

# Fast Report ©



## Manual del usuario (x) Harbour - Xailer

Introducción.....	3
Características principales de FR.....	4
Tema 1. Componentes del generador de informes.....	5
Tema 2. Uso del generador de informes.....	6
Tema 3. Uso del diseñador.....	7
Tema 4. Barras de herramientas del diseñador de Informes.....	11
Tema 5. Teoría de Bandas.....	14
Tema 6. Configurando el entorno de trabajo.....	17
Opciones del diseñador.....	17
Opciones de página.....	18
Opciones del Informe.....	19
Tema 7. Creando nuestro primer informe columnar.....	21
Tema 8. Creando un informe de etiquetas.....	24
Tema 9. El objeto texto o memo.....	26
Mostrando expresiones en un objeto texto de FR.....	28
Campos de tablas de datos.....	28
Caracteres especiales como corchetes [ ] o < >.....	28
Formateando un texto y resaltándolo automáticamente.....	30
El editor de Macroexpresiones.....	31
Tema 10. Ampliar FastReport con nuestras propias funciones.....	33
Tema 11. Manipular la vista previa de FR.....	34
Tema 12. FASTREPH.PRG y Xailer.....	37
Tema 13. Métodos de FastReport.....	38
13.1 Métodos constructor y destructor.....	38
13.2 Métodos para cargar diseños, diseñar informes y grabar estos diseños.....	39
13.3 Métodos para presentar, imprimir y exportar informes.....	42
13.4 Trabajando con variables y bases de datos.....	50
15.4.1 Variables.....	50
15.4.2 Trabajando con WorkAreas (Areas de trabajo).....	60
13.5 Opciones de Impresión.....	69
13.6 Opciones del diseñador visual.....	72
13.7 Opciones de la Vista Previa.....	73
13.8 Otros métodos de la clase frReportManager.....	80
13.9 Generación Manual de Reportes.....	81
13.10 Opciones del motor de impresión.....	85
13.11 Métodos universales del objeto frReportManager.....	88
Tema 14. Ejemplos de informes hechos con FastReport.....	92
Informe columnar simple.....	92
Informe columnar con grupos anidados.....	96
Informe columnar maestro - detalle.....	100
Informe columnar con subinforme.....	102
Informes con índices o miniaturas.....	104
Informe columnar con multicopia y un texto identificativo por copia.....	106
Informe con multiples diseños de página.....	108
Informe con gráficas de datos.....	110
Informe de tablas cruzadas.....	112
Impresión de documentos sobre plantilla escaneada.....	114
Informe sin DataSets, impresión de un recibo.....	115
Informe hoja de calendario.....	118
Informe columnar con líneas de altura variable.....	120
Impresión de documentos de tipo factura.....	122
Informe columnar de tarjetas con código de barras y fotografía.....	131
Apéndice 1: Atajos del teclado y del ratón en el editor.....	133

## **Introducción.**

FastReport (FR) es un generador visual de reportes (informes) para el compilador (x)Harbour que proporciona a las aplicaciones la capacidad de generar informes de múltiples tipos, rápida y eficientemente.

Este generador de reportes provee de todas las herramientas necesarias para desarrollar informes complejos y muy atractivos a la vista.

FastReport tiene variantes para varios lenguajes de programación incluyendo:

- Microsoft Visual Studio.
- Microsoft Visual Basic.
- Microsoft Access.
- Microsoft FoxPro.
- SyBase PowerBuilder .
- Delphi, que es el lenguaje nativo en el cual fue desarrollado el generador de informes.
- xHarbour.
- Alaska xBase.
- .Net

FastReport para (x)Harbour está basado en el generador de informes FastReport de la empresa Fast Report Inc. <http://www.fast-report.com/es/> y fue específicamente adaptada a xHb por Sergey Spirin de la empresa ParitetSoft [http://www.paritetsoft.ru/FastReport\\_para\\_\(x\)Harbour.htm](http://www.paritetsoft.ru/FastReport_para_(x)Harbour.htm) que es quien comercializa esta versión en particular con licencia de Fast Report Inc.

FastReport para (x)Harbour se integra con el compilador Harbour o xHarbour por medio de una clase que se provee en código fuente (.PRG) y que contiene todos los métodos, propiedades y eventos para controlar todos los aspectos de la generación de informes desde nuestro código fuente.

FastReport para (x)Harbour funciona para aplicaciones (x)Harbour en modo consola (MS-DOS) y también con todas las interfaces gráficas compatibles con este compilador, incluyendo Xailer, Visual xHarbour, MiniGUI, OOHG, FiveWin, etc.

La interfaz visual de FastReport para (x)Harbour es muy simple de utilizar, además de no requerir complicados procesos de instalación, no se necesita instalar absolutamente nada en cada equipo que haga uso de él, solo añadir una DLL a la carpeta de la aplicación y un pequeño archivo del idioma específico.

FastReport para (x)Harbour se integra nativamente con el lenguaje de programación, pudiendo hacer uso de variables, métodos o funciones definidas en nuestros programas y con soporte para cualquier formato de datos que nuestra aplicación. Puede usar desde tablas DBFs con índices NTX o CDX, ADTs de Advantage Database Server y datos provenientes de motores de base de datos por medio ODBC o ADO o datasets de MySQL - MariaDB o SQLite e incluso simples arrays de datos.

Con FastReport se puede hacer todo tipo de impresos desde informes columnares, etiquetas, tarjetas de identificación, gráficas estadísticas, códigos de barras, hojas de calendario, informes de tablas cruzadas, hasta impresos del tipo de facturas, albaranes, tickets de gran longitud, impresión de fotografías dentro de todo lo anterior, impresión de texto enriquecido, etc.

Todo lo que se pueda imprimir desde un programa en cualquier tipo de impresora se puede hacer con FR.

También sirve para generar informes desde un CGI en Html que pueden ser mostrados directamente en páginas Web de consulta.

## Características principales de FR

- Sin complicadas instalaciones, solo necesita un DLL que se distribuye sin royalties y que va en la carpeta del programa sin instalación ni registro alguno.
  - Poderoso diseñador visual, multiidioma incluido el castellano, fácil de usar y disponible en tiempo de ejecución del programa.
  - Totalmente WYSIWYG
  - Orientado a “bandas” de información.
  - Soporta todo tipo de componentes: Líneas, cuadros, textos, imágenes, OLE, graficas estadísticas, códigos de barras, richtext, etc.
  - Capacidad para exportar los informes a múltiples formatos con un gran número de opciones (PDF, HTML, RTF, CSV, TXT, XLS, XML, JPG, BMP, TIFF, GIF, ODS, ODT, eMail), desde la vista previa de la impresión o por código fuente con formulario de opciones o directamente por código fuente y si se desea sin mostrar absolutamente nada en pantalla.
  - Soporte para todo tipo de impresoras incluidas las de matriz de puntos.
  - Editor de scripts interno con sintaxis resaltada.
  - Arquitectura multi-lenguaje que permite usar varios lenguajes para scripting como PascalScript, C++Script, Javascript y BasicScript.
- NOTA:** Para que funcione todo el soporte de (x)Harbour ha de utilizarse únicamente PascalScript
- Depurador del código fuente de scripting.
  - Acceso en tiempo de ejecución a variables, funciones, matrices y tablas de datos de la aplicación (x)Harbour.
  - Gran velocidad en la generación de los informes y en su conversión a cualquier formato.
  - Arquitectura de librerías fácilmente expandible. Hay librerías de terceros para realizar por ejemplo códigos QR bidimensionales que el FR no contempla.
  - Vista previa de los informes con opciones de búsqueda de texto y edición de datos en vista previa
  - Rotación de textos de 0 a 360 grados
  - Justificación de párrafos.
  - Los objetos memo (para textos) soportan etiquetas html simples (Font,color,b,i,u,sub,sup).
  - Tipos de fuentes, tamaños, estilos y colores.
- NOTA:** se puede usar cualquier fuente del PC de desarrollo, pero si se ha de distribuir la aplicación y el usuario final no tiene instalada la fuente utilizada el texto en esta fuente será sustituido por otra diferente y en las exportaciones a formatos como PDF puede que se pierda.
- Flujo de textos con WordWrapping.
  - Posibilidad de incluir URLs en el texto con link funcional desde la vista previa.
  - Ampliación (Zoom) en el diseñador visual.
  - Reglas y guías.
  - Rejilla configurable en mm, pulgadas y pixeles.
  - Asistente para los informes más comunes.
  - Copia de objetos al portapapeles de Windows.
  - Deshacer / Rehacer ilimitado.
  - Los informes pueden contener sus propios formularios, por ejemplo para pedir parámetros antes de preparar un informe. Se pueden tener tantos diálogos como sea necesario. FastReport usa el mismo diseñador para crear y mantener los diálogos y soporta los controles de Windows mas comúnmente usados como botones, gets, checkboxes, etc. También podemos llamar a diálogos de nuestra aplicación y recoger lo que en ellos se introduzca para configurar un informe, antes de mostrarlo e imprimirlo.
  - Almacenamiento de los informes en formato XML, informes XML comprimidos (compatibles con GZip) también están soportados.
  - Salida bidireccional de texto para idiomas específicos.
  - Permite integrar componentes de terceros a través de OLE como por ejemplo hojas de Excel.
  - Todo tipo de tamaños de página, incluyendo tamaños personalizados o incluso sin límite.
  - Incluye una Demo 100% operativa con la única limitación de no generar más de 5 páginas de texto.

## Tema 1. Componentes del generador de informes

FastReport consta de 4 archivos principales:

1. **FRSYSTH.DLL** - es la librería base que será usada por la aplicación. Se puede colocar en cualquier directorio del disco duro, aunque lo recomendable es que se encuentra en la misma donde se instala el archivo ejecutable. No precisa de instalación ni registro alguno.
2. **FASTREPH.PRG** - Archivo con código fuente que contiene todas las clases, métodos, procedimientos y funciones necesarias para trabajar con FastReport. Este archivo debe de ser linkado al proyecto del programa a generar.
3. **FASTREPH.CH** - es el archivo de cabecera (include file) que contiene las constantes predefinidas para el uso del archivo FASTREP.CH, se debe agregar al proyecto.
4. **SPANISH.FRC** - El archivo de idioma para cada idioma concreto, en nuestro caso español, que es el archivo que contiene todos los textos del editor visual y del previsualizador. Se debe de colocar en el mismo directorio del disco duro donde se encuentre la DLL. Hay 30 idiomas disponibles y se cargan desde código pudiendo cargarse uno u otro si se desea.

Para comenzar a trabajar con FastReport básicamente tendremos que agregar a nuestro proyecto los archivos de código fuente provistos con FR: el FASTREPH.PRG y el FASTREPH.CH y desde nuestro código fuente llamar a los métodos necesarios para instanciar y manipular FR.

Una vez que tenemos nuestro archivo EXE generado, será necesario tener disponible la DLL que contiene a FR, así como el archivo que contiene la traducción de la interfaz del usuario Spanish.Frc.

~~Para descargar este conjunto de archivos ha de hacerse desde [http://www.paritetsoft.ru/FastReport\\_para\\_\(x\)Harbour.htm](http://www.paritetsoft.ru/FastReport_para_(x)Harbour.htm) pudiendo descargarse en un primer momento la versión demostrativa y pudiendo adquirirla después a un precio de unos 300,00 €. En esta misma web podemos descargar también diferentes manuales (solo en inglés) y ejemplos ya compilados con xHb o con Xailer y ver la potencia de los informes que genera FR.~~

## **Tema 2. Uso del generador de informes**

FastReport no es un programa al uso, no es un archivo EXE independiente que podamos ejecutar, en realidad es un generador de informes que se empotra en la aplicación que se esté programando y que lo contiene de forma que solo funciona cuando dicha aplicación está en ejecución.

Para ello bastará simplemente que en el código fuente de dicha aplicación se introduzca el siguiente código.

```
oFr:=frReportManager():New()
oFr:DesignReport()
oFr:DestroyFR()
```

Y al correr la aplicación y ejecutar este código, se mostrará el diseñador completo con todas sus características, con lo que se podrá diseñar un informe y grabarlo con un nombre como por ejemplo “Informe de ventas.fr3” (a pesar de que FastReport se encuentra en su versión 4 la extensión por defecto sigue siendo FR3) que es un archivo Xml que podemos abrir con cualquier editor de texto, e incluso editar si fuera necesario.

