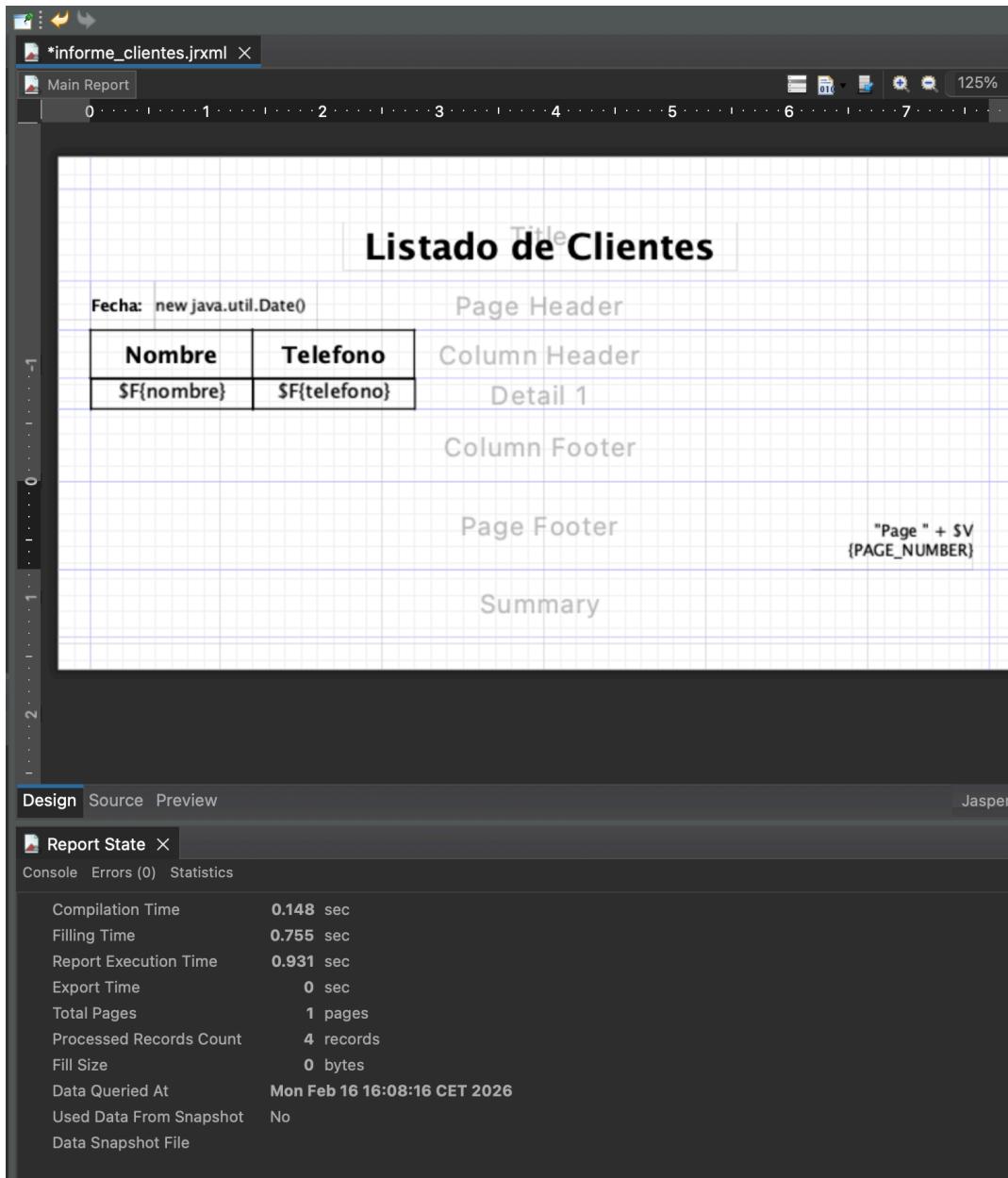


Entregable 5: Jaspersoft Informes

Ejercicio 1:

Plantilla informe JRXML:



Compilar JRXML y rellenar:

```

private final String path = "src/main/resources/informe_clientes.jrxml";

public JasperReport readAndCompileFile() {
    JasperReport compiledReport = null;

    try {
        compiledReport = JasperCompileManager.compileReport(path);
        System.out.println("Archivo compilado con éxito mi rey");

    } catch (JRException e) {
        System.out.println("Error al compilar el archivo JRXML: " + e.getMessage());
        e.printStackTrace();
    }
    return compiledReport;
}

public JasperPrint fillFile(JasperReport jasperReport) {
    Map<String, Object> map = new HashMap<>();
    JasperPrint jasperPrint = null;
    try {
        jasperPrint = JasperFillManager.fillReport(jasperReport, map, ConnectionJDBC.connection());
        System.out.println("Archivo llenado con éxito");
    } catch (JRException e) {
        System.out.println("Error al llenar el archivo JRXML: " + e.getMessage());
        e.printStackTrace();
    }
    return jasperPrint;
}

