Generando reportes gráficos con Jfreechart en Netbeans 5.5

Autor: Javier Luna Velásquez [1]
javierluna55@gmail.com
Fecha: 05 de julio del 2007

I.- Introducción

Cuando se piensa el porqué se desarrolla sistemas de información en una organización, la respuesta parece bastante obvia, para registrar y controlar los movimientos y transacciones que se van generando con el correr del tiempo, sin embargo existe aun un motivo mas importante y no tan obvio, es que la información puede usarse para respaldar la toma de decisiones. Sin importar el tamaño de una empresa u organización, la conducción exitosa de la misma estará drásticamente influenciada por la precisión de sus registros y la adopción de decisiones acertadas.

Precisamente si nuestra base de datos no contiene información precisa, las salidas del sistema arrojarán datos erróneos, llevando esto a una toma de decisiones ineficiente.

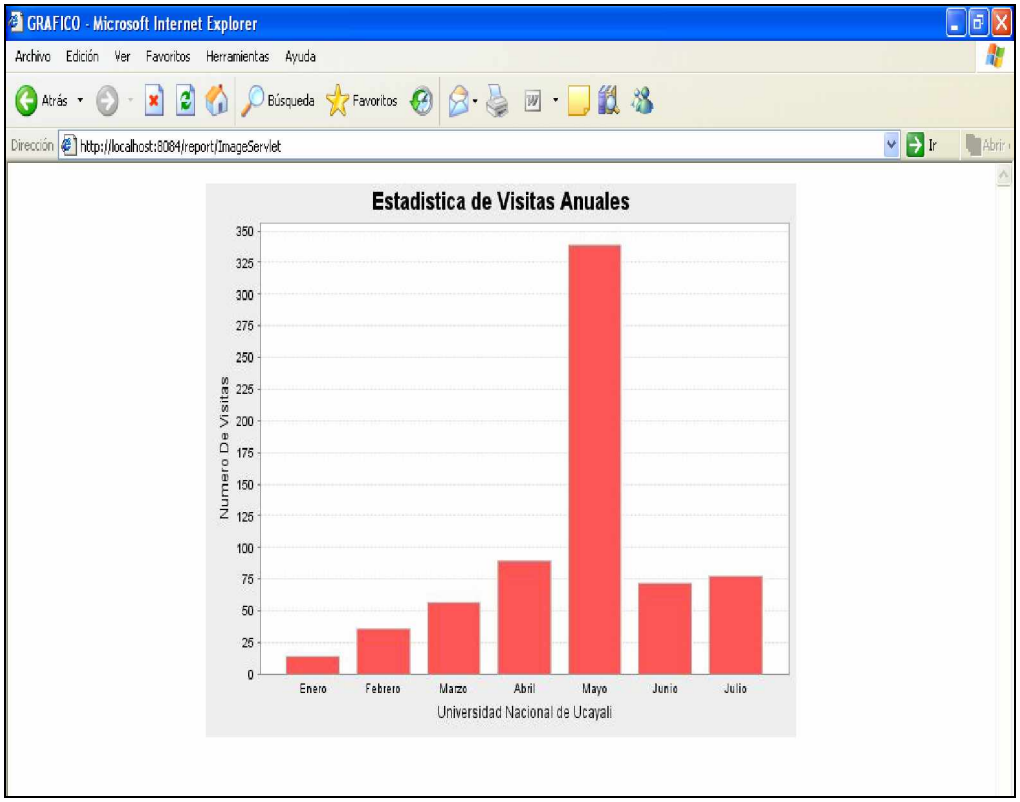
Normalmente en un sistema de información las salidas o reportes se usan para mostrarle al usuario en forma resumida las transacciones que se han almacenado en un tiempo determinado (diario, mensual, anual) y en base a ello pueda tomar una acción determinada.

Estos reportes se pueden mostrar en forma de texto pero está comprobado que es mucho más fácil y rápido interpretar una imagen, así que si algunos de nuestros reportes arrojaran gráficos en lugar de solo texto plano sería de mucha ayuda al momento de analizarlos y tomar alguna decisión en base a ellos.

II.- Generando Reportes Gráficos

El objetivo de este artículo es crear un Reporte en modo gráfico usando la librería jfreechart 1.0.3, el ejemplo consistirá en mostrar un reporte que muestre el número de asistencias anuales que se registraron en una organización determinada, mostrando dicho gráfico por meses.

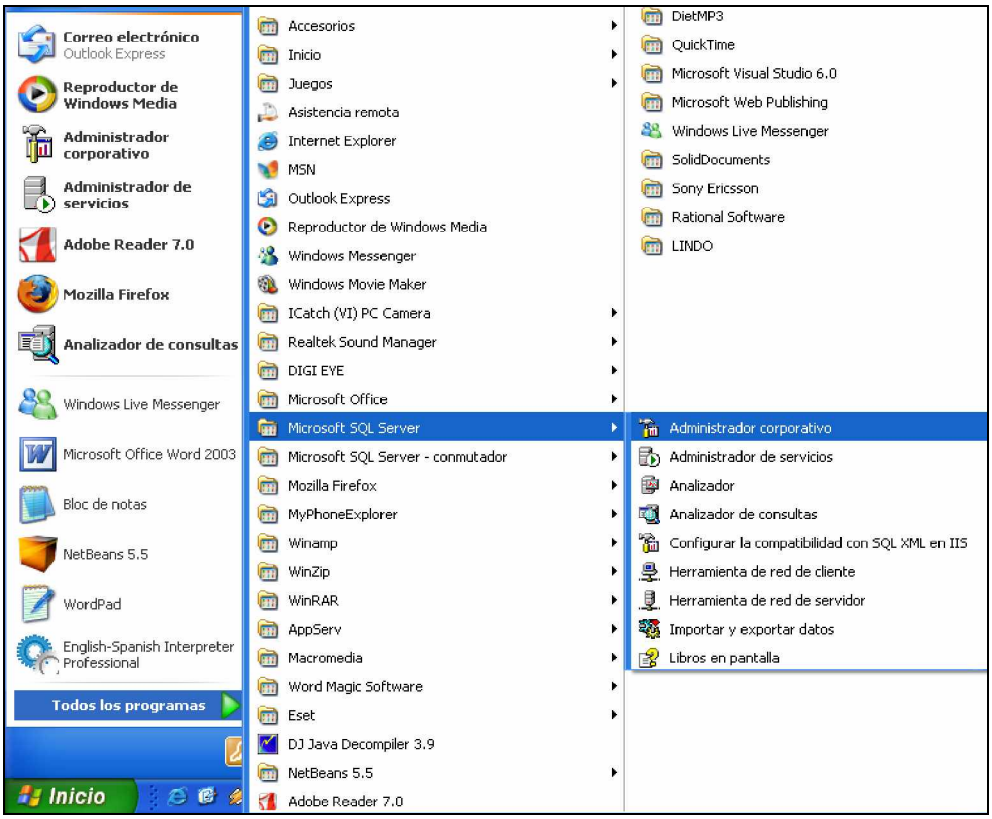
Tendremos que crear nuestra Base de Datos llamada “reporte” que contendrá una tabla llamada “asistencia” y nos conectaremos nativamente al Netbeans para generar el siguiente reporte:



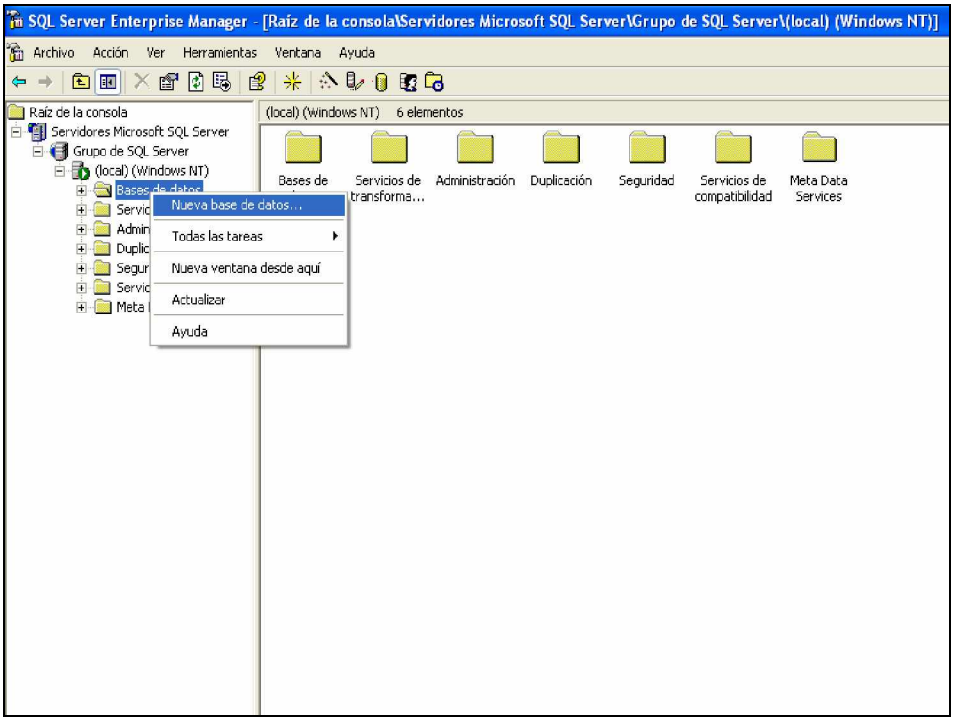
Las herramientas que vamos a usar para el desarrollo de esta aplicación son:

- Ø Entorno de Desarrollo: Netbeans 5.5, para descargarlo http://www.netbeans.org/
- Ø Base de Datos: SQL Server 2000
- Ø Librería jfreechart-1.0.3 y jcommon-1.0.6, para su descarga http://sourceforge.net/project/showfiles.php?group_id=15494&package_id=12428

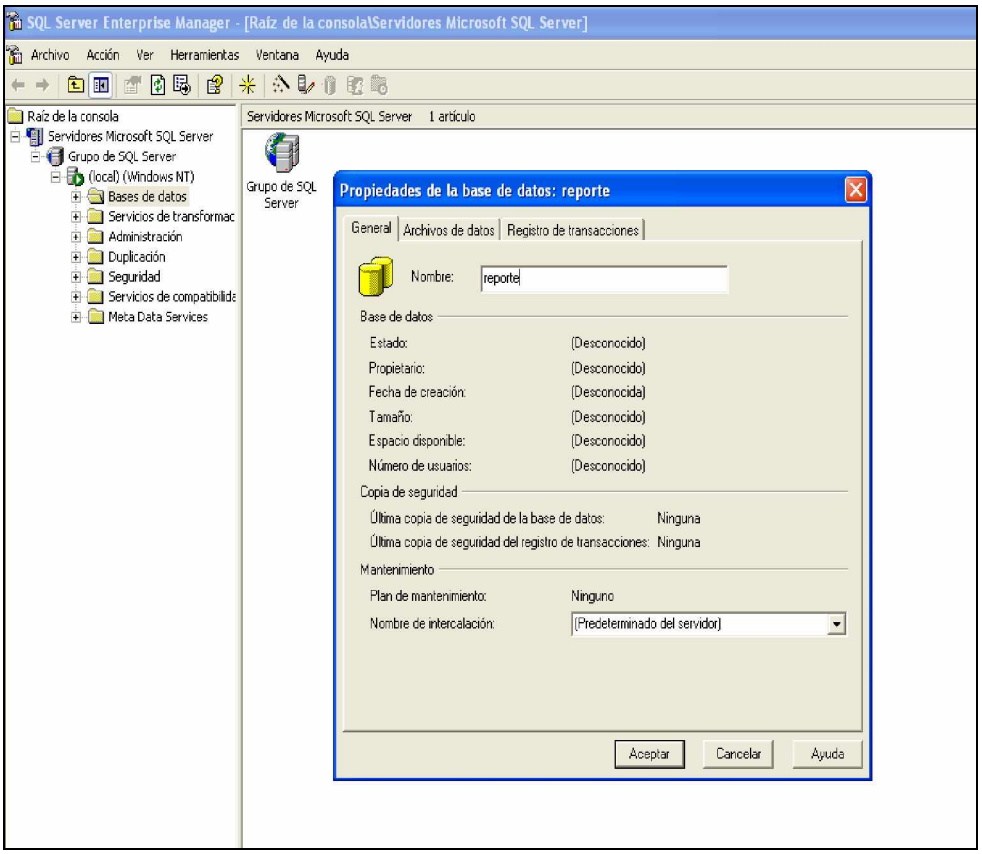
Para empezar vamos a crear nuestra base de datos en SQL Server 2000, para eso entramos al Administrador Corporativo del SQL Server:



