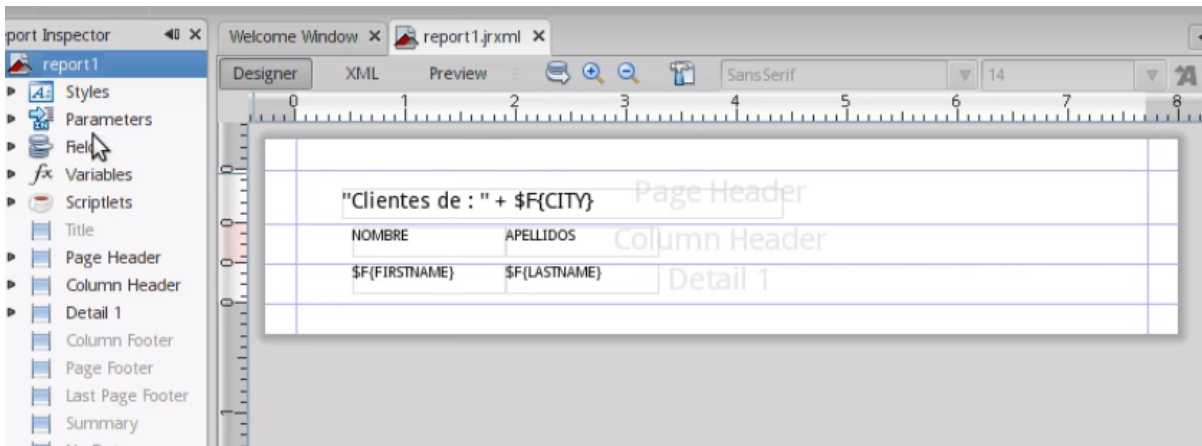
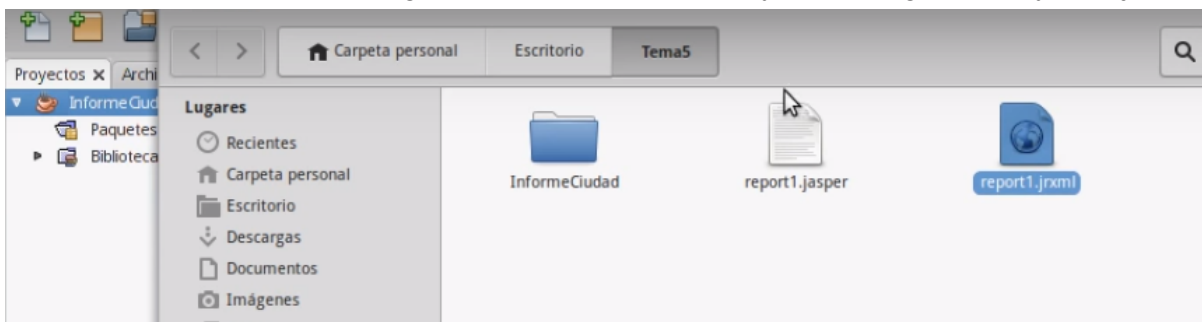


