

1. O3 Report Studio	2
1.1 Creando un Reporte nuevo	3
1.2 Área de Trabajo	4
1.3 Especificando los datos a utilizar	7
1.3.1 Definiendo una Fuente de Datos (Data Source)	7
1.3.2 Definiendo un Conjunto de Datos (Data Set)	9
1.3.2.1 Definiendo un Conjunto de Datos (Data Set) a partir de una vista O3	9
1.3.2.2 Definiendo un Conjunto de Datos (Data Set) a partir de una vista O3 extendida	11
1.3.2.3 Definiendo un Conjunto de Datos (Data Set) a través de MDX	14
1.3.2.3.1 Consultas MDX para reportes	15
1.3.2.3.2 Desplegando la fecha de actualización de un Datamart	16
1.4 Parametrizando el reporte	18
1.4.1 Parametrización Simple	18
1.4.2 Parametrización en Cascada	23
1.4.3 Parametrización con Fechas	27
1.4.4 UserLogin como parámetro del reporte en ePortal	38
1.5 Diseñando la forma del reporte	39
1.6 Formateando la información	43
1.7 Funciones de Birt	47
1.7.1 BirtComp	47
1.7.2 BirtMath	55
1.7.3 BirtStr	61
1.8 Reporte con Número Variable de Columnas	67
1.9 Almacenando el reporte	71
1.10 Envío automático de reportes	72
1.11 Dando formato a las leyendas de una gráfica	73

O3 Report Studio

A partir de esta versión, Ideasoft O3 incorpora un nuevo componente para generación de reportes estáticos. Este componente está fuertemente basado en Eclipse BIRT, que ha sido extendido para integrarse con los modelos multidimensionales de O3.

Este nuevo componente de Ideasoft O3, incluye un entorno de creación de reportes (a nivel del Desktop), un entorno de ejecución de los mismos a nivel del Web y del Desktop. El acceso a los modelos multidimensionales desde el reporteador puede hacerse en forma simplificada mediante referencias a vistas o utilizando el lenguaje de consultas multidimensionales MDX. O3 soporta para esto un subconjunto de las sentencias de consulta de MDX que es un estándar de facto de la industria.

Mediante este componente, los reportes de O3 pueden generar excelentes niveles de presentación en múltiples formatos, incluyendo PDF, HTML, MS WORD, etc., creados a partir de información proveniente de los modelos multidimensionales así como en forma directa de las bases de datos relacionales.

Este módulo se encuentra totalmente integrado con las capacidades analíticas de O3 lo cual representa una extensión significativa de las capacidades de O3 en cuanto a la explotación de datos.

Este Manual

El objetivo de este Manual es presentar el módulo para generación de reportes disponible desde la versión 5.0 de O3, brindando una guía sencilla acerca de su uso con ejemplos concretos que pueden ser seguidos en una instalación cualquiera de O3.

Se recomienda además la lectura del material bibliográfico detallado en la sección de bibliografía más abajo para reportes más complejos o por detalles específicos de la herramienta.

Esta documentación está organizada de acuerdo a los pasos necesarios para construir un reporte vistoso utilizando los datos almacenados en el cubo de Vinos incluido en la instalación de O3.

How To O3 Report

How To O3 Report
<ul style="list-style-type: none">• Almacenando el reporte• Área de Trabajo• Creando un Reporte nuevo• Dando formato a las leyendas de una gráfica• Diseñando la forma del reporte• Envío automático de reportes• Especificando los datos a utilizar• Formateando la información• Funciones de Birt• Parametrizando el reporte• Reporte con Número Variable de Columnas

Bibliografía

Libro	Autor
BIRT - A Field Guide te Reporting	Diana Peh/Aletha Honnemann/Nola Hauge

Documentación Online

Autor	Link
Eclipse Documentation	http://help.eclipse.org/helios/nav/9_0

- Creando un Reporte nuevo
- Área de Trabajo
- Especificando los datos a utilizar
- Parametrizando el reporte
- Diseñando la forma del reporte
- Formateando la información

- Funciones de Birt
- Reporte con Número Variable de Columnas
- Almacenando el reporte
- Envío automático de reportes
- Dando formato a las leyendas de una gráfica

Creando un Reporte nuevo

Esta sección describe cómo construir un nuevo reporte paso a paso utilizando un ejemplo sencillo con datos que provienen del cubo Vinos distribuido con el producto.

El ejemplo terminado puede consultarse aquí.

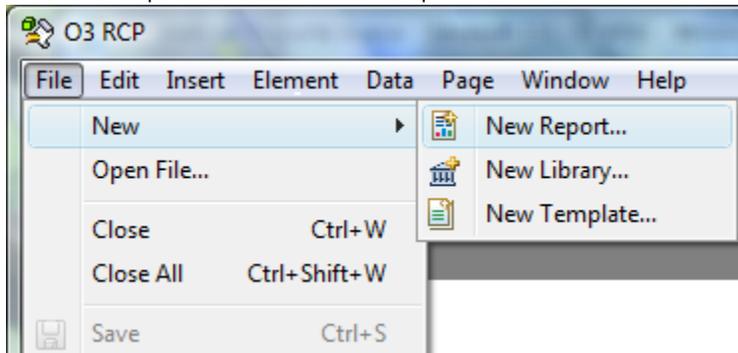
Creando un nuevo Reporte

Los reportes pueden crearse de dos formas diferentes:

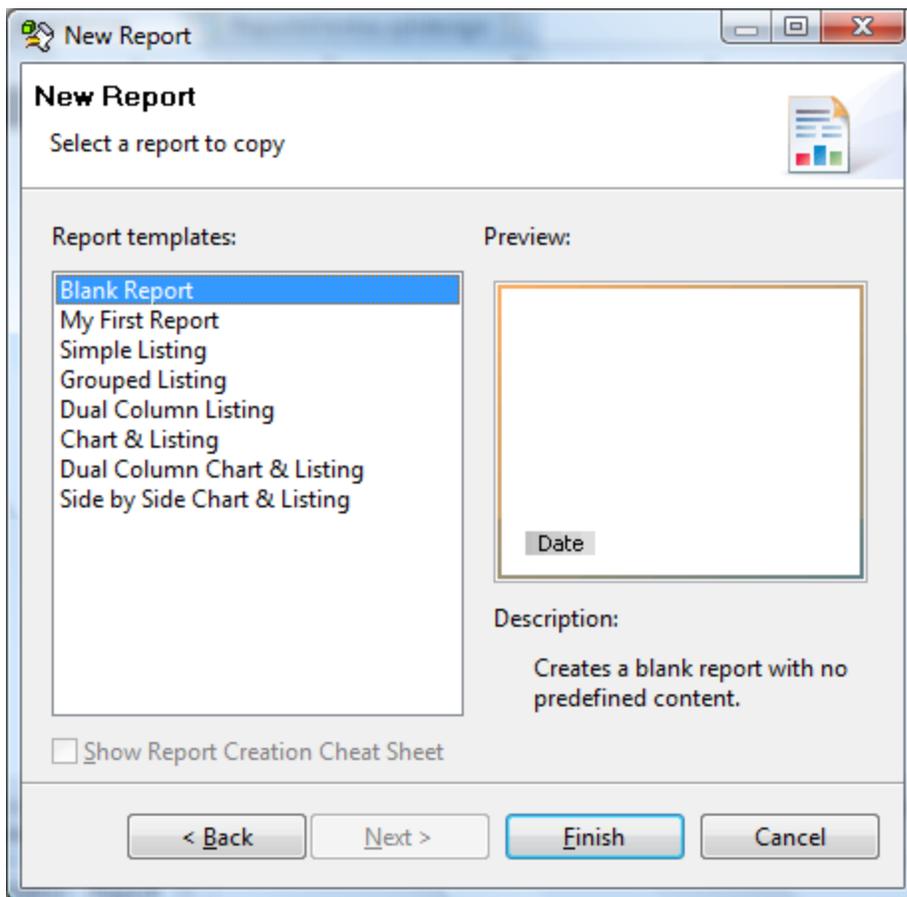
- Comenzando con un diseño en blanco.
- Usando una plantilla prediseñada de reporte. En este caso se sugieren los pasos necesarios para ir armando el reporte ajustándose a un diseño preestablecido.

A los efectos de este tutorial se utiliza un reporte en blanco.

1. Seleccione la opción de menú File->New Report



2. Ingrese mi_reporte.rptdesign como nombre del Reporte
3. Seleccione **Blank Report** y finalmente presione el botón **Finish**

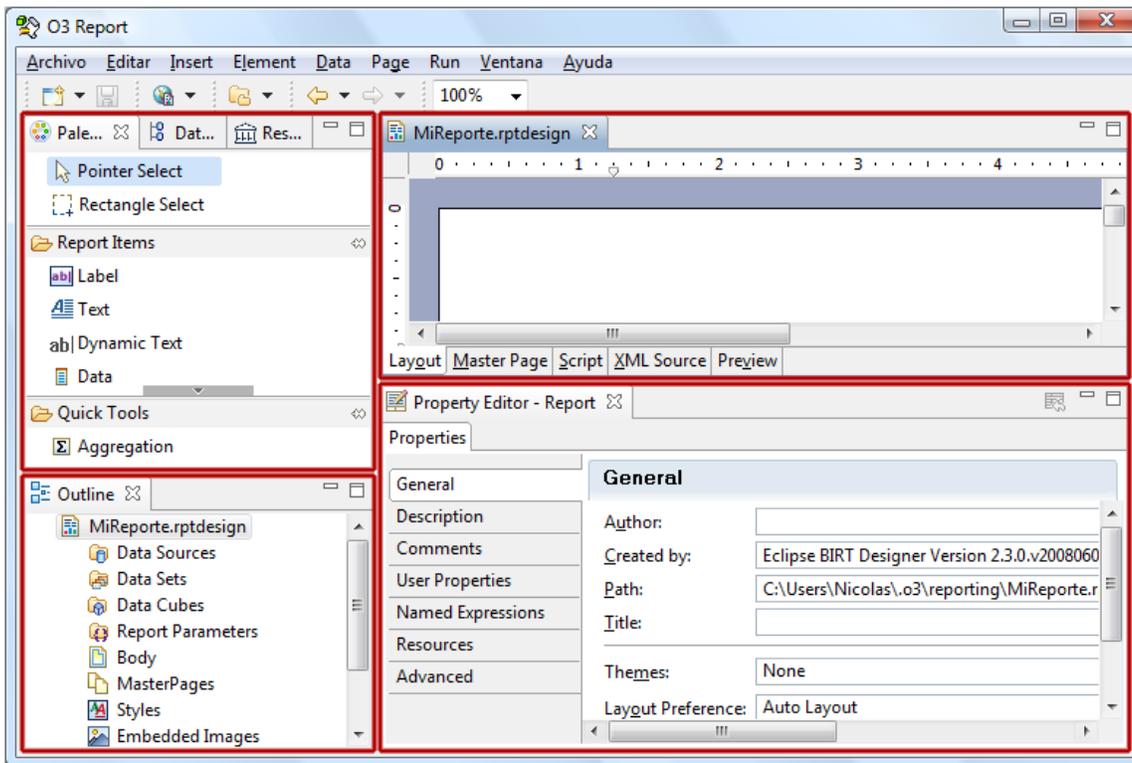


Con estos pasos queda disponible en nuevo reporte para comenzar a ubicar los elementos sobre él

Área de Trabajo

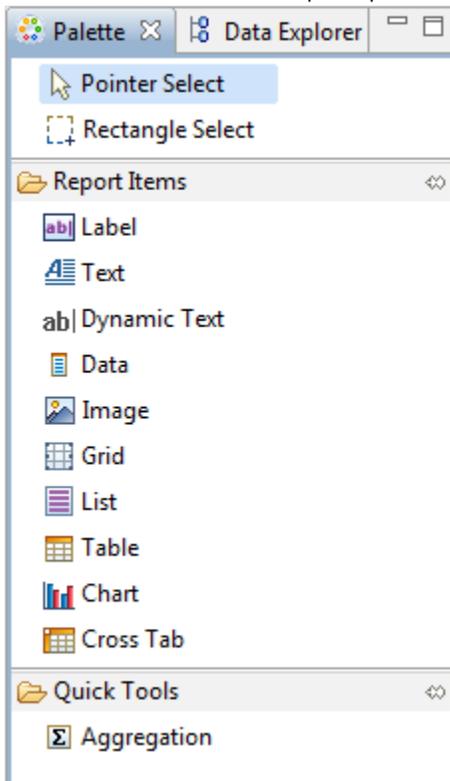
Esta sección describe el área de trabajo personalizable de O3 Report.

Área de Trabajo



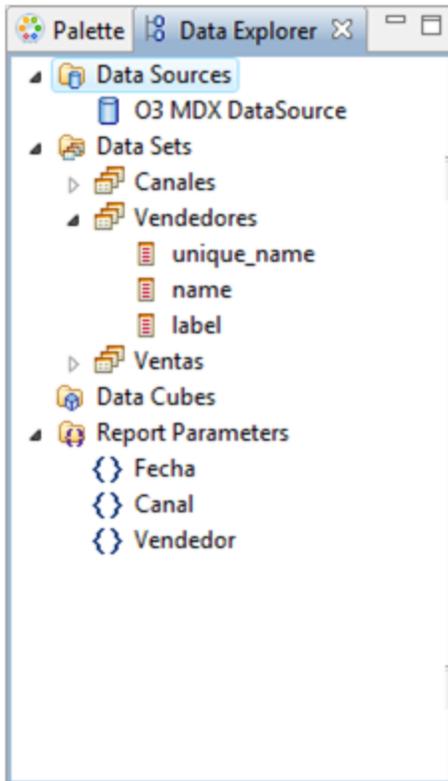
Paleta (Palette)

- Muestra los elementos que se puede utilizar en un reporte.



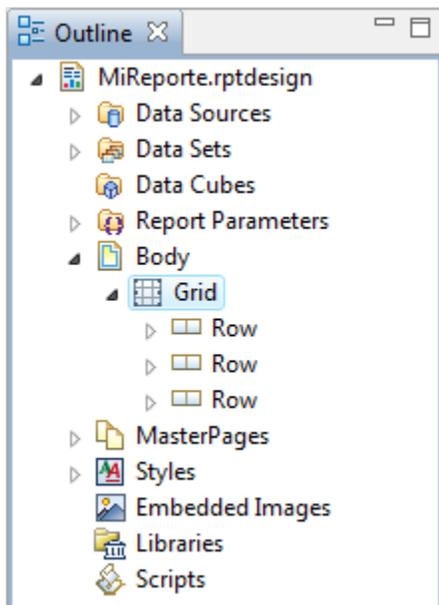
Explorador de Datos (Data Explorer)

- Muestra los datos disponibles para el reporte.
- Permite crear nuevas fuentes de datos, conjuntos de datos y parámetros.



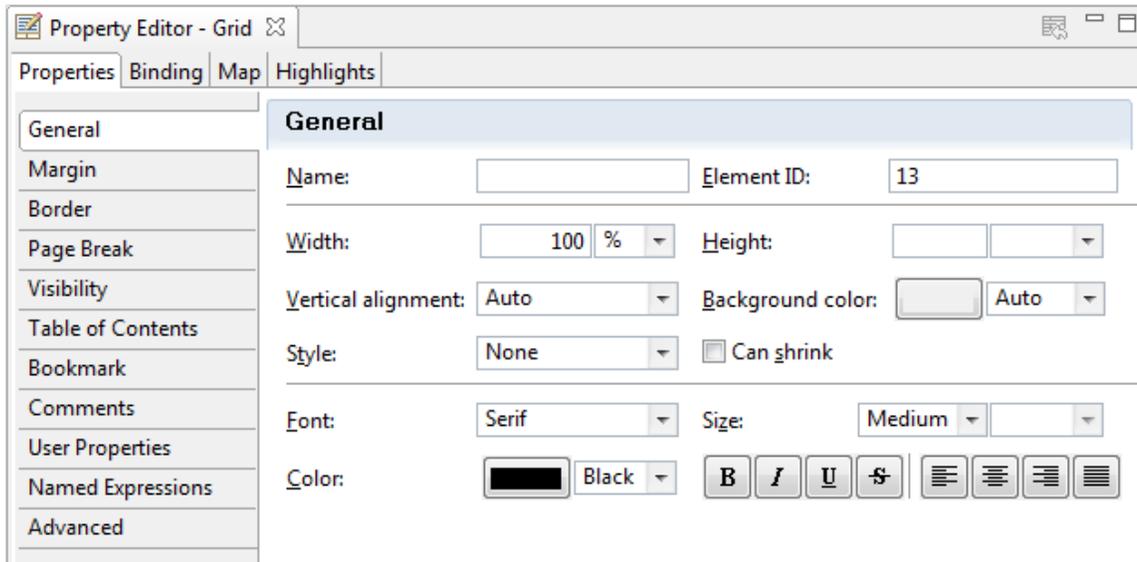
Esquema (Outline)

- Muestra la estructura del reporte como un árbol.
- Se pueden seleccionar elementos desde aquí para eliminar o modificar.



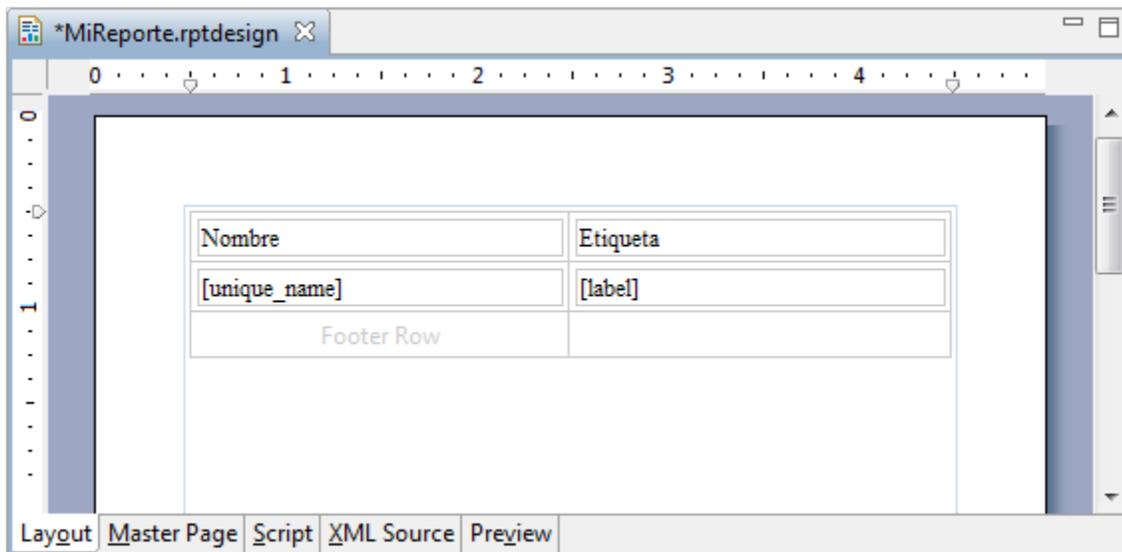
Editor de Propiedades

- Permite editar las propiedades del elemento que esté seleccionado.
- Sirve para aplicar estilos y formato de los elementos.



Editor del Reporte

- Múltiples reportes a la vez.
 - Layout: es la hoja donde se crea y diagrama el reporte.
 - Master Page: define la hoja del reporte (tamaño, márgenes, fondo, encabezado, pie de página).
 - Script: permite agregar JavaScript para controlar la generación del reporte.
 - XML Source: muestra el XML que se genera al diseñar el reporte.
 - Preview: muestra una vista preliminar del reporte con un subconjunto de los datos.



Especificando los datos a utilizar

Antes de poder diseñar el reporte es necesario especificar sobre qué datos se trabajará y cuál será el mecanismo de conexión.

Este tutorial construye un reporte utilizando datos almacenados en un cubo de O3.

La extracción de estos datos se realiza mediante 2 pasos:

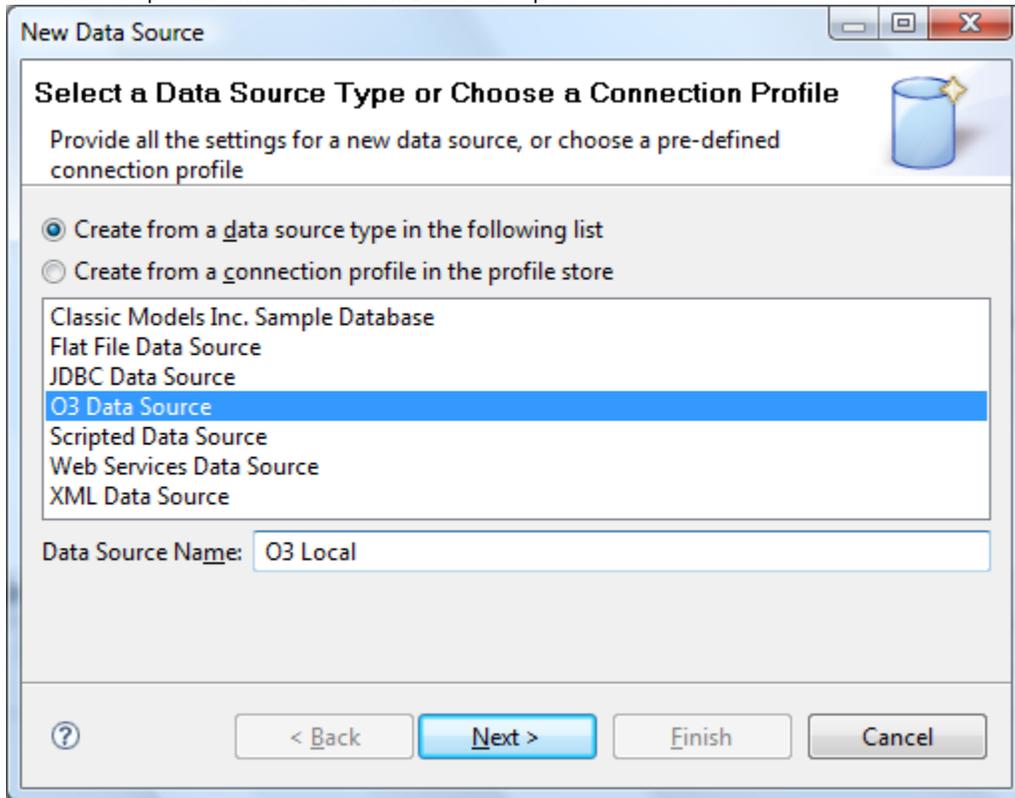
- Definiendo una Fuente de Datos (Data Source)
- Definiendo un Conjunto de Datos (Data Set)

Es posible tener varios conjuntos de datos definidos sobre la misma fuente de datos

Definiendo una Fuente de Datos (Data Source)

Definiendo una Fuente de Datos (Data Source)

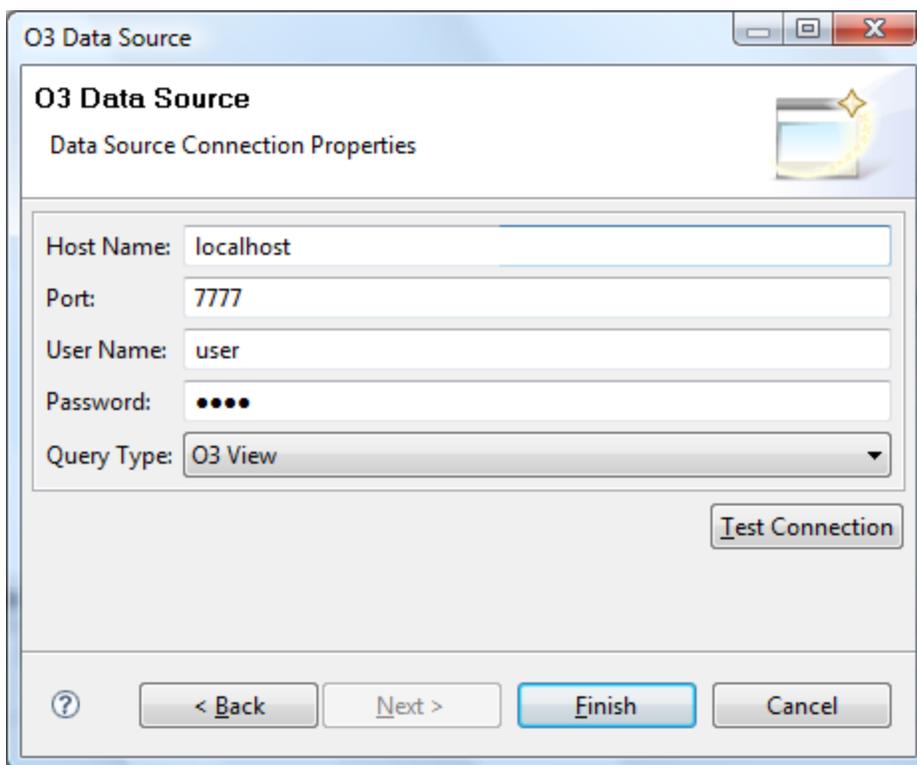
1. Seleccione la opción del menú Data->New Data Source para crear una nueva Fuente de datos



2. Elija el tipo de fuente de datos para O3 (O3 Data Source)
Esto permite definir una fuente de datos a cubos y vistas de O3. Estas fuentes de datos a O3 pueden ser de dos tipos:
 - Fuentes que acceden a vistas almacenadas en el servidor
 - Fuentes que acceden a cubos almacenados en el servidor y utilizan el lenguaje MDX para especificar la consulta
3. Indique el nombre de la fuente de datos (Data Source Name)
En este ejemplo llamaremos "O3 Local" a la fuente de datos
4. Ingrese los parámetros de la Fuente de datos

Host Name:	El nombre o dirección IP del servidor donde se encuentra ejecutando O3. En nuestro caso localhost
Port:	Puerto del servidor indicado en el parámetro anterior donde se encuentra ejecutando el servidor de O3. En nuestro caso usaremos el puerto por defecto 7777
User Name:	Nombre del usuario definido en O3 que se usará para conectarse al servidor. En nuestro caso usaremos el usuario user
Password:	Contraseña del usuario definido en el campo anterior En nuestro caso usaremos la password del usuario user
Query Type:	Indica el tipo de Fuente de datos que se usará: Consulta MDX, Vista O3 o Vista O3 Extendida.

