

UNIDAD DIDÁCTICA 5

CONFECCIÓN DE INFORMES

**MÓDULO PROFESIONAL:
DESARROLLO DE INTERFACES**



CESUR
Tu Centro Oficial de FP

Índice

RESUMEN INTRODUCTORIO	2
INTRODUCCIÓN	3
CASO INTRODUCTORIO	3
1. INFORMES	4
1.1 Informes incrustados y no incrustados en la aplicación.....	5
1.2 Herramientas gráficas integradas en el IDE y externas al mismo	6
2. JASPERSOFT	10
2.1 Instalación	11
2.2 Creando un nuevo informe. Conexión con las fuentes de datos	13
2.3 Estructura general. Secciones	20
2.4 Ejecución de consultas	23
2.5 Encabezados y pies. Numeración de líneas, recuentos y totales	26
2.6 Valores calculados	31
2.7 Filtrado de datos	37
2.8 Informes con agrupamiento, recuentos parciales y subtotales	41
2.9 Sub-Informes	43
2.10 Formatos de salida	45
2.11 Imágenes. Gráficos	47
2.12 Parámetros	53
2.13 Librerías para generación de informes. Clases, métodos y atributos	59
RESUMEN FINAL	63

RESUMEN INTRODUCTORIO

En esta unidad introduciremos uno de los apartados más importantes cuando trabajamos con información en una empresa como es el de los informes y análisis de información.

Comenzaremos realizando una definición y un análisis de las diferentes herramientas que disponemos en el mercado, encontrando herramientas y suites muy interesantes gratuitas que, además, se pueden integrar perfectamente dentro de espacios de programación.

Analizaremos las herramientas gráficas disponibles para la generación de informes, tanto las integradas en los entornos de desarrollo (IDE) como las externas al mismo, evaluando sus funcionalidades y cómo se adaptan a necesidades de desarrollo. Una de esas herramientas es JasperSoft, una suite muy madura, realizada con Java y, por lo tanto, que se integra perfectamente con nuestro entorno.

Estudiaremos la estructura general de un informe en JasperSoft, familiarizándonos con sus secciones y comprendiendo cómo organizar la información. Veremos cómo se ejecutan consultas para extraer datos y veremos cómo incorporar encabezados y pies de página, incluyendo numeración de líneas, recuentos y totales.

Exploraremos el uso de valores calculados y técnicas de filtrado de datos para generar informes más personalizados. Aprenderemos a crear informes con agrupamientos, recuentos parciales y subtotales, así como a integrar subinformes para manejar estructuras de datos más complejas.

Además, veremos cómo enriquecer nuestros informes con imágenes y gráficos, mejorando la visualización y comprensión de la información presentada.

En último lugar, realizaremos un análisis de las librerías y clases que nos proporciona JasperSoft para poder integrar incluso dentro de nuestros proyectos realizados con Java.

INTRODUCCIÓN

El tratamiento de la información, la gestión de los datos y la creación de informes para su posterior análisis se ha convertido en uno de los aspectos más importantes para cualquier cargo directivo dentro de una empresa.

Los informes son elementos que permiten dar sentido a la información almacenada en las bases de datos generados a partir de las múltiples aplicaciones de gestión que pueda tener una empresa.

Para poder realizar y gestionar dichos informes, existen en el mercado múltiples opciones, pero todas ellas tienen en común muchos aspectos que hacen que una vez estudiada en profundidad una, utilizar cualquier otra sea muy sencillo.

CASO INTRODUCTORIO

Acabas de ser contratado por una franquicia de tiendas de ventas de ropa. Esta franquicia tiene, desde hace ya tiempo, diversas aplicaciones tanto en tienda como en la central para la generación de información y datos, por lo que ahora mismo los diferentes directores de área se plantean incorporar herramientas para el análisis de la información y generación de informes.

Al finalizar esta unidad tendremos los conocimientos para poder desarrollar informes con una de las herramientas propuestas.

1. INFORMES

El primer paso consiste en realizar un análisis de las herramientas disponibles en el sector para poder seleccionar la que más se acople con las necesidades de negocio y técnicas de la empresa.

Durante el proceso de investigación y búsqueda se concluye que existen herramientas que puedes integrar dentro de tu entorno de desarrollo como Jaspersoft Studio o Eclipse Birt. A partir de esta selección, podrás establecer los primeros criterios de búsqueda y prueba.

Los informes son la guinda del pastel real cuando se trata de obtener información útil de una base de datos. Proporcionan información en un formato que cualquier persona puede entender y, como la mayoría de los informes están diseñados para impresión, dan a sus datos un elemento de portabilidad.

La construcción de informes es, sin duda, un proyecto más complejo que la creación de consultas o formularios y, por lo tanto, es una tarea que requiere un poco de planificación.

Los informes suelen incluir números de página, fechas y tiempos en que se imprimieron, agrupaciones de registros y sumarios de datos, entre otra información.

The screenshot shows a SAP Crystal Reports interface displaying a pre-printed form titled "REPORT OF NEW EMPLOYEE(S)". The form is organized into sections for employer information, employee details, and contact information. The employer information includes the name "RIDIN' ROUND BIKES, LLC" and address "40 ROCKY RIVER RD, LOS ANGELES, CA 90037". The employee details section lists three employees: LAURA CALLAHAN, LAURENT PEREIRA, and TIM SMITH, each with their respective social security numbers, addresses, and employment dates. The form is displayed within a software window with a menu bar (File, Edit, View, Insert, Format, Database, Report, Window, Help) and a toolbar. A report explorer on the right side of the window shows the structure of the report, including sections like "Pre-Printed Form", "Page Header", "Page Header a", "Page Header b", "Text1: RIDIN' ROUND BIKES, LLC", "Text2: 40 ROCKY RIVER RD", "Text3: LOS ANGELES", "Text4: 90037", "Text5: 310-555-1212", "Text6: 90037", "Text7: 310-555-1212", "Text8: 90037", "Text9: 310-555-1212", "Text10: 90037", "Text11: 310-555-1212", "Text12: 90037", "Text13: 310-555-1212", "Text14: 90037", "Text15: 310-555-1212", "Text16: 90037", "Text17: 310-555-1212", "Text18: 90037", "Text19: 310-555-1212", "Text20: 90037", "Text21: 310-555-1212", "Text22: 90037", "Text23: 310-555-1212", "Text24: 90037", "Text25: 310-555-1212", "Text26: 90037", "Text27: 310-555-1212", "Text28: 90037", "Text29: 310-555-1212", "Text30: 90037", "Text31: 310-555-1212", "Text32: 90037", "Text33: 310-555-1212", "Text34: 90037", "Text35: 310-555-1212", "Text36: 90037", "Text37: 310-555-1212", "Text38: 90037", "Text39: 310-555-1212", "Text40: 90037", "Text41: 310-555-1212", "Text42: 90037", "Text43: 310-555-1212", "Text44: 90037", "Text45: 310-555-1212", "Text46: 90037", "Text47: 310-555-1212", "Text48: 90037", "Text49: 310-555-1212", "Text50: 90037", "Text51: 310-555-1212", "Text52: 90037", "Text53: 310-555-1212", "Text54: 90037", "Text55: 310-555-1212", "Text56: 90037", "Text57: 310-555-1212", "Text58: 90037", "Text59: 310-555-1212", "Text60: 90037", "Text61: 310-555-1212", "Text62: 90037", "Text63: 310-555-1212", "Text64: 90037", "Text65: 310-555-1212", "Text66: 90037", "Text67: 310-555-1212", "Text68: 90037", "Text69: 310-555-1212", "Text70: 90037", "Text71: 310-555-1212", "Text72: 90037", "Text73: 310-555-1212", "Text74: 90037", "Text75: 310-555-1212", "Text76: 90037", "Text77: 310-555-1212", "Text78: 90037", "Text79: 310-555-1212", "Text80: 90037", "Text81: 310-555-1212", "Text82: 90037", "Text83: 310-555-1212", "Text84: 90037", "Text85: 310-555-1212", "Text86: 90037", "Text87: 310-555-1212", "Text88: 90037", "Text89: 310-555-1212", "Text90: 90037", "Text91: 310-555-1212", "Text92: 90037", "Text93: 310-555-1212", "Text94: 90037", "Text95: 310-555-1212", "Text96: 90037", "Text97: 310-555-1212", "Text98: 90037", "Text99: 310-555-1212", "Text100: 90037", "Text101: 310-555-1212", "Text102: 90037", "Text103: 310-555-1212", "Text104: 90037", "Text105: 310-555-1212", "Text106: 90037", "Text107: 310-555-1212", "Text108: 90037", "Text109: 310-555-1212", "Text110: 90037", "Text111: 310-555-1212", "Text112: 90037", "Text113: 310-555-1212", "Text114: 90037", "Text115: 310-555-1212", "Text116: 90037", "Text117: 310-555-1212", "Text118: 90037", "Text119: 310-555-1212", "Text120: 90037", "Text121: 310-555-1212", "Text122: 90037", "Text123: 310-555-1212", "Text124: 90037", "Text125: 310-555-1212", "Text126: 90037", "Text127: 310-555-1212", "Text128: 90037", "Text129: 310-555-1212", "Text130: 90037", "Text131: 310-555-1212", "Text132: 90037", "Text133: 310-555-1212", "Text134: 90037", "Text135: 310-555-1212", "Text136: 90037", "Text137: 310-555-1212", "Text138: 90037", "Text139: 310-555-1212", "Text140: 90037", "Text141: 310-555-1212", "Text142: 90037", "Text143: 310-555-1212", "Text144: 90037", "Text145: 310-555-1212", "Text146: 90037", "Text147: 310-555-1212", "Text148: 90037", "Text149: 310-555-1212", "Text150: 90037", "Text151: 310-555-1212", "Text152: 90037", "Text153: 310-555-1212", "Text154: 90037", "Text155: 310-555-1212", "Text156: 90037", "Text157: 310-555-1212", "Text158: 90037", "Text159: 310-555-1212", "Text160: 90037", "Text161: 310-555-1212", "Text162: 90037", "Text163: 310-555-1212", "Text164: 90037", "Text165: 310-555-1212", "Text166: 90037", "Text167: 310-555-1212", "Text168: 90037", "Text169: 310-555-1212", "Text170: 90037", "Text171: 310-555-1212", "Text172: 90037", "Text173: 310-555-1212", "Text174: 90037", "Text175: 310-555-1212", "Text176: 90037", "Text177: 310-555-1212", "Text178: 90037", "Text179: 310-555-1212", "Text180: 90037", "Text181: 310-555-1212", "Text182: 90037", "Text183: 310-555-1212", "Text184: 90037", "Text185: 310-555-1212", "Text186: 90037", "Text187: 310-555-1212", "Text188: 90037", "Text189: 310-555-1212", "Text190: 90037", "Text191: 310-555-1212", "Text192: 90037", "Text193: 310-555-1212", "Text194: 90037", "Text195: 310-555-1212", "Text196: 90037", "Text197: 310-555-1212", "Text198: 90037", "Text199: 310-555-1212", "Text200: 90037", "Text201: 310-555-1212", "Text202: 90037", "Text203: 310-555-1212", "Text204: 90037", "Text205: 310-555-1212", "Text206: 90037", "Text207: 310-555-1212", "Text208: 90037", "Text209: 310-555-1212", "Text210: 90037", "Text211: 310-555-1212", "Text212: 90037", "Text213: 310-555-1212", "Text214: 90037", "Text215: 310-555-1212", "Text216: 90037", "Text217: 310-555-1212", "Text218: 90037", "Text219: 310-555-1212", "Text220: 90037", "Text221: 310-555-1212", "Text222: 90037", "Text223: 310-555-1212", "Text224: 90037", "Text225: 310-555-1212", "Text226: 90037", "Text227: 310-555-1212", "Text228: 90037", "Text229: 310-555-1212", "Text230: 90037", "Text231: 310-555-1212", "Text232: 90037", "Text233: 310-555-1212", "Text234: 90037", "Text235: 310-555-1212", "Text236: 90037", "Text237: 310-555-1212", "Text238: 90037", "Text239: 310-555-1212", "Text240: 90037", "Text241: 310-555-1212", "Text242: 90037", "Text243: 310-555-1212", "Text244: 90037", "Text245: 310-555-1212", "Text246: 90037", "Text247: 310-555-1212", "Text248: 90037", "Text249: 310-555-1212", "Text250: 90037", "Text251: 310-555-1212", "Text252: 90037", "Text253: 310-555-1212", "Text254: 90037", "Text255: 310-555-1212", "Text256: 90037", "Text257: 310-555-1212", "Text258: 90037", "Text259: 310-555-1212", "Text260: 90037", "Text261: 310-555-1212", "Text262: 90037", "Text263: 310-555-1212", "Text264: 90037", "Text265: 310-555-1212", "Text266: 90037", "Text267: 310-555-1212", "Text268: 90037", "Text269: 310-555-1212", "Text270: 90037", "Text271: 310-555-1212", "Text272: 90037", "Text273: 310-555-1212", "Text274: 90037", "Text275: 310-555-1212", "Text276: 90037", "Text277: 310-555-1212", "Text278: 90037", "Text279: 310-555-1212", "Text280: 90037", "Text281: 310-555-1212", "Text282: 90037", "Text283: 310-555-1212", "Text284: 90037", "Text285: 310-555-1212", "Text286: 90037", "Text287: 310-555-1212", "Text288: 90037", "Text289: 310-555-1212", "Text290: 90037", "Text291: 310-555-1212", "Text292: 90037", "Text293: 310-555-1212", "Text294: 90037", "Text295: 310-555-1212", "Text296: 90037", "Text297: 310-555-1212", "Text298: 90037", "Text299: 310-555-1212", "Text300: 90037", "Text301: 310-555-1212", "Text302: 90037", "Text303: 310-555-1212", "Text304: 90037", "Text305: 310-555-1212", "Text306: 90037", "Text307: 310-555-1212", "Text308: 90037", "Text309: 310-555-1212", "Text310: 90037", "Text311: 310-555-1212", "Text312: 90037", "Text313: 310-555-1212", "Text314: 90037", "Text315: 310-555-1212", "Text316: 90037", "Text317: 310-555-1212", "Text318: 90037", "Text319: 310-555-1212", "Text320: 90037", "Text321: 310-555-1212", "Text322: 90037", "Text323: 310-555-1212", "Text324: 90037", "Text325: 310-555-1212", "Text326: 90037", "Text327: 310-555-1212", "Text328: 90037", "Text329: 310-555-1212", "Text330: 90037", "Text331: 310-555-1212", "Text332: 90037", "Text333: 310-555-1212", "Text334: 90037", "Text335: 310-555-1212", "Text336: 90037", "Text337: 310-555-1212", "Text338: 90037", "Text339: 310-555-1212", "Text340: 90037", "Text341: 310-555-1212", "Text342: 90037", "Text343: 310-555-1212", "Text344: 90037", "Text345: 310-555-1212", "Text346: 90037", "Text347: 310-555-1212", "Text348: 90037", "Text349: 310-555-1212", "Text350: 90037", "Text351: 310-555-1212", "Text352: 90037", "Text353: 310-555-1212", "Text354: 90037", "Text355: 310-555-1212", "Text356: 90037", "Text357: 310-555-1212", "Text358: 90037", "Text359: 310-555-1212", "Text360: 90037", "Text361: 310-555-1212", "Text362: 90037", "Text363: 310-555-1212", "Text364: 90037", "Text365: 310-555-1212", "Text366: 90037", "Text367: 310-555-1212", "Text368: 90037", "Text369: 310-555-1212", "Text370: 90037", "Text371: 310-555-1212", "Text372: 90037", "Text373: 310-555-1212", "Text374: 90037", "Text375: 310-555-1212", "Text376: 90037", "Text377: 310-555-1212", "Text378: 90037", "Text379: 310-555-1212", "Text380: 90037", "Text381: 310-555-1212", "Text382: 90037", "Text383: 310-555-1212", "Text384: 90037", "Text385: 310-555-1212", "Text386: 90037", "Text387: 310-555-1212", "Text388: 90037", "Text389: 310-555-1212", "Text390: 90037", "Text391: 310-555-1212", "Text392: 90037", "Text393: 310-555-1212", "Text394: 90037", "Text395: 310-555-1212", "Text396: 90037", "Text397: 310-555-1212", "Text398: 90037", "Text399: 310-555-1212", "Text400: 90037", "Text401: 310-555-1212", "Text402: 90037", "Text403: 310-555-1212", "Text404: 90037", "Text405: 310-555-1212", "Text406: 90037", "Text407: 310-555-1212", "Text408: 90037", "Text409: 310-555-1212", "Text410: 90037", "Text411: 310-555-1212", "Text412: 90037", "Text413: 310-555-1212", "Text414: 90037", "Text415: 310-555-1212", "Text416: 90037", "Text417: 310-555-1212", "Text418: 90037", "Text419: 310-555-1212", "Text420: 90037", "Text421: 310-555-1212", "Text422: 90037", "Text423: 310-555-1212", "Text424: 90037", "Text425: 310-555-1212", "Text426: 90037", "Text427: 310-555-1212", "Text428: 90037", "Text429: 310-555-1212", "Text430: 90037", "Text431: 310-555-1212", "Text432: 90037", "Text433: 310-555-1212", "Text434: 90037", "Text435: 310-555-1212", "Text436: 90037", "Text437: 310-555-1212", "Text438: 90037", "Text439: 310-555-1212", "Text440: 90037", "Text441: 310-555-1212", "Text442: 90037", "Text443: 310-555-1212", "Text444: 90037", "Text445: 310-555-1212", "Text446: 90037", "Text447: 310-555-1212", "Text448: 90037", "Text449: 310-555-1212", "Text450: 90037", "Text451: 310-555-1212", "Text452: 90037", "Text453: 310-555-1212", "Text454: 90037", "Text455: 310-555-1212", "Text456: 90037", "Text457: 310-555-1212", "Text458: 90037", "Text459: 310-555-1212", "Text460: 90037", "Text461: 310-555-1212", "Text462: 90037", "Text463: 310-555-1212", "Text464: 90037", "Text465: 310-555-1212", "Text466: 90037", "Text467: 310-555-1212", "Text468: 90037", "Text469: 310-555-1212", "Text470: 90037", "Text471: 310-555-1212", "Text472: 90037", "Text473: 310-555-1212", "Text474: 90037", "Text475: 310-555-1212", "Text476: 90037", "Text477: 310-555-1212", "Text478: 90037", "Text479: 310-555-1212", "Text480: 90037", "Text481: 310-555-1212", "Text482: 90037", "Text483: 310-555-1212", "Text484: 90037", "Text485: 310-555-1212", "Text486: 90037", "Text487: 310-555-1212", "Text488: 90037", "Text489: 310-555-1212", "Text490: 90037", "Text491: 310-555-1212", "Text492: 90037", "Text493: 310-555-1212", "Text494: 90037", "Text495: 310-555-1212", "Text496: 90037", "Text497: 310-555-1212", "Text498: 90037", "Text499: 310-555-1212", "Text500: 90037", "Text501: 310-555-1212", "Text502: 90037", "Text503: 310-555-1212", "Text504: 90037", "Text505: 310-555-1212", "Text506: 90037", "Text507: 310-555-1212", "Text508: 90037", "Text509: 310-555-1212", "Text510: 90037", "Text511: 310-555-1212", "Text512: 90037", "Text513: 310-555-1212", "Text514: 90037", "Text515: 310-555-1212", "Text516: 90037", "Text517: 310-555-1212", "Text518: 90037", "Text519: 310-555-1212", "Text520: 90037", "Text521: 310-555-1212", "Text522: 90037", "Text523: 310-555-1212", "Text524: 90037", "Text525: 310-555-1212", "Text526: 90037", "Text527: 310-555-1212", "Text528: 90037", "Text529: 310-555-1212", "Text530: 90037", "Text531: 310-555-1212", "Text532: 90037", "Text533: 310-555-1212", "Text534: 90037", "Text535: 310-555-1212", "Text536: 90037", "Text537: 310-555-1212", "Text538: 90037", "Text539: 310-555-1212", "Text540: 90037", "Text541: 310-555-1212", "Text542: 90037", "Text543: 310-555-1212", "Text544: 90037", "Text545: 310-555-1212", "Text546: 90037", "Text547: 310-555-1212", "Text548: 90037", "Text549: 310-555-1212", "Text550: 90037", "Text551: 310-555-1212", "Text552: 90037", "Text553: 310-555-1212", "Text554: 90037", "Text555: 310-555-1212", "Text556: 90037", "Text557: 310-555-1212", "Text558: 90037", "Text559: 310-555-1212", "Text560: 90037", "Text561: 310-555-1212", "Text562: 90037", "Text563: 310-555-1212", "Text564: 90037", "Text565: 310-555-1212", "Text566: 90037", "Text567: 310-555-1212", "Text568: 90037", "Text569: 310-555-1212", "Text570: 90037", "Text571: 310-555-1212", "Text572: 90037", "Text573: 310-555-1212", "Text574: 90037", "Text575: 310-555-1212", "Text576: 90037", "Text577: 310-555-1212", "Text578: 90037", "Text579: 310-555-1212", "Text580: 90037", "Text581: 310-555-1212", "Text582: 90037", "Text583: 310-555-1212", "Text584: 90037", "Text585: 310-555-1212", "Text586: 90037", "Text587: 310-555-1212", "Text588: 90037", "Text589: 310-555-1212", "Text590: 90037", "Text591: 310-555-1212", "Text592: 90037", "Text593: 310-555-1212", "Text594: 90037", "Text595: 310-555-1212", "Text596: 90037", "Text597: 310-555-1212", "Text598: 90037", "Text599: 310-555-1212", "Text600: 90037", "Text601: 310-555-1212", "Text602: 90037", "Text603: 310-555-1212", "Text604: 90037", "Text605: 310-555-1212", "Text606: 90037", "Text607: 310-555-1212", "Text608: 90037", "Text609: 310-555-1212", "Text610: 90037", "Text611: 310-555-1212", "Text612: 90037", "Text613: 310-555-1212", "Text614: 90037", "Text615: 310-555-1212", "Text616: 90037", "Text617: 310-555-1212", "Text618: 90037", "Text619: 310-555-1212", "Text620: 90037", "Text621: 310-555-1212", "Text622: 90037", "Text623: 310-555-1212", "Text624: 90037", "Text625: 310-555-1212", "Text626: 90037", "Text627: 310-555-1212", "Text628: 90037", "Text629: 310-555-1212", "Text630: 90037", "Text631: 310-555-1212", "Text632: 90037", "Text633: 310-555-1212", "Text634: 90037", "Text635: 310-555-1212", "Text636: 90037", "Text637: 310-555-1212", "Text638: 90037", "Text639: 310-555-1212", "Text640: 90037", "Text641: 310-555-1212", "Text642: 90037", "Text643: 310-555-1212", "Text644: 90037", "Text645: 310-555-1212", "Text646: 90037", "Text647: 310-555-1212", "Text648: 90037", "Text649: 310-555-1212", "Text650: 90037", "Text651: 310-555-1212", "Text652: 90037", "Text653: 310-555-1212", "Text654: 90037", "Text655: 310-555-1212", "Text656: 90037", "Text657: 310-555-1212", "Text658: 90037", "Text659: 310-555-1212", "Text660: 90037", "Text661: 310-555-1212", "Text662: 90037", "Text663: 310-555-1212", "Text664: 90037", "Text665: 310-555-1212", "Text666: 90037", "Text667: 310-555-1212", "Text668: 90037", "Text669: 310-555-1212", "Text670: 90037", "Text671: 310-555-1212", "Text672: 90037", "Text673: 310-555-1212", "Text674: 90037", "Text675: 310-555-1212", "Text676: 90037", "Text677: 310-555-1212", "Text678: 90037", "Text679: 310-555-1212", "Text680: 90037", "Text681: 310-555-1212", "Text682: 90037", "Text683: 310-555-1212", "Text684: 90037", "Text685: 310-555-1212", "Text686: 90037", "Text687: 310-555-1212", "Text688: 90037", "Text689: 310-555-1212", "Text690: 90037", "Text691: 310-555-1212", "Text692: 90037", "Text693: 310-555-1212", "Text694: 90037", "Text695: 310-555-1212", "Text696: 90037", "Text697: 310-555-1212", "Text698: 90037", "Text699: 310-555-1212", "Text700: 90037", "Text701: 310-555-1212", "Text702: 90037", "Text703: 310-555-1212", "Text704: 90037", "Text705: 310-555-1212", "Text706: 90037", "Text707: 310-555-1212", "Text708: 9003

Uno de los propósitos de gran utilidad de los informes es mostrar resultados que no se pueden representar directamente en un formato de tabla de base de datos. Un informe tiene que permitir ver los detalles importantes a primera vista:

- Agrupaciones.
- Cálculos agrupados como totales, subtotales, medias y otros.
- Resúmenes.
- Datos cruzados.

1.1 Informes incrustados y no incrustados en la aplicación

Al planear la creación de una aplicación con un entorno de desarrollo integrado (IDE), una de las consideraciones más importantes es si se utilizan informes incrustados o no incrustados. Saber cuáles son los aspectos fundamentales que afectan la incrustación de informes nos ayudará a elegir la mejor estructura para el proyecto.