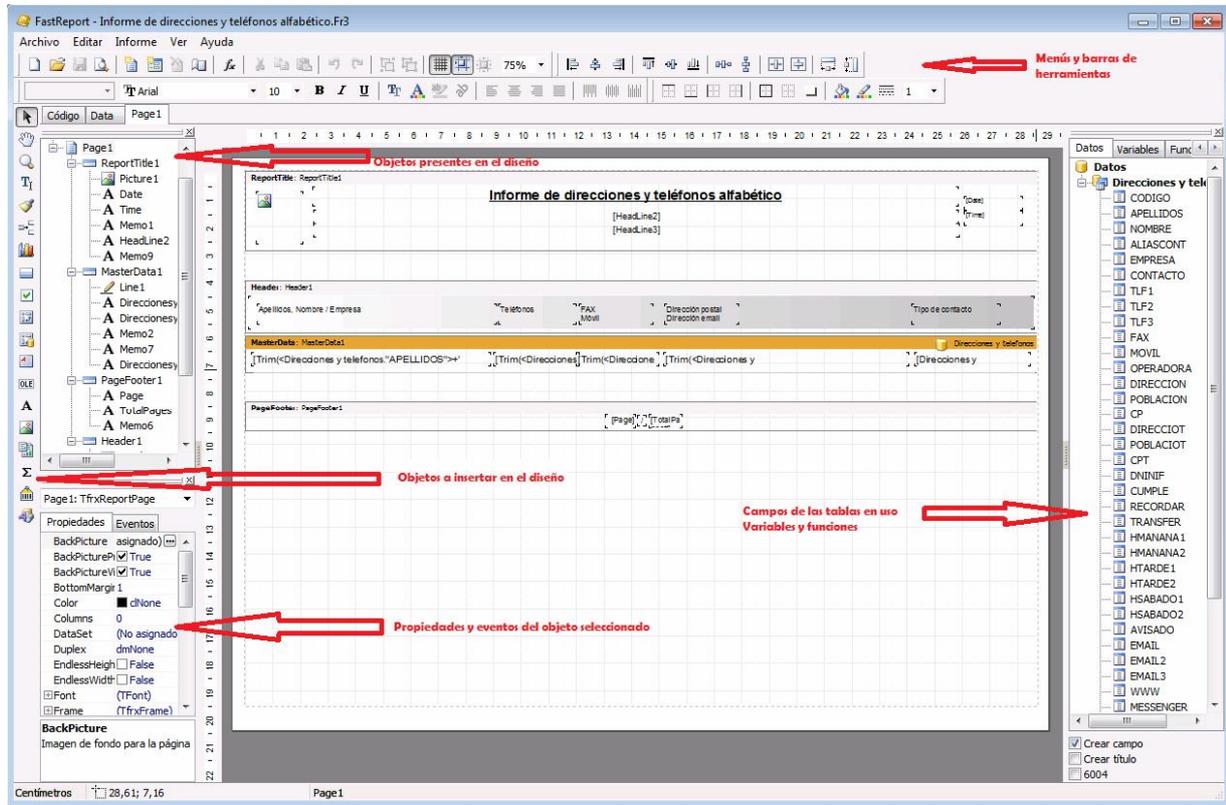
Pero para poder lanzar un informe ya confeccionado a vista previa o a impresión, habrá que añadir un par de líneas más.

```
oFr:=frReportManager():New()
oFr:LoadFromFile(“Informe de ventas.fr3”)
oFr:ShowReport()
oFr:DestroyFR()
```

Realmente simple verdad, pero FR no solo es esto ya que además dispone de un buen número de otros métodos y propiedades para realizar informes desde lo más básico hasta lo más sofisticado.

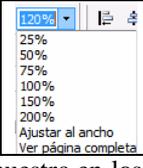
## Tema 3. Uso del diseñador.

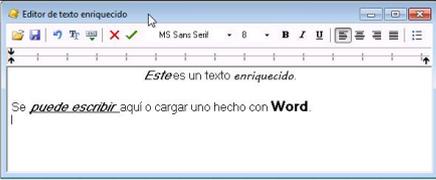
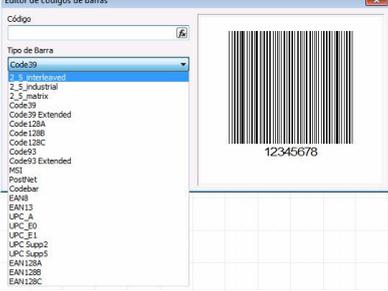
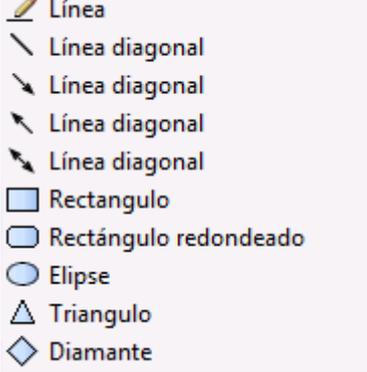
Una vez mostrado en pantalla el diseñador visual, este tendrá el siguiente aspecto:



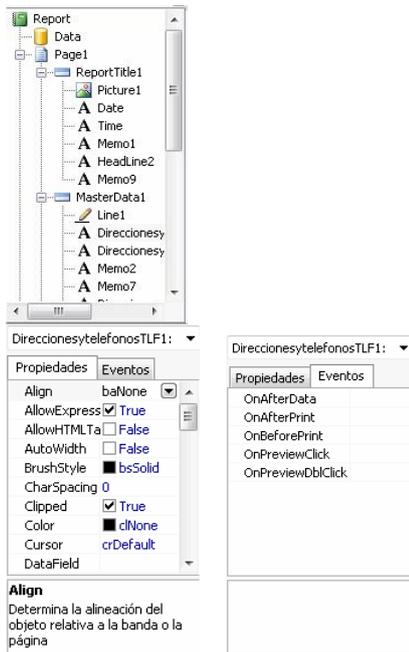
### Objetos a insertar en el diseño.

A la izquierda se muestran una serie de acciones y objetos a insertar en el diseño, objetos de variados tipos.

 Seleccionar	Selector estándar que sirve para la acción de seleccionar los objetos del diseño y para moverlos.	
 Arrastrar	Se utiliza para mover la hoja en diseño de FastReport cuando no cabe en la pantalla, sería lo mismo que utilizar las barras de desplazamiento.	
 Zoom	Se usa para ampliar o reducir el diseño, se puede utilizar también Ctrl+Rueda del ratón o el Zoom de la barra de herramientas superior.	
 Texto	Es la herramienta más útil de todas ya que básicamente lo que se muestra en los informes es texto. Permite dibujar un área de un tamaño que luego incluirá el texto. Más adelante se mostrarán sus múltiples capacidades.	
 Copiar formato	Seleccionando previamente un objeto y pulsando en este control se copia el formato, tipo de letra, tamaño, colores, ajustes y demás del objeto para que después haciendo clic en otro objeto le sean transferidos.	
 Bandas	Son el elemento contenedor del resto de elementos visuales del informe, de gran importancia y que serán vistos con detalle más adelante.	
 Gráficas	Permiten incorporar al informe gráficas de tipo barras, líneas y sectores con los datos de una tabla o introducidos manualmente.	
 Degradados	Permite hacer áreas degradadas de dos colores sobre el diseño.	
 CheckBox	Permite introducir un checkbox en el documento, que asociado después a un campo de valor lógico se muestra lleno <input checked="" type="checkbox"/> o vacío <input type="checkbox"/>	
 Informe cruzado	Permite construir una tabla de datos cruzados sobre datos suministrados desde el diseñador.	
 Informe datos cruzados	Permite construir una tabla de datos cruzados sobre datos de las tablas en uso. Mas adelante se verá la gran potencia de esta herramienta.	

 Texto RTF	<p>Muestra un pequeño editor de textos que permite editar un texto con todo tipo de letras, o abrir un documento previamente hecho en Word y grabado como RTF. Como curiosidad se pueden insertar en el texto campos de las tablas, fórmulas, variables o funciones de nuestra aplicación.</p>	
 Objeto OLE	<p>Incorpora al documento cualquier objeto de cualquier otro programa que tenga OLE, por ejemplo podremos introducir una hoja de Excel de una carpeta del disco y además si lo deseamos vincular el objeto a esa hoja de Excel y así si cambia la hoja de Excel se readapta el objeto OLE automáticamente.</p>	
 Texto	<p>Al igual que el anterior icono  introduce un campo de texto pero con esta opción no podemos darle un tamaño directo sino que toma un tamaño por defecto, aunque luego se podrá cambiar su tamaño. Más adelante se mostrarán sus múltiples capacidades.</p>	
 Imagen	<p>Permite añadir al texto una imagen, que puede ser una imagen cualquiera del disco, una referencia a una imagen como variable (un logo que puede cambiar) o una referencia a una imagen almacenada en un campo de una tabla, de forma que en un listado columnar una de las columnas podrían ser fotografías.</p>	
 Subinforme	<p>Aunque lógicamente FastReport como diseñador de informes permite hacer grupos de información un subinforme va aún más lejos ya que permite hacer informes dentro de informes, p.e. si creamos un informe columnar de facturas se puede hacer en cada cambio de factura un informe de los albaranes que la conforman. El subinforme se puede hacer con la misma tabla o con tablas diferentes pero al fin y al cabo forma una sola unidad y sus datos pueden aparecer mezclados. También veremos más adelante que FastReport permite hacer informes múltiples en un solo informe que saldrían consecutivos como uno único pero no mezclados.</p>	
 Texto del sistema	<p>Inserta en el documento un campo con un valor que puede ser una variable del sistema, una fórmula o un valor calculado. Se puede hacer lo mismo insertando un texto normal y usando el generador de expresiones.</p>	
 Código de barras	<p>Permite introducir en el diseño códigos de barras unidimensionales de múltiples tipos a los que se puede asociar un campo de la tabla o introducir un valor calculado o directamente un número de forma manual.</p> <p>Los códigos de barras bidimensionales pueden hacerse a través de herramientas de terceros.</p>	
 Dibujo	<p>Permite dibujar dentro del diseño objetos de diferente tipo a los que después se puede cambiar el tipo de línea, su grosor, su color de línea y de fondo usando las herramientas de la barra superior:</p> 	

Si se está habituado al uso del IDE de Xailer se verá que el resto de la distribución es muy parecida ya que a la izquierda de forma parecida al gestor de proyectos, se encuentra la lista de los objetos presentes en el diseño en un orden jerárquico, cada uno dentro de su contenedor y debajo de estos las propiedades y eventos del objeto seleccionado en el diseñador.



La lista de objetos del diseño permite situarse en un concreto para poder trabajar con el aunque lógicamente sería mas sencillo seleccionarlo en el propio diseño.

Pero en el caso de tener objetos que en el diseño están tapados por otros o sean difíciles de seleccionar por cualquier otro motivo, podemos recurrir a esta lista que nos muestra en orden jerárquico todos los objetos dentro de cada objeto contenedor.

Cada objeto seleccionado muestra sus propiedades y sus eventos que pueden llamar a métodos programables desde el propio FR.

Cada propiedad nos muestra debajo una pequeña explicación de para que sirve y nos permite modificar su valor de forma asistida.

Cada distinto tipo de objeto tendrá diferentes propiedades y eventos, muchos son comunes como el color o la alineación, pero otros son específicos de cada objeto y por lo tanto es imposible condensar ahora todos ellos, pero en este manual se verán referencias a muchos de ellos.

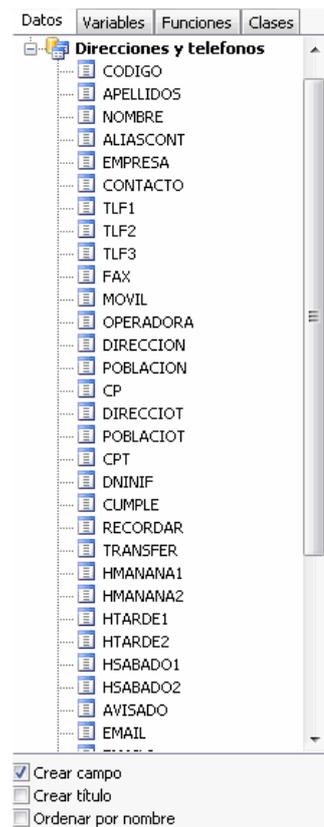
A la derecha del diseño se puede ver otra área, donde encontramos diversos objetos que se pueden incorporar al diseño.

Primeramente la solapa datos muestra las tablas que hemos suministrado a FastReport para este informe, con todos sus campos, sean del tipo que sean.

Bastará con seleccionarlos y arrastrarlos al diseño para agregarlos al informe, cuidado para que esto funcione ha de estar marcado abajo el checkbox de crear campo.

Si además se marca el de crear título al seleccionar el campo y llevarlo al diseño aparece sobre el campo el nombre que tiene en la tabla, el que estamos viendo en la lista.