A modo de ejemplo, se presenta la configuración para O3 View:



5. Verificar que la conexión se estableció en forma correcta presionando el botón **Test Connection...**

Definiendo un Conjunto de Datos (Data Set)

Definiendo un Conjunto de Datos (Data Set)

- Definiendo un Conjunto de Datos (Data Set) a partir de una vista O3
- Definiendo un Conjunto de Datos (Data Set) a partir de una vista O3 extendida
- Definiendo un Conjunto de Datos (Data Set) a través de MDX

Definiendo un Conjunto de Datos (Data Set) a partir de una vista O3

Definiendo un Conjunto de Datos (Data Set) a partir de una vista O3

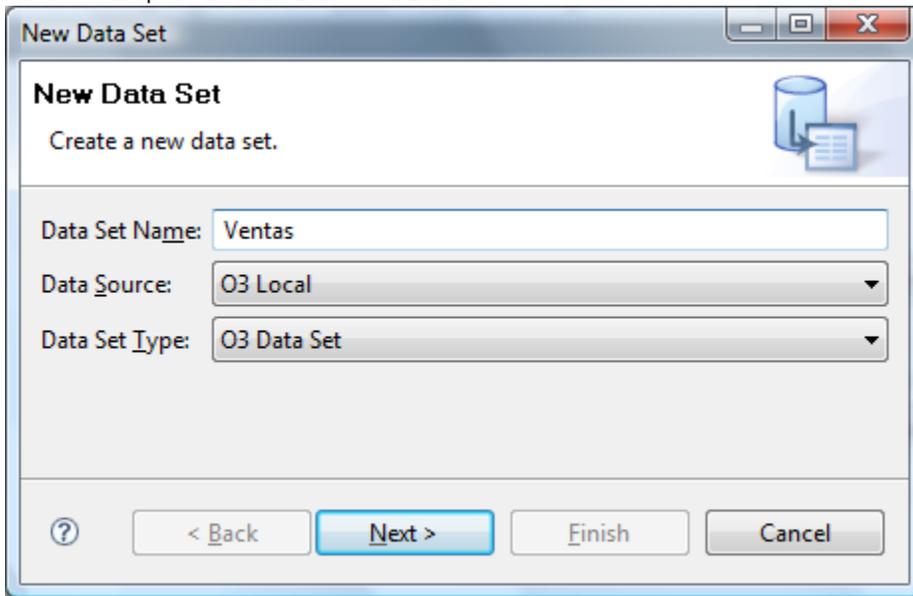
Para el caso concreto de este tutorial se utiliza una vista sobre el cubo de vinos que se define como se muestra en la siguiente imagen

		Medidas	
Vendedores	Vinos	Medidas	
Vendedor	Vino	Litros Vendidos	Venta Neta
G & P Corp.	Moscat Semillon	450	\$ 273.341,00
	Nebbiolo	10.800	\$ 7.024.984,00
	Pinot Blanc	900	\$ 602.942,00
	Pinot Noir	900	\$ 551.000,00
	Ruby Cabernet	3.600	\$ 2.319.081,00
	Sauvignon Blanc	4.500	\$ 2.790.518,00
Link S.R.L.	Cabernet Sauvignon, Merlot	3.150	\$ 2.216.611,00
	Cabernet, Tannat, Merlot	3.150	\$ 2.105.037,00
	Chenin, Gewurzt	2.250	\$ 1.535.941,00
	Moscatel, Merlot	4.950	\$ 2.990.956,00
	Pinot Blanc, Merlot	900	\$ 566.106,00
	Pinot Noir, Merlot, Malbec	2.700	\$ 1.819.582,00
	Sauvignon, Pinot Blanc	3.150	\$ 2.158.091,00

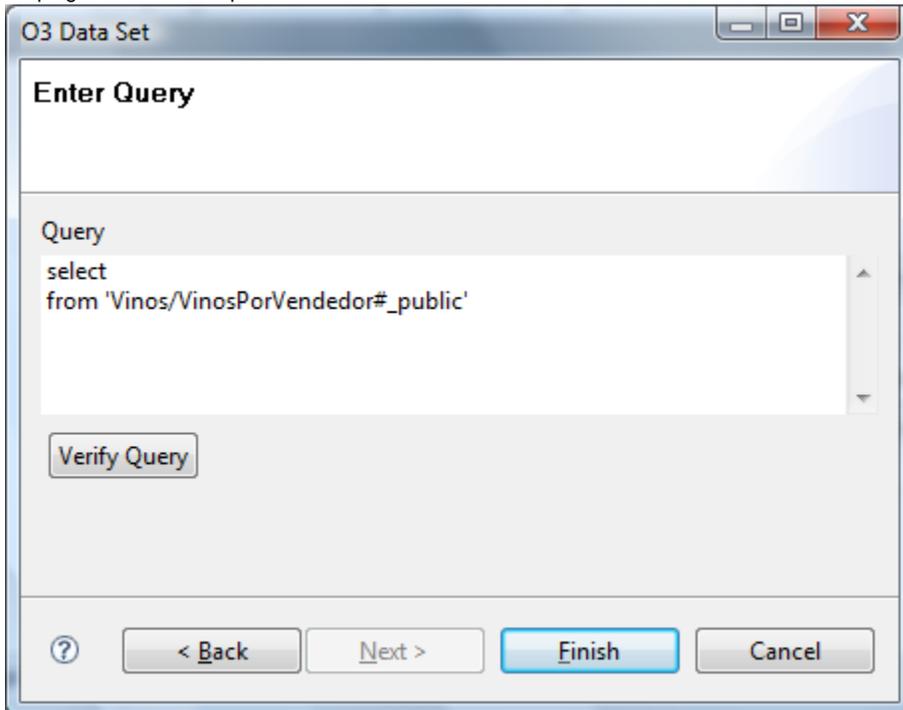
En esta vista se seleccionan las medidas Litros Vendidos y Venta Neta como columnas y en el eje vertical se colocan las dimensiones de Vendedores y Vinos en sus respectivos niveles Vendedor y Vino.

Una vez definida la vista y almacenada en el servidor con el nombre **VinosPorVendedor** con alcance público se deben seguir los siguientes pasos:

1. Seleccione la opción del menú Data->New Data Set



2. Indique el nombre del Conjunto de Datos (Data Set Name)
3. Seleccione la Fuente de datos sobre la que se desea definir el conjunto de datos
En este ejemplo la fuente de datos se llama "O3 Local".
4. Seleccionar **O3 Data Set** como tipo de Conjunto de Datos
5. En la ventana de definición del nuevo Conjunto de Datos, ingrese la consulta que se muestra a continuación para obtener los datos desplegados en la vista pública VinosPorVendedor. Esta vista debe haber sido creada con anterioridad en el servidor de O3



En la consulta es importante destacar la cláusula **FROM** de la clausulta que hace referencia a '**Vinos/VinosPorVendedor#_public**'. Este es el nombre con el que se hace referencia a la vista necesaria para este reporte.

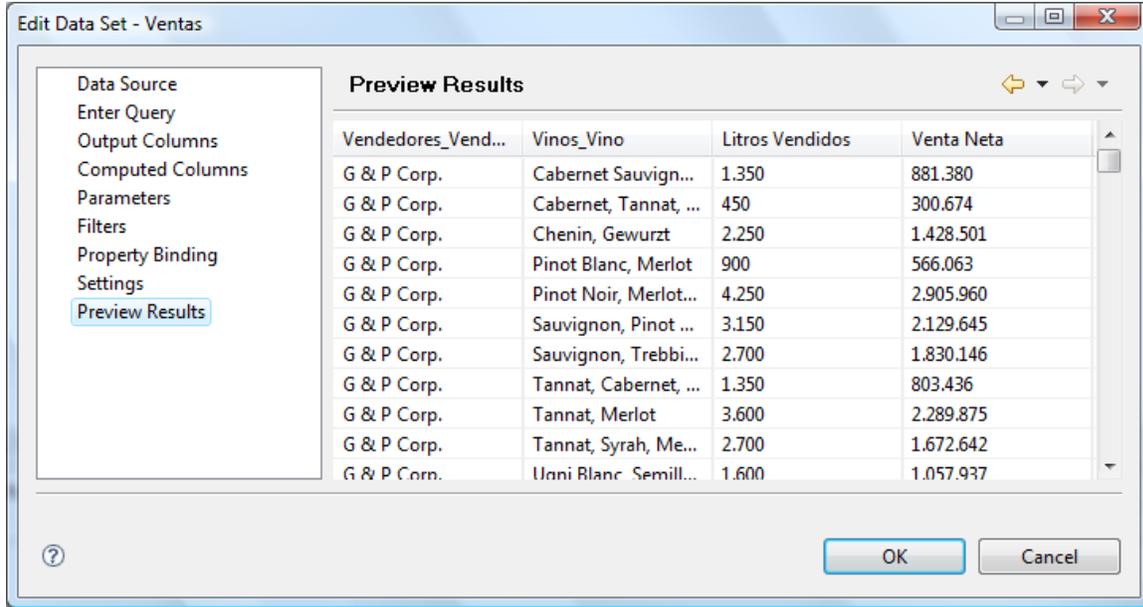
Vinos	Es el nombre del cubo tal como está publicado en el servidor
VinosPorVendedor	Es el nombre de la vista de donde se extraen los datos
#_public	Indica que la vista es una vista pública.

Si la vista fuera privada o de un rol particular:

#_private	Indica que la vista es una vista privada.
#comercial	Indica que la vista es una vista del rol comercial.

Estos datos pueden ser sacados de la barra de estado del navegador web, luego de source=Vinos/VinosPorVendedor#_public en la página de marcadores.

6. Presionar el botón **Finish** para aceptar los cambios
7. La ventana que se muestra a continuación permite modificar parámetros adicionales del Conjunto de Datos, así como obtener una Vista Preliminar (Preview Results) de la consulta



Definiendo un Conjunto de Datos (Data Set) a partir de una vista O3 extendida

Funcionalidad disponible a partir de la versión 6.3.2.

Página en construcción

Definiendo un Conjunto de Datos (Data Set) a partir de una vista O3 extendida

Para el caso concreto de este tutorial se utiliza una vista sobre el cubo de vinos que se define como se muestra en la siguiente imagen

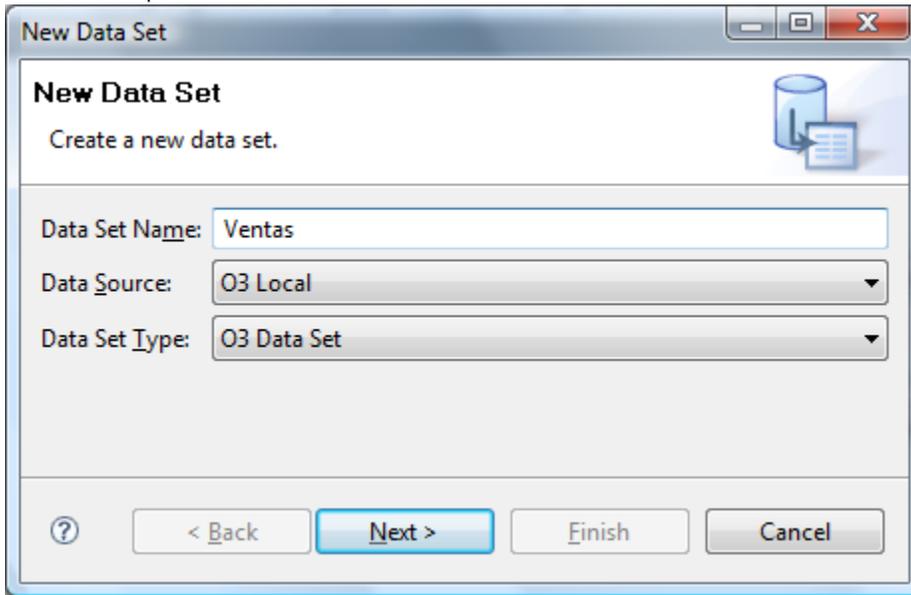
Vendedores	Vinos	Medidas	
		Litros Vendidos	Venta Neta
G & P Corp.	Moscat Semillon	450	\$ 273.341,00
	Nebbiolo	10.800	\$ 7.024.984,00
	Pinot Blanc	900	\$ 602.942,00
	Pinot Noir	900	\$ 551.000,00
	Ruby Cabernet	3.600	\$ 2.319.081,00
Link S.R.L.	Sauvignon Blanc	4.500	\$ 2.790.518,00
	Cabernet Sauvignon, Merlot	3.150	\$ 2.216.611,00
	Cabernet, Tannat, Merlot	3.150	\$ 2.105.037,00
	Chenin, Gewurzt	2.250	\$ 1.535.941,00
	Moscatel, Merlot	4.950	\$ 2.990.956,00
	Pinot Blanc, Merlot	900	\$ 566.106,00
	Pinot Noir, Merlot, Malbec	2.700	\$ 1.819.582,00
Sauvignon, Pinot Blanc	3.150	\$ 2.158.091,00	

En esta vista se seleccionan las medidas Litros Vendidos y Venta Neta como columnas y en el eje vertical se colocan las dimensiones de

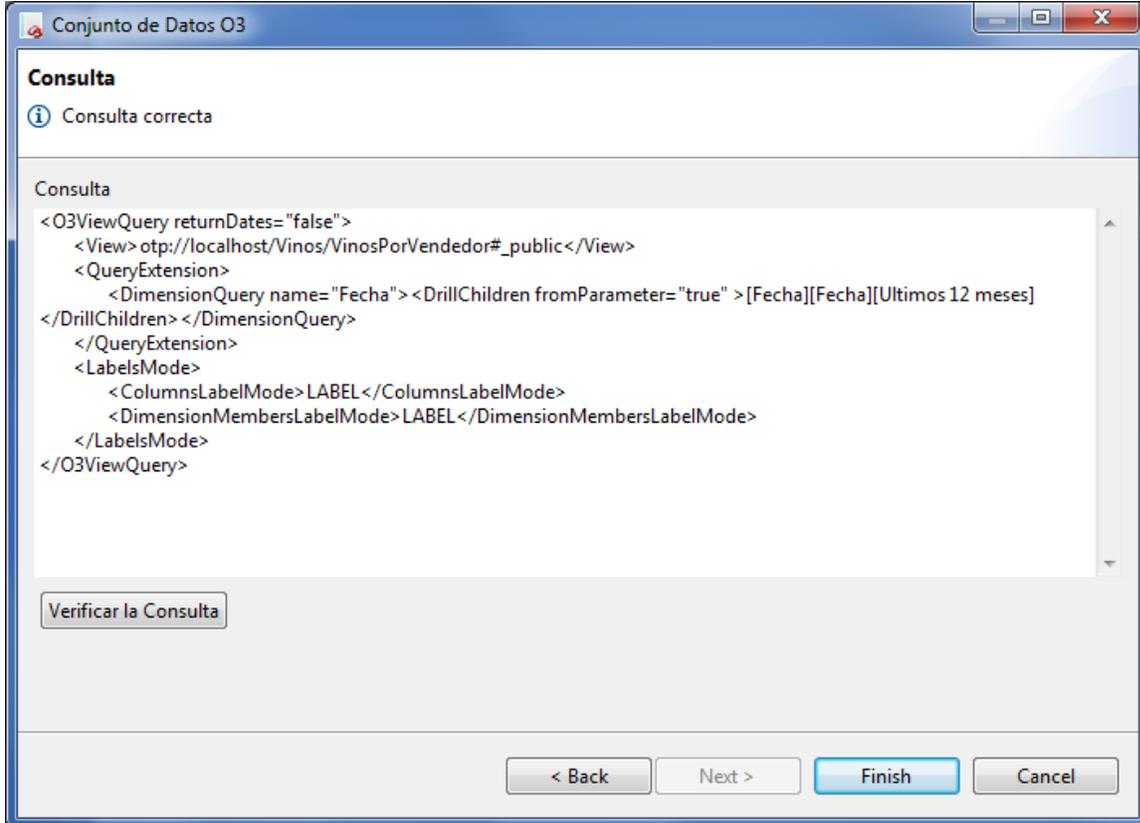
Vendedores y Vinos en sus respectivos niveles Vendedor y Vino.

Una vez definida la vista y almacenada en el servidor con el nombre **VinosPorVendedor** con alcance público se deben seguir los siguientes pasos:

1. Seleccione la opción del menú Data->New Data Set



2. Indique el nombre del Conjunto de Datos (Data Set Name)
3. Seleccione la Fuente de datos sobre la que se desea definir el conjunto de datos
En este ejemplo la fuente de datos se llama "O3 Local".
4. Seleccionar **O3 Data Set** como tipo de Conjunto de Datos
5. En la ventana de definición del nuevo Conjunto de Datos, ingrese la consulta que se muestra a continuación para obtener los datos desplegados en la vista pública VinosPorVendedor. Esta vista debe haber sido creada con anterioridad en el servidor de O3



En la consulta es importante destacar el nodo <View> con el valor: **'Vinos/VinosPorVendedor#_public'**. Este es el nombre con el que se hace referencia a la vista necesaria para este reporte.

Vinos	Es el nombre del cubo tal como está publicado en el servidor
VinosPorVendedor	Es el nombre de la vista de donde se extraen los datos
#_public	Indica que la vista es una vista pública.

Si la vista fuera privada o de un rol particular:

#_private	Indica que la vista es una vista privada.
#comercial	Indica que la vista es una vista del rol comercial.

Estos datos pueden ser sacados de la barra de estado del navegador web, luego de source=Vinos/VinosPorVendedor#_public en la página de marcadores.

Las opciones de configuración para <DimensionQuery> son:

Explorar Hijos

```
<DrillChildren fromParameter="true">[Date][Date][2010]</DrillChildren>
```

Explorar Nivel

```
<DrillLevel><LevelName>Mes</LevelName></DrillLevel>
```

Explorar Nivel desde-hasta

```
<LevelRange>
  <Level>Mes</Level>
  <From fromParameter="true">[Date]</From>
  <To fromParameter="true">[Date]</To>
</LevelRange>
```

Lista de Elementos

```
<MemberList>
  <Member> </Member>
  <Member> </Member>
  <Member> </Member>
  <Member> </Member>
</MemberList>
```

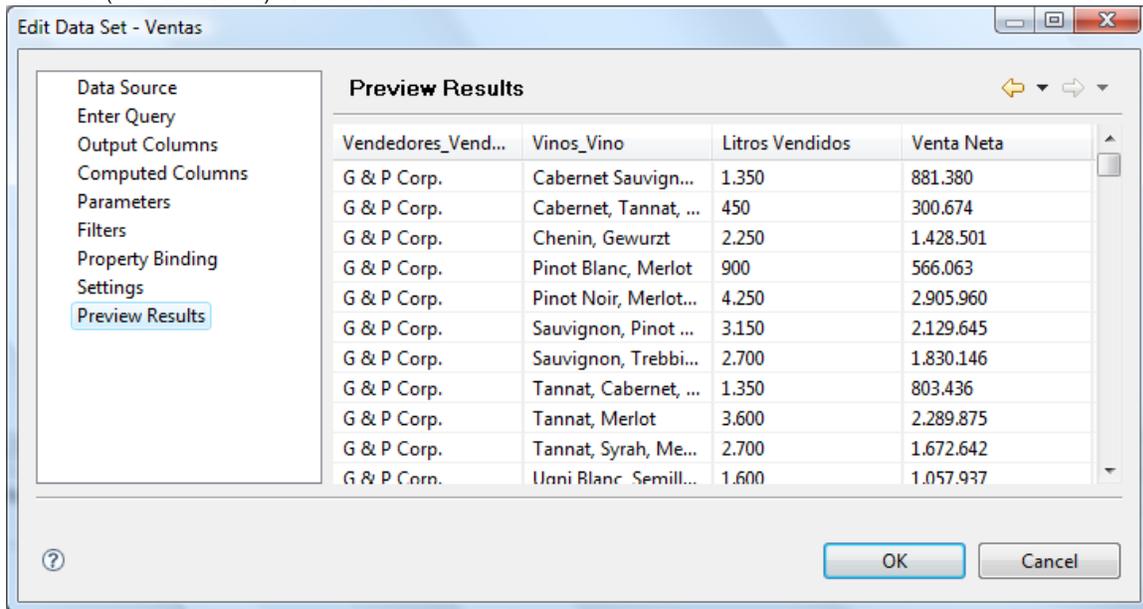
Fechas Relativas

```

<RelativeDate>
  <RelativeLevel>(Nombre dimension relativa)</RelativeLevel>
    <ReferenceDate>(Fecha base en formato YYYY-MM-DD)</ReferenceDate>
</RelativeDate>

```

6. Presionar el botón **Finish** para aceptar los cambios
7. La ventana que se muestra a continuación permite modificar parámetros adicionales del Conjunto de Datos, así como obtener una Vista Preliminar (Preview Results) de la consulta



Definiendo un Conjunto de Datos (Data Set) a través de MDX

Definiendo un Conjunto de Datos a través de MDX

Los pasos anteriores permitieron definir un Conjunto de Datos utilizando una vista previamente almacenada en el servidor.

O3 permite un método alternativo que no requiere salvar ninguna vista, sino que por el contrario permite al diseñador del reporte indicar la consulta directamente en el propio reporte.

Este método alternativo utiliza el lenguaje de consulta MDX, el cual permite especificar consultas multidimensionales sobre los cubos publicados en el servidor de O3

1. Crear una nueva fuente de datos (Data Source) llamada O3 Local MDX con los siguientes parámetros

Host Name:	localhost
Port:	7777
User Name:	user
Password:	user
Query Type:	MDX Query

2. Crear un nuevo conjunto de datos (Data Set) llamado Ventas MDX con los siguientes parámetros

Name:	Ventas MDX
Data Source:	O3 Local MDX
Data Set Type:	O3 Data Set

3. Ingresar la siguiente consulta MDX para este nuevo conjunto de datos

```

SELECT
  {[Measures].[Litros Vendidos], [Measures].[Venta Neta]} ON COLUMNS
  ,CROSSJOIN({[Vendedores].[Vendedor].members}, {[Vinos].[Vino].members}) ON ROWS
FROM [Vinos]

```

En esta consulta se genera una tabla de datos que contendrá:

- Una fila por cada Vino vendido por cada Distribuidor
- Para cada combinación de Vino con Distribuidor mostrará 2 columnas (las medidas Litros Vendidos y Venta Neta)

Distribuidor	Vino	Litros Vendidos	Venta Neta
Distribuidor 1	Vino 1	10	1500
Distribuidor 1	Vino 2	15	2700
Distribuidor 2	Vino 1	13	1780
...
Distribuidor 3	Vino 3	12	720

La cláusula CROSSJOIN de la consulta es la que permite lograr la anidación de las dimensiones de Distribuidores y Vinos que también existía en la Vista.