¿Qué diferencia hay entre los informes incrustados y los informes no incrustados? Un informe incrustado es un informe que se ha importado a un proyecto del IDE o que se ha creado dentro del IDE. Cuando un informe se incrusta en el proyecto, automáticamente se genera una clase contenedora para el informe.

Un informe no incrustado es un informe externo al proyecto del IDE. Hay muchas maneras de acceder al informe para cargarlo en un modelo de objetos a fin de habilitar la interacción mediante programación, pero el informe siempre será externo al proyecto del IDE. Vamos a comprobar cómo funciona un informe incrustado y uno no incrustado:

En el **informe incrustado**, cuando el informe se importa al proyecto o se crea en él, se genera una clase contenedora que normalmente tiene el mismo nombre que el informe. Esta clase contiene o representa el informe en el proyecto.

Cuando pasa esto, todo el código del proyecto interactúa con la clase del informe que se ha creado para representarlo en vez de hacerlo con el archivo mismo del informe original.

Al compilar el proyecto, tanto el informe como su clase contenedora se incrustan en el ensamblaje, tal y como pasaría con cualquier otro recurso del proyecto.

En el **informe no incrustado**, el acceso siempre se obtiene externamente. El software puede acceder de varias maneras. Por ejemplo:

- El informe puede estar en una unidad de disco externo.
- El informe puede estar disponible como servicio web de informes.

Nunca se importan informes no incrustados en el proyecto y, por lo tanto, nunca se crea ninguna clase contenedora de informe, a diferencia de los informes incrustados. En vez de esto, el informe no incrustado se carga en uno de los modelos de objetos en tiempos de ejecución.

Para simplificar la implementación del proyecto, es preferible que se utilicen informes incrustados. Tendremos que trabajar con menos archivos y no nos tendremos que preocupar de si los informes están colocados correctamente en la ruta de directorio de ficheros definida. Además, esta solución es más segura, puesto que los informes no se exponen a modificaciones.

Si estamos aprendiendo a desarrollar y a implementar, nos será más fácil incrustar los informes. Por ejemplo, si trabajáis con el entorno a desarrollo Microsoft Visual Studio, los informes, una vez incrustados, siempre aparecen como una clase del proyecto, están disponibles desde el IntelliSense y se ven en el explorador de objetos. No nos tenemos que preocupar de si se mueven o se eliminan en el directorio de ficheros ni de escribir la ruta correctamente.

Los informes incrustados son más sencillos y seguros, pero comportan más trabajo:

- No se pueden modificar sin volver a compilar todo el proyecto.
- Hay límites en cuanto al tamaño que puede tener un informe incrustado.
- Un informe muy grande se compila como un recurso incrustado.

Por otro lado, si los informes se tienen que modificar regularmente, es preferible que se utilicen informes no incrustados. De este modo, podremos acceder y los podremos modificar más fácilmente, sin tener que preocuparnos por la necesidad de volver a compilar el proyecto.

1.2 Herramientas gráficas integradas en el IDE y externas al mismo

Hoy en día, la mayoría de los entornos de desarrollo contienen herramientas para el desarrollo y generación de informes. Son lo que se conoce con el nombre de motores de informes y que permiten, mediante una interfaz gráfica de usuario, indicar la estructura y diseño final de los elementos que aparecerán en el informe.

Si nuestro entorno de desarrollo no presenta un motor de informes siempre podemos recurrir a una herramienta de generación de informes externa al IDE.

En la actualidad existen muchos motores de informes, tanto libres como propietarios, entre los que se pueden destacar los siguientes:

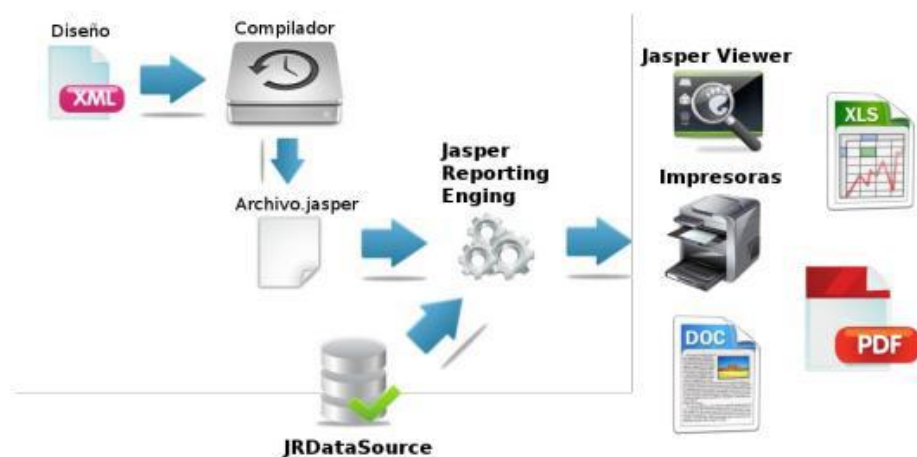
1. SAP Crystal Reports.

Es una solución intuitiva de generación de informes que ayuda a los desarrolladores a crear informes flexibles, fiables y llenos de funcionalidades, e integrarlos en aplicaciones tanto de cliente ligero como completo. La solución Crystal Reports consta de:

- Diseñador de informes.
- Integración con aplicaciones como, por ejemplo, con Visual Studio.
- Administración y distribución de los informes.

2. JasperSoft.

Es una herramienta que tiene una versión community y, por lo tanto, de versión libre y que permite la generación de informes en formatos PDF, HTML, XLS, CSV y XML. Además, como ventaja podemos destacar que está escrita completamente en Java y, por tanto, puede ser usada en aplicaciones web y, por supuesto, en aplicaciones Java.



Una de sus principales ventajas es la generación de informes en XML que funciona de la siguiente manera: genera un archivo en XML con extensión .jrxml que contiene el diseño y estructura del informe y, una vez compilado este archivo, se genera otro con extensión .jasper, que es el que se rellena con la información extraída del origen de datos.

3. Eclipse Birt.

BIRT (*Bussiness Intelligence Reporting Tools*) es un proyecto de software de código abierto que permite la generación de informes para aplicaciones web, basadas en Java o en Java EE.

Los informes de BIRT, al igual que en el caso de JasperSoft, se hacen en XML y son compatibles con gran cantidad de orígenes de datos.

BIRT tiene dos componentes principales: un diseñador de informes visuales basado en Eclipse y un componente que se puede agregar al servidor de aplicaciones y que genera informes en tiempo de ejecución. Además, también incluye un motor de gráficos integrado en el diseñador de informes.



PARA SABER MÁS

Aquí tienes más información sobre las diferentes herramientas y posibilidades de reporters en Java:



4. Pentaho.

Pentaho Reporting es una clase de Java que permite la generación de informes. Como en el caso de otras librerías y herramientas, proporciona soporte para múltiples formatos de salida como PDF, Excel, XHTML, Texto Plano, XML o CSV.

También existe un diseñador gráfico independiente para la definición del informe y después incorporarlo a la aplicación.



EJEMPLO PRÁCTICO

Después de haber evaluado y revisado varias herramientas del mercado, dejas para la decisión final JasperSoft y Eclipse Birt. Teniendo en cuenta las siguientes características técnicas de la empresa:

- Aplicaciones realizadas con J2EE y Spring.
- IDE de desarrollo Eclipse.
- Servidor de datos y aplicaciones centralizado.
- Aplicaciones de ventas, logística y administración contra esos sistemas.

¿Cuál de las dos herramientas elegirías?

Tienes en cuenta lo siguiente:

- Tanto BIRT como JasperSoft Studio son herramientas construidas sobre Eclipse. Por lo tanto, no habrá ninguna diferencia entre una y otra.
- BIRT permite la realización de múltiples fuentes de datos en un mismo informe, algo que es muy interesante para la integración de datos.
- JasperSoft tiene más elementos visuales y gráficos, aunque BIRT se puede ampliar fácilmente y customizar de una forma más amplia si se tiene experiencia.
- Ambos pueden usar los motores para integrar dentro de las aplicaciones directamente.

Ante esta comparativa, la empresa se decanta por BIRT, ya que proporciona una mayor flexibilidad y, además, permite alcanzar múltiples fuentes de información, imprescindible para la presentación de los informes.



VÍDEO DE INTERÉS

En el vídeo puedes ver una introducción a Pentaho, una tecnología increíble para realizar BI:



2. JASPERSOFT

Una vez que has seleccionado JasperSoft como la herramienta que usarás para realizar tus informes y desarrollos dentro de la empresa, necesitas realizar unas primeras pruebas y tener un primer entorno de desarrollo preparado.

Durante el proceso de preparación del entorno con JasperSoft, puedes conectar directamente a la base de datos que la empresa ya tiene creada utilizando un conector JDBC, además de otras fuentes como archivos XML u hojas de cálculo. Generarás informes en formatos como PDF y HTML con información de la base de datos, incluyendo valores calculados, gráficos e imágenes.

Dentro del paquete de programas de Jaspersoft Community nos encontramos con JasperSoft Studio y JasperReports Library, que es uno de los motores gratuitos más usados en el mundo para aplicaciones Java.

JasperReports Library 7.0.1 (2024-10-08)

JasperReports Library 7.0.1 (.zip)

JasperReports Library 7.0.1 (.tar.gz)

[More download options](#)

JasperSoft Library

Fuente: Elaboración propia

Es una herramienta escrita completamente en Java, de ahí la perfecta incorporación con nuestros proyectos también desarrollados con Java.



VÍDEO DE INTERÉS

Aquí podrás ver cómo JasperSoft se usa también en otras herramientas como Odoo:



2.1 Instalación

Para la instalación descargaremos la versión gratuita de la herramienta, dentro de la web y seleccionaremos el editor Jaspersoft Studio 7.0.0. La versión gratuita nos solicitará que nos creemos una cuenta en la web de Jaspersoft.



ENLACE DE INTERÉS

Accede para encontrar un acceso directo a la descarga de Jaspersoft:



Download Jaspersoft

Experience why Jaspersoft is one of the most downloaded reporting software of all time through one of our two product editions – commercial or community – to see which one is best for you.



Jaspersoft® community edition

Experience what you can do with Jaspersoft for free.

Our community edition allows you to explore some of the most popular capabilities Jaspersoft offers. Built on an open source foundation, the community edition includes JasperReports® Library and Jaspersoft Studio, allowing you to design, build, and export highly-formatted, pixel-perfect reports, in an array of file formats.

Download Now



Jaspersoft® commercial edition

Everything you need to design and deliver any report you need.

Our commercial edition lets you design, distribute and securely manage self-service ad hoc and other reports, dashboards, and visualizations. Deliver them in real-time or on a predetermined schedule, quickly and efficiently, in any format.

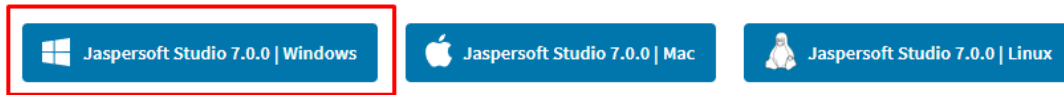
With Jaspersoft's commercial edition you will receive 24/7 customer support, secure multi-tenant management, product warranties and indemnification.

Take a 30-day Trial

Jaspersoft Versión Gratuita

Fuente: Elaboración propia

Jaspersoft Studio 7.0.0 (2024-06-21)

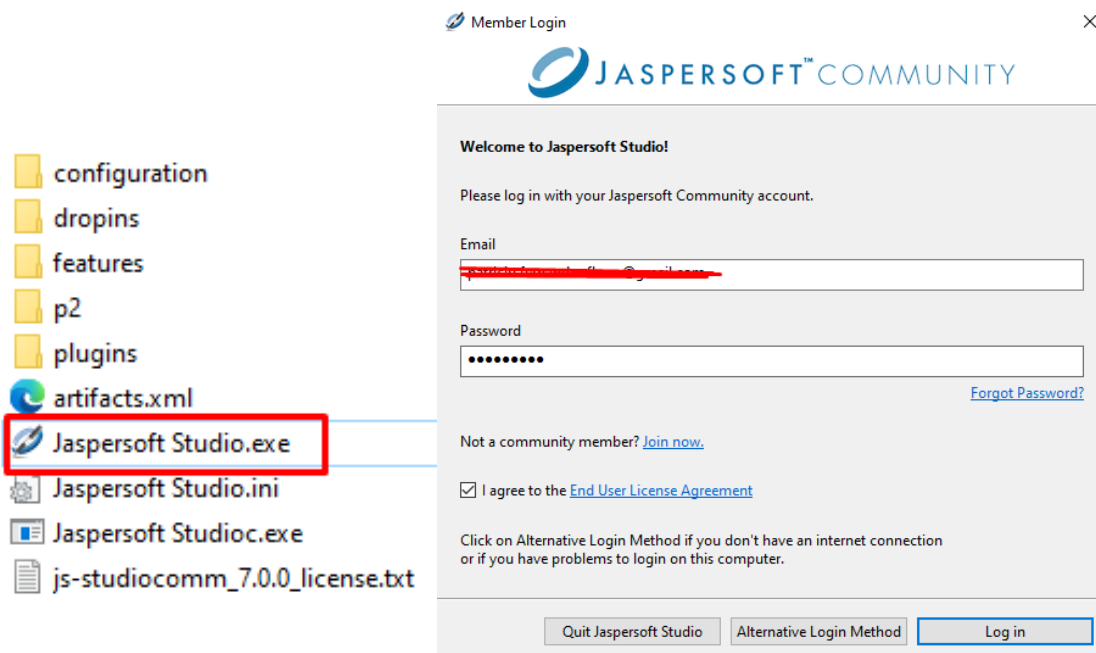


[More download options](#)

Descarga JaspersoftStudio

Fuente: Elaboración propia

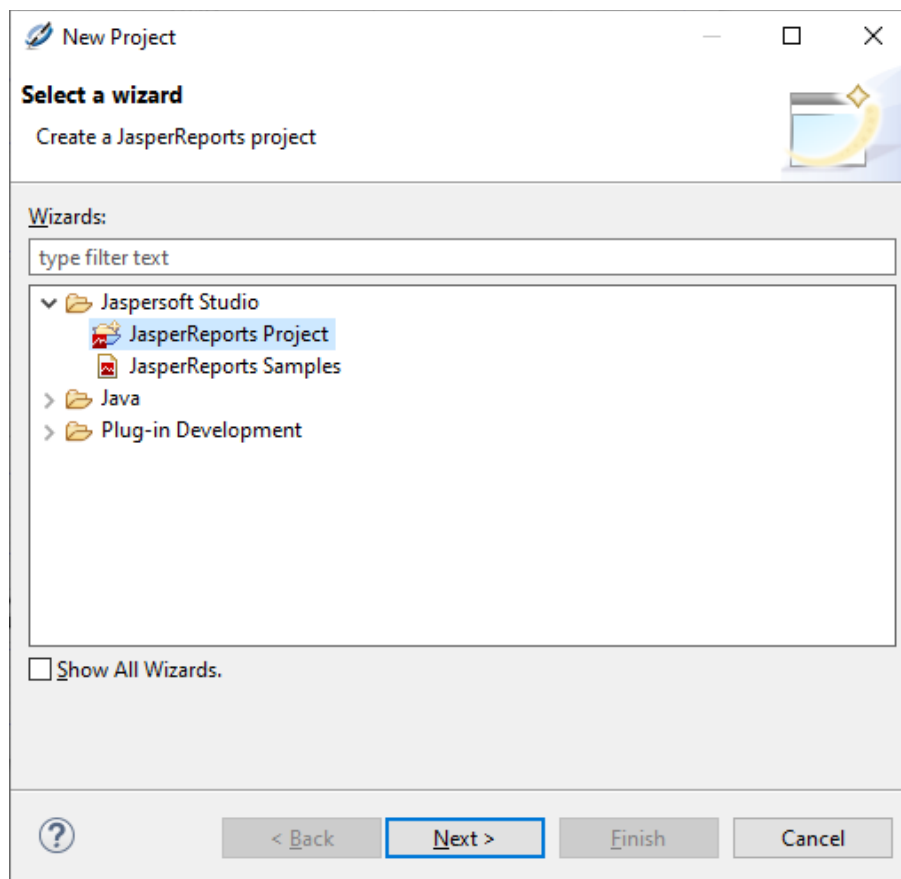
Una vez descargados descomprimos los ficheros, dentro de la carpeta encontraremos un fichero llamado “Jaspersoft Studio.exe” que nos permitirá ejecutar la interfaz. Una vez lo lancemos nos solicitará nuestra cuenta creada previamente.



Ejecutando JaspersoftStudio

Fuente: Elaboración propia

A continuación, crearemos un proyecto, para ello pulsaremos en “File -> New -> Project..” pulsaremos en “JasperReport Project” y le daremos un nombre por ejemplo: Reporte Clase.

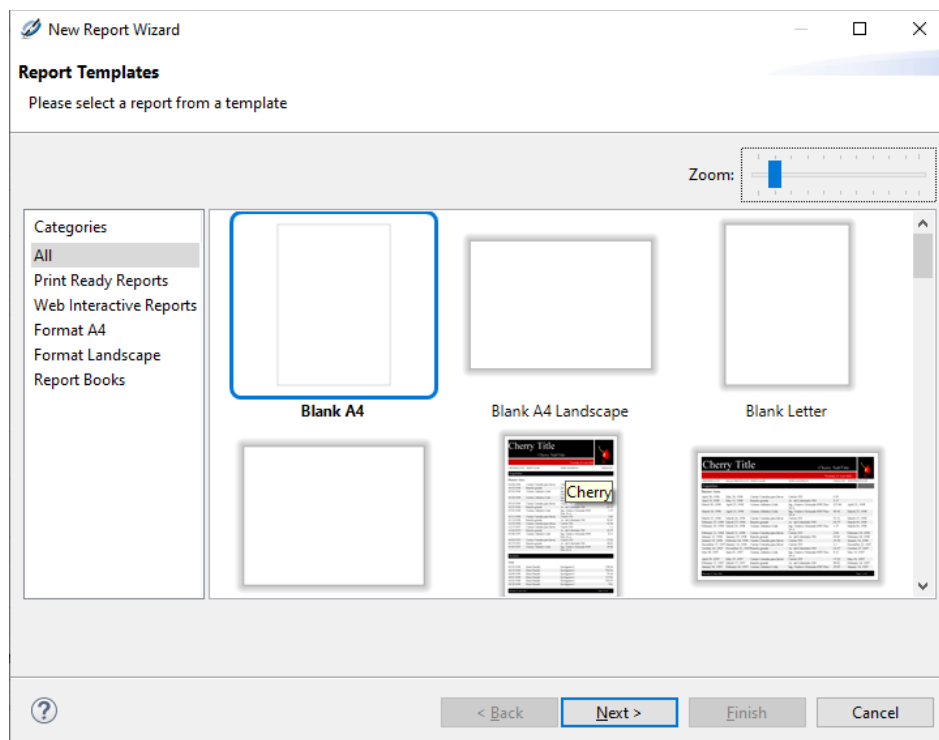


Creación de proyecto JaspersoftStudio

Fuente: Elaboración propia

2.2 Creando un nuevo informe. Conexión con las fuentes de datos

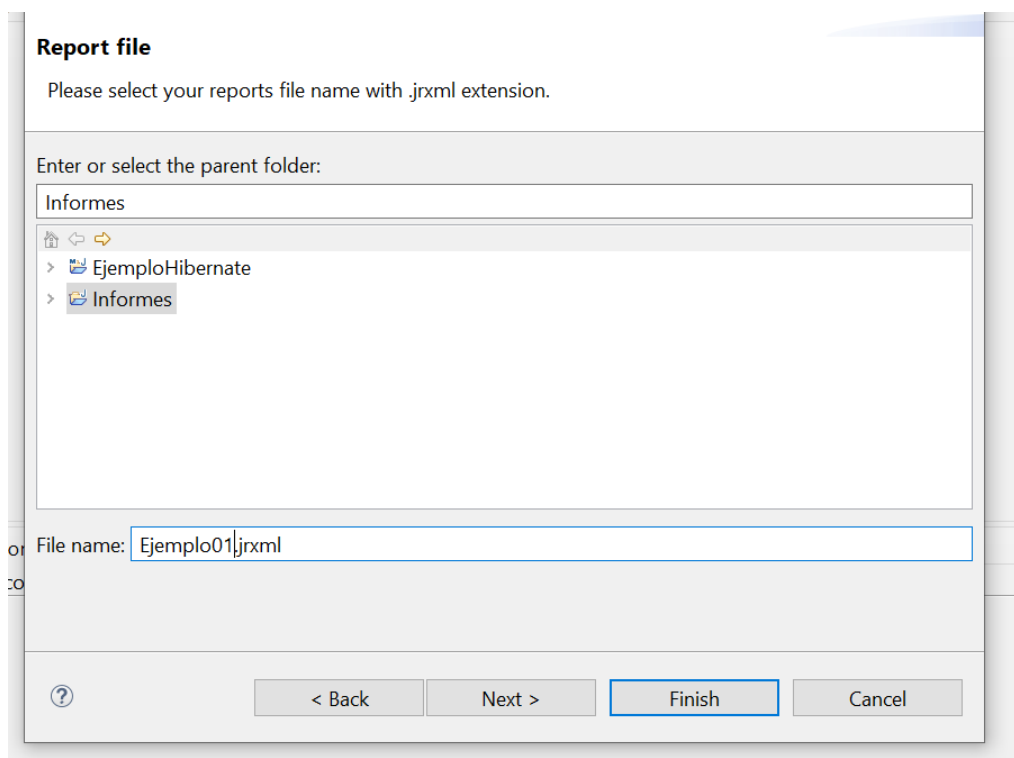
Para crear un nuevo informe, tendremos que ir al menú File->New->Jasper Report. En este punto nos aparecerán todas las posibles plantillas. Para comenzar, elegiremos la plantilla en blanco.



Selección de plantilla

Fuente: Elaboración propia

Elegiremos el proyecto Java donde queremos incluir nuestros informes, así como el nombre de nuestro informe.



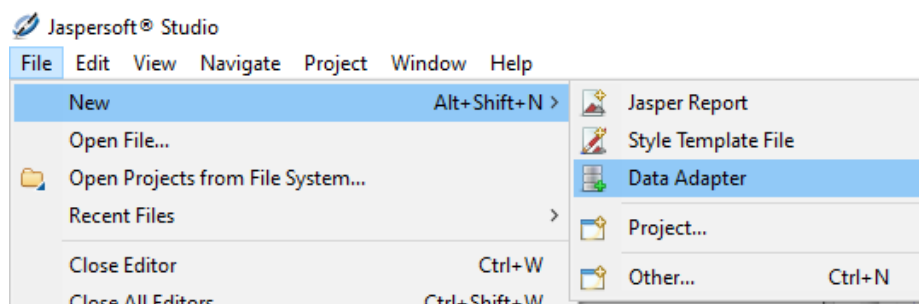
Nombre de informe

Fuente: Elaboración propia

Uno de los primeros pasos para crear un informe siempre es identificar un origen de datos. Algunas fuentes de datos son las que se considerarían bases de datos tradicionales como Microsoft SQL Server, Oracle, el Informix o MySQL que vamos a usar nosotros. Otras fuentes de datos son formas más abstractas de datos como ficheros de registro, correo electrónico, XML, hojas de cálculo y bases de datos multidimensionales (OLAP).

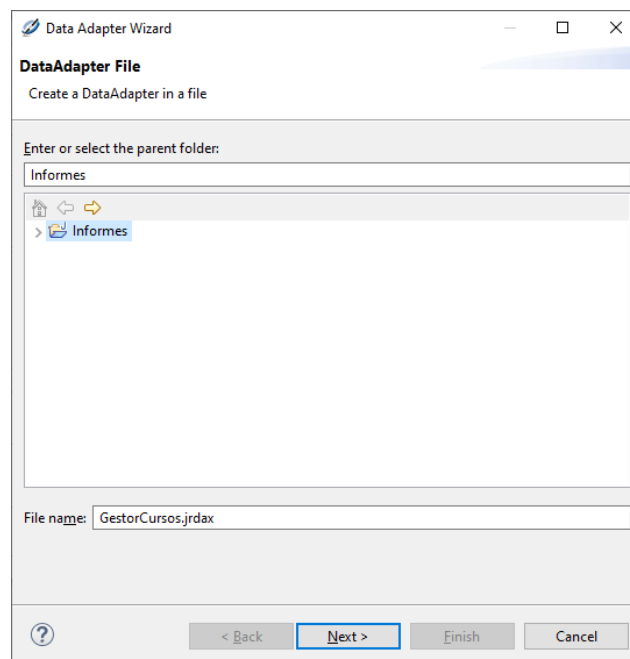
Para apoyar a todas las fuentes de datos y, a la vez, preservar una experiencia de usuario, el apoyo para cada fuente de datos se proporciona por medio de un controlador de base de datos. Un controlador de base de datos actúa como una pasarela entre la aplicación y un tipo específico de base de datos o tecnología de acceso a datos.