Si se desea también se puede seleccionar el checkbox ordenar por nombre y tendremos la lista anterior en orden alfabético.



Si cambiamos de pestaña a Variables se obtiene una lista con variables del sistema FastReport y además se pueden tener variables que hayan sido enviadas desde nuestra aplicación al instanciar FastReport.

Pero además FastReport puede obtener el valor de cualquier variable definida en nuestra aplicación directamente mediante una función GetHBVar salvo el valor de las matrices y para obtener el valor de un elemento de una matriz también tenemos otra función GetHBArrayVar.

Estas funciones y muchas más se encuentra en la siguiente solapa llamada Funciones.

Las hay de variados tipos



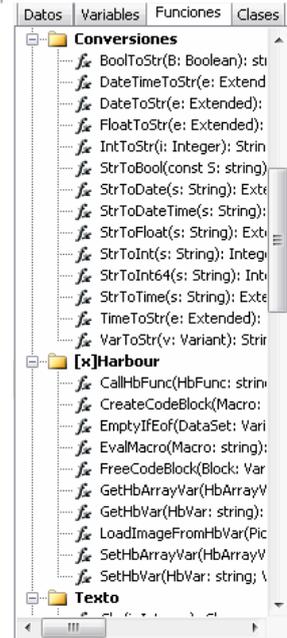
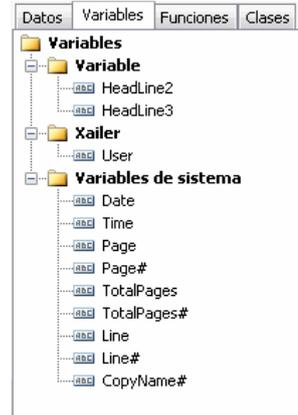
Pero entre todas ellas, la magia de esta versión de FastReport para (x)Harbour, está en la sección específica que contiene para este compilador/es ya que nos permite un control bidireccional total desde FastReport ya que puede llamar a variables, arrays, funciones de (x)Hb o de nuestro programa, e imprimir cosas en consecuencia.

De esta manera la integración con nuestro código es total.

Se puede arrastrar una función desde aquí al diseño o también se puede usar el generador de expresiones dentro de cualquier campo memo (campo de texto) para construir una función de forma asistida.

Además FastReport permite ampliar las funciones con las que trabaja añadiéndole nuevas funciones desde nuestro código fuente usando la función AddFunction( )

Por último tenemos la solapa de Clases que son aquellas clases que nos provee FastReport de forma nativa para poder hacer entre otras muchas cosas conexiones directas a Bases de Datos ADO, construir ventanas de diálogo, etc.



## Tema 4. Barras de herramientas del diseñador de Informes

El diseñador de informes proporciona las herramientas necesarias para definir el contenido y la apariencia del informe, y mostrarlo tal y como va a ser visto por el usuario final.

En la parte superior de la ventana contiene varias barras de herramientas que pueden moverse donde se desee.

- ✓ Estandar
- ✓ Texto
- ✓ Marco
- ✓ Paleta de Alineación

Las barras son cuatro, la barra estándar, la de texto, la de marco y la paleta de alineación.

En caso de no ver esta última que no es mostrada por defecto vaya al menú Ver – Barras de herramientas para seleccionarla y que sea mostrada en adelante.



La información sobre la situación de barra y todos los demás elementos personalizables se guardan en un archivo del sistema y se restauran la próxima vez que se accede al mismo.

### Barra de herramientas Estándar.



Icono	Nombre	Descripción
	Nuevo informe	Crea un nuevo informe en blanco.
	Abrir informe	Abre un informe previamente grabado en el disco [Ctrl]+[O]
	Grabar informe	Graba en el disco el informe actual del diseñador [Ctrl]+[S]
	Vista previa	Muestra la vista previa del informe [Ctrl]+[P]
	Nueva página	Añade una nueva página al diseñador. Un informe multipágina es un informe con más de un diseño, siendo cada página un informe en si mismo (con tantas páginas de papel como necesite) que se suma al siguiente, imprimiéndose todo seguido como uno solo.
	Nuevo diálogo	Añade un nuevo diálogo (form) al diseño. Un diálogo se puede utilizar para presentar al usuario una serie de opciones con las que parametrizar el informe final.
	Borrar página	Borra una página del diseñador.
	Propiedades de la página	Muestra el diálogo de propiedades de la página. Cada Página del diseñador puede tener unas propiedades diferentes.
	Variables	Muestra el diálogo del editor de variables, donde se pueden ver las variables de FastReport, aquellas que hayan sido cargadas desde código y se pueden añadir, editar o eliminar variables de la lista.
	Cortar	Corta al portapapeles el elemento seleccionado [Ctrl]+[X]
	Copiar	Copia al portapapeles el elemento seleccionado [Ctrl]+[C]
	Pegar	Pega el elemento que haya en el portapapeles [Ctrl]+[V]
	Deshacer	Deshace la última operación, o las anteriores una a una [Ctrl]+[Z]
	Rehacer	Rehace la última operación deshecha, o las posteriores una a una [Ctrl]+[Y]
	Mostrar rejilla	Muestra la rejilla de diseño. El tamaño de la rejilla puede ser modificado en el menú Ver – Opciones.
	Alineación a la rejilla	Si está seleccionada esta opción el movimiento de los objetos se ajusta a la rejilla, si no lo está el movimiento es libre.
	Ajusta a rejilla	Ajusta al punto más cercano de la rejilla al elemento/s seleccionado/s
	Zoom	Establece el zoom de pantalla.

**Barra de herramientas Texto.**

Icono	Nombre	Descripción
	Estilo	Permite seleccionar un estilo para un objeto de texto. Un estilo es un conjunto de características aplicables a un texto como fuente, tamaño, color, etc. que se pueden guardar con un nombre. Para definir un estilo hay que ir al menú Informe - Estilos.
	Fuente	Permite seleccionar un fuente para aplicar al texto/s seleccionado/s
	Tamaño del fuente	Permite seleccionar un tamaño para aplicar al texto/s seleccionado/s. También puede introducirse manualmente.
	Negrita	Activa o desactiva la negrita para el texto seleccionado
	Cursiva	Activa o desactiva la cursiva para el texto seleccionado
	Subrayado	Activa o desactiva el subrayado para el texto seleccionado
	Diálogo de fuentes	Muestra el diálogo de características de la fuente para poder hacer todo lo anteriormente descrito desde aquí
	Color del fuente	Permite seleccionar un color para aplicar al texto/s seleccionado/s
	Resaltar	Resalta el texto cambiando su color de letra y de fondo en base a una condición, por ejemplo que los valores negativos se muestren en rojo con el fondo azul.
	Rotación	Permite rotar el texto seleccionado en saltos de 45°
	Ajuste izquierdo	Ajusta el texto a la izquierda de su área contenedora
	Ajuste centrado	Ajusta el texto al centro de su área contenedora
	Ajuste derecho	Ajusta el texto a la derecha de su área contenedora
	Ajustado al ancho	Ajusta el texto a la izquierda y a la derecha de su área contenedora quedando recto por ambos lados salvo para la última línea.
	Ajuste arriba	Ajusta el texto arriba de su área contenedora
	Ajuste centro	Ajusta el texto en el centro vertical de su área contenedora
	Ajuste abajo	Ajusta el texto abajo de su área contenedora

## Barra de herramientas de marco



Icono	Descripción
	Pinta la parte superior del cuadro del objeto/s seleccionado/s.
	Pinta la parte inferior del cuadro del objeto/s seleccionado/s.
	Pinta la parte izquierda del cuadro del objeto/s seleccionado/s.
	Pinta la parte derecha del cuadro del objeto/s seleccionado/s.
	Pinta los cuatro lados del cuadro del objeto/s seleccionado/s.
	Borra la línea de los cuatro lados del cuadro del objeto/s seleccionado/s.
	Pinta un sombra al cuadro del objeto/s seleccionado/s.
	Establece el color de fondo del cuadro del objeto/s seleccionado/s.
	Establece el color de la línea del cuadro del objeto/s seleccionado/s.
	Establece el tipo de línea del cuadro del objeto/s seleccionado/s.
1	Establece el grosor de la línea del cuadro del objeto/s seleccionado/s.

## Barra de herramientas de alineación

NOTA: Hay que seleccionar varios objetos para alinearlos y siempre manda el primero seleccionado, que no se ve alterado en posición ni tamaño, ajustándose los demás a su posición o tamaño.

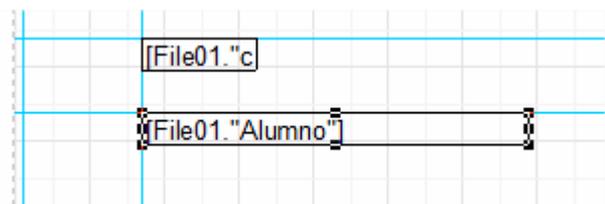


Icono	Descripción
	Alinea los objetos seleccionados por su izquierda
	Centra los objetos seleccionados horizontalmente
	Alinea los objetos seleccionados por su derecha
	Alinea los objetos seleccionados por arriba
	Centra los objetos seleccionados verticalmente
	Alinea los objetos seleccionados por abajo
	Distribuye el espacio horizontal entre los objetos seleccionados sin mover ni el primero ni el último
	Distribuye el espacio vertical entre los objetos seleccionados sin mover ni el primero ni el último
	Centra los objetos horizontalmente en su banda
	Centra los objetos verticalmente en su banda
	Iguala la anchura de los objetos (al primer seleccionado)
	Iguala la altura de los objetos (al primer seleccionado)

También es posible utilizar guías de alineación que se pueden obtener haciendo clic sobre las reglas superior o izquierda (han de estar activadas la opciones Reglas y Guías del menú Ver) y sin soltar el clic arrastrando el ratón hacia el diseño obtendremos una línea azul vertical u horizontal respectivamente. Se pueden obtener tantas como se necesiten.

Estas guías son “magnéticas” haciendo que los controles que se aproximen a ellas se “peguen” de forma que se alinean de forma precisa.

Para quitarlas basta con hacer clic en ellas y arrastrarlas de nuevo hasta las reglas o utilizar la opción Eliminar Guías del menú Ver.



## Tema 5. Teoría de Bandas

FastReport, como todos los diseñadores visuales, funciona mediante “Bandas”.

Las bandas son contenedores de datos especializados cada una en un tipo de presentación. Se llaman bandas por que al seleccionarlos y mostrarlos sobre el diseño, aparecen directamente de lado a lado del papel.

Los distintos tipos de bandas de FastReport son:

**Título del informe:** Todo lo que se introduzca en esta banda aparece solo una vez en la parte superior de la primera página de todo el informe, se pueden poner aquí el título, un logotipo, la fecha, la hora o el usuario que ha generado el informe, etc.

**Sumario del informe:** Todo lo que se introduzca en esta banda aparece solo una vez en la parte inferior de la última página de todo el informe por encima del pie de página, se pueden poner aquí los resultados totales de unas columnas numéricas, una firma digitalizada, etc.

**Cabecera de página:** Todo lo que se introduzca en esta banda aparece en todas las páginas del informe en la parte superior, en la primera página debajo del título, se puede poner aquí el nombre de las columnas del informe.

**Pie de página:** Todo lo que se introduzca en esta banda aparece en todas las páginas del informe en la parte inferior, en la última página debajo del sumario, se puede poner aquí el suma y sigue de los cálculos de las columnas del informe, el número de página de páginas, etc.

**Cabecera:** Todo lo que se introduzca en esta banda aparece solo en la primera página del informe en la parte superior debajo del título, se puede poner aquí un subtítulo con las condiciones de lo que se está mostrando en el informe “Desde fecha .... hasta Fecha ...” aunque esto podría ir también en el título.

**Pie:** Todo lo que se introduzca en esta banda aparece solo en la última página del informe en la parte inferior debajo del sumario y encima del pie de página, se puede poner aquí anotaciones, condiciones generales del documento, etc. aunque esto podría ir también en el sumario.

**Datos maestros:** Todo lo que se introduzca en esta banda aparece una vez por cada registro de la tabla seleccionada. Aquí es donde se indican los campos de un informe columnar. Ojo al poner esta banda hay que indicar a FastReport de que tabla tomará sus datos. En un mismo informe puede haber una o más bandas de datos maestros.

**Datos de detalle:** Es una banda asociada a una banda de datos maestros, para poder imprimir informes Maestro-Detalle. Lógicamente trabajará con dos tablas que previamente deberán de haber sido relacionadas tal y como veremos más adelante. FastReport permite tener hasta 6 niveles de detalle dentro del informe, con lo que podríamos tener hasta 7 tablas relacionadas en cascada.