 Por más detalles acerca de la sintaxis y las opciones que brinda el lenguaje MDX soportado por O3 referirse a [Consultas MDX para reportes](#)

Desplegando la fecha de actualización de un Datamart

[Desplegando la fecha de actualización de un Datamart](#)

Consultas MDX para reportes

Consultas MDX para reportes

El objetivo de este tutorial es introducir el lenguaje de consulta MDX soportado desde la versión 5.0 de O3 como lenguaje para hacer consultas desde el reporteador.

El tutorial está organizado en base a un conjunto de casos utilizando el modelo del cubo *Demo* como base de consulta. Para ver un ejemplo en forma rápida ver [más abajo](#).

- [Introducción a MDX de O3](#)
- [Estructura y datos del cubo Demo](#)
- [Caso 1 Lista de elementos en ejes](#)
- [Caso 2 Lista de elementos de un nivel en ejes](#)
- [Caso 3 Filtro básico en dimensiones](#)
- [Caso 4 Lista de elementos según jerarquía](#)
- [Caso 5 Combinando formas de listar elementos](#)
- [Caso 6 Combinación de varias dimensiones en ejes](#)
- [Caso 7 Ocultar elementos en ejes](#)
- [Consulta MDX de metadata](#)

Ejemplo

Una sentencia MDX de consulta tiene la siguiente estructura básica.

```

SELECT
  <colExpresion> ON COLUMNS,
  <rowExpresion> ON ROWS
FROM <cubeName>
WHERE <filterExpresion>

```

Describiremos sus elementos a través de un ejemplo. Supongamos que del modelo de análisis de ventas de nombre *Demo* queremos consultar las unidades vendidas en *France* a los clientes clasificados como *Major Accounts*.

La consulta MDX sería la siguiente:

```
SELECT
    {Customers.[Major Accounts]} ON COLUMNS,
    {Location.[France]} ON ROWS
FROM Demo
WHERE (Measures.[Units Sold])
```

El cubo conteniendo la información se indica en el **FROM** de la consulta. *Major Accounts* es un miembro del nivel de más arriba de la dimensión *Customers*, por lo tanto la forma de referirse a él es mediante la expresión *Customers.[Major Accounts]*. En esta consulta se ha decidido que este valor aparezca como columna en el resultado, de ahí que se declara **ON COLUMNS**. Análogamente, *France* es un miembro del nivel más arriba de la dimensión *Location*. La forma de referirse a él es *Location.France* y para que aparezca en las filas se especifica **ON ROWS**. Por último, en el **WHERE** de la consulta se indica la medida que se quiere utilizar. Las medidas pertenecen a una dimensión de nombre *Measures*. La forma de referirse a la medida deseada en este caso es *Measures.[Units Sold]*.

El resultado de esta consulta puede leerse: "En Francia, las unidades vendidas a los clientes clasificados como de Grandes Cuentas es 2841".

Desplegando la fecha de actualización de un Datamart

En ocasiones es necesario desplegar como parte del reporte la fecha en que la información de un datamart fue actualizada.

Para ello **O3 Report** dispone de un mecanismo que permite consultar esta información utilizando MDX

Definiendo la consulta

1. Agregue una nueva consulta MDX con la siguiente sentencia

```
SELECT {CubeInfo.LastModifiedDate} ON COLUMNS
FROM Vinos
```

En esta sentencia, **CubeInfo** indica que se requiere información del Datamart, y **LastModifiedDate** identifica la la fecha de la última actualización del mismo.

En el ejemplo que se muestra a continuación se nombró la consulta como "*Fecha de Actualización de Vinos*"

Desplegando la fecha de actualización en el reporte

Existen varias formas en que puede incluirse esta información en el reporte.

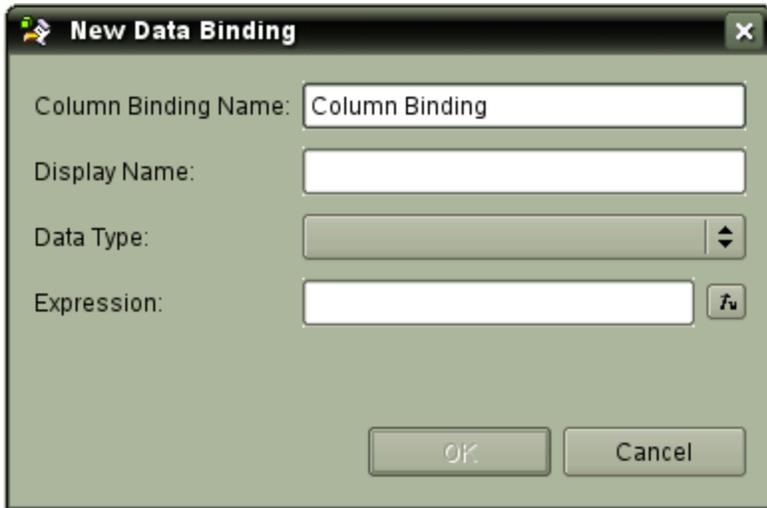
Por ejemplo es posible agregar una Tabla, como se describe en la sección [Especificando los datos a utilizar](#)

Otra alternativa que se muestra a continuación es utilizar un elemento *Data* que no requiera pertenecer a ninguna Tabla

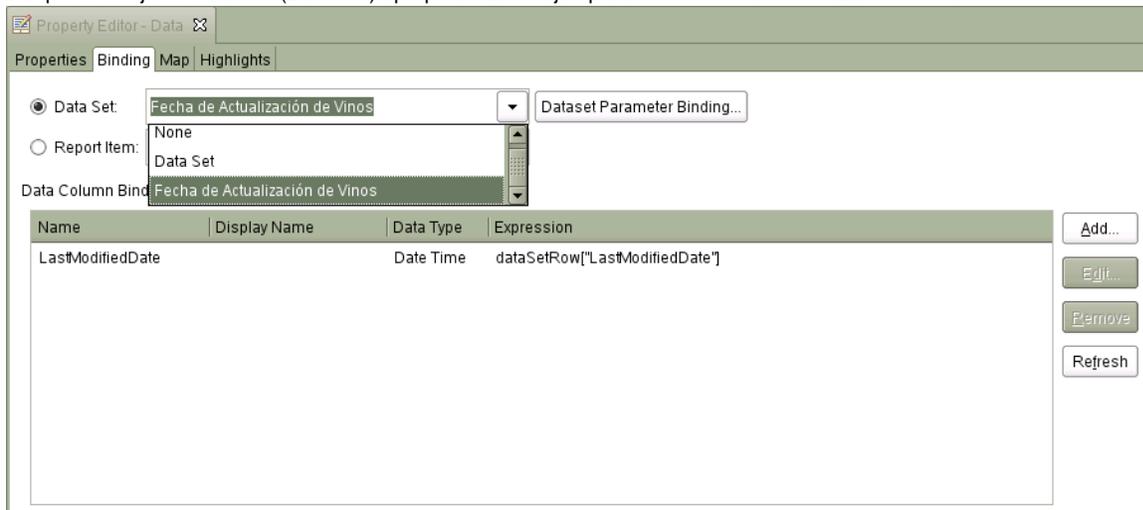
1. Arrastrar el item *Data* desde la Paleta hasta el área de diseño del reporte.



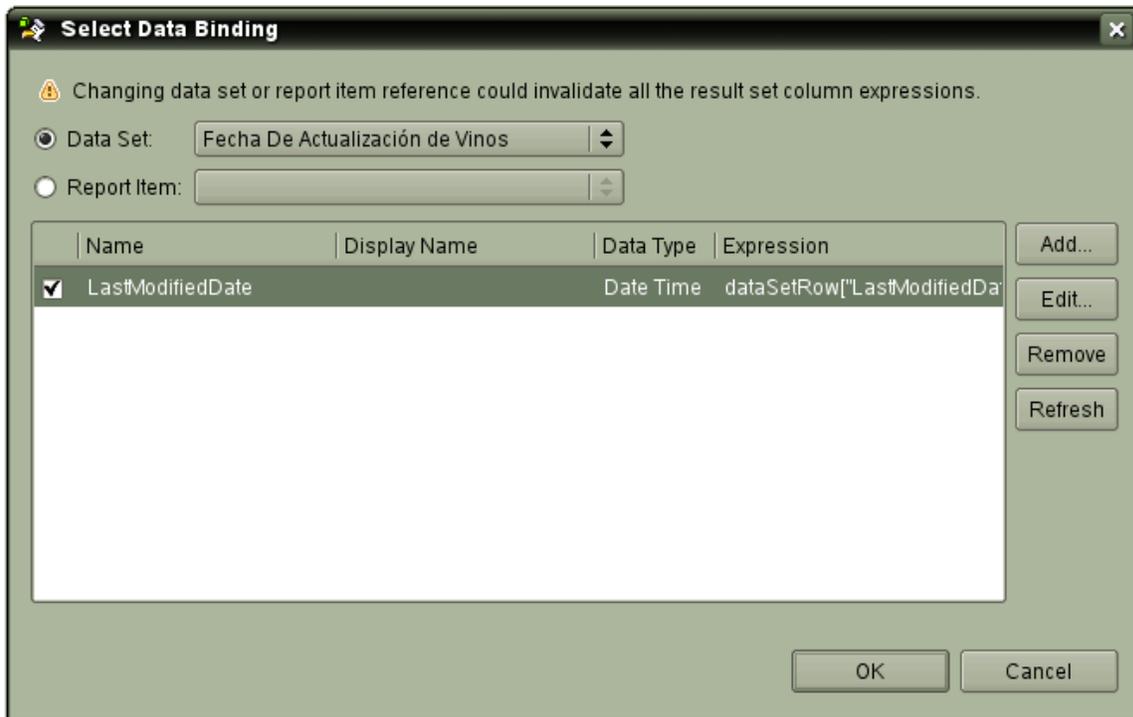
Esto mostrará automáticamente la siguiente ventana solicitando la información que se desea mostrar.



2. Cancele la edición en esta ventana ya que se requiere antes indicarle de qué Conjunto de datos (Data Set) se obtendrán los datos.
3. En la pantalla de edición de propiedades seleccione la opción Binding
4. Indique el Conjunto de datos (Data Set) apropiado. En el ejemplo sería "Fecha de Actualización de Vinos"



5. Finalmente resta indicar, para el elemento Data, qué columna del Conjunto de Datos contiene la información
6. Presione el botón derecho sobre el Data recién agregado, y seleccione la opción Change Data Column. Aparecerá la siguiente pantalla:



7. Active la casilla junto a la columna LastModifiedDate y acepte los cambios

Parametrizando el reporte

Los reportes pueden ser parametrizados con dos tipos de parámetros:

Parámetros simples y parámetros en cascada, en los siguientes links aparecen ejemplos prácticos de como usar cada tipo de parámetro.

- [Parametrización Simple](#)
En la parametrización simple los parámetros que se definen son parámetros directos sobre el reporte.
- [Parametrización en Cascada](#)
Por otra parte en la parametrización en cascada los parámetros dependen entre sí, siendo recalculados al modificar alguno.
- [Parametrización con Fechas](#)
La parametrización es muy útil para trabajar con fechas.

- [Parametrización Simple](#)
- [Parametrización en Cascada](#)
- [Parametrización con Fechas](#)
- [UserLogin como parámetro del reporte en ePortal](#)

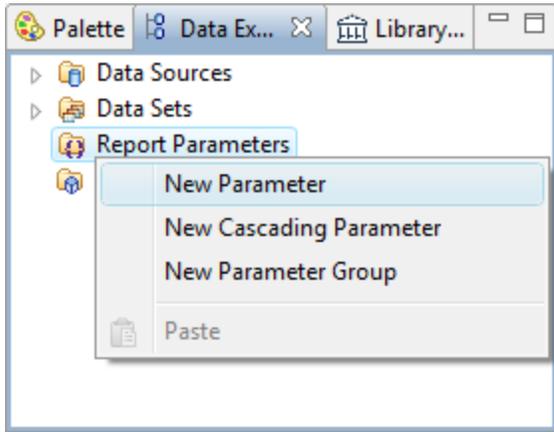
Parametrización Simple

Con frecuencia los reportes requieren cierto grado de parametrización para que el propio usuario pueda filtrar la información que se despliega, restringiendo así el análisis a subconjuntos de la información.

En esta sección se describe como agregar un parámetro al reporte que permita desplegar los datos de un único vendedor sin que esto afecte la estructura del reporte.

Creando un nuevo parámetro

1. Desde el panel Data Explorer presionar el botón derecho del mouse sobre el elemento Report Parameters
2. Crear un nuevo parámetro



Esta acción abre el editor de parámetros

3. Indique los siguientes datos del parámetro

Name:	El nombre del parámetro En nuestro caso Vendedor
Data type:	Tipo de dato del parámetro En nuestro caso String
Display Type:	Forma de mostrar los datos En nuestro caso List Box

4. La selección de la forma en que se despliegan los datos como **List Box** despliega un conjunto de opciones adicionales propias de esta forma de visualización.

La lista desplegable (list box) podrá contener opciones estáticas o dinámicas.

El caso **estático** permite al diseñador del reporte indicar de forma estática la lista de elementos que se incluirán en la lista.

El caso **dinámico** permite al diseñador definir una consulta que cargue la lista con datos almacenados, por ejemplo, en un cubo de O3

5. Indique el tipo de lista dinámica

Este tipo de parámetro requiere de la existencia de un Conjunto de Datos (Data Set) desde donde se tomarán los valores para cargar la lista.

- Utilizando el botón **Create New...** cree un nuevo Conjunto de datos como se indica a continuación:

Name	Vendedores
Data Source	O3 Local MDX
Data Set Type	O3 Data Set

- Ingrese la siguiente consulta MDX en la especificación del Data Set

```
SELECT {[Vendedores].[Vendedor].members} ON COLUMNS
FROM [Vinos]
```

 Para más detalles sobre la definición de un **Data Set** consultar la sección [Especificando los datos a utilizar](#)

- Acepte los cambios del editor del Data Set
- Complete los datos del parámetro con la nueva información:

Data Set:	Conjunto de datos al cual está asociado el parámetro. En nuestro caso usaremos el conjunto de datos Vendedores
Select value column:	Campo del Data Set que contiene la clave del parámetro. En nuestro caso usaremos unique_name

Select display text:

Campo del Data Set que contiene la etiqueta del parámetro a mostrar al usuario
En nuestro caso **label**

10. Finalmente el editor debería mostrar lo siguiente

11. Presione el botón **OK** para concluir la creación del Parámetro. Este aparecerá en la ventana **Data Explorer** bajo **Report Parameters**.

En este punto es posible ejecutar el reporte para ver cómo se muestra una ventana donde se solicita el valor del parámetro de entre una lista de opciones que son obtenidas del cubo de O3

Filtrar Consulta

Utilizando el Parámetro en la consulta

Una vez definido el parámetro es necesario utilizarlo en la consulta para que filtre la misma y devuelva únicamente el Vendedor seleccionado por el usuario.

Para ello modificaremos el Conjunto de Datos (Data Set) definido en la sección **Especificando los datos a utilizar**.

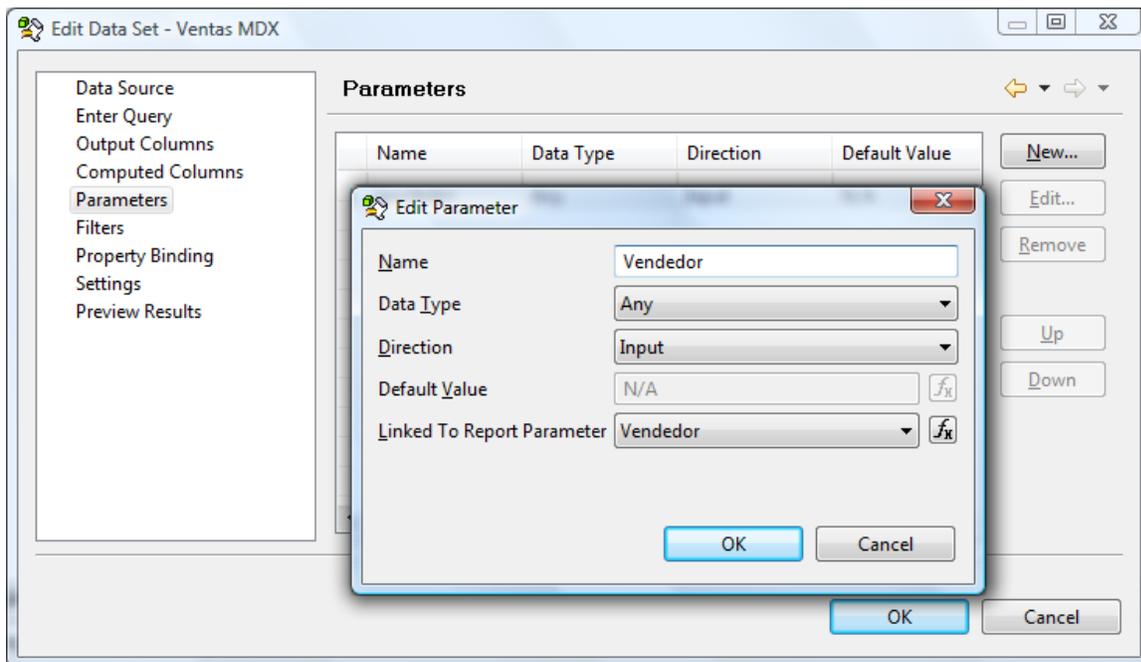
1. Desde el Data Explorer, hacer doble click sobre el Conjunto de Datos **Ventas MDX**.
2. Modificar la consulta MDX para que contenga lo siguiente

```
SELECT
  {[Measures].[Litros Vendidos], [Measures].[Venta Neta]} ON COLUMNS
  ,CROSSJOIN({[Vendedores].[Vendedor].?}, {[Vinos].[Vino].members}) ON ROWS
FROM [Vinos]
```

Notar el signo de interrogación en la sección correspondiente a las filas. Este signo marca el lugar donde se sustituirá el valor del parámetro cuando se ejecute la consulta.

3. Seleccione **Parameters** en el editor del Data Set
4. Agregue un nuevo parámetro con el botón **New...**
5. Configure el parámetro de acuerdo a lo siguiente

Name:	El nombre del parámetro En nuestro caso Vendedor
Data Type:	Tipo del parámetro En nuestro caso no requiere modificación
Direction:	Dirección del parámetro (Entrada o Salida) En nuestro caso usaremos input
Default Value:	Valor que tendrá el parámetro por defecto En nuestro caso N/A
Linked To Report Parameter:	Parámetro al cual está asociado En nuestro caso Vendedor



6. Finalmente resta ejecutar el reporte para ver cómo se filtran los datos para mostrar únicamente las ventas del Vendedor seleccionado

Utilizando un filtro en la consulta

Una vez definido el parámetro es necesario utilizarlo en la consulta para que filtre la misma y devuelva únicamente el Vendedor seleccionado por el usuario.

Para ello modificaremos el Conjunto de Datos (Data Set) definido en la sección [Especificando los datos a utilizar](#).

1. Desde el Data Explorer, hacer doble click sobre el Conjunto de Datos **Ventas MDX**.
2. Modificar la consulta MDX para que contenga lo siguiente

```
SELECT
  {[Measures].[Litros Vendidos], [Measures].[Venta Neta]} ON COLUMNS
  ,CROSSJOIN({[Vendedores].[Vendedor].members}, {[Vinos].[Vino].members}) ON ROWS
FROM [Vinos]
```

3. Seleccione **Filters** en el editor del Data Set
4. Agregue un nuevo filtro con el botón **New...**
5. Configure el filtro de acuerdo a lo siguiente

Expression:	Definición de la expresión. En nuestro caso row["Vendedores_Vendedor"]
Operator:	Lista de operadores. En nuestro caso Equals o Match
Value 1:	Valor a comparar con la expresión. En nuestro caso usaremos params["Vendedor"].value

6. Modificar en la definición del parámetro **Select value column unique_name** por **label**.
7. Finalmente resta ejecutar el reporte para ver cómo se filtran las ventas para el Vendedor seleccionado.

Parametrización en Cascada

En muchos casos, en los que el usuario debe elegir un valor entre una larga lista para un parámetro, es conveniente ir aproximando la búsqueda en varios pasos.

En el caso que se desarrolla en este tutorial, hemos definido hasta el momento un parámetro para filtrar el reporte para un vendedor determinado. Una forma de facilitar la localización del vendedor es catalogarlo primero entre Distribuidor y Venta Directa.

De esta forma se podría solicitar primero el Canal para luego elegir el vendedor de la lista de vendedores de ese canal.

Esta sección explica cómo establecer esta dependencia entre los parámetros a través de **Parámetros en Cascada**.

Creando Parámetros en Cascada



Nota

Se recomienda haber leído antes la sección [Parametrización Simple](#)

1. Cree un conjunto de datos (Data Set) para construir la lista de Canales

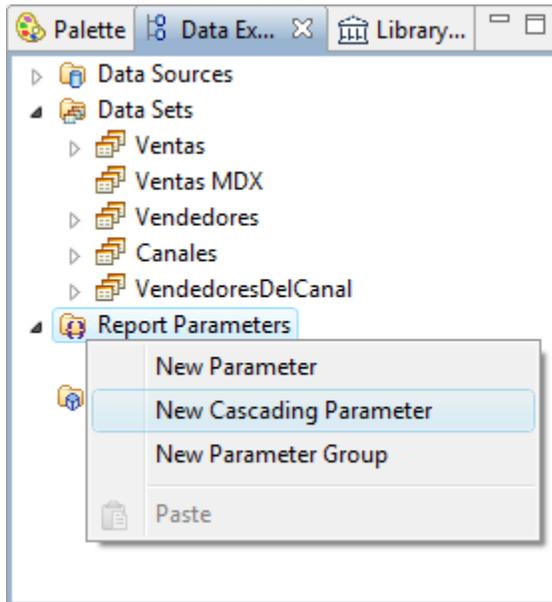
Name:	Canales
Data Source:	O3 Local MDX
Data Set Type:	O3 Data Set
Query	<pre>SELECT {[Vendedores].[Canal].members} ON COLUMNS FROM [Vinos]</pre>

2. Cree un conjunto de datos (Data Set) para construir la lista de Vendedores de cada Canal

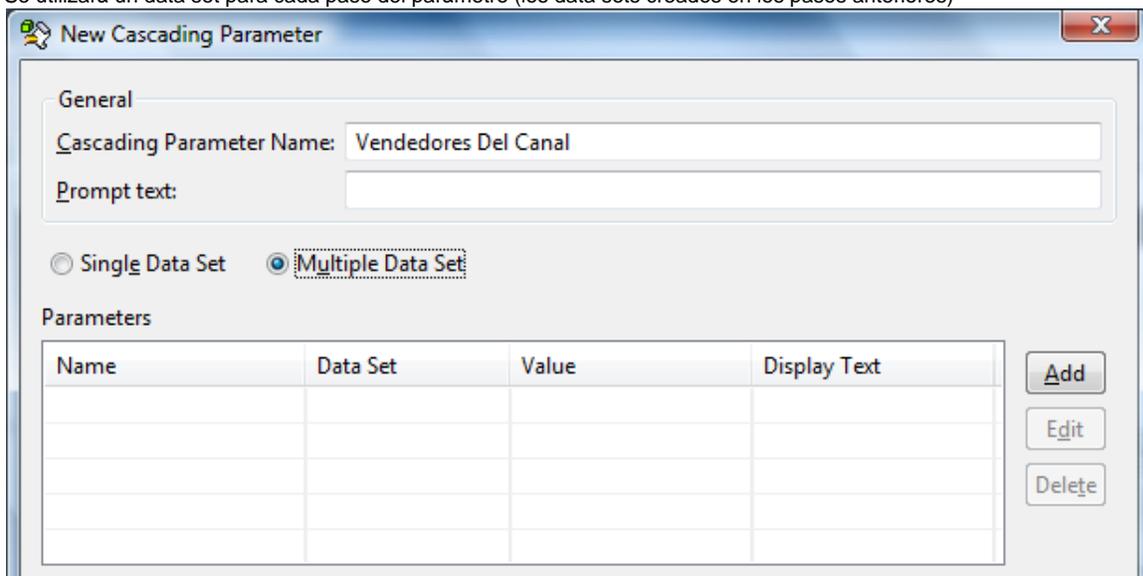
Name:	VendedoresDelCanal
--------------	--------------------

Data Source:	O3 Local MDX
Data Set Type:	O3 Data Set
Query	<pre>SELECT {[Vendedores].?.children} ON COLUMNS FROM [Vinos]</pre> <p>El signo de interrogación será luego asociado al parámetro que identifique el Canal. Ignorar cualquier mensaje de error que se muestre debido a que no puede resolver el parámetro</p>

- Desde el panel Data Explorer presionar el botón derecho del mouse sobre el elemento Report Parameters
- Crear un nuevo Parámetro en Cascada



- Ingrese **Vendedores Del Canal** como nombre del Parámetro
 - Seleccione **Multiple Data Set**
- Se utilizará un data set para cada paso del parámetro (los data sets creados en los pasos anteriores)



- Presione el botón **Add** para agregar un parámetro a la cascada.
- Ingrese los siguientes datos para el primer Parámetro:

Name:	Canal
Data Set:	Canales
Value:	unique_name
Display Text:	label

9. Presione el botón **Add** para agregar un segundo parámetro a la cascada.
10. Ingrese los siguientes datos para el segundo Parámetro:

Name:	VendedorDelCanal
Data Set:	VendedoresDelCanal
Value:	unique_name
Display Text:	label

11. El editor del parámetro en cascada debería quedar como se muestra

Edit Cascading Parameter

General

Cascading Parameter Name:

Prompt text:

Single Data Set Multiple Data Set

Parameters

Name	Data Set	Value	Display Text
Canal	Canales	unique_name	label
VendedorDelCanal	VendedoresDelC...	unique_name	label

Properties

Name:

Prompt text:

Data type:

Display type:

Default value:

More Options

Help Text:

Format as:

List Limit: values Is Required

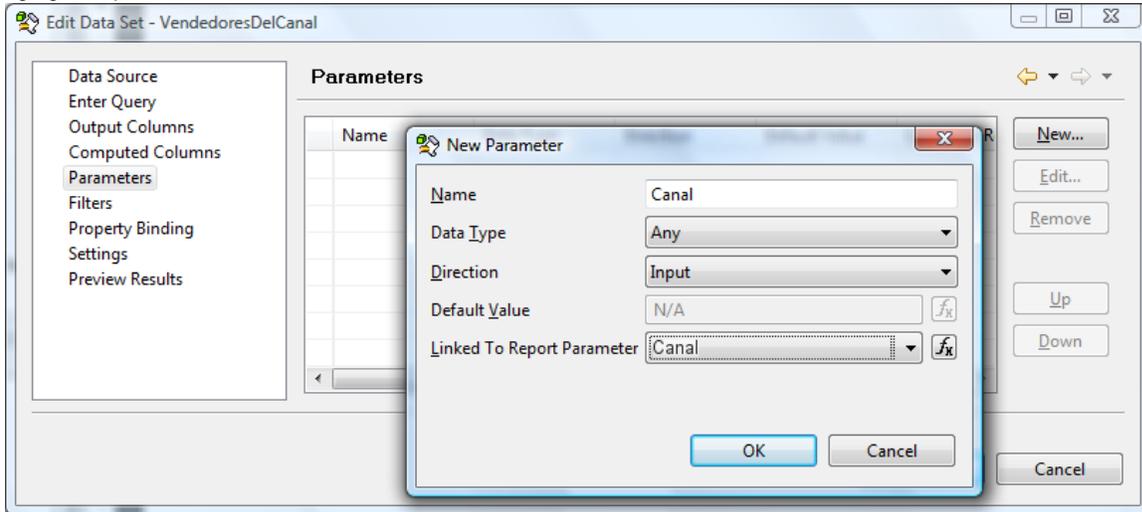
12. Para continuar presione **OK**.

