



GUSTAVO CORONEL
DESARROLLA SOFTWARE



ENTERPRISE JAVA DEVELOPER DESARROLLO WEB CON SPRING BOOT

REPORTES CON JASPERREPORT

Eric Gustavo Coronel Castillo

www.desarrollasoftware.com

gcoronelc@gmail.com

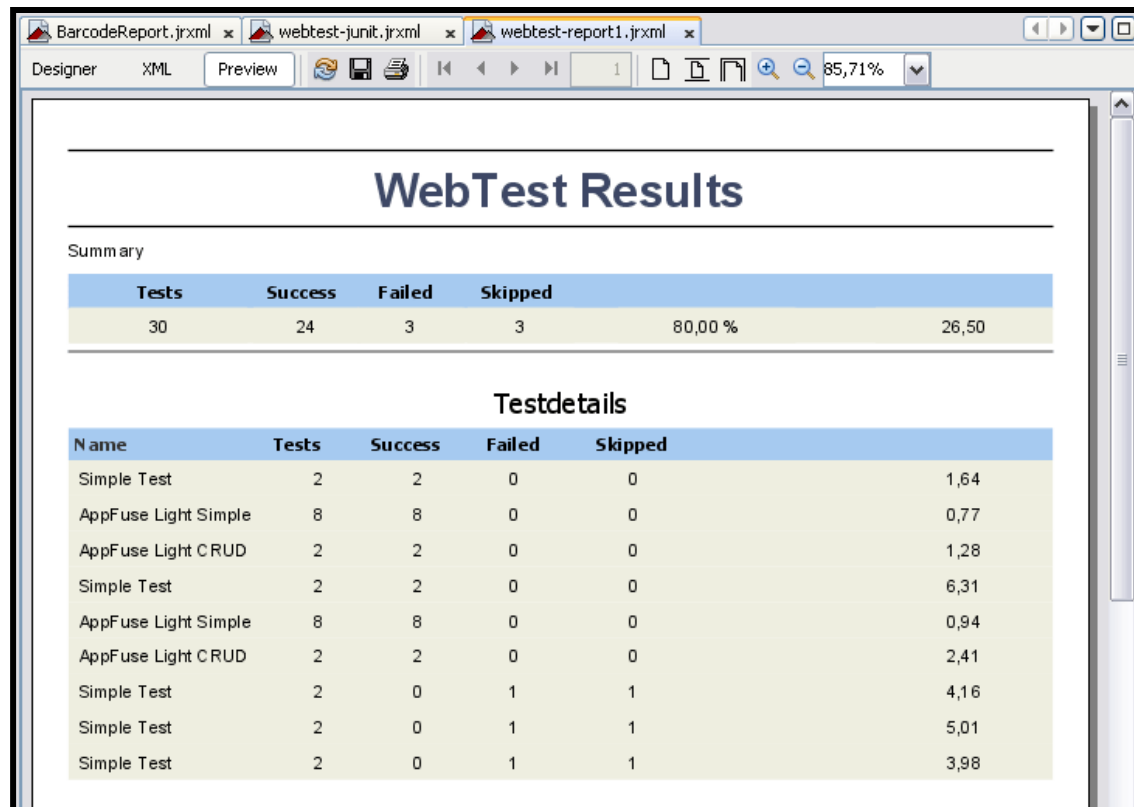
LOGRO ESPERADO

Al finalizar esta lección el participante debe manejar reportes con JasperReport.



Objetivo

Obtener reportes impresos.



The screenshot shows a web browser window with three tabs: 'BarcodeReport.jrxml', 'webtest-junit.jrxml', and 'webtest-report1.jrxml'. The active tab displays a 'WebTest Results' report. The report has a 'Summary' section with a table showing overall statistics and a 'Testdetails' section with a table showing individual test results.

