.

Se incluyen en esta sección los paneles utilizados para organizar los distintos controles del mapa. Basicamente se utilizan 4 paneles:

- Panel de Selección de obras (disponible en "captura de obras" y "reportes").  
Se listan todos los grupos de obras almacenados en la base de datos, acompañados de un checkbox para ver u ocultar el grupo. La estructura básica del panel es:

```

<g id="panelSelecObra" x="50" >
  <rect x="756.8" y="0" width="140" height="570" style="fill:#CCCC99"/>
  <g id="check" transform="translate(140,0)" style="visibility:visible;">

  //////////ACA SE INCLUYEN TODOS LOS GRUPOS //////////
  ////////////////////DE OBRAS, CON SU CHECKBOX//// //////////////

  </g>
</g>

```

- Panel para agregar obras (disponible en "captura de obras"). Se listan todas los tipos de obras almacenadas en la base de datos. Cuando el usuario desea adherir una nueva obra al mapa, accede a este panel, hace clic en la obra deseada para seleccionarla, y la coloca sobre el mapa. La estructura general de este panel es la siguiente:

```

<g id="panelUbicObra">
  <rect x="616.8" y="0" width="140" height="570" style="fill:#AACC99"/>
  <g id="simbl">

  //////////ACA SE INCLUYEN TODAS LAS OBRAS //////////
  //////////REGISTRADAS EN LA BASE DE DATOS//////////

  </g>
</g>

```

- Panel de reportes (disponible en "reportes"). Lista dos reportes útiles para el usuario:
  - o Avance Financiero.
  - o Avance Físico.

Cuando se hace click sobre los enlaces a cada uno de los reportes mencionados, se cambias propiedades SVG del Mapa. El panel está estructurado esencialmente de la siguiente forma:

```

<g id="panelRepObra">
  <rect x="616.8" y="0" width="140" height="570" style="fill:#AACC99"/>
  <g id="reportes" transform="translate(644,50)">
  ////////////ESTE ES EL LINK AL REPORTE DE AVANCE FINANCIERO//////////
  <a><text style="font-size:11;" onclick="colorear_finan();WinOpenReporte('01');">
    <tspan x="0" y="0">Avance Financiero</tspan></text></a>
  <rect id="rep_finan" x="-5" y="-12" width="100" height="15"
    style="visibility:hidden;fill:none;fill-opacity=0;stroke:black;stroke-width:1;"/>

  ////////////ESTE ES EL LINK AL REPORTE DE AVANCE FISICO//////////
  <a><text style="font-size:11;" onclick="colorear_fisico();WinOpenReporte('02');">
    <tspan x="0" y="30">Avance Físico</tspan></text></a>
  <rect id="rep_fisico" x="-5" y="18" width="100" height="15"
    style="visibility:hidden;fill:none;fill-opacity=0;stroke:black;stroke-width:1;"/>
  </g>
</g>

```

- Panel de Navegación.  
Incluye todos los elementos orientados a la navegación por el mapa:

- o Barra de zoom.

Control destinado al cambio de niveles de visualización del mapa. Su estructura en código SVG es la siguiente:

```
<g transform="translate(660,15) scale(0.75)" id="zoombarlevel" style="fill-rule:nonzero;clip-rule:nonzero;fill:#DDDDDD;stroke:rgb(0,0,0);stroke-miterlimit:4;font-family:'AbadiMT-CondensedExtraBold';font-size:11">
  <path id="z25" style="stroke:none" d="M0.81 99.14 L0.8 113.66 L29.29 113.64 L30.02 99.14 z"/>
  <path id="z50" style="stroke:none" d="M0.81 85.43 L0.81 99.14 L30.02 99.14 L32.5 85.46 z"/>
  <path id="z100" style="stroke:none" d="M0.81 71.52 L0.81 85.58 L32.5 85.46 L36.7 71.52 z"/>
  <path id="z200" style="stroke:none" d="M0.81 57.04 L0.81 71.52 L36.7 71.52 L43 57.1 z"/>
  <path id="z400" style="stroke:none" d="M0.61 42.86 L0.8 57.04 L43 57.1 L51.4 42.86 z"/>
  <path id="z600" style="stroke:none" d="M0.81 28.68 L0.61 42.86 L51.4 42.86 L62.16 28.68 z"/>
  <path id="z800" style="stroke:none" d="M0.81 14.57 L0.81 28.68 L62.09 28.68 L75.45 14.57 z"/>
  <path id="z1000" style="stroke:none" d="M0.81 0.05 L93.44 0.05 L75.64 14.57 L0.81 14.57 z"/>
  <a>
    <text onmouseover="ver(evt)" onmouseout="nover(evt)" onclick="cambiarzoom(10)"> <tspan x="3px" y="10px" style="stroke:none;fill:red">1000%</tspan></text>
    <text onmouseover="ver(evt)" onmouseout="nover(evt)" onclick="cambiarzoom(8)"> <tspan x="3px" y="24px" style="stroke:none;fill:red">800%</tspan></text>
    <text onmouseover="ver(evt)" onmouseout="nover(evt)" onclick="cambiarzoom(6)"> <tspan x="3px" y="38px" style="stroke:none;fill:red">600%</tspan></text>
    <text onmouseover="ver(evt)" onmouseout="nover(evt)" onclick="cambiarzoom(4)"> <tspan x="3px" y="52px" style="stroke:none;fill:red">400%</tspan></text>
    <text onmouseover="ver(evt)" onmouseout="nover(evt)" onclick="cambiarzoom(2)"> <tspan x="3px" y="66px" style="stroke:none;fill:red">200%</tspan></text>
    <text onmouseover="ver(evt)" onmouseout="nover(evt)" onclick="cambiarzoom(1)"> <tspan x="3px" y="80px" style="stroke:none;fill:red">100%</tspan></text>
    <text onmouseover="ver(evt)" onmouseout="nover(evt)" onclick="cambiarzoom(0.5)"> <tspan x="3px" y="94px" style="stroke:none;fill:red">50%</tspan></text>
    <text onmouseover="ver(evt)" onmouseout="nover(evt)" onclick="cambiarzoom(0.25)"> <tspan x="3px" y="108px" style="stroke:none;fill:red">25%</tspan></text>
  </a>
</g>
```

- o Control de desplazamiento.

Constituído por 13 controles que manipulan el desplazamiento del mapa a lo largo de la pantalla de visualización. Su estructura SVG es:

```
<g id="navegador" transform="translate(650,110)" style="fill-rule:nonzero;clip-rule:nonzero;fill:#FDFFCA;stroke:#000000;stroke-miterlimit:4;">
  <a>
    <path id="centro_nav" style="stroke:none" onclick="Ir(307.5,312.5)" onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M41.32 34.22 C41.32 38.33 37.99 41.65 33.89 41.65 C29.78 41.65 26.46 38.32 26.46 34.22 C26.46 30.11 29.79 26.79 33.89 26.79 C38 26.79 41.32 30.12 41.32 34.22 z"/>
    <path id="der1_nav" style="stroke:none" onclick="desplazar('der',1)" onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M40.68 27.82 C40.68 27.82 46.53 34.35 40.43 40.88 L49.39 34.29 L40.68 27.83 z"/>
  </a>
</g>
```

```

    <path id="der2_nav" style="stroke:none" onclick="desplazar('der',2)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M42.24 25.8 L50.83 34.19
L42.21 42.49 L60.25 34.04 z"/>
    <path id="der3_nav" style="stroke:none" onclick="desplazar('der',3)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M43.71 24.57 L64.24 34
L43.47 43.73 L44.62 44.91 L68 33.97 L44.92 23.42 L43.71 24.56 z"/>
    <path id="izq1_nav" style="stroke:none" onclick="desplazar('izq',1)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M27.27 40.63 C27.27
40.63 21.42 34.1 27.52 27.57 L18.56 34.16 L27.27 40.62 z"/>
    <path id="izq2_nav" style="stroke:none" onclick="desplazar('izq',2)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M25.36 42.39 L16.77 34
L25.39 25.7 L7.35 34.15 z"/>
    <path id="izq3_nav" style="stroke:none" onclick="desplazar('izq',3)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M24.3 43.38 L3.76 33.95
L24.53 24.22 L23.38 23.04 L0 33.97 L23.08 44.52 L24.29 43.38 z"/>
    <path id="bajar1_nav" style="stroke:none" onclick="desplazar('abajo',1)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M40.51 40.89 C40.51
40.89 33.98 46.74 27.45 40.64 L34.04 49.6 L40.5 40.89 z"/>
    <path id="bajar2_nav" style="stroke:none" onclick="desplazar('abajo',2)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M42.17 42.5 L33.78 51.09
L25.48 42.47 L33.93 60.51 z"/>
    <path id="bajar3_nav" style="stroke:none" onclick="desplazar('abajo',3)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M43.45 43.64 L34.02 64.17
L24.3 43.41 L23.12 44.56 L34.05 67.95 L44.6 44.87 L43.46 43.66 z"/>
    <path id="subir1_nav" style="stroke:none" onclick="desplazar('arriba',1)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M27.37 27.47 C27.37
27.47 33.9 21.62 40.43 27.72 L33.84 18.76 L27.38 27.47 z"/>
    <path id="subir2_nav" style="stroke:none" onclick="desplazar('arriba',2)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M25.48 25.79 L33.87 17.2
L42.17 25.82 L33.72 7.77 z"/>
    <path id="subir3_nav" style="stroke:none" onclick="desplazar('arriba',3)"
onmouseover="changeCol(evt,#FFCC00)" onmouseout="regresaCol(evt)" d="M24.49 24.3 L33.92 3.77
L43.65 24.54 L44.83 23.39 L33.89 0 L23.34 23.08 L24.48 24.29 z"/>
  </a>
</g>

```

- Mapa de orientación.

Constituido por una versión reducida del código SVG que define al mapa principal. Incluye solo el código de los contornos principales de cantones, municipios, etc. No existe detalle de calles, nombres de calle, lugares de referencia, etc. (Ver *Mapa en Sección de Mapa* para ver referencia a la estructura SVG del mapa).

- Visualización de coordenadas.

Un área de texto destinada a mantener rastreo constante de las coordenadas del mapa. Su estructura SVG se presenta en el siguiente código:

```

<g id="postext">
  <g transform="translate(590,250)" style="fill:#FFFFFF;">
    <text id="XPos" x="50" y="130">Posicion X :</text>
    <text id="YPos" x="50" y="150">Posicion Y :</text>
    <text id="ClickPos" x="50" y="170">Ultimo Click :</text>
  </g>
</g>

```

</g>

- Botones de acceso a otros paneles.