Una vez dentro hacemos anticlik en *Base de Datos* y escogemos la opción *Nueva Base de Datos:*



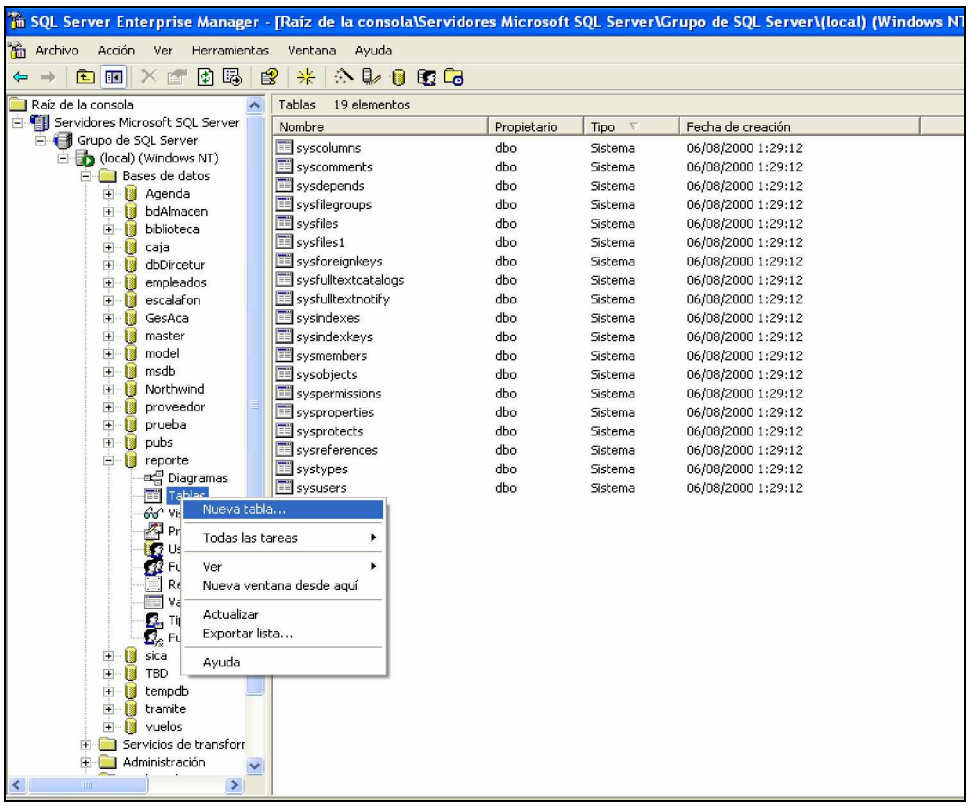
Le ponemos nombre a nuestra base de Datos: “reporte” y le damos clic en *Aceptar*:



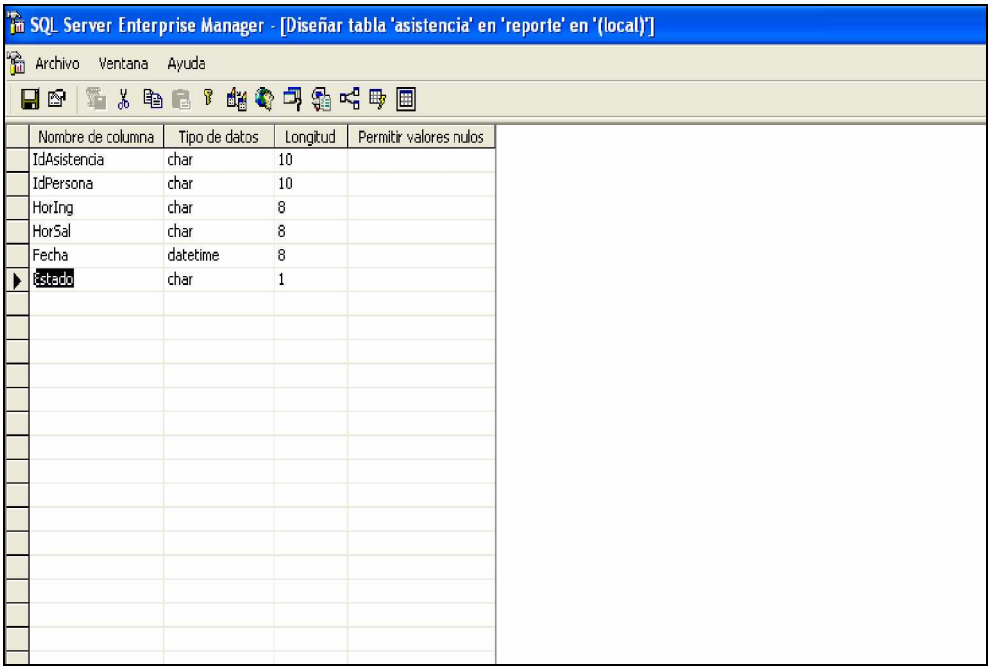
Ahora vamos a crear nuestra Tabla “asistencia”, que nos permitirá registrar las visitas de los usuarios, con la siguiente estructura:

- *IdAsistencia:* Es la Primary Key de la Tabla.
- *IdPersona:* Es un Foreign Key que almacena el código de los usuarios.
- *HorIng:* Almacena la hora de ingreso.
- *HorSal:* Almacena la hora de salida
- *Fecha:* Almacena la fecha de la visita
- *Estado:* Almacena el estado de la asistencia (E=Entrada, S=Salida)

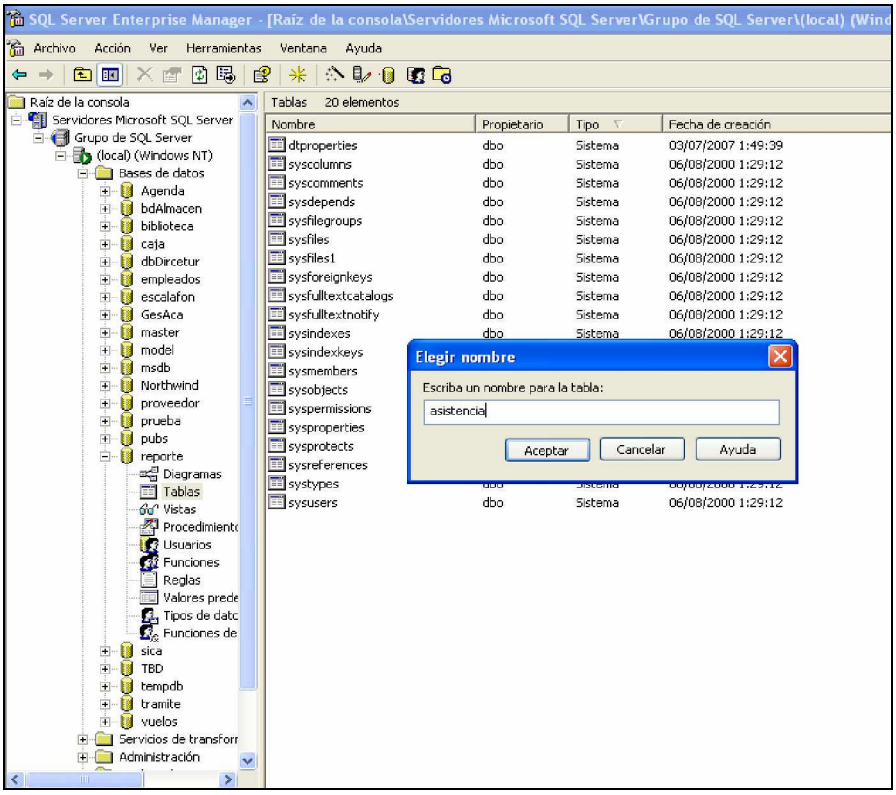
Escogemos nuestra Base de Datos “reporte”, hacemos anticlick en *Tablas* y escogemos *Nueva Tabla*:



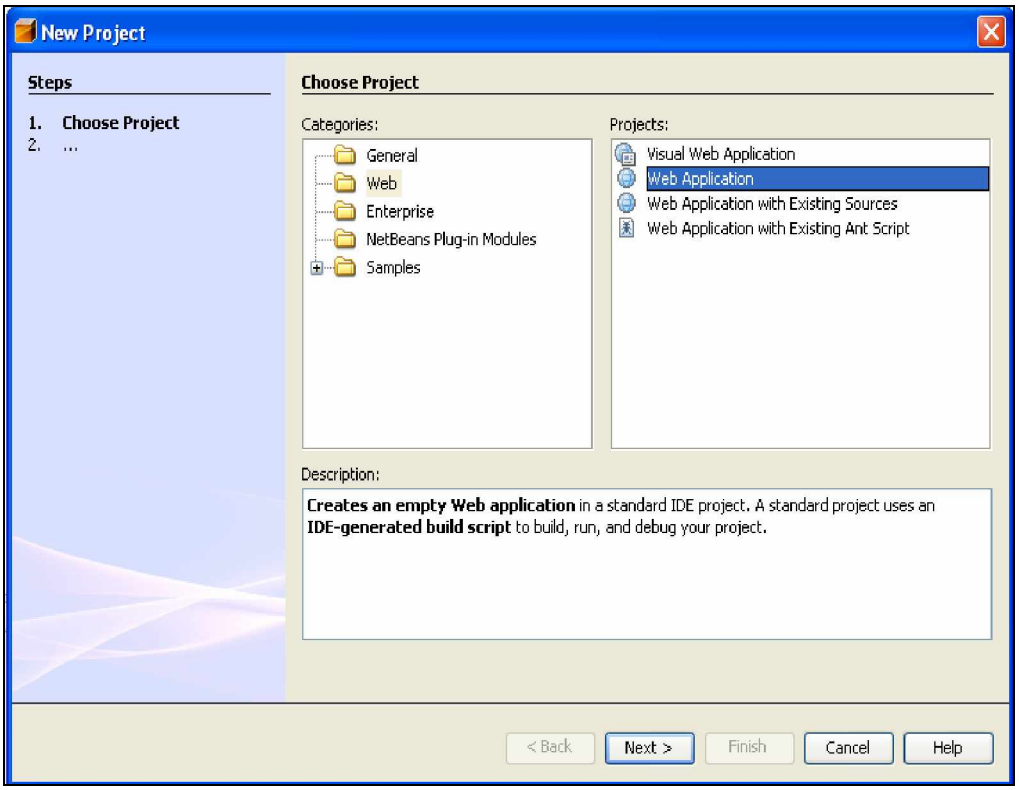
Empezamos a diseñar la tabla con los campos antes mencionados y su tipo de dato respectivo:



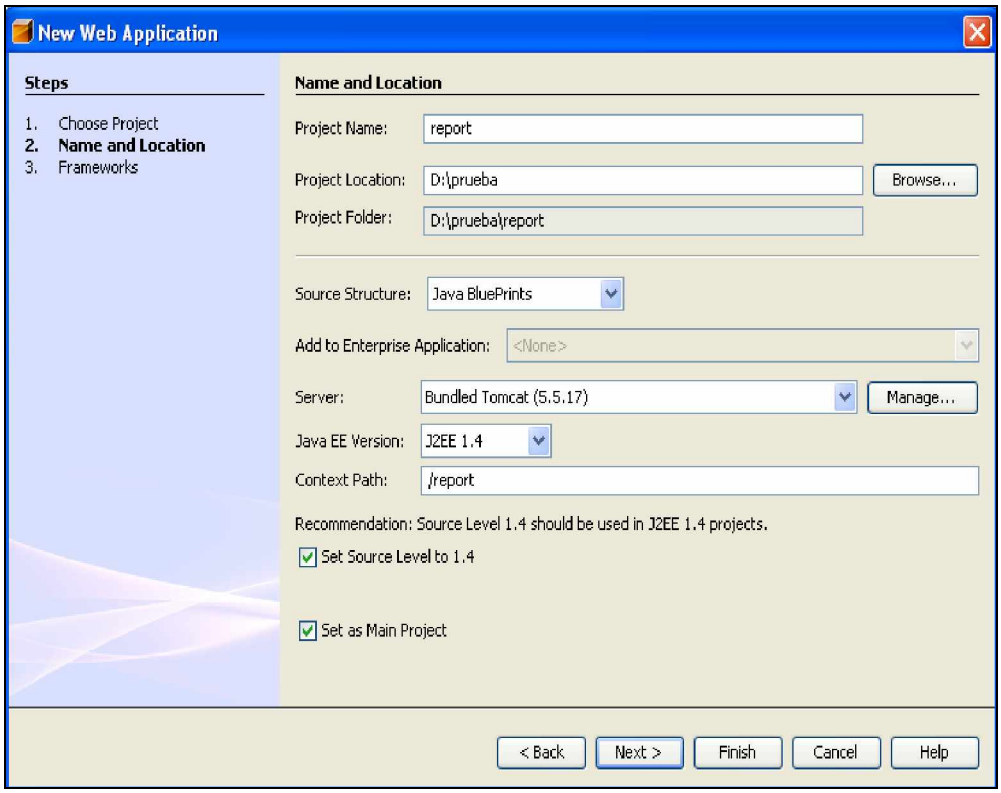
Finalmente el SQL Server nos pedirá un nombre para nuestra tabla:



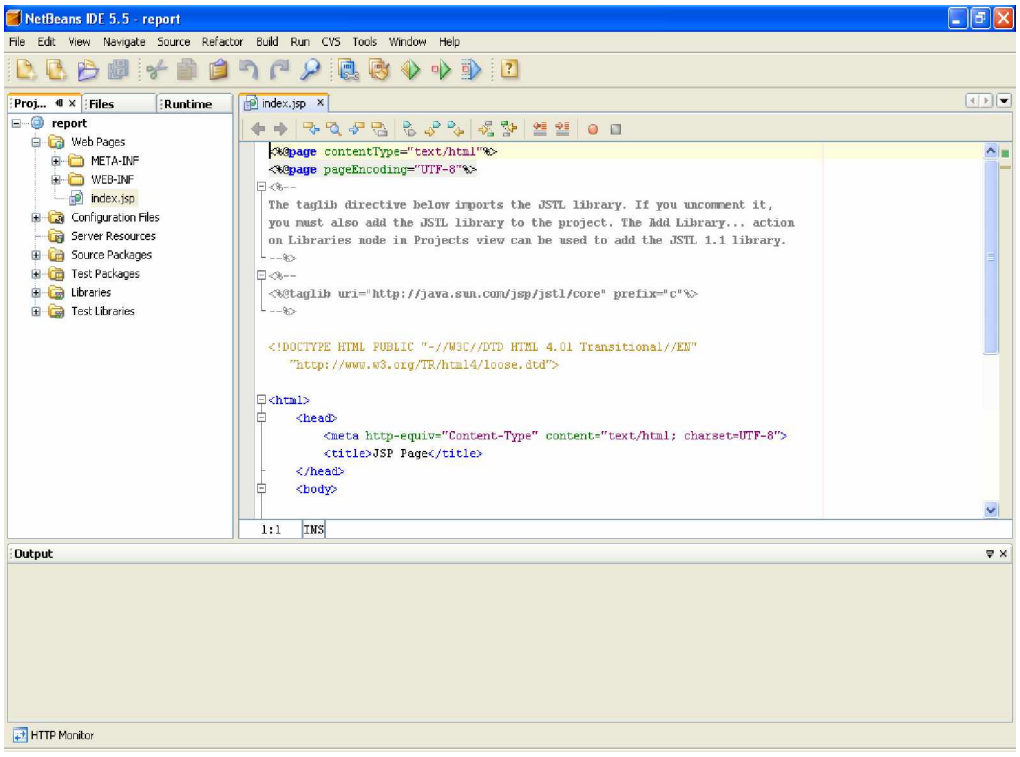
Ahora abrimos el Netbeans 5.5 y creamos un nuevo proyecto Web, para eso hacemos clic en *File* y luego *New Proyect*:



Hacemos clic en *Next* y aparecerá la siguiente pantalla donde poner el nombre de nuestro proyecto y la ubicación:

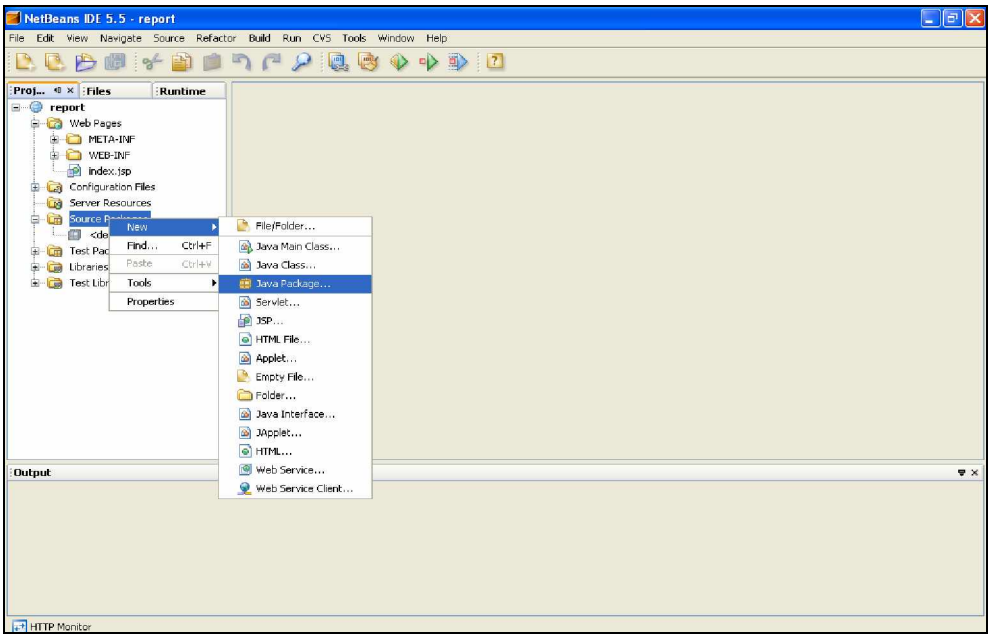


En esta pantalla le damos clic en *Finish*, esta es la pantalla por defecto que nos debe aparecer, esta contiene un *index.jsp:*

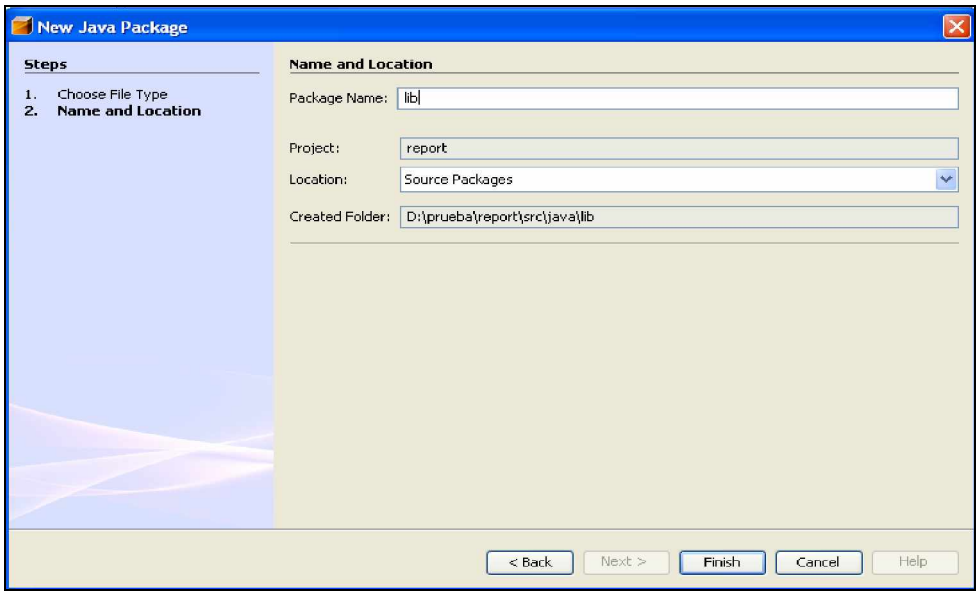


En éste punto nos toca conectarnos al SQL Server 2000 en forma nativa, para ello utilizamos JDBC (Java DataBase Connectivity) la cual es un conjunto de clases e interfaces escritos en Java que ofrecen una API completa para la programacion de Bases de Datos de diferentes proveedores a través de sentencias SQL, difiere con OBDC (Open DataBase Connectivity: API de Microsoft para conectarse a bases de datos) porque JDBC esta escrito 100% en Java en cambio ODBC esta en C, además el rendimiento usando ODBC en bases de datos grandes disminuye.

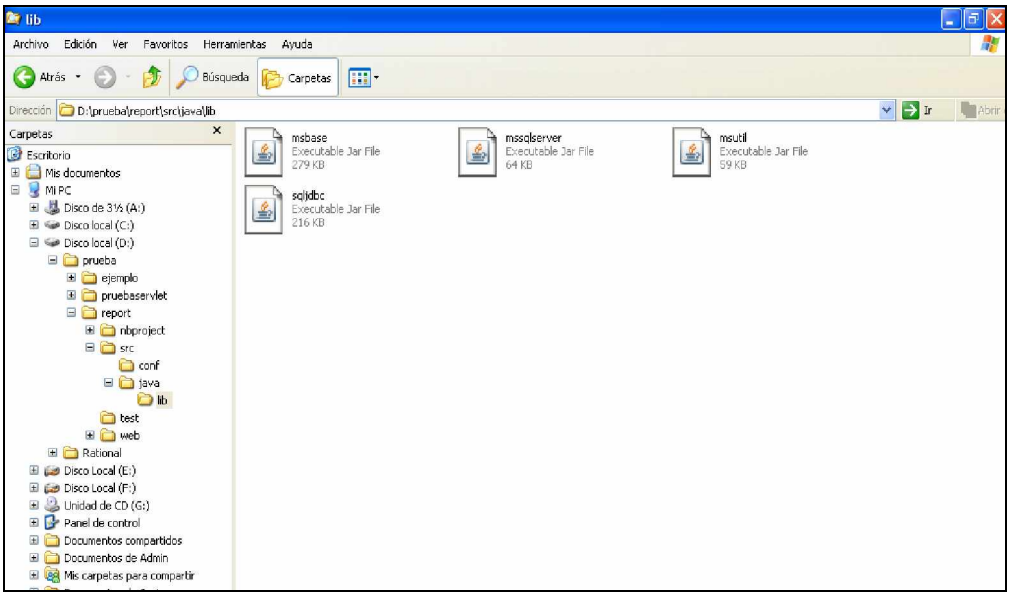
Ahora bien lo que tenemos que hacer es crear un paquete llamado lib, dentro de nuestro proyecto en *Source Package* hacemos anticlik y escogemos *New* y *Java Package:*



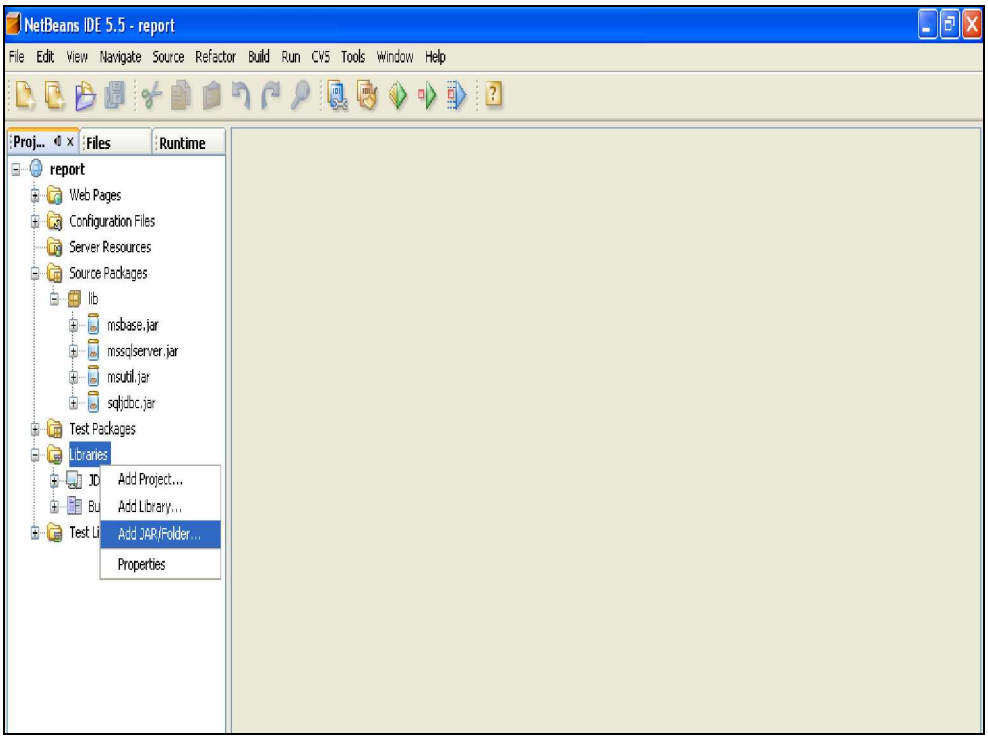
Le ponemos nombre a nuestro paquete y hacemos clic en *Finish*:



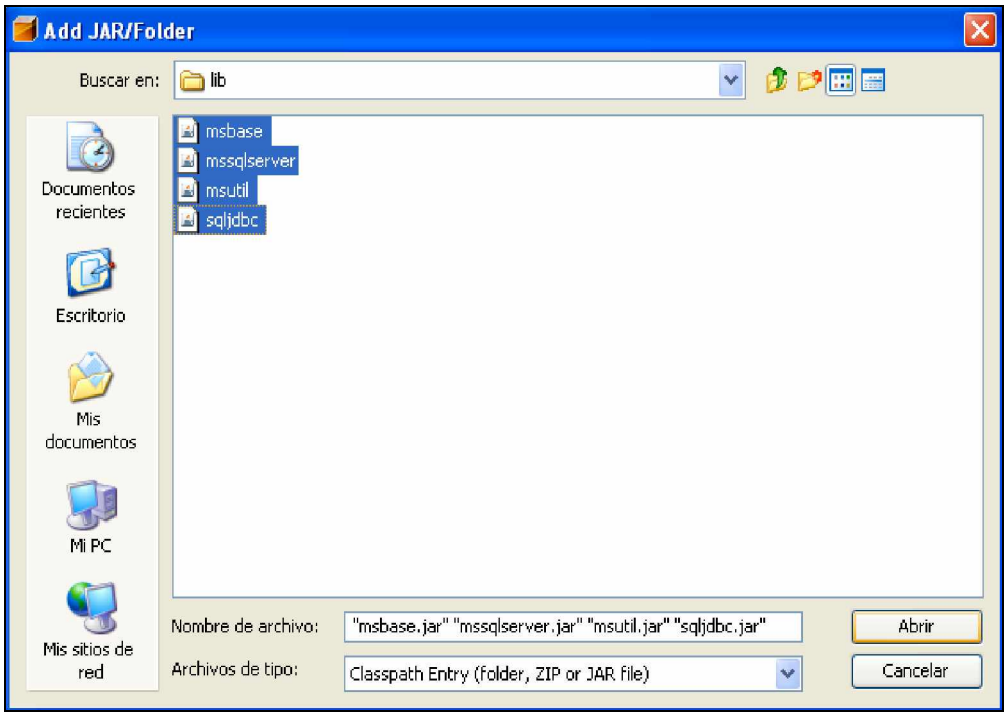
Ahora nos falta copiar los jar de conexión a la carpeta que hemos creado en nuestro proyecto, para ello vamos a la ruta donde lo hemos guardado y buscamos la carpeta *scr*, luego *java* y finalmente *lib*, en mi caso se encuentra en esta dirección: ***D:\prueba\report\src\java\lib***



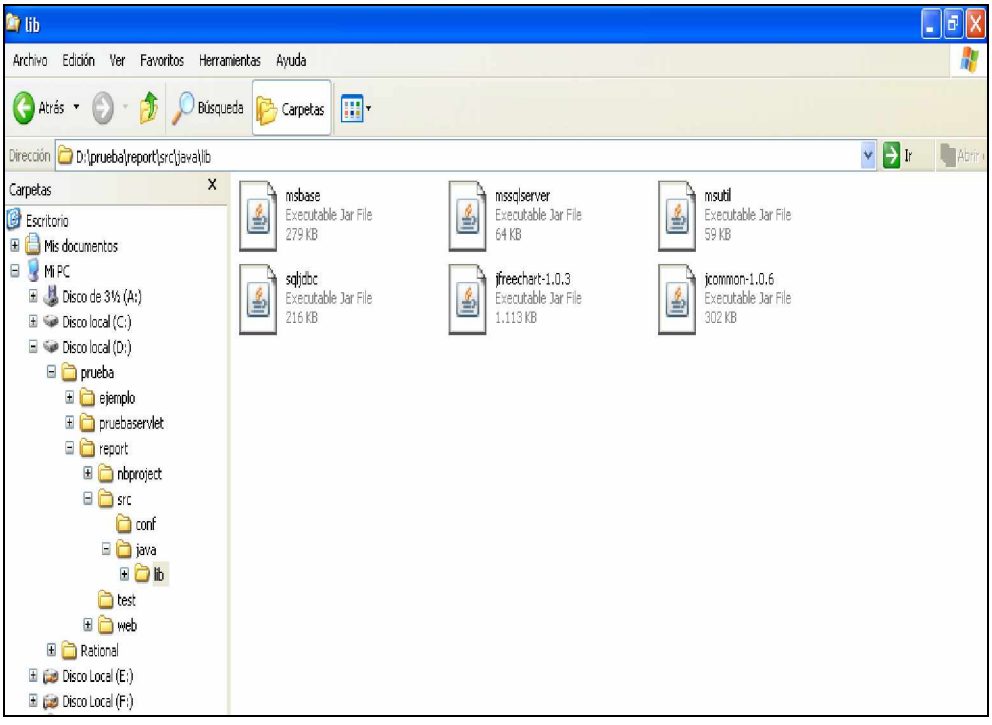
Si regresamos al Netbeans veremos que se han añadido los jar a nuestro paquete *lib*, ahora debemos agregarlos a las librerías del Proyecto, para ello hacemos anticlik en *Libraries* y escogemos la opción *Add Jar/Folder*



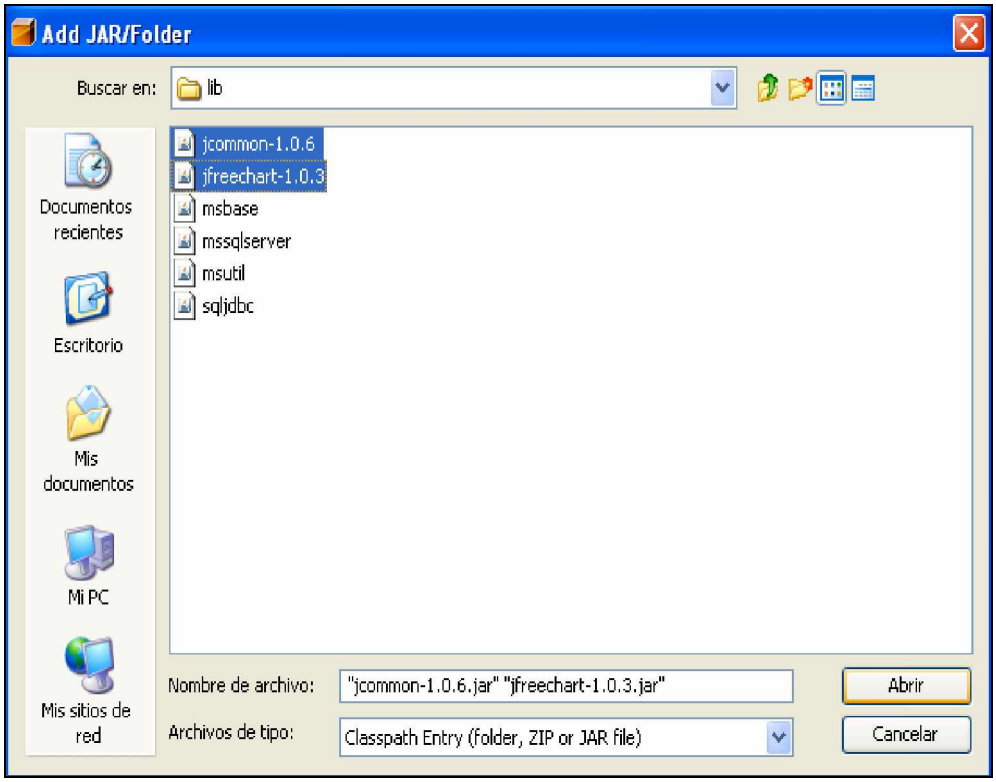
En la pantalla que nos aparece vamos a agregar los jar que anteriormente hemos copiado en el paquete *lib,* los seleccionamos y le damos clic en *Abrir*:



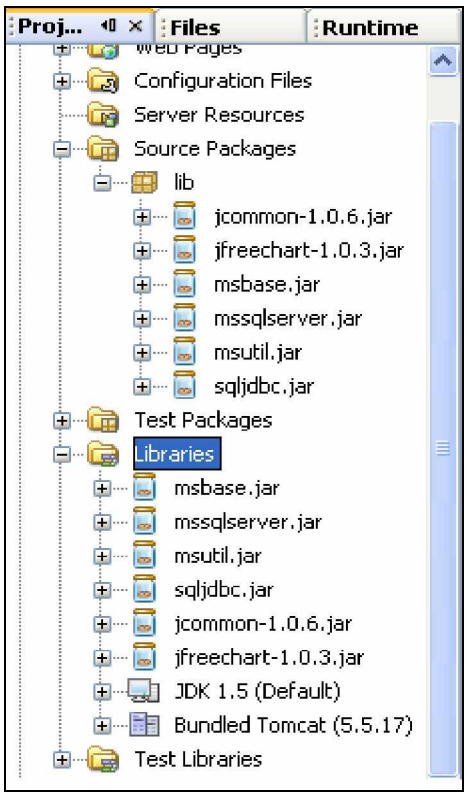
Para la conexión eso es todo lo necesario, ahora debemos agregar los jar que nos permitirán generar los gráficos en el reporte, éstos jar son el jfreechart-1.0.3 y el jcommon-1.0.6 que previamente hemos descargado. Simplemente debemos copiarlos en el paquete *lib (*donde están los jar de conexión*)* creado anteriormente.



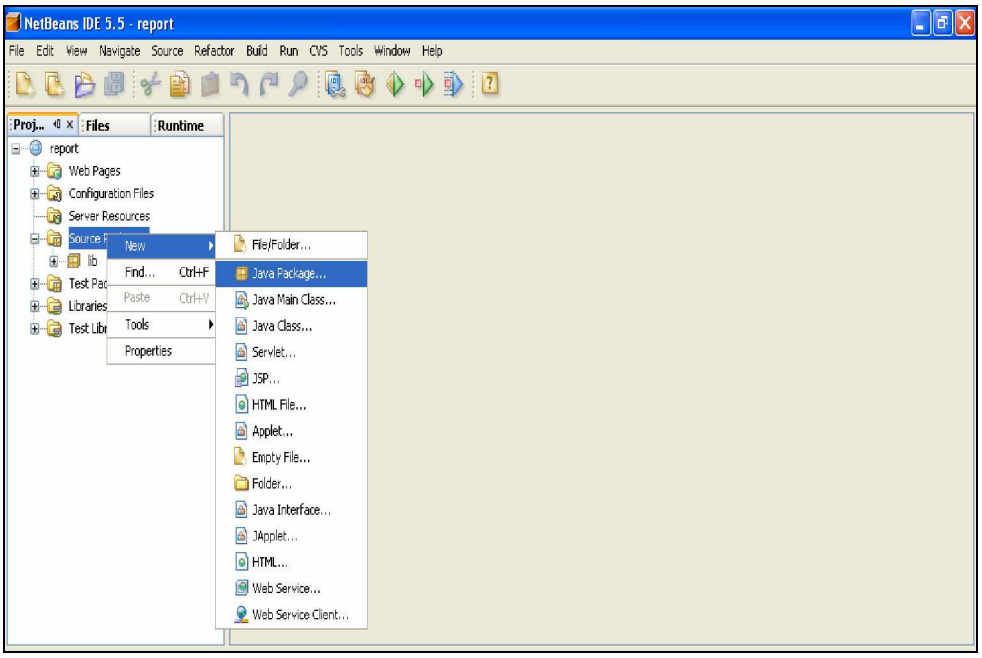
Y finalmente los agregamos a las librerías:



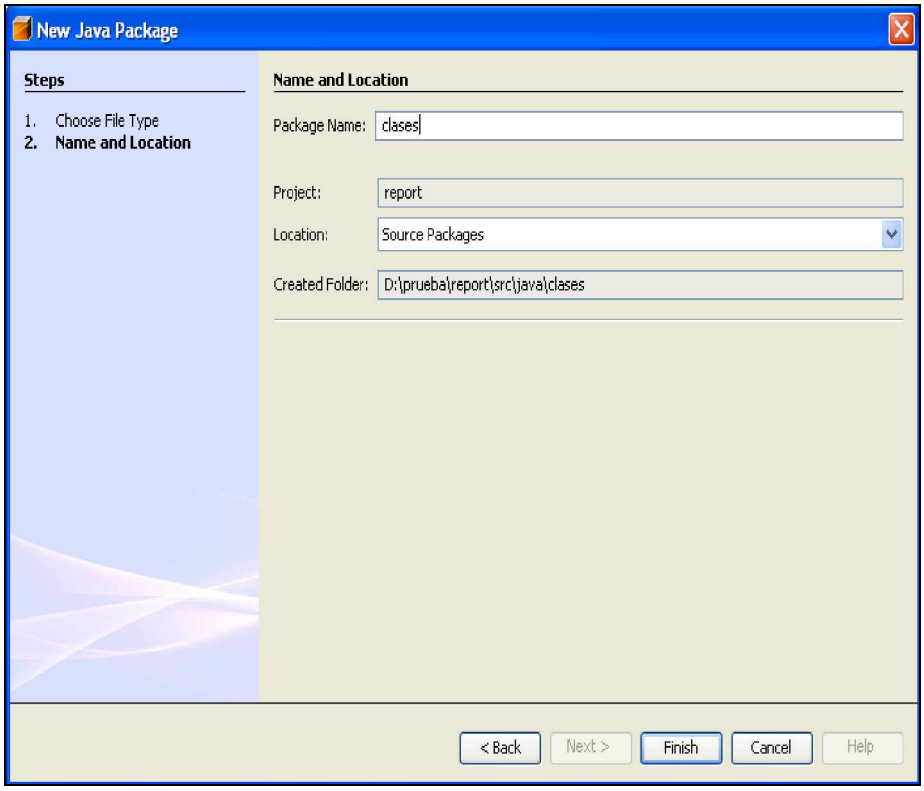
Finalmente con todas las librerías agregadas nuestro proyecto debe quedar así:



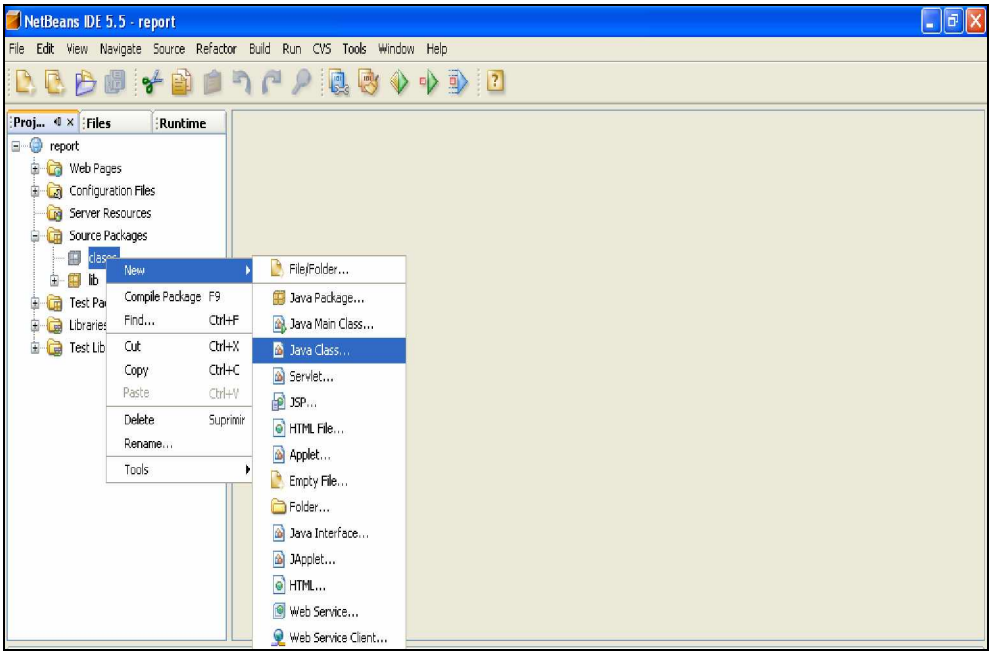
Bien espero que hasta aquí no tengamos ningún problema y podamos seguir adelante, lo que toca es crear nuestras clases: en primera instancia vamos a crear nuestra clase de acceso a la Base de Datos. Para ello vamos a crear un paquete llamado *clases* donde estarán todas las clases, valga la redundancia, a ser usadas en nuestro proyecto. Hacemos anticlik en *Source Packages, New* y luego *Java Package:*



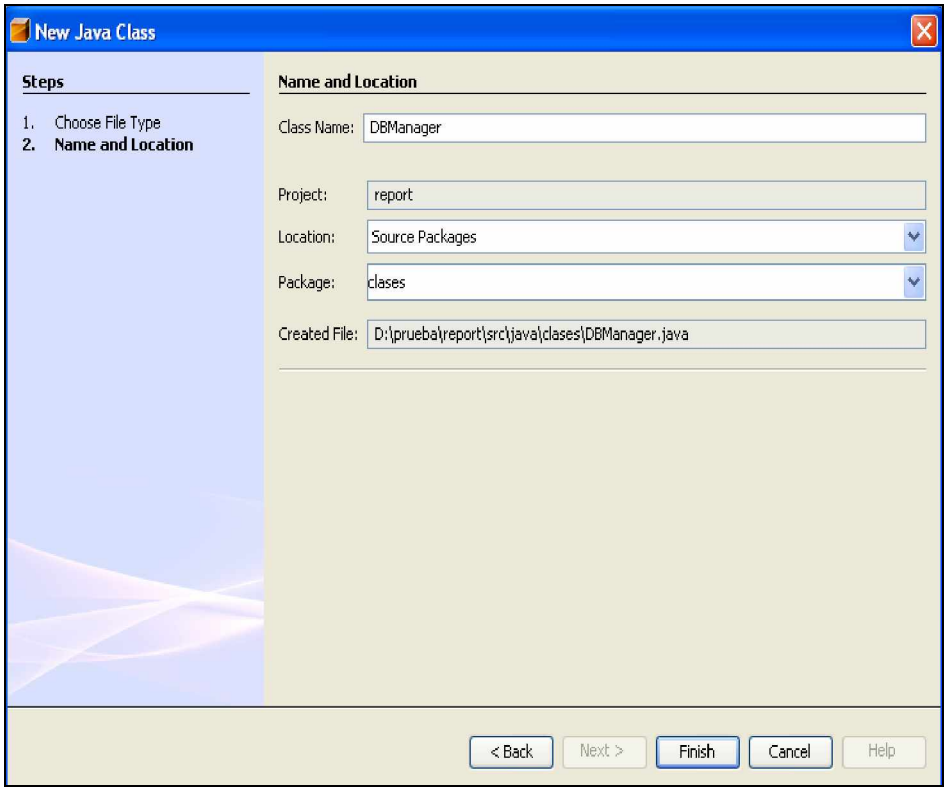
Y ponemos nombre a nuestro paquete y le damos *Finish*:



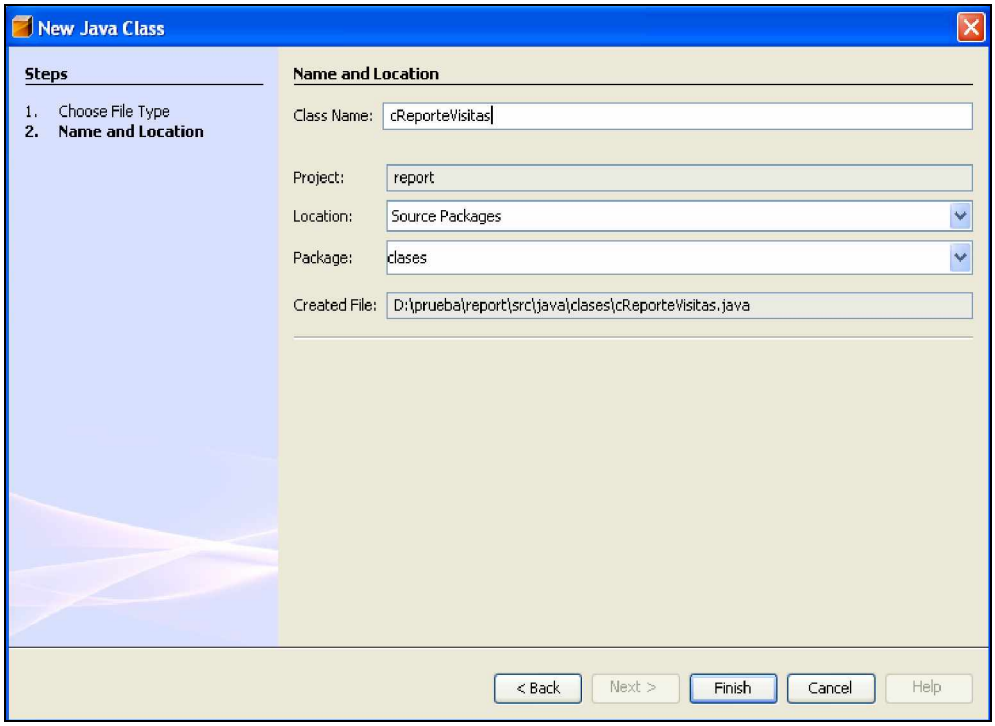
Ahora dentro de ese paquete vamos a crear la clase DBManager para conectarnos al Motor de Base de Datos, hacemos anticlik en el paquete *clases* y escogemos New - *Java Class:*



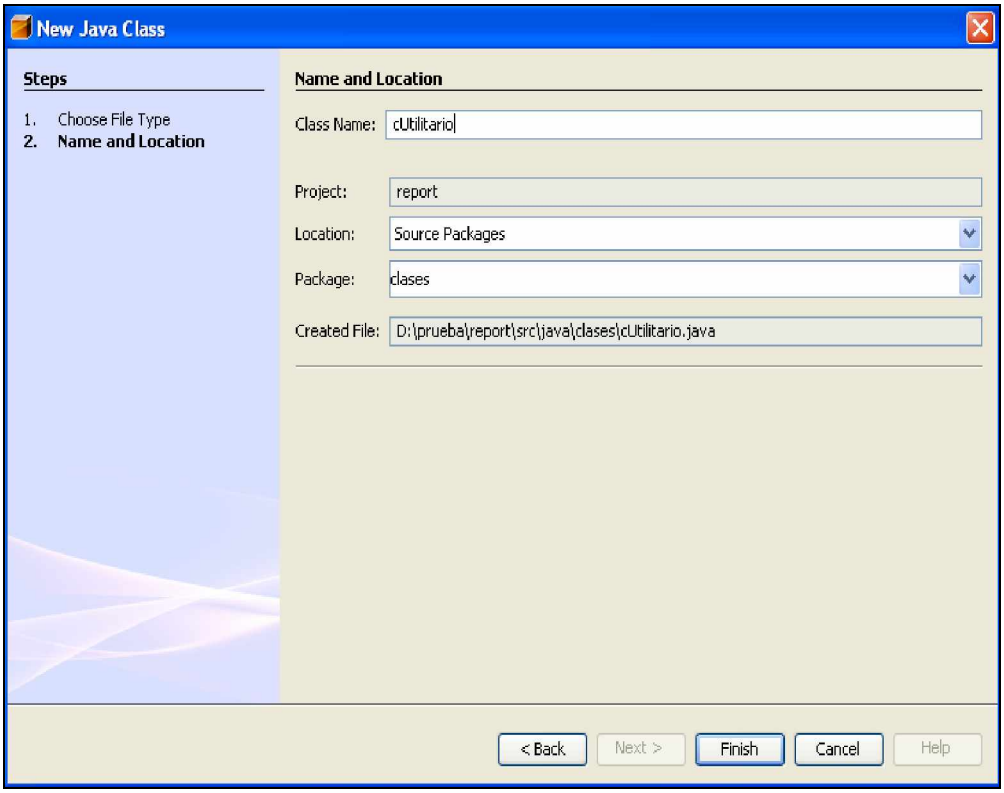
Ponemos nombre a nuestra clase y le damos *Finish*:



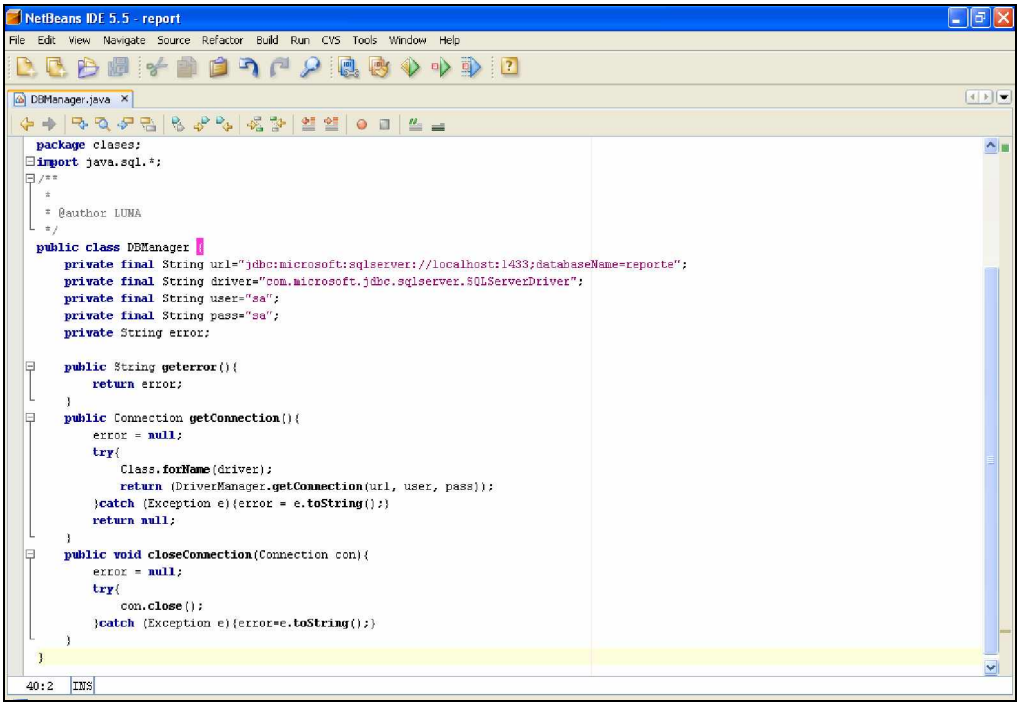
De la misma forma creamos nuestra clase cReporteVisitas que nos permitirá acceder a la tabla asistencia y extraer la información necesaria para luego generar el gráfico:



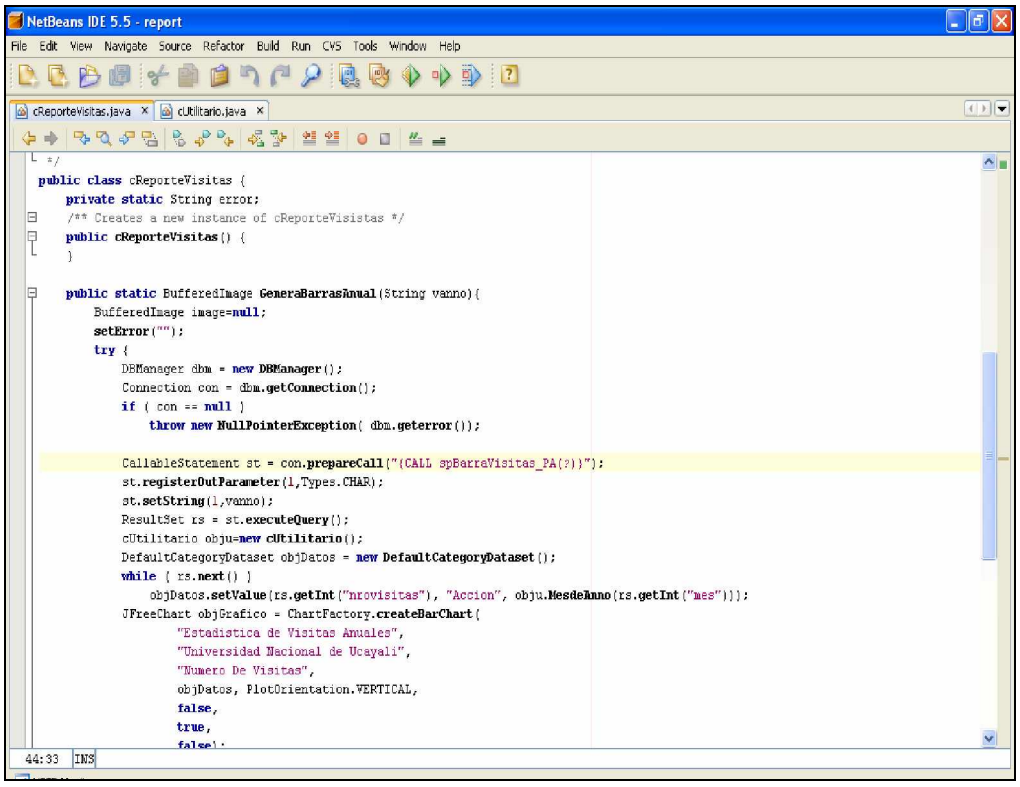
Y también nuestra clase cUtilitario que contendrá un método que devuelve el nombre del mes según el número que le indiquemos como parámetro:



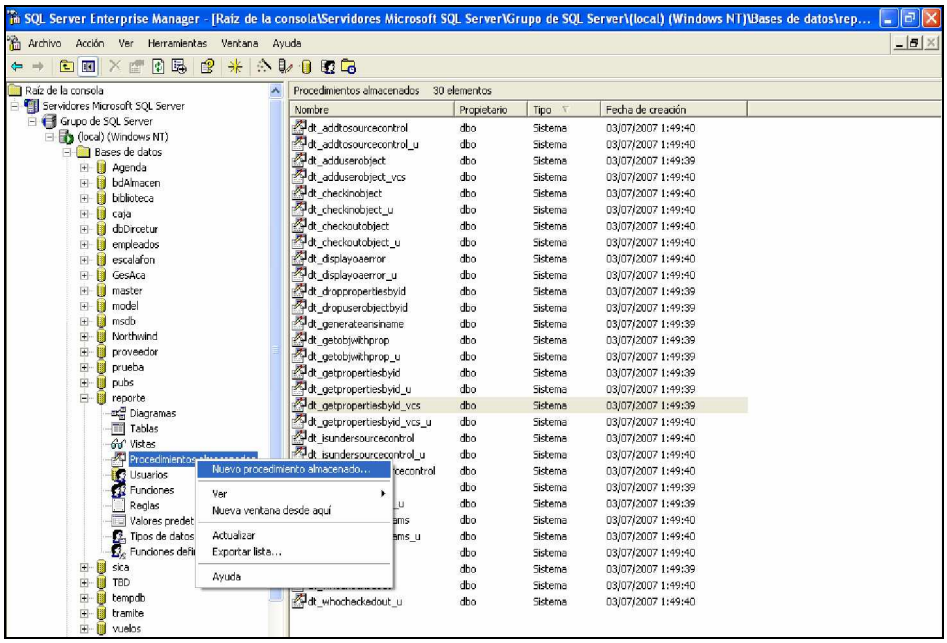
Aquí vemos el código fuente de nuestra clase DBManager:



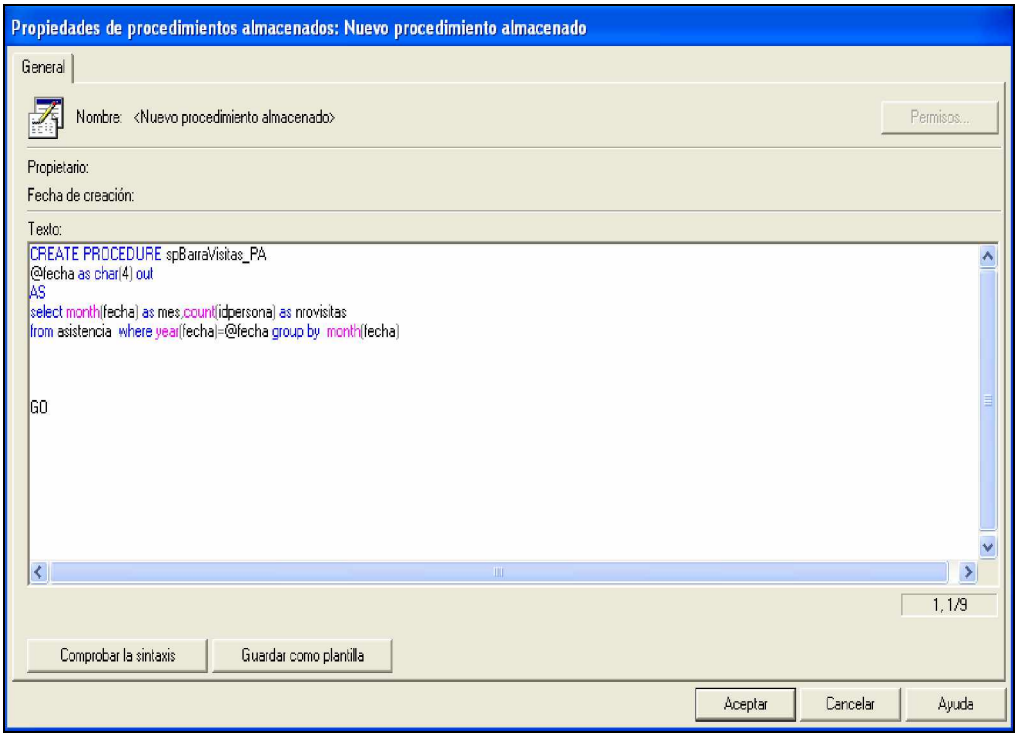
La clase cReporteVisitas contiene un método que ejecuta una consulta SQL a la tabla *asistencia* indicando el número del mes y la cantidad de visitantes que hubo para que con esos datos pueda generar una imagen gracias la librería jfreechart, en éste caso generaremos un diagrama de barras pero esta potente librería permite generar todo tipo de gráficos.



Vamos a crear el procedimiento almacenado que extraerá los datos necesarios para generar el gráfico, para ello entramos al SQL Server 2000 y escogemos nuestra Base de Datos *reporte* y luego hacemos anticlik en *Procedimientos Almacenados* y escogemos *Nuevo Procedimiento Almacenado:*