EJEMPLO PARA INTEGRAR UN INFORME EN APP JAVA DAD

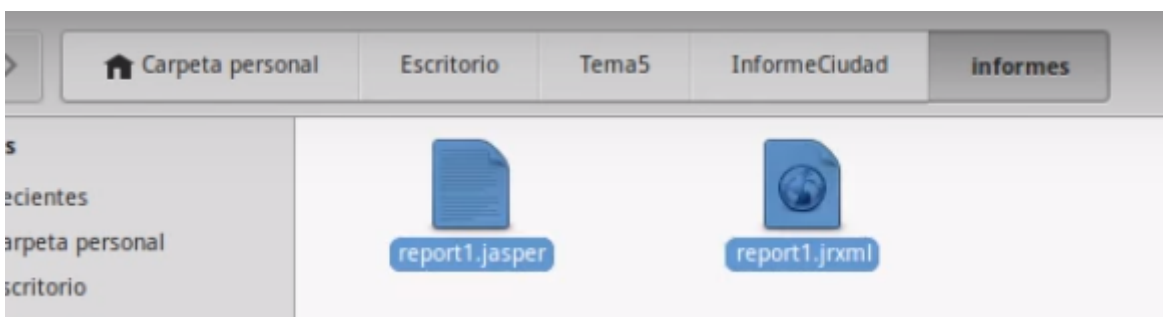
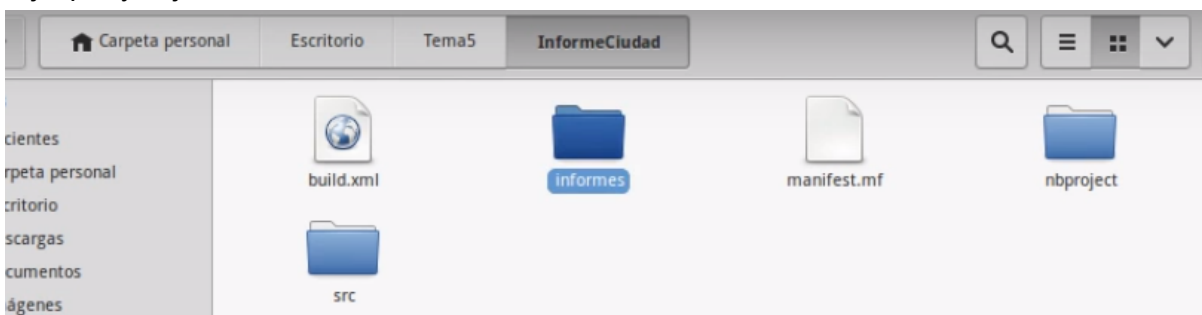
Vamos a partir de un informe sencillo, a integrarlo en un app en java, por ejemplo que tenga un combo y que dependiendo del valor del combo que seleccione, lo pase como parámetro y genere el informe.



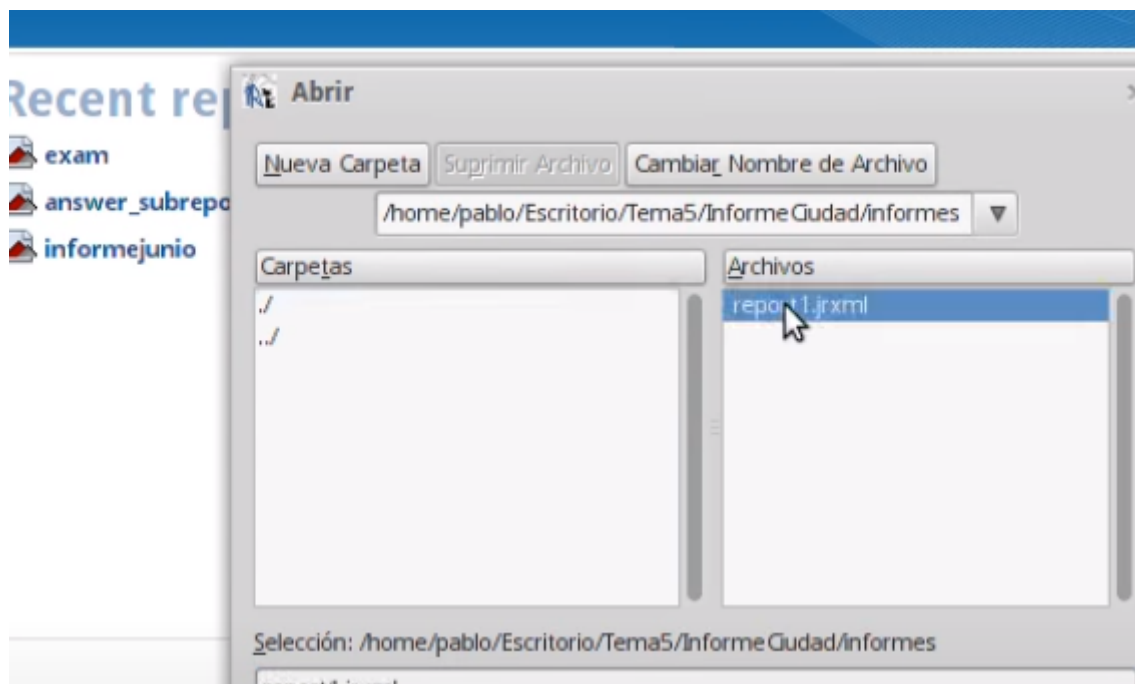
Creamos una app en java con un nuevo proyecto, por ejemplo con el nombre InformeCiudad, pero primero vamos a conocer un poco la organización del informe que ya ha sido generado y a mejorarla



El informe compilado es el jasper, al que tenemos que acceder, para tener todo dentro de nuestro proyecto te propongo que creemos una carpeta que se llama informes dentro de nuestro proyecto y ahí moveremos el jasper y el jrxml.