Relacionando los Parámetros de la Cascada

Hasta este punto se definió una Cascada de 2 Parámetros y 2 Conjuntos de datos, uno para cada parámetro de la cascada.

Lo que resta es relacionar los parámetros para que uno actúe sobre el otro.

1. Editar el Conjunto de Datos (Data Set) **VendedoresDelCanal**
2. Agregar un parámetro tal como se muestra a continuación



Probando lo realizado

Para probar lo realizado cree un nuevo **Data Set** con los siguientes datos:

- * **Name:** Ventas por Canal y Vendedor
- * **Data Source:** O3 MDX View
- * **Data Set Type:** O3 Data Set
- * **Query** La siguiente expresion en **MDX**:

```
SELECT
    {Measures.[Litros Vendidos]} ON COLUMNS,
    {Vinos.[Vino].members} ON ROWS
FROM Vinos
WHERE ([Vendedores].[?])
```



Nota

En la consulta aparece un signo de interrogación, lo que indica es que en ese lugar hay un parámetro. Agregue el parámetro **Vendedor Por Canal** en el conjunto de datos como se indica en [Parametrización Simple](#)

Cree una tabla con la información de Ventas por Canal y Vendedor como se explica en [Diseñando la forma del reporte](#)

Seleccione **Preview** y aparecerá la siguiente pantalla:

Enter Parameters

Parameters marked with * are required.

{ } Vendedor: *

Federico Moreira

{ } Canal: *

{ } Vendedor Por Canal: *

OK Cancel

En **Canal** aparecerán todos los canales de vendedores del cubo de **Vinos**, seleccione un canal y automáticamente se actualizará **Vendedor Por Canal**.

En **Vendedor Por Canal** aparecerán todos los vendedores del canal seleccionado, seleccione un vendedor y presione **OK** para ver el detalle de los litros de vinos vendidos por ese vendedor.

Por ejemplo si elige como canal "Venta Directa" y como vendedor "Federico Moreira" el reporte mostrará:

Vino	Litros Vendidos
Cabernet Sauvignon, Merlot	4500
Cabernet, Tannat, Merlot	0
Chenin, Gewurzt	2250
...	...
Ugni Blanco	0

Si usted desea cambiar el valor del parámetro deberá hacer click en **Show Report Parameters** y se volverá a abrir la ventana anterior, pudiendo elegir un nuevo canal y vendedor.

Parametrización con Fechas

Esta sección describe como agregar un parámetros que manejen fechas.

El ejemplo terminado puede consultarse aquí.

Creando Parámetros sobre Fechas



Nota

Se recomienda haber leído antes [Parametrización en Cascada](#).

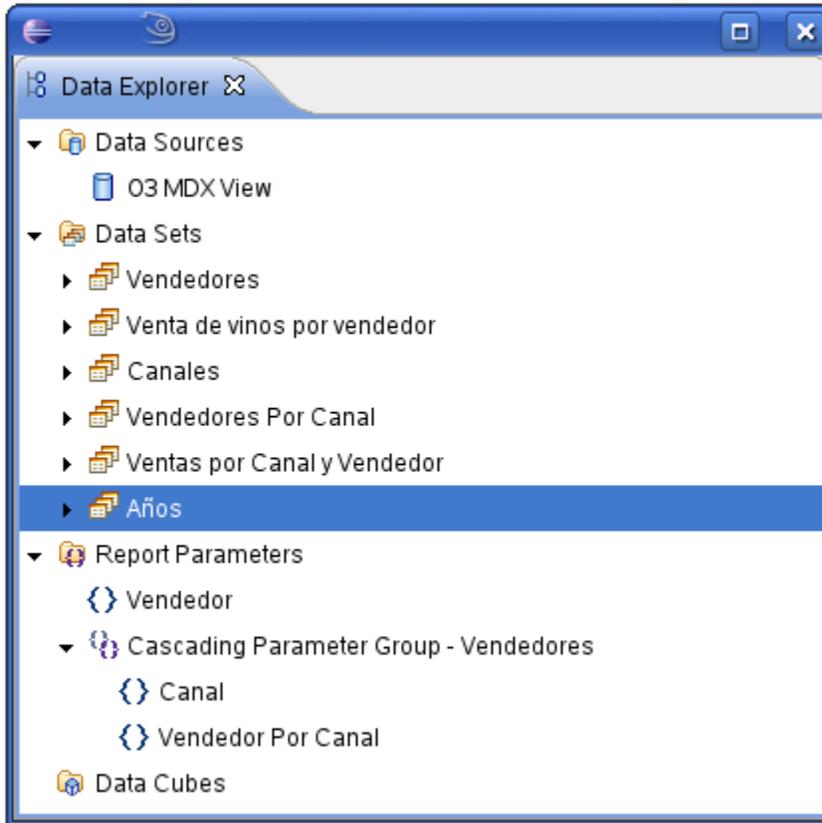
En el siguiente ejemplo se muestra cómo definir un reporte donde ver litros de vinos vendidos por todos los vendedores, parametrizando en los meses.

Primero cree un conjunto de datos (**Data Set**) que contenga la siguiente información:

- * **Name:** Años
- * **Data Source:** O3 MDX View
- * **Data Set Type:** O3 Data Set
- * **Query** La siguiente expresión en **MDX:**

```
SELECT {[Fecha].[Año].members} ON COLUMNS  
FROM [Vinos]
```

Una vez definido el **Data Set** se podrá visualizar el mismo desde la vista **Data Explorer** (Explorador de datos) del O3Reporting:



Para continuar realice los siguientes pasos:

1. Haga click derecho en **Report Parameters** y elija la opción **New Cascading Parameter** y cree una cascada de parametros con nombre **Fechas**.

New Cascading Parameter

General

Cascading Parameter Name:

Prompt text:

Single Data Set Multiple Data Set

Parameters

Name	Data Set	Value	Display Text

Add Edit Delete

Properties

Name:

Prompt text:

Data type:

Display type:

Default value:

More Options

Help Text:

Format as:

List Limit: values Is Required

OK Cancel

2. Presione **Add** para agregar un parámetro a la cascada.
3. Ingrese los siguientes datos:

Name:	El nombre del parámetro En nuestro caso Año
Data Set:	Conjunto de datos al cual está asociado el parámetro. En nuestro caso usaremos el conjunto de datos Años
Value:	Valor que tendrá el parámetro. En nuestro caso usaremos unique_name
Display Text:	Texto a mostrar en el repotye En nuestro caso label

Add Cascading Parameter

Name: Año

Data Set: Años

Value: unique_name

Display Text: label

OK Cancel

4. Presione **OK** para continuar, aparecerá la siguiente pantalla (Recuerde que **Multiple Data Set** debe estar seleccionado):

Edit Cascading Parameter

General

Cascading Parameter Name:

Prompt text:

Single Data Set Multiple Data Set

Parameters

Name	Data Set	Value	Display Text
Año	Años	unique_name	label

Properties

Name:

Prompt text:

Data type:

Display type:

Default value:

More Options

Help Text:

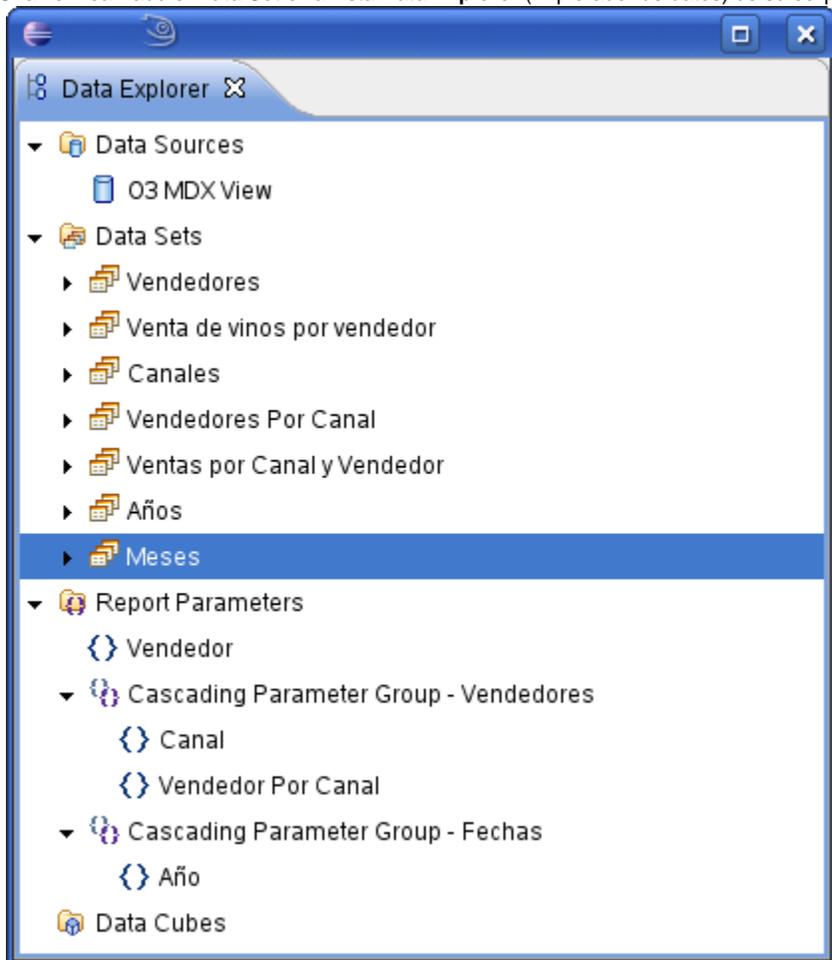
Format as:

List Limit: values Is Required

5. Presione **OK** para continuar.
6. Cree un conjunto de datos (**Data Set**) que contenga la siguiente información:
 - * **Name:** Meses
 - * **Data Source:** O3 MDX View
 - * **Data Set Type:** O3 Data Set
 - * **Query** La siguiente expresion en **MDX**

```
SELECT Descendants([Fecha].?, [Fecha].[Mes]) ON COLUMNS
FROM [Vinos]
```

7. Agregue el parámetro **Año** al conjunto de datos creados como se indica en [Parametrización Simple](#)
Una vez realizado el **Data Set** el la vista **Data Explorer** (Explorador de datos) de su eclipse deberá aparecer de la siguiente manera:



8. Haga click derecho sobre **Cascading Parameter Group - Fechas** y seleccione la opción **Edit**. Aparecerá la siguiente pantalla:

Edit Cascading Parameter

General

Cascading Parameter Name:

Prompt text:

Single Data Set Multiple Data Set

Parameters

Name	Data Set	Value	Display Text
Año	Años	unique_name	label

Add Edit Delete

Properties

Name:

Prompt text:

Data type:

Display type:

Default value:

More Options

Help Text:

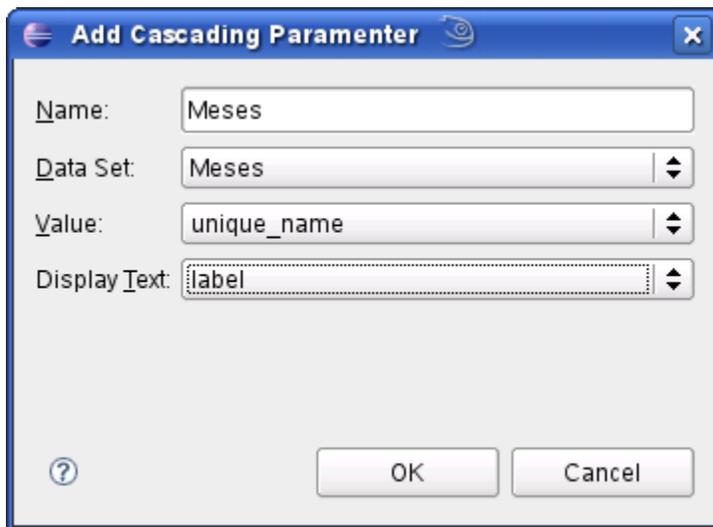
Format as:

List Limit: values Is Required

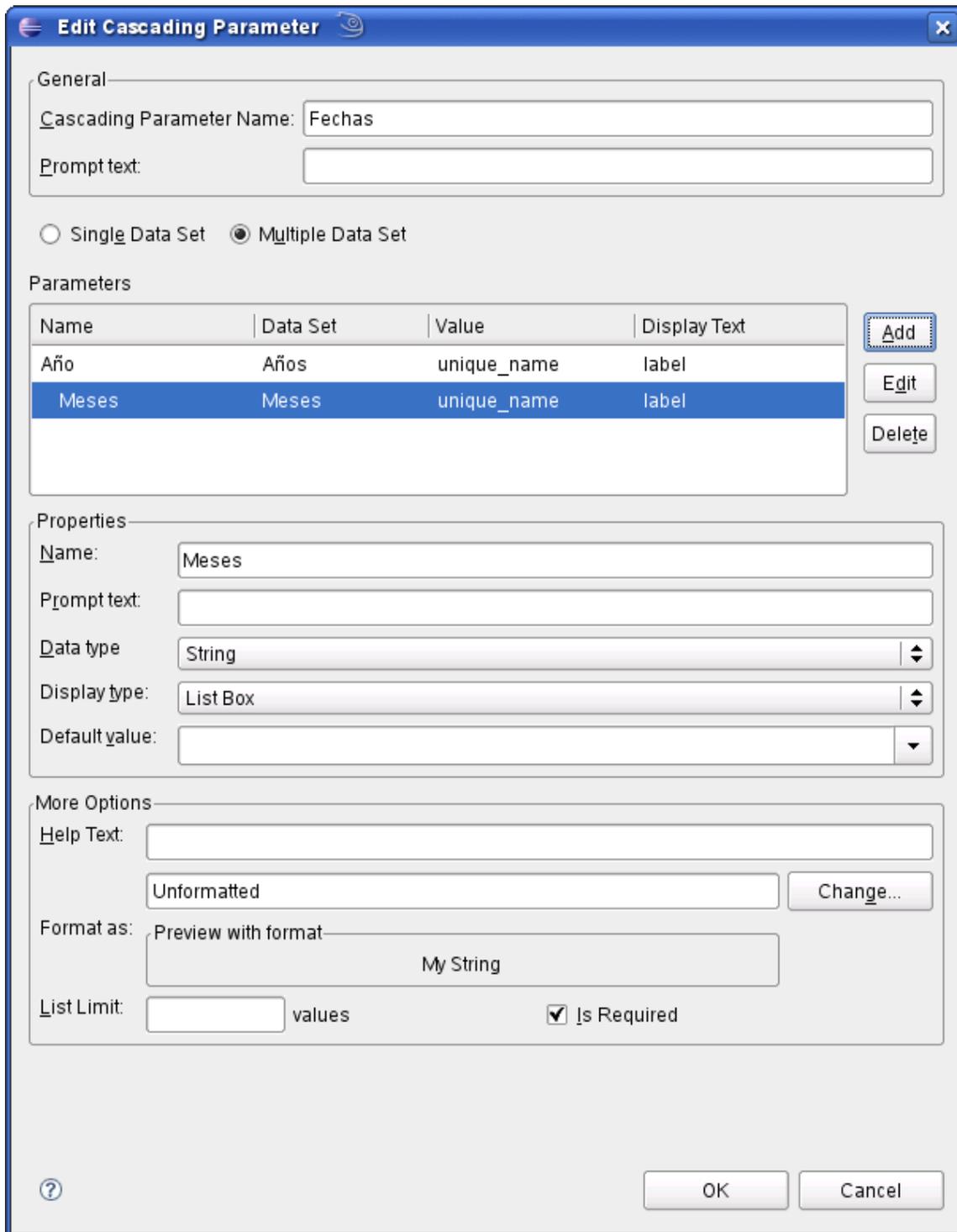
OK Cancel

9. Seleccione **Add** para agregar un nuevo parámetro a la cascada.
10. Ingrese los siguientes datos:

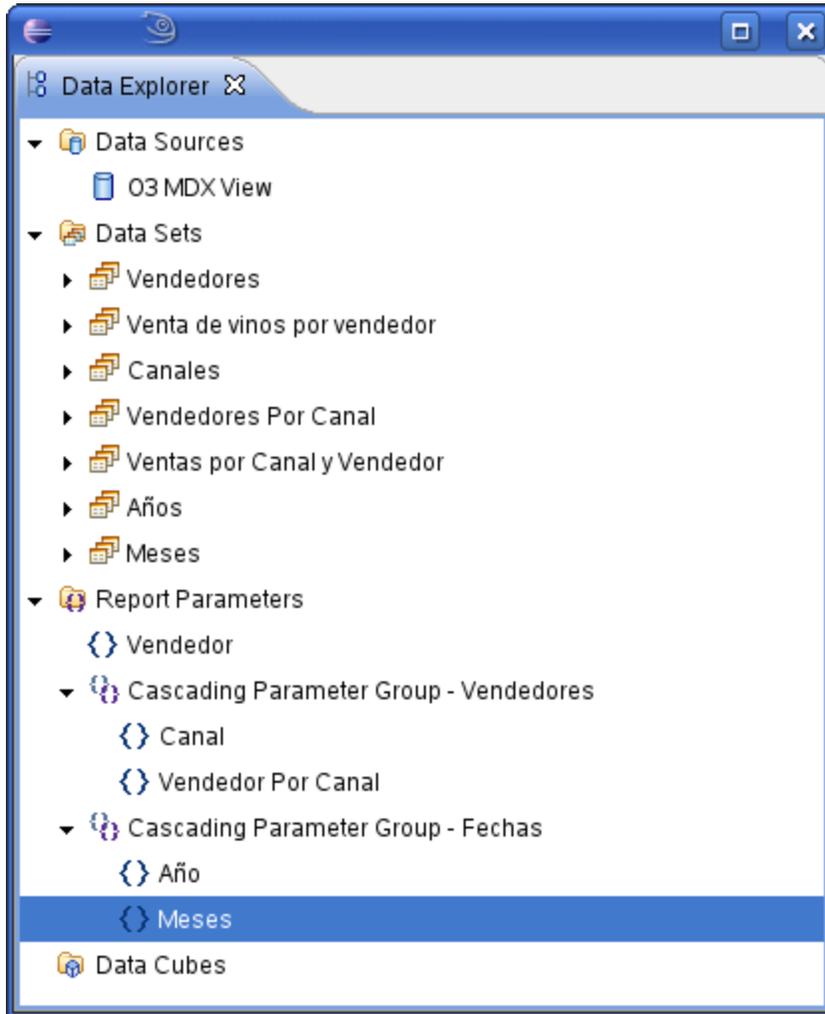
Name:	El nombre del parámetro En nuestro caso Mes
Data Set:	Conjunto de datos al cual está asociado el parámetro. En nuestro caso usaremos el conjunto de datos Meses
Value:	Valor que tendrá el parámetro. En nuestro caso usaremos unique_name
Display Text:	Texto a mostrar en el repotye En nuestro caso label



11. Para continuar presione **OK**. Aparecerá la siguiente pantalla:



12. Presione **OK** para continuar.
La vista **Data Explorer** (Explorador de datos) de su eclipse deberá aparecer de la siguiente manera:



13. Cree un conjunto de datos (**Data Set**) que contenga la siguiente información:

* **Name:** Ventas por Vendedor y Meses

* **Data Source:** O3 MDX View

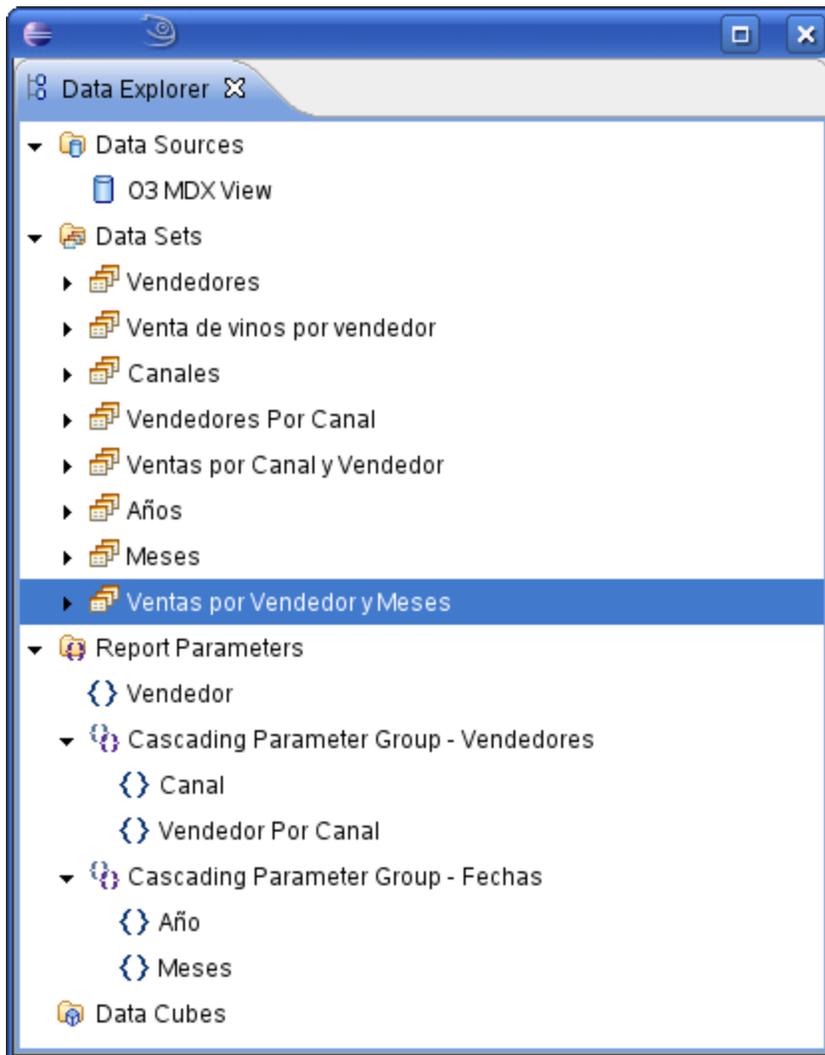
* **Data Set Type:** O3 Data Set

* **Query** La siguiente expresión en **MDX**:

```
SELECT
  {[Fecha].?} ON COLUMNS,
  {[Vendedores].[Vendedor].members} ON ROWS
FROM [Vinos]
WHERE Measures.[Litros Vendidos]
```

14. Agregue el parámetro **Meses** al conjunto de datos creados como se indica en [Parametrización Simple](#)

Una vez realizado el **Data Set** en la vista **Data Explorer** (Explorador de datos) de su eclipse deberá aparecer de la siguiente manera:



15. Cree una tabla con la información de Ventas por Vendedor y Meses como se explica en [Diseñando la forma del reporte](#)
16. Seleccione **Preview**. Aparecerá la siguiente pantalla:

Parameters marked with * are required.