Permiten navegar entre los distintos paneles disponibles. Todos guardan la misma estructura, que se presenta a continuación para un solo botón, "btnUbic":

```
<g id="btns" transform="translate(618,475) scale(1.25)">
  <g style="fill-rule:nonzero;clip-rule:nonzero;fill:none;stroke:#FFFFFF;stroke-
width:2;stroke-miterlimit:4; font-family:'ArialMT'; font-size:9.0065;">
    <g id="btnUbic">
      <g onclick="verPanelU()">
        <g id="UbicOcultar">
          <path d="M98.88,3.571-5.05,5.2115.35,5.35"/>
          <path d="M105.14,3.39L100.1,8.615.35,5.35"/>
          <path style="stroke:#FFFFFF;"
            d="M98.04,3.661-5.05,5.2115.35,5.35"/>
          <path style="stroke:#FFFFFF;"
            d="M104.29,3.481-5.05,5.2115.35,5.35"/>
        </g>
        <g id="UbicMostrar">
          <path d="M99.54,3.5415.05,5.211-5.35,5.35"/>
          <path d="M93.28,3.3615.05,5.211-5.35,5.35"/>
          <path style="stroke:#FFFFFF;"
            d="M100.39,3.6315.05,5.211-5.35,5.35"/>
          <path style="stroke:#FFFFFF;"
            d="M94.13,3.4515.05,5.211-5.35,5.35"/>
        </g>
      </g>
      <g>
        <text id="txt_ubic1" transform="matrix(1 0 0 1 3.0444 11.9541)"
          style="fill:#FFFFFF;stroke:none;" visibility="visible">
          <tspan x="0" y="0">Agregar Obras</tspan></text>
        <text id="txt_ubic2" transform="matrix(1 0 0 1 3.0444 11.9541)"
          style="fill:#FFFFFF;stroke:none;" visibility="hidden">
          <tspan x="0" y="0">Seleccion de obras</tspan></text>
        <g style="stroke:#FFFFFF;stroke-width:1;">
          <path d="M110.18,16.97H0.75V0.5h109.43v16.47z"/>
          <path d="M109.93,16.99H88.4V0.53h21.53v16.47z"/>
        </g>
      </g>
    </g>
  </g>
</g>
```

## 2) Sección de Mapa.

Es esta sección del código SVG se incluyen todos los elementos necesarios para mostrar información en el mapa. Estos componentes se agrupan en:

- Mapa base. Contornos de cantones, municipios, calles, nombres, etc. Esta parte es la que contiene más información dentro del código SVG. Su estructura básica es la siguiente:

```

<svg id_pais estilo_propiedades_eventos_metodos_utilizados ">
<Definición de scripts utilizados/>
<g id="id_departamento" estilo_eventos_metodos_utilizados >
  <path path_del_departamento>
    <g id="id_municipio" estilo_eventos_metodos_utilizados >
      <path path_del_municipio>
        <g id="id_canton" estilo_eventos_usados>
          <path path_de_canton>
            elementos_de_canton
            manzanas nombres_calle lugares
          </g>
        <g id="id_canton" estilo_eventos_usados>
          <path path_del_canton>
            elementos del canton
            manzanas nombres_calle lugares
          </g>
        .
        .
      </g>
    .
    .
  </g>
</g>

```

</svg>

- Obras.

Todas las obras introducidas en el sistema por el usuario. Están organizadas dentro de un grupo SVG. Luego de consultar la base de datos se introducen en esta área:

```

<g id="obras">
  //////////SE ORGANIZAN LAS OBRAS EN GRUPOS////////////////////////////////////
  //////////ACA SE INCLUYEN LOS TEXTOS DE LAS OBRAS////
  //////////SE INCLUYE ADEMAS LA IMAGEN DE LA OBRA////
</g>

```

- Tooltips.

Se incluyen acá los cuadros informativos con datos acerca de las obras. Estos se muestran u ocultan cuando el usuario ubica el puntero del ratón sobre o fuera del area correspondiente a la imagen de una obra. Estan organizados de la siguiente forma:

```

<g id="toolt">
  //////////////////////////////////TEXTO DEL TOOLTIP////////////////////////////////////
  //////////ESTRUCTURA GRAFICA DEL CUADRO (relleno, outline)////
</g>

```

- Reporte financiero.

Información oculta, utilizada para generar los repotes gráficos de avance financiero.

```
<g id="repfn">
    ///REFERENCIA A CADA CANTON y su información financiera///
</g>
```

- Reporte fisico.

Información oculta, utilizada para generar los repotes gráficos de avance fisico.

```
<g id="repfs">
    ///REFERENCIA A CADA CANTON y su información financiera///
</g>
```

## Procedimientos Almacenados y Triggers

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure AvanceFinancieroObras @institucion int ,@obra int ,@anio int
,@usuario varchar(10)
as
```

```
select codigo_proyecto,
nombre_proyecto,
convert(varchar(20),'-') as prog,
convert(varchar(20),'-') as ejec,convert(varchar(20),'-') as porc
into #RESULTADOS
from proyectos
where codigo_obra=@obra
```

```
if @institucion<>999
BEGIN
select p.codigo_proyecto,
convert(varchar(20),convert(money,sum(s.programado_financiero)),1) as prog,
convert(varchar(20),convert(money,sum(s.ejecucion_financiero)),1) as ejec,
porc=
    Case sum(s.programado_financiero)
```

```

                When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_financiero)/sum(s.programado_fin
anciero)*100)),1)+'%'
            end
into #RESULTADOS2
from seguimiento s,proyectos p,responsables r
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=@obra
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
and s.año=@año
group by p.codigo_proyecto

update #RESULTADOS set prog=#RESULTADOS2.prog,
ejec=#RESULTADOS2.ejec,
porc=#RESULTADOS2.porc
from #RESULTADOS,#RESULTADOS2
where #RESULTADOS.codigo_proyecto=#RESULTADOS2.codigo_proyecto

END
else
BEGIN
select p.codigo_proyecto,
convert(varchar(20),convert(money,sum(s.programado_financiero)),1) as prog,
convert(varchar(20),convert(money,sum(s.ejecucion_financiero)),1) as ejec,
porc=
    Case sum(s.programado_financiero)
        When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_financiero)/sum(s.programado_fin
anciero)*100)),1)+'%'
    end
into #RESULTADO3
from seguimiento s,proyectos p
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=@obra
and s.año=@año
group by p.codigo_proyecto

update #RESULTADOS set prog=#RESULTADO3.prog,
ejec=#RESULTADO3.ejec,
porc=#RESULTADO3.porc
from #RESULTADOS,#RESULTADO3
where #RESULTADOS.codigo_proyecto=#RESULTADO3.codigo_proyecto

```

END

```
select nombre_proyecto as Proyecto,prog as [Monto Programado],ejec as [Monto
Ejecutado], porc as [Porcentaje de Avance]
from #RESULTADOS
order by codigo_proyecto
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

/\*  
\*\*\*\*\*  
\*/

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure AvanceFinancieroObrasCanton @institucion int ,@canton int
,@anio int ,@usuario varchar(10)
as
```

```
select codigo_obra,
'<a
href=/servlet/Reportes?Codigo=05&USUARIO='+@usuario+'&parametro='+convert(var
char(4),codigo_obra)+'&institucion='+convert(varchar(4),@institucion)+'&anio='+conve
rt(varchar(4),@anio)+'>'+nombre_obra+'</a>' as nombre_obra,
convert(varchar(20),'-') as prog,
convert(varchar(20),'-') as ejec,convert(varchar(20),'-') as porc
into #RESULTADOS
from obras_fisicas
where codigo_canton=@canton
```

```
if @institucion<>999
BEGIN
select o.codigo_obra,
```

```

convert(varchar(20),convert(money,sum(s.programado_financiero)),1) as prog,
convert(varchar(20),convert(money,sum(s.ejecucion_financiero)),1) as ejec,
porc=
    Case sum(s.programado_financiero)
        When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_financiero)/sum(s.programado_financiero)*100)),1)+'%'
    end
into #RESULTADOS2
from seguimiento s,proyectos p,obras_fisicas o,responsables r
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=o.codigo_obra
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
and s.año=@anio
and o.codigo_canton=@canton
group by o.codigo_obra

update #RESULTADOS set prog=#RESULTADOS2.prog,
ejec=#RESULTADOS2.ejec,
porc=#RESULTADOS2.porc
from #RESULTADOS,#RESULTADOS2
where #RESULTADOS.codigo_obra=#RESULTADOS2.codigo_obra

END
else
BEGIN
select o.codigo_obra,
convert(varchar(20),convert(money,sum(s.programado_financiero)),1) as prog,
convert(varchar(20),convert(money,sum(s.ejecucion_financiero)),1) as ejec,
porc=
    Case sum(s.programado_financiero)
        When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_financiero)/sum(s.programado_financiero)*100)),1)+'%'
    end
into #RESULTADO3
from seguimiento s,proyectos p,obras_fisicas o
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=o.codigo_obra
and s.año=@anio
and o.codigo_canton=@canton
group by o.codigo_obra

```

```
update #RESULTADOS set prog=#RESULTADO3.prog,
ejec=#RESULTADO3.ejec,
porc=#RESULTADO3.porc
from #RESULTADOS,#RESULTADO3
where #RESULTADOS.codigo_obra=#RESULTADO3.codigo_obra
```

END

```
select nombre_obra as Obra,prog as [Monto Programado],ejec as [Monto Ejecutado],
porc as [Porcentaje de Avance]
from #RESULTADOS
where prog<>'-'
order by codigo_obra
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

/\*  
\*\*\*\*\*  
\*/

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure AvanceFisicoObras @institucion int ,@obra int ,@anio int
,@usuario varchar(10)
as
```

```
select codigo_proyecto,
nombre_proyecto,
convert(varchar(20),'-') as porc
into #RESULTADOS
from proyectos
where codigo_obra=@obra
```

```
if @institucion<>999
BEGIN
select p.codigo_proyecto,
porc=
```

```

        Case sum(s.programado_fisico)
            When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_fisico)/sum(s.programado_fisico)*
100)),1)+'%'
            end
into #RESULTADOS2
from seguimiento s,proyectos p,responsables r
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=@obra
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
and s.año=@año
group by p.codigo_proyecto

update #RESULTADOS set
porc=#RESULTADOS2.porc
from #RESULTADOS,#RESULTADOS2
where #RESULTADOS.codigo_proyecto=#RESULTADOS2.codigo_proyecto