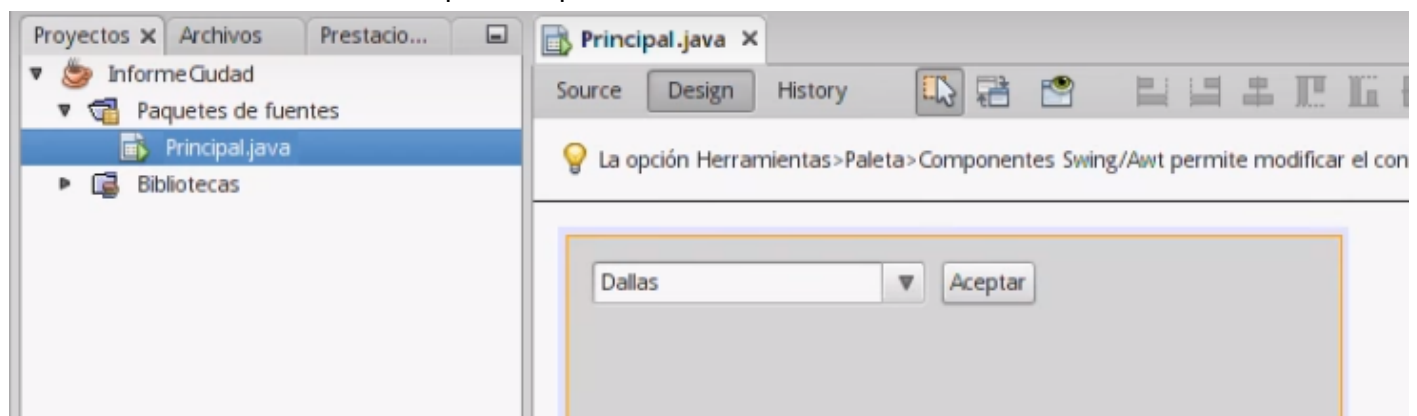


Vamos a ireport y cerraremos el jrxml y lo abriremos desde su nueva ruta.



En nuestro proyecto vamos a crear un nuevo JFrame, por ejemplo con dos items que sabemos que existen (en este caso dos ciudades), solo para hacer la prueba y ver que funciona, lo ideal sería leer la base de datos y rellenar ese combo con las ciudades existentes, pero por ahora lo que nos interesa es generar el informe con el parámetro elegido.