**Cabecera de grupo:** Permite establecer grupos de datos, todo lo que se introduzca en esta banda aparece en cada ruptura de grupo. La ruptura se puede hacer por un campo en concreto, una suma de campos o cualquier función que devuelva un resultado como el mes de una fecha MonthOf(<Facturas."FECHAFACTURA">). FastReport permite el uso de múltiples bandas de cabecera de grupo anidadas.



**Pie de grupo:** Para cada grupo establecido por una cabecera de grupo puede haber un pié de grupo al finalizar el mismo antes del siguiente grupo. Aquí se pueden totalizar campos numéricos para los valores del grupo.

**Hija:** Una banda hija es una banda que cuelga de cualquiera de las bandas anteriores y esta asociada a la banda de la que cuelga comportándose exactamente como ella. La mejor manera de hacer una banda Hija no es desde el selector de bandas sino haciendo clic derecho del ratón directamente sobre una banda ya presente en el diseño y añadiendo una banda hija directamente desde aquí, así no habrá duda de a quien pertenece. Una banda puede contener bandas hijas de 1 a n y se utilizan mediante programación, para mostrar una u otra en función de lo que se desee. Por ejemplo si una MasterData se deja vacía y se le añaden 3 bandas hijas, al mostrarse el listado se puede decidir para cada línea de datos cual es la banda a presentar.

**NOTA:** Las bandas de Pie de página no admiten “Bandas Hijas”

**Cabecera de columnas:** En principio para un informe columnar puro, se comporta como una banda Cabecera de página, todo lo que se introduzca en esta banda aparece en todas las páginas del informe en la parte superior de las columnas, en la primera página debajo del título, se puede poner aquí el nombre de las columnas del informe. La diferencia es que si aparte de un informe columnar mezclamos otras cosas en el mismo informe parte columnar y parte no, será mejor utilizar esta última si queremos que los nombres de las columnas vayan justo en el comienzo de las mismas.

**Pié de columnas:** En principio para un informe columnar puro, se comporta como una banda Pie de página, todo lo que se introduzca en esta banda aparece en todas las páginas del informe en la parte inferior de las columnas, en la última página debajo del sumario, se pueden poner aquí totales de las columnas del informe. La diferencia es que si aparte de un informe columnar mezclamos otras cosas en el mismo informe parte columnar y parte no, será mejor utilizar esta última si queremos que los totales de las columnas vayan justo al final de las mismas.

**Superpuesto:** Todo lo que se introduzca en esta banda aparece directamente en el fondo de la página impresa, superponiéndose el resto de las bandas sobre esta. Se puede utilizar para imprimir marcas de agua en la página

### Bandas verticales.



Son prácticamente las mismas bandas ya descritas pero con la particularidad de que serán mostradas en vertical, es decir, de izquierda a derecha.

No solo no le veo mucho sentido sino que además no funciona muy bien, por ejemplo una banda de Datos maestros vertical no imprime todos los registros sino el primero y punto.

Además si queremos hacer esto para aprovechar la altura del papel es mejor ponerlo en apaisado y utilizar las bandas normales en horizontal.

**NOTA:** Las bandas siempre van de un margen hasta el otro del papel, no se pueden hacer bandas que ocupen solo una parte de éste.

**Trabajar sin Bandas.**

FastReport también permite hacer cosas fuera de las bandas. Cualquier cosa que situemos directamente sobre el fondo del papel del diseño se mostrará directamente en todas las páginas, solo que si lo que ponemos son campos de una tabla se mostrarán solo los del registro activo en la tabla enviada a FR.

Por ejemplo, si queremos imprimir una factura y tenemos dos tablas una de datos de la cabecera y otra con las líneas de la factura podemos enviar ambas tablas a FR, la de cabecera con el registro activo seleccionado y la de líneas con las líneas previamente filtradas, seleccionadas con un scope o con una select de sql. Entonces pasamos a diseñar el modelo de la factura directamente sobre el papel con los campos de la cabecera y creamos una banda de datos maestros para las líneas y listo.

The image shows a screenshot of the FastReport design environment. On the left, the design grid is visible, showing various fields and tables being placed on the page. On the right, the rendered output of the form is shown, including a header with company information, a table of items, and a summary section at the bottom.

**Form Header:**

- Logo: FRODO S.L.
- Address: CORRCA, B. I.
- Company: POLIGU INDUSTRIAL DE BARATRO
- City: BAYO AMURRO
- Area: ARREA

**Table of Items:**

NO ALB.	FECHA	DESCRIPCION	NETO Kg.	Unid.	PRECIO
6		FOR LEVENTE DE MOUNO A. L. C 666/9996 CON EQUIPO D			
6		MAN			

**Summary Section:**

BASE IMPONIBLE	% IVA	IMPORTE IVA	TOTAL FACTURA
18000,00	1000	2880,00	20.880,00

**Payment Information:**

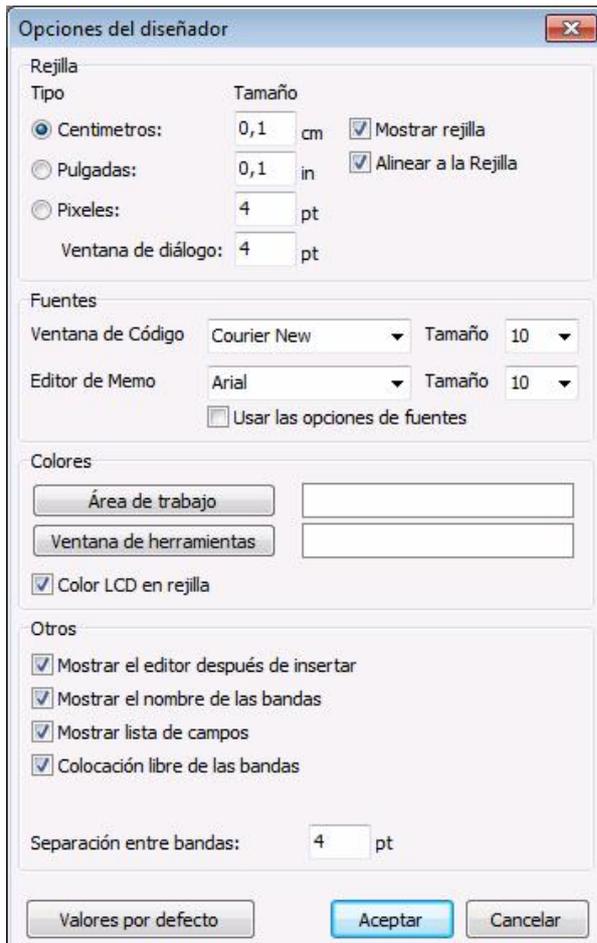
FORMA DE PAGO	NO CUENTA	VENCIMIENTOS

## Tema 6. Configurando el entorno de trabajo.

### Opciones del diseñador

Estas opciones son guardadas en un archivo del sistema y son recuperadas en la próxima sesión.

Para acceder a las opciones del diseñador, seleccione el menú "Ver - Opciones...".



Aquí puede configurar las unidades deseadas (centímetros, pulgadas, píxeles), y especificar el paso para cada unidad.

También puede cambiar de unidad desde el diseñador, haciendo doble clic en la parte izquierda de la barra de estado donde se muestran las unidades actuales.

También puede especificar si la rejilla debe verse. Esto también puede realizarse con los botones   de la barra de herramientas.

Puede establecer una fuente para la ventana del editor de código y para el editor de objetos de "Texto". Si selecciona la opción "Usar las opciones de fuentes" (no recomendado) la fuente en la ventana del editor de texto será la misma que la fuente del objeto que se está editando, lo cual puede resultar bastante incómodo, si la letra en el diseño es por ejemplo muy pequeña o muy grande.

Si el fondo blanco del diseñador y diálogos no le gusta, puede modificarlo con los botones "Área de trabajo" y las "Ventanas de herramientas".

La opción "Color LCD en rejilla" aumenta el contraste de las líneas de la rejilla un poco, y mejora su visibilidad en los monitores LCD.

Si la opción "Mostrar el editor después de insertar" está activada, el editor se mostrará cada vez que un objeto es insertado. Cuando va a crear un gran número de objetos en blanco, se recomienda desactivar esta opción.

Con la opción "Mostrar el nombre de las bandas", puede desactivar los nombres de estas para ahorrar un poco de espacio en la página.

La opción "Mostrar lista de campos" muestra una lista desplegable dentro del campo cuando está conectado a una base de datos para cambiar este campo por otro. Esto puede ser incómodo si los campos son muy pequeños ya que será muy difícil seleccionarlos.

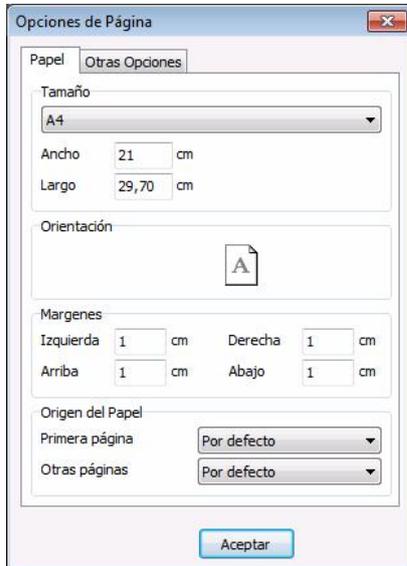
Si la opción "Colocación libre de las bandas" está desactivada las bandas se agrupan de forma automática en la página de acuerdo a su función. Si la activamos podremos disponer las bandas como queramos.

Por último se puede establecer la separación mínima entre estas bandas.

## Opciones de página.

Estas opciones se aplican al diseño actual, en cada nuevo diseño deberán de adaptarse si se desea.

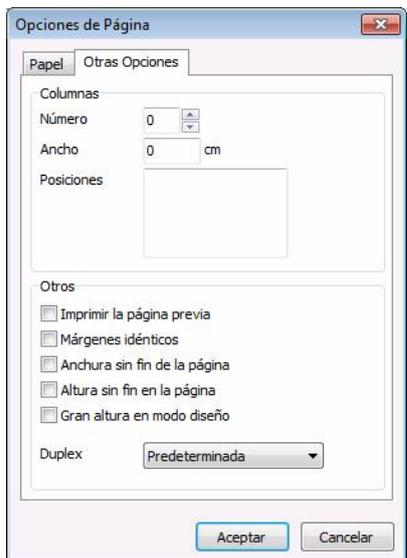
Los parámetros de página se asignan a través del menú "Archivo|Opciones de Página", pulsando en el botón de la barra  o haciendo doble-clic en el espacio en blanco de la página. El diálogo tiene dos solapas:



En este diálogo, se puede seleccionar el tamaño del papel de impresión, su orientación, así como establecer los márgenes.

Los cambios son automáticamente mostrados en el diseñador para trabajar de la forma más realista posible.

En la sección "Origen del papel" se puede seleccionar una bandeja de la impresora para imprimir la primera página y otra para el resto de las páginas del informe.



En la segunda parte, se puede establecer el número de columnas, su anchura y separación para informes de varias columnas, como por ejemplo etiquetas.

Los cambios son automáticamente mostrados en el diseñador para trabajar de la forma más realista posible.

La opción "Imprimir la página previa" permite imprimir páginas, comenzando en el espacio en blanco de la página de diseño anterior. Esta opción se puede utilizar en caso de que un diseño consta de varias páginas.

La opción "Márgenes idénticos" cambia los márgenes derecho e izquierdo de la página, incluso para páginas durante una vista previa o imprimir un informe.

La "Anchura y altura sin fin de la página" hace que el papel a imprimir no tenga límite. Es útil para imprimir en impresoras de tickets donde el papel es un rollo continuo o para generar páginas HTML

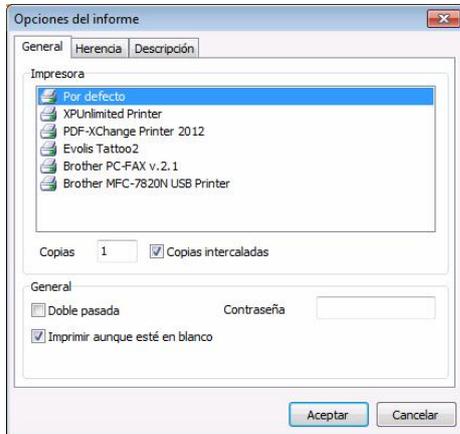
La opción "Gran altura en modo diseño" no tener limite de altura en el diseñador. Esta característica puede ser útil si se usan muchas bandas en la página. Esto sólo afecta a la altura de la página de diseño, el destino final en papel es del tamaño del papel seleccionado.

Por último la opción duplex sirve para la impresión en impresoras de doble cara.