{ } Vendedor: *
 Federico Moreira

{ } Canal: *
 Distribuidor

{ } Vendedor Por Canal: *
 Mykonos

{ } Año: *

{ } Meses: *

OK Cancel

17. Seleccione el **Año** para el cual desea ver los litros vendidos por los vendedores. Se actualizará automáticamente los valores de **Meses**.
18. Seleccione el **Mes** para el cual desea ver el reporte.
19. Seleccione **OK**

Por ejemplo si elige como **año** "2002" y como **mes** "mar / 2002" el reporte mostrará:

Vino	Litros Vendidos
G & P Corp.	5850
Link S.R.L.	1800
Mykonos	0
...	...
Susana Garcia	2700

Si se desea cambiar el valor del parámetro deberá hacer click en **Show Report Parameters** y se volverá a abrir la ventana anterior, pudiendo elegir un nuevo **año** y **mes**.

UserLogin como parámetro del reporte en ePortal

En el Enterprise Portal al ejecutar un reporte siempre se le pasa como parámetro el login del usuario que está logueado. Es posible filtrar la información de O3 tomando en cuenta este parámetro.

Esto es independiente de la aplicación de los permisos definidos en O3, los que se aplican siempre independientemente del envío del parámetro.

En el caso de que el usuario sea un vendedor y el reporte muestre datos de venta, es posible filtrar las ventas y mostrar solamente las del vendedor.

Procedimiento con un cubo de O3

Para esto es necesario contar con el login del usuario de alguna forma en el cubo, que puede ser teniendo una dimensión con el login (en este caso, la dimensión **no** debe definirse como oculta), o teniendo el login del vendedor como etiqueta larga del nivel vendedor.

- En el caso de tener una dimensión con el login, en O3 Report se puede realizar una consulta MDX o una vista, siempre con la condición de que la dimensión login venga en los datos.

- En el caso de setear el login como etiqueta larga, es necesario definir una vista utilizando las etiquetas largas, y que la misma cuente con la dimensión vendedor.

En ambos casos primero se debe crear el parámetro del reporte con el nombre "userLogin", ya que así se llama el parámetro que le pasa el Enterprise Portal. El mismo debe ser definido como oculto (hidden).

Luego se crea un data set para que traiga los datos, de una forma o de otra, y en el mismo data set se aplica un filtro para que el parametro login coincida con el login de los datos.

Consulta a una base de datos

En caso de que el reporte obtenga los datos de una consulta a la base, es posible filtrar los datos en la consulta a la base, i.e.: que la base aplique el filtro y traer en el data set solamente los datos del usuario. Esta opción no está disponible en la consulta a un cubo de O3, donde los datos siempre son filtrados por el reporte luego de recuperados.

En este caso es posible aplicar un parámetro al data set, vinculándolo con el parámetro del reporte, para luego filtrar en la consulta con el caracter '?', de esta forma:

Consulta de ventas

```
select cliente, monto
from Ventas
where vendedor_login=?
group by cliente
```

Restricciones

El envío del parámetro userLogin se realiza solamente en el Enterprise Portal, **no** se realiza en O3 Web. De todas formas, siempre se aplican los permisos definidos sobre los cubos en O3.

Diseñando la forma del reporte

Ahora que ya se definieron los datos que se utilizarán en el reporte, así como los mecanismos para obtenerlos, es posible diseñar la estructura del reporte distribuyendo los datos en la página.

Todo reporte se organiza en diferentes secciones y estas secciones se ubican una a continuación de la otra en forma vertical.

Cada sección a su vez incluye un contenedor que es un elemento gráfico donde se ubica el resto de los componentes del reporte, como ser etiquetas, datos, imágenes, etc.

Existen 3 tipos de contenedores

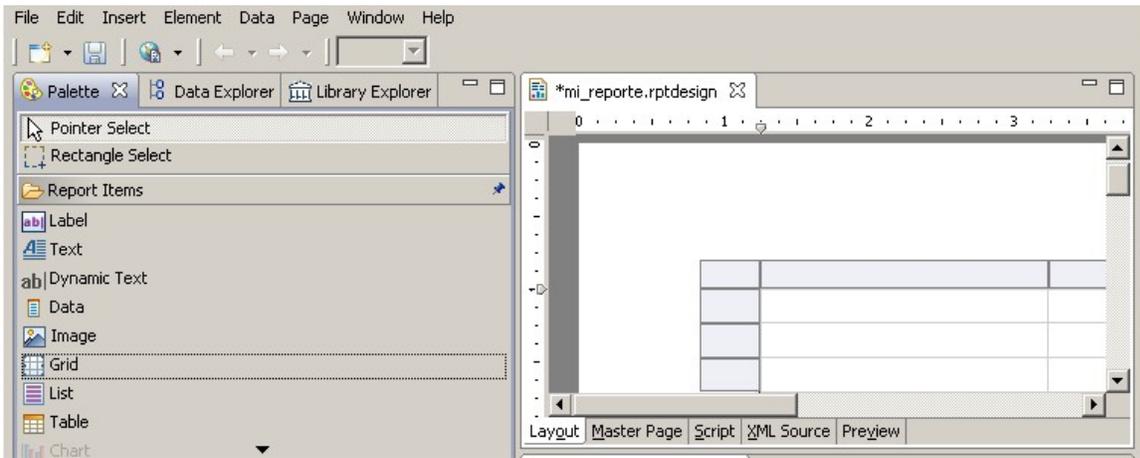
- Grilla
- Tabla
- Lista

En este tutorial se utiliza una Tabla para ubicar los datos del reporte dada su flexibilidad y facilidad de uso. En particular la tabla permite:

- Iterar a través de todos los registros provenientes de un Conjunto de Datos
- Ubicar la información de una manera muy sencilla en un formato de filas y columnas

1. Active la Paleta (Pallette)

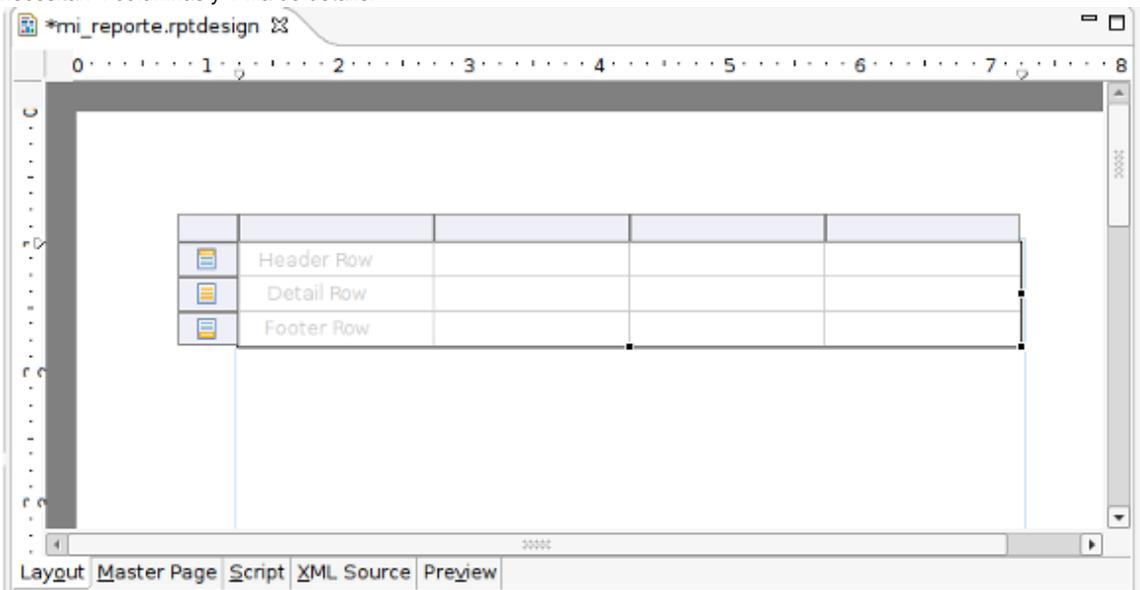
La Paleta enumera todos los elementos que se pueden distribuir en un reporte



Observar que el cuadrante de arriba a la izquierda contiene tres tab: Palette, Data Explorer, Library Explorer

2. Arrastre una tabla desde la paleta a la hoja donde se irá definiendo el reporte.

Esta acción solicita la cantidad de columnas que tendrá la tabla y la cantidad de filas de detalle necesarias. Para este ejemplo se necesitan 4 columnas y 1 fila de detalle.

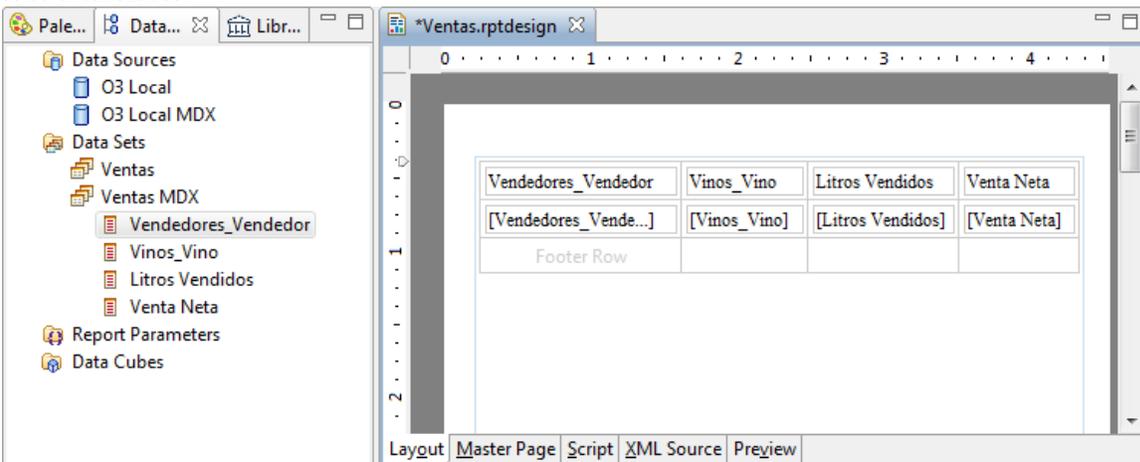


1. Active el Explorador de Datos (Data Explorer)

El Explorador de Datos permite navegar por los datos disponibles para el reporte. En particular muestra las Fuentes de Datos y Conjuntos de Datos definidos en los pasos anteriores (O3 Local MDX y Ventas MDX respectivamente).

2. Expanda el Conjunto de Datos **Ventas MDX** definido para este reporte. Es posible ver los campos incluidos en dicho conjunto.
3. Arrastre cada uno de los campos para soltarlos sobre las celdas de detalle de la tabla (Detail Row).

Esta acción irá poblando la tabla y agregando en la fila de encabezado un título para cada columna. Más tarde será posible cambiar estos encabezados.



De esta forma sencilla queda definido el reporte en su formato más básico con los datos obtenidos desde un cubo de O3.

Para ver cómo quedaría el reporte una vez ejecutado es posible seleccionar el tab **Preview** de la ventana de reporte o el botón **View Report** de la barra de herramientas (File >> View Report >> View report in web viewer)

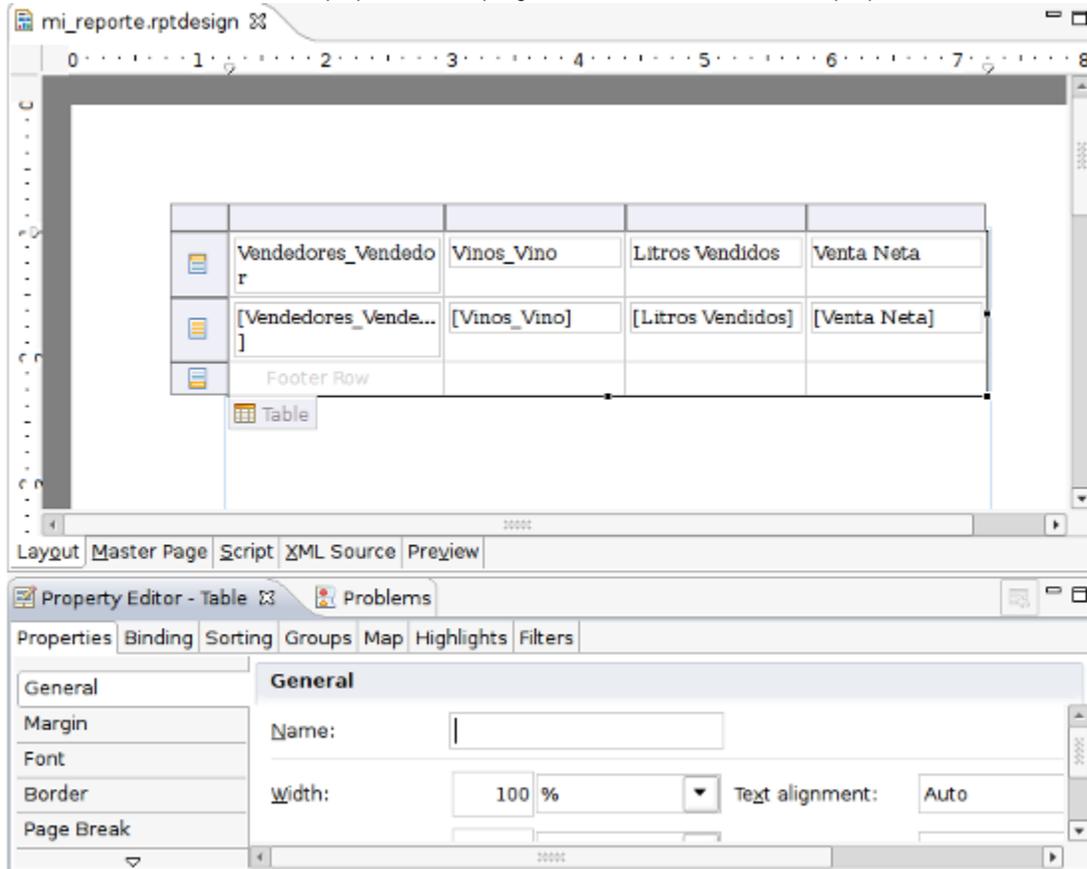
Ordenando los registros del reporte

Cuando la información llega desde la consulta, ésta no necesariamente viene ordenada. Al momento de definir el reporte es posible indicar un criterio de ordenación de los datos diferente del establecido en la propia consulta o vista.

Para esto se usa el editor de propiedades.

1. Seleccionar la tabla. Esto se hace haciendo click sobre la solapa con la etiqueta **Table** que aparece cuando se mueve el mouse por encima de la tabla

Al seleccionar la tabla el editor de propiedades despliega todos los atributos de la tabla que pueden ser modificados



2. Seleccione el tab **Sorting** para habilitar las opciones de ordenamiento de los registros de la tabla
3. Presione el botón **Add** para agregar un nuevo criterio de ordenamiento
4. Seleccione el campo por el cual se ordenarán los registros, por ejemplo Litros Vendidos

Es posible agregar varios campos al criterio de ordenamiento

El resultado de aplicar el nuevo orden puede observarse viendo una vista preliminar del reporte.

Agrupando registros del reporte

Muchas veces es deseable construir reportes que agrupan un conjunto de registros para facilitar su lectura o para mostrar datos adicionales para cada grupo.

En el caso que se expone en este tutorial parece natural presentar el reporte agrupando las ventas de vino por Vendedor, de modo que el nombre del Vendedor aparezca una única vez como encabezado del grupo y luego se muestre el detalle de los vinos vendidos por dicho Vendedor.

Esta operación de agrupar registros dentro de un reporte comunmente se conoce como agregar **Cortes de Control**.

En el ejemplo generaremos un corte de control para cada Vendedor

1. Seleccione la tabla
2. Seleccione el tab **Groups** del Editor de Propiedades para habilitar las opciones de agrupamiento
3. Presione el botón **Add** para agregar un nuevo criterio de agrupamiento

- Ingrese Vendedores como nombre del grupo
- Seleccione el campo Vendedor (Vendedores_Vendedor) como criterio de agrupamiento. Adicionalmente se puede elegir el orden en que estos grupos se despliegan (ascendente o descendente)

Group Details

Name:

Group On:

Interval: **Range:**

Use fixed base value for interval:

Hide Detail

Bookmark:

Sort Direction

Ascending
 Descending

Page break

Before:
After:
Inside:

Repeat Header

- Acepte los cambios en el editor de grupos
Esta acción agrega una nueva fila de encabezado y de pie para el nuevo grupo y coloca en el encabezado el campo correspondiente al criterio de agrupamiento, en este caso el campo Vendedores_Vendedor

Vendedores_Vendedor	Vinos_Vino	Litros Vendidos	Venta Neta
[Vendedores_Vende...]			
[Vendedores_Vende...]	[Vinos_Vino]	[Litros Vendidos]	[Venta Neta]
Group Footer Row			
Footer Row			

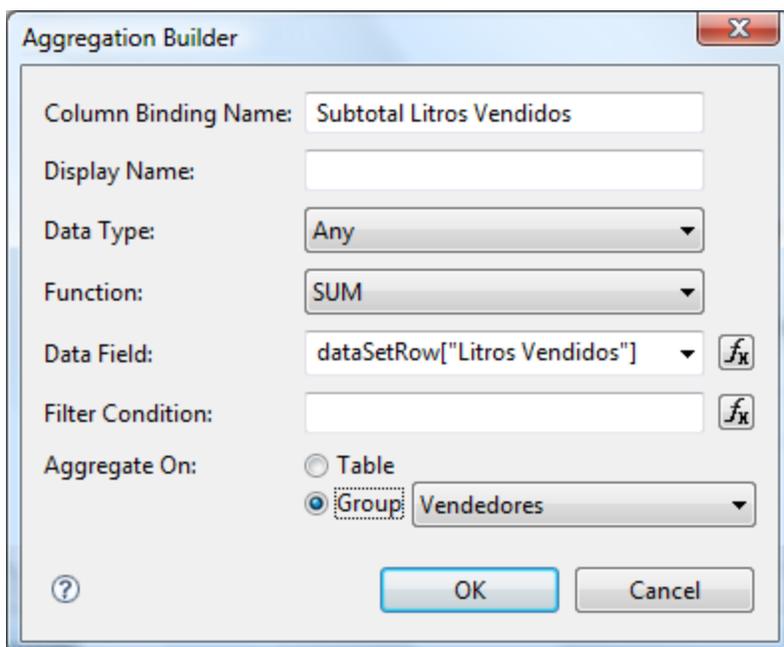
- Al hacer esto tanto el campo de Vendedor en la fila de detalle, como el título de vendedor en el encabezado de la tabla, carecen de sentido por lo que se pueden eliminar

Header Row	Vinos_Vino	Litros Vendidos	Venta Neta
[Vendedores_Vende...]			
Detail Row	[Vinos_Vino]	[Litros Vendidos]	[Venta Neta]
Group Footer Row			
Footer Row			

Agregando Totales Parciales

El paso siguiente consiste en agregar los subtotales por Vendedor tanto para los Litros Vendidos como para la Venta Neta

- Seleccionar la Paleta (Palette)
- Arrastrar una Agregación (Aggregation) en la fila correspondiente al pie del grupo (Group Footer Row) justo debajo de Litros Vendidos
Esto despliega el Editor de Agregaciones donde se define un nuevo lazo (binding) para el subtotal deseado.
Los binding permiten asignarle un nombre a un dato cualquiera o cálculo lo cual facilita la tarea de ubicarlo luego en el reporte
- Ingresar **Subtotal Litros Vendidos** como nombre del binding.
- Seleccionar el campo Litros Vendidos como campo de datos (Data Field)
- Seleccionar el grupo Vendedores como grupo de agregación (Aggregate On)



6. Al aceptar los cambios aparece en la tabla un nuevo elemento totalizador de la columna de Litros Vendidos por Vendedor.
7. Repetir la misma operación para la Venta Neta

Header Row	Vinos_Vino	Litros Vendidos	Venta Neta
[Vendedores_Vende...]			
Detail Row	[Vinos_Vino]	[Litros Vendidos]	[Venta Neta]
Group Footer Row		Σ [Subtotal Litros ...]	Σ [Subtotal Venta N...]
Footer Row			

Formateando la información

En esta sección se detallan los pasos necesarios para hacer que el reporte luzca mejor. En particular se verá cómo

- Alinear las celdas de datos
- Formateo de elementos numéricos
- Formateo de los encabezados
- Ajustes a la fuente utilizada

Tal como se vio en secciones anteriores, al seleccionar un elemento del reporte, el Editor de Propiedades muestra atributos relacionados con el elemento seleccionado permitiendo así modificar su comportamiento o presentación.

Esto permite modificar el formato de cada uno de los elementos del reporte en forma independiente logrando una gran flexibilidad.

Sin embargo, también es posible ajustar el formato del reporte definiendo estilos que permitan controlar las propiedades de varios elementos similares en forma simultánea. Este tutorial explica cómo formatear el reporte haciendo uso de los estilos.

Definiendo Estilos

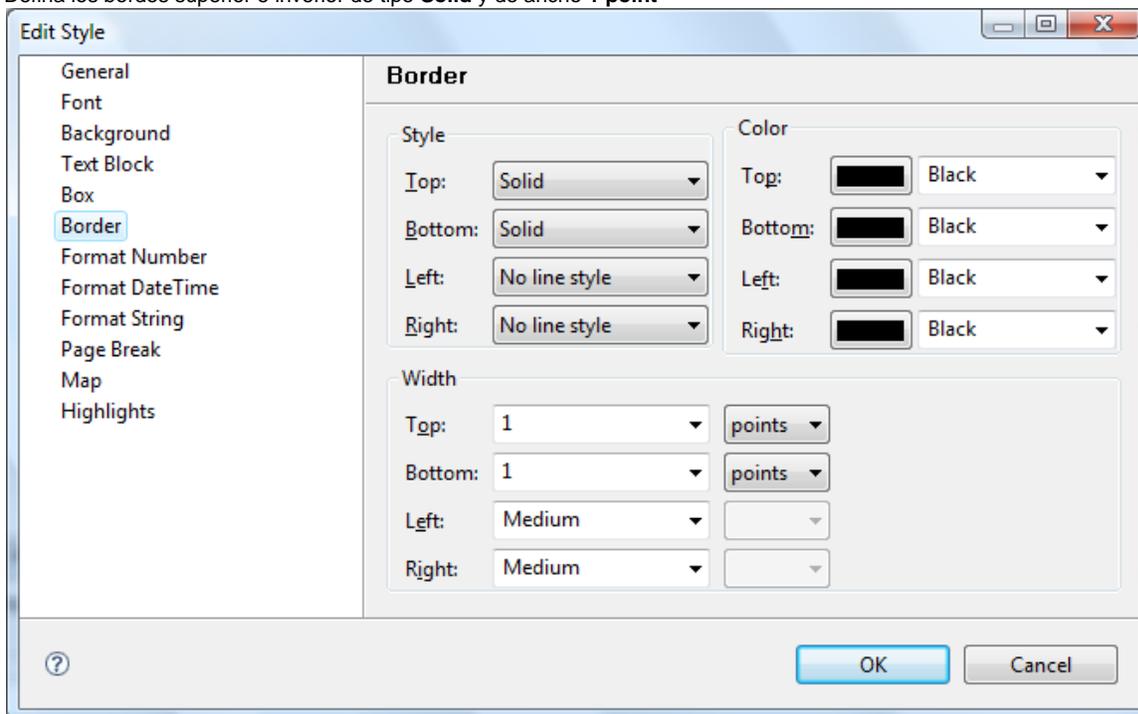
En el reporte construido hasta el momento se pueden identificar los siguientes tipos de elementos

- Títulos en el encabezado de la tabla
- Nombre del Vendedor en el encabezado del grupo de Vendedores
- Nombre del Vino en la fila de detalle
- Valores numéricos correspondientes a Cantidad de Litros y Venta Neta en la fila de detalle
- Subtotales para los valores numéricos

Esta clasificación sugiere la necesidad de contar con estilos diferentes para cada caso.