En la pantalla y conexión con la base de datos crearemos un nuevo origen de datos a través del conector JDBC, tal y como muestra la imagen. Para ello deberemos crear una base de datos MySQL previamente y dejarla activa.



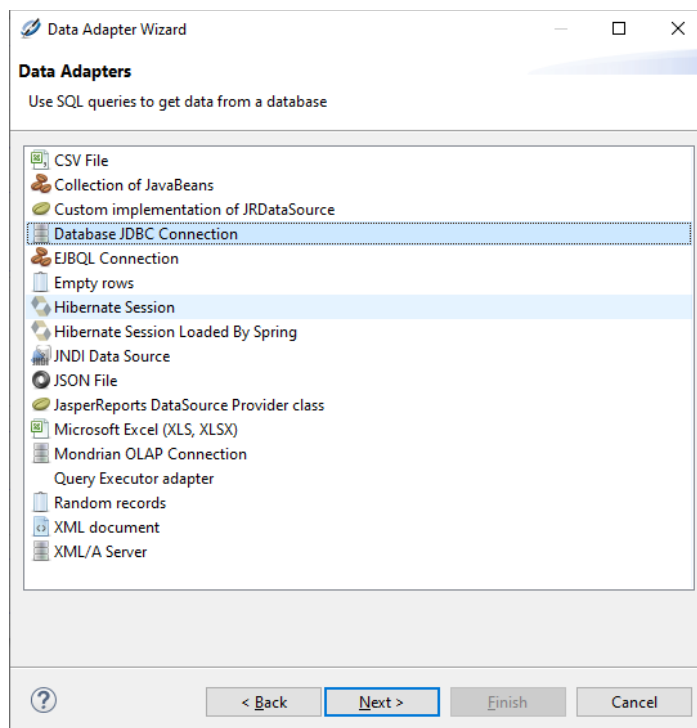
Creación de fuente de datos

Fuente: Elaboración propia



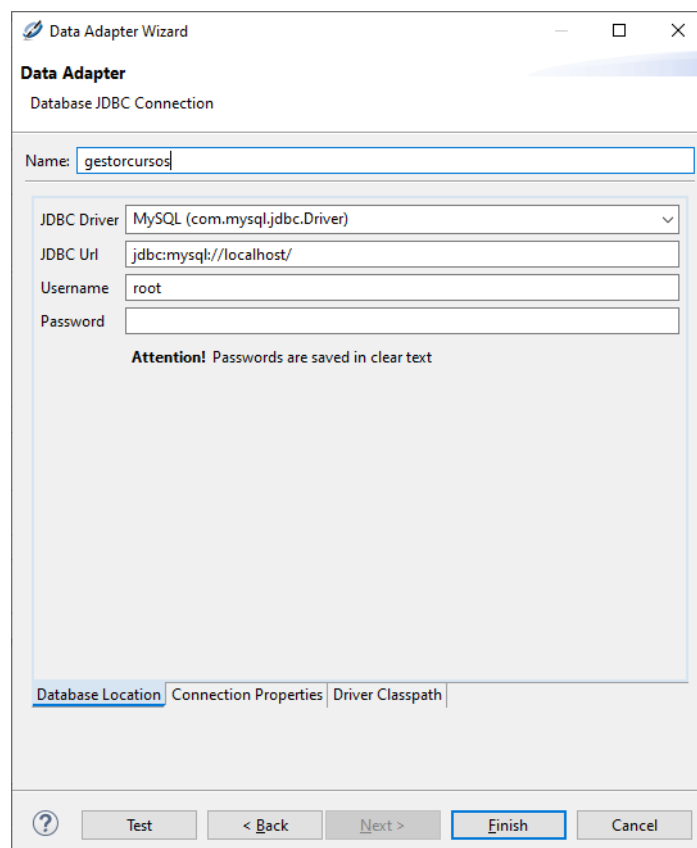
Creación de fuente de datos

Fuente: Elaboración propia



Selección de conector

Fuente: Elaboración propia



Parámetros del conector

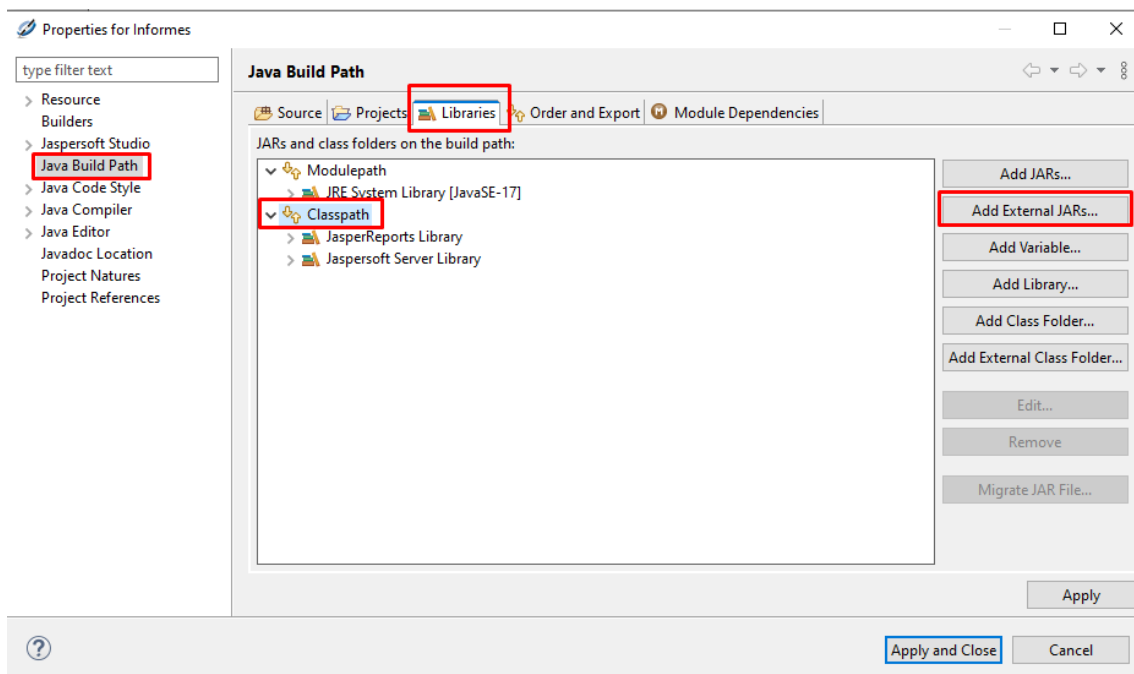
Fuente: Elaboración propia

Una vez finalizada nos permitirá probar la conexión, pero antes hay que importar el conector de MySQL.



Pulsaremos botón derecho en el proyecto y le daremos a “Properties”.

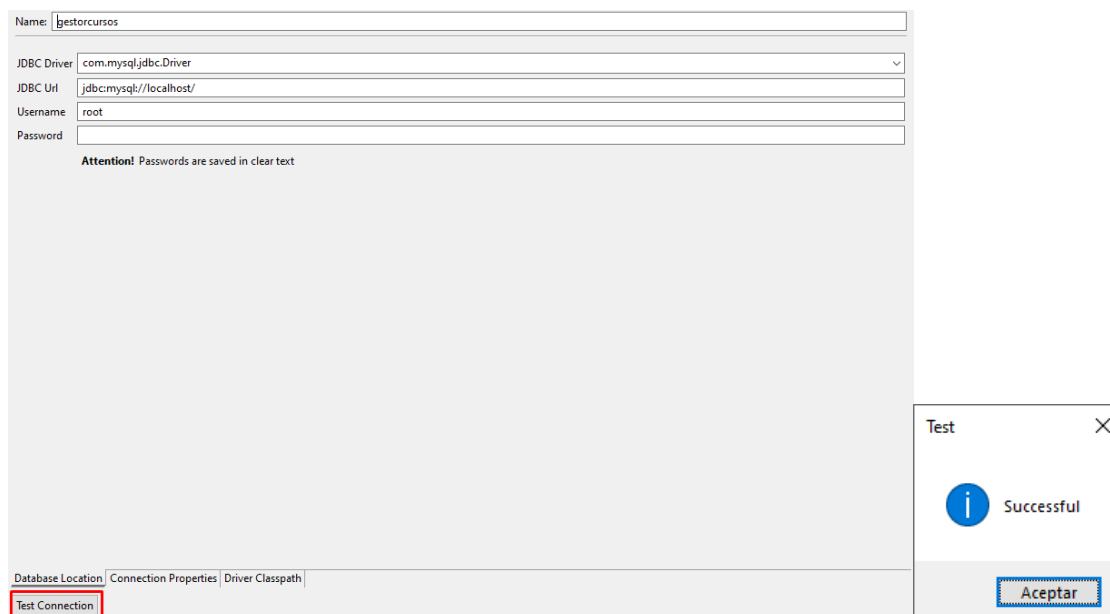
Dentro de “Properties” iremos a “Java Build Path” y “Libraries”, pulsaremos en “Classpath” y “Add External Jars”. Ahí seleccionaremos el conector JDBC que suele encontrarse en la ruta: “C:\Program Files (x86)\MySQL\Connector J 8.0”.



Importando el conector

Fuente: Elaboración propia

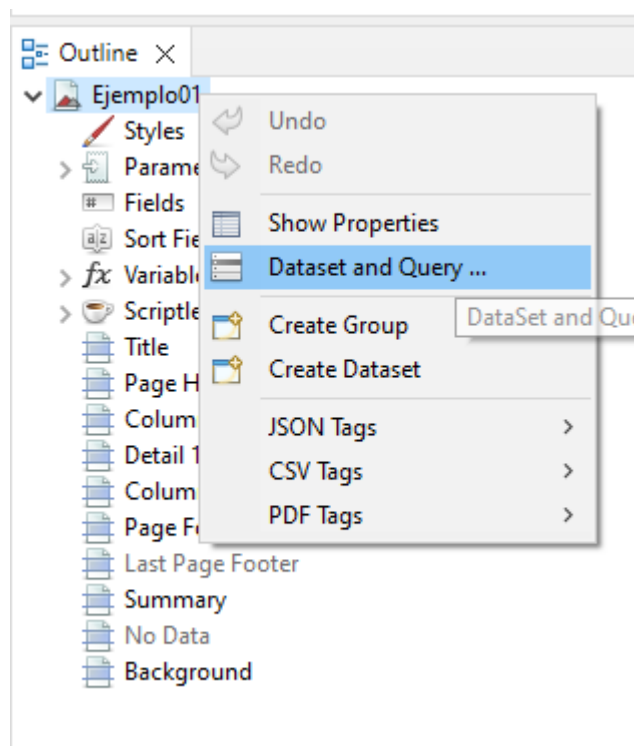
Pulsaremos “Apply and Close” y ya sí podemos probar la conexión.



Probando Conexión

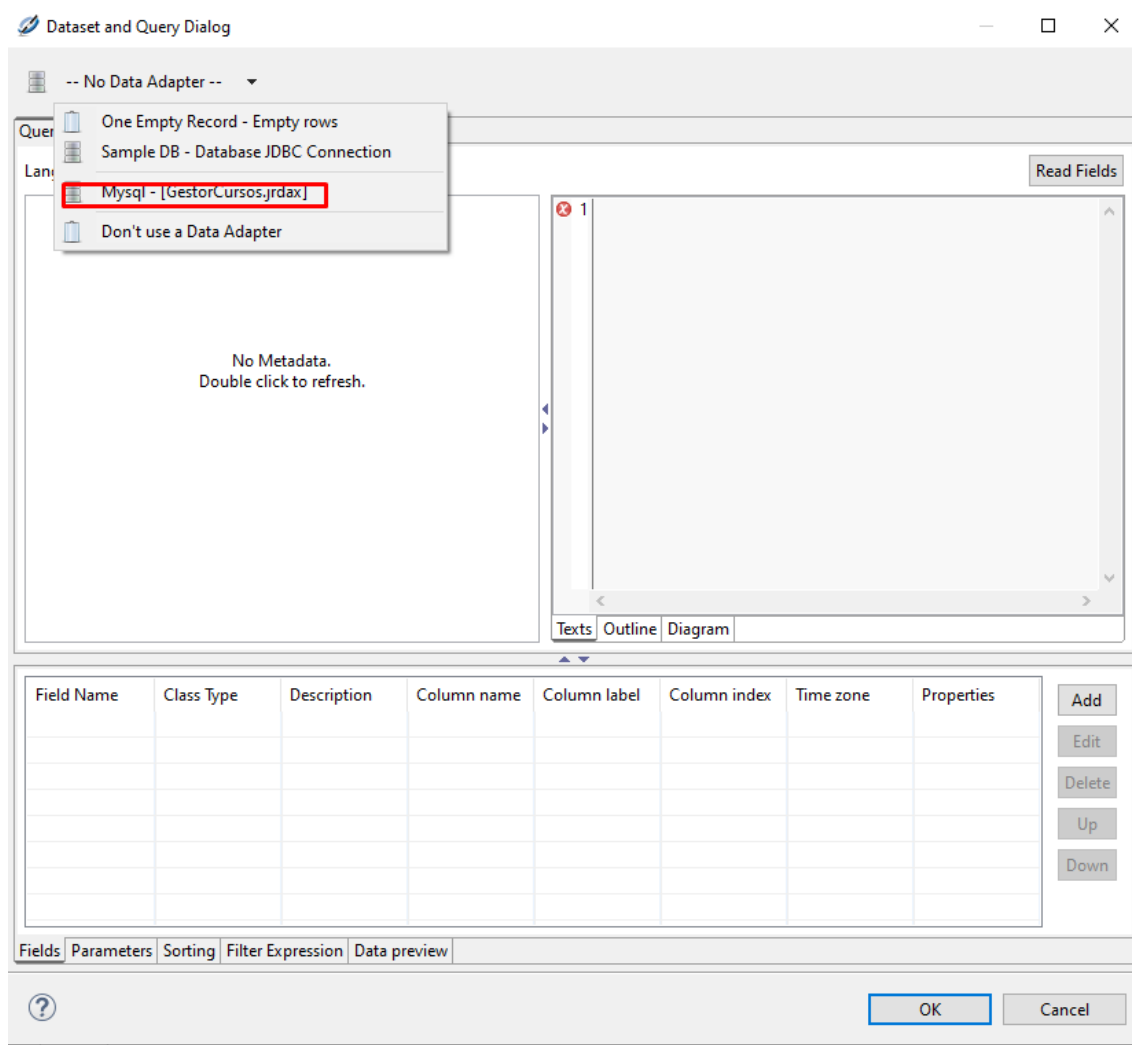
Fuente: Elaboración propia

Ya que funciona podemos configurar el origen de datos del reporte como el adaptador que acabamos de crear. Para ello, en el árbol del reporte, pulsamos botón derecho en el padre y “Dataset and Query”, luego se pone como origen de datos el adaptador que acabamos de crear.



Configurando el Adaptador en el Reporte

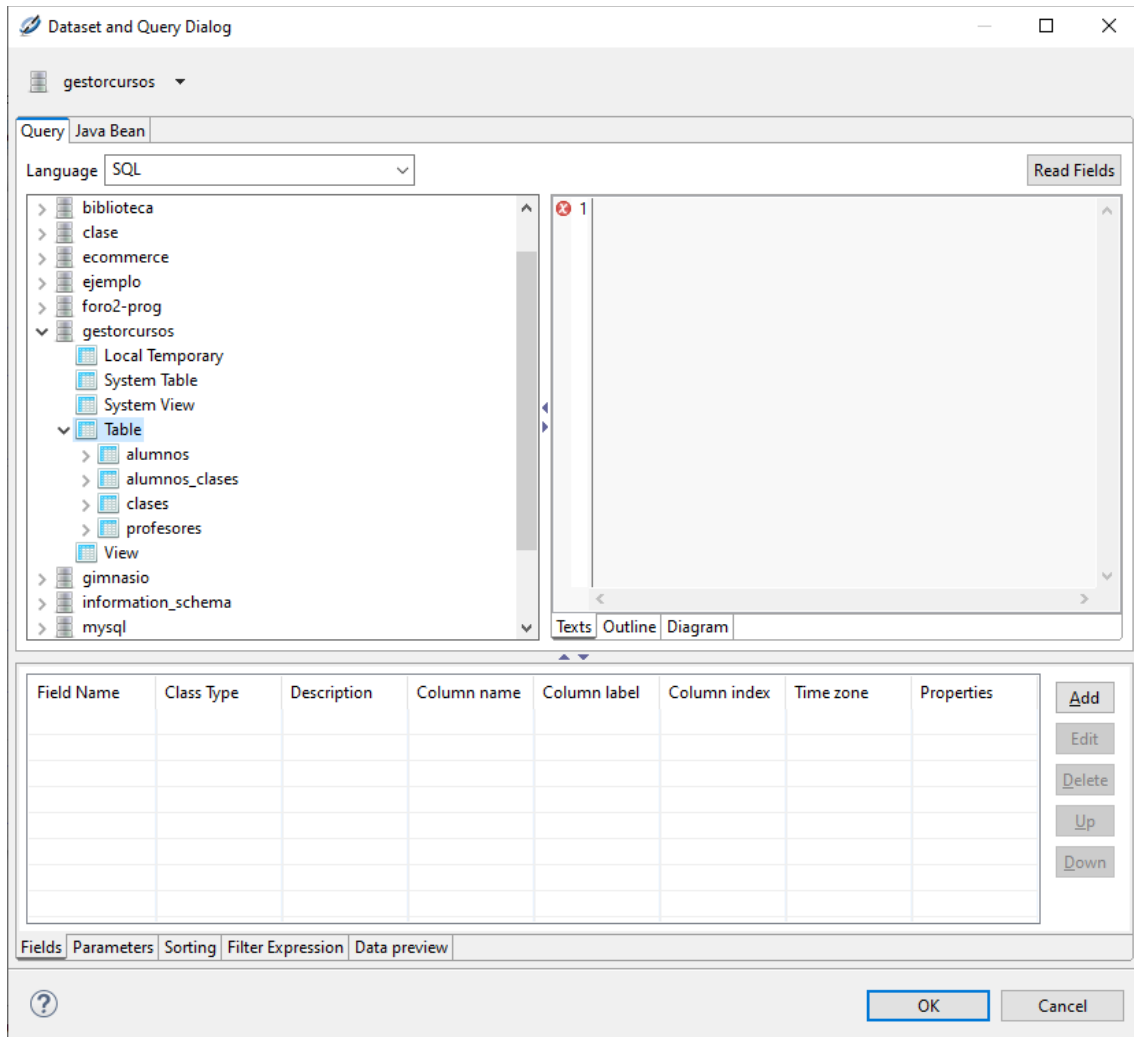
Fuente: Elaboración propia



Configurando el Adaptador en el Reporte

Fuente: Elaboración propia

Una vez configurados los parámetros básicos, en la pantalla de Data Source nos debe aparecer las diferentes tablas y bases de datos.

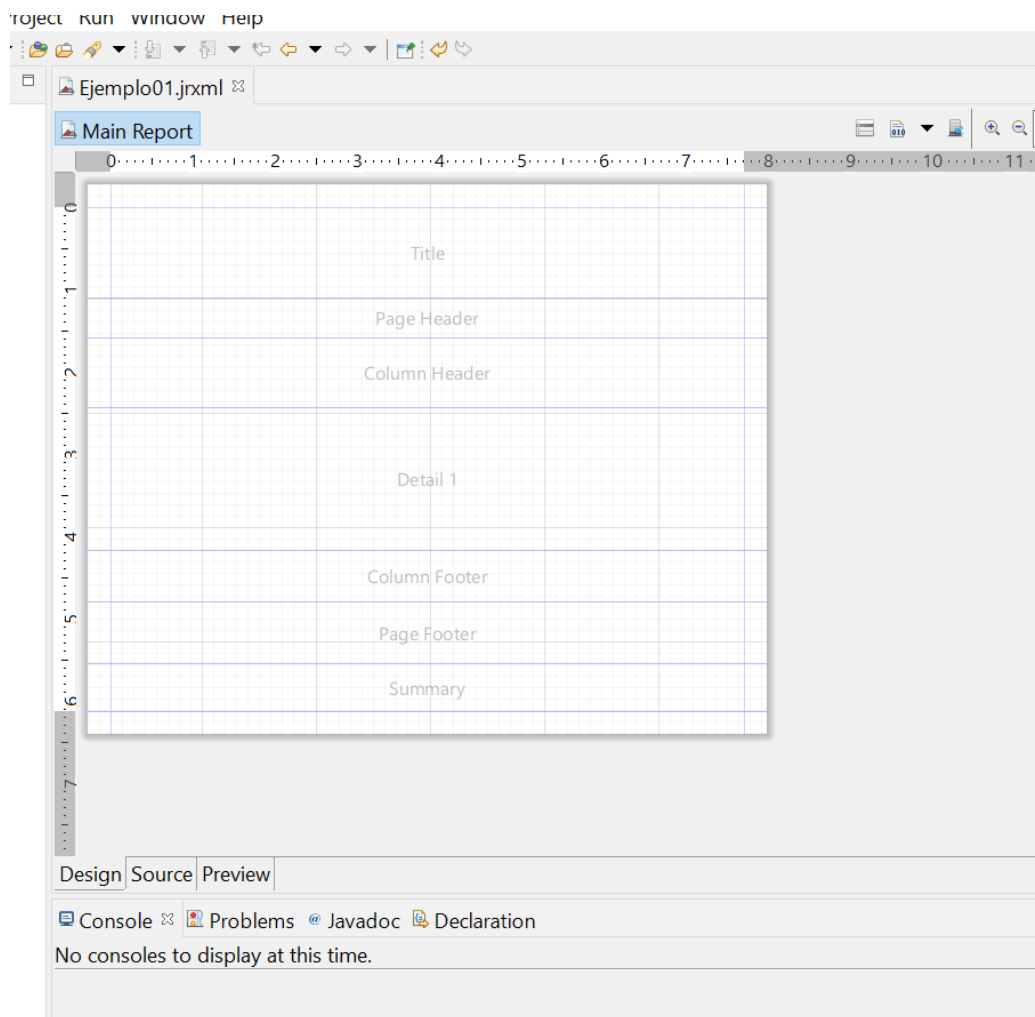


Campos Base de Datos

Fuente: Elaboración propia

2.3 Estructura general. Secciones

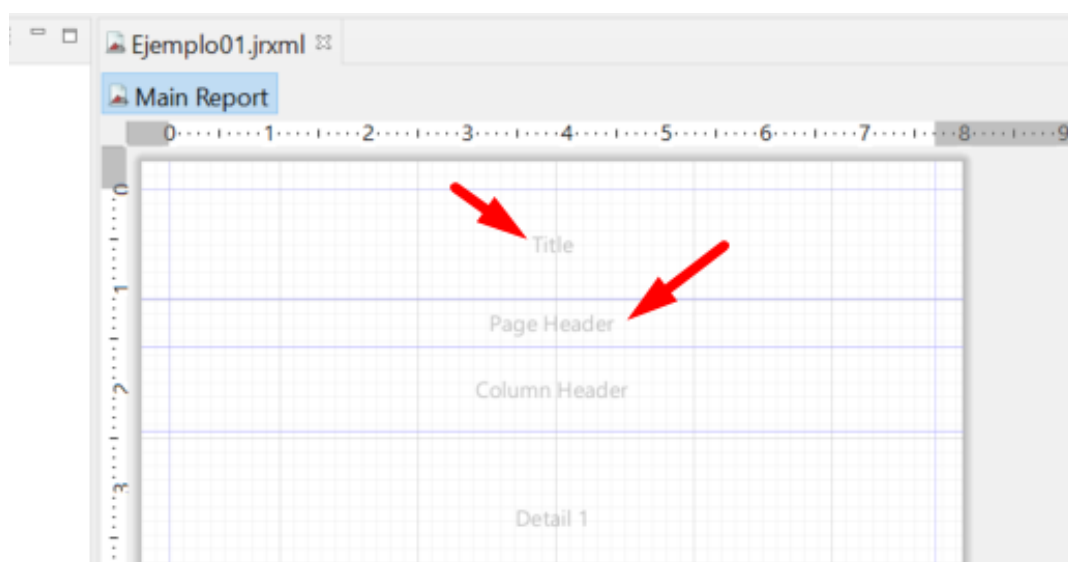
Los informes usan varias secciones para controlar el diseño y las agrupaciones de los informes. Estas secciones siempre se organizan en un orden específico, como si se tratara de un esquema. A veces, una sección en particular puede estar vacía. Por ejemplo, es posible que no se quiera utilizar el pie del informe si no se tiene que posar nada.