## Opciones del Informe

Puede modificar los parámetros del informe Desde "Informe|Opciones...".

Este diálogo tiene tres solapas:



La opción "Impresora predeterminada" envía a la impresora por defecto o a la que se seleccione en el momento de la impresión, pero se puede "atar" un informe a una impresora concreta de las instaladas en el sistema. Esto significa que la impresión de un informe se realizará por esa impresora. Esto podría ser útil por ejemplo para enviar la impresión de talones de pago a una impresora matricial por ejemplo

También puede establecer el número de copias a imprimir por defecto para este informe y especificar, si es necesario para llevar a cabo la intercalación. Estos valores aparecerán después en el diálogo de impresión y podrán ser modificados.

Si se selecciona "Doble pasada", la generación del informe se realizará en dos pasos. Durante la primera pasada, se forma un informe, y se divide en páginas, pero los cálculos de los resultados no se efectúan ya que se realizan en un segundo paso.

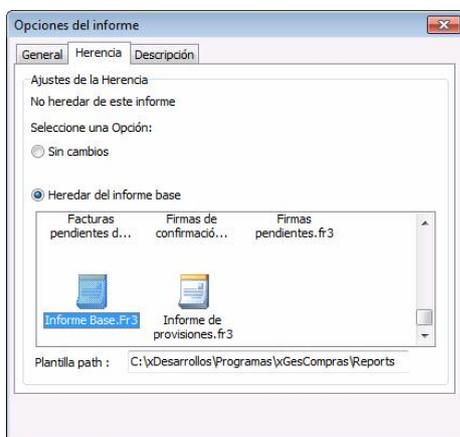
A menudo, esta opción se utiliza en los casos en que en un informe hay una necesidad de que el número total de páginas en el mismo, es decir, la información de la "Página 1 de 15".

Se calcula el número total de páginas durante la primera pasada y está disponible través de la variable del sistema "TotalPages". El error más frecuente es un intento de utilizar esta variable en un informe de un solo paso, en este caso devuelve "0".

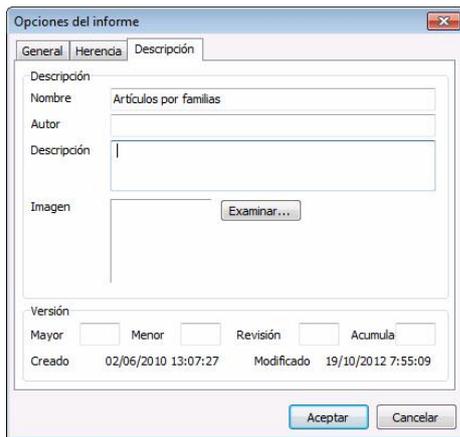
Otro uso es para visualizar una suma en el encabezado de grupo, que por lo general se calcula y se muestra en el pie de grupo.

La "Imprimir aunque esté en blanco" permite la construcción de un informe, que no contiene líneas. Si esta opción está desactivada, no se pueden construir informes en blanco.

El campo "Contraseña" permite la creación de una contraseña que el usuario deberá introducir antes de la apertura de un informe.



En este diálogo, se puede hacer que un informe herede visualmente de un informe base.



Por último se pueden establecer una serie de datos sobre el informe que son meramente informativos.

## Tema 7. Creando nuestro primer informe columnar.

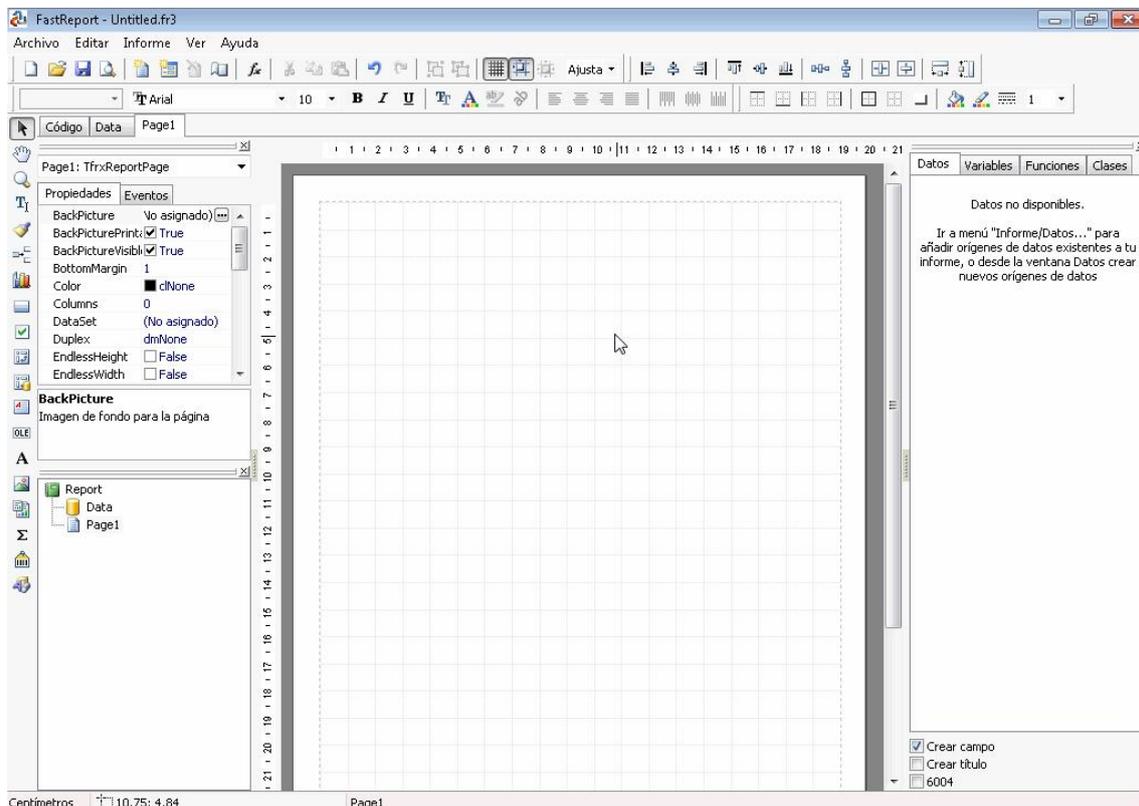
Aunque veremos que FastReport puede hacer muchos tipos de informes, los que más habitualmente se utilizan en las aplicaciones son los informes columnares.

Veamos como podemos crear uno sencillo. Partiremos desde un programa cualquiera en el que hay una tabla de clientes de tipo DBF, deberemos de abrir el archivo, instanciar un objeto FastReport y enviar la tabla a este. Después pondremos las columnas a imprimir y listo.

Vamos paso a paso.

```
Use .\Datos\Clientes.Dbf
oFr:=frReportManager():New( )
oFr:SetWorkArea(Alias(), Select()) //Se debe de indicar el alias y el área del archivo dbf
oFr:DesignReport()
oFr:DestroyFR()
```

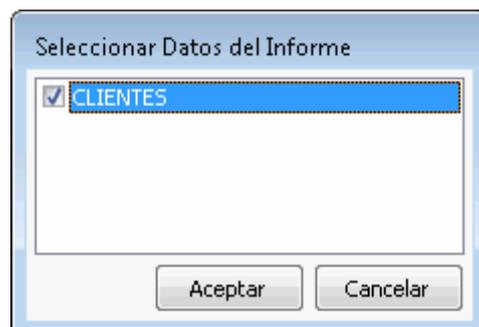
Con estas líneas aparece en pantalla el editor visual vacío:



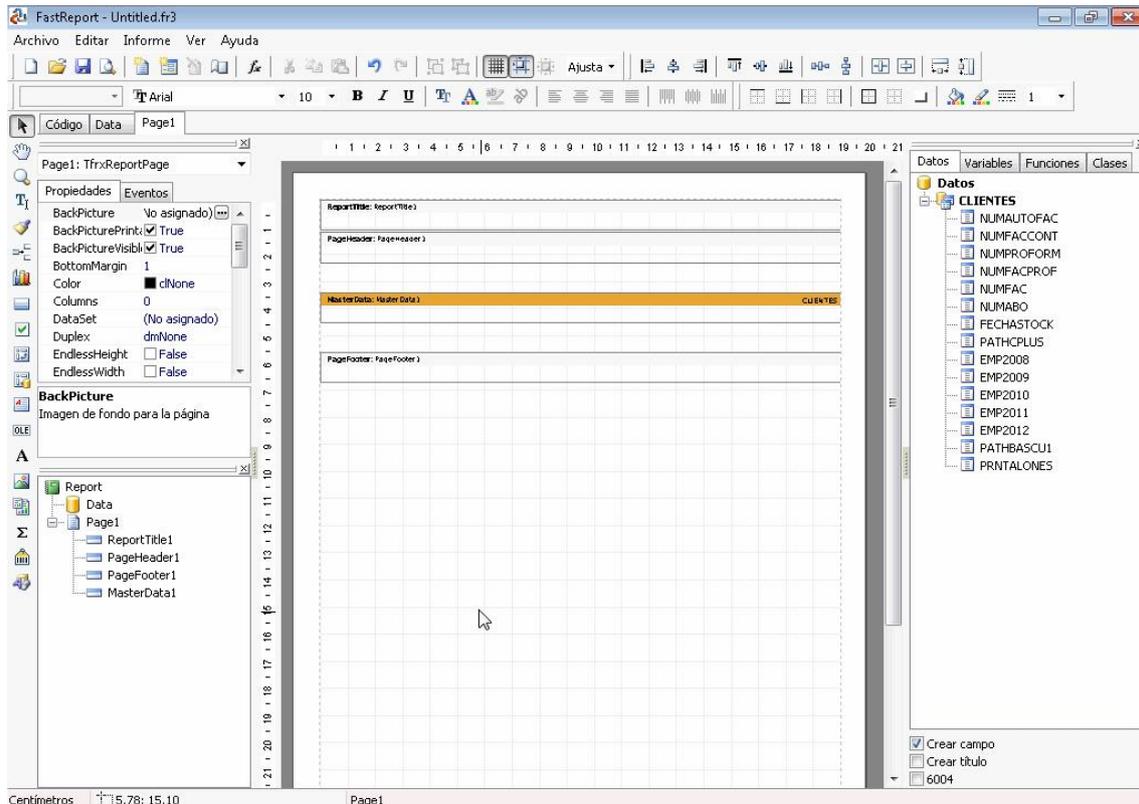
Como se ve a la derecha se indica que Datos no disponibles, entonces debemos de ir al menú Informes-Datos donde se muestran las tablas que ha recibido el editor en este caso solo una y la seleccionamos. Si hubiera recibido varias podríamos seleccionar todas las que quisiéramos que aporten algo al informe.

Hecho esto aparecen automáticamente todos los campos de la tabla o tablas seleccionadas a la derecha.

Ahora es momento de poner unas bandas, por ejemplo



una banda de título, una de cabecera de página, una de pie de página y finalmente la mas importante banda de datos maestros que al ser creada nos preguntará con que tabla queremos que este ligada.



Ahora podemos arrastrar los campos que queremos a la banda MasterData teniendo cuidado de que queden dentro, hay que insistir en esto puesto que si por descuido se salen un poco de la banda entonces ya pertenecen al fondo del diseño y saldrán justo en esa posición. Si hace falta haremos la banda más alta.

NOTA: Si queremos cambiar el fuente y tamaño del objeto podemos hacerlo con la barra de herramientas habitual pero cada nuevo objeto insertado tomará las propiedades del último objeto modificado.



Una vez insertados los campos pulsamos F9 para ver el resultado y el informe muestra los datos enviados desde nuestro programa.

Si ahora añadimos un texto con un título, los nombres de los campos como textos en la banda de cabecera y el n° de página en el pie desde las variables del sistema tendremos terminado el informe.