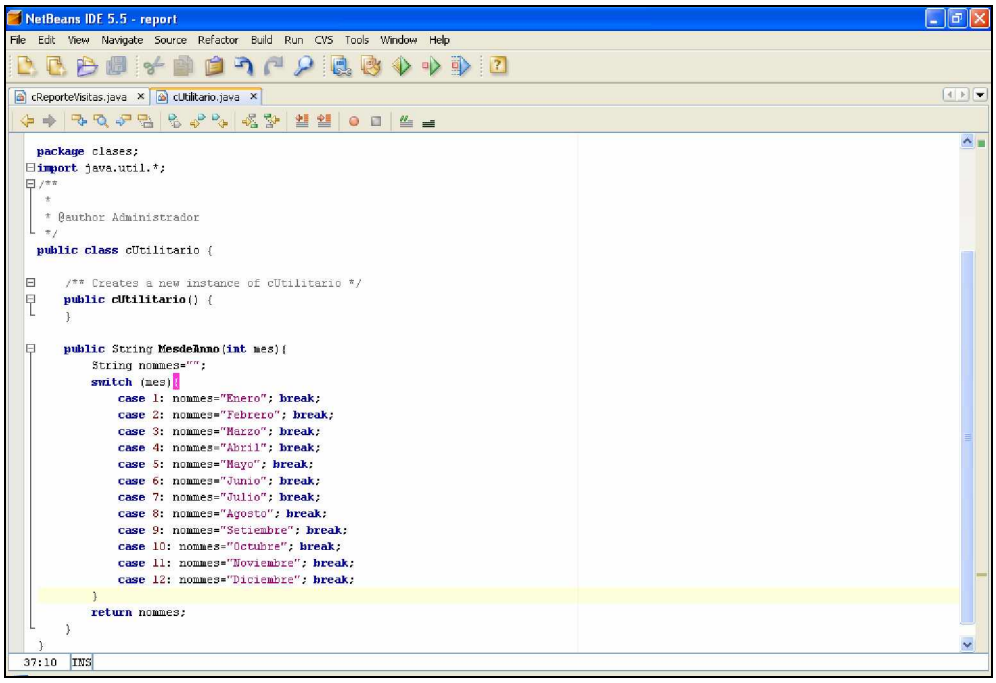


En la siguiente pantalla creamos nuestro Procedimiento y le damos *Aceptar:*

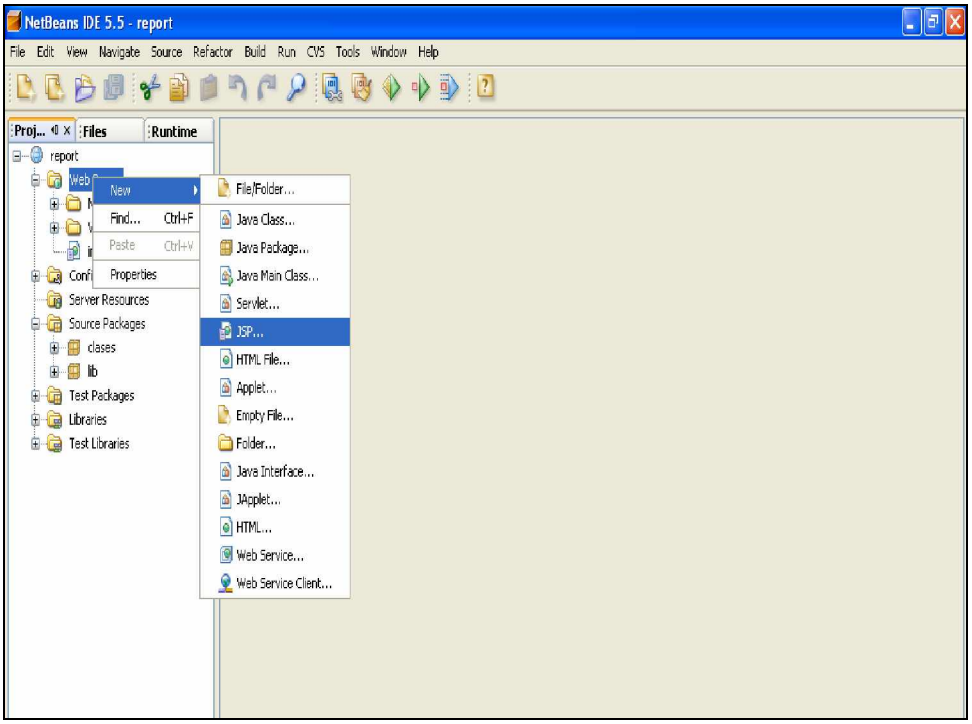


Listo!!!, Nuestro Procedimiento almacenado esta creado.

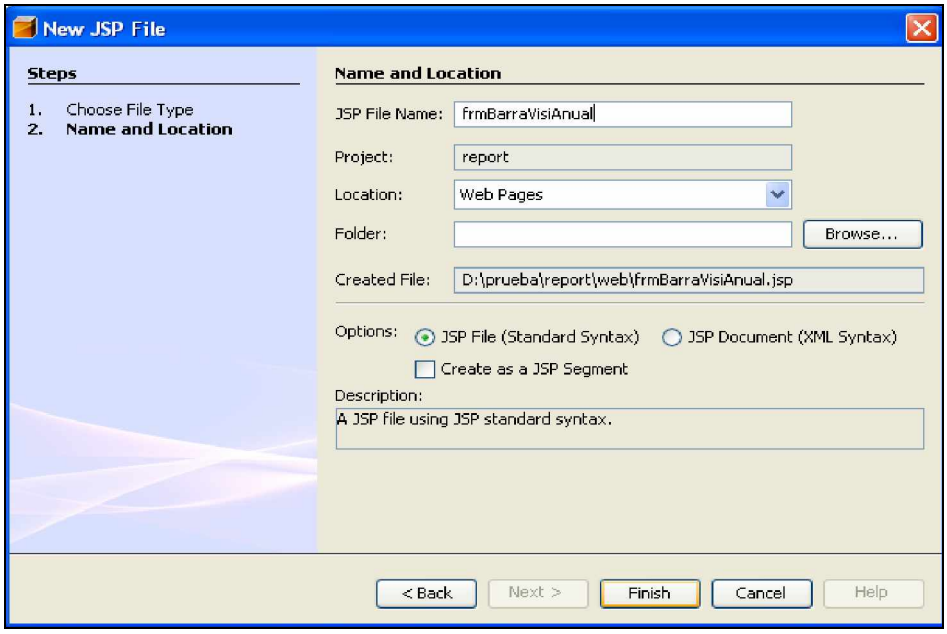
Por otro lado la clase cUtilitario solo contiene un método que devuelve el nombre del mes del año según el número que le enviemos como parámetro:



Ahora vamos a crear nuestro jsp que enviará los parámetros de la consulta que queremos realizar, para ello hacemos vamos a nuestro proyecto y hacemos anticlik en Web Pages escogemos *New* y luego *JSP:*



Le ponemos nombre al JSP y le damos *Finish*:



Nuestro frmBarraVisiAnual.jsp será simple y lo único que tendrá es un combo que muestre dinámicamente los años en los que queremos extraer el reporte, en nuestro caso empezará en 2007 y terminará en el año actual aquí el código de esa rutina:

```
<select name="cboanno" >
    <option value="">Seleccione--Año</option>
    <% for(int i=2007;i<=Integer.parseInt(a);i++){%>
    <option value="<%=i%>"><%=i%></option>
    <%}%>
</select>
```

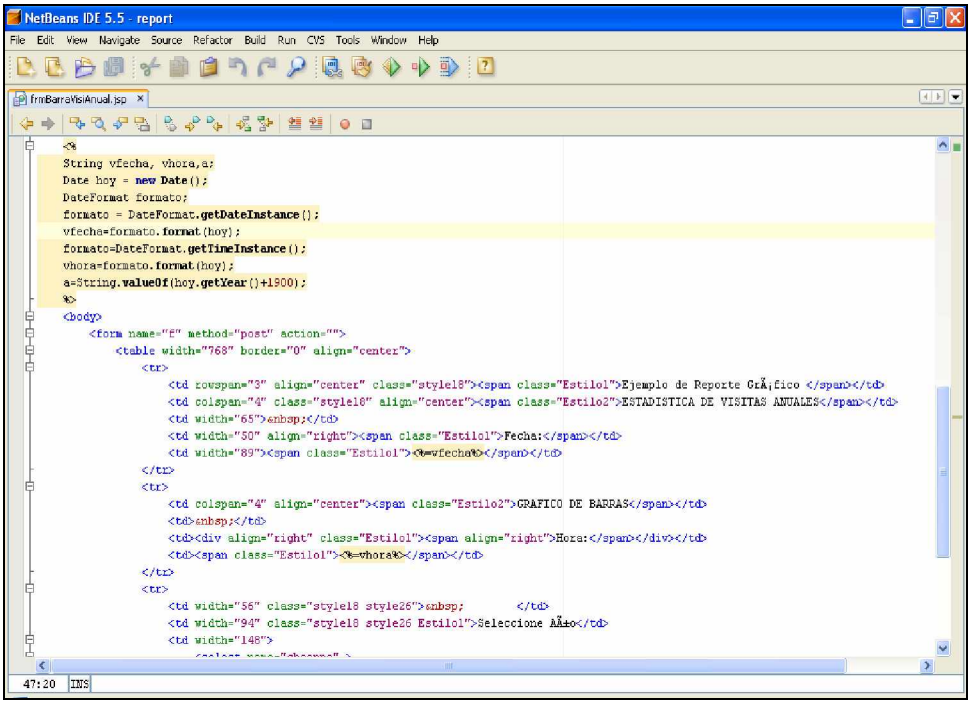
Donde “a” es el valor del año actual extraído previamente en el jsp, si queremos que genere desde el año 2000 simplemente cambiamos el valor inicial de “i” y generará los años 2000,2001,2002,2003,2004,2005,2006 y 2007

Para extraer la fecha y la hora actual del sistema usamos:

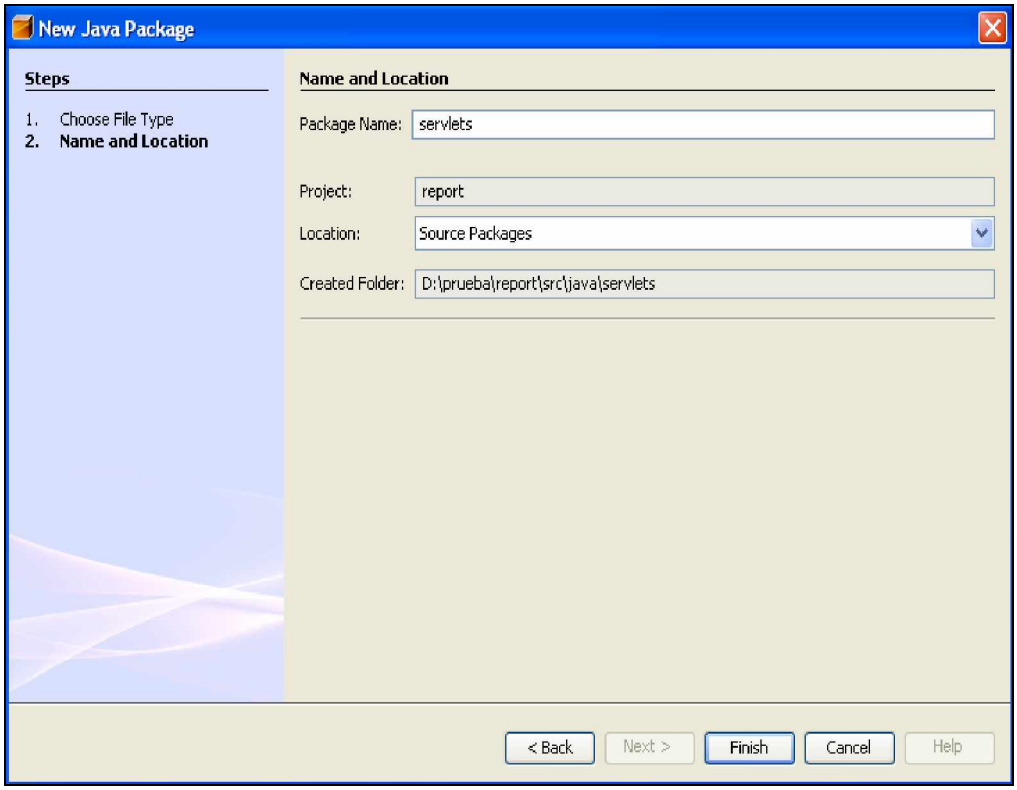
```
<%    String vfecha, vhora,a;
      Date hoy = new Date();
      DateFormat formato;
      formato = DateFormat.getDateInstance();
      vfecha=formato.format(hoy);
      formato=DateFormat.getTimeInstance();
      vhora=formato.format(hoy);
      a=String.valueOf(hoy.getYear()+1900);    %>
```

Donde “vfecha” contiene la fecha actual, “vhora” contiene la hora actual y “a” es el año en el que nos encontramos

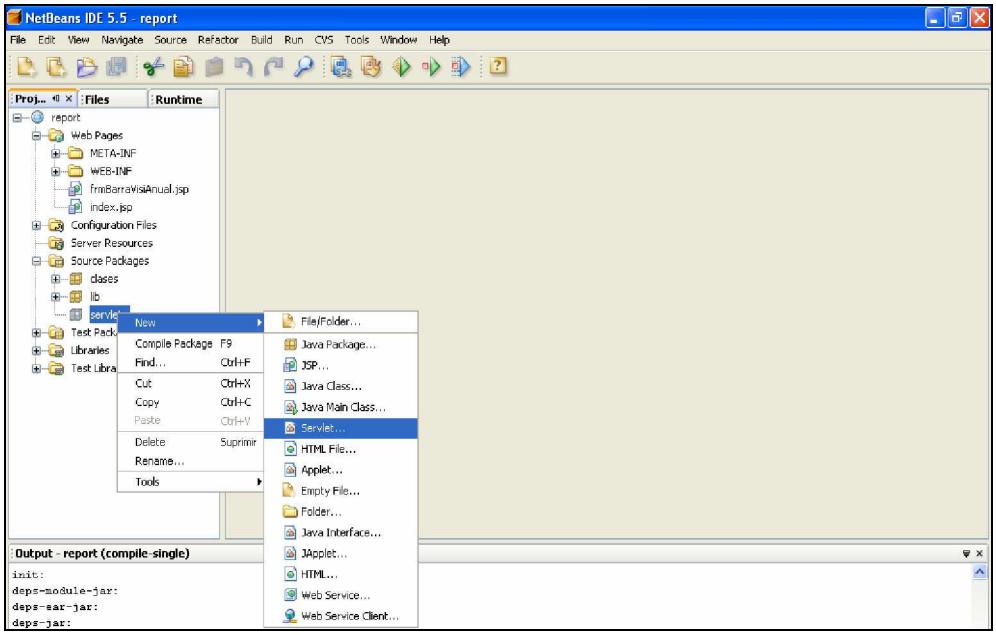
Aquí tenemos el código completo:



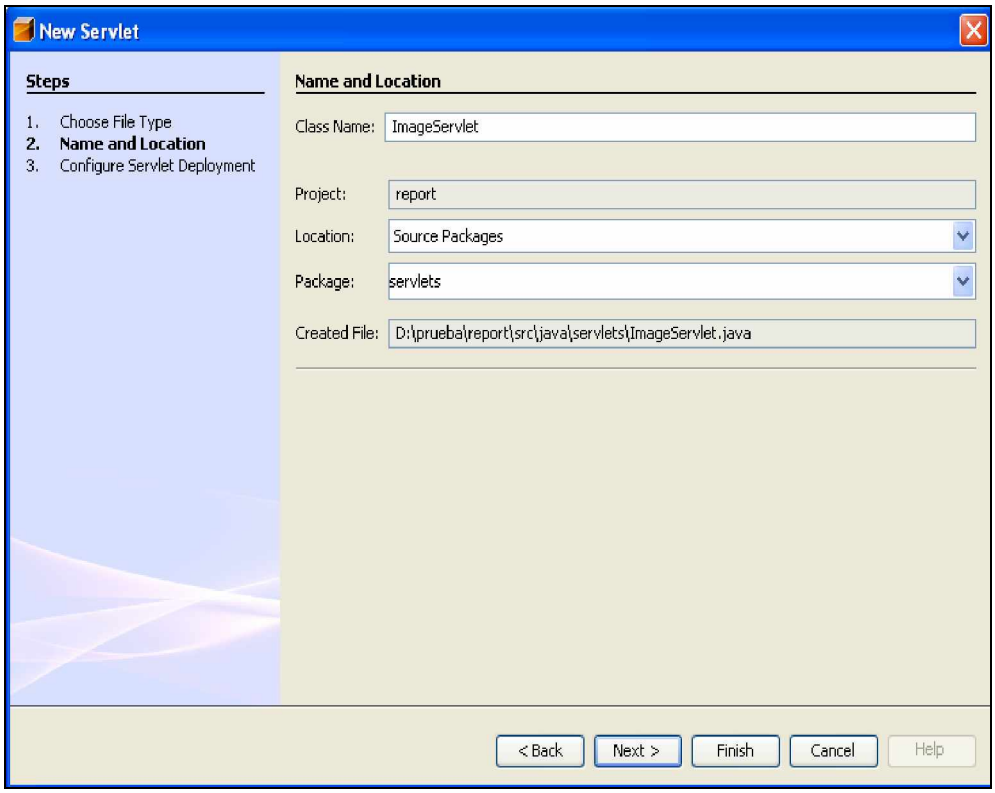
Nuestro jsp enviará los datos a un servlet, el cual disparará el reporte en forma de imagen, asi que vamos a crearlo. Para ello tenemos que crear un paquete en nuestro proyecto llamado *servlets*



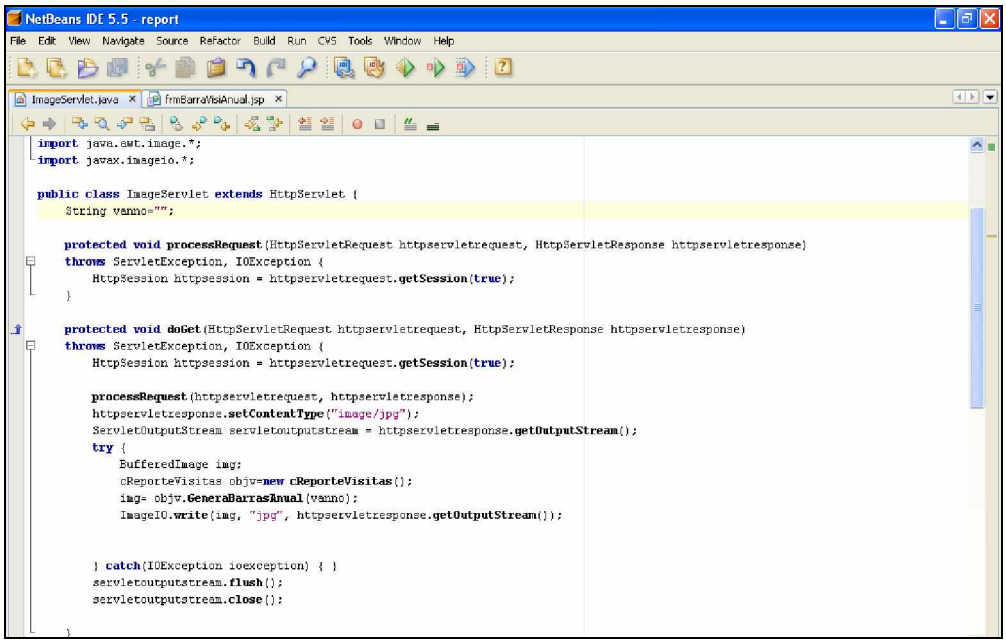
Y agregamos un nuevo servlet llamado ImageServlet:

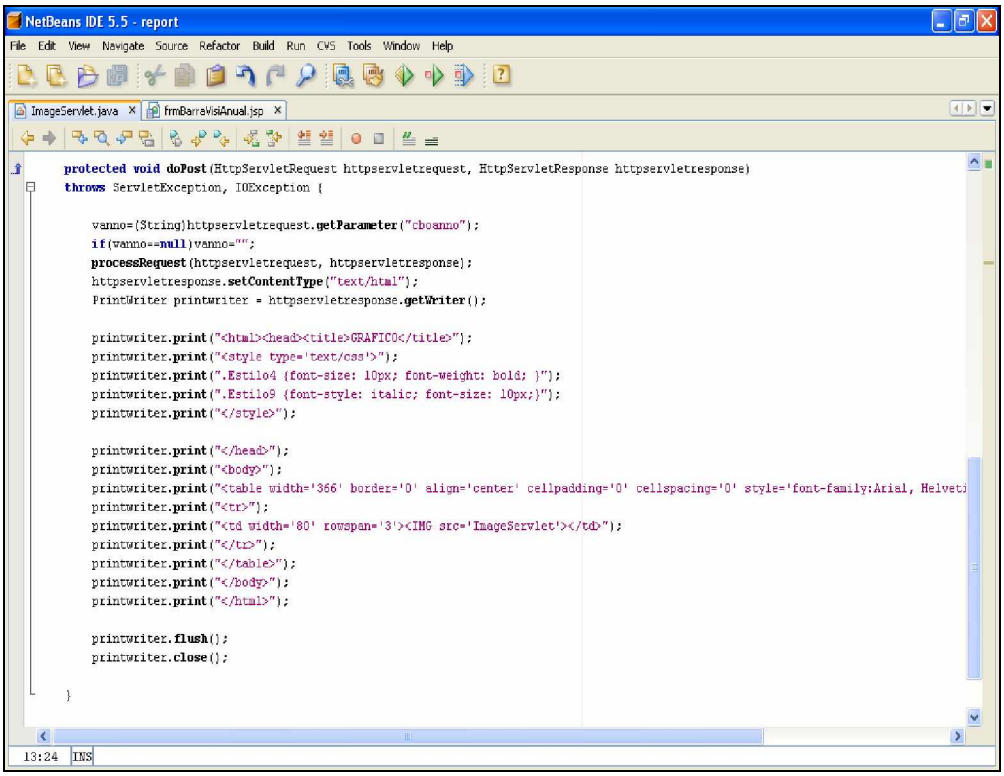


Le ponemos el nombre y le damos *Finish*

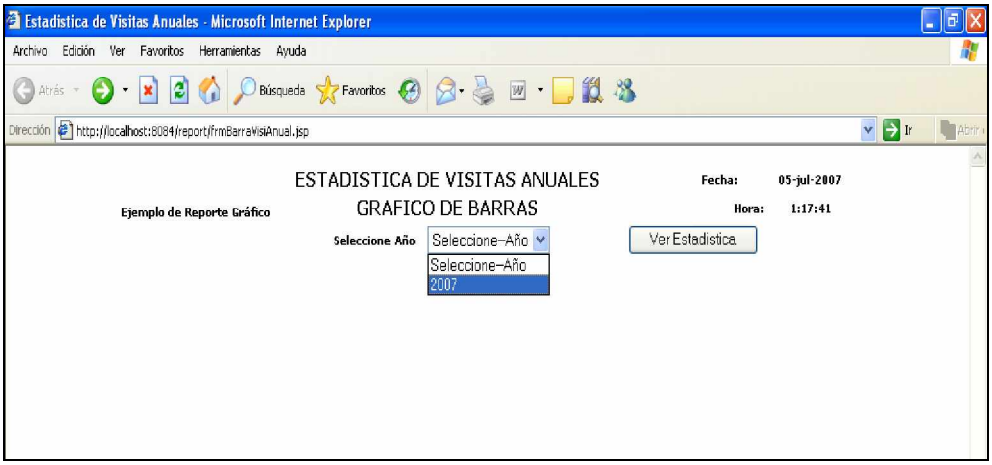


Lo que hace el servlet es instanciar un objeto de la clase cReporteVisitas e invocar a su método GenerarBarrasAnual(var) donde “var” es el año que le enviamos desde el jsp para generar el gráfico. Finalmente nos devuelve un HTML con la imagen incrustada:

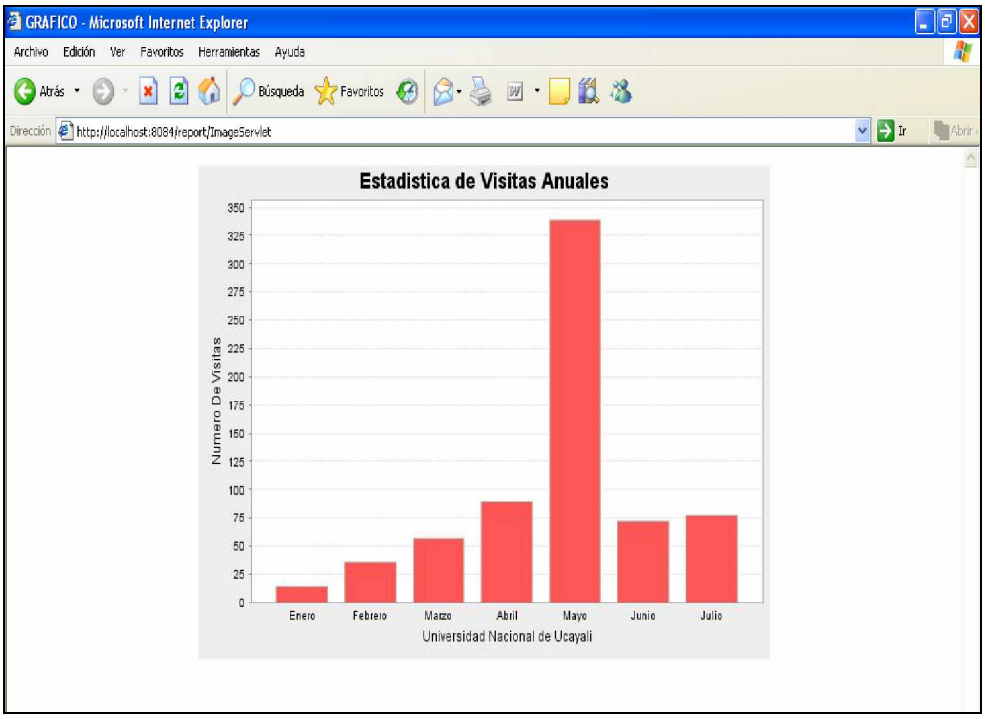




Solo nos queda ejecutar nuestro JSP y verificar su funcionamiento:



Y el resultado es el siguiente:



III.- Conclusiones

- Ø Los reporte gráficos se pueden crear de manera sencilla y rápida utilizando componentes reutilizables.

- Ø La versatilidad de ésta librería nos permite generar muchos tipos de gráficos, según la necesidad del desarrollador.

- Ø Siempre debemos separar el Modelo (que en este caso son nuestras clases), la Vista (los JSP's) y el controlador (los servlet's) para respetar el patrón de desarrollo MVC.

- Ø Los reportes gráficos permiten una interpretación mas ágil que usando reporte en modo texto.

---

[1] **Acerca del Autor**

- **Bachiller en Ingeniería de Sistemas de la Universidad Nacional de Ucayali.**
- **Analista-Programador de Aplicaciones Web bajo la plataforma J2EE.**
- **Actualmente labora en la Oficina de Desarrollo de Software de la Universidad Nacional de Ucayali.**