Estilo para el Encabezado de la Tabla

1. Seleccione la opción de menú **Element -> New Style...**
Esto despliega el editor de estilos. Este editor puede invocarse mediante otros mecanismos como por ejemplo botón derecho sobre el elemento al que luego se le asignará el estilo
2. Ingrese **Encabezado** como nombre del Estilo (Custom Style)
3. Seleccione el grupo de atributos **Text Block** a la izquierda
4. Seleccione la alineación del texto (Text alignment) centrada
5. Seleccione el grupo de atributos **Border**
6. Defina los bordes superior e inferior de tipo **Solid** y de ancho **1 point**



7. Acepte los cambios para cerrar el Editor de Estilos
Con esto se ha creado el nuevo estilo Encabezado el cual podrá asignarse a cualquier elemento del reporte
8. Seleccione la etiqueta **Vinos_Vino** del encabezado de la tabla
9. En el Editor de Propiedades de dicha etiqueta, en el grupo General, seleccione el estilo (style) recién creado
El reporte se ajustará para formatear la etiqueta de acuerdo al formato del estilo
10. Asigne el estilo a las etiquetas **Litros Vendidos** y **Venta Neta**

Estilo para el Encabezado del Grupo de Vendedores

1. Cree un nuevo estilo llamado Encabezado Grupo
2. Seleccione el grupo de atributos **Background** y defina como color de fondo RGB(240,240,240)
3. Asigne este estilo a la etiqueta correspondiente al nombre del Vendedor **Vendedores_Vendedor**

Hasta el momento el reporte debería verse similar a la siguiente figura

Header Row	Vinos_Vino	Litros Vendidos	Venta Neta
[Vendedores_Vende...]			
Detail Row	[Vinos_Vino]	[Litros Vendidos]	[Venta Neta]
Group Footer Row		Σ[Subtotal Litros ...]	Σ[Subtotal Venta N...]
Footer Row			

Juntando Celdas

El encabezado del grupo se vería más vistoso si abarcara el ancho total del reporte. Esto puede lograrse juntando celdas (Merge Cells)

1. Seleccione la celda donde se encuentra la etiqueta **Vendedores_Vendedor**
Notar que para seleccionar la celda debe hacerse click en el espacio disponible entre las líneas de división de la tabla y el borde de la etiqueta. Verificar en el Editor de Propiedades que se ha seleccionado la celda y no la etiqueta. El título de dicho editor debería decir **Properties Editor - Cell** y no **Properties Editor - Data**
2. Seleccione el resto de las celdas de la fila de encabezado de grupo mientras se mantiene presionada la tecla **SHIFT**

La selección se irá agrandando hasta abarcar todas las celdas de dicha fila

3. Presione el botón derecho del mouse y seleccione la opción **Merge Cells**

Las celdas de la fila de encabezado del grupo se juntarán de modo que quede una sola que abarque el ancho total de la tabla. El estilo con el color de fondo deja más en evidencia esto

Estilo para valores numéricos de detalle

1. Cree un nuevo estilo llamado Valor
2. Seleccione el grupo de propiedades **Text Block** y defina la alineación a la derecha
3. Seleccione el grupo de propiedades **Format Number**
4. Elija el formato predefinido **Fixed** con 2 decimales y active el checkbox del separador de miles
5. Asigne este nuevo estilo a los elementos Litros Vendidos y Venta Neta de la fila de detalle

Estilo para los subtotales

1. Cree un nuevo estilo llamado Subtotal
2. Ajuste la alineación a la derecha y el formato numérico similar al del estilo Valor
3. Defina un borde superior que de la idea de totalización
4. Asigne este nuevo estilo a los subtotales de Litros Vendidos y Venta Neta

Indentación de la fila de detalle

Por último ajustaremos el ancho de la primera columna para que tenga un tamaño fijo y dé la noción de indentación necesaria para identificar los vinos que corresponden a cada vendedor

1. Seleccione la tabla
Esto muestra selectores para las filas y columnas alrededor de la tabla
2. Seleccione la primera columna
3. Especifique un ancho de 1 cm para dicha columna en el Editor de Propiedades

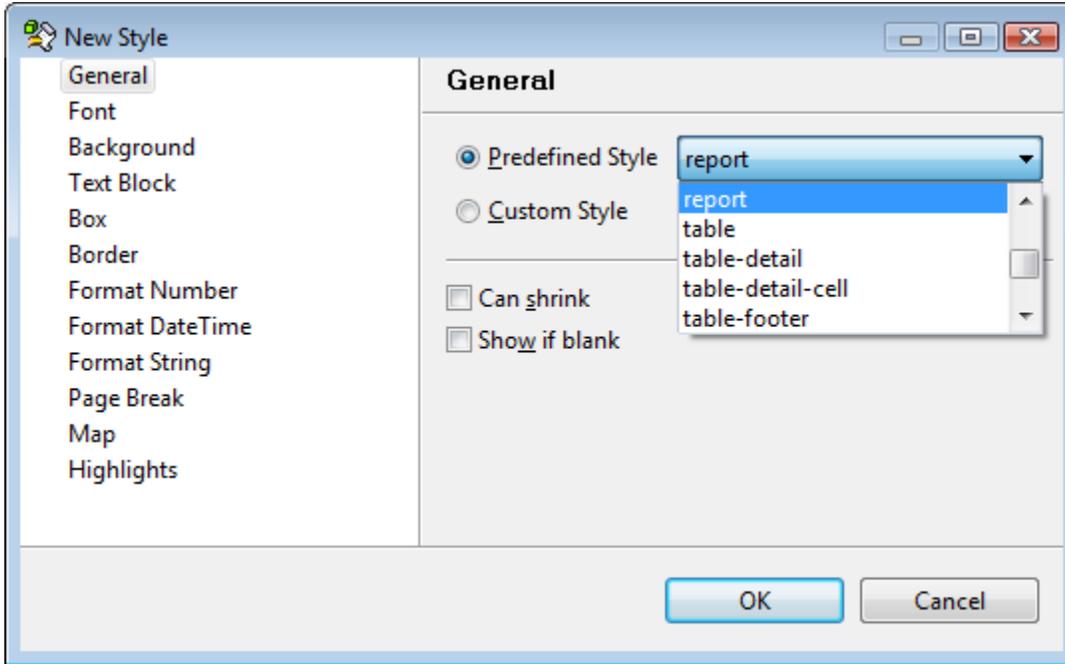
El reporte formateado debería verse más o menos así

Header R	Vinos_Vino	Litros Vendidos	Venta Neta
[Vendedores_Vende...]			
etail R	[Vinos_Vino]	[Litros Vendidos]	[Venta Neta]
o Foote		Σ [Subtotal Litros ...]	Σ [Subtotal Venta N...]
oter R			

Estilos predefinidos

Existe un conjunto de estilos ya predefinidos con la herramienta que permiten formatear elementos famosos del reporte. Tal es el caso de tablas, filas, celdas, etc.

A continuación se muestra el editor de estilos donde se ha seleccionado la opción de Estilo Predefinido, lo cual permite seleccionar el estilo que se desea modificar.



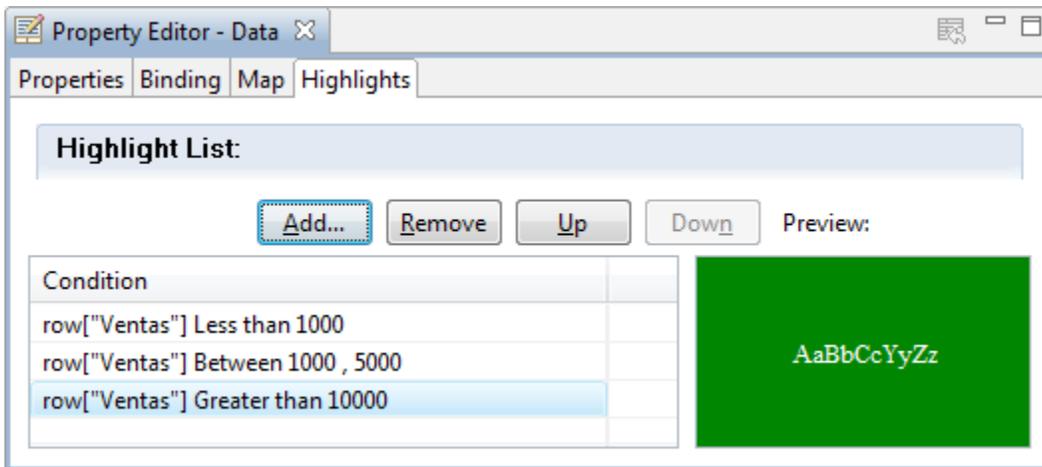
En particular la imagen muestra la selección del estilo **report** que permite configurar el formato de todo el reporte.

Formato Condicional

Permite resaltar elementos cuando ocurre alguna condición, la cual puede ser sobre el propio dato u otro.

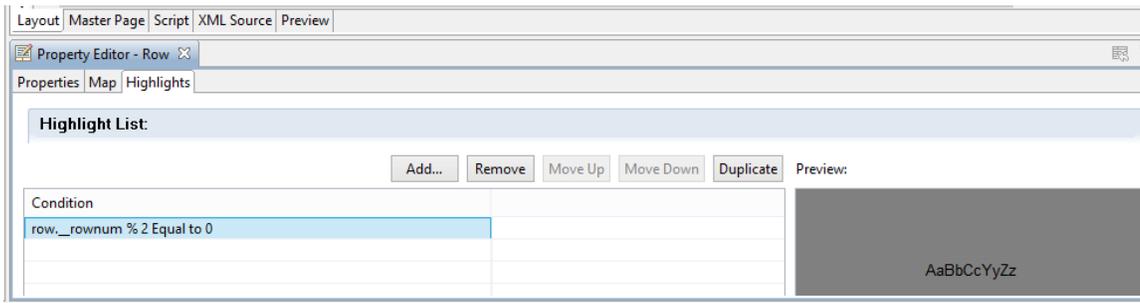
Por ejemplo:

- Resaltar ventas por debajo de un monto.
- Resaltar Vendedores con ventas por debajo del promedio.
- Resaltar valores negativos.
- Coloración de acuerdo a una meta (rojo, amarillo, verde).



Otra alternativa puede ser alternar el color de las filas, facilitando la lectura de la tabla con muchos datos.

Referencia [aqui](#)



The screenshot shows the BIRT Report Viewer window. The title bar says 'BIRT Report Viewer'. The report content is titled 'Reporte de Ventas 2008'. It displays a table with two columns: 'Vino' and 'Litros Vendidos'. The table contains the following data:

Vino	Litros Vendidos
Cabernet Sauvignon, Merlot	41850,0
Cabernet, Tannat, Merlot	41150,0
Chenin, Gewurzt	45250,0
Moscatoel, Merlot	29700,0
Pinot Blanc, Merlot	42500,0
Pinot Noir, Merlot, Malbec	40900,0
Sauvignon, Pinot Blanc	36000,0
Sauvignon, Trebbiano, Chardonnay	32650,0
Tannat, Cabernet, Merlot	36900,0
Tannat, Merlot	50400,0

Funciones de Birt

03 Report cuenta con funciones propias de Birt, que podremos utilizar al momento de definir una expresión.

Aquí se detallan alguna de las clases disponibles con las funciones que cada una provee:

- BirtComp
- BirtMath
- BirtStr

BirtComp

Clase BirtComp

La clase BirtComp proporciona funciones para comparar valores, por ejemplo, para probar si un valor es igual, mayor o menor que otro valor. Todas las funciones devuelven verdadero o falso.

BirtComp.anyOf

Esta función prueba si uno o más valores se encuentra en un campo.

Sintaxis

```
BirtComp.anyOf (fuente, objetivo1, ..., objetivoN)
```

Parámetros

fuente - El campo que contiene los valores a comparar.

objetivo - El valor o los valores a encontrar en la fuente.

Devoluciones

True si un valor objetivo coincide con un valor en la fuente; devuelve false en caso contrario.

Ejemplos

La prueba del siguiente ejemplo si Canadá, México o EE.UU. son los valores en el campo País. Si alguno de los países está en el campo, la función devuelve true.

```
BirtComp.anyOf (row ["País"], "Canadá", "México", "EE.UU.")
```

La prueba del siguiente ejemplo, si 15 de enero 2009 o 31 de enero 2009 son valores en el campo payDate:

```
BirtComp.anyOf (row ["payDate"], "15/1/2009", "31/01/2009")
```

La prueba del siguiente ejemplo si dos códigos de los productos específicos son los valores en el campo de código de producto. Si bien el valor aparece en el campo, la cadena de obsoletos se muestra, de lo contrario el valor ProductCode se muestra tal y como aparece en el campo.

```
if (BirtComp.anyOf (row ["ProductCode"], "S18_1749", "S18_2248")) {  
  DisplayString = "obsoletos"  
}  
else {  
  row ["ProductCode"]  
}
```

BirtComp.between

Esta función prueba si un valor está entre dos valores específicos.

Sintaxis

```
BirtComp.between (fuente, objetivo1, objetivo2)
```

Parámetros

fuente - El valor que se prueba.

objetivo1 - El primer valor en el rango de valores para comparar.

objetivo2 - El segundo valor en el rango de valores para comparar.

Devoluciones

True si el valor de fuente es entre el objetivo1 y los valores objetivo; devuelve false en caso contrario.

Ejemplos

Las siguientes expresiones prueban si los valores del campo están dentro de un rango específico de valores:

```
BirtComp.between (row ["SalesTotal"], 10000, 20000)
BirtComp.between (row ["CustomerName"], "A", "M")
BirtComp.between (row ["OrderDate"], "01/01/2009", "01/31/2009")
```

El siguiente ejemplo calcula las fechas de envío. Si un valor de OrderDate es en diciembre de 2008 (entre el 12/01/08 y 12/31/08), añadir 5 días para el valor OrderDate. Si un valor de OrderDate es en un mes que no sea diciembre, añadir 3 días para el valor OrderDate.

```
if (BirtComp.between (row ["OrderDate"], "01/12/2008", "31/12/2008")) {
  fechaEnvio = BirtDateTime.addDay (row ["OrderDate"], 5)
}
else {
  ShipDate = BirtDateTime.addDay (row ["OrderDate"], 3)
}
```

BirtComp.compareString

Esta función prueba si un valor de cadena coincide con otro valor de la cadena, teniendo en cuenta las condiciones especificadas.

Sintaxis

```
BirtComp.compareString (fuente1, fuente2, ignoreCase, recorte)
```

Parámetros

fuente1 - El primer valor de cadena a utilizar en la comparación.

fuente2 - El segundo valor de cadena a utilizar en la comparación.

ignoreCase - Especifique true para realizar una comparación entre mayúsculas y minúsculas. Especifique false para realizar una comparación entre mayúsculas y minúsculas.

recorte - Especifique true para eliminar los espacios en blanco iniciales o finales antes de comparar los dos valores. Los espacios en blanco en medio de una cadena no se quitan. Especifique false si desea que la comparación incluya espacios en blanco iniciales o finales.

Devoluciones

True si el valor coincide con el valor fuente1 fuente2; devuelve false en caso contrario.

Ejemplos

Las siguientes expresiones comparar cadenas con diferentes condiciones especificadas:

```
BirtComp.compareString ("Jackson", "Jackson", false,) // devuelve true
BirtComp.compareString ("Jackson", "Jackson", false, true) // devuelve false
BirtComp.compareString ("Jackson", "Jackson", false, true) // devuelve true
BirtComp.compareString ("Jackson", "Jackson", false, true) // devuelve false
BirtComp.compareString ("Jackson", "Jackson", true, true) // devuelve true
BirtComp.equalTo
```

BirtComp.equalTo

Esta función prueba si un valor es igual a otro valor.

Sintaxis

```
BirtComp.equalTo (fuente, destino)
```

Parámetros

fuente - El primer valor a utilizar en la comparación.

objetivo - El segundo valor a utilizar en la comparación.

Devoluciones

True si el valor de fuente es igual al valor objetivo; devuelve false en caso contrario.

Ejemplos

Las siguientes expresiones prueban si los valores del campo son iguales a los valores especificados:

```
BirtComp.equalTo (row ["SalesTotal"], 10000)
BirtComp.equalTo (row ["País"], "Francia")
BirtComp.equalTo (row ["OrderDate"], "15/02/2009")
```

BirtComp.greaterOrEqual

Esta función prueba si un valor es mayor o igual a otro valor.

Sintaxis

```
BirtComp.greaterOrEqual (fuente, destino)
```

Parámetros

fuente - El primer valor a utilizar en la comparación.

objetivo - El segundo valor a utilizar en la comparación.

Devoluciones

True si el valor de fuente es mayor o igual al valor objetivo; devuelve false en caso contrario.

Ejemplos

Las siguientes expresiones prueban si los valores del campo son mayores o iguales a los valores especificados:

```
BirtComp.greaterOrEqual (row ["SalesTotal"], 10000)
BirtComp.greaterOrEqual (row ["ProductCode"], "S50_4000")
BirtComp.greaterOrEqual (row ["OrderDate"], "15/02/2009")
```

BirtComp.greaterThan

Esta función prueba si un valor es mayor que otro valor.

Sintaxis

```
BirtComp.greaterThan (fuente, destino)
```

Parámetros

fuelle - El primer valor a utilizar en la comparación.

objetivo - El segundo valor a utilizar en la comparación.

Devoluciones

True si el valor de fuente es mayor que el valor objetivo; devuelve false en caso contrario.

Ejemplos

Las siguientes expresiones prueban si los valores del campo son mayores que los valores especificados:

```
BirtComp.greaterThan (row ["SalesTotal"], 10000)
BirtComp.greaterThan (row ["CustomerName"], "M")
BirtComp.greaterThan (row ["OrderDate"], "15/02/2009")
```

BirtComp.lessOrEqual

Esta función prueba si un valor es menor o igual a otro valor.

Sintaxis

```
BirtComp.lessOrEqual (fuente, destino)
```

Parámetros

fuelle - El primer valor a utilizar en la comparación.

objetivo - El segundo valor a utilizar en la comparación.

Devoluciones

True si el valor de fuente es menor o igual al valor objetivo; devuelve false en caso contrario.

Ejemplos

Las siguientes expresiones prueban si los valores del campo son menores o iguales a los valores especificados:

```
BirtComp.lessOrEqual (row ["SalesTotal"], 10000)
BirtComp.lessOrEqual (row ["ProductCode"], "S18_4000")
BirtComp.lessOrEqual (row ["OrderDate"], "15/02/2009")
```

BirtComp.lessThan

Esta función prueba si un valor es menor que otro valor.

Sintaxis

```
BirtComp.lessThan (fuente, destino)
```

Parámetros

fuelle - El primer valor a utilizar en la comparación.

objetivo - El segundo valor a utilizar en la comparación.

Devoluciones

True si el valor de fuente es menor que el valor objetivo; devuelve false en caso contrario.

Ejemplos

Las siguientes expresiones prueban si los valores del campo son inferiores a los valores especificados:

```
BirtComp.lessThan (row ["SalesTotal"], 10000)
BirtComp.lessThan (row ["CustomerName"], "M")
BirtComp.lessThan (row ["OrderDate"], "15/02/2009")
```

BirtComp.like

Esta función prueba si un valor de cadena coincide con un patrón. El patrón debe utilizar la sintaxis SQL de patrones.

Sintaxis

```
BirtComp.like (fuente, destino)
```

Parámetros

fuente - El valor de cadena de evaluar.

objetivo - El modelo de cadena para que coincida. Debe incluir el patrón entre comillas dobles ("). Utilice la caja correcta al escribir los caracteres que desea hacer coincidir. Usted puede utilizar el siguiente carácter especial en un patrón:

- Un carácter de porcentaje (%) coincide con cero o más caracteres. Por ejemplo, %ace% coincide con cualquier cadena que contenga la subcadena ace, tales como Facebook y MySpace. No coincide con Ace Corporation porque esta cadena contiene la letra A en mayúscula, y no la a minúscula.
- Un guión bajo (_) que coincide exactamente con uno de los caracteres.

Para hacer coincidir un porcentaje literal (%) o carácter guión bajo (_), se precedera a los caracteres con dos caracteres de barras invertidas (\).

Por ejemplo, para que coincida con S_10, utilice el siguiente patrón:

```
S\_10
```

Para el 50%, utilice el siguiente patrón:

```
50\%
```

Devoluciones

True si el valor de fuente coincide con el valor objetivo; devuelve false en caso contrario.

Ejemplos

El ejemplo siguiente devuelve verdadero para los valores en el campo de código de producto que empiezan con S18:

```
BirtComp.like (row ["ProductCode"], "S18%")
```

En el siguiente ejemplo real de los valores productName que contienen la subcadena Ford precedida por un solo carácter devuelve:

```
BirtComp.like (row ["ProductName"], "_Ford%")
```

BirtComp.match

Esta función prueba si un valor de cadena coincide con un patrón. El patrón debe utilizar la sintaxis de expresiones regulares de JavaScript.

Sintaxis

```
BirtComp.match (fuente, destino)
```

Parámetros

fuente - El valor de cadena de evaluar.

objetivo - El modelo de cadena para que coincida. En la sintaxis de las expresiones regulares de JavaScript, se especifica un patrón dentro de un par de caracteres barra diagonal (/).

Usted puede utilizar cualquier carácter especial con el apoyo de expresiones JavaScript regulares, como las siguientes:

- Un signo de interrogación (?) Coincide con cero o una ocurrencia del carácter anterior a la misma.
- Un asterisco (*) que coincide con cero o cualquier número de apariciones del carácter anterior a la misma.
- Un carácter de punto (.) Coincide con cualquier carácter.
- Un carácter de intercalación (^) especifica que la subcadena a buscar es en el principio de una cadena.
- Un carácter que especifica una búsqueda entre mayúsculas y minúsculas.

Para hacer coincidir un carácter especial literalmente, se precedera al carácter especial con un carácter barra invertida (\).

Por ejemplo, para que coincida con S * 10, utilice el siguiente patrón:

```
/S\ * 10/
```

Para el 50%, utilice el siguiente patrón:

```
/50 %/
```

Devoluciones

True si el valor de fuente coincide con el valor objetivo; devuelve false en caso contrario.

Ejemplos

El ejemplo siguiente devuelve verdadero para los valores en el campo de código de producto que empiezan con S18:

```
BirtComp.match (row ["ProductCode"], / S18 ^ /)
```

En el siguiente ejemplo real de los valores productName que contienen la subcadena Ford regresa:

```
BirtComp.match (row ["ProductName"], / Ford /)
```

BirtComp.notBetween

Esta función prueba si un valor no se encuentra entre dos valores especificados.

Sintaxis

```
BirtComp.notBetween (fuente, objetivo1, objetivo2)
```

Parámetros

fuente - El valor que se prueba.

objetivo1 - El primer valor en el rango de valores para comparar.

objetivo2 - El segundo valor en el rango de valores para comparar.

Devoluciones

True si el valor de fuente no es entre el objetivo1 y los valores objetivo; devuelve false en caso contrario.

Ejemplos

Las siguientes expresiones prueban si los valores del campo para ver si están fuera de un determinado rango de valores:

```
BirtComp.notBetween (row ["SalesTotal"], 10000, 20000)
BirtComp.notBetween (row ["CustomerName"], "A", "M")
BirtComp.notBetween (row ["OrderDate"], "01/01/2009", "01/31/2009")
```

El siguiente ejemplo calcula las fechas de envío. Si un valor OrderDate no es en diciembre de 2008 (no entre 12/01/08 y 12/31/08), añadir 3 días para el valor OrderDate. Si un valor de OrderDate es en diciembre, cinco días para añadir el valor OrderDate.

```
if (BirtComp.notBetween (row ["OrderDate"], "01/12/2008", "31/12/2008")) {
    fechaEnvio = BirtDateTime.addDay (row ["OrderDate"], 3)
}
else {
    ShipDate = BirtDateTime.addDay (row ["OrderDate"], 5)
}
```

BirtComp.notEqual

Esta función prueba si un valor no es igual a otro valor.