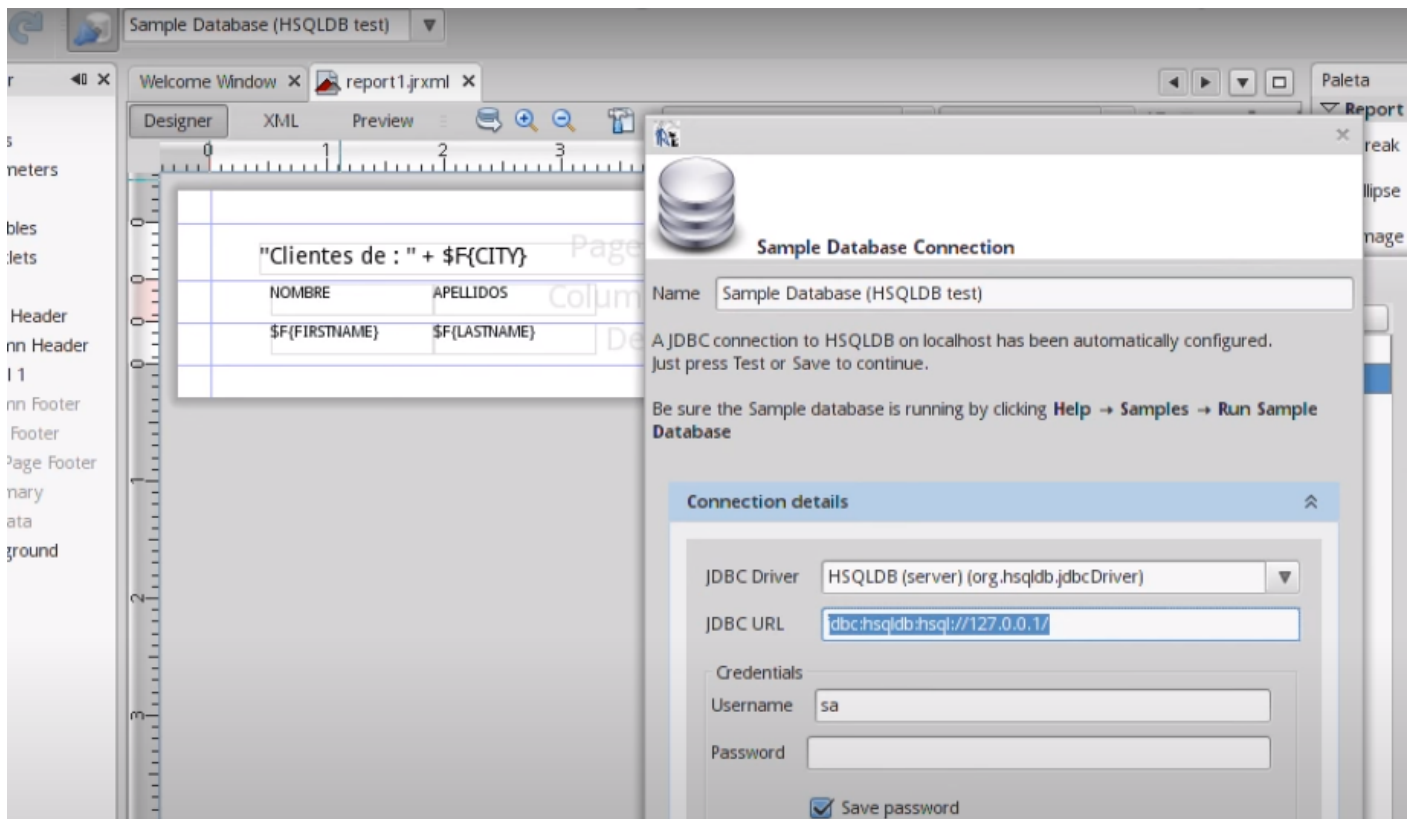
Vamos además a meter un botón para aceptar.



Es necesario usar los try y catch porque llamaremos a un código externo, es necesario realizar la conexión de la base de datos, primero cargaremos el driver del manejador de la base de datos, en este ejemplo que podrás comprobar, se está usando otro manejador y es el que utilizan, pero en tu caso deberás poner el que estamos utilizando. La opción `e.printStackTrace()` es para imprimir los errores en consola para ir controlando nuestro código.

```
private void jButtonAceptarActionPerformed(java.awt.event.ActionEvent evt) {
    try
    {
        Class.forName("org.hsqldb.jdbcDriver").newInstance();
    }
    catch (Throwable e)
    {
        e.printStackTrace();
    }
}
```

Para la siguiente línea de código que vamos a generar que es la conexión, vamos a consultar la url que estamos utilizando en la ventana de conexión y la agregaremos en nuestro código



```
try
{
    Class.forName("org.hsqldb.jdbcDriver").newInstance();
    Connection connection = DriverManager.getConnection("jdbc:hsqldb:hsqldb://127.0.0.1/");
}
catch (Throwable e)
```