Direcciones de clientes			Dirección			Población			C.P.		
Dirección	Población	C. P.	Dirección	Población	C.P.	Dirección	Población	C.P.	Dirección	Población	C.P.
VEGA DE UGARTE, SW	BASABURI	45910	KAREAGA, Nº15	BARACALDO	45903	RIBERA DE LA RIA, SW	SESTAO	45910	RIBERA DE LA RIA, SW	SESTAO	45910
PUERTO DE ARETIO, 16	MALLASA	45269	MAS D E LA RIVERA, SW	SANTASDORNI PANKOIA	45770	OSKIZURI, SW	ERMITA	45260	MAS D E LA RIVERA, SW	SANTASDORNI PANKOIA	45770
Bº BALLOTTI, SW	VALLE DE TRAPAGA	45510	MEMOIFETE TALDEA, 24	TRAPAGA	45510	Bº AMARREKAL, 13A	TRAPAGA	45510	MEMOIFETE TALDEA, 24	TRAPAGA	45510
PASEO DE LOS FUEROS, 14	AZPEITIA	20730	PO LO INO. EL CAMPILLO, 4	GALLARTA	45503	PO LO INO. EL CAMPILLO, 4	GALLARTA	45503	Bº AMARREKAL, 13A	TRAPAGA	45510
AVDA. BILBAO, 10	ORTUELLA	45330	Bº KORTERRERA, 37	AMOREBIETA	45340	PAULINO MENDOZA, 1-6ºA	IZURZA	45213	Bº KORTERRERA, 37	AMOREBIETA	45340
PASEO DE LA MADONALENA, 22	SALMASEDA	45510	CLAS ERAS, Nº5	MORALZARZAL	25411	CALATSA SUBILLA, 30º	NAVARRA	31751	CLAS ERAS, Nº5	MORALZARZAL	25411
EL JUNCAL, SW (UGARTE)	VALLE DE TRAPAGA	45510	ERANDIO-OOINDA	ERANDIO-OOINDA	45560	ERANDIO-OOINDA	ERANDIO-OOINDA	45560	CALATSA SUBILLA, 30º	NAVARRA	31751
RETUERTO	BARACALDO	45903	SANTANDER	SANTANDER	35011	SANTANDER	SANTANDER	35011	ERANDIO-OOINDA	ERANDIO-OOINDA	45560
BF ARTZ, 24 -APTDO. CORREOS Nº3-	ZUMARRAGA	20100	BOLUETA	BOLUETA	45204	BOLUETA	BOLUETA	45204	SANTANDER	SANTANDER	35011
AFARTADO 51	EUBA	45250	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510	BOLUETA	BOLUETA	45204
OTRA UGARTE A TRAPAGA, SW	VALLE DE TRAPAGA	45510	SU RO ER A-BARACALDO	BARACALDO	45903	SU RO ER A-BARACALDO	BARACALDO	45903	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510
C/VEGA DE LA TAPIA, SW	ERANDIO-OOINDA	45560	ERANDIO-OOINDA	ERANDIO-OOINDA	45560	ERANDIO-OOINDA	ERANDIO-OOINDA	45560	SU RO ER A-BARACALDO	BARACALDO	45903
C/TRA. OOINDA A ERANDIO Nº0, SW/LA	ERANDIO-OOINDA	45560	SANTANDER	SANTANDER	35011	SANTANDER	SANTANDER	35011	ERANDIO-OOINDA	ERANDIO-OOINDA	45560
NU EVA MONTAÑA, SW	SANTANDER	35011	BOLUETA	BOLUETA	45204	BOLUETA	BOLUETA	45204	SANTANDER	SANTANDER	35011
PLAZA SANTANA, 10	BOLUETA	45204	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510	BOLUETA	BOLUETA	45204
OTRA GALINDO A UGARTE, SW	VALLE DE TRAPAGA	45510	BARACALDO	BARACALDO	45903	BARACALDO	BARACALDO	45903	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510
KAREAGA, Nº15	BARACALDO	45903	SAQUINTO	SAQUINTO	45500	SAQUINTO	SAQUINTO	45500	BARACALDO	BARACALDO	45903
C/ BEMBAIRO, 2-1º	SAQUINTO	45500	NUÑIKA	NUÑIKA	45362	NUÑIKA	NUÑIKA	45362	SAQUINTO	SAQUINTO	45500
Bº VISTA ALBORA Nº100	NUÑIKA	45362	SONDICA	SONDICA	45940	SONDICA	SONDICA	45940	NUÑIKA	NUÑIKA	45362
PORTUBIEA, Nº12 Nº0, SY 7	SONDICA	45940	AMURRIO	AMURRIO	01470	AMURRIO	AMURRIO	01470	SONDICA	SONDICA	45940
BARRIO SAGARRIBAL, Nº2	AMURRIO	01470	EUBA	EUBA	20600	EUBA	EUBA	20600	AMURRIO	AMURRIO	01470
PF URQUIZU 13-1- PLANTA (APTDO. C. 14)	EUBA	20600	BILBAO	BILBAO	45102	BILBAO	BILBAO	45102	AMURRIO	AMURRIO	01470
LUCHARA, Nº6- 3ºB	BILBAO	45102	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510	EUBA	EUBA	20600
POL. IND. DE UGARTE, SW	VALLE DE TRAPAGA	45510	TRAPAGARAN	TRAPAGARAN	45510	TRAPAGARAN	TRAPAGARAN	45510	BILBAO	BILBAO	45102
NU EVA MONTAÑA, SW	SANTANDER	35011	SESTAO	SESTAO	45910	SESTAO	SESTAO	45910	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510
Bº EL JUNCAL, SW	TRAPAGARAN	45510	EL JUNCAL SW, EDJUNTES, PPTO 16-3º	TRAPAGARAN	45510	EL JUNCAL SW, EDJUNTES, PPTO 16-3º	TRAPAGARAN	45510	SANTANDER	SANTANDER	35011
CHAVARRI, 6	SESTAO	45910	PINO ORANA DA, AVDA BILBAO, SW	ORTUELLA	45300	PINO ORANA DA, AVDA BILBAO, SW	ORTUELLA	45300	TRAPAGARAN	TRAPAGARAN	45510
OTRA SALO EDILLO, SW	TRAPAGARAN	45510	AVDA. CATALUÑA, 242	ZARAGOZA	50114	AVDA. CATALUÑA, 242	ZARAGOZA	50114	SESTAO	SESTAO	45910
C/ BILBAO Nº5-6ºB	GALDAKANO	45960	C/PORTO COLOM Nº5-LOCAL Nº2	ALDOCOA	28524	C/PORTO COLOM Nº5-LOCAL Nº2	ALDOCOA	28524	TRAPAGARAN	TRAPAGARAN	45510
LETIZIAR, 37	BARACALDO	45901	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510	VALLE DE TRAPAGA	VALLE DE TRAPAGA	45510	GALDAKANO	GALDAKANO	45960
105, RUE BUCAU	BO RDEAUX	33000	BASABURI	BASABURI	45910	BASABURI	BASABURI	45910	BARACALDO	BARACALDO	45901
Bº UGARTE, SW PAB. BLOOMING	BASABURI	45910	BARRO MALOC, 15	AULESTIA	45330	BARRO MALOC, 15	AULESTIA	45330	BO RDEAUX	BO RDEAUX	33000
RIBERA DE ERANDIO, 7	ERANDIO	45960	POL. IND. INDUSTRIAL DE BARATXO	AMURRIO	01470	POL. IND. INDUSTRIAL DE BARATXO	AMURRIO	01470	BASABURI	BASABURI	45910
C/TRA. DURANA OO-EL RRO-VATXO Nº0	APATAMONASTERIO	45291	C/AMINO ORIBO NºM 1500	PARLA	28500	C/AMINO ORIBO NºM 1500	PARLA	28500	ERANDIO	ERANDIO	45960
POLIGONO INDUSTRIAL ZUBIETA, SW	AMOREBIETA	45340	C/IBARRA Nº6	BERGARA	20510	C/IBARRA Nº6	BERGARA	20510	APATAMONASTERIO	APATAMONASTERIO	45291
CHEMIN DE QUITEONDE, B.P. 1	AMOREBIETA	45340	MONTESQUIZA Nº30, SP 120	MADRID	28010	MONTESQUIZA Nº30, SP 120	MADRID	28010	AMOREBIETA	AMOREBIETA	45340
EDIFICIO FUENTES, C/TRA. JUNCAL, SW	UGARTE	45510	PINO LAPATZA, C/ARASTOI Nº2	BASABURI	45910	PINO LAPATZA, C/ARASTOI Nº2	BASABURI	45910	AMOREBIETA	AMOREBIETA	45340
Bº LA ESTACION, SW	NU EVA MONTAÑA	35011	C/ ARDENA L CENEBROS 4, 3º B	OSLU DA	28200	C/ ARDENA L CENEBROS 4, 3º B	OSLU DA	28200	UGARTE	UGARTE	45510
Bº LA MINA, SW	GAJANO	35752	C/ PLAYA-BARRI, Nº5	LUCHARA-ERANDIO	45960	C/ PLAYA-BARRI, Nº5	LUCHARA-ERANDIO	45960	NU EVA MONTAÑA	NU EVA MONTAÑA	35011
P. LAURERA 3ª FASE-27	TRAPAGARAN	45510	OTRAJUA D RIBU, SW	OLABERRIA	20212	OTRAJUA D RIBU, SW	OLABERRIA	20212	GAJANO	GAJANO	35752
			ARENE BIDEA Nº20	BERANCO	45440	ARENE BIDEA Nº20	BERANCO	45440	TRAPAGARAN	TRAPAGARAN	45510

Ahora solo queda grabarlo dándole un nombre como por ejemplo “Direcciones de Clientes.Fr3” y a partir de este momento podremos llamarlo desde el programa simplemente con estas líneas:

```

Use .\Datos\Clientes.Dbf
oFr:=frReportManager():New( )
oFr:SetWorkArea(Alias(), Select())
oFr:LoadFromFile(“Direcciones de Clientes.Fr3”)
oFr:PrepareReport()
oFr:ShowReport() //Lo muestra en el previsualizador
oFr:DestroyFR()
    
```

Hasta aquí no es complicado, pero se puede complicar algo más si queremos enviar 10 tablas a FR, si queremos que sean tDatasets de Xailer que provienen de Sql o si queremos imprimir tMemDataSets con tablas de memoria, etc. aunque más adelante veremos como podemos unificar y simplificar todo esto.

Veamos ahora más detalladamente el uso de estas instrucciones básicas de FastReport que hemos utilizado y alguna nueva.

## Tema 8. Creando un informe de etiquetas.

Pasamos a crear un nuevo informe y ver algunas características más de FR.

En este caso comenzamos igual que antes enviando a FastReport una tabla de datos de clientes llamada Etiquetas:

```
Use .\Datos\Etiquetas.Dbf
oFr:=frReportManager():New( )
oFr:SetWorkArea("Etiquetas", 1) //Se debe de indicar el alias y el área del archivo dbf
oFr:DesignReport()
oFr:DestroyFR()
```

y una vez aparezca el diseñador añadimos una banda de datos maestros (MasteDataBand), como las etiquetas no necesitan ni título ni cabeceras ni pie, es suficiente.

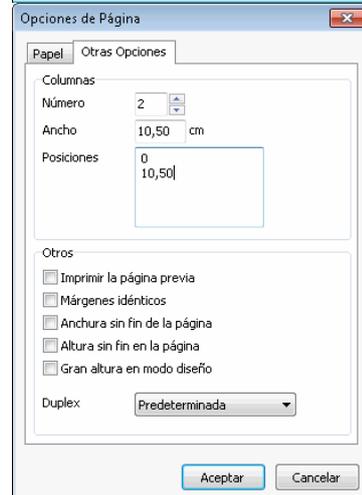
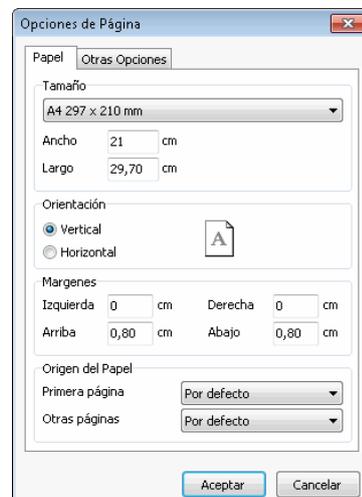
Las etiquetas normalmente vienen en varias columnas, dos por ejemplo. Para poder hacer esto hacemos doble clic sobre el fondo de la página y acceder a las opciones de página.

Debemos de establecer los márgenes de la hoja a los que exactamente tengan las etiquetas a veces ni siquiera tienen márgenes y se dejan a cero.

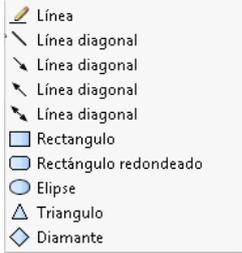
Pasando ahora a la segunda solapa indicamos un número de columnas, dos en este caso y se ve que FastReport reparte el espacio restante entre los márgenes izquierdo y derecho entre dos, anchura de cada una 10,50 cm.

El caso más rebuscado de unas etiquetas es que tengan un canal central de separación. Entonces deberemos modificar a mano la anchura y la posición de inicio de la segunda columna.