Sintaxis

```
BirtComp.notEqual (fuente, destino)
```

Parámetros

fuente - El primer valor a utilizar en la comparación.

objetivo - El segundo valor a utilizar en la comparación.

Devoluciones

True si el valor de fuente no es igual al valor objetivo; devuelve false en caso contrario.

Ejemplos

Las siguientes expresiones prueban si los valores del campo no son iguales a los valores especificados:

```
BirtComp.notEqual (row ["SalesTotal"], 10000)
BirtComp.notEqual (row ["País"], "Francia")
BirtComp.notEqual (row ["OrderDate"], "15/02/2009")
```

BirtComp.notLike

Esta función prueba si un valor de cadena coincide con un patrón. El patrón debe utilizar la sintaxis SQL de patrones.

Sintaxis

```
BirtComp.notLike (fuente, destino)
```

Parámetros

fuente - El valor de cadena de evaluar.

objetivo - El modelo de cadena para que coincida. Debe incluir el patrón entre comillas dobles (""). Utilice la caja correcta al escribir los caracteres que desea hacer coincidir. Usted puede utilizar el siguiente carácter especial en un patrón:

- Un carácter de porcentaje (%) coincide con cero o más caracteres. Por ejemplo, %ace% coincide con cualquier cadena que contenga la subcadena ace, tales como Facebook y MySpace. No coincide con Ace Corporation porque esta cadena contiene la letra A en mayúscula, y no la a minúscula.
- Un guión bajo (_) que coincide exactamente con uno de los caracteres.

Para hacer coincidir un porcentaje literal (%) o carácter guión bajo (_), se precedera a los caracteres con dos caracteres de barras invertidas (\).

Por ejemplo, para que coincida con S_10, utilice el siguiente patrón:

```
S\_10
```

Para el 50%, utilice el siguiente patrón:

```
50\%
```

Devoluciones

True si el valor de fuente no coincide con el valor objetivo; devuelve false en caso contrario.

Ejemplos

El ejemplo siguiente devuelve false para los valores en el campo de código de producto que empiezan con S18:

```
BirtComp.notLike (row ["ProductCode"], "S18%")
```

El ejemplo siguiente devuelve false para los valores de productName que contienen la subcadena Ford precedidos por un solo carácter:

```
BirtComp.notLike (row ["ProductName"], "_Ford%")
```

BirtMath

Clase BirtMath

La clase BirtMath proporciona funciones para manipular datos numéricos, por ejemplo, para sumar, restar, dividir, multiplicar, y redondear números.

BirtMath.add

Esta función devuelve la suma de dos números.

Sintaxis

```
BirtMath.add (n1, n2)
```

Parámetros

n1 - El primer número a utilizar en el cálculo.

n2 - El segundo número a utilizar en el cálculo.

Devoluciones

Un número que resulte de sumar n1 y n2.

Ejemplo

El ejemplo siguiente devuelve la suma de los valores en los campos SalesTotal y Shipping:

```
BirtMath.add( row["SalesTotal"], row["Shipping"] )
```

BirtMath.ceiling

Esta función redondea un número hacia arriba, alejado de 0, al múltiplo más cercano especificado.

Sintaxis

```
BirtMath.ceiling (n, importancia)
```

Parámetros

n - El número a redondear al alza.

importancia - Los múltiplos para redondear n.

Devoluciones

Un número que resulta del redondeo. Si n es un múltiplo exacto de importancia, no se redondea.

Ejemplos

```
BirtMath.ceiling ()
```

Se utiliza comúnmente para redondear al alza los precios.

```
BirtMath.ceiling (row ["precio"], 0,05)
```

Si el valor del precio es 20.52, la expresión devuelve 20.55.

```
BirtMath.ceiling (row ["precio"], 0,1)
```

Si el valor del precio es 20.52, la expresión devuelve 20.60. Si el valor del precio es 20.40, la expresión devuelve 20.40. No hay redondeo se debe a que 20,40 es un múltiplo de 0,1.

```
BirtMath.ceiling (row ["precio"], 1)
```

Si el valor del precio es 20.30, la expresión devuelve 21.

BirtMath.divide

Esta función devuelve el resultado de dividir un número por otro.

Sintaxis

```
BirtMath.divide (dividendo, divisor)
```

Parámetros

dividendos - El número que se divide.

divisor - El número por el cual los dividendos se dividen.

Devoluciones

Un número que resulte de dividir dividendo por el divisor.

Ejemplos

El siguiente ejemplo muestra el resultado de dividir un número específico:

```
BirtMath.divide (10, 2) // devuelve 5  
BirtMath.divide (2, 10) // devuelve 0.2
```

El ejemplo siguiente devuelve el resultado de dividir los valores del campo total de valores en el campo Cantidad:

```
BirtMath.divide (row ["total"], row ["Cantidad"])
```

BirtMath.mod

Esta función devuelve el resto después de dividir un número por un divisor.

Sintaxis

```
BirtMath.mod (número, divisor)
```

Parámetros

número - El número que se divide.

divisor - El número por el cual el primer número está dividido. Debe especificar un número distinto de cero.

Devoluciones

El resto después de dividir un número por el divisor.

Ejemplos

Los siguientes ejemplos muestran los resultados que devuelve la función de números específicos:

```
BirtMath.mod (10, 3) // devuelve 1  
BirtMath.mod (10, 5) // devuelve 0  
BirtMath.mod (10, 6) // devuelve 4
```

El siguiente ejemplo comprueba si los números en el campo de grado son pares o impares. Cuando el divisor es 2, la función devuelve 0 para los

números pares, y 1 para números impares.

```
BirtMath.mod (row ["Grado"], 2)
```

BirtMath.multiply

Esta función devuelve la multiplicación de dos números.

Sintaxis

```
BirtMath.multiply (n1, n2)
```

Parámetros

n1 - El primer número a utilizar en el cálculo.

n2 - El segundo número a utilizar en el cálculo.

Devoluciones

Un número que resulte de multiplicar n1 y n2.

Ejemplo

En el ejemplo siguiente se devuelve el resultado de multiplicar los valores en los campos QuantityOrdered y PriceEach:

```
BirtMath.multiply (row ["QuantityOrdered"], row ["PriceEach"])
```

BirtMath.round

Esta función redondea un número a un número determinado de dígitos.

Sintaxis

```
BirtMath.round (número, dec)
```

Parámetros

número - El número para el redondeo.

dec - El número de dígitos de redondeo para todo el número. Si se omite este argumento, la función supone 0.

Devoluciones

Un número redondeado a un número determinado de dígitos.

Ejemplos

Los siguientes ejemplos muestran los resultados que devuelve la función de números específicos:

```
BirtMath.round (1.545,50) // devuelve 1546  
BirtMath.round (1545,56, 1) // devuelve 1545.6  
BirtMath.round (1545,23, 1) // devuelve 1545.2  
BirtMath.round (1.545,50, -1) // devuelve 1550
```

Las rondas siguiente ejemplo los valores en el campo PriceEstimate para devolver un número entero:

```
BirtMath.round (row ["PriceEstimate"], 0)
```

BirtMath.roundDown

Esta función redondea un número a un número determinado de dígitos.

Sintaxis

```
BirtMath.roundDown (número, dec)
```

Parámetros

número - El número para el redondeo.

dec - El número de dígitos de redondeo para todo el número. Si se omite este argumento, la función supone 0.

Devoluciones

Un número redondeado a un número determinado de dígitos.

Ejemplos

Los siguientes ejemplos muestran los resultados que devuelve la función de números específicos:

```
BirtMath.roundDown (1.545,50) // devuelve 1545  
BirtMath.roundDown (1545,56, 1) // devuelve 1545.5  
BirtMath.roundDown (1545,23, 1) // devuelve 1545.2  
BirtMath.roundDown (1.545,50, -1) // devuelve 1540
```

Las rondas de ejemplo siguiente se establecen los valores en el campo PriceEstimate para devolver un número entero:

```
BirtMath.roundDown (row ["PriceEstimate"], 0)
```

BirtMath.roundUp

Esta función redondea un número hasta un número determinado de dígitos.

Sintaxis

```
BirtMath.roundUp (número, dec)
```

Parámetros

número - El número para el redondeo.

dec - El número de dígitos de redondeo para todo el número. Si se omite este argumento, la función supone 0.

Devoluciones

Un número redondeado a un número determinado de dígitos.

Ejemplos

Los siguientes ejemplos muestran los resultados que devuelve la función de números específicos:

```
BirtMath.roundUp (1.545,50) // devuelve 1546
BirtMath.roundUp (1545,56, 1) // devuelve 1545.6
BirtMath.roundUp (1545,23, 1) // devuelve 1545.3
BirtMath.roundUp (1.545,50, -1) // devuelve 1550
```

Las rondas de ejemplo el seguimiento de los valores en el campo PriceEstimate para devolver un número entero:

```
BirtMath.roundUp (row ["PriceEstimate"], 0)
```

BirtMath.safeDivide

Esta función devuelve el resultado de dividir un número por otro, evitar una división por cero condición.

Sintaxis

```
BirtMath.safeDivide (dividendo, divisor, ifZero)
```

Parámetros

dividendos - El número que se divide.

divisor - El número por el cual los dividendos se dividen.

ifZero - El valor que se devolverá cuando el divisor es cero.

Devoluciones

Un número que resulte de dividir dividendo por el divisor.

Ejemplos

El siguiente ejemplo muestra los resultados que devuelve la función de números específicos:

```
BirtMath.safeDivide (10, 2, 0) // devuelve 5
BirtMath.safeDivide (10, 0, 0) // devuelve 0
```

En el ejemplo siguiente se devuelve el resultado de dividir los valores en el ámbito de los ingresos por valores en el campo de volumen. Si el valor de volumen es 0, la función devuelve 0.

```
BirtMath.safeDivide (row ["Ingresos"], row ["Volumen"], 0)
```

BirtMath.subtract

Esta función devuelve el resultado de restar un número de otro.

Sintaxis

```
BirtMath.subtract (n1, n2)
```

Parámetros

n1 - El número de los que a restar.

n2 - El número a restar de n1.

Devoluciones

Un número que resulta de restar n2 de n1.

Ejemplo

El siguiente ejemplo muestra los resultados de restar números específicos:

```
BirtMath.subtract (12, 4) // devuelve 8  
BirtMath.subtract (4, 12) // devuelve -8
```

En el ejemplo siguiente se devuelve el resultado de restar descuento del total:

```
BirtMath.subtract (row ["total"], row ["Descuento"])
```

BirtStr

Clase BirtStr

La clase BirtStr proporciona funciones para manipular cadenas, por ejemplo, para concatenar cadenas, recortar espacios adicionales, ver las partes de una cadena, o una cadena en mayúsculas o minúsculas.

BirtStr.charLength

Esta función devuelve la longitud de una cadena dada.

Sintaxis

```
BirtStr.charLength (fuente)
```

Parámetro

fuente - La cadena de evaluar.

Devoluciones

El número de caracteres en la cadena especificada.

Ejemplos

En el ejemplo siguiente se devuelve la longitud de una cadena específica:

```
BirtStr.charLength ("Julie Murphy") // devuelve 12
```

El ejemplo siguiente devuelve la longitud de cada valor en el campo CustomerName:

```
BirtStr.charLength (row ["CustomerName"])
```

BirtStr.concat

Esta función devuelve la cadena que resulta de la concatenación de cadenas especificadas.

Sintaxis

```
BirtStr.concat (Source1, ..., sourceN)
```

Parámetro

Source1, ..., sourceN - Las cadenas para concatenar.

Devoluciones

La cadena que resulta de la concatenación de una serie de cadenas.

Ejemplo

En el ejemplo siguiente se devuelve una dirección completa mediante la concatenación de los valores de varios campos:

```
BirtStr.concat (row ["AddressLine1"], "", row ["AddressLine2"], "", row ["ciudad"],  
"", row ["Código postal"], "row [" Estado " ] , "", row ["País"])
```

BirtStr.indexOf

Esta función devuelve la posición de una subcadena especificada en una cadena determinada.

Sintaxis

```
BirtStr.indexOf (destino, origen, principio)
```

Parámetros

objetivo - La subcadena a buscar. La búsqueda distingue entre mayúsculas y minúsculas.

fuentes - La cadena en la cual se va a buscar una subcadena especificada.

inicio - (Opcional). La posición en la cadena de origen, donde comienza la búsqueda. Si se omite este argumento, la función comienza la búsqueda desde el primer carácter de la cadena.

Devoluciones

La posición numérica de la subcadena en la cadena. El primer carácter de una cadena que comienza en 0. Si la subcadena no se encuentra, la función devuelve -1.

Ejemplos

En el ejemplo siguiente se devuelve la posición numérica de caracteres especificado en cadenas específicas:

```
BirtStr.indexOf (" ", "Julie Murphy") // devuelve 5
BirtStr.indexOf ("-", "Modela-1234-567") // devuelve 6
BirtStr.indexOf ("-", "Modela-1234-567", 7) // devuelve 11
```

En el ejemplo siguiente se utiliza `BirtStr.indexOf ()`, en relación con `BirtStr.left ()` para mostrar los caracteres que preceden al carácter de espacio en un nombre de cliente. El `BirtStr.left ()` función extrae una subcadena de una longitud especificada, a partir del primer carácter. En este ejemplo, la longitud de la subcadena de pantalla es igual a la posición numérica del carácter de espacio.

```
spaceCharPosition = BirtStr.indexOf (" ", row ["customerName"]);
displayFirstName = BirtStr.left (row ["customerName"], spaceCharPosition);
```

Si el nombre del cliente es Julie Murphy, la expresión devuelve Julie.

BirtStr.left

Esta función extrae una subcadena de una longitud especificada de una cadena, a partir de la más a la izquierda, o en primer lugar,.

Sintaxis

```
BirtStr.left (fuente, n)
```

Parámetros

fuente - La cadena de la que extraer una subcadena.

n - El número de caracteres para extraer, a partir del primer carácter.

Devoluciones

Una subcadena de una longitud específica.

- Si se omite n, el número de caracteres a extraer, la función devuelve el primer carácter único.
- Si n es cero, la función devuelve una cadena vacía.
- Si n es mayor que la longitud de la cadena, la función devuelve la cadena completa.

Ejemplos

El ejemplo siguiente devuelve las subcadenas de diferentes longitudes de cadenas específicas:

```
BirtStr.left ("Julie Murphy", 5) // devuelve Julie
BirtStr.left ("Julie Murphy", 12) // devuelve Julie Murphy
```

En el ejemplo siguiente se utiliza `BirtStr.indexOf ()`, en relación con `BirtStr.left ()` para mostrar los caracteres que preceden al carácter de espacio en un nombre de cliente. El `BirtStr.left ()` función extrae una subcadena de una longitud especificada, a partir del primer carácter. En este ejemplo, la longitud de la subcadena de pantalla es igual a la posición numérica del carácter de espacio.

```
spaceCharPosition = BirtStr.indexOf (" ", row ["customerName"]);
displayFirstName = BirtStr.left (row ["customerName"], spaceCharPosition);
```

Si el nombre del cliente es Julie Murphy, la expresión devuelve Julie.

BirtStr.right

Esta función extrae una subcadena de una longitud especificada de una cadena, a partir de la más a la derecha, o el último carácter.

Sintaxis

```
BirtStr.right (fuente, n)
```

Parámetros

fuente - La cadena de la que extraer una subcadena.

n - El número de caracteres para extraer, a partir del último carácter.

Devoluciones

Una subcadena de una longitud específica.

- Si se omite n, en el número de caracteres a extraer, la función devuelve el último carácter único.
- Si n es cero, la función devuelve una cadena vacía.
- Si n es mayor que la longitud de la cadena, la función devuelve la cadena completa.

Ejemplos

El ejemplo siguiente devuelve las subcadenas de diferentes longitudes de cadenas específicas:

```
BirtStr.right ("Julie Murphy", 6) // devuelve Murphy  
BirtStr.right ("Julie Murphy", 12) // devuelve Julie Murphy
```

En el ejemplo siguiente se utiliza `BirtStr.right ()`, en relación con la `BirtStr.indexOf ()` y `BirtStr.charLength ()` para mostrar los caracteres que aparecen después de que el carácter de espacio en un nombre de cliente. En este ejemplo se supone que el número de caracteres después del guión varía. Por lo tanto, la longitud de la cadena completa (devuelto por `BirtStr.charLength ()`) menos la longitud hasta el guión (que devuelve `BirtStr.indexOf ()`) es el número de caracteres para mostrar.

```
spaceCharPosition = BirtStr.indexOf (" ", row ["customerName"]);  
displayLastName = BirtStr.right (row ["customerName"], (BirtStr.charLength (row  
["customerName"] - (spaceCharPosition + 1)))
```

Si el nombre del cliente es Julie Murphy, la expresión devuelve Murphy. Si el nombre del cliente es Kwai Li, la expresión devuelve Li.

BirtStr.search

Esta función devuelve la posición de una subcadena especificada en una cadena determinada. La subcadena puede contener caracteres comodín.

Sintaxis

```
BirtStr.search (patrón, la fuente, el índice)
```

Parámetros

patrón - El patrón de la cadena a buscar. La búsqueda distingue entre mayúsculas y minúsculas. Puede usar los siguientes caracteres comodín en un patrón:

- Un asterisco, coincide con cero o más caracteres, espacios incluidos. Por ejemplo, `n * t` partidos `tn`, el `estaño` y el `adolescente`.
- Un signo de interrogación, coincide exactamente con uno de los personajes. Por ejemplo, `no? N` coincide con `canela`, `diez`, `estaño`, y una `tonelada`. No coincide con la `adolescencia` o `tn`.

Para coincidir con un asterisco literal o signo de interrogación en una cadena, preceden a los personajes con dos caracteres de barra invertida (`\`). Por ejemplo, para encontrar la subcadena `R * 10`, utilice el modelo de cadena siguientes:

"R\ * 10"

fuente - La cadena en la que buscar una subcadena especificada.

índice - (Opcional). La posición en la cadena de origen, donde comienza la búsqueda. Si se omite este argumento, la función comienza la búsqueda desde el primer carácter de la cadena.

Devoluciones

La posición numérica de la subcadena en la cadena. El primer carácter de una cadena que comienza en 0. Si la subcadena no se encuentra, la función devuelve -1.

Ejemplos

En el ejemplo siguiente se devuelve la posición numérica de los patrones de cadena especificada en cadenas específicas:

```
BirtStr.search ("XM?", "XMS-ModelA-1234-567") // devuelve 0
BirtStr.search ("Modela \*", "XMS-ModelA-1234-567") // devuelve 4
BirtStr.search ("-", "XMS-ModelA-1234-567", 4) // devuelve 10
```

En el siguiente ejemplo para el patrón de texto, S*A, en cada valor en el campo ProductCode. Si el código de producto es KBS5412A, la expresión devuelve 2.

```
BirtStr.search ("S*A", row ["ProductCode"])
```

En el ejemplo siguiente BirtStr.search usos (), en relación con BirtStr.left () para mostrar los caracteres que preceden al patrón de texto, * Modelo, en nombre de un producto. El BirtStr.left () función extrae una subcadena de una longitud especificada, a partir del primer carácter. En este ejemplo, la longitud de la subcadena de pantalla es igual a la posición numérica del patrón de texto.

```
stringPatternPosition = BirtStr.search ("-Model*", row ["ProductName"]);
DisplayString = BirtStr.left (row ["ProductName"], stringPatternPosition);
```

Si el nombre del producto es XMS-Modela-1234, la expresión devuelve XMS.

BirtStr.toLower

Esta función convierte todas las letras de una cadena en minúsculas.

Sintaxis

```
BirtStr.toLower (fuente)
```

Parámetro

fuente - La cadena para convertir a minúsculas.

Devoluciones

La cadena especificada en letras minúsculas

Ejemplo

El ejemplo siguiente muestra todos los valores en el campo ProductLine en minúsculas:

```
BirtStr.toLowerCase (row ["ProductLine"])  
BirtStr.toUpperCase
```

BirtStr.toUpperCase

Esta función convierte todas las letras de una cadena en mayúsculas.

Sintaxis

```
BirtStr.toUpperCase (fuente)
```

Parámetro

fuente - La cadena para convertir a mayúsculas.

Devoluciones

La cadena especificada en mayúsculas

Ejemplo

El ejemplo siguiente muestra todos los valores en el campo customerName en mayúsculas:

```
BirtStr.toUpperCase (row ["customerName"])
```

BirtStr.trim

Esta función devuelve una cadena con todos los espacios iniciales y finales en blanco eliminados. No quita los caracteres en blanco entre las palabras.

Sintaxis

```
BirtStr.trim (fuente)
```

Parámetro

fuente - La cadena de la que para eliminar los espacios iniciales y finales en blanco.

Devoluciones

Una cadena con todos los espacios iniciales y finales en blanco eliminados.

Ejemplo

En el ejemplo siguiente se utiliza BirtStr.trim () para eliminar todos los espacios iniciales y finales en blanco de los valores en los campos de datos Nombre y Apellidos.

```
BirtStr.trim (row ["Nombre"]) + " " + BirtStr.trim (row ["Apellido"])
```

BirtStr.trimLeft

Esta función devuelve una cadena con todos los espacios iniciales y finales en blanco eliminados. No quita los caracteres en blanco entre las palabras.

Sintaxis

```
BirtStr.trimLeft (fuente)
```

Parámetro

fuente - La cadena de la que para eliminar principales caracteres en blanco.

Devoluciones

Una cadena con todos los espacios principales en blanco eliminados.

Ejemplo

En el ejemplo siguiente concatena un literal de cadena con cada valor en el campo `customerName`. `BirtStr.trimLeft ()` elimina todos los caracteres en blanco que precede el valor `customerName` modo que no hay caracteres en blanco entre la cadena literal y el valor `customerName`.

```
"Nombre del cliente:" + BirtStr.trimLeft (row ["customerName"])
```

BirtStr.trimRight

Esta función devuelve una cadena con todos los caracteres en blanco de cola eliminado. No quita los caracteres en blanco entre las palabras.

Sintaxis

```
BirtStr.trimRight (fuente)
```

Parámetro

fuente - La cadena de la que se arrastra para eliminar caracteres en blanco.

Devoluciones

Una cadena con todos los caracteres en blanco de cola eliminado.

Ejemplo

En el ejemplo siguiente concatena cada valor en el campo de comentario con un punto y coma, a continuación, con un valor en el campo de acción. `BirtStr.trimRight ()` elimina todos los caracteres en blanco después del valor de comentario de modo que no hay caracteres en blanco entre la cadena de comentario y el punto y coma.

```
BirtStr.trimRight (row ["comentario"]) + " " + row ["Acción"])
```

Reporte con Número Variable de Columnas

Ante el hecho de tener que confeccionar un Reporte con las ventas que se van realizando durante el año, y necesitamos ver esta información mes a mes.

Como ya hemos visto, siempre estamos trabajando con un *Data Sets* fijo, con cantidad de columnas ya conocidas, pero este caso es distinto, ya que cuando pase el tiempo (meses), se irán agregando columnas.

O3Report nos posibilita esto a través de la funcionalidad de utilizar *CrossTab*

Caso de Ejemplo

Reporte que nos informe los litros vendidos de los distintos vinos por año.

Utilizaremos el Datamart de Vinos que se brinda con la instalación.