END
else
BEGIN
select p.codigo_proyecto,
porc=
        Case sum(s.programado_fisico)
            When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_fisico)/sum(s.programado_fisico)*
100)),1)+'%'
            end
into #RESULTADO3
from seguimiento s,proyectos p
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=@obra
and s.año=@año
group by p.codigo_proyecto

update #RESULTADOS set
porc=#RESULTADO3.porc
from #RESULTADOS,#RESULTADO3
where #RESULTADOS.codigo_proyecto=#RESULTADO3.codigo_proyecto

END

```

```
select nombre_proyecto as Proyecto,porc as [Porcentaje de Avance]
from #RESULTADOS
order by codigo_proyecto
```

```
--exec AvanceFisicoObras 1,32,2003,'DATOSMINED'
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/*-*****-*/
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure AvanceFisicoObrasCanton @institucion int ,@canton int ,@anio
int ,@usuario varchar(10)
as
```

```
select codigo_obra,
'<a
href=/servlet/Reportes?Codigo=06&USUARIO='+@usuario+'&parametro='+convert(var
char(4),codigo_obra)+'&institucion='+convert(varchar(4),@institucion)+'&anio='+conve
rt(varchar(4),@anio)+'>'+nombre_obra+'</a>' as nombre_obra,
convert(varchar(20),'-') as porc
into #RESULTADOS
from obras_fisicas
where codigo_canton=@canton
```

```
if @institucion<>999
BEGIN
select o.codigo_obra,
porc=
    Case sum(s.programado_fisico)
```

```

                When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_fisico)/sum(s.programado_fisico)*
100)),1)+'%'
            end
into #RESULTADOS2
from seguimiento s,proyectos p,obras_fisicas o,responsables r
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=o.codigo_obra
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
and s.año=@anio
and o.codigo_canton=@canton
group by o.codigo_obra

update #RESULTADOS set porc=#RESULTADOS2.porc
from #RESULTADOS,#RESULTADOS2
where #RESULTADOS.codigo_obra=#RESULTADOS2.codigo_obra

END
else
BEGIN
select o.codigo_obra,
porc=
        Case sum(s.programado_fisico)
                When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_fisico)/sum(s.programado_fisico)*
100)),1)+'%'
            end
into #RESULTADO3
from seguimiento s,proyectos p,obras_fisicas o
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=o.codigo_obra
and s.año=@anio
and o.codigo_canton=@canton
group by o.codigo_obra

update #RESULTADOS set porc=#RESULTADO3.porc
from #RESULTADOS,#RESULTADO3
where #RESULTADOS.codigo_obra=#RESULTADO3.codigo_obra

END

```



```

                When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_financiero)/sum(s.programado_fin
anciero)*100)),1)+'%'
            end
into #RESULTADOS2
from cantones c,seguimiento s,proyectos p,obras_fisicas o,responsables r
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=o.codigo_obra
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
and s.año=@anio
and o.codigo_canton=c.codigo_canton
group by c.codigo_canton

update #RESULTADOS set prog=#RESULTADOS2.prog,
ejec=#RESULTADOS2.ejec,
porc=#RESULTADOS2.porc
from #RESULTADOS,#RESULTADOS2
where #RESULTADOS.codigo_canton=#RESULTADOS2.codigo_canton

END
else
BEGIN
select c.codigo_canton,
convert(varchar(20),convert(money,sum(s.programado_financiero)),1) as prog,
convert(varchar(20),convert(money,sum(s.ejecucion_financiero)),1) as ejec,
porc=
    Case sum(s.programado_financiero)
        When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_financiero)/sum(s.programado_fin
anciero)*100)),1)+'%'
    end
into #RESULTADO3
from cantones c,seguimiento s,proyectos p,obras_fisicas o
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=o.codigo_obra
and s.año=@anio
and o.codigo_canton=c.codigo_canton
group by c.codigo_canton

update #RESULTADOS set prog=#RESULTADO3.prog,
ejec=#RESULTADO3.ejec,
porc=#RESULTADO3.porc
from #RESULTADOS,#RESULTADO3

```

```
where #RESULTADOS.codigo_canton=#RESULTADO3.codigo_canton
```

```
END
```

```
select nombre_canton as Canton,prog as [Monto Programado],ejec as [Monto Ejecutado],  
porc as [Porcentaje de Avance]  
from #RESULTADOS  
order by codigo_canton
```

```
GO  
SET QUOTED_IDENTIFIER OFF  
GO  
SET ANSI_NULLS ON  
GO
```

```
/*  
*****  
*/
```

```
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_NULLS ON  
GO
```

```
ALTER procedure DetalleAvanceFisico @institucion int,@parametro char(1),@anio  
int,@usuario varchar(10)  
as
```

```
select codigo_canton,  
'<a  
href=/servlet/Reportes?Codigo=03&USUARIO='+@usuario+'&parametro='+convert(var  
char(4),codigo_canton)+'&institucion='+convert(varchar(4),@institucion)+'&anio='+con  
vert(varchar(4),@anio)+'>'+nombre_canton+'</a>' as nombre_canton,  
convert(varchar(20),'-') as porc  
into #RESULTADOS  
from cantones  
where codigo_municipio=601
```

```
if @institucion<>999  
BEGIN  
select c.codigo_canton,
```

```

porc=
    Case sum(s.programado_fisico)
        When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_fisico)/sum(s.programado_fisico)*
100)),1)+'%'
    end
into #RESULTADOS2
from cantones c,seguimiento s,proyectos p,obras_fisicas o,responsables r
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=o.codigo_obra
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
and s.año=@año
and o.codigo_canton=c.codigo_canton
group by c.codigo_canton

update #RESULTADOS set porc=#RESULTADOS2.porc
from #RESULTADOS,#RESULTADOS2
where #RESULTADOS.codigo_canton=#RESULTADOS2.codigo_canton

END
else
BEGIN
select c.codigo_canton,
porc=
    Case sum(s.programado_fisico)
        When 0 then convert(varchar(20),'0.00%') else
convert(varchar(20),convert(money,(sum(s.ejecucion_fisico)/sum(s.programado_fisico)*
100)),1)+'%'
    end
into #RESULTADO3
from cantones c,seguimiento s,proyectos p,obras_fisicas o
where s.codigo_proyecto=p.codigo_proyecto
and s.codigo_obra=p.codigo_obra
and p.codigo_obra=o.codigo_obra
and s.año=@año
and o.codigo_canton=c.codigo_canton
group by c.codigo_canton

update #RESULTADOS set porc=#RESULTADO3.porc
from #RESULTADOS,#RESULTADO3
where #RESULTADOS.codigo_canton=#RESULTADO3.codigo_canton

END

```

```
select nombre_canton as Canton,porc as [Porcentaje de Avance]
from #RESULTADOS
order by codigo_canton
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/*- - - - -*/
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure InversionTipoObra @parametro char(1) as
select distinct codigo_unidad,nombre_unidad, convert(numeric(20,2),0) as monto
into #RESULTADO
from unidades_medida
```

```
select a.codigo_unidad,sum(c.programado_financiero) as monto
into #RESULTADO2
from obras_fisicas a,proyectos p,seguimiento c,unidades_medida u
where a.codigo_obra=p.codigo_obra
and p.codigo_proyecto=c.codigo_proyecto
and u.codigo_unidad=a.codigo_unidad
group by a.codigo_unidad
```

```
update #RESULTADO set monto=#RESULTADO2.monto
from #RESULTADO,#RESULTADO2
where #RESULTADO.codigo_unidad=#RESULTADO2.codigo_unidad
```

```
insert into #RESULTADO
select 'TOTAL',sum(monto) from #RESULTADO
```

```
select nombre_unidad as [Tipo Obra],
monto as Inversión
```

```
from #RESULTADO
order by codigo_unidad
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/*- - - - -*/
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure InversionTipoObraMunicipioCanton @mun int
as
```

```
select c.codigo_canton, c.nombre_canton, count(p.codigo_proyecto) as cantidad,
t.codigo_tipo, t.nombre_tipo, convert(numeric(20,2),0) as monto
into #RESULTADO
from proyectos p, cantones c, obras_fisicas o, tipos_proyecto t
where o.codigo_obra=p.codigo_obra
and o.codigo_canton=c.codigo_canton
and p.codigo_tipo=t.codigo_tipo
group by c.codigo_canton, c.nombre_canton, t.codigo_tipo, t.nombre_tipo
```

```
insert into #RESULTADO
select 99, 'TOTAL', count(p.codigo_proyecto),
t.codigo_tipo, t.nombre_tipo, convert(numeric(20,2),0) as monto
from proyectos p, cantones c, obras_fisicas o, tipos_proyecto t
where o.codigo_obra=p.codigo_obra
and o.codigo_canton=c.codigo_canton
and p.codigo_tipo=t.codigo_tipo
group by t.codigo_tipo, t.nombre_tipo
```

```
select c.codigo_canton, t.codigo_tipo, sum(s.programado_financiero) as monto
into #RESULTADO2
from proyectos p, cantones c, obras_fisicas o, tipos_proyecto t, seguimiento s
```



```

ALTER procedure ListadoProyectos @obra int,@usuario varchar(10)
as

declare @perfil varchar(10)

select @perfil=cod_per from sisusuper where codigo_responsable=@usuario

if @perfil='CONSULSETE' or @perfil='ADMIN'
BEGIN
select p.nombre_proyecto as Proyecto,
p.empleos_proyecto as Empleos,
p.beneficiarios_proyecto as Beneficiarios,
e.nombre_etapa as Etapa,
t.nombre_tipo as [Tipo de Proyecto]
from proyectos p,etapas e,tipos_proyecto t
where e.codigo_etapa=p.codigo_etapa
and t.codigo_tipo=p.codigo_tipo
and p.codigo_obra= @obra
END
else
BEGIN
declare @institucion int
select @institucion=codigo_institucion from responsables where
codigo_responsable=@usuario

select p.nombre_proyecto as Proyecto,
p.empleos_proyecto as Empleos,
p.beneficiarios_proyecto as Beneficiarios,
e.nombre_etapa as Etapa,
t.nombre_tipo as [Tipo de Proyecto]
from proyectos p,etapas e,tipos_proyecto t,responsables r
where e.codigo_etapa=p.codigo_etapa
and t.codigo_tipo=p.codigo_tipo
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion = @institucion
and p.codigo_obra= @obra
END