Estructura de un informe
Fuente: Elaboración propia

A continuación, se muestran las secciones que podrían aparecer en un informe y que podemos ver en la imagen:

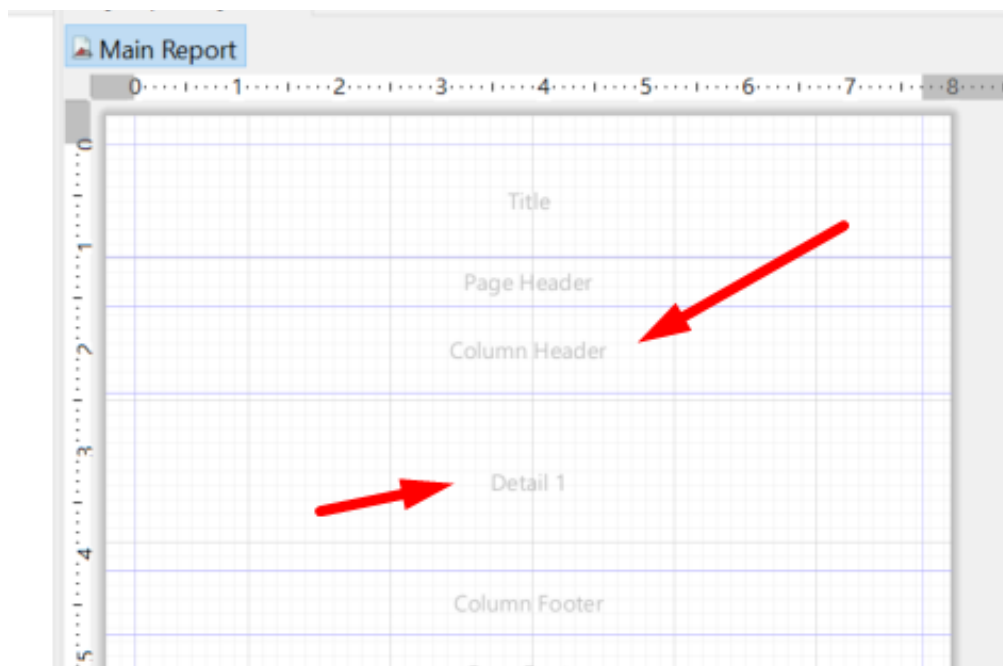
- **Cabecera del informe:** aparece una vez a comienzos de un informe. A menudo incluye el título del informe y la fecha en que se produce. También se puede usar para insertar gráficos y mesas cruzadas que hagan referencia en todo el informe.



Cabecera del informe y de página

Fuente: Elaboración propia

- **Cabecera de página:** se utiliza para añadir información que aparecerá a la parte superior de cada página del informe. Se puede usar para escribir el encabezamiento de columna, el número de página o la fecha.
- **Cabecera de grupo:** se usa en el grupo de registros de un informe para indicar donde empieza cada grupo. Normalmente incluye el valor del campo que se utiliza para la agrupación. Si tenéis múltiples niveles de agrupación, cada uno de estos niveles tendrá su propio grupo encabezado.
- **Detalle:** es la sección más interna. Constituye el cuerpo del informe y se imprime una vez por registro (fila). Muestra los detalles de cada registro. La mayoría de los datos del informe aparecen en esta sección.



Cabecera de grupo y de detalle

Fuente: Elaboración propia

- **Pie de página:** se usa para especificar la información que se quiere que aparezca a la parte inferior de cada página del informe, como el número de página o la fecha.
- **Pie del informe:** contiene la información que se quiere que aparezca de un golpe en la parte final del informe (por ejemplo, totales generales). También puede haber los gráficos y las mesas cruzadas que incluyen datos relativos en todo el informe. A menudo se deja en blanco.

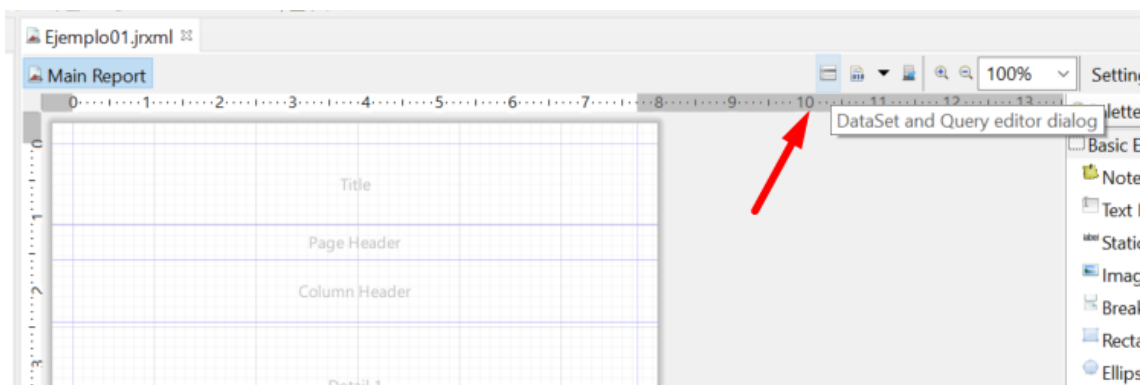
2.4 Ejecución de consultas

En cuanto a los informes basados en bases de datos, se necesitaría una manera de especificar qué campos de las tablas tienen que aparecer, qué filtros se tienen que aplicar y cómo se tienen que ordenar los datos.

El lenguaje SQL proporciona esta posibilidad. La sintaxis de este lenguaje se basa en un sistema de envío de las instrucciones SQL al servidor de base de datos SQL.

Cada directriz es una petición para hacer una operación de base de datos: crear un archivo de base de datos, incorporar mesas y campos a una base de datos, añadir registros a mesas o recuperar datos de bases de datos. El servidor SQL analiza la instrucción SQL y efectúa la operación solicitada. Por ejemplo, si la declaración es una petición de datos, el servidor recoge los datos y los devuelve al programa cliente porque los trate.

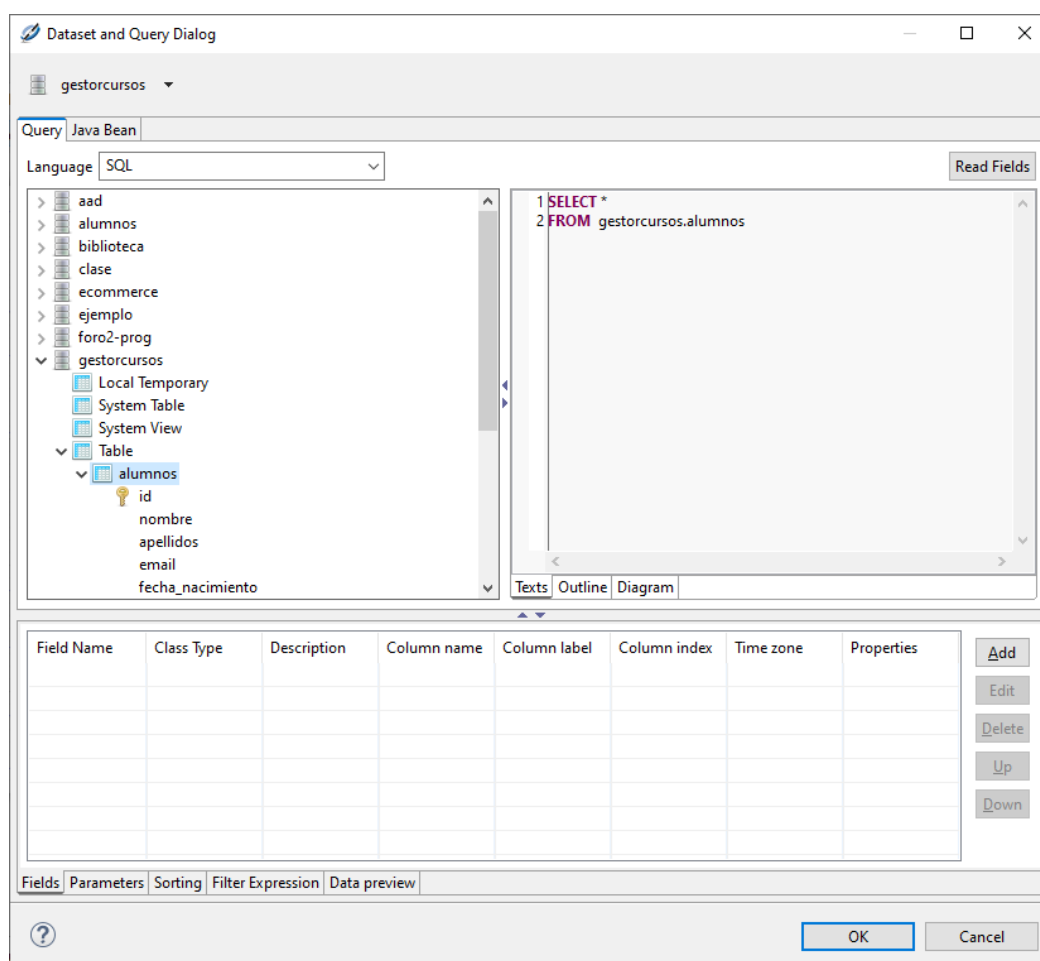
Con JasperSoft tenemos un ayudante para poder definir justamente las consultas que necesitamos para nuestro informe. En primer lugar, abriremos el asistente a través del botón, tal y como marca la imagen:



Fuentes de datos

Fuente: Elaboración propia

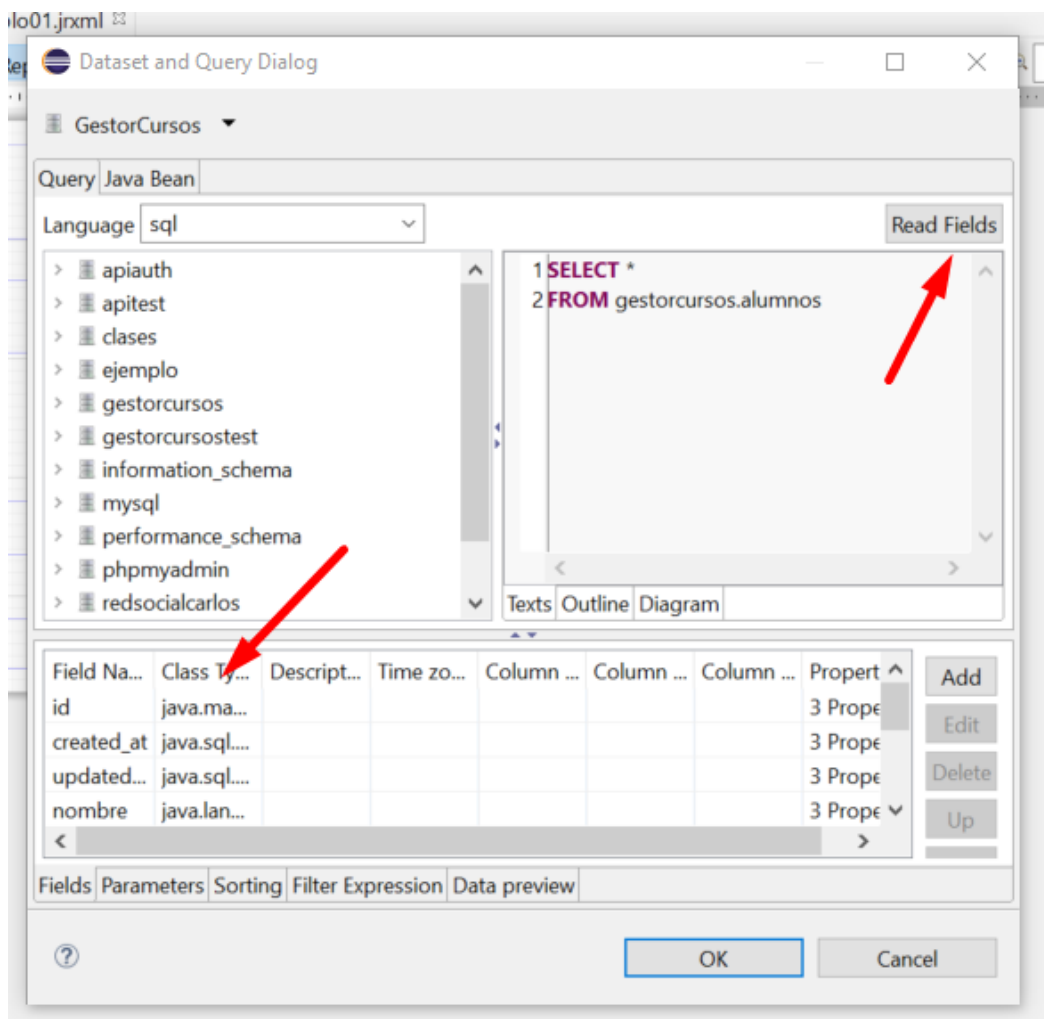
En el editor podemos visualizar las diferentes tablas, campos para formar justamente la expresión que queremos incorporar a nuestro informe:



Selección de los datos y consulta

Fuente: Elaboración propia

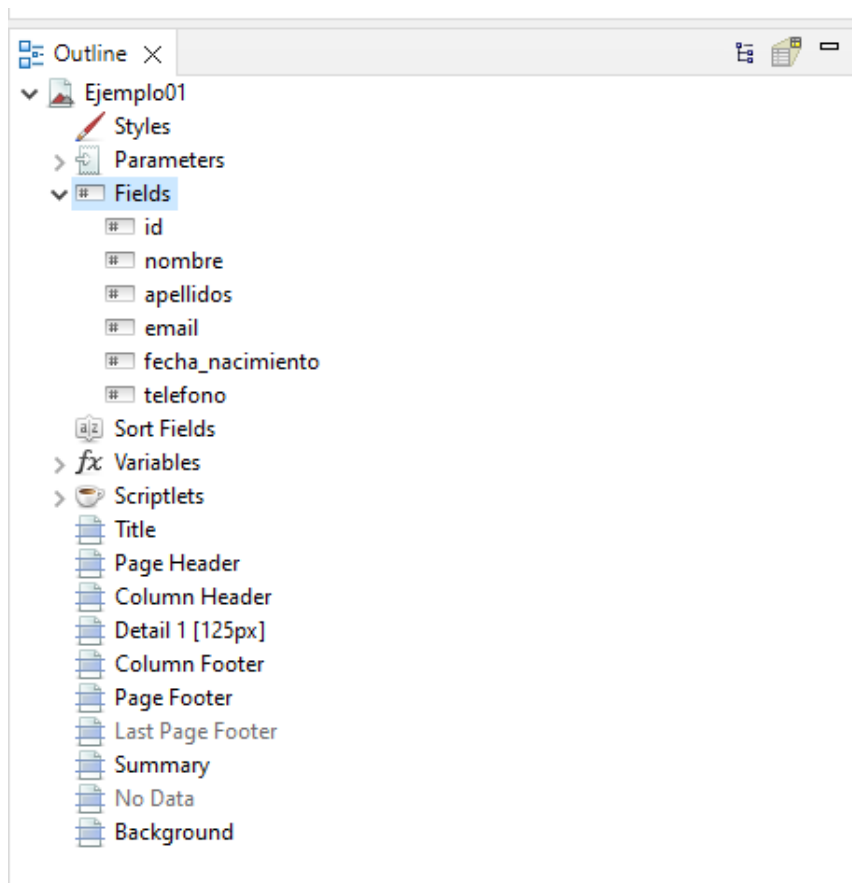
Una vez escrita la consulta apretaremos al botón Read Fields para tenerlos disponibles dentro de nuestro informe.



Carga de datos en documento

Fuente: Elaboración propia

Cuando demos a OK, tendremos disponibles los campos en nuestro informe para ser utilizados.



Datos disponibles

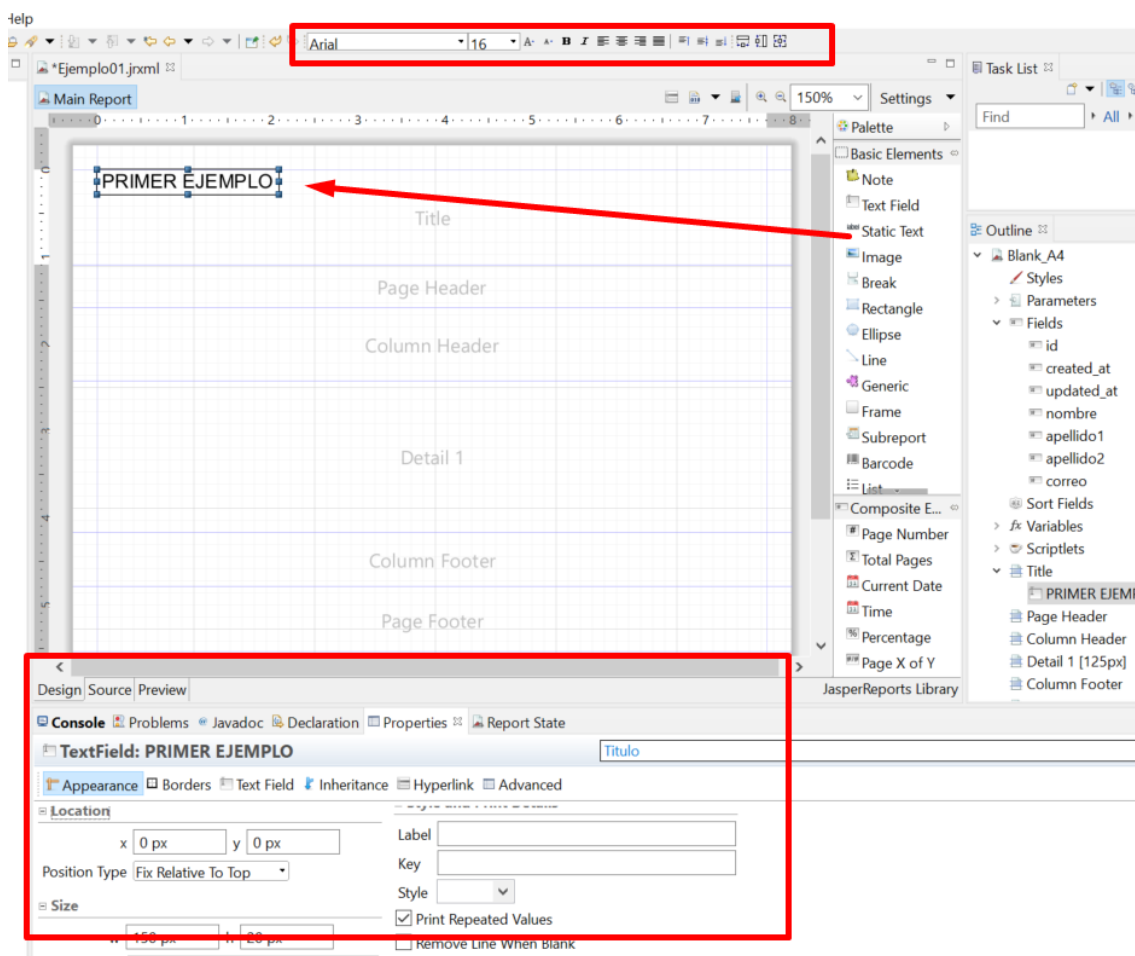
Fuente: Elaboración propia

2.5 Encabezados y pies. Numeración de líneas, recuentos y totales

Tal y como adelantábamos anteriormente, en los encabezados y en los pies incorporaremos normalmente diferentes elementos que nos permiten hacer más usable nuestro informe: títulos, información general, información del lugar y fecha, información del número de página o información del departamento o la persona.

Como podemos observar, esta información no suele ser dependiente de los datos que traemos desde la base de datos, aunque dependerá mucho del tipo y la finalidad del mismo.

Para poner un título en nuestro informe usaremos el elemento “Static Text”, tal y como mostramos en la imagen.



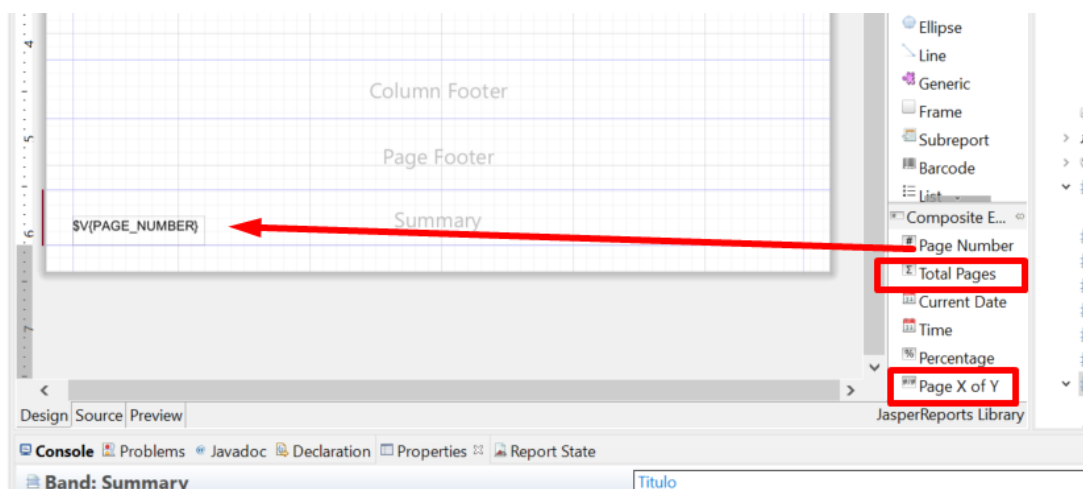
Static Text

Fuente: Elaboración propia

Mediante ese componente podemos incorporar tanto textos fijos como texto a partir de expresiones. Como vemos también en la imagen, tenemos propiedades para:

- Cambiar el tipo de fuente y texto.
- Cambiar las propiedades del componente.

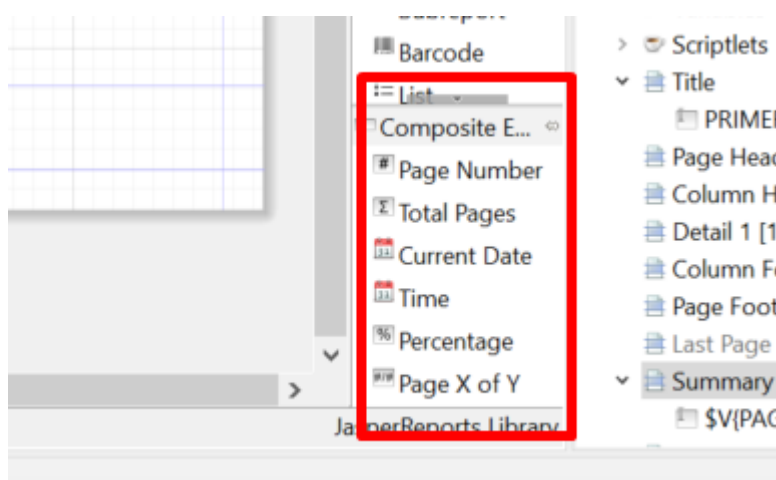
Otros componentes interesantes que podemos incorporar a nuestro informe son los que tenemos referentes al número de páginas y el total de páginas o la página X de Y. Estos datos suelen aparecer en el pie del informe.



Campos sobre el número de páginas

Fuente: Elaboración propia

Otros campos sobre la hora y fecha los podemos observar en el mismo menú sobre los campos que estamos trabajando.



Otros campos sobre Encabezados y Pies

Fuente: Elaboración propia

Como podemos observar en la imagen tenemos todos los campos preparados para poder ser integrado dentro de los informes:

- **Page Number:** corresponde al campo que nos permite introducir la página actual en el documento.
- **Total Page o totales:** corresponde al indicador de totales de páginas que tiene nuestro informe una vez compilado.
- **Current Date:** corresponde con la fecha actual en la que se previsualiza o se compila el documento. Es conveniente realizar un formateo de la misma para poder visualizar correctamente la fecha.

- **Time:** complementario al anterior campo en el que en este caso se introduce la hora y minutos dentro de la previsualización o el compilado del documento.
- **Page X of Y:** corresponde a la representación de la página actual frente al total de páginas.

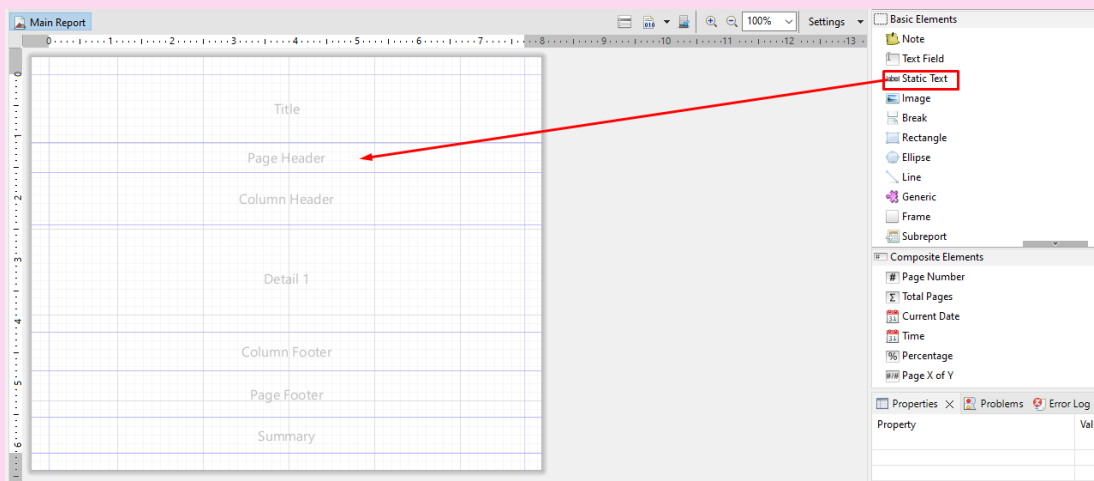
Todos estos campos de forma automática realizan un recuento, una visualización de totales y de las páginas actuales, teniendo en cuenta que dichos campos se calculan en el momento del compilado y previsualización.