WebTest Results

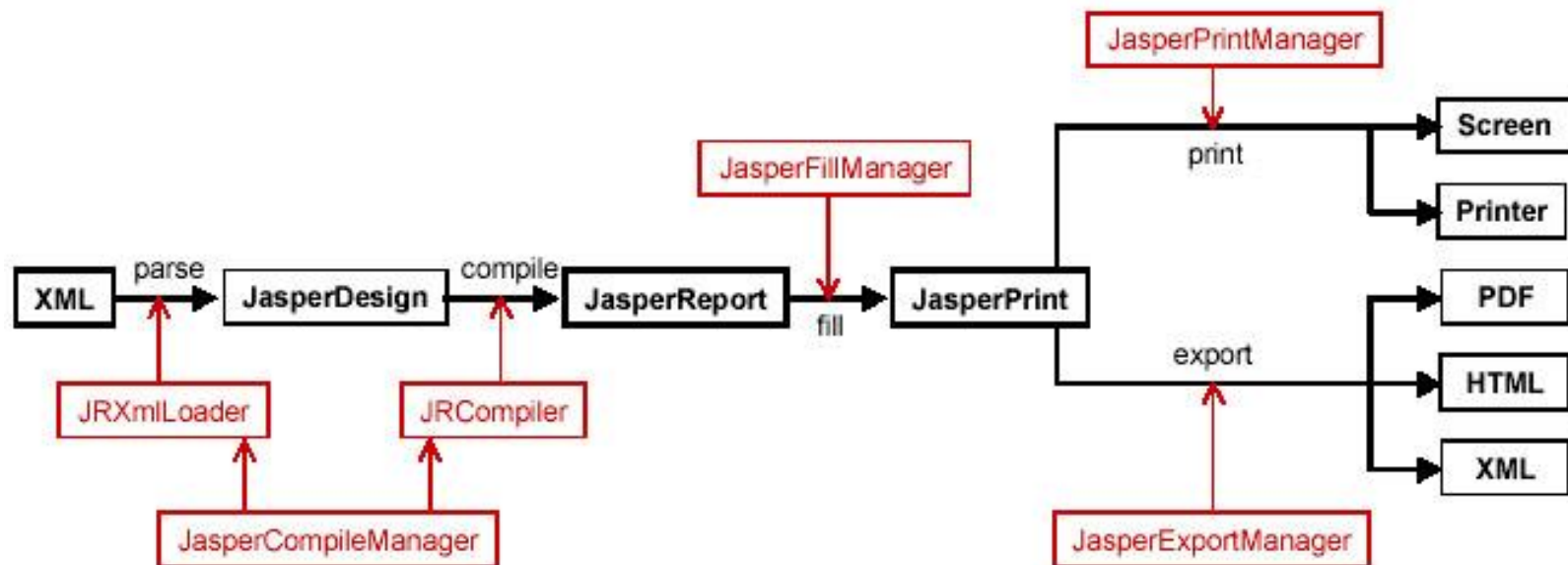
Summary

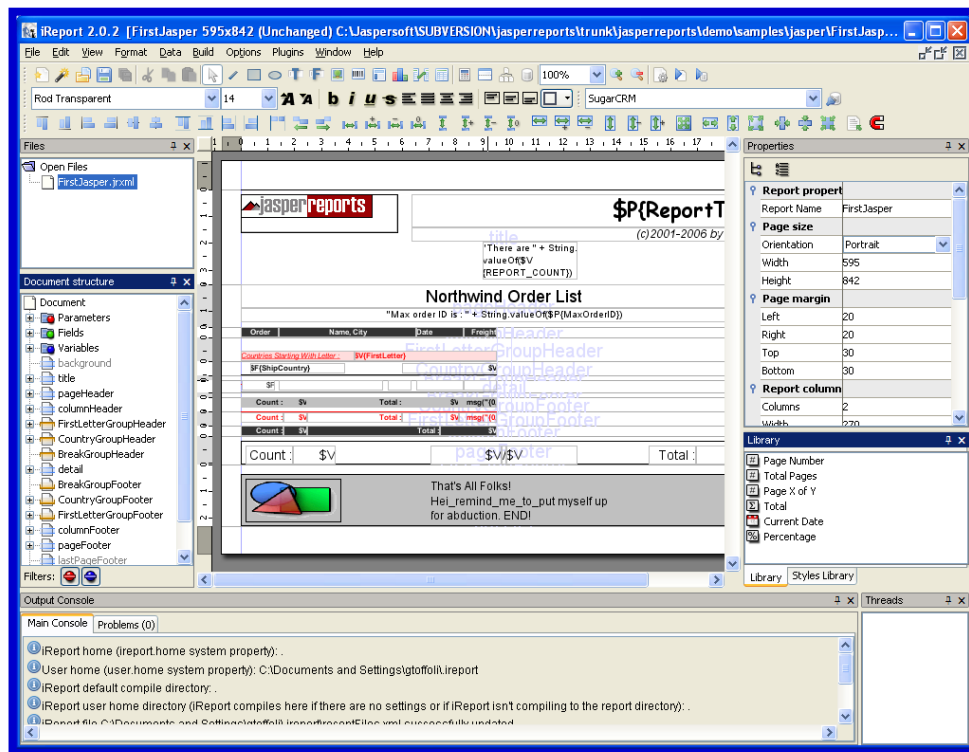
Tests	Success	Failed	Skipped		
30	24	3	3	80,00 %	26,50

Testdetails

Name	Tests	Success	Failed	Skipped	
Simple Test	2	2	0	0	1,64
AppFuse Light Simple	8	8	0	0	0,77
AppFuse Light CRUD	2	2	0	0	1,28
Simple Test	2	2	0	0	6,31
AppFuse Light Simple	8	8	0	0	0,94
AppFuse Light CRUD	2	2	0	0	2,41
Simple Test	2	0	1	1	4,16
Simple Test	2	0	1	1	5,01
Simple Test	2	0	1	1	3,98

- Es una librería Open Source en Java para generar informes desde **plantillas XML** y diferentes fuentes de datos.
- La salida puede ser mostrada en una ventana, impresa o escrita en archivos XML, PDF, archivos de Excel y HTML.
- Pasos para trabajar con **JasperReport**
 - Crear el archivo de diseño del reporte (*.jrxml)
 - Cargar el archivo de diseño del reporte (*.jrxml)
 - Compilar el archivo de diseño del reporte (*.jasper)
 - Llenar los datos al reporte.
 - Enviar el reporte a (Pantalla, PDF, XLS, etc)
- **Home Page:** <http://www.jasperforge.org>





Herramienta gráfica para diseñar informes, usa la librería **JasperReport** y genera los archivos *.jrxml (XML) y *.jasper (COMPILADO)

Home Page: <http://www.jasperforge.org/>

JAR dependientes

- El JAR JasperReport depende de otras librerías para realizar diversas tareas por ejemplo.
 - SAX 2.0 XML Parser
 - Jakarta Commons Digester Component
 - Jakarta Commons BeanUtils Component
 - Jakarta Commons Collections Component
 - Jakarta Commons Logging Component
 - iText
 - JFreeChart
 - Jakarta POI (v1.5.1+)
 - etc.

JRXML

```
<?xml version="1.0" encoding="UTF-8"?>
<jasperReport name="report1" pageWidth="595" pageHeight="842" columnWidth="535"
leftMargin="20" rightMargin="20" topMargin="20" bottomMargin="20">
  <queryString>
    <![CDATA[select * from cliente]]>
  </queryString>
  <field name="nombre" class="java.lang.String">
    <fieldDescription><![CDATA[]]></fieldDescription>
  </field>
  <detail>
    <textFieldExpression class="java.lang.String">
      <![CDATA[${nombre}]]>
    </textFieldExpression>
  </detail>
</jasperReport>
```