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```



```

        Case sum(s.programado_financiero)
            When 0 then 0.00 else
            (sum(s.ejecucion_financiero)/sum(s.programado_financiero)*100)
        end
    into #RESULTADO2
    from seguimiento s,proyectos p,obras_fisicas o,responsables r
    where s.codigo_obra=p.codigo_obra
    and s.codigo_proyecto=p.codigo_proyecto
    and o.codigo_obra=p.codigo_obra
    and s.año=@codigo_anio
    and p.codigo_responsable=r.codigo_responsable
    and r.codigo_institucion=@institucion
    group by o.codigo_canton

    update #RESULTADO set valor=#RESULTADO2.porc
    from #RESULTADO,#RESULTADO2
    where #RESULTADO.codigo_canton=#RESULTADO2.codigo_canton

    select codigo_canton,valor from #RESULTADO

```

END

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

/\*\*\*\*\*/

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

ALTER    procedure PorcentajeAvanceFisicoCanton @institucion int,@codigo_anio int
as

```

```

select codigo_canton,convert(numeric(8,2),-1) as valor

```

```

into #RESULTADO
from cantones where codigo_municipio=601

if @institucion=999
BEGIN
select o.codigo_canton,
porc=
    Case sum(s.programado_fisico)
        When 0 then 0.00 else
(sum(s.ejecucion_fisico)/sum(s.programado_fisico)*100)
    end
into #RESULTADO1
from seguimiento s,proyectos p,obras_fisicas o
where s.codigo_obra=p.codigo_obra
and s.codigo_proyecto=p.codigo_proyecto
and o.codigo_obra=p.codigo_obra
and s.año=@codigo_anio
group by o.codigo_canton

update #RESULTADO set valor=#RESULTADO1.porc
from #RESULTADO,#RESULTADO1
where #RESULTADO.codigo_canton=#RESULTADO1.codigo_canton

select codigo_canton,valor from #RESULTADO

END
else
BEGIN
select o.codigo_canton,
porc=
    Case sum(s.programado_fisico)
        When 0 then 0.00 else
(sum(s.ejecucion_fisico)/sum(s.programado_fisico)*100)
    end
into #RESULTADO2
from seguimiento s,proyectos p,obras_fisicas o,responsables r
where s.codigo_obra=p.codigo_obra
and s.codigo_proyecto=p.codigo_proyecto
and o.codigo_obra=p.codigo_obra
and s.año=@codigo_anio
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
group by o.codigo_canton

```



```
update #RESULTADO set mes ='Julio' where cod_mes=7
update #RESULTADO set mes ='Agosto' where cod_mes=8
update #RESULTADO set mes ='Septiembre' where cod_mes=9
update #RESULTADO set mes ='Octubre' where cod_mes=10
update #RESULTADO set mes ='Noviembre' where cod_mes=11
update #RESULTADO set mes ='Diciembre' where cod_mes=12
```

```
select mes as Mes,Programado,Ejecución from #RESULTADO order by cod_mes
```

```
END
ELSE
BEGIN
select s.mes as cod_mes,convert(varchar(15),'?') as mes,sum(s.programado_financiero) as
Programado,
sum(s.ejecucion_financiero) as Ejecución
into #RESULTADO2
from seguimiento s,proyectos p,responsables r
where s.año=@anio
and s.codigo_proyecto=p.codigo_proyecto
and r.codigo_responsable=p.codigo_responsable
and r.codigo_institucion=@institucion
group by mes
```

```
update #RESULTADO2 set mes ='Enero' where cod_mes=1
update #RESULTADO2 set mes ='Febrero' where cod_mes=2
update #RESULTADO2 set mes ='Marzo' where cod_mes=3
update #RESULTADO2 set mes ='Abril' where cod_mes=4
update #RESULTADO2 set mes ='Mayo' where cod_mes=5
update #RESULTADO2 set mes ='Junio' where cod_mes=6
update #RESULTADO2 set mes ='Julio' where cod_mes=7
update #RESULTADO2 set mes ='Agosto' where cod_mes=8
update #RESULTADO2 set mes ='Septiembre' where cod_mes=9
update #RESULTADO2 set mes ='Octubre' where cod_mes=10
update #RESULTADO2 set mes ='Noviembre' where cod_mes=11
update #RESULTADO2 set mes ='Diciembre' where cod_mes=12
```

```
select mes as Mes,Programado,Ejecución from #RESULTADO2 order by cod_mes
END
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

/\*  
\*\*\*\*\*  
\*/

```
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_NULLS ON  
GO
```

```
ALTER procedure ProgramacionEjecucionEtapa @usuario varchar(10),@institucion  
int,@anio int  
as
```

```
if(@institucion=999)  
BEGIN  
select '<a  
href=/servlet/ReportesListado?CODIGO=01&USUARIO='+@usuario+'&PARAMETRO  
='+convert(varchar(5),e.codigo_etapa)+'&INSTITUCION='+convert(varchar(3),@institu  
cion)+'&ANIO='+convert(varchar(4),@anio)+'>'+e.nombre_etapa+'</a>' as [Etapa],  
sum(s.programado_financiero) as Programado,  
sum(s.ejecucion_financiero) as Ejecutado  
from proyectos p,seguimiento s,etapas e  
where p.codigo_proyecto=s.codigo_proyecto  
and p.codigo_etapa=e.codigo_etapa  
and s.año=@anio  
group by e.codigo_etapa,e.nombre_etapa
```

```
END  
ELSE  
BEGIN
```

```
select '<a  
href=/servlet/ReportesListado?CODIGO=01&USUARIO='+@usuario+'&PARAMETRO  
='+convert(varchar(5),e.codigo_etapa)+'&INSTITUCION='+convert(varchar(3),@institu  
cion)+'&ANIO='+convert(varchar(4),@anio)+'>'+e.nombre_etapa+'</a>' as [Etapa],  
sum(s.programado_financiero) as Programado,  
sum(s.ejecucion_financiero) as Ejecutado  
from proyectos p,seguimiento s,etapas e,responsables r  
where p.codigo_proyecto=s.codigo_proyecto  
and p.codigo_etapa=e.codigo_etapa  
and p.codigo_responsable=r.codigo_responsable
```

```
and r.codigo_institucion=@institucion
and s.año=@anio
group by e.codigo_etapa,e.nombre_etapa
```

```
END
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/*  
*****  
*/
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure ProgramacionEjecucionFuente @usuario varchar(10),@institucion
int,@anio int
as
```

```
IF (@institucion=999)
BEGIN
select '<a
href=/servlet/ReportesListado?CODIGO=02&USUARIO='+@usuario+'&INSTITUCIO
N='+convert(varchar(3),@institucion)+'&ANIO='+convert(varchar(4),@anio)+'&PARA
METRO='+f.codigo_fuente+'>'+fp.nombre_fuente+'</a>' as [Fuente de Financiamiento],
sum(s.programado_financiero) as Programación,
sum(s.ejecucion_financiero) as Ejecución
from proyectos_fuentes f,proyectos p,seguimiento s,fuentes_financiamiento fp
where p.codigo_proyecto=s.codigo_proyecto
and p.codigo_proyecto=f.codigo_proyecto
and s.año=@anio
and f.codigo_fuente=fp.codigo_fuente
group by f.codigo_fuente,fp.nombre_fuente
END
ELSE
```

```

BEGIN
select '<a
href=/servlet/ReportesListado?CODIGO=02&USUARIO='+@usuario+'&INSTITUCIO
N='+convert(varchar(3),@institucion)+'&ANIO='+convert(varchar(4),@anio)+'&PARA
METRO='+f.codigo_fuente+'>'+fp.nombre_fuente+'</a>' as [Fuente de Financiamiento],
sum(s.programado_financiero) as Programación,
sum(s.ejecucion_financiero) as Ejecución
from proyectos_fuentes f,proyectos p,seguimiento s,responsables
r,fuentes_financiamiento fp
where p.codigo_proyecto=s.codigo_proyecto
and p.codigo_proyecto=f.codigo_proyecto
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
and s.año=@anio
and f.codigo_fuente=fp.codigo_fuente
group by f.codigo_fuente,fp.nombre_fuente

```

```
END
```

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

```

/*****

```

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

ALTER procedure ProgramacionEjecucionInstitucion @usuario
varchar(10),@institucion int,@anio int
as

```

```

if (@institucion=999)
BEGIN
select '<a
href=/servlet/ReportesConsolidados?ANIO='+convert(varchar(4),@anio)+'&USUARIO=

```

```

'+@usuario+'&INSTITUCION='+convert(varchar(3),i.codigo_institucion)+'>'+i.nombre_
institucion+'</a>' as Institución,
sum(s.programado_financiero) as Programado,
sum(s.ejecucion_financiero) as Ejecutado
from seguimiento s,proyectos p,instituciones i,responsables r
where s.codigo_proyecto=p.codigo_proyecto
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=i.codigo_institucion
and s.año=@anio
group by i.codigo_institucion,i.nombre_institucion
END
ELSE
BEGIN
select i.nombre_institucion as Institución,
sum(s.programado_financiero) as Programado,
sum(s.ejecucion_financiero) as Ejecutado
from seguimiento s,proyectos p,instituciones i,responsables r
where s.codigo_proyecto=p.codigo_proyecto
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=i.codigo_institucion
and i.codigo_institucion=@institucion
and s.año=@anio
group by i.nombre_institucion

END

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

/*****

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

ALTER    procedure ProgramacionEjecucionUnidadesMedida @usuario
varchar(10),@institucion int,@anio int

```

as

```
if(@institucion=999)
BEGIN
select '<a
href=/servlet/ReportesListado?CODIGO=03&USUARIO='+@usuario+'&PARAMETRO
='+convert(varchar(5),u.codigo_unidad)+'&INSTITUCION='+convert(varchar(3),@instit
ucion)+'&ANIO='+convert(varchar(4),@anio)+'>'+u.nombre_unidad+'</a>' as [Tipos de
Obra],
sum (s.programado_financiero) as Programado,
sum(s.ejecucion_financiero) as Ejecutado
from unidades_medida u,proyectos p,obras_fisicas o,seguimiento s
where s.codigo_proyecto=p.codigo_proyecto
and p.codigo_obra=o.codigo_obra
and o.codigo_unidad=u.codigo_unidad
and s.año=@anio
group by u.codigo_unidad,u.nombre_unidad
```

END

ELSE

BEGIN