No olvides agregar el pass y contraseña si nos conectamos al ireport con eso

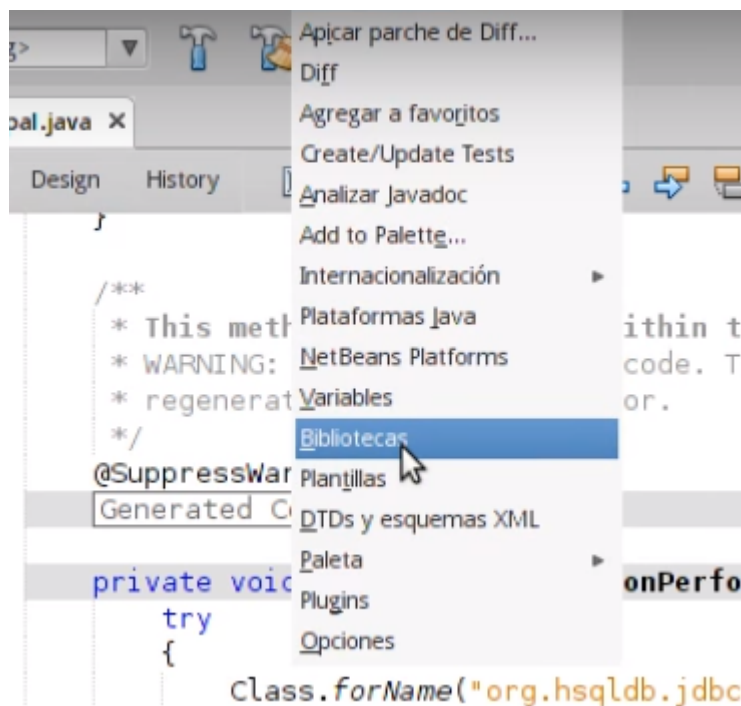
```
Class.forName("org.hsqldb.jdbcDriver").newInstance();
Connection connection = DriverManager.getConnection("jdbc:hsqldb:hsqldb://127.0.0.1/", "sa", "");
```

Ahora vamos a generar el informe pasando los parámetros con la clase map, poniendo bastante atención que el parámetro coincida mayúsculas y minúsculas con el que estamos usando en ireport

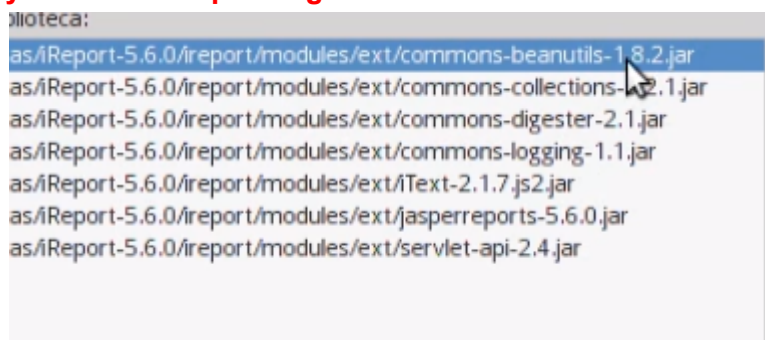
```
try
{
    Class.forName("org.hsqldb.jdbcDriver").newInstance();
    Connection connection = DriverManager.getConnection("jdbc:hsqldb:hsqldb://127.0.0.1/", "sa", "");

    Map parametros = new HashMap();
    parametros.put("CIUDAD", jComboBoxCiudad.getSelectedItem());
}
catch (Throwable e)
```

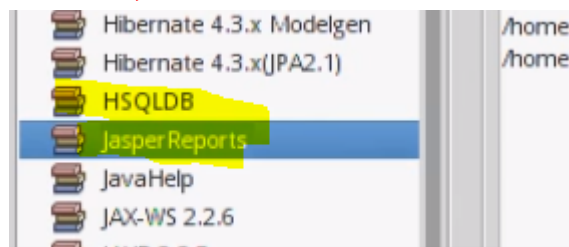
Ahora las líneas que siguen son correspondientes a la llamada a las bibliotecas, que si no hacemos esta verificación no nos reconocerá las librerías. Así que vamos a bibliotecas