Compilando y Visualizando el JRXML









```
try{
    // Base de datos
    Class.forName("com.mysql.jdbc.Driver");
    Connection cn = DriverManager.getConnection("jdbc:mysql://localhost/db","root", "admin");
    // Parámetros
    Map pars = new HashMap();
    // Usando el archivo JRXML
    JasperReport rep=JasperCompileManager.compileReport("rep.jrxml");
    JasperPrint print = JasperFillManager.fillReport(rep, pars, cn);
    // Mostrando el reporte
    JasperViewer viewer = new JasperViewer(print,false);
    viewer.setTitle("Mi Reporte");
    viewer.setVisible(true);
}catch (Exception ex){
    JOptionPane.showMessageDialog(null, ex.getMessage());
}
```

Trabajando con BEANS

```
try{
    // Datos
    Collection lista = objDAO.TraerListaBeans();
    JRBeanCollectionDataSource data;
    data = new JRBeanCollectionDataSource(lista);
    // Parámetros
    Map pars = new HashMap();
    pars.put("USUARIO", "Gustavo Coronel");
    pars.put("LOGO", "logo.gif");
    //Usando el archivo JASPER (compilado)
    JasperReport rep=(JasperReport)JRLoader.loadObject("rep.jasper");
    JasperPrint print = JasperFillManager.fillReport(rep, pars, data);
    //Creando y Visualizando PDF
    JasperExportManager.exportReportToPdf(print);
    JasperViewer.viewReport(print,false);
}catch (Exception ex){
    JOptionPane.showMessageDialog(null, ex.getMessage());
}
```

Reportes en Web

```
try {
    IArticuloDAO articuloDAO = DAOFactory.getDAOFactory().getArticuloDAO();
    ArrayList<ArticuloTO> lista = articuloDAO.consultarPorNombre("%");
    JRBeanCollectionDataSource data;
    data = new JRBeanCollectionDataSource(lista);
    ServletConfig config = getServletConfig();
    ServletContext sc = config.getServletContext();
    InputStream inputStream = sc.getResourceAsStream("/reportes/repoArticulos.jasper");
    if (inputStream == null) { throw new Exception("Reporte no existe.<br>"); }
    JasperReport jasperReport = (JasperReport) JRLoader.loadObject(inputStream);
    byte[] bytes = JasperRunManager.runReportToPdf(jasperReport, null, data);
    response.setContentType("application/pdf");
    response.setContentLength(bytes.length);
    ServletOutputStream out = response.getOutputStream();
    out.write(bytes, 0, bytes.length);
    out.flush();
    out.close();
} catch (Exception e) {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("Error: " + e.getMessage());
    out.close();
}
```

- ▼  WebContent
 - >  js
 - >  META-INF
 - ▼  reportes
 -  repoArticulos.jasper
 -  repoArticulos.jrxml
 -  repoMovimientos.jasper
 -  repoMovimientos.jrxml

Reportes en Web

```
try {
    String cuenta = request.getParameter("cuenta");
    EurekaService service = new EurekaService();
    List<Map<String, ?>> lista = service.getMovimientos(cuenta);
    JRMapCollectionDataSource data = new JRMapCollectionDataSource(lista);
    Map pars = new HashMap<>();
    String reporte = "/pe/eureka/reportesx/repoMovimientos.jasper";
    InputStream is = getClass().getResourceAsStream(reporte);
    if(is == null){ throw new Exception("Reporte no existe."); }
    JasperReport rep = (JasperReport) JRLoader.loadObject(is);
    byte[] bytes = JasperRunManager.runReportToPdf(rep, null, data);
    response.setContentType("application/pdf");
    response.setContentLengthLong(bytes.length);
    ServletOutputStream out = response.getOutputStream();
    out.write(bytes, 0, bytes.length);
    out.flush();
    out.close();
} catch (Exception e) {
    . . .
}
```

- ▼ 📁 SRC
 - > 📁 pe.eureka.controller
 - > 📁 pe.eureka.db
 - > 📁 pe.eureka.dto
 - > 📁 pe.eureka.prueba
 - ▼ 📁 pe.eureka.reportes
 - 📄 repoMovimientos.jasper
 - 📄 repoMovimientos.jrxml
 - > 📁 pe.eureka.service

Gracias