```
select '<a
href=/servlet/ReportesListado?CODIGO=03&USUARIO='+@usuario+'&PARAMETRO
='+convert(varchar(5),u.codigo_unidad)+'&INSTITUCION='+convert(varchar(3),@instit
ucion)+'&ANIO='+convert(varchar(4),@anio)+'>'+u.nombre_unidad+'</a>' as [Tipos de
Obra],
sum (s.programado_financiero) as Programado,
sum(s.ejecucion_financiero) as Ejecutado
from unidades_medida u,proyectos p,obras_fisicas o,seguimiento s,responsables r
where s.codigo_proyecto=p.codigo_proyecto
and p.codigo_obra=o.codigo_obra
and o.codigo_unidad=u.codigo_unidad
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
and s.año=@anio
group by u.codigo_unidad,u.nombre_unidad
END
```

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

/\*\*\*\*\*\*

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure ProyectosEtapa @etapa int, @institucion int, @anio int
as
```

```
if(@institucion=999)
BEGIN
select o.nombre_obra as Obra,p.nombre_proyecto as Proyecto,t.nombre_tipo as [Tipo de
Proyecto],
p.empleos_proyecto as [Empleos Generados],
p.beneficiarios_proyecto as [Beneficiarios],
convert(varchar(20),convert(money,sum(s.programado_financiero))) as [Monto
Programado],
convert(money,convert(varchar(20),sum(s.ejecucion_financiero))) as [Monto Ejecutado]
from proyectos p,tipos_proyecto t,seguimiento s,obras_fisicas o
where p.codigo_tipo=t.codigo_tipo
and p.codigo_etapa= @etapa
and p.codigo_proyecto=s.codigo_proyecto
and s.año=@anio
and o.codigo_obra=p.codigo_obra
group by
o.nombre_obra,p.nombre_proyecto,t.nombre_tipo,p.empleos_proyecto,p.beneficiarios_pr
oyecto
END
else
BEGIN
select o.nombre_obra as Obra,p.nombre_proyecto as Proyecto,t.nombre_tipo as [Tipo de
Proyecto],
p.empleos_proyecto as [Empleos Generados],
p.beneficiarios_proyecto as [Beneficiarios],
convert(money,convert(varchar(20),sum(s.programado_financiero))) as [Monto
Programado],
convert(money,convert(varchar(20),sum(s.ejecucion_financiero))) as [Monto Ejecutado]
from proyectos p,tipos_proyecto t,seguimiento s,responsables r,obras_fisicas o
where p.codigo_tipo=t.codigo_tipo
and p.codigo_etapa= @etapa
```

```
and p.codigo_proyecto=s.codigo_proyecto
and s.año=@anio
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
and o.codigo_obra=p.codigo_obra
group by
o.nombre_obra,p.nombre_proyecto,t.nombre_tipo,p.empleos_proyecto,p.beneficiarios_pr
oyecto
```

END

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

/\*\*\*\*\*/

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure ProyectosFuente @fuente char(10),@institucion int,@anio int
as
if (@institucion=999)
BEGIN
```

```
select o.codigo_obra,o.nombre_obra,p.codigo_proyecto,p.nombre_proyecto,
t.nombre_tipo, e.nombre_etapa,
sum(s.programado_financiero ) as monto_total,
convert(numeric(20,2),0) as monto_fuente
into #RESULTADO1
from obras_fisicas o,proyectos p,seguimiento s,etapas e,tipos_proyecto t
where o.codigo_obra=p.codigo_obra
and p.codigo_obra=s.codigo_obra
and p.codigo_proyecto=s.codigo_proyecto
and p.codigo_tipo=t.codigo_tipo
and p.codigo_etapa=e.codigo_etapa
and s.año=@anio
```

```
group by
o.codigo_obra,o.nombre_obra,p.codigo_proyecto,p.nombre_proyecto,t.nombre_tipo,e.no
mbre_etapa
```

```
select f.codigo_obra,f.codigo_proyecto,f.monto
into #RESULTADO2
from proyectos_fuentes f
where f.codigo_fuente=@fuente
```

```
update #RESULTADO1 set monto_fuente=#RESULTADO2.monto
from #RESULTADO1,#RESULTADO2
where #RESULTADO1.codigo_obra=#RESULTADO2.codigo_obra
and #RESULTADO1.codigo_proyecto=#RESULTADO2.codigo_proyecto
```

```
delete #RESULTADO1 where monto_fuente=0
```

```
select nombre_obra as Obra, nombre_proyecto as Proyecto,
nombre_tipo as [Tipo de Proyecto],nombre_etapa as [Etapa],
monto_Total as [Monto Total de Inversión],
monto_fuente as [Monto Financiado]
from #RESULTADO1
```

```
END
else
BEGIN
```

```
select o.codigo_obra,o.nombre_obra,p.codigo_proyecto,p.nombre_proyecto,
t.nombre_tipo, e.nombre_etapa,
sum(s.programado_financiero ) as monto_total,
convert(numeric(20,2),0) as monto_fuente
into #RESULTADO3
from obras_fisicas o,proyectos p,seguimiento s,etapas e,tipos_proyecto t,responsables r
where o.codigo_obra=p.codigo_obra
and p.codigo_obra=s.codigo_obra
and p.codigo_proyecto=s.codigo_proyecto
and p.codigo_tipo=t.codigo_tipo
and p.codigo_etapa=e.codigo_etapa
and s.año=@anio
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
group by
o.codigo_obra,o.nombre_obra,p.codigo_proyecto,p.nombre_proyecto,t.nombre_tipo,e.no
mbre_etapa
```

```
select f.codigo_obra,f.codigo_proyecto,f.monto
into #RESULTADO4
from proyectos_fuentes f
where f.codigo_fuente=@fuente
```

```
update #RESULTADO3 set monto_fuente=#RESULTADO4.monto
from #RESULTADO3,#RESULTADO4
where #RESULTADO3.codigo_obra=#RESULTADO4.codigo_obra
and #RESULTADO3.codigo_proyecto=#RESULTADO4.codigo_proyecto
```

```
delete #RESULTADO3 where monto_fuente=0
```

```
select nombre_obra as Obra, nombre_proyecto as Proyecto,
nombre_tipo as [Tipo de Proyecto],nombre_etapa as [Etapa],
monto_Total as [Monto Total de Inversión],
monto_fuente as [Monto Financiado]
from #RESULTADO3
```

```
END
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/*****/
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure ProyectosObra @obra int,@institucion int,@anio int
as
```

```
if @institucion=999
BEGIN
select p.nombre_proyecto as Proyecto,
t.nombre_tipo as [Tipo de Proyecto],
```

```

e.nombre_etapa as Etapa,
p.empleos_proyecto as [Empleos Generados],
p.beneficiarios_proyecto as [Beneficiarios],
sum(s.programado_financiero) as [Monto Programado],
sum(s.ejecucion_financiero) as [Monto Ejecutado]
from proyectos p,obras_fisicas o,tipos_proyecto t,etapas e,seguimiento s
where o.codigo_obra=p.codigo_obra
and p.codigo_proyecto=s.codigo_proyecto
and p.codigo_obra=s.codigo_obra
and p.codigo_tipo=t.codigo_Tipo
and p.codigo_etapa=e.codigo_etapa
and s.año=@anio
and o.codigo_obra=@obra
group by p.nombre_proyecto,t.nombre_tipo,e.nombre_etapa,
p.empleos_proyecto,p.beneficiarios_proyecto
END
ELSE
BEGIN
select p.nombre_proyecto as Proyecto,
t.nombre_tipo as [Tipo de Proyecto],
e.nombre_etapa as Etapa,
p.empleos_proyecto as [Empleos Generados],
p.beneficiarios_proyecto as [Beneficiarios],
sum(s.programado_financiero) as [Monto Programado],
sum(s.ejecucion_financiero) as [Monto Ejecutado]
from proyectos p,obras_fisicas o,tipos_proyecto t,etapas e,seguimiento s,responsables r
where o.codigo_obra=p.codigo_obra
and p.codigo_proyecto=s.codigo_proyecto
and p.codigo_obra=s.codigo_obra
and p.codigo_tipo=t.codigo_Tipo
and p.codigo_etapa=e.codigo_etapa
and s.año=@anio
and o.codigo_obra=@obra
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
group by p.nombre_proyecto,t.nombre_tipo,e.nombre_etapa,
p.empleos_proyecto,p.beneficiarios_proyecto
END

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

/\*\*\*\*\*/

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure RetornarAnios
@buscarpor char(1),
@codigo varchar(3),
@consulta char(1)
as
```

```
if @buscarpor='N'
begin
    if @consulta='0'
        Select Count(*) from Seguimiento
    else
        Select distinct (año) From Seguimiento
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

/\*\*\*\*\*/

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure RetornarDepartamentos
@buscarpor char(1),
@codigo varchar(3),
@consulta char(1)
```

```

as

if @buscarpor='N'
begin
    if @consulta='0'
        Select Count(*) from Departamentos
    else
        Select * From Departamentos
end
else
begin
    if @buscarpor='i'
    begin
        if @consulta='0'
        begin
            Select Count(distinct d.codigo_departamento)
            From Proyectos p, Obras_Fisicas o, cantones c, municipios m,
departamentos d,
            Responsables r, Instituciones i
            Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
            and c.codigo_municipio=m.codigo_municipio
            and m.codigo_departamento=d.codigo_departamento
            and p.codigo_responsable=r.codigo_responsable
            and r.codigo_institucion=i.codigo_institucion
            and i.codigo_institucion=@codigo
        end
        else
        begin
            Select distinct d.*
            From Proyectos p, Obras_Fisicas o, cantones c, municipios m,
departamentos d,
            Responsables r, Instituciones i
            Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
            and c.codigo_municipio=m.codigo_municipio
            and m.codigo_departamento=d.codigo_departamento
            and p.codigo_responsable=r.codigo_responsable
            and r.codigo_institucion=i.codigo_institucion
            and i.codigo_institucion=@codigo
        end
    end
    else
    begin
        if @buscarpor='t'
        begin

```

```

        if @consulta='0'
        begin
            Select Count(*)
            From Proyectos p, Obras_Fisicas o, cantones c, municipios
m, departamentos d,
            Unidades_Medida u
            Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
            and c.codigo_municipio=m.codigo_municipio
            and m.codigo_departamento=d.codigo_departamento
            and o.codigo_unidad=u.codigo_unidad
            and u.codigo_unidad=@codigo
        end
        else
        begin
            Select d.*
            From Proyectos p, Obras_Fisicas o, cantones c, municipios
m, departamentos d,
            Unidades_Medida u
            Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
            and c.codigo_municipio=m.codigo_municipio
            and m.codigo_departamento=d.codigo_departamento
            and o.codigo_unidad=u.codigo_unidad
            and u.codigo_unidad=@codigo
        end
    end
end
end

```

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

/\*\*\*/

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON