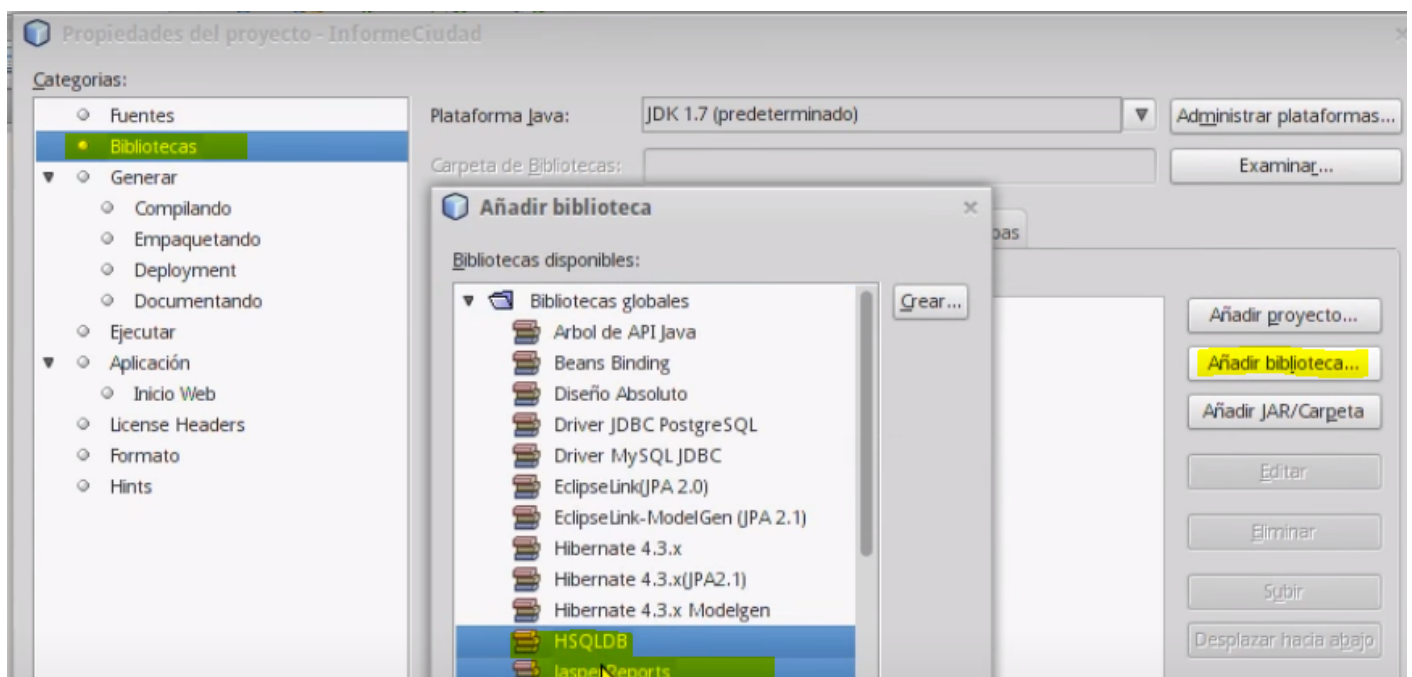
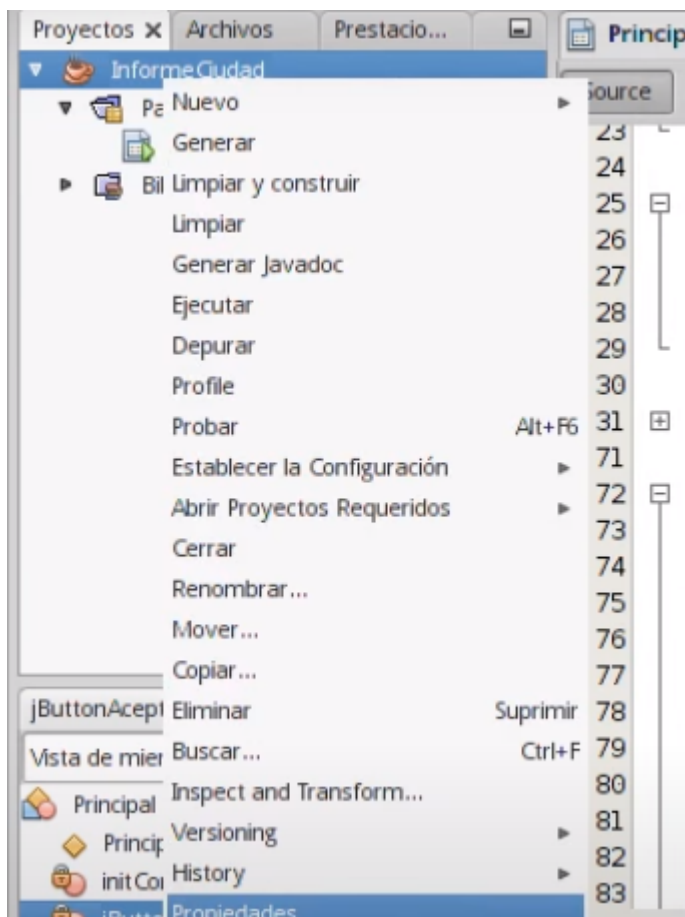
y verificamos que tengamos las librerías



y también verificamos en la librería que tenemos el jar del driver correspondiente de nuestra conexión, si no tienes nada de esto crea las librerías e introdúcelas ahí.