EJEMPLO PRÁCTICO

Tu jefe te solicita un informe que agregue encabezados y pies a un informe en JasperSoft, además de incluir numeración de páginas y totales. ¿Cuáles son los pasos que deberías realizar para crear este documento?

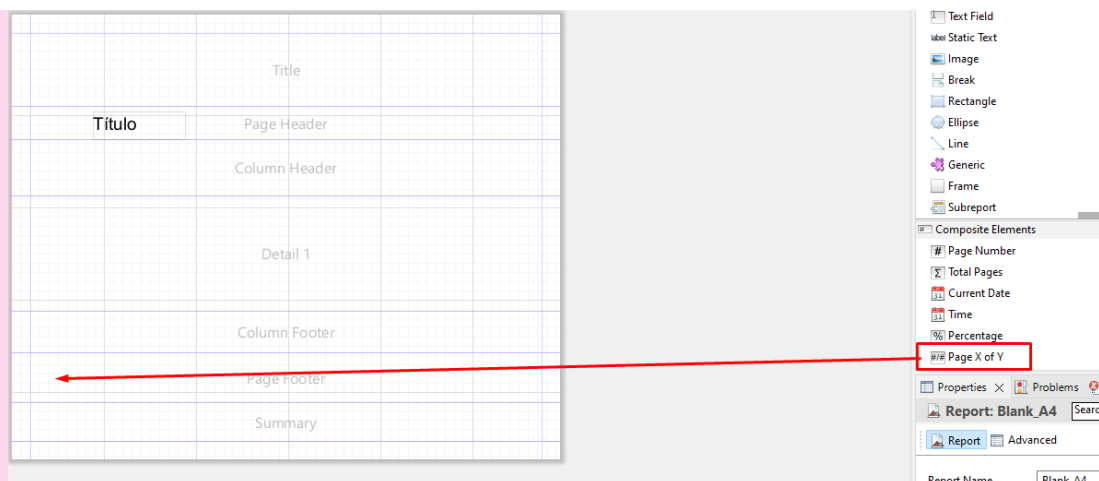
- Creas un informe con JasperSoft.
- A continuación, agregas el encabezado, arrastras un “Static Text” a “Header” y le pones: encabezado.



Encabezado

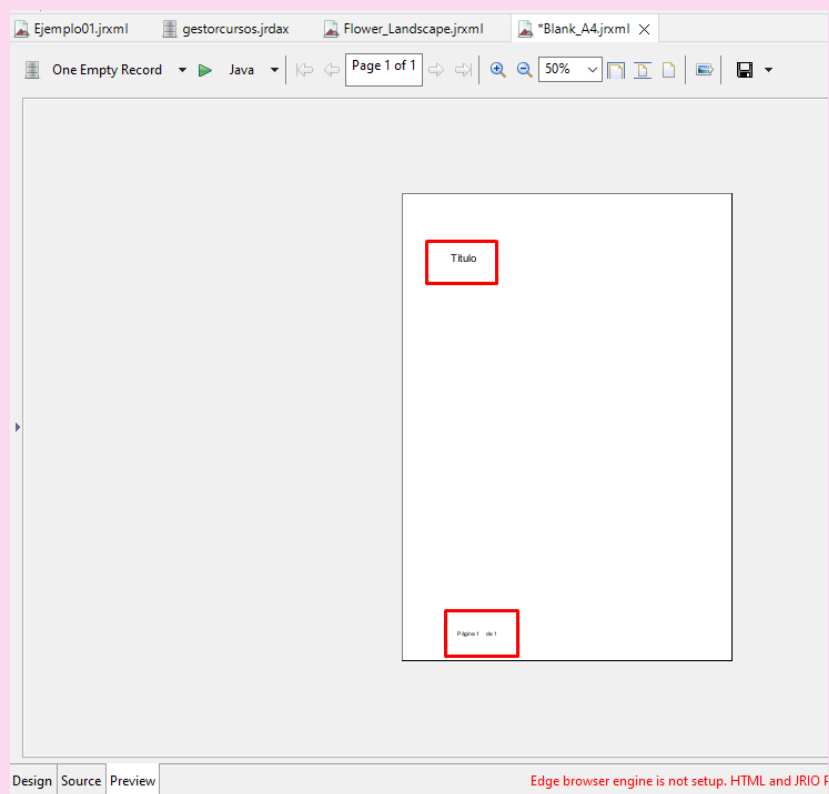
Fuente: Elaboración propia

- Ahora añades el número de página con el total de páginas al pie de página (“Page Footer”). Adapta el texto al español: “Page” por “página” y “of” por “de”.



Página Informe
Fuente: Elaboración propia

- Ejecutas el informe y ves el resultado.

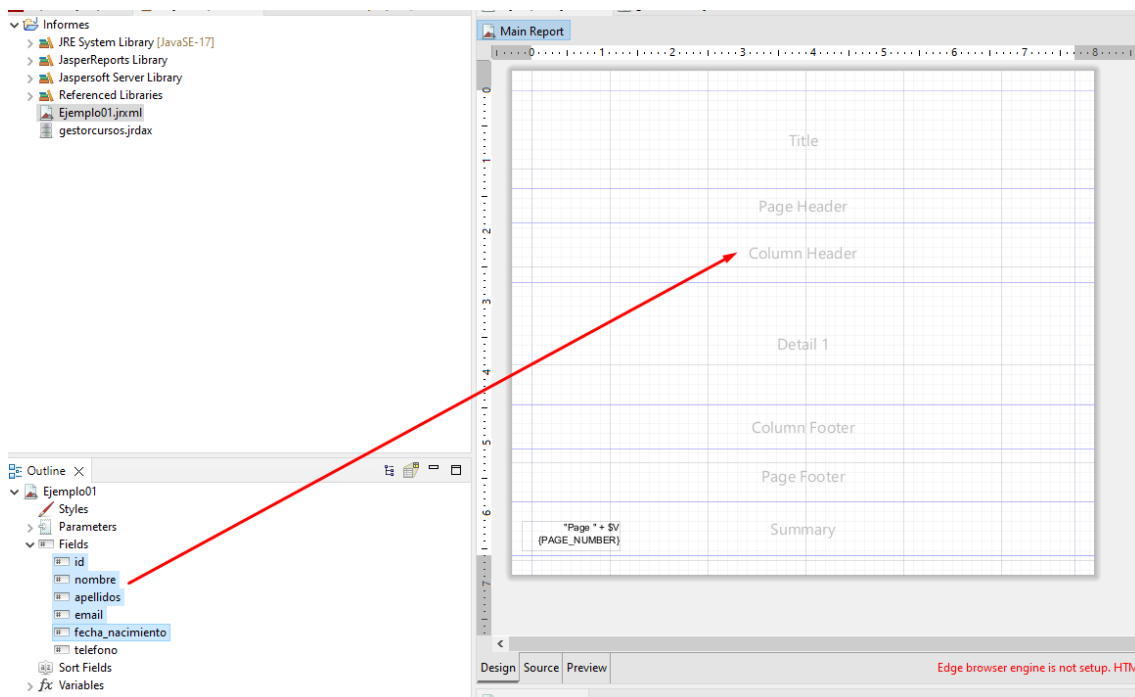


Resultado Informe
Fuente: Elaboración propia

2.6 Valores calculados

Una vez que hemos incorporado a nuestro informe un conjunto de datos desde la base de datos, estos aparecen directamente como objetos arrastrables que podemos incorporar a nuestro documento.

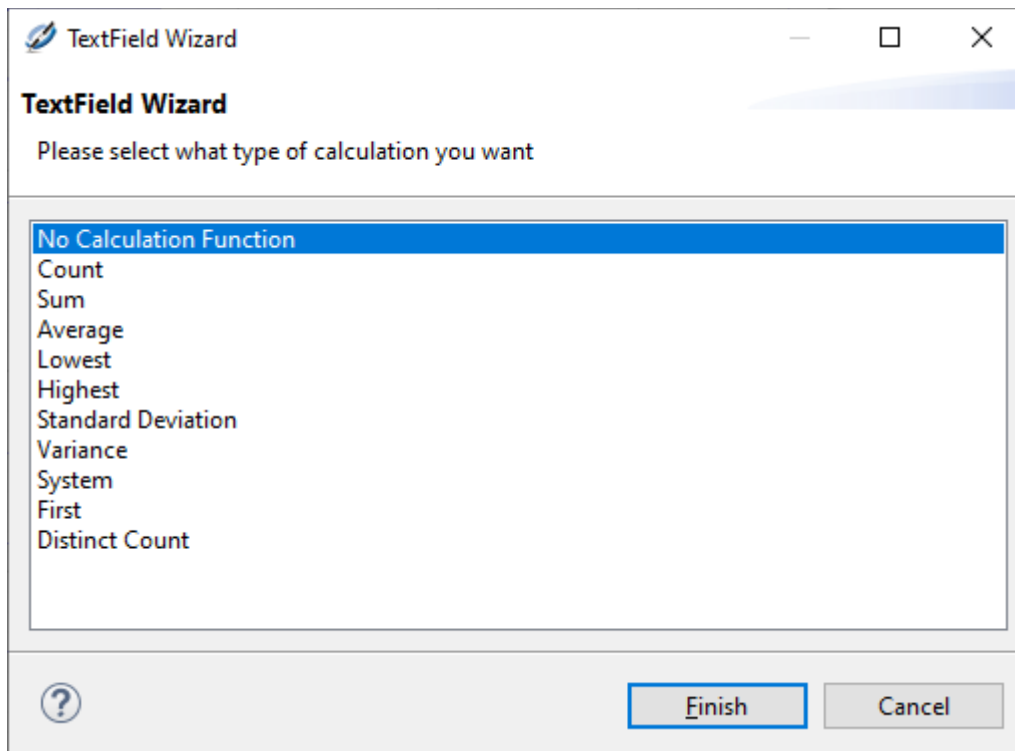
Para ello, elegiremos uno o varios campos y los arrastraremos a nuestra sección denominada Detail.



Campos de la base de datos

Fuente: Elaboración propia

Una vez que los arrastramos nos aparece un ayudante que nos permite realizar cálculos si los necesitamos a partir de estos campos.



Cálculo sobre campos

Fuente: Elaboración propia

Los campos valores calculados que podemos ver en la imagen son uno de los muchos ejemplos que podemos encontrar. Podemos dividirlos en dos tipos de campos calculados:

- Funciones y cálculos a partir de los campos importados de la base de datos. En ese caso, tal y como hemos visto, podemos usar el propio campo arrastrado al informe para después modificar su contenido o expresión de cálculo.
- Valores agrupados, donde tendremos cálculos como la suma, media, varianza, etc. Estos campos necesitan de una preparación previa de la consulta SQL a la base de datos, principalmente proporcionados por los campos GROUP BY.

Después de la incorporación de un campo de este estilo a nuestro informe el resultado puede ser muy parecido al de la incorporación de un *static text*.

Title				
Page Header				
\$F{id}	\$F{nombre}	\$F{apellidos}	\$F{email}	\$F{fecha_nacimiento}
Column Header				
Detail 1				
Column Footer				
Page Footer				
Summary				
<div> "Page " + \$V {PAGE_NUMBER} </div>				

Variables en la zona de detalle

Fuente: Elaboración propia



EJEMPLO PRÁCTICO

El jefe de inventarios solicita que crees un informe en JasperSoft Studio que muestre el valor total del stock de cada producto (precio por cantidad de stock). ¿Cómo lo realizarías?

- Se proporciona un script para poder probar el informe en cualquier base de datos:

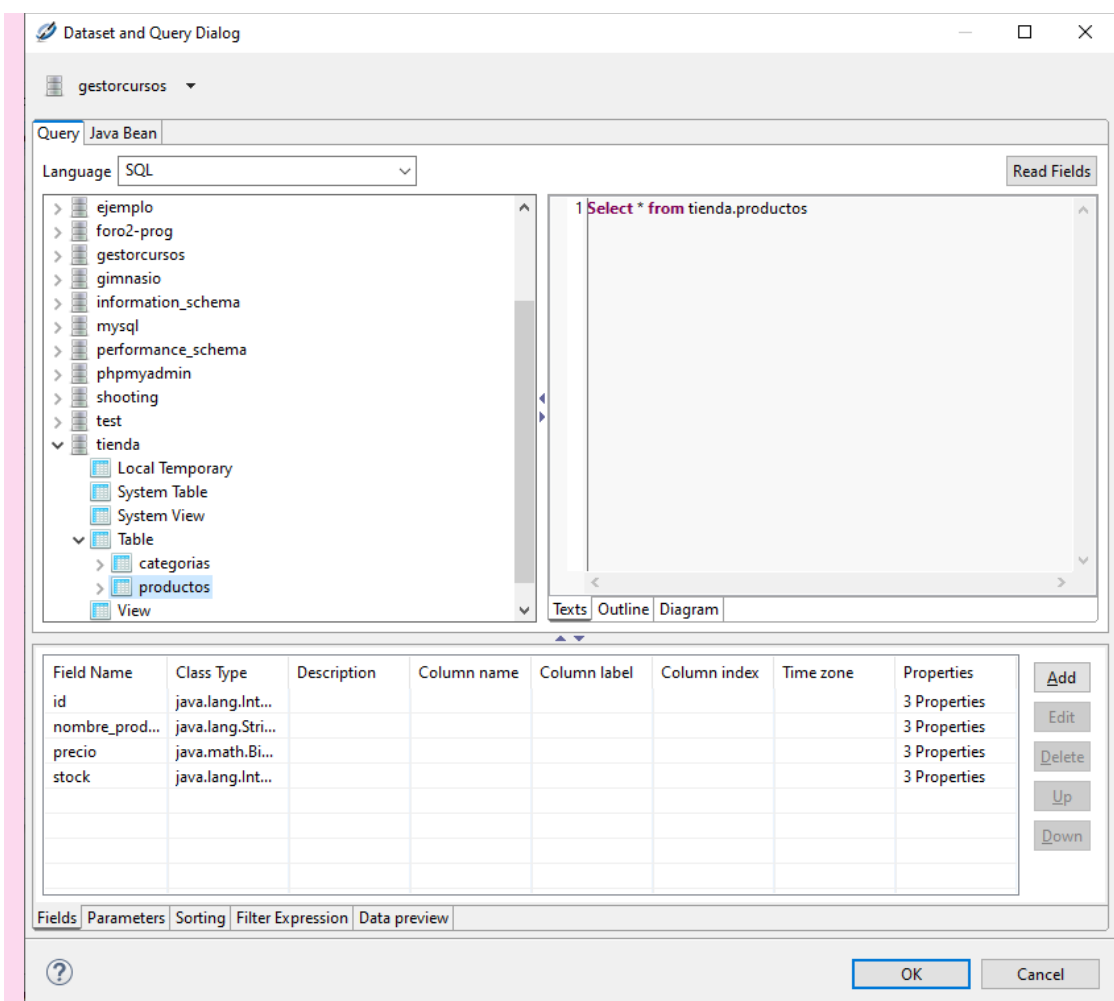
```
-- Crear la base de datos
CREATE DATABASE tienda;

-- Usar la base de datos recién creada
USE tienda;

-- Crear la tabla de productos
CREATE TABLE productos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre_producto VARCHAR(150) NOT NULL,
    precio DECIMAL(10, 2),
    stock INT NOT NULL
);

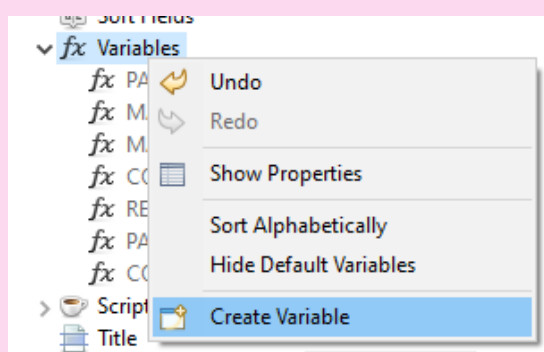
-- Insertar algunos productos con stock
INSERT INTO productos (nombre_producto, precio, stock) VALUES
('Televisor LED 40"', 350.00, 10),
('Smartphone 5G', 600.00, 15),
('Camiseta deportiva', 25.00, 100),
('Pantalón vaquero', 45.00, 50),
('Sofá 3 plazas', 499.99, 5),
('Juego de sábanas', 55.50, 30),
('Muñeca de trapo', 15.00, 200),
('Camión de juguete', 30.00, 150);
```

- En JasperSoft Studio, creas un nuevo informe usando la plantilla en blanco y lo conectas a la base de datos de la tabla productos.



Conexión BBDD
Elaboracion Propia

- Creas el campo calculado mediante la creación de una nueva variable para calcular el precio total del stock: `${precio}.multiply(new java.math.BigDecimal(${F{stock}})`



Properties Problems Error Log

fx Variable: Total Expression

Object Advanced

Name: Total

Value Class Name: java.math.BigDecimal

Description:

Calculation: No Calculation Function

Expression: `$F{precio}.multiply(new`

Initial Value Expression:

Increment type:

Incrementer Factory Class Name:

Reset type:

Expression Editor (java)

Expression Editor

Edit the JasperReports expression that returns the expected type.

`$F{precio}.multiply(new java.math.BigDecimal($F{stock}))`

Parameters

Fields

Variables

Resources

Built-in Functions

- Date & Time
- Logical
- Message Bundles
- Numeric / Mathematical
- Report
- Text
- User Defined Expressions
- Recent Expressions

type filter text

id Field Integer

nombre_producto Field String

precio Field BigDecimal

stock Field Integer

type filter text

bitCount(int) int

byteValue() byte

compare(int, int) int

compareTo(Integer) int

compareTo(Object) int

compareUnsigned(int, int) int

decode(String) Integer

describeConstable() java.util.Optional

divideUnsigned(int, int) int

doubleValue() double

equals(Object) boolean

floatValue() float

getClass() Class

getInteger(String) Integer

getInteger(String, Integer) Integer

getInteger(String, int) Integer

hashCode(int) int

hashCode() int

highestOneBit(int) int

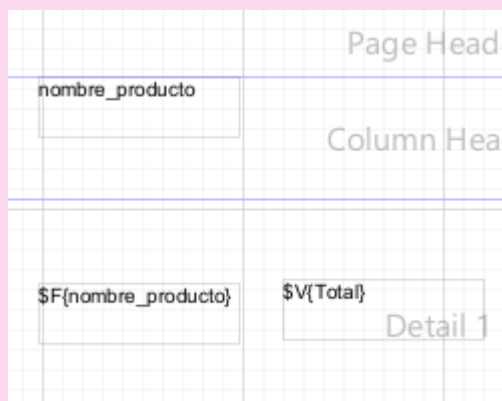
intValue() int

longValue() long

Finish Cancel

Creación de Variable
Fuente: Elaboración Propia

- Por último, creas el informe que con los datos.



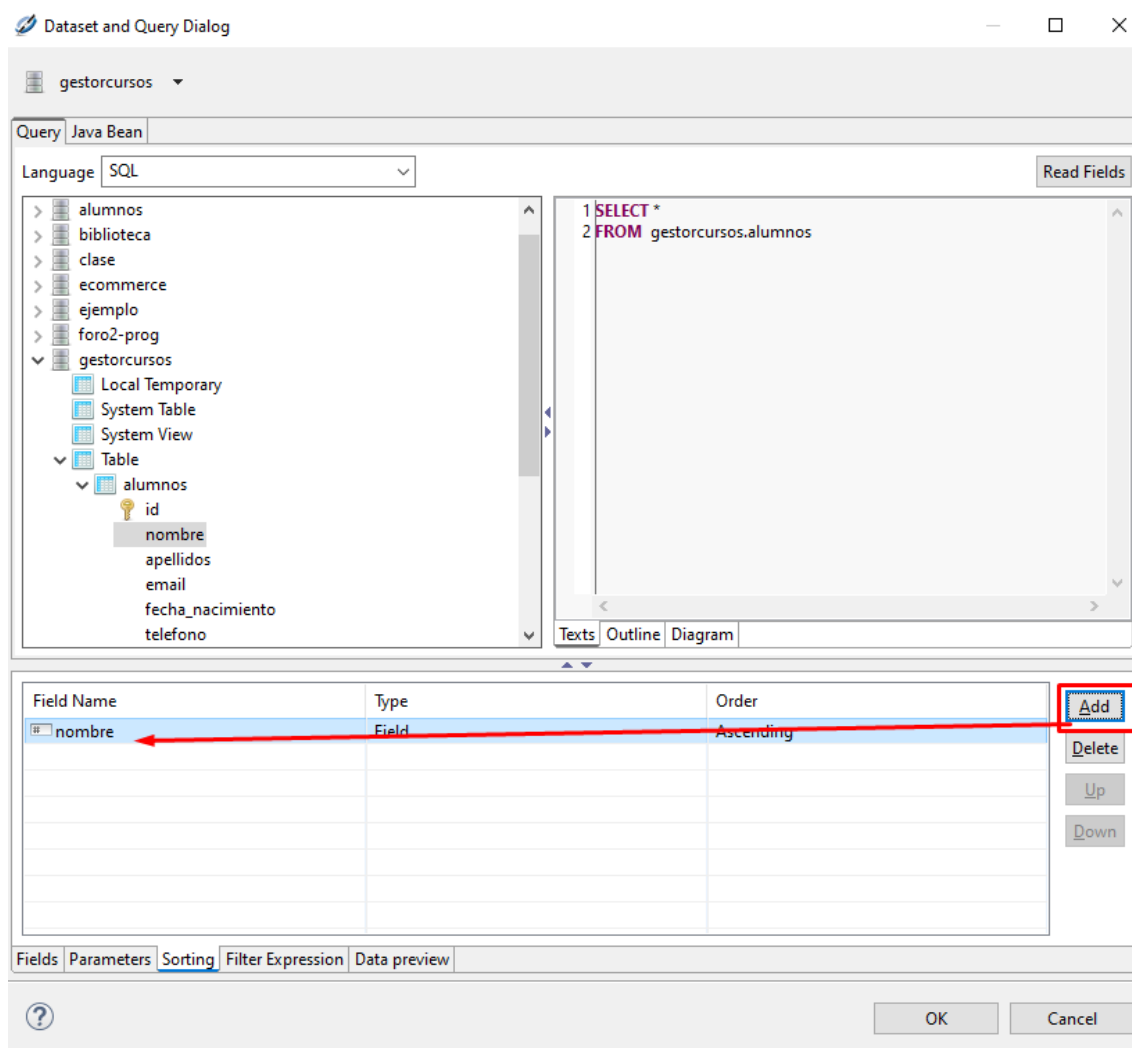
nombre_producto	
Televisor LED 40"	3500.00
Smartphone 5G	9000.00
Camiseta deportiva	2500.00
Pantalón vaquero	2250.00

Informe con la variable nueva
Fuente: Elaboración Propia

2.7 Filtrado de datos

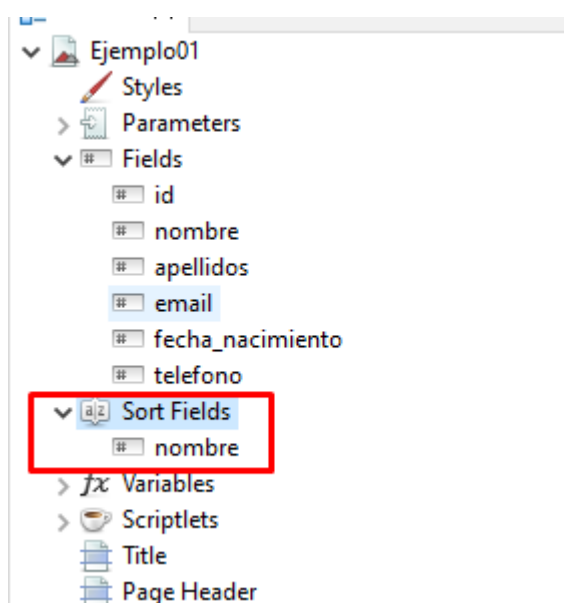
Para el filtrado debemos usar nuevamente el menú de acceso a datos donde disponemos justamente de la posibilidad de incorporar parámetros de filtrado.

Al hacer click sobre el botón Add, podemos añadir el campo o el conjunto de campos sobre los que queremos realizar un filtrado. Apareciendo la selección sobre el origen de datos y pudiendo elegir cómo interactúa con los datos.



Selección de campos

Fuente: Elaboración propia



Creación de filtro de datos

Fuente: Elaboración propia



ENLACE DE INTERÉS

Accede a esta página web, cuyo contenido complementa la documentación sobre la creación de un informe y su uso:



EJEMPLO PRÁCTICO

Tu jefe te ha solicitado un informe que muestre solo los productos cuyo precio sea superior a 100. Debes utilizar un filtro en JasperSoft Studio para obtener este resultado. ¿Cómo lo realizarías?