```

GO

```
ALTER procedure RetornarInstituciones
```

```
@buscarpor char(1),
```

```
@codigo varchar(3),
```

```
@consulta char(1)
```

```
as
```

```
if @buscarpor='N'
```

```
begin
```

```
    if @consulta='0'
```

```
        Select Count(*) from Instituciones
```

```
    else
```

```
    begin
```

```
        if @consulta='1'
```

```
            Select * From Instituciones
```

```
        else
```

```
            Select * From Instituciones where Codigo_Institucion=@codigo
```

```
    end
```

```
end
```

```
else
```

```
begin
```

```
    if @buscarpor='d'
```

```
    begin
```

```
        if @consulta='0'
```

```
        begin
```

```
            Select Count(*) From Instituciones
```

```
            Where codigo_institucion in
```

```
            (
```

```
                Select codigo_institucion from responsables
```

```
                Where codigo_responsable in
```

```
                (
```

```
                    Select p.codigo_responsable
```

```
                    From Proyectos p, Obras_Fisicas o, cantones c,
```

```
municipios m, departamentos d
```

```
                    Where p.codigo_obra=o.codigo_obra and
```

```
o.codigo_canton=c.codigo_canton
```

```
                    and c.codigo_municipio=m.codigo_municipio
```

```
                    and
```

```
m.codigo_departamento=d.codigo_departamento
```

```
                    and d.codigo_departamento=convert(int,@codigo)
```

```
                )
```

```
            )
```

```
        end
```

```
    else
```

```

begin
  Select * From Instituciones
  Where codigo_institucion in
  (
    Select codigo_institucion from responsables
    Where codigo_responsable in
    (
      Select p.codigo_responsable
      From Proyectos p, Obras_Fisicas o, cantones c,
municipios m, departamentos d
      Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
      and c.codigo_municipio=m.codigo_municipio
      and
m.codigo_departamento=d.codigo_departamento
      and d.codigo_departamento=convert(int,@codigo)
    )
  )
end
else
begin
  if @buscarpor='t'
  begin
    if @consulta='0'
    begin
      Select Count(*) From Instituciones
      Where codigo_institucion in
      (
        Select codigo_institucion from responsables
        Where codigo_responsable in
        (
          Select p.codigo_responsable
          From Proyectos p, Obras_Fisicas o,
Unidades_Medida as u
          Where p.codigo_obra=o.codigo_obra
          and o.codigo_unidad=u.codigo_unidad
          and u.codigo_unidad=@codigo
        )
      )
    end
  else
  begin
    Select * From Instituciones
    Where codigo_institucion in
    (

```

```

Unidades_Medida as u
        Select codigo_institucion from responsables
        Where codigo_responsable in
        (
            Select p.codigo_responsable
            From Proyectos p, Obras_Fisicas o,

            Where p.codigo_obra=o.codigo_obra
            and o.codigo_unidad=u.codigo_unidad
            and u.codigo_unidad=@codigo
        )
    end
end
end
end

```

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

/\*\*\*\*\*\*

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

ALTER procedure RetornarMunicipios
@buscarpor char(1),
@codigo varchar(3),
@consulta char(1)
as

if @buscarpor='N'
begin
    if @consulta='0'
        Select Count(*) from Municipios
    else

```

```

                Select * From Municipios
end
else
begin
    if @buscarpor='i'
    begin
        if @consulta='0'
        begin
            Select Count(distinct d.codigo_departamento)
            From Proyectos p, Obras_Fisicas o, cantones c, municipios m,
departamentos d,
            Responsables r, Instituciones i
            Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
            and c.codigo_municipio=m.codigo_municipio
            and m.codigo_departamento=d.codigo_departamento
            and p.codigo_responsable=r.codigo_responsable
            and r.codigo_institucion=i.codigo_institucion
            and i.codigo_institucion=@codigo
        end
        else
        begin
            Select distinct d.*
            From Proyectos p, Obras_Fisicas o, cantones c, municipios m,
departamentos d,
            Responsables r, Instituciones i
            Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
            and c.codigo_municipio=m.codigo_municipio
            and m.codigo_departamento=d.codigo_departamento
            and p.codigo_responsable=r.codigo_responsable
            and r.codigo_institucion=i.codigo_institucion
            and i.codigo_institucion=@codigo
        end
    end
    else
    begin
        if @buscarpor='t'
        begin
            if @consulta='0'
            begin
                Select Count(*)
                From Proyectos p, Obras_Fisicas o, cantones c, municipios
m, departamentos d,
                Unidades_Medida u

```

```

Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
and c.codigo_municipio=m.codigo_municipio
and m.codigo_departamento=d.codigo_departamento
and o.codigo_unidad=u.codigo_unidad
and u.codigo_unidad=@codigo
end
else
begin
Select distinct m.*
From Proyectos p, Obras_Fisicas o, cantones c, municipios
m, departamentos d,
Unidades_Medida u
Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
and c.codigo_municipio=m.codigo_municipio
and m.codigo_departamento=d.codigo_departamento
and o.codigo_unidad=u.codigo_unidad
and u.codigo_unidad=@codigo
end
end
else
if @buscarpor='d'
begin
if @consulta='0'
begin
Select Count(*)
From Proyectos p, Obras_Fisicas o, cantones c,
municipios m, departamentos d,
Unidades_Medida u
Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
and c.codigo_municipio=m.codigo_municipio
and
m.codigo_departamento=d.codigo_departamento
and o.codigo_unidad=u.codigo_unidad
and d.codigo_departamento=@codigo
end
end
else
begin
Select distinct m.*
From Proyectos p, Obras_Fisicas o, cantones c,
municipios m, departamentos d,
Unidades_Medida u
Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton

```

```

                                and c.codigo_municipio=m.codigo_municipio
                                and
m.codigo_departamento=d.codigo_departamento
                                and o.codigo_unidad=u.codigo_unidad
                                and d.codigo_departamento=@codigo
                                end
                                end
                                end
end

```

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

/\*\*\*\*\*\*

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

ALTER procedure RetornarObras
@institucion int,
@tipoObra varchar(50),
@departamento int,
@etapa int,
@municipio int,
@anio int,
@todos int =0

```

as

```

if @institucion<>999
begin
    if @tipoObra<>'-'
    begin
        if @departamento<>-1

```

```

begin
  if @etapa<>-1
  begin
    if @municipio<>-1
    begin
      SELECT distinct u.* FROM obras_fisicas o,
      proyectos p, responsables r, instituciones i,
      departamentos d, municipios m, cantones c, etapas e,
      unidades_medida u
      WHERE o.codigo_obra=p.codigo_obra and
      p.codigo_responsable=r.codigo_responsable and
      r.codigo_institucion=i.codigo_institucion
      and o.codigo_canton=c.codigo_canton and
      c.codigo_municipio=m.codigo_municipio
      and
      m.codigo_departamento=d.codigo_departamento
      and p.codigo_etapa=e.codigo_etapa
      and i.codigo_institucion=@institucion and
      e.codigo_etapa=@etapa
      and d.codigo_departamento=@departamento
      and o.codigo_unidad=u.codigo_unidad and
      u.codigo_unidad=@tipoObra
      and m.codigo_municipio=@municipio
    end
    begin
      SELECT distinct u.* FROM obras_fisicas o,
      proyectos p, responsables r, instituciones i,
      departamentos d, municipios m, cantones c, etapas e,
      unidades_medida u
      WHERE o.codigo_obra=p.codigo_obra and
      p.codigo_responsable=r.codigo_responsable and
      r.codigo_institucion=i.codigo_institucion
      and o.codigo_canton=c.codigo_canton and
      c.codigo_municipio=m.codigo_municipio
      and
      m.codigo_departamento=d.codigo_departamento
      and p.codigo_etapa=e.codigo_etapa
      and i.codigo_institucion=@institucion and
      e.codigo_etapa=@etapa
      and d.codigo_departamento=@departamento
      and o.codigo_unidad=u.codigo_unidad and
      u.codigo_unidad=@tipoObra
    end
  end
  else
  begin

```

```

        if @municipio<>-1
        begin
            SELECT distinct u.* FROM obras_fisicas o,
            proyectos p, responsables r, instituciones i,
            departamentos d, municipios m, cantones c,
            unidades_medida u
            WHERE o.codigo_obra=p.codigo_obra and
            p.codigo_responsable=r.codigo_responsable and
            r.codigo_institucion=i.codigo_institucion
            and o.codigo_canton=c.codigo_canton and
            c.codigo_municipio=m.codigo_municipio
            and
            m.codigo_departamento=d.codigo_departamento
            and i.codigo_institucion=@institucion
            and d.codigo_departamento=@departamento
            and o.codigo_unidad=u.codigo_unidad and
            u.codigo_unidad=@tipoObra
            and m.codigo_municipio=@municipio
        end
        else
        begin
            SELECT distinct u.* FROM obras_fisicas o,
            proyectos p, responsables r, instituciones i,
            departamentos d, municipios m, cantones c,
            unidades_medida u
            WHERE o.codigo_obra=p.codigo_obra and
            p.codigo_responsable=r.codigo_responsable and
            r.codigo_institucion=i.codigo_institucion
            and o.codigo_canton=c.codigo_canton and
            c.codigo_municipio=m.codigo_municipio
            and
            m.codigo_departamento=d.codigo_departamento
            and i.codigo_institucion=@institucion
            and d.codigo_departamento=@departamento
            and o.codigo_unidad=u.codigo_unidad and
            u.codigo_unidad=@tipoObra
        end
    end
end
else
begin
    if @etapa<>-1
    begin
        if @municipio<>-1
        begin
            SELECT distinct u.* FROM obras_fisicas o,

```

```

municipios m, cantones c,
proyectos p, responsables r, instituciones i,
etapas e, unidades_medida u
WHERE o.codigo_obra=p.codigo_obra
and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
and p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
and p.codigo_etapa=e.codigo_etapa
and i.codigo_institucion=@institucion and
e.codigo_etapa=@etapa
and o.codigo_unidad=u.codigo_unidad and
u.codigo_unidad=@tipoObra
and m.codigo_municipio=@municipio
end
else
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p, responsables r, instituciones i,
etapas e, unidades_medida u
WHERE o.codigo_obra=p.codigo_obra and
p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
and p.codigo_etapa=e.codigo_etapa
and i.codigo_institucion=@institucion and
e.codigo_etapa=@etapa
and o.codigo_unidad=u.codigo_unidad and
u.codigo_unidad=@tipoObra
end
end
else
begin
if @municipio<>-1
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p, responsables r, instituciones i,
municipios m, cantones c,
unidades_medida u
WHERE o.codigo_obra=p.codigo_obra and
p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
and i.codigo_institucion=@institucion
and o.codigo_unidad=u.codigo_unidad and
u.codigo_unidad=@tipoObra