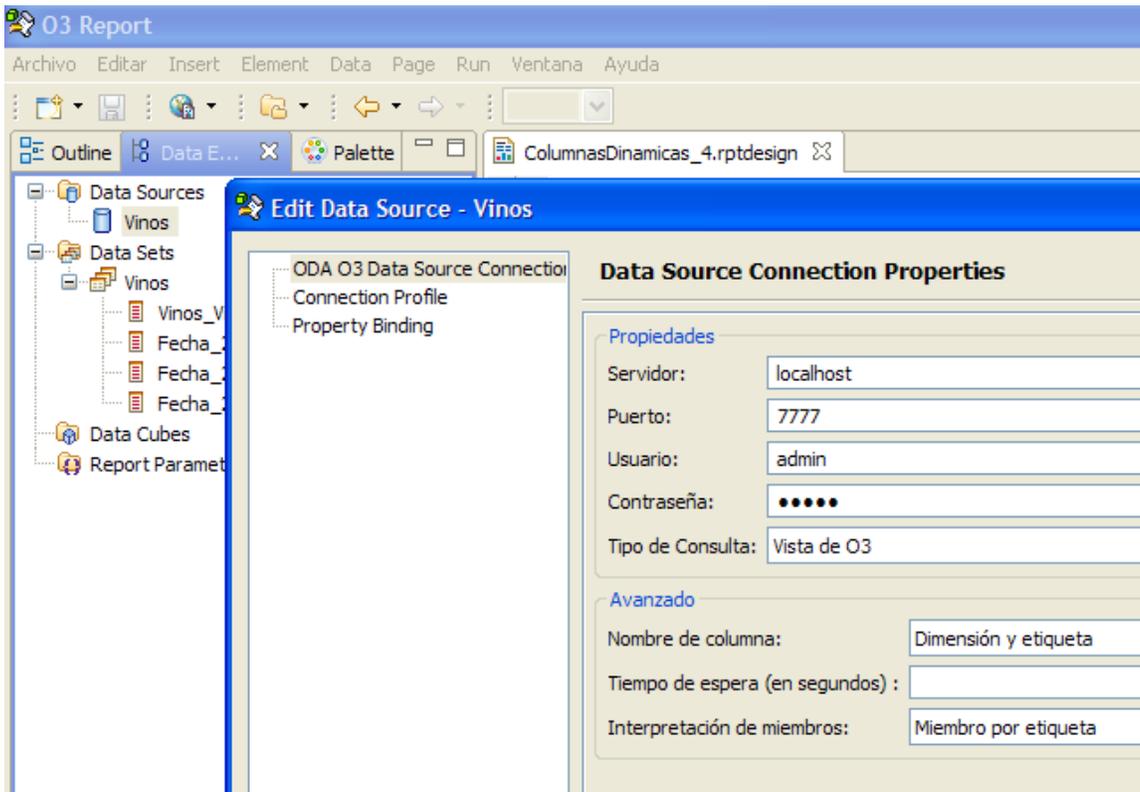
Partiendo que vamos a utilizar como *Data Set* una vista de O3, debemos considerar de configurar la misma anidando las dimensiones correspondientes a la información que queremos utilizar en el reporte y en las columnas la o las medidas.

Creando el Reporte



Fecha	Vinos	Medidas
Año	Vino	Litros Vendidos
2007	Cabernet Sauvignon, Merlot	4.050
	Cabernet, Tannat, Merlot	9.000
	Chenin, Gewurzt	7.200
	Moscatel, Merlot	6.750
	Pinot Blanc, Merlot	4.500
	Pinot Noir, Merlot, Malbec	16.200
	Sauvignon, Pinot Blanc	15.300
	Sauvignon, Trebbiano, Chardonnay	9.450
	Tannat, Cabernet, Merlot	9.450
	Tannat, Merlot	22.050
	Tannat, Syrah, Merlot	6.300
	Ugni Blanc, Semillon	11.250
	Brut Blanc de Noirs	2.700
	Cabernet Franc Rosado	8.550
	Cabernet Franc Tinto	7.200
Cabernet Sauvignon Roble	13.500	

En un nuevo **O3Report**, creamos un nuevo *Data Source*



O3 Report

Archivo Editar Insert Element Data Page Run Ventana Ayuda

Outline Data E... Palette ColumnasDinamicas_4.rptdesign

Edit Data Source - Vinos

ODA O3 Data Source Connection
Connection Profile
Property Binding

Data Source Connection Properties

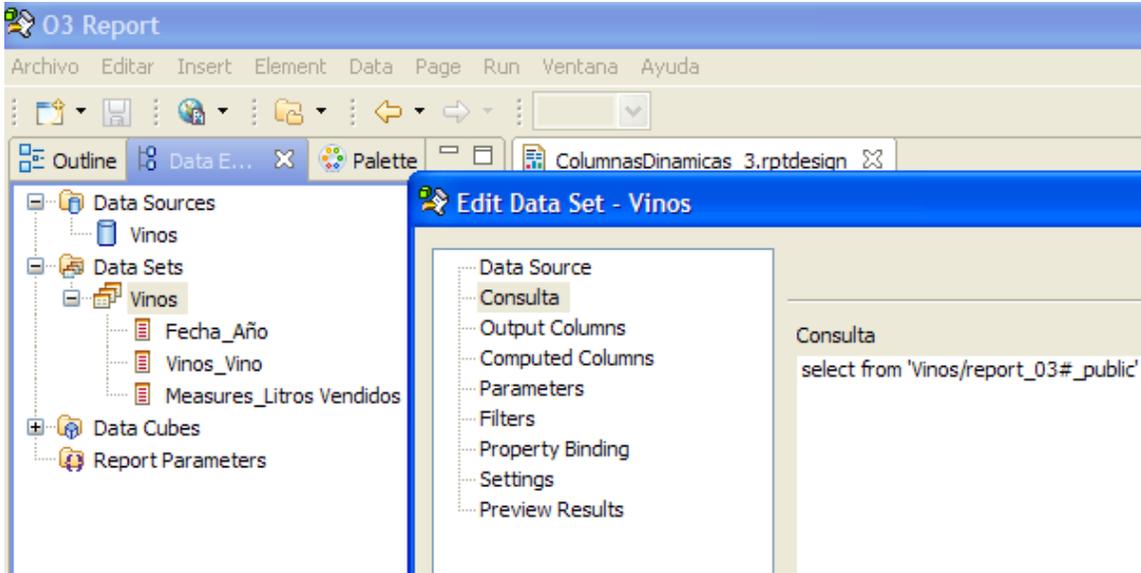
Propiedades

Servidor: localhost
Puerto: 7777
Usuario: admin
Contraseña: ●●●●●
Tipo de Consulta: Vista de O3

Avanzado

Nombre de columna: Dimensión y etiqueta
Tiempo de espera (en segundos):
Interpretación de miembros: Miembro por etiqueta

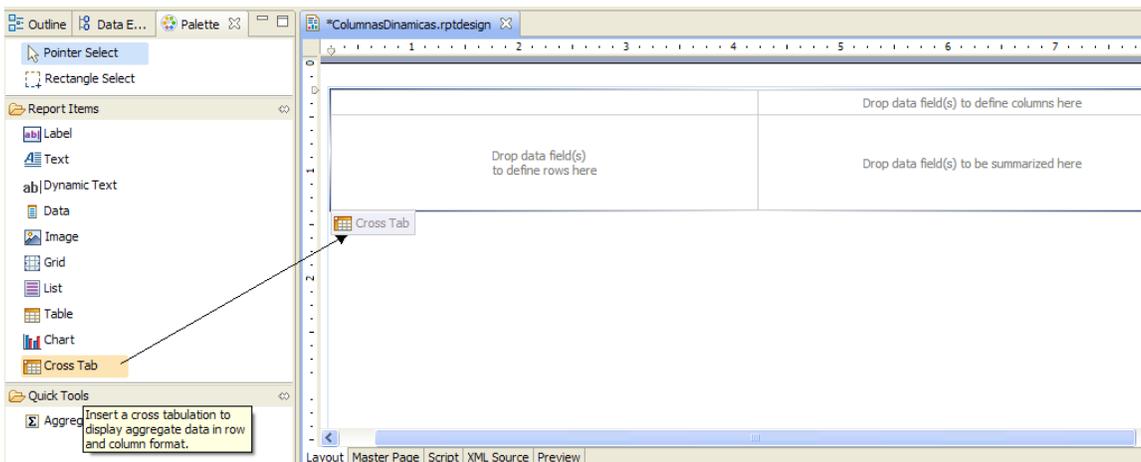
y a continuación un *Data Sets*, definida por la vista que guardamos anteriormente.



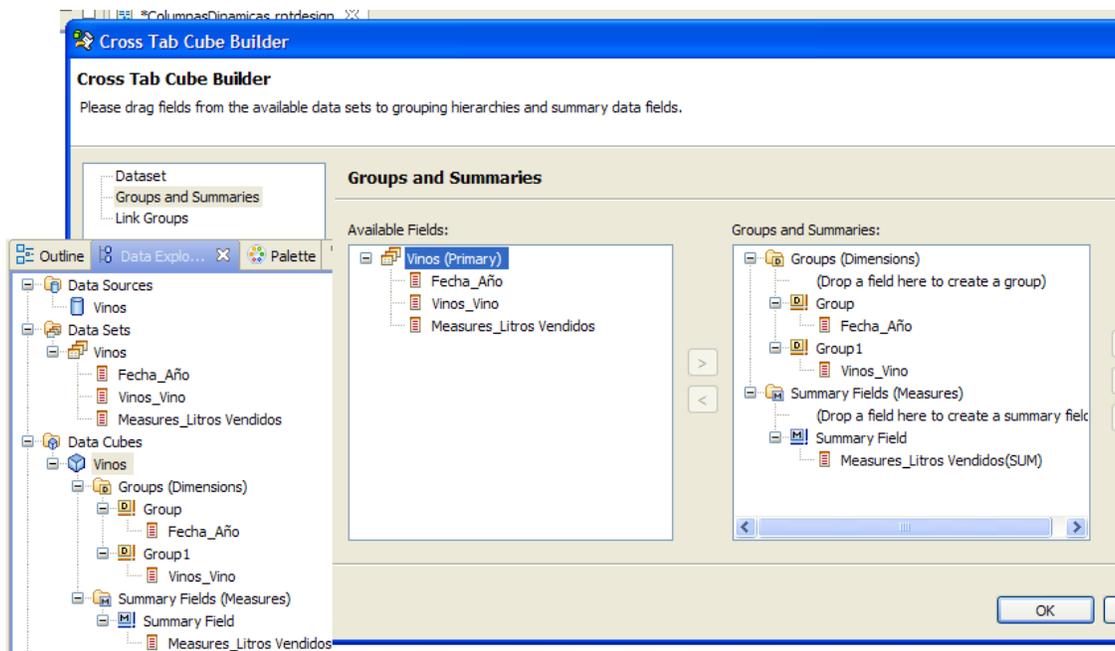
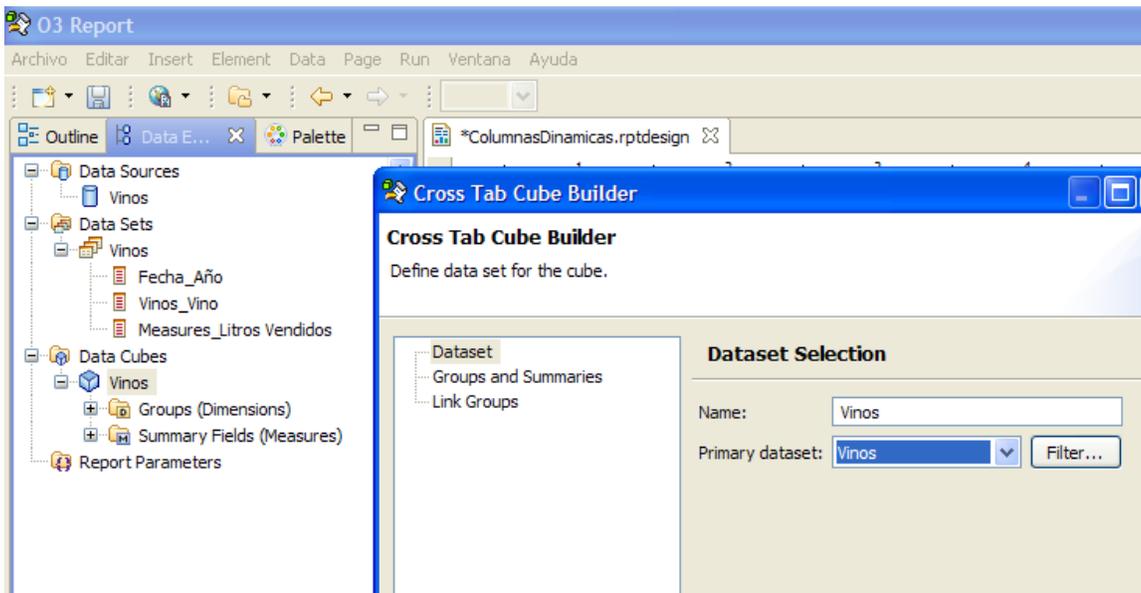
Uso de Cross Tab

Como indicáramos anteriormente en vez de utilizar una tabla, vamos a utilizar un elemento de **O3Report** llamado *Cross Tab*

En la ventana de *Palette* encontramos el elemento *Cross Tab* y lo arrastramos a la posición que deseamos en el Reporte.



Generamos un nuevo *Data Cubes* (Vinos) y los respectivos Groups (Dimensiones) y Summaries (Medidas), que utilizaremos en el reporte.



Luego arrastramos los elementos que deseamos pertenezcan en el reporte, Año para la celda superior derecha (número de columnas variables), celda inferior izquierda para los Vinos y la celda inferior derecha para los litros vendidos.

Se pueden realizar todos los mejores de formato de información y de celdas como de costumbre.

De esta manera si aplicamos la generación del reporte, obtendremos el mismo con tres años de ventas (tres columnas).

	2007	2008	2009
	Litros Vendidos	Litros Vendidos	Litros Vendidos
Brut Blanc de Noirs	2700,0	9000,0	9000,0
Cabernet Franc Rosado	8550,0	9450,0	11700,0
Cabernet Franc Tinto	7200,0	16650,0	8550,0
Cabernet Sauvignon Roble	13500,0	5850,0	11250,0
Cabernet Sauvignon Rosado	16650,0	1800,0	3150,0
Cabernet Sauvignon Tinto	11700,0	6300,0	15750,0
Cabernet Sauvignon, Merlot	4050,0	10350,0	20700,0

Luego de cargar al datamart con nueva información (año 2010), generamos el reporte y obtenemos una columna nueva ... con el nuevo año.

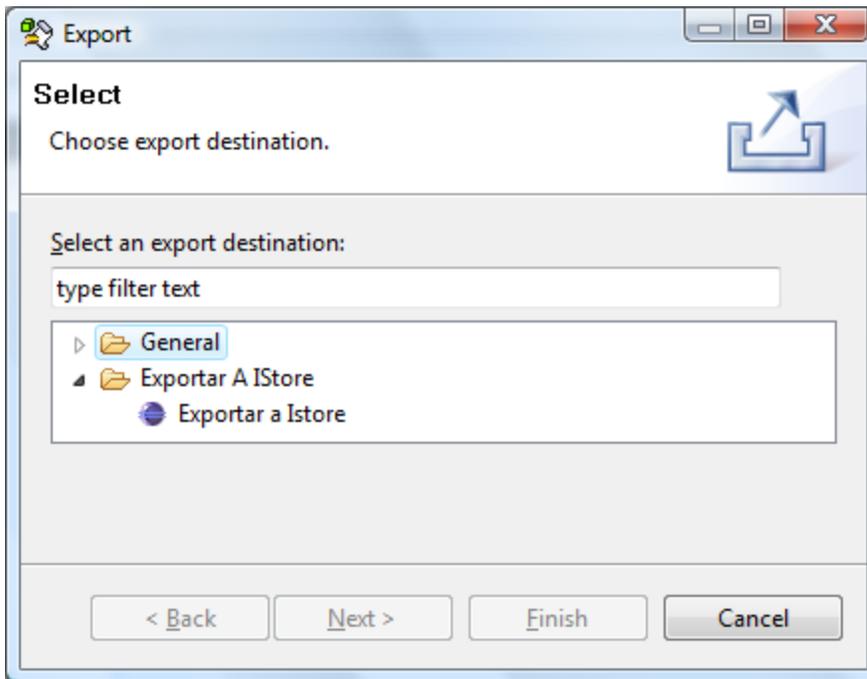
	2007	2008	2009	2010
	Litros Vendidos	Litros Vendidos	Litros Vendidos	Litros Vendidos
Brut Blanc de Noirs	2700,0	9000,0	9000,0	13700,0
Cabernet Franc Rosado	8550,0	9450,0	11700,0	16200,0
Cabernet Franc Tinto	7200,0	16650,0	8550,0	3150,0
Cabernet Sauvignon Roble	13500,0	5850,0	11250,0	6750,0
Cabernet Sauvignon Rosado	16650,0	1800,0	3150,0	6500,0
Cabernet Sauvignon Tinto	11700,0	6300,0	15750,0	9000,0
Cabernet Sauvignon, Merlot	4050,0	10350,0	20700,0	6750,0
Cabernet, Tannat, Merlot	9000,0	15300,0	7200,0	9650,0
Chardonnay	8100,0	11700,0	14400,0	4300,0
Chenin	19350,0	6300,0	5850,0	15500,0
Chenin, Gewurzt	7200,0	25650,0	450,0	11950,0
Folle Blanche	10800,0	13050,0	4500,0	13050,0
Garnacha	6300,0	11700,0	11250,0	6300,0
Gewurztraminer	4050,0	5850,0	8100,0	11700,0
Lambrusco	11250,0	14400,0	5850,0	

Almacenando el reporte

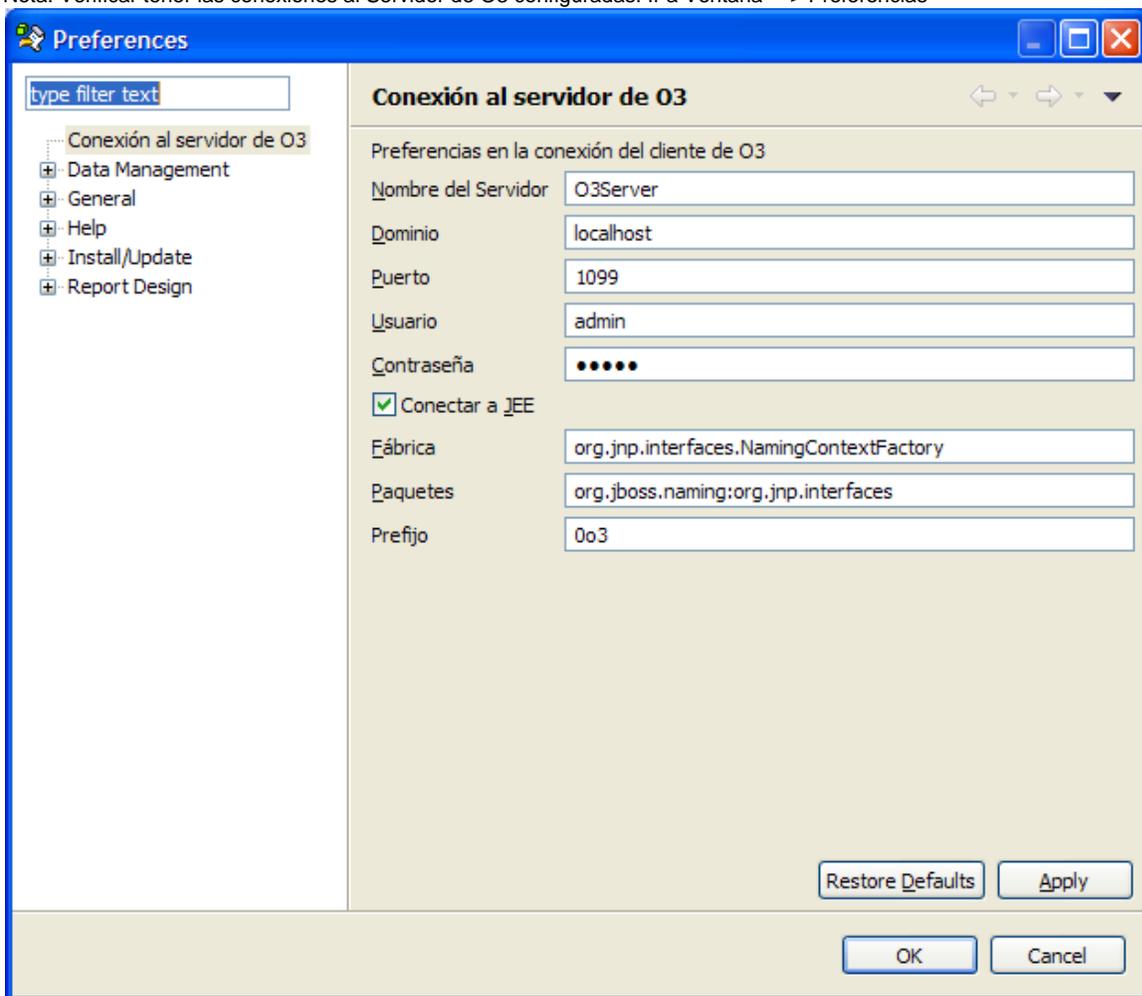
Una vez definido el reporte se debe grabar la definición del mismo. Esta operación tiene por objetivo almacenar el diseño del reporte realizado.

Para que el usuario pueda consultar el reporte es necesario que el mismo esté disponible en el servidor. Para ello es preciso Exportar el reporte luego de ser diseñado.

1. Desde el menú File, seleccionar la opción Exportar
2. Seleccionar la opción Exportar a IStore. Esto permite que el reporte sea generado en el servidor y quede accesible para los usuarios



3. Nota: Verificar tener las conexiones al Servidor de O3 configuradas. Ir a Ventana ---> Preferencias



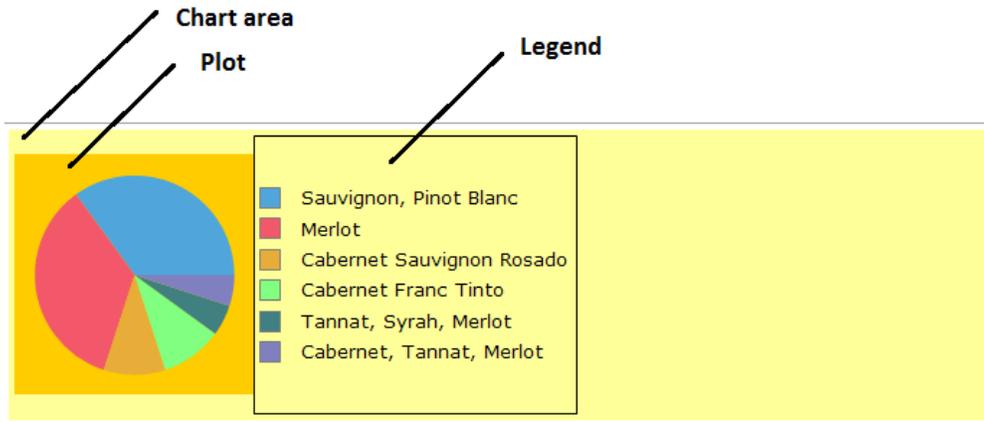
Envío automático de reportes

Dando formato a las leyendas de una gráfica

En esta sección se explican los elementos básicos para dar formato a las leyendas de una gráfica.

Los elementos básicos de una gráfica son los siguientes, y se muestran en el gráfico a continuación.

- Chart area (contiene todas las partes de una gráfica, como ser el título, el gráfico, las leyendas)
- Plot (contiene el gráfico en sí)
- Legend (las leyendas del gráfico)



Ejemplo de uso

Para mostrar la configuración del formato, se muestra un ejemplo de gráfica de torta con un problema, y se muestran los pasos a seguir para resolverlo.

A continuación se muestra una captura de un reporte. Como se puede observar, las leyendas de la gráfica se muestran cortadas. Para resolver este problema es necesario ajustar el ancho de las leyendas, y el ancho de la gráfica. Primero el ancho de las leyendas, porque se están mostrando truncadas.



Luego de resuelto esto, las leyendas quedan con ancho distinto en cada gráfica lo que resulta en plots de distintos tamaños (porque estos se ajustan al tamaño de la leyenda). Por lo tanto es necesario definir un ancho fijo para los gráficos.

Inicialmente, cuando se crea la gráfica, todos los plots se muestran del mismo tamaño a pesar de tener un tamaño que se ajusta automáticamente, porque el espacio asignado para la leyenda (Legend Wrapping Width) es muy restringido y todas se muestran cortadas.

Haciendo doble click en una gráfica en Report, se abre el cuadro de edición. En la tercera pestaña, "Format Chart" se encuentran las opciones de formato. Clickeamos en "Legend" en el árbol de la izquierda. Abajo clickeamos en el botón "Layout", para que se abra la ventana Legend Layout, como se muestra en la siguiente imagen.

- Plot width: 100
- Legend Wrapping width: 200
- Legend: Max Percent: 100

Andrea merteris



En esta última captura se observa que no todos los elementos de la leyenda son desplegados. Esto se resuelve aumentando la altura del elemento de la gráfica (Property Editor -> Properties -> General -> Height).