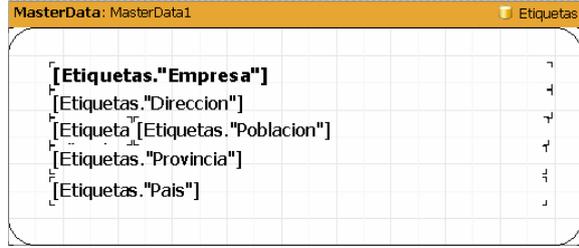
El resultado sería el siguiente:



Ahora bastará con arrastrar los campos que queramos se impriman en la etiqueta y si queremos añadir un pequeño efecto visual.



Desde el botón dibujar  obtenemos el menú de dibujos y seleccionamos el dibujo de rectángulo redondeado y lo dibujamos alrededor de la etiqueta.

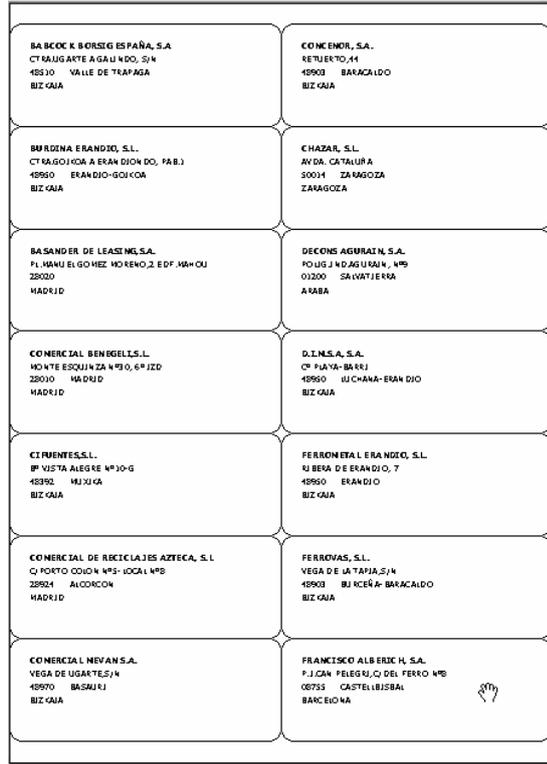


De esta manera se mostrará en la vista previa como si fueran los bordes de las etiquetas reales.

Y por último seleccionamos este rectángulo y nos fijamos a la izquierda entre sus propiedades en la propiedad llamada Printable y la ponemos a False.

En FastReport se puede tomar cualquier objeto y decirle si se va a ver en la vista previa (propiedad Visible) y si va a ser impreso o no (propiedad Printable).

Así podemos presentar en una vista previa una factura completa con logotipo, datos de la empresa, líneas y cuadros, aunque luego no se impriman ya que la empresa dispone de papel ya preimpreso y solo se imprimen los textos.



Este sistema de varias columnas también vale para hacer informes columnares que por sus pocos datos puedan caber en 2 columnas.

Un detalle, en este caso en la banda de cabecera de página habrá que escribir los nombres de las columnas dos veces.

Hay que fijarse en otro detalle importante, las columnas se conforman escribiendo los registros de la tabla de arriba abajo, terminando siempre la primera columna para empezar con la segunda.

**Prefijos provinciales españoles**

Nota: Aunque ya no se utilizan como tales para llamadas interprovinciales, siguen figurando como parte del número de teléfono de los abonados.

Provincia	Prefijo	Provincia	Prefijo
Álava	945	Navarra	948 - 948
Albacete	967	Ourense	988
Alicante	96	Palencia	979
Almería	950	Palmas, Las	929 - 929
Asturias	98	Pontevedra	986 - 986
Ávila	920	Rioja, La	941
Badajoz	924 - 924	Salamanca	923
Baleares	971 - 971	Segovia	921
Barcelona	93	Sevilla	95
Burgos	947	Soñá	975
Cáceres	927	Sta. C. de Tenerife	922 - 922
Cádiz	956 - 956	Taragona	977 - 977
Cantabria	942	Teruel	978
Castellón	964	Toledo	925
Ceuta	959 - 959	Valencia	96
Ciudad Real	926	Valladolid	983
Córdoba	957	Vizcaya	94
Coruña, A	981 - 981	Zamora	980
Cuenca	969	Zaragoza	976 - 976
Girona	972 - 972		
Granada	958 - 958		
Guadalajara	949		
Guzmán	943 - 943		
Huelva	959		
Huesca	974		
Jáén	953		

## Tema 9. El objeto texto o memo.

Este es el objeto más útil y versátil de cuantos tenemos en FR. Se usa para insertar en un diseño tanto un texto escrito, como un campo de una tabla, como una fórmula, cálculo, total, variable, función de FastReport o de xHb, textos con tags de Html, links Html que funcionan al pulsar en ellos en la vista previa, campos memo o Blob, o expresiones con una mezcla de todo ello.

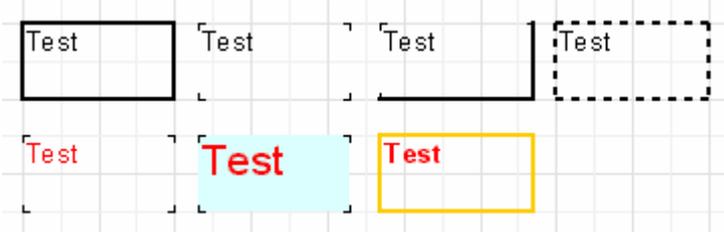
Para insertar un campo de texto en el diseño se puede hacer de múltiples maneras

**T** Permite dibujar un área de un tamaño que luego incluirá el texto.

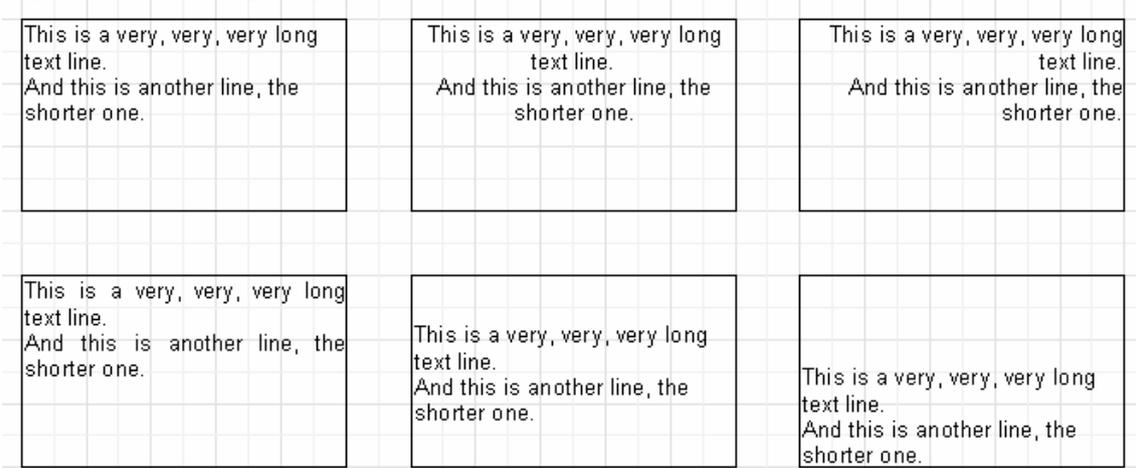
**A** Introduce un campo de texto pero con esta opción no podemos darle un tamaño directo sino que toma un tamaño por defecto, aunque luego se podrá cambiar su tamaño.

También se crea un campo de texto si desde el panel derecho se arrastra un campo de una tabla, una variable o una función.

Además los objetos texto tienen un sin fin de propiedades y posibilidades como el uso de fuentes y tamaños diversos, colores de texto, de fondo, de marco, estilos, etc.



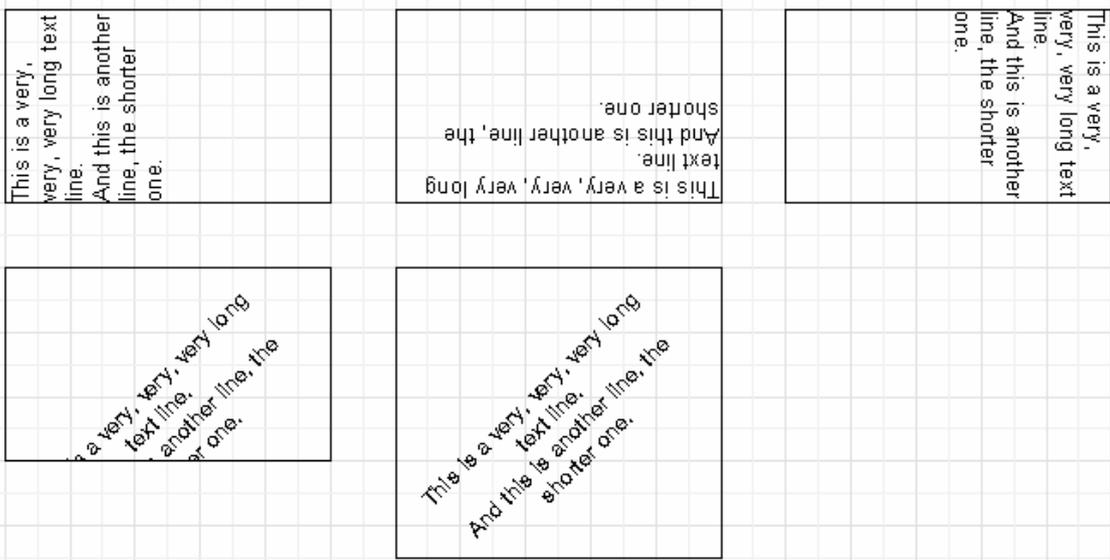
Ajustes de todo tipo incluyendo “Ajuste de palabras”, es decir si se estrecha el marco contenedor se reajustan las líneas sin partir palabras:



Aunque el “Ajuste de palabras” puede ser desactivado desde el menú contextual del objeto.

Admite rotación en intervalos de 45° pulsando sobre  y seleccionando

0°  
45°  
90°  
180°  
270°



Si se quiere rotar unos grados específicos no múltiplos de 45° se puede indicar con la propiedad Rotation del objeto cualquier valor de 0 a 360.

Además de poder modificar un buen grupo más de características visuales como un brush de fondo, el espaciado entre caracteres y entre líneas, etc. desde las propiedades del objeto, mostradas en el panel izquierdo.

### Uso de Tags HTML

Se pueden usar Tags simples de HTML, estos han de estar situados dentro del texto del objeto y están desactivados por defecto. Para activarlos hay que activarlos desde el menú contextual con la opción “Permitir etiquetas HTML” o desde la propiedad “AllowHTMLTags” como True.

Los tag soportados no son muchos pero suficientes para el formateo de un texto simple:

<b> - negrita - </b>

<i> - cursiva - </i>

<u> - subrayado - </u>

<sub> - subscript - </sub>

<sup> - superscript - </sup>

<font color> - color del fuente - </ font>

<nowrap> - desactiva el “WordWrapping” o “Ajuste de palabras” - </ nowrap>

Por ejemplo el contenido de un campo de tipo texto podría ser:

texto <b>texto en negrita</b> <i>texto en cursiva</i> <b><i>negrita y cursiva</i></b></i>

E = mc<sup>2</sup>

A<sub>1</sub> = B<sup>2</sup>

este es un texto normal, <font color=red>y este es uno en color rojo</font>

este es un texto normal, <font color="#FF8030">y este es uno en color naranja</font>

Con lo cual FastReport mostraría esto

texto **texto en negrita** *texto en cursiva* ***negrita y cursiva***  
*cursiva*  
 E = mc<sup>2</sup>  
 A<sub>1</sub> = B<sup>2</sup>  
 este es un texto normal, **y este es uno en color rojo**  
 este es un texto normal, **y este es uno en color naranja**

Los Tag Html se pueden usar también para resaltar campos de datos de una tabla, incluso con condiciones como que si un número es negativo lo ponga en rojo.

## Mostrando expresiones en un objeto texto de FR

Una de las características más importantes de este versátil objeto es la posibilidad de mostrar expresiones, es decir textos variables como un campo de una tabla o una fórmula.

Las expresiones además pueden ser mezcladas con texto normal para construir textos más ricos como:

Buenos días, hoy es [DATE].

Que en la vista previa o impresión se convertiría en la fecha en curso:

Buenos días, hoy es 24-02-2010

Nótese que los textos normales se escriben sin más pero todo aquello que vaya entre los símbolos [ ] será algo que FastReport deberá de evaluar en ejecución y bien podría ser un campo de una tabla [Etiquetas."Direccion"] o una función del sistema [DATE] o una variable de FastReport o de nuestro programa u operaciones de todo tipo [1+2\*(3+4)]

Pero si en un texto normal queremos incluir los símbolos [ ] FastReport intentará evaluar su contenido pudiendo dar error, entonces para ese objeto texto se debe de desactivar la propiedad "Allow Expressions" y si además queremos que un texto pueda incluir [ ] y además una expresión [DATE] si desactivamos la opción saldrán los [ ] del texto pero no se evaluará [DATE] que saldrá escrito tal cual, entonces podemos obligar a que se evalúe cambiando [DATE] por <DATE>.