- Puedes utilizar este Script para probar este informe:

```
-- Crear la base de datos
CREATE DATABASE tienda;

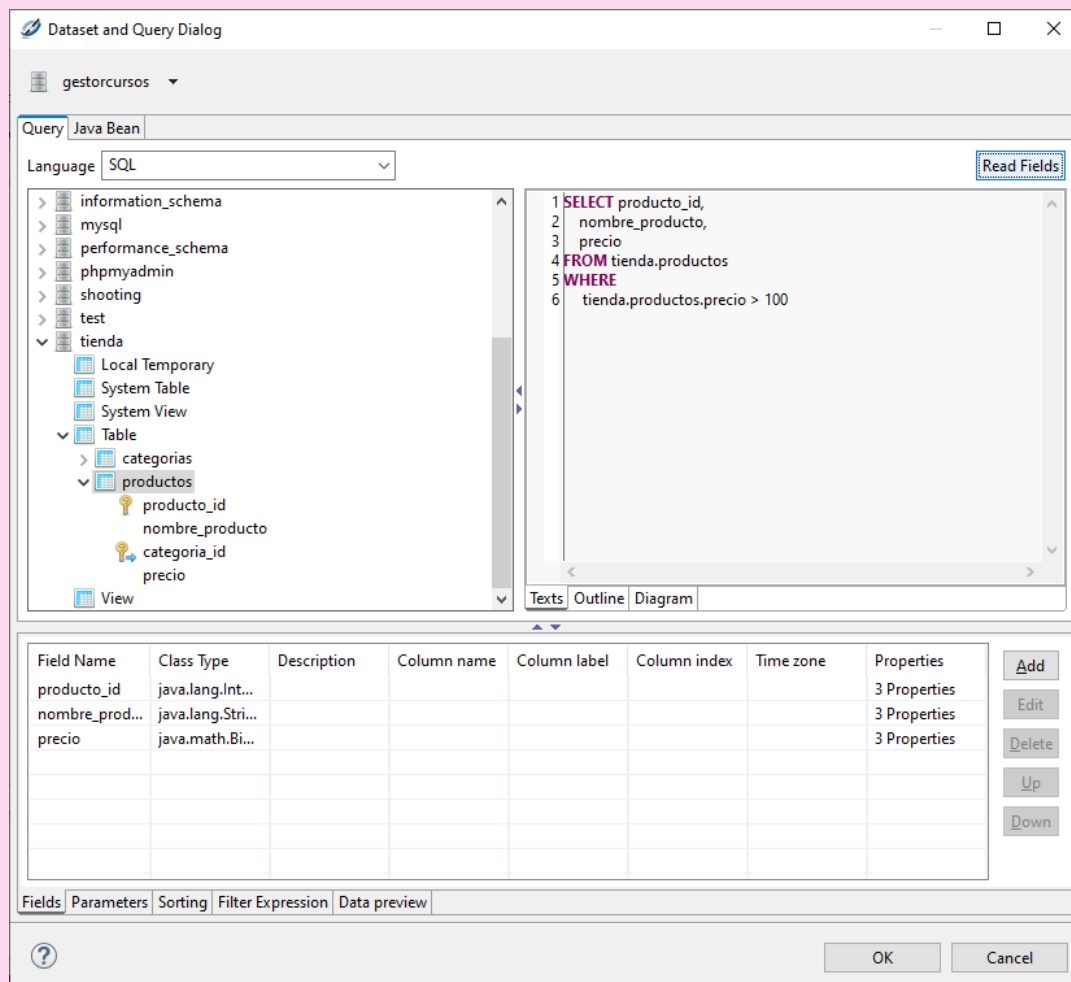
-- Usar la base de datos recién creada
USE tienda;

-- Crear la tabla de productos
CREATE TABLE productos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre_producto VARCHAR(150) NOT NULL,
    precio DECIMAL(10, 2),
    stock INT NOT NULL
);

-- Insertar algunos productos con stock
INSERT INTO productos (nombre_producto, precio, stock) VALUES
('Televisor LED 40"', 350.00, 10),
('Smartphone 5G', 600.00, 15),
('Camiseta deportiva', 25.00, 100),
('Pantalón vaquero', 45.00, 50),
('Sofá 3 plazas', 499.99, 5),
('Juego de sábanas', 55.50, 30),
('Muñeca de trapo', 15.00, 200),
('Camión de juguete', 30.00, 150);
```

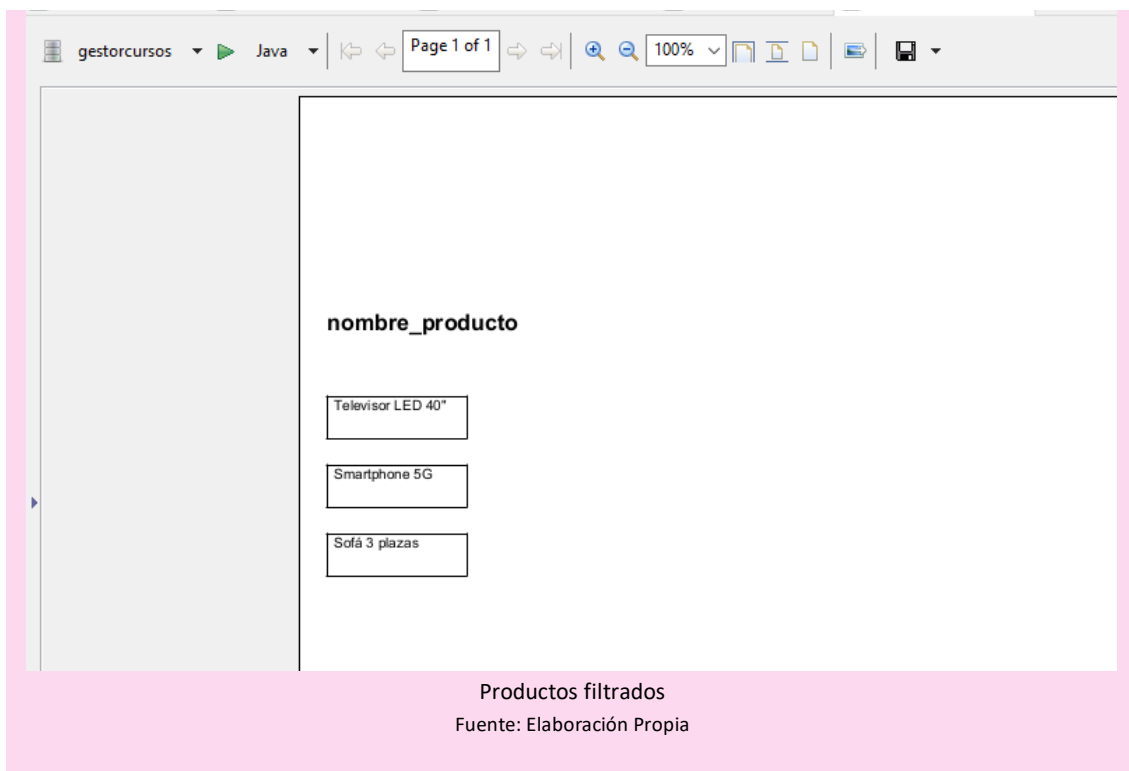

- En JasperSoft Studio, creas un nuevo informe usando la plantilla en blanco.
- Una vez creado el informe, en el apartado de la fuente de datos, selecciona la opción de escribir una consulta SQL (“Dataset and Query”) e introduce la siguiente consulta:

```
SELECT producto_id,nombre_producto,precio
FROM tienda.productos
WHERE tienda.productos.precio > 100
```



Consulta Filtro
Fuente: Elaboración Propia

- Arrastras los campos al informe y verás que solo se cargarán los productos cuyo precio es mayor que 100.

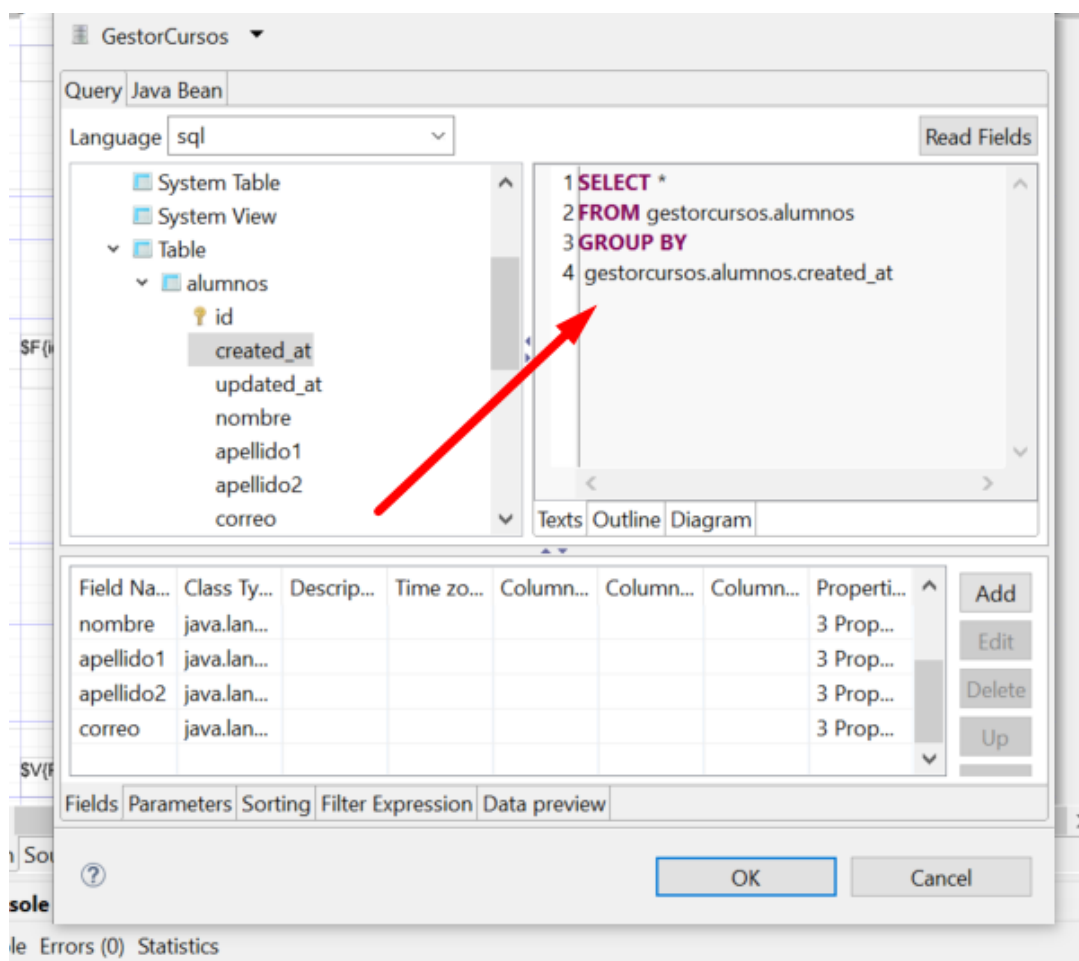


2.8 Informes con agrupamiento, recuentos parciales y subtotales

Existen tres mecanismos para poder realizar agrupamiento, recuentos y subtotales:

- A través de la definición de los datos.
- A través del cálculo de una nueva expresión o variable.
- A través del cálculo a partir del campo.

En el primer caso usaríamos la pantalla de origen de datos que hemos estado usando en otras ocasiones. En este caso usaremos directamente el SQL para poder realizar esas acciones, tanto de cálculo como de agrupaciones. Tal y como vemos en la imagen, hemos definido la agrupación mediante el campo `created_at`.

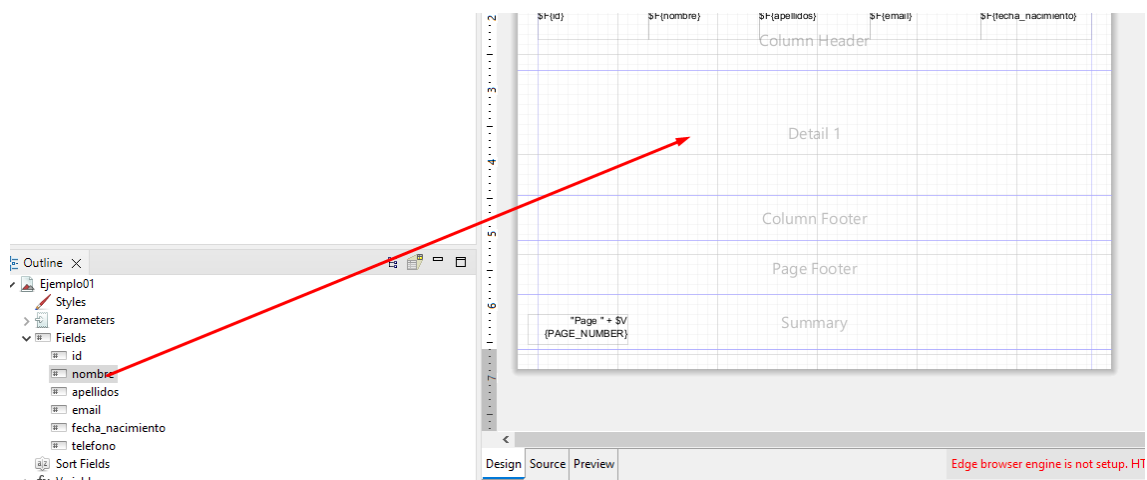


Agrupaciones por SQL

Fuente: Elaboración propia

En una segunda opción podemos usar el cálculo de variables e incorporarlas al informe. Por ejemplo, si incluimos un nuevo Text Field en el Page Footer donde se calcule el número de registros que hay de un determinado elemento.

Esta misma acción lo podemos realizar de una forma más automática cuando arrastramos un determinado campo a las secciones normalmente en el área de resumen como vemos en la imagen.



Cálculo en informe

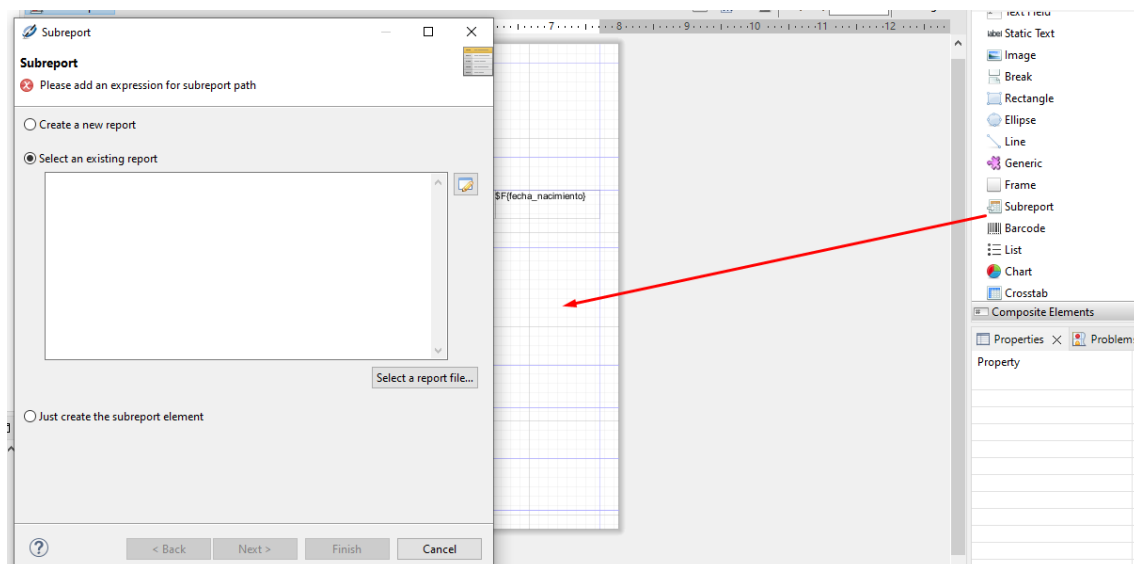
Fuente: Elaboración propia

Como vemos, al arrastrar el campo de la base de datos, nos aparece el menú de posibilidades para poder realizar cálculos agregados. En este caso se trata de un campo tipo VARCHAR, por lo que el único cálculo será el de un conteo del número de elementos, bien por página o bien total dependiendo de dónde coloquemos el campo.

2.9 Sub-Informes

Los sub-Informes permiten la creación de un nuevo informe dentro de otro. El motivo de la realización de sub-Informes es principalmente el de añadir un detalle una vez que se ha extraído información de una determinada fila. Imaginemos que tenemos una base de datos de recursos humanos y tenemos los datos de nuestros empleados y, a su vez, querríamos conocer el número de horas por semanas de un determinado empleado. Un mecanismo sería el uso de los sub-Informes.

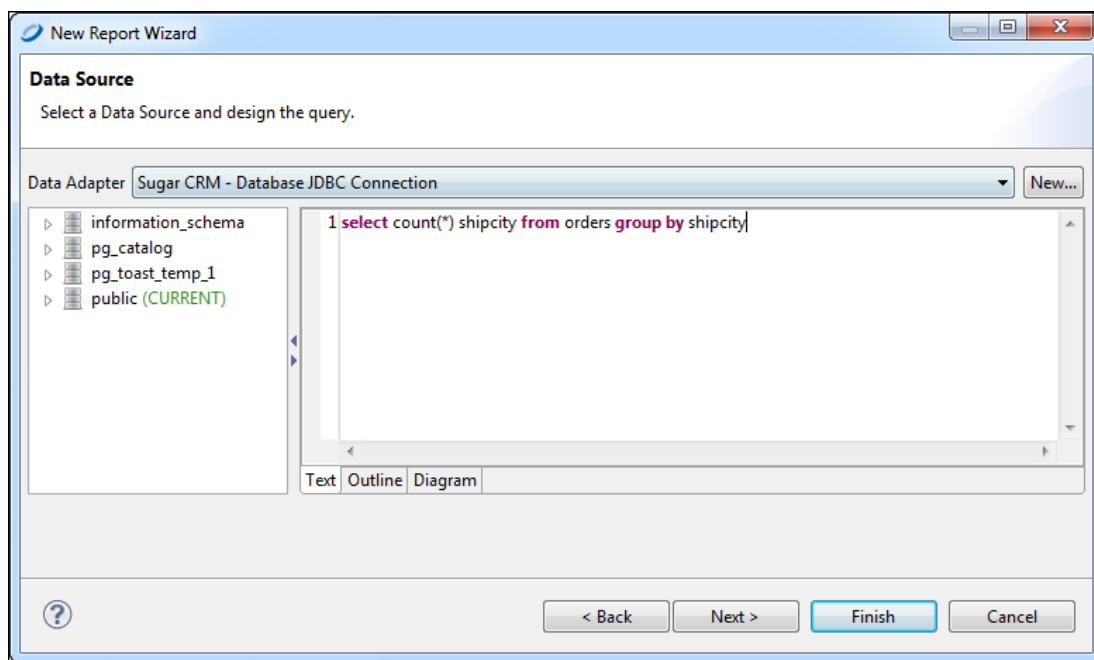
Dentro de JasperSoft tenemos el asistente de Subreport al arrastrar el elemento a la zona de detalle, tal y como vemos en la imagen.



Subreport

Fuente: Elaboración propia

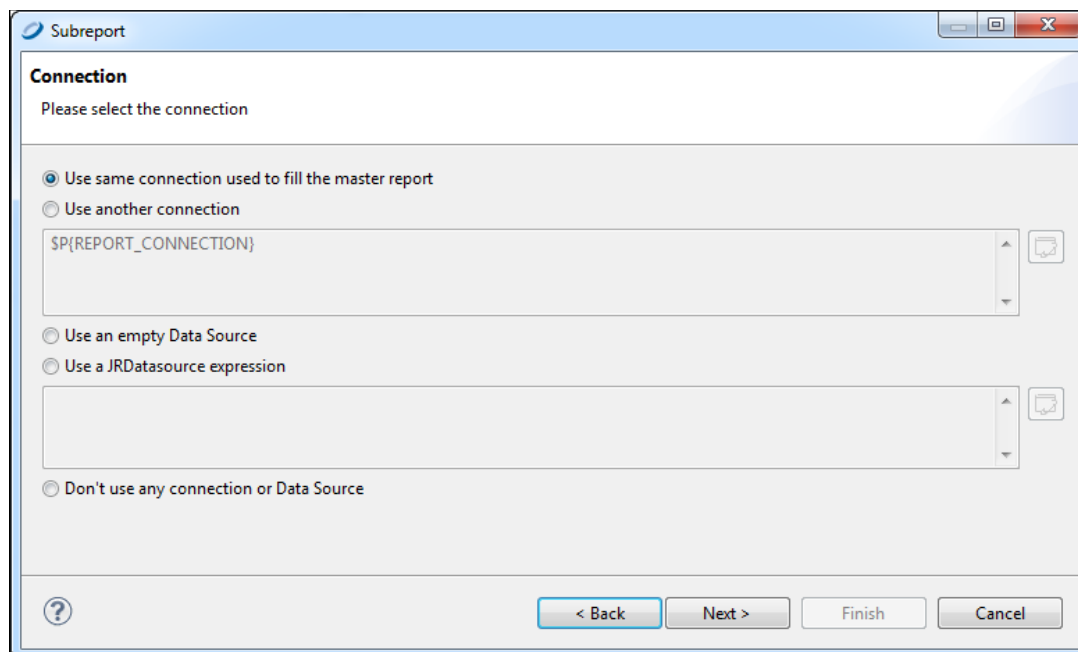
Una vez abierto elegiremos un nuevo informe y se nos comenzará a abrir todos los pasos de creación de un nuevo informe. La pantalla de Data Source será imprescindible para poder realizar una selección correcta de la información a presentar.



Fuente de la información en el subreport

Fuente: <https://community.jaspersoft.com/documentation/tibco-jaspersoft-studio-user-guide/v640/creating-new-report-subreport-wizard>

Eligiendo el mismo adaptador que teníamos en el informe origen, tendremos por lo tanto ligado todos los datos del informe y del sub-Informe.



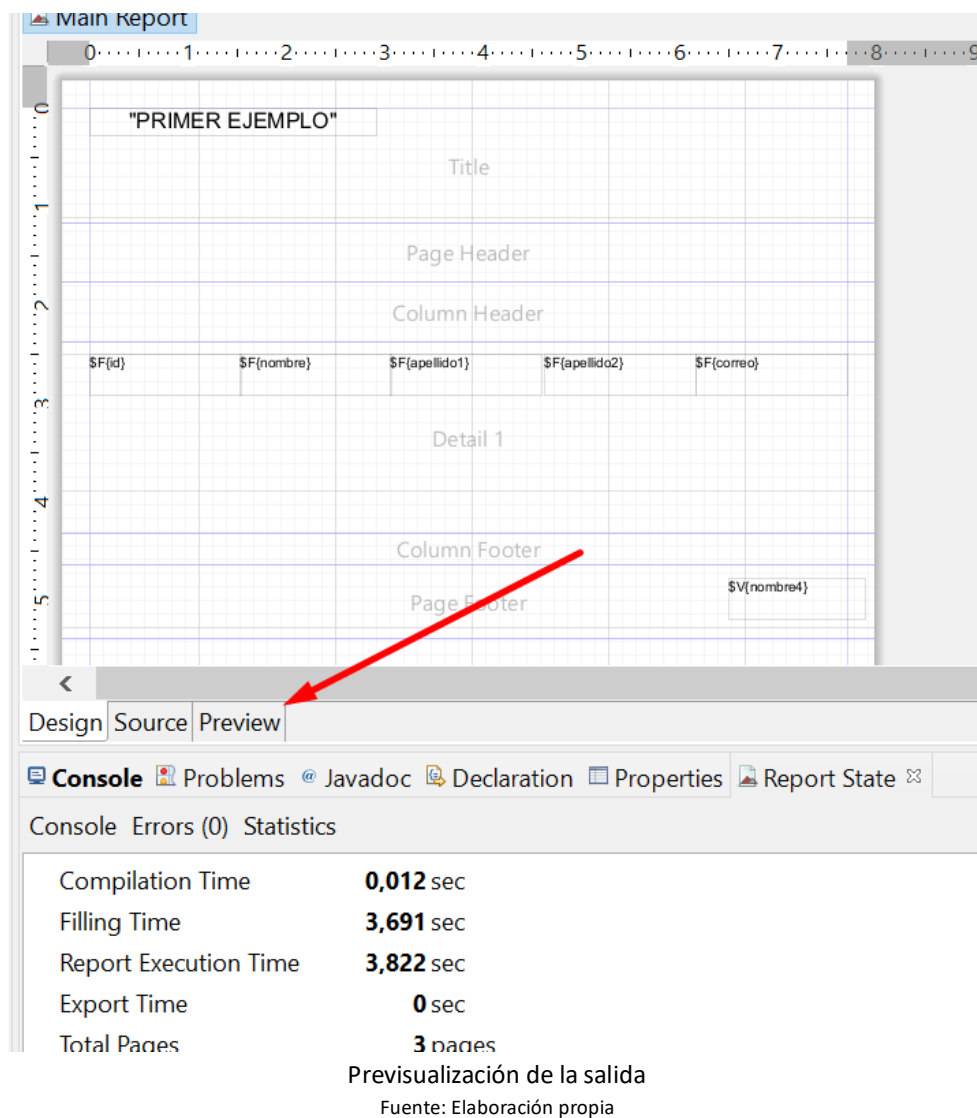
Conector

Fuente: <https://community.jaspersoft.com/documentation/tibco-jaspersoft-studio-user-guide/v640/creating-new-report-subreport-wizard>

2.10 Formatos de salida

Una vez que tenemos diseñado un informe dentro de nuestro paquete JasperSoft y Eclipse debemos o podemos realizar un proceso de previsualización o de compilación.