```

```

        and m.codigo_municipio=@municipio
    end
    else
        begin
        SELECT distinct u.* FROM obras_fisicas o,
        proyectos p, responsables r, instituciones i,
        unidades_medida u
        WHERE o.codigo_obra=p.codigo_obra and
        p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
        and i.codigo_institucion=@institucion
        and o.codigo_unidad=u.codigo_unidad and
u.codigo_unidad=@tipoObra
        end
    end
end
else
begin
    if @departamento<>-1
    begin
        if @etapa<>-1
        begin
            if @municipio<>-1
            begin
                SELECT distinct u.* FROM obras_fisicas o,
                proyectos p, responsables r, instituciones i,
                departamentos d, municipios m, cantones c, etapas e,
unidades_medida u
                WHERE o.codigo_obra=p.codigo_obra and
                p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
                and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
                and
m.codigo_departamento=d.codigo_departamento
                and p.codigo_etapa=e.codigo_etapa
                and i.codigo_institucion=@institucion and
e.codigo_etapa=@etapa
                and d.codigo_departamento=@departamento
                and o.codigo_unidad=u.codigo_unidad
                and m.codigo_municipio=@municipio
            end
            else
            begin

```

```

SELECT distinct u.* FROM obras_fisicas o,
proyectos p, responsables r, instituciones i,
departamentos d, municipios m, cantones c, etapas e,
unidades_medida u
WHERE o.codigo_obra=p.codigo_obra and
p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
and
m.codigo_departamento=d.codigo_departamento
and p.codigo_etapa=e.codigo_etapa
and i.codigo_institucion=@institucion and
e.codigo_etapa=@etapa
and d.codigo_departamento=@departamento
and o.codigo_unidad=u.codigo_unidad
end
end
else
begin
if @municipio<>-1
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p, responsables r, instituciones i,
departamentos d, municipios m, cantones c,
unidades_medida u
WHERE o.codigo_obra=p.codigo_obra and
p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
and
m.codigo_departamento=d.codigo_departamento
and i.codigo_institucion=@institucion
and d.codigo_departamento=@departamento
and o.codigo_unidad=u.codigo_unidad
and m.codigo_municipio=@municipio
end
end
else
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p, responsables r, instituciones i,
departamentos d, municipios m, cantones c,
unidades_medida u
WHERE o.codigo_obra=p.codigo_obra and

```

```

                                p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
                                and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
                                and
m.codigo_departamento=d.codigo_departamento
                                and i.codigo_institucion=@institucion
                                and d.codigo_departamento=@departamento
                                and o.codigo_unidad=u.codigo_unidad
                                end
                                end
                                end
                                else
                                begin
                                if @etapa<>-1
                                begin
                                if @municipio<>-1
                                begin
                                SELECT distinct u.* FROM obras_fisicas o,
                                proyectos p, responsables r, instituciones i,
municipios m, cantones c,
                                etapas e, unidades_medida u
                                WHERE o.codigo_obra=p.codigo_obra and

                                p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
                                and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
                                and p.codigo_etapa=e.codigo_etapa
                                and i.codigo_institucion=@institucion and
e.codigo_etapa=@etapa
                                and o.codigo_unidad=u.codigo_unidad
                                and m.codigo_municipio=@municipio
                                end
                                else
                                begin
                                SELECT distinct u.* FROM obras_fisicas o,
                                proyectos p, responsables r, instituciones i,
                                etapas e, unidades_medida u
                                WHERE o.codigo_obra=p.codigo_obra and
                                p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
                                and p.codigo_etapa=e.codigo_etapa
                                and i.codigo_institucion=@institucion and
e.codigo_etapa=@etapa
                                and o.codigo_unidad=u.codigo_unidad

```

```

end
end
else
begin
    if @municipio<>-1
    begin
        SELECT distinct u.* FROM obras_fisicas o,
        proyectos p, responsables r, instituciones i,
municipios m, cantones c,
        unidades_medida u
        WHERE o.codigo_obra=p.codigo_obra and
        p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
        and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
        and i.codigo_institucion=@institucion
        and o.codigo_unidad=u.codigo_unidad
        and m.codigo_municipio=@municipio
    end
    else
    begin
        SELECT distinct u.* FROM obras_fisicas o,
        proyectos p, responsables r, instituciones i,
        unidades_medida u
        WHERE o.codigo_obra=p.codigo_obra and
        p.codigo_responsable=r.codigo_responsable and
r.codigo_institucion=i.codigo_institucion
        and i.codigo_institucion=@institucion
        and o.codigo_unidad=u.codigo_unidad
    end
    end
end
end
end
else
begin
    if @tipoObra<>'-1'
    begin
        if @departamento<>-1
        begin
            if @etapa<>-1
            begin
                if @municipio<>-1
                begin
                    SELECT distinct u.* FROM obras_fisicas o,

```

```

                                proyectos p,
                                departamentos d, municipios m, cantones c, etapas e,
unidades_medida u
                                WHERE o.codigo_obra=p.codigo_obra
                                and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
                                and
m.codigo_departamento=d.codigo_departamento
                                and p.codigo_etapa=e.codigo_etapa
                                and e.codigo_etapa=@etapa
                                and d.codigo_departamento=@departamento
                                and o.codigo_unidad=u.codigo_unidad and
u.codigo_unidad=@tipoObra
                                and m.codigo_municipio=@municipio
                                end
                                else
                                begin
                                SELECT distinct u.* FROM obras_fisicas o,
                                proyectos p,
                                departamentos d, municipios m, cantones c, etapas e,
unidades_medida u
                                WHERE o.codigo_obra=p.codigo_obra
                                and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
                                and
m.codigo_departamento=d.codigo_departamento
                                and p.codigo_etapa=e.codigo_etapa
                                and e.codigo_etapa=@etapa
                                and d.codigo_departamento=@departamento
                                and o.codigo_unidad=u.codigo_unidad and
u.codigo_unidad=@tipoObra
                                end
                                end
                                else
                                begin
                                if @municipio<>-1
                                begin
                                SELECT distinct u.* FROM obras_fisicas o,
                                proyectos p,
                                departamentos d, municipios m, cantones c,
unidades_medida u
                                WHERE o.codigo_obra=p.codigo_obra
                                and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
                                and
m.codigo_departamento=d.codigo_departamento

```

```

                                and d.codigo_departamento=@departamento
                                and o.codigo_unidad=u.codigo_unidad and
u.codigo_unidad=@tipoObra
                                and m.codigo_municipio=@municipio
                                end
                                else
                                begin
                                SELECT distinct u.* FROM obras_fisicas o,
                                proyectos p,
                                departamentos d, municipios m, cantones c,
                                unidades_medida u
                                WHERE o.codigo_obra=p.codigo_obra
                                and o.codigo_canton=c.codigo_canton and
                                c.codigo_municipio=m.codigo_municipio
                                and
                                m.codigo_departamento=d.codigo_departamento
                                and d.codigo_departamento=@departamento
                                and o.codigo_unidad=u.codigo_unidad and
                                u.codigo_unidad=@tipoObra
                                end
                                end
                                end
                                else
                                begin
                                if @etapa<>-1
                                begin
                                if @municipio<>-1
                                begin
                                SELECT distinct u.* FROM obras_fisicas o,
                                proyectos p, municipios m, cantones c,
                                etapas e, unidades_medida u
                                WHERE o.codigo_obra=p.codigo_obra
                                and p.codigo_etapa=e.codigo_etapa
                                and o.codigo_canton=c.codigo_canton and
                                c.codigo_municipio=m.codigo_municipio
                                and e.codigo_etapa=@etapa
                                and o.codigo_unidad=u.codigo_unidad and
                                u.codigo_unidad=@tipoObra
                                and m.codigo_municipio=@municipio
                                end
                                end
                                end
                                else
                                begin
                                SELECT distinct u.* FROM obras_fisicas o,
                                proyectos p,
                                etapas e, unidades_medida u
                                WHERE o.codigo_obra=p.codigo_obra

```

```

and p.codigo_etapa=e.codigo_etapa
and e.codigo_etapa=@etapa
and o.codigo_unidad=u.codigo_unidad and
u.codigo_unidad=@tipoObra
end
end
else
begin
if @municipio<>-1
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p, municipios m, cantones c,
unidades_medida u
WHERE o.codigo_obra=p.codigo_obra
and o.codigo_unidad=u.codigo_unidad and
u.codigo_unidad=@tipoObra
and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
and m.codigo_municipio=@municipio
end
else
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p,
unidades_medida u
WHERE o.codigo_obra=p.codigo_obra
and o.codigo_unidad=u.codigo_unidad and
u.codigo_unidad=@tipoObra
end
end
end
end
else
begin
if @departamento<>-1
begin
if @etapa<>-1
begin
if @municipio<>-1
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p, responsables r,
departamentos d, municipios m, cantones c, etapas e,
unidades_medida u

```

```

WHERE o.codigo_obra=p.codigo_obra

and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
and
m.codigo_departamento=d.codigo_departamento
and p.codigo_etapa=e.codigo_etapa
and e.codigo_etapa=@etapa
and d.codigo_departamento=@departamento
and o.codigo_unidad=u.codigo_unidad
and m.codigo_municipio=@municipio
end
else
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p, responsables r,
departamentos d, municipios m, cantones c, etapas e,
unidades_medida u

WHERE o.codigo_obra=p.codigo_obra

and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
and
m.codigo_departamento=d.codigo_departamento
and p.codigo_etapa=e.codigo_etapa
and e.codigo_etapa=@etapa
and d.codigo_departamento=@departamento
and o.codigo_unidad=u.codigo_unidad
end
end
else
begin
if @municipio<>-1
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p,
departamentos d, municipios m, cantones c,
unidades_medida u

WHERE o.codigo_obra=p.codigo_obra
and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
and
m.codigo_departamento=d.codigo_departamento
and d.codigo_departamento=@departamento
and o.codigo_unidad=u.codigo_unidad
and m.codigo_municipio=@municipio