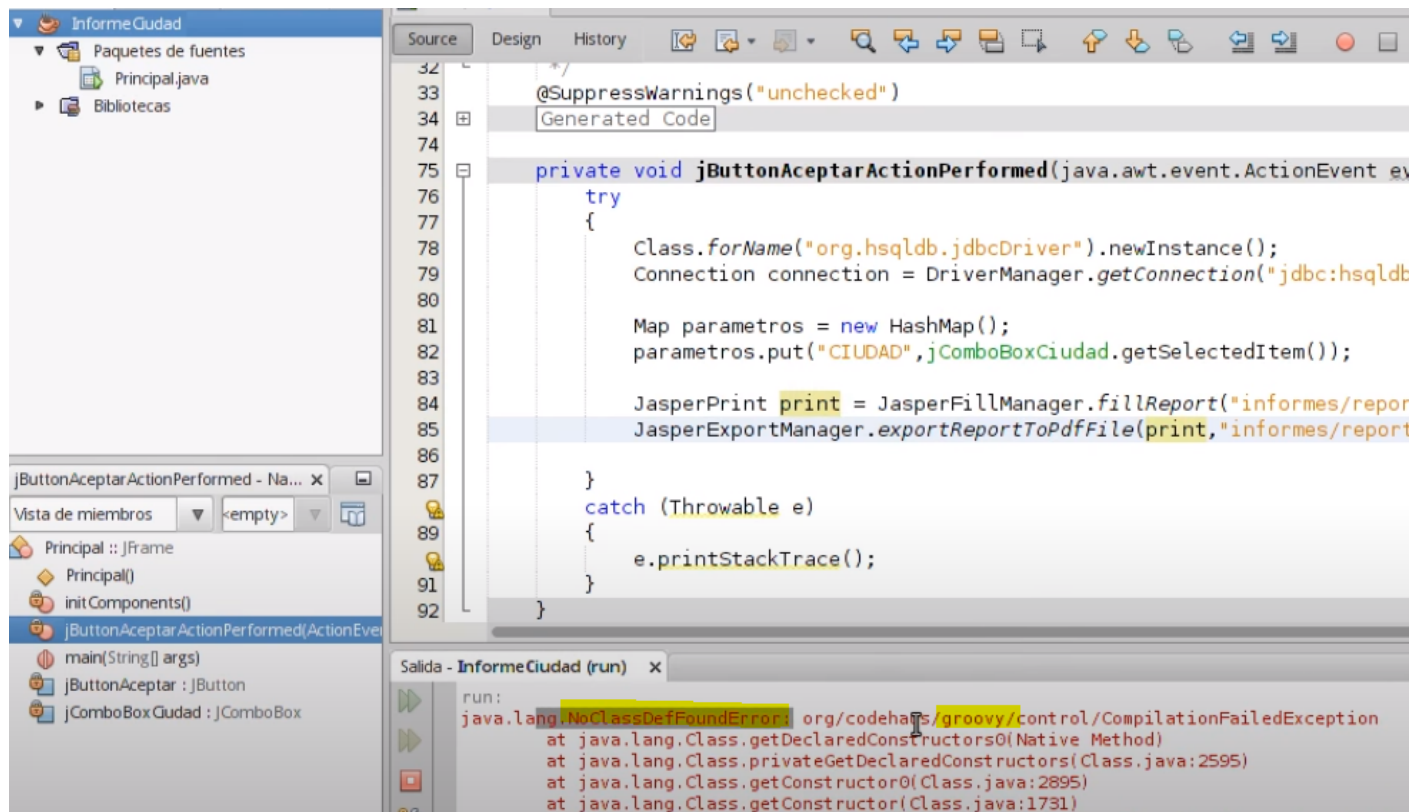
Vamos ahora a añadir estas librerías a nuestro proyecto en el menú propiedades