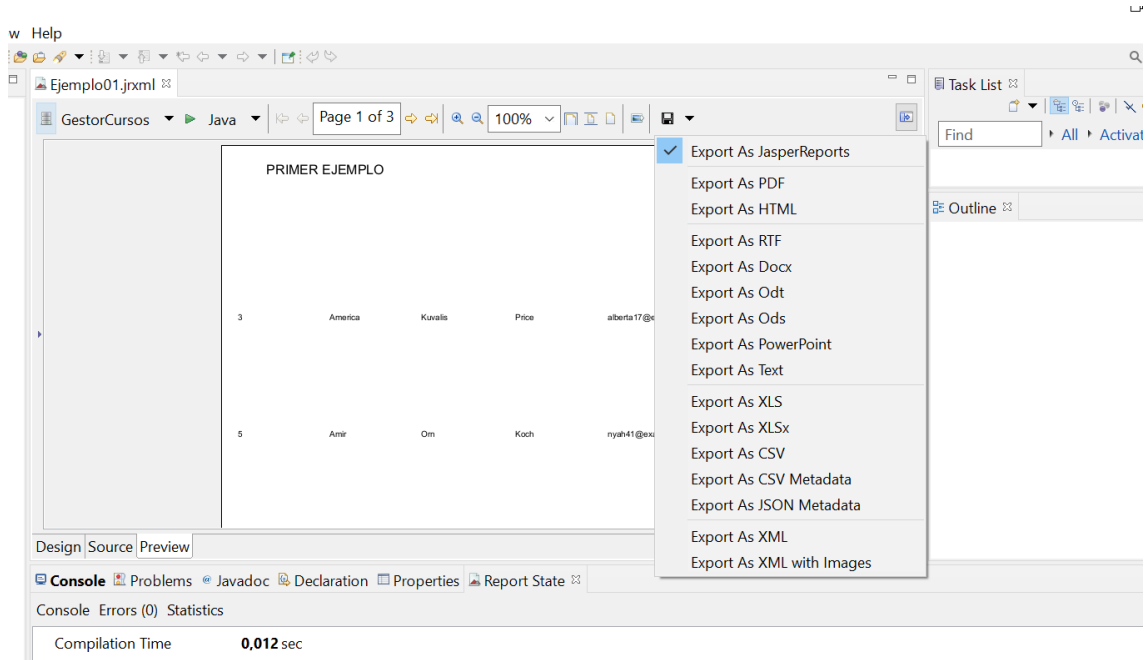
En este proceso, el diseño realizado recoge la información de la base de datos, aplica el diseño de nuestro informe y forma el informe final.



Una vez que ya tenemos compilado el informe tenemos la opción de salvar nuestro informe de dos formas:

- A. Exportar para poder ser visualizado externamente por cualquier usuario sin el paquete de JasperSoft.
- B. Guardar como un informe propio de JasperSoft para su posterior modificación.

En la imagen podemos ver los formatos que permite Jaspersoft para su exportación que son muchos:



Exportación de informe

Fuente: Elaboración propia

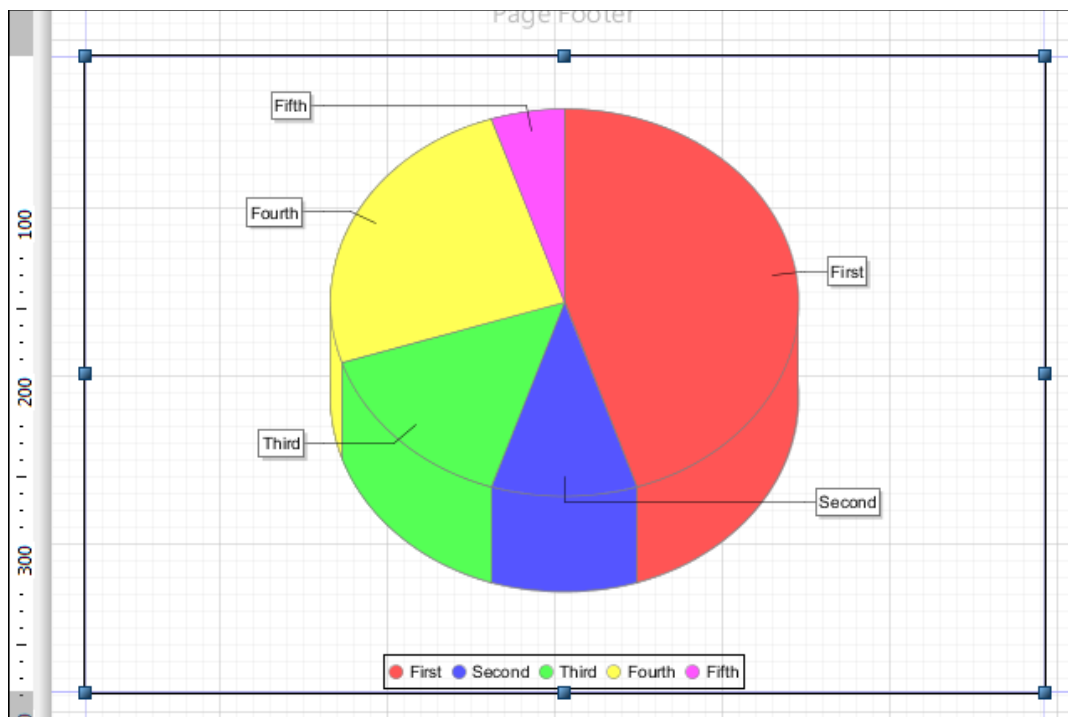
Como podemos observar, tenemos desde formatos totalmente compilados sin posibilidad de poder trabajar después la información a formatos donde posteriormente podemos interactuar con ese “preparado” de datos como Excel, CSV, etc.

2.11 Imágenes. Gráficos

Dentro de un informe nos encontramos diferentes tipos de elementos multimedia que nos ayudan a representar la información de una forma mucho más gráfica:

- Las **imágenes** nos ayudan a hacer más atractiva la presentación. Suelen ser elementos estáticos y se utilizan para logos, imágenes de productos, de usuarios, etc.
- Los **gráficos** nos permiten presentar la información de una forma más dinámica. En concreto, dentro de JasperSoft Studio, nos encontramos con múltiples posibilidades: tarta, categoría, tiempo, cy, valor, etc.

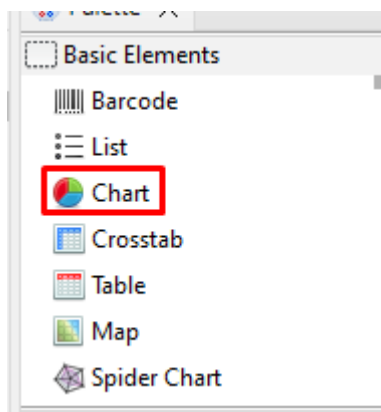
Es en este último apartado donde tenemos una mayor complejidad al mismo tiempo que nos proporciona la posibilidad de interacción con los datos.



Ejemplo de gráfico de tarta

Fuente: https://community.jaspersoft.com/documentation/tibco-jaspersoft-studio-user-guide/v640/creating-simple-chart#jss-user_charts_505037509_1028827

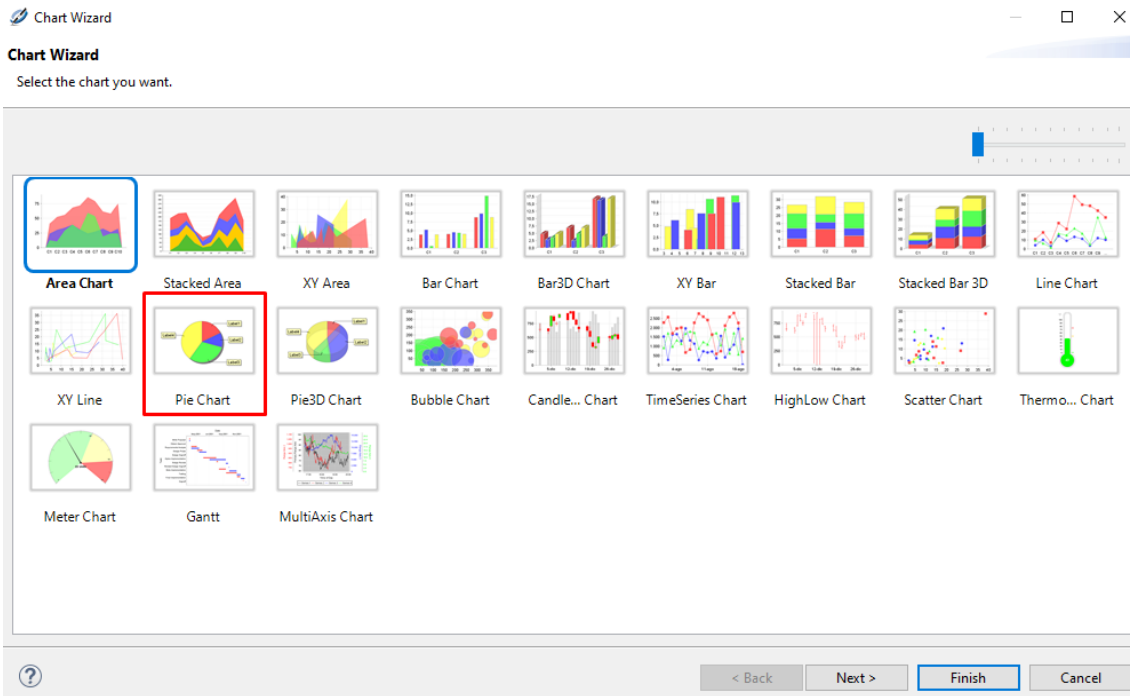
Para añadir el gráfico en la ventana de elementos tenemos el ítem “Chart” que lo arrastraremos al informe como cualquier otro elemento.



Elemento Gráfico

Fuente: Elaboración propia

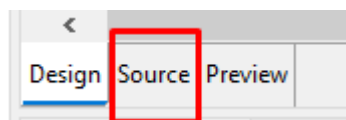
Luego nos saldrá un asistente donde seleccionaremos el tipo de gráfico, en este caso seleccionaremos el gráfico de tarta.



Gráficos Predefinidos

Fuente: Elaboración propia

Si pulsamos en “Next” y la pantalla no reacciona, es que tenemos un problema muy común con el asistente de gráficos de algunas versiones de JasperSoft. Para solucionarlo pulsaremos en “Finish”, acto seguido iremos a la pestaña de “Source”.



Pestaña Source

Fuente: Elaboración propia

En la etiqueta dataset incluiremos el siguiente texto:

```
resetGroup="DEPARTMENT_NAME" resetType="Group" incrementType="None"
```

Acto seguido guardaremos.

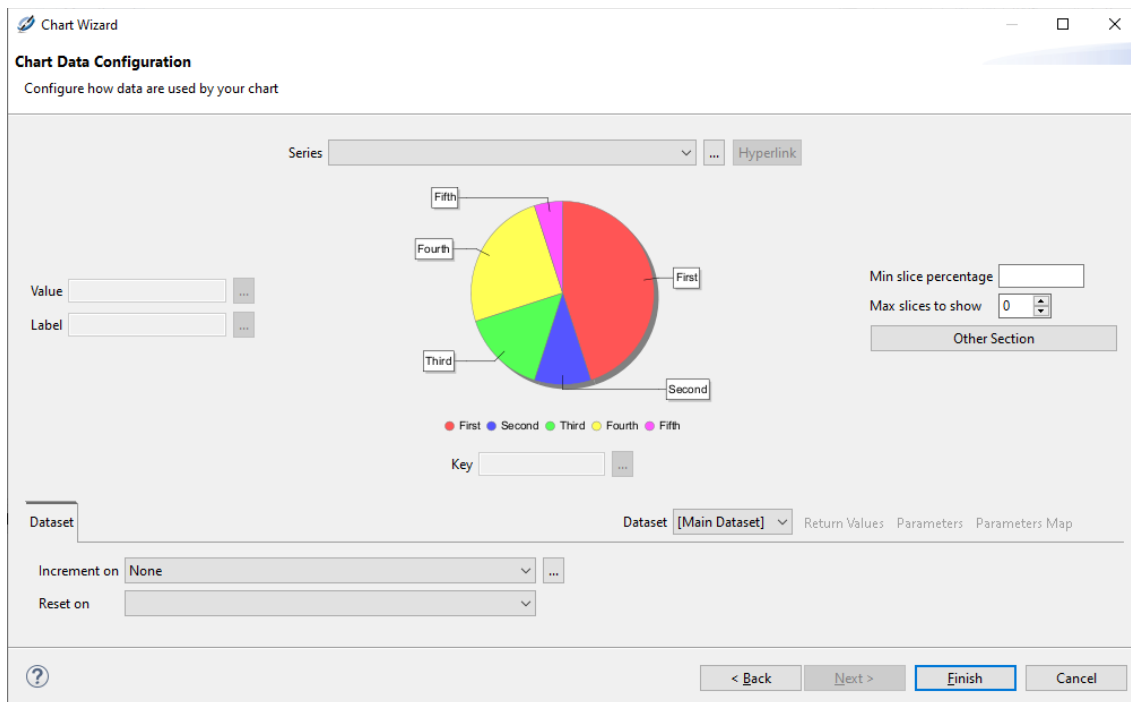
```
1<!-- Created with Jaspersoft Studio version 7.0.0.final using JasperReports Library version 7.0.0-b478feaa9aab4375eba71de77b4ca138ad2f6
2<jasperReport name="Ejemplo01" language="java" pageWidth="595" pageHeight="842" columnWidth="555" leftMargin="20" rightMargin="20" top
3<query language="sql"><![CDATA[]]></query>
4<title height="79" splitType="Stretch"/>
5<pageHeader height="35" splitType="Stretch"/>
6<columnHeader height="242" splitType="Stretch">
7<element kind="chart" chartType="pie" uuid="e9669f53-f2c6-4311-bf35-2b1083cf5e7f" x="224" y="42" width="200" height="200" eval
8<dataset kind="pie"/>
9<plot>
10<itemLabel/>
11</plot>
12</element>
13</columnHeader>
14<detail>
15<band height="245" splitType="Stretch"/>
16</detail>
17</jasperReport>
```

```
<dataset kind="pie" resetGroup="DEPARTMENT_NAME" resetType="Group"/>
```

Solución Problema Gráficos

Fuente: Elaboración propia

Volveremos a la pestaña “Design” y haremos doble click sobre el gráfico. Ya se nos debería abrir el asistente sin problemas.



Asistente Gráficos

Fuente: Elaboración propia



ENLACE DE INTERÉS

Para entrar en la profundidad que se merece, realizar un detalle y comprender todas las posibilidades de los posibles formatos, es recomendable visitar el apartado específico dedicado a los gráficos:





PARA SABER MÁS

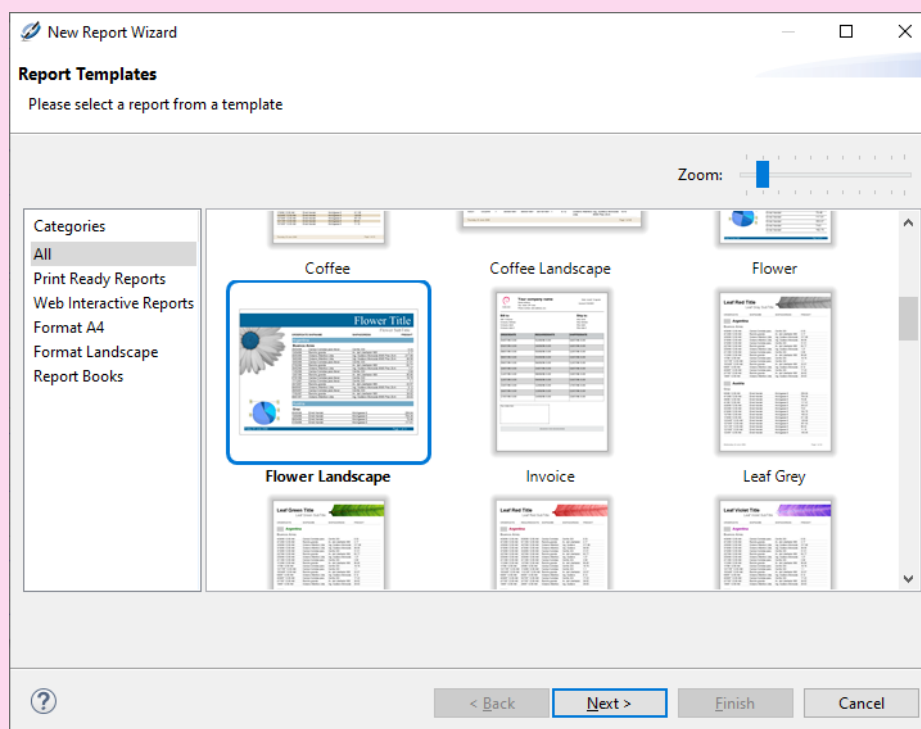
Aquí podrás acceder a la documentación en castellano de versiones anteriores:



EJEMPLO PRÁCTICO

El jefe de ventas necesita tener un informe muy visual, a poder ser usando gráficos para representar las ventas por tienda de forma mensual. Para ello, tu jefe te plantea atacar a la base de datos, extraer un resumen de los datos de forma agrupada y un gráfico de tarta, todo ello en una sola página. ¿Cuáles son los pasos que deberíamos realizar para crear este documento?

- Creas un nuevo informe tomando como plantilla “Flower Landscape”, ya que proporciona justamente la incorporación de una tabla de datos y gráficos.

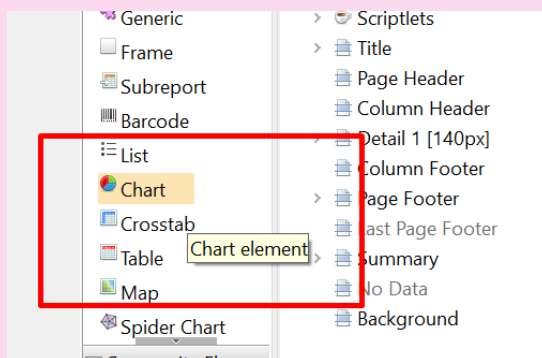


Ejemplo de gráfico de tarta
Fuente: Elaboración propia

- A partir de esa plantilla, sigues los pasos realizando la siguiente consulta a la base de datos:

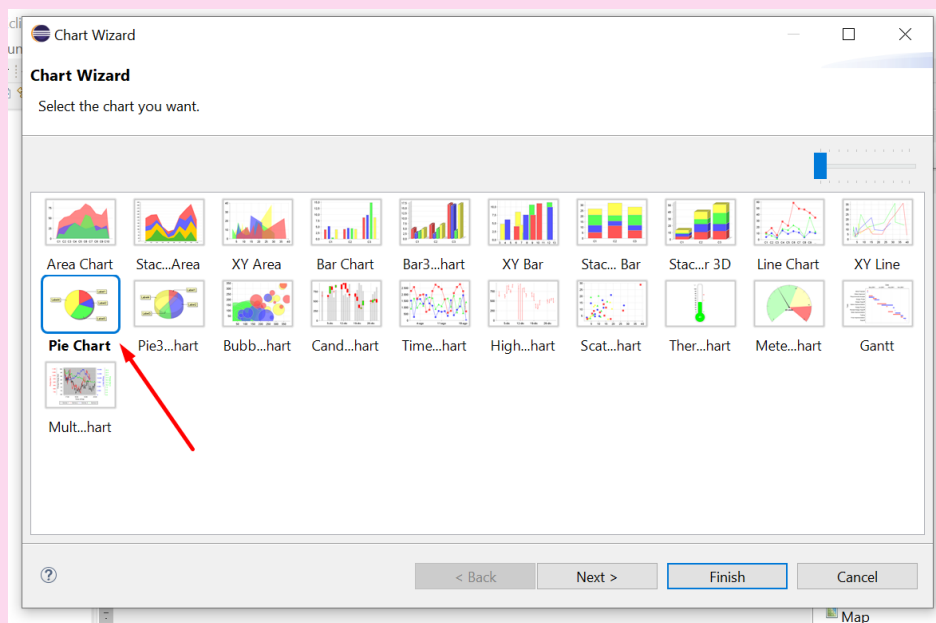
SQL SELECT nombreTienda, provincia, ventas, fechaVenta **FROM** Ventas **GROUP BY** nombreTienda

- Teniendo esa información, realizas una presentación de datos mediante ese agrupado, filtrando por tiendas.
- Para incluir un nuevo gráfico, eliges el elemento Chart de la paleta de elementos.



Elemento gráfico tartas

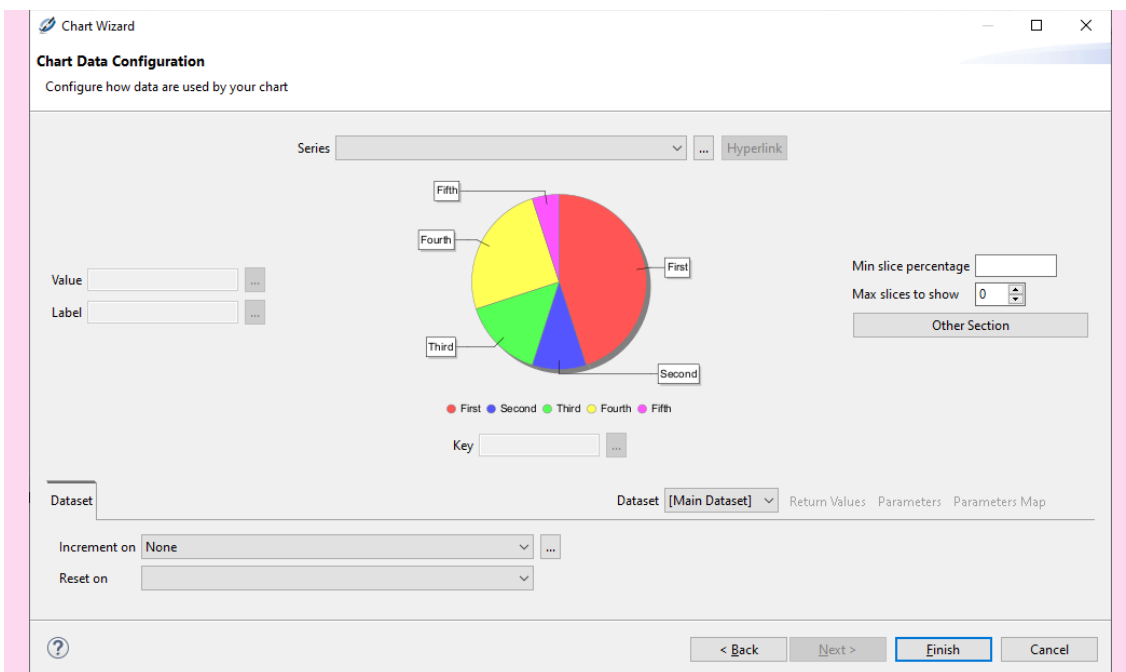
Fuente: Elaboración propia



Tipo tartas

Fuente: Elaboración propia

- El apartado más importante será incluir los datos desde la consulta y para eso los incluyes a partir del menú series, tal y como ves en la imagen.

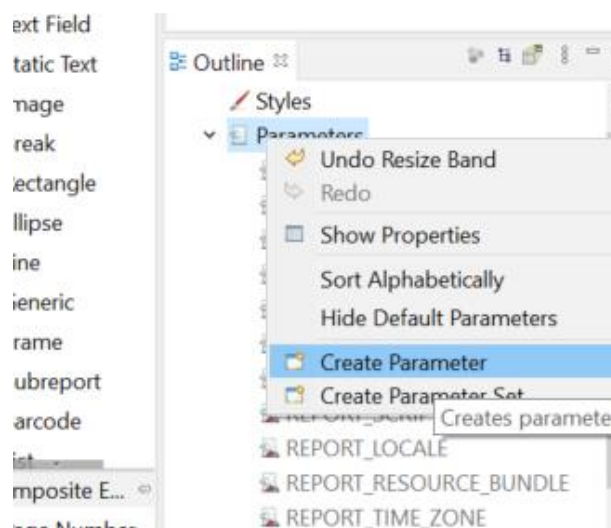


Selección de datos
Fuente: Elaboración propia

- Solo quedará incluir el gráfico al informe.

2.12 Parámetros

Los parámetros dentro de un informe permiten la comunicación entre el entorno, el usuario y el documento o informe. Un parámetro puede tener un valor por defecto y será evaluado en el momento de compilar el informe. Para administrar los parámetros debemos hacer clic derecho sobre el ítem Parameters y elegir Create Parameter, tal y como muestra la imagen.