```

```

end
else
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p,
departamentos d, municipios m, cantones c,
unidades_medida u
WHERE o.codigo_obra=p.codigo_obra
and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
and
m.codigo_departamento=d.codigo_departamento
and d.codigo_departamento=@departamento
and o.codigo_unidad=u.codigo_unidad
end
end
end
else
begin
if @etapa<>-1
begin
if @municipio<>-1
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p, municipios m, cantones c,
etapas e, unidades_medida u
WHERE o.codigo_obra=p.codigo_obra
and p.codigo_etapa=e.codigo_etapa
and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
and e.codigo_etapa=@etapa
and o.codigo_unidad=u.codigo_unidad
and m.codigo_municipio=@municipio
end
else
begin
SELECT distinct u.* FROM obras_fisicas o,
proyectos p,
etapas e, unidades_medida u
WHERE o.codigo_obra=p.codigo_obra
and p.codigo_etapa=e.codigo_etapa
and e.codigo_etapa=@etapa
and o.codigo_unidad=u.codigo_unidad
end
end
end
else

```

```

begin
  if @municipio<>-1
  begin
    SELECT distinct u.* FROM obras_fisicas o,
    proyectos p, municipios m, cantones c,
    unidades_medida u
    WHERE o.codigo_obra=p.codigo_obra
    and o.codigo_unidad=u.codigo_unidad
    and o.codigo_canton=c.codigo_canton and
c.codigo_municipio=m.codigo_municipio
    and m.codigo_municipio=@municipio
  end
  else
  begin
    SELECT distinct u.* FROM obras_fisicas o,
    proyectos p,
    unidades_medida u
    WHERE o.codigo_obra=p.codigo_obra
    and o.codigo_unidad=u.codigo_unidad
  end
end
end
end
end
end

```

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

/\*\*\*/

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

```

```

ALTER procedure RetornarTipoObra

```

```

@buscarpor char(1),
@codigo varchar(3),
@consulta char(1)
as

if @buscarpor='N'
begin
    if @consulta='0'
        Select Count(*) from Unidades_Medida
    else
        Select * From Unidades_Medida
end
else
begin
    if @buscarpor='d'
    begin
        if @consulta='0'
        begin
            Select Count(*) From Unidades_Medida
            Where codigo_unidad in
            (
                Select o.codigo_unidad
                From Proyectos p, Obras_Fisicas o, cantones c, municipios
m, departamentos d
                Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
                and c.codigo_municipio=m.codigo_municipio
                and m.codigo_departamento=d.codigo_departamento
                and d.codigo_departamento=convert(int,@codigo)
            )
        end
        else
        begin
            Select * From Unidades_Medida
            Where codigo_unidad in
            (
                Select o.codigo_unidad
                From Proyectos p, Obras_Fisicas o, cantones c, municipios
m, departamentos d
                Where p.codigo_obra=o.codigo_obra and
o.codigo_canton=c.codigo_canton
                and c.codigo_municipio=m.codigo_municipio
                and m.codigo_departamento=d.codigo_departamento
                and d.codigo_departamento=convert(int,@codigo)
            )
        end
    end
end

```

```

end
else
begin
    if @buscarpor='i'
    begin
        if @consulta='0'
        begin
            Select Count(*) From Unidades_Medida
            Where codigo_unidad in
            (
                Select o.codigo_unidad
                From Proyectos p, Obras_Fisicas o, Responsables as
                r, Instituciones i

                Where p.codigo_obra=o.codigo_obra
                and p.codigo_responsable=r.codigo_responsable
                and r.codigo_institucion=i.codigo_institucion
                and i.codigo_institucion=@codigo
            )
        end
        else
        begin
            Select * From Unidades_Medida
            Where codigo_unidad in
            (
                Select o.codigo_unidad
                From Proyectos p, Obras_Fisicas o, Responsables as
                r, Instituciones i

                Where p.codigo_obra=o.codigo_obra
                and p.codigo_responsable=r.codigo_responsable
                and r.codigo_institucion=i.codigo_institucion
                and i.codigo_institucion=@codigo
            )
        end
    end
end
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

/\*- - - - -\*/

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure TiposObra @unidad int,@institucion int,@anio int,@usuario
varchar(10)
```

```
as
```

```
if @institucion=999
```

```
BEGIN
```

```
select o.codigo_obra,o.nombre_obra,
convert(numeric(20,2),0) as [Empleos],
sum(s.programado_financiero) as [Monto Programado],
sum(s.ejecucion_financiero) as [Monto Ejecutado]
into #RESULTADO1
```

```
from obras_fisicas o,seguimiento s,unidades_medida u
where o.codigo_obra=s.codigo_obra
and o.codigo_unidad=u.codigo_unidad
and u.codigo_unidad=@unidad
and s.año=@anio
group by o.codigo_obra,o.nombre_obra
```

```
select o.codigo_obra,sum(p.empleos_proyecto)as empleos
into #RESULTADO2
```

```
from obras_fisicas o,proyectos p,unidades_medida u
where o.codigo_obra=p.codigo_obra
and o.codigo_unidad=u.codigo_unidad
and u.codigo_unidad=@unidad
group by o.codigo_obra
```

```
update #RESULTADO1 set empleos=#RESULTADO2.empleos
from #RESULTADO1,#RESULTADO2
where #RESULTADO1.codigo_obra=#RESULTADO2.codigo_obra
```

```
select '<a
```

```
href=/servlet/ReportesListado?CODIGO=04&USUARIO='+@usuario+'&PARAMETRO
='+convert(varchar(5),codigo_obra)+'&INSTITUCION='+convert(varchar(3),@institucio
n)+'&ANIO='+convert(varchar(4),@anio)+'>'+nombre_obra+'</a>' as [Obra],
```

```
empleos as [Empleos Generados],
[Monto Programado],
```

[Monto Ejecutado]  
from #RESULTADO1

```
END
ELSE
BEGIN
select o.codigo_obra,o.nombre_obra,
convert(numeric(20,2),0) as [Empleos],
sum(s.programado_financiero) as [Monto Programado],
sum(s.ejecucion_financiero) as [Monto Ejecutado]
into #RESULTADO3
from obras_fisicas o,seguimiento s,proyectos p,responsables r,unidades_medida u
where o.codigo_obra=s.codigo_obra
and o.codigo_obra=p.codigo_obra
and p.codigo_proyecto=s.codigo_proyecto
and p.codigo_obra=s.codigo_obra
and p.codigo_responsable=r.codigo_responsable
and r.codigo_institucion=@institucion
and o.codigo_unidad=u.codigo_unidad
and u.codigo_unidad=@unidad
and s.año=@año
group by o.codigo_obra,o.nombre_obra
```

```
select o.codigo_obra,sum(p.empleos_proyecto)as empleos
into #RESULTADO4
from obras_fisicas o,proyectos p,unidades_medida u
where o.codigo_obra=p.codigo_obra
and o.codigo_unidad=u.codigo_unidad
and u.codigo_unidad=@unidad
group by o.codigo_obra
```

```
update #RESULTADO3 set empleos=#RESULTADO4.empleos
from #RESULTADO3,#RESULTADO4
where #RESULTADO3.codigo_obra=#RESULTADO4.codigo_obra
```

```
select '<a
href=/servlet/ReportesListado?CODIGO=04&USUARIO='+@usuario+'&PARAMETRO
='+convert(varchar(5),codigo_obra)+'&INSTITUCION='+convert(varchar(3),@institucio
n)+'&ANIO='+convert(varchar(4),@año)+'>'+nombre_obra+'</a>' as [Obra],
empleos as [Empleos Generados],
[Monto Programado],
[Monto Ejecutado]
from #RESULTADO3
END
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/*-----*/
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
ALTER procedure ToolTipObra @obra int
as
```

```
select o.codigo_obra, o.nombre_obra,
convert(numeric(25,2),0) as inversion,
convert(numeric(25,2),0) as proyectos,
convert(numeric(25,2),0) as necesidades,
convert(numeric(25,2),0) as empleos,
convert(numeric(25,2),0) as porcentaje
into #RESULTADO
from obras_fisicas o
where o.codigo_obra=@obra
```

```
select p.codigo_obra,sum(s.programado_financiero) as inversion,
porcentaje=
    Case sum(s.ejecucion_financiero)
    When 0 then 0.00 else
sum(s.programado_financiero)/sum(s.ejecucion_financiero)*100
    end
into #RESULTADO1
from proyectos p,seguimiento s
where s.codigo_obra=p.codigo_obra
and s.codigo_proyecto=p.codigo_proyecto
and p.codigo_obra=@obra
group by p.codigo_obra
```

```
update #RESULTADO set inversion=#RESULTADO1.inversion,
porcentaje=#RESULTADO1.porcentaje
from #RESULTADO,#RESULTADO1
where #RESULTADO.codigo_obra=#RESULTADO1.codigo_obra
```

```
select codigo_obra,count(codigo_proyecto) as proyectos into #RESULTADO2 from
proyectos where codigo_obra=@obra group by codigo_obra
select codigo_obra,count(codigo_necesidad) as necesidades into #RESULTADO3 from
necesidades where codigo_obra=@obra group by codigo_obra
select codigo_obra,sum(empleos_proyecto) as empleos into #RESULTADO4 from
proyectos where codigo_obra=@obra group by codigo_obra
```

```
update #RESULTADO set proyectos =#RESULTADO2.proyectos
from #RESULTADO,#RESULTADO2
where #RESULTADO.codigo_obra=#RESULTADO2.codigo_obra
```

```
update #RESULTADO set necesidades =#RESULTADO3.necesidades
from #RESULTADO,#RESULTADO3
where #RESULTADO.codigo_obra=#RESULTADO3.codigo_obra
```

```
update #RESULTADO set empleos =#RESULTADO4.empleos
from #RESULTADO,#RESULTADO4
where #RESULTADO.codigo_obra=#RESULTADO4.codigo_obra
```

```
select nombre_obra,convert(varchar(20),convert(money,inversion),1),
proyectos,necesidades,empleos,
convert(varchar(7),porcentaje)+'%'
from #RESULTADO
```

```
--exec ToolTipObra 32
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/**/
```

```
CREATE TRIGGER CambiarMenu ON [dbo].[MENU]
FOR INSERT, UPDATE
as
declare @cod_men varchar(20)
declare @sup_men varchar(20)

select @cod_men=codigo_menu, @sup_men=superior_menu from inserted

if @sup_men='00'
update menu set superior_menu=@cod_men where codigo_menu=@cod_men
```

```
/**/
```

```
CREATE TRIGGER NecesidadPorDefecto ON [dbo].[OBRAS_FISICAS]
FOR INSERT
AS
declare @codigo_obra int
select @codigo_obra=codigo_obra from inserted

declare @diahoy datetime
set @diahoy=getdate()

declare @admin char(10)

select @admin=codigo_responsable from Perfiles_Responsables where
codigo_perfil='ADMIN'

insert into necesidades values(@codigo_obra,1,'Ninguna','Ninguna',0,@diahoy,@admin)
```