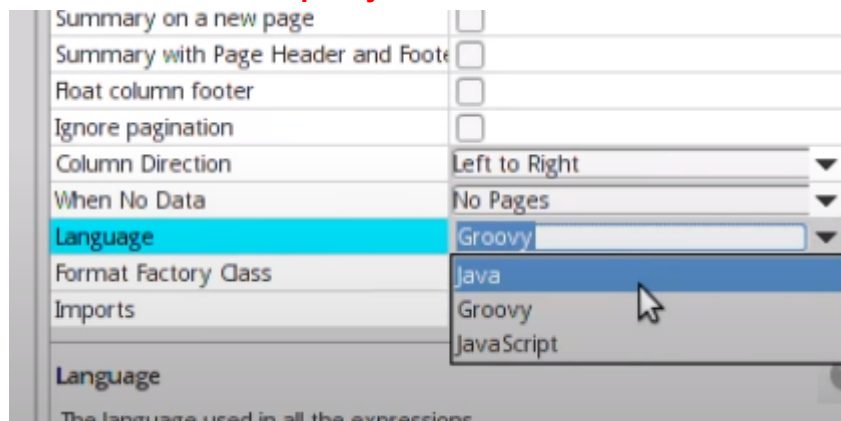
En el ejemplo agregaremos dos nuevas líneas, la correspondiente a la impresión del informe y la generación de un documento pdf (si quieres elige mejor la salida por pantalla para que veas el resultado inmediatamente)

```
JasperPrint print = JasperFillManager.fillReport("informes/report1.jasper", parametros,connection);
JasperExportManager.exportReportToPdfFile(print,"informes/report1.pdf");
```

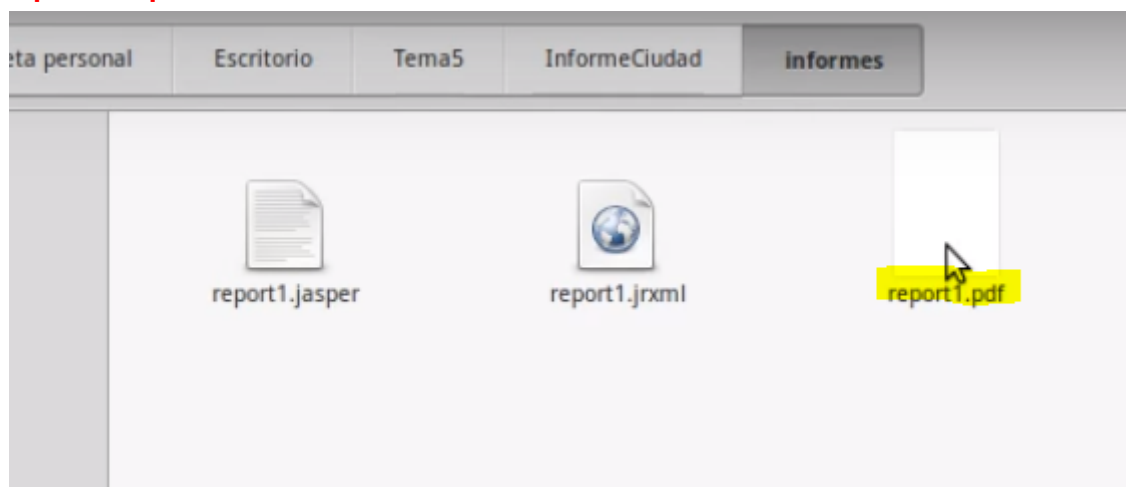
Probablemente veremos el famoso error y ahora lo resolveremos, vamos a poner atención al tema del lenguaje groovy



Para ello iremos a ireport y localizaremos en las herramientas el campo referente al lenguaje groovy



Cambiamos a Java para dejarlo con posibilidad más nativa recompilamos el reporte. Al ejecutarlo si lo dejaste con la clase de generación pdf tendrás que ir a la carpeta y verificar que ahí está el reporte impreso.



En este caso debería de funcionar correctamente.