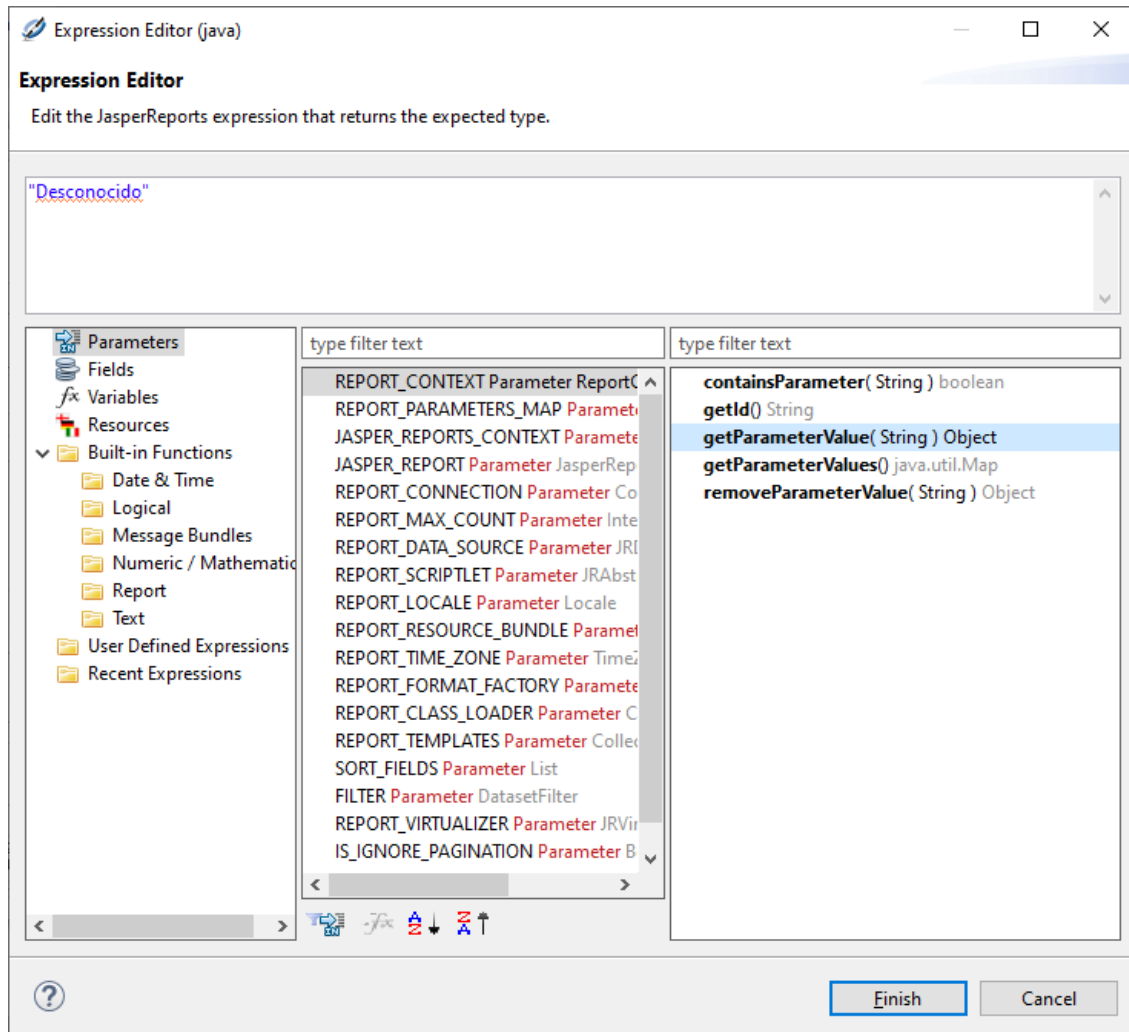
Creación de un nuevo parámetro

Fuente: Elaboración propia

Si, además, vemos las propiedades de ese parámetro tendremos las siguientes opciones:

- Nombre del parámetro.
- Clase o tipo del parámetro.
- Valor por defecto, en el caso de que no se le dé ningún valor cuando se le solicite.
- Clic para solicitar dicho valor al usuario, Is For Prompting.

Un ejemplo sería el siguiente:



Definición de un parámetro

Fuente: Elaboración propia



EJEMPLO PRÁCTICO

El jefe de ventas necesita tener un informe muy visual, una lista de productos filtrados por categoría utilizando JasperSoft Studio. Este informe se conectará a una base de datos MySQL y permitirá filtrar los productos por su categoría para visualizar únicamente aquellos que pertenecen a una categoría específica. ¿Cuáles son los pasos que deberías realizar para crear este informe?

- Se proporciona un Script para poder crear la base de datos:

```
-- Crear la base de datos
CREATE DATABASE tienda;

-- Usar la base de datos recién creada
USE tienda;

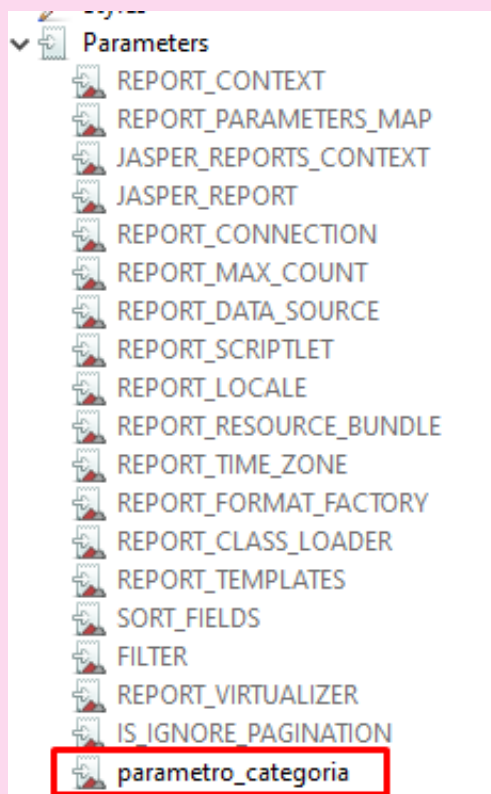
-- Crear la tabla de categorías
CREATE TABLE categorias (
    categoria_id INT AUTO_INCREMENT PRIMARY KEY,
    nombre_categoria VARCHAR(100) NOT NULL
);

-- Insertar algunas categorías
INSERT INTO categorias (nombre_categoria) VALUES
('Electrónica'),
('Ropa'),
('Hogar'),
('Juguetes');

-- Crear la tabla de productos
CREATE TABLE productos (
    producto_id INT AUTO_INCREMENT PRIMARY KEY,
    nombre_producto VARCHAR(150) NOT NULL,
    categoria_id INT,
    precio DECIMAL(10, 2),
    CONSTRAINT fk_categoria FOREIGN KEY (categoria_id) REFERENCES
categorias(categoria_id)
);

-- Insertar algunos productos
INSERT INTO productos (nombre_producto, categoria_id, precio) VALUES
('Televisor LED 40"', 1, 350.00),
('Smartphone 5G', 1, 600.00),
('Camiseta deportiva', 2, 25.00),
('Pantalón vaquero', 2, 45.00),
('Sofá 3 plazas', 3, 499.99),
('Juego de sábanas', 3, 55.50),
('Muñeca de trapo', 4, 15.00),
('Camión de juguete', 4, 30.00);
```


- Primero, creas una consulta SQL que permita filtrar los productos por la categoría seleccionada. Para ello, utilizas un parámetro llamado parametro_categoria que el usuario podrá modificar para realizar el filtrado.

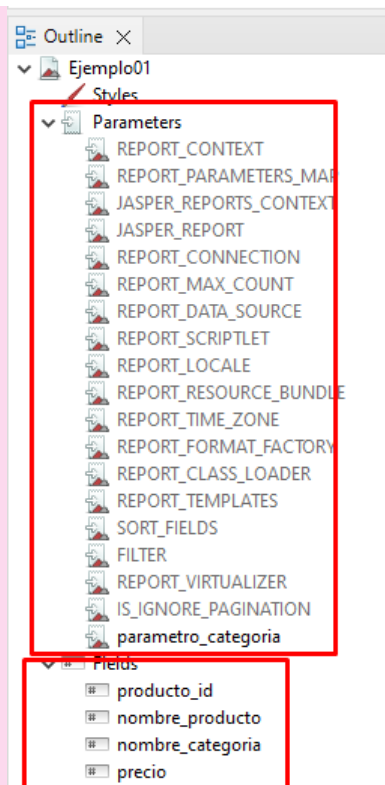


Creación de parámetros

Fuente: Elaboración propia

- Ahora añades esta consulta que incluye el parámetro de filtrado:

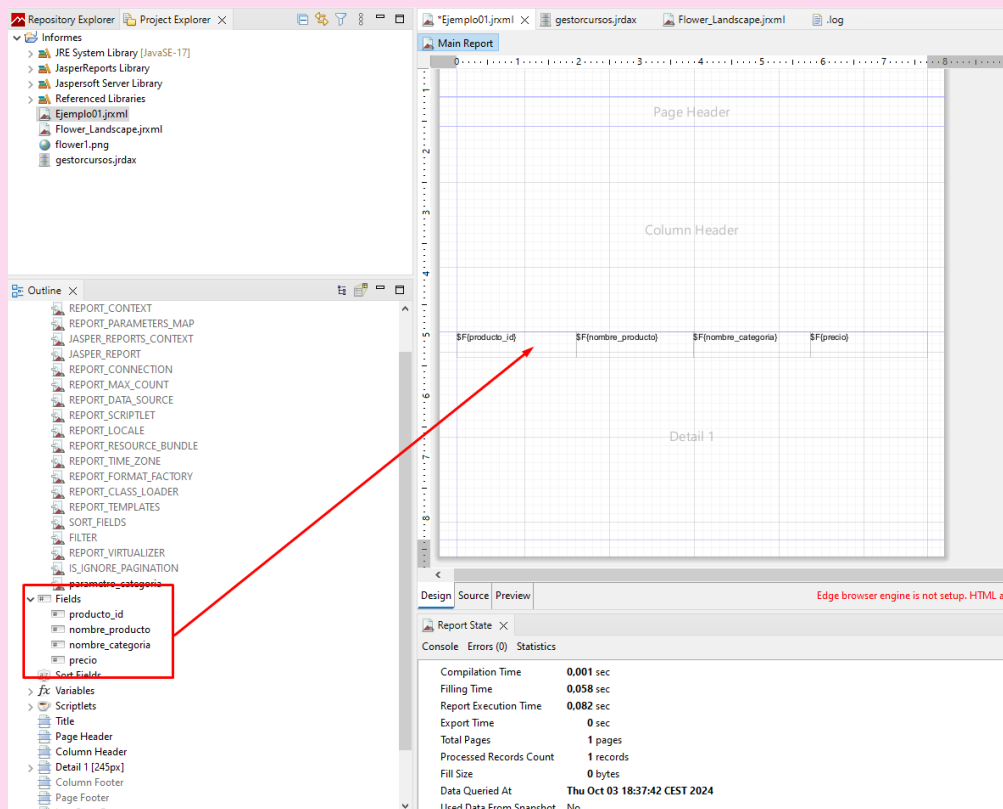
```
SELECT p.producto_id, p.nombre_producto, c.nombre_categoria, p.precio
FROM tienda.productos p
JOIN tienda.categorias c ON p.categoria_id = c.categoria_id
WHERE c.nombre_categoria = ${parametro_categoria}
```



Creación de Consulta

Fuente: Elaboración propia

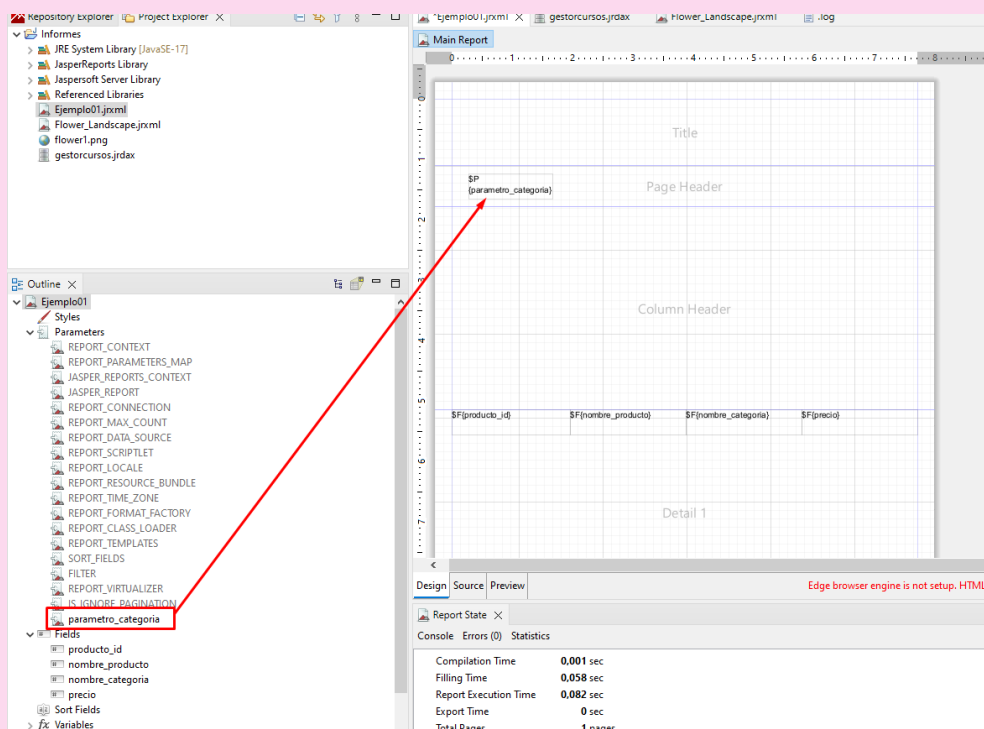
- Ahora hay que arrastrar los campos al informe detalle:



Informe Detalle

Fuente: Elaboración propia

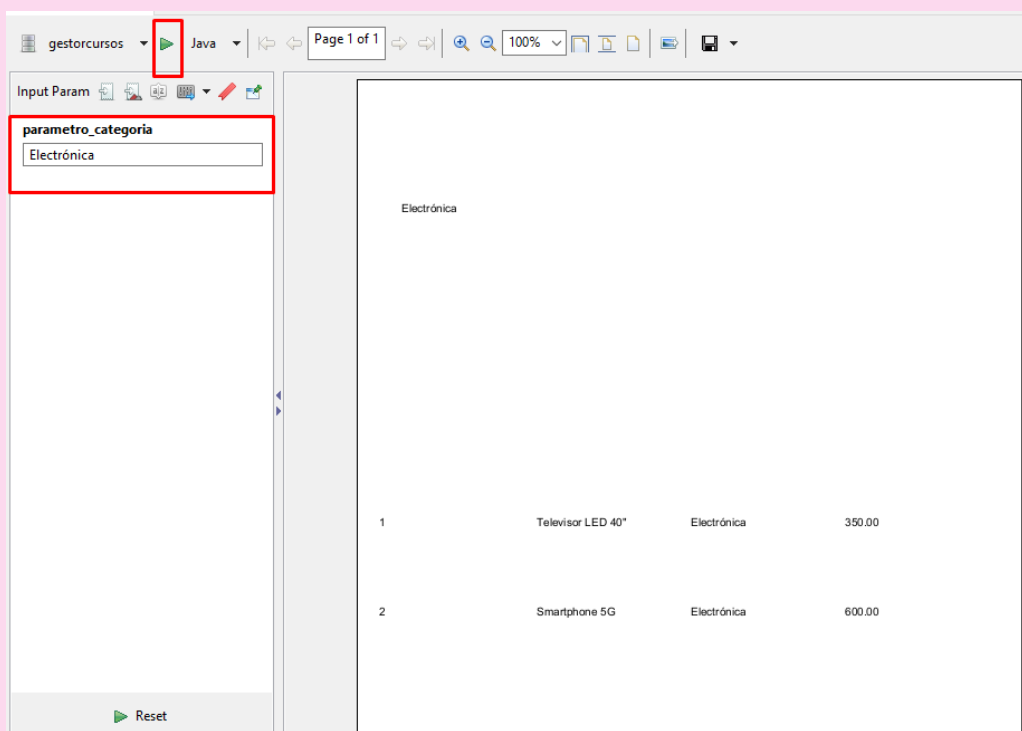
- Ahora arrastras el filtrado a la parte superior del informe.



Adjunto Detalle

Fuente: Elaboración propia

- Guardas el informe y lo ejecutas introduciéndole una categoría por parámetro.



Ejecución Informe

Fuente: Elaboración propia

2.13 Librerías para generación de informes. Clases, métodos y atributos

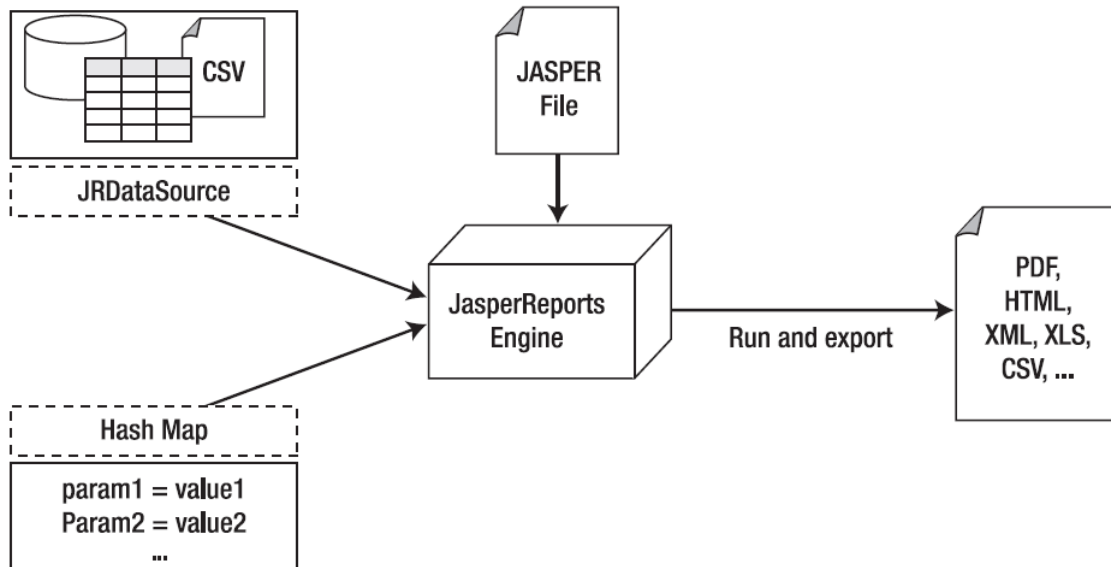
Durante toda la unidad hemos estado trabajando con JasperSoft Studio, el cual nos permite, dentro de un proyecto, definir una serie de informes directamente con la interfaz gráfica. Para hacer esto, JasperSoft Studio configura los informes utilizando una librería llamada JasperReports Library (que se puede descargar en la versión community).

Esta librería define un informe mediante el uso de ficheros XML, compuestos por una serie de secciones, que podemos ver a través del visor fuente en la interfaz:

```
<a name="kanchor251"></a><?xml version="1.0" encoding="UTF-8"?>
<jasperReport
xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd"
name="My first report" pageWidth="595" pageHeight="842"
columnWidth="535"
leftMargin="20" rightMargin="20" topMargin="20"
bottomMargin="20">
  <queryString language="SQL">
    <![CDATA[select * from address order by city]]>
  </queryString>
  <field name="ID" class="java.lang.Integer">
    <fieldDescription><![CDATA[]]></fieldDescription>
  </field>
  <field name="FIRSTNAME" class="java.lang.String">
    <fieldDescription><![CDATA[]]></fieldDescription>
  </field>
  <field name="LASTNAME" class="java.lang.String">
    <fieldDescription><![CDATA[]]></fieldDescription>
  </field>
  <field name="STREET" class="java.lang.String">
    <fieldDescription><![CDATA[]]></fieldDescription>
  </field>
  <field name="CITY" class="java.lang.String">
    <fieldDescription><![CDATA[]]></fieldDescription>
  </field>
```

En resumen, cuando utilizamos Jaspersoft Studio se realiza un proceso que nos permite generar los informes de forma dinámica.



Esquema de funcionamiento de JasperSoft Studio

Fuente: <https://community.jaspersoft.com/documentation/tibco-jaspersoft-studio-user-guide/v640/data-sources-and-print-format>

La compilación del archivo JRXML, que representa el diseño del informe, se realiza mediante el método `compileReport()`, que se encuentra en la clase `JasperCompileManager`. Mediante la compilación, el diseño del informe se carga en un objeto de diseño de informe que luego se serializa y se almacena en el disco (clase `JasperReport`).

Este objeto serializado se utiliza cuando la aplicación desea llenar el diseño de informe especificado con datos. De hecho, la compilación de un diseño de informe implica la compilación de todas las expresiones Java definidas en el archivo JRXML que representa el diseño del informe.

Las clases principales a usar cuando se trabaja con JasperReports Engine son:

- `net.sf.jasperreports.engine.JasperCompileManager`
- `net.sf.jasperreports.engine.JasperFillManager`
- `net.sf.jasperreports.engine.JasperPrintManager`
- `net.sf.jasperreports.engine.JasperExportManager`

Estas clases representan una fachada al motor JasperReports y que tienen varios métodos estáticos que simplifican el acceso a la funcionalidad de la API: se pueden utilizar para compilar un diseño de informe JRXML, para completar un informe, imprimirlo o exportarlo a otros formatos de documento (PDF, HTML, XML).

Además de estas clases, también se puede trabajar directamente con clases de exportador específicas, en caso de que se necesite exportar sus informes a XLS, RTF, ODT, TXT u otros formatos de documento para los que no existe un método auxiliar correspondiente en JasperExportManager o cuando se necesite configurar el proceso de exportación y adaptarlo a sus necesidades específicas. Estas implementaciones de exportador se pueden encontrar en el paquete *net.sf.jasperreports.engine.export* de la biblioteca JasperReports.

Por último, si se necesita mostrar el informe dentro de una aplicación *Swing*, puede utilizar el componente JRViewer que se envía con la biblioteca y consta de un componente *javax.swing.JPanel* integrable y configurable. JasperViewer es una aplicación Swing independiente que utiliza el componente JRViewer para mostrar informes en formato propietario (objetos JasperPrint serializados) o en formato XML.



ENLACE DE INTERÉS

Accede a la web para disponer del Javadoc del JavaReports Library:



EJEMPLO PRÁCTICO

Has realizado un informe que muestra en detalle la cantidad en stock por tienda de una determinada prenda de ropa, informe que es muy útil para el departamento de compras. Sin embargo, cada vez que quieren extraer el informe te solicitan el mismo. ¿Se puede crear una automatización para que el mismo departamento lo pudiera extraer?

Una vez que dispones del informe, pruebas su contenido y lo exportas en un formato que puedas después usar con el motor de JasperReports Library.

El código final de la aplicación quedaría como sigue:

```
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.export.*;
import java.util.*;
public class JasperTest
{
    public static void main(String[] args)
    {
        String fileName = "/departamento/compras/stock.jasper";
        String outFileName = "/departamento/compras/stock.pdf";
        HashMap hm = new HashMap();
        try
        {
            JasperPrint print = JasperFillManager.fillReport(
                fileName,
                hm,
                new JREmptyDataSource());
            JRExporter exporter =
                new net.sf.jasperreports.engine.export.JRPdfExporter();
            exporter.setParameter(
                JRExporterParameter.OUTPUT_FILE_NAME,
                outFileName);
            exporter.setParameter(
                JRExporterParameter.JASPER_PRINT, print);
            exporter.exportReport();
            System.out.println("Se ha creado el fichero: " + outFileName);
        }
        catch (JRException e)
        {
            e.printStackTrace();
            System.exit(1);
        }
        catch (Exception e)
        {
            e.printStackTrace();
            System.exit(1);
        }
    }
}
```

RESUMEN FINAL

En las empresas, actualmente se ha convertido en imprescindible poder manejar la información, realizar informes, analizar los datos y poder adaptarlos a las necesidades de cada departamento o momento.

Hoy en día nos encontramos con múltiples herramientas como Crystal Reports o Eclipse Birt, que funcionan con Java y que permiten integración con nuestras herramientas de desarrollo.

En concreto, JasperSoft es una herramienta muy potente que cuenta con una versión gratuita que permite integrarse con nuestro entorno de desarrollo Eclipse.

Con JasperSoft Studio, podremos realizar informes a partir de una o múltiples fuentes de datos disponiendo la información a través de múltiples secciones. Mediante las herramientas de agrupamiento, filtrado y otras muchas propias del lenguaje SQL, podremos además incorporar un filtrado y presentación mejorado de la información, además de disponer de múltiples herramientas dentro de JasperSoft Studio.

Por último, mediante la exportación a ficheros manejables por otras herramientas de usuario o la exportación en el formato XML propio, podremos reutilizar y extender las posibilidades de nuestros informes creados a priori.