```

Exportar a PDF:

```

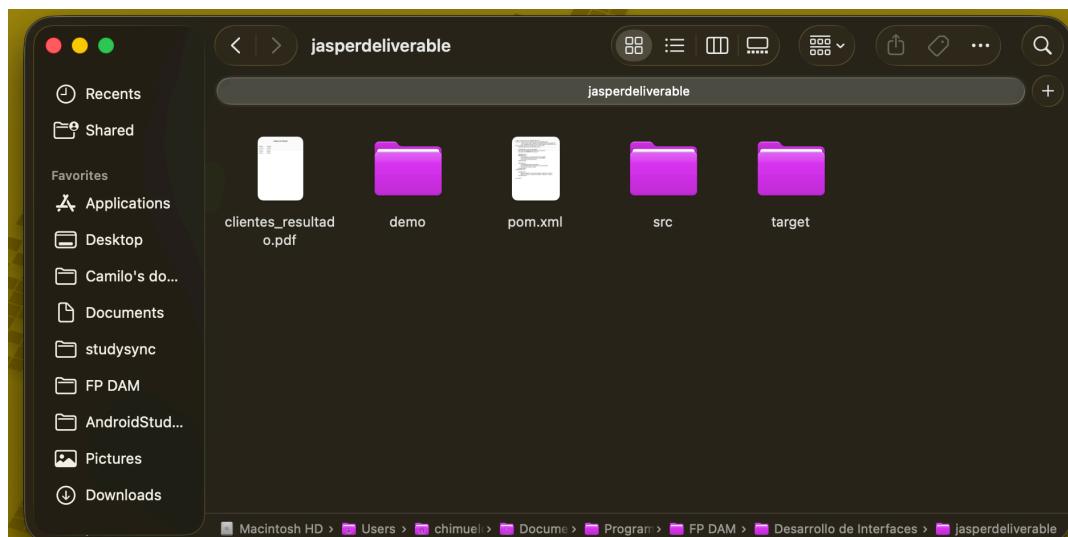
public void exportToPdf(JasperPrint jasperPrint) {

    String rutaSalida = "clientes_resultado.pdf";

    try {
        JasperExportManager.exportReportToPdfFile(jasperPrint, rutaSalida);
        System.out.println("PDF generado correctamente en: " + rutaSalida + "!");
    } catch (JRException e) {
        System.err.println("Error al exportar a PDF: " + e.getMessage());
        e.printStackTrace();
    }
}

```

Como no se especifica una ruta de salida, el programa lo guarda en la misma carpeta del proyecto:



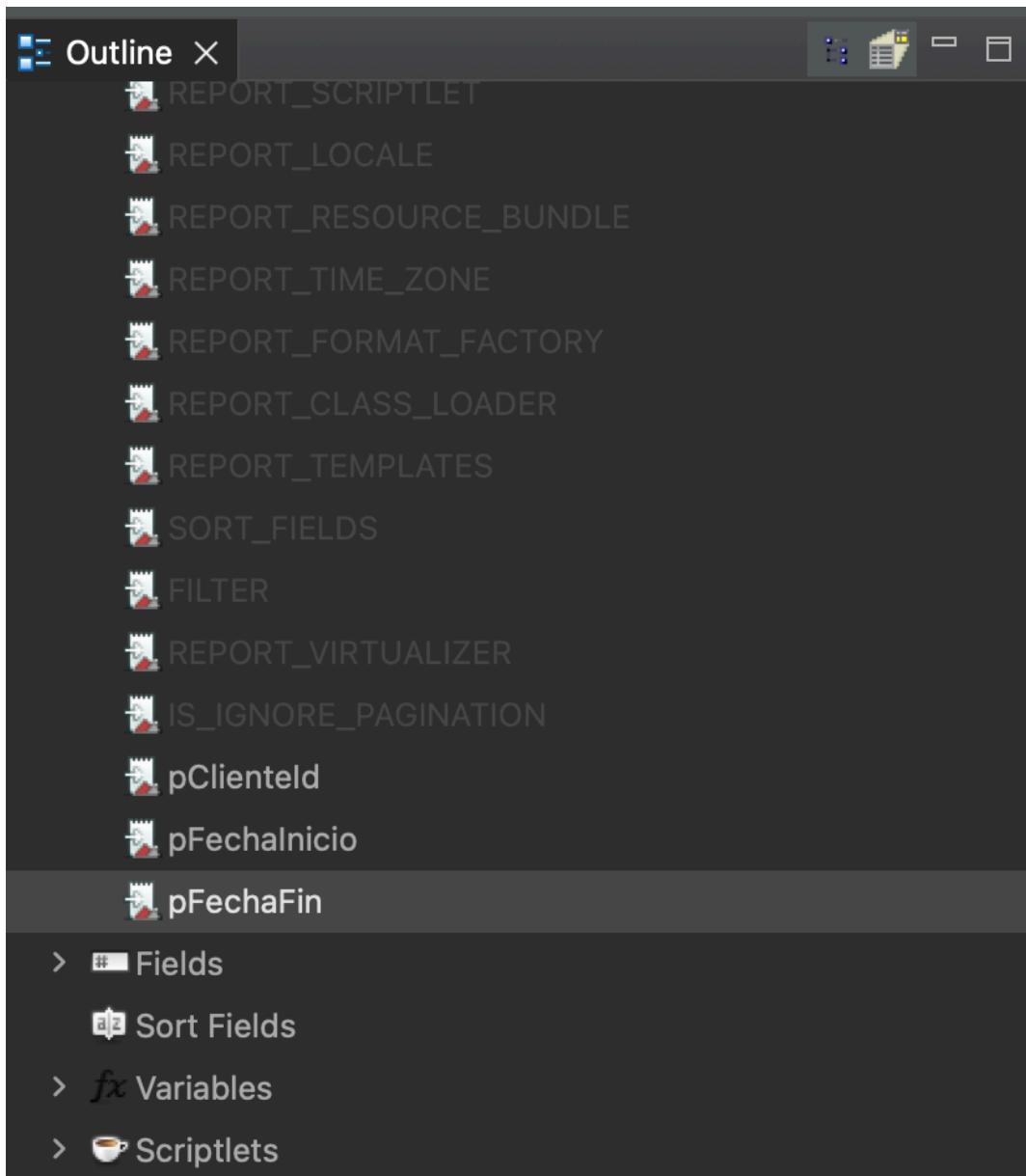
Nombre	Telefono
Juan Camilo	123456789
Ana Lopez	666111222
Luis Garcia	666333444
Marta Sanz	666555666

Ejercicio 2:

Definir parámetros:

Definí los 3 parametros dichos en el enunciado, estos son los que se usan en ReportService en el metodo

`fillFile()` a través de un `Map<>`



Creación de Variables:

Se crearon las variables, les agregué una función de `"Sum"` y `"Count"` para evitar hacer esto en código en Java



Creación del Map<String, Object>

Lo que se pasó como "Object" fue la fecha de inicio y fin, esto se conectó con las variables creadas en JasperSoft

```

public static void main(String[] args) {
    ReportService reportService = new ReportService();

    JasperReport report = reportService.readAndCompileFile();

    if (report != null) {
        for (int i = 1; i <= 3; i++) {
            Map<String, Object> parametros = new HashMap<>();

            if (i == 1) {

                parametros.put("pClienteId", 1);
                parametros.put("pFechaInicio", Date.valueOf("2000-01-01"));
                parametros.put("pFechaFin", Date.valueOf("2099-12-31"));
            } else if (i == 2) {

                parametros.put("pClienteId", 3);
                parametros.put("pFechaInicio", Date.valueOf("2000-01-01"));
                parametros.put("pFechaFin", Date.valueOf("2026-12-31"));
            } else {

                parametros.put("pClienteId", 4);
                parametros.put("pFechaInicio", Date.valueOf("2000-01-01"));
                parametros.put("pFechaFin", Date.valueOf("2026-12-31"));
            }

            // Rellenar y exportar con un nombre diferente cada vez
            JasperPrint impresion = reportService.fillFile(report, parametros);

            if (impresion != null) {
                reportService.exportToPdf(impresion, "pedidos_reporte_" + i +
                ".pdf");
            }
        }
    }
}

```

```
    }  
}
```

Ejecución del programa 3 veces:

Para lograr esto, se creó el bucle for mostrado arriba, donde en cada iteración, crea un nuevo Map con filtros diferentes para cada cliente, y va siguiendo uno a uno el flujo de trabajo:

1. Compilar
2. Rellenar
3. Exportar a PDF