## Campos de tablas de datos

Los campos de tablas que se arrastran desde el panel de la derecha también pasan a ser textos con una expresión dentro [Etiquetas."Direccion"] aunque si se hace doble clic sobre ellos se puede entrar a modificarlos y podemos añadir texto C/[Etiquetas."Direccion"] o nuevos campos mezclados [Etiquetas."CodPos"] - [Etiquetas."Poblacion"]

Si el campo que se arrastra es un campo memo a una banda de datos, lógicamente no cabrá su texto entero dentro del marco proporcionado y por lo tanto será truncado, se puede hacer mas grande pero nunca sabremos que tamaño exacto darle para que todo quepa y además si lo hacemos mas alto a la fuerza todas las líneas de la banda tomarán esa altura. Para hacer que FastReport calcule automáticamente la altura de cada memo y ajuste cada línea hay que hacerlo de la siguiente manera

Situamos el campo memo y lo dejamos con la altura estándar de una línea, después vamos a su propiedad StretchMode y lo ajustamos a smMaxHeight y para terminar seleccionamos la propia banda e indicamos que su propiedad Stretched sea True, de esta manera el memo crecerá lo que necesite en cada línea. Esto se puede usar para cualquier otro campo de texto de tamaño considerable.

### Caracteres especiales como corchetes [ ] o < >

Como se ha podido observar los campos de las tablas o las variables son encerrados por FastReport entre corchetes, cuando hay un corchete el programa sabe que ha de resolver lo que tiene dentro para dar un resultado, pero lo que esté fuera de estos corchetes es escrito tal cual como un texto.

**Nombre:** [Clientes."Nombre"]      **Resultará en:**      **Nombre:** José García

A veces sucede que al tratarse de una expresión compleja los corchetes no bastan

**[IF(Length(<DatosEmpresa."telefono1">)>0,"Tel.:'+<DatosEmpresa."telefono1">,"")]**

Por que encierran la expresión global a resolver, Entonces se usan los < > para indicar los campos que han de resolverse dentro de la expresión global y los textos irían encerrados entre comillas simples ' '

Si queremos hacer que nuestro objeto imprima corchetes y no se les considere como expresiones, por ejemplo, si tenemos que mostrar el texto a[1] := 10

FastReport considerará [1] una expresión a evaluar y mostrará si puede su resultado a1 := 10 y esto no es lo que queremos La primera manera de evitar esto es desactivando la propiedad "AllowExpressions"

en la lista de propiedades del objeto memo, pero esto hará que no se evalúe ninguna expresión dentro del memo y si queremos mezclar el texto anterior con una expresión `a[1] := [TotalPages]` no evaluará el total de páginas del documento.

Para esto se puede activar la propiedad "AllowExpressions" y cambiar la propiedad "ExpressionDelimiters" indicando que en lugar de usar como delimitador el estándar [,] use otro como por ejemplo <> (la coma divide el operador de inicio el de cierre) y entonces indicar la expresión como `a[1] := <TotalPages>` con lo cual conseguiremos que se imprima `a[1] :=12` en el caso de ser 12 las páginas totales.

El único símbolo que no se puede usar como delimitador es el porcentaje %, %

## Formateando un texto y resaltándolo automáticamente.

Cuando se edita un texto se muestra dentro de una ventana de edición como esta:

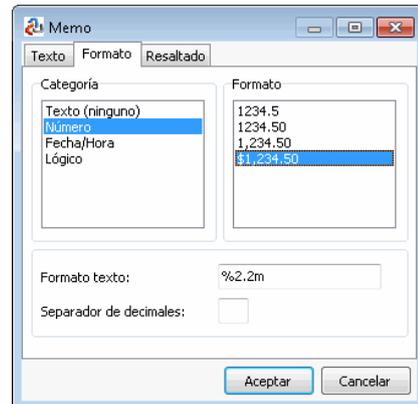


En la parte superior dispone de tres solapas siendo la primera la de edición del texto y las siguientes de formato y resaltado.

En la pestaña de formato se puede ver que se puede aplicar de una forma rápida un formato de tipo Numérico, Fecha/Hora o Lógico.

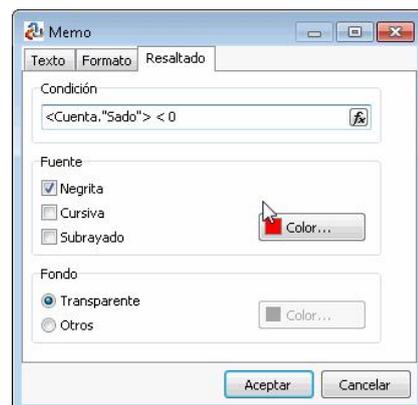
El primer formato Texto es como no asignar nada el texto será mostrado tal cual es.

Seleccionando numérico se muestra una lista de cuatro posibles combinaciones, General, decimales fijos, millares con decimales fijos y moneda. Seleccionando cualquiera de ellos se muestra debajo el Picture que va a ser aplicado, Picture que se encuentra en formato Delphi, no hay que olvidar que es el lenguaje en que está hecho FR, y que podemos modificar si lo deseamos a mano. Ojo en el formato moneda ya se aplica directamente la que esté por defecto en el sistema y el separador de decimales también, así que es mejor no indicar nada en ellos.



El resaltado permite añadir al texto las opciones de negrita, cursiva, subrayado, color de texto y de fondo, también se pueden seleccionar estas opciones desde la barra de herramientas, pero aquí además se puede establecer una condición que ha de cumplirse para que se aplique este resaltado.

Por lo tanto un texto puede tener dos tonalidades la que se aplique desde la barra de herramientas que será la normal y la que se escriba aquí que será la especial, por ejemplo para resaltar números negativos en rojo.



## El editor de Macroexpresiones.

Dentro del editor de textos, tenemos también un generador y editor de macroexpresiones que facilita enormemente la construcción de funciones, formulas, etc. utilizando nuestros campos y variables.

Si nos fijamos en la parte superior de la ventana dentro de la primera solapa vemos cuatro botones



El primero es el generador de macroexpresiones, pulsando sobre el se muestra un diálogo nuevo con múltiples solapas y opciones.

En la primera solapa llamada datos se muestran todas las tablas enviadas a FastReport desde el programa y sus campos. Haciendo doble clic en uno de ellos lo pasa a la zona inferior que es donde se irá construyendo la expresión, pero no es lo primero que haremos, ya que pretendemos hacer algo más completo.

Si pasamos a la segunda solapa veremos también las variables de FastReport y las que hayamos enviado desde el programa pero pasemos a la tercera solapa que es la más interesante, la llamada funciones.

Las funciones que se muestran son las que incorpora FastReport y las que hayamos añadido desde el programa y vienen agrupadas por temas, las hay de tipo matemático, de fecha y hora, de conversión entre formatos, de texto, etc.

Cada una de ellas muestra al seleccionarla una pequeña ayuda en la parte media de la ventana para indicarnos primero para que sirve y segundo para que veamos que parámetros necesita.

Por ejemplo DayOf(Date) devuelve el día de una fecha así que si tenemos un campo fecha en nuestros datos y queremos que FastReport imprima solo el día deberemos de hacer doble clic sobre DayOf con lo que en la parte inferior se mostrará la función vacía DayOf() y segundo debemos de introducir el cursor dentro del paréntesis para ir a la pestaña de datos, buscar el campo fecha y hacer doble clic en el para que se inserte dentro de los paréntesis, con lo que se mostraría la expresión completa: DayOf(<SqlLisMov."fecha">)

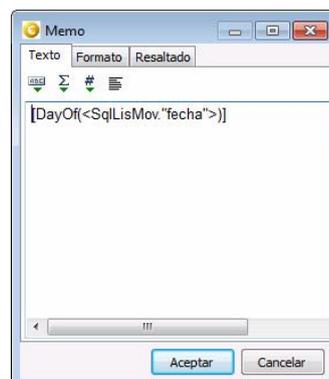
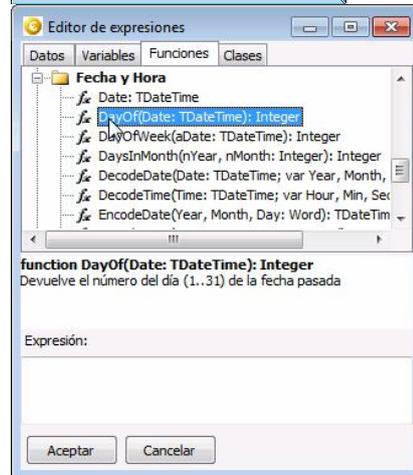
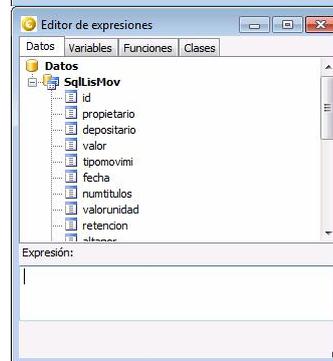
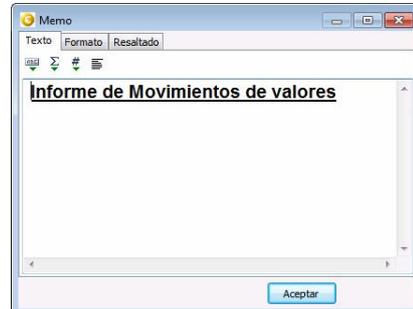
Ahora basta con aceptar para que la expresión sea pasada al editor de textos y aceptar de nuevo para que pase al editor.

Este es un ejemplo sencillo de lo que se puede hacer con el generador de expresiones pero se puede llegar a un nivel de sofisticación increíble usando funciones de todo tipo, como por ejemplo hacer que mediante un IIF se imprima un campo u otro, etc.

Por ejemplo

```
[IIF(YearOf(<SqlLisMov."fecha">)<YearOf(Date),'Años pasados',<SqlLisMov."fecha">)]
```

Si el año de la fecha de movimiento es menor al año actual escribe el texto 'Años pasados' y si no pon la fecha de



movimiento

Otro ejemplo

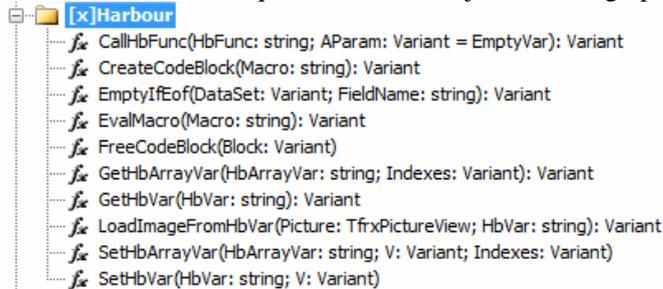
```
[IIF(<SqlMovimientos."Tipo">='A','Nuevo movimiento', IIF(<SqlMovimientos."Tipo">='M',  
'Modificación', 'Baja'))]
```

**Nota:** Como se ha mencionado los corchetes [ ] se usan para delimitar una expresión, que debe de ser evaluada en tiempo de ejecución para dar un resultado, pero si se ha de utilizar dentro de una expresión un campo a su vez ha de ser delimitado por los símbolos < >

## Tema 10. Ampliar FastReport con nuestras propias funciones

Las funciones que encontramos en FastReport nos suenan mucho a las que tenemos en cualquier sistema de programación, pero al ser este programa Delphi debemos de tener alguna consideración, por ejemplo las comparaciones con campos lógicos no pueden expresarse como .T. o .F. sino como True o False escrito entero.

De todas las funciones que tiene vamos a fijarnos en un grupo muy concreto llamado [x]Harbour.



Estas funciones nos permiten una comunicación bidireccional con nuestro programa ya que al igual que desde nuestro programa hemos llamado a FR, también FastReport puede llamar a nuestro programa. Con la primera función CallHbFunc podemos llamar a cualquier función de [x]Hb o de nuestro código fuente, pasarle los parámetros necesarios desde FastReport y aquello que nos devuelva sea un texto, fecha, número o lógico sea escrito en el informe o evaluado por FR.

Por ejemplo: [CallHbFunc('Tipomovimi', [<SqlLisMov."tipomovimi">])]

Llama a una función de mi programa Function TipoMovimi(cTipo) que devuelve un texto a FastReport que es el que va a ser impreso

```
//Devuelve el tipo de movimiento a partir de una clave de un carácter
Function Tipomovimi(cTipo)
Local cTipos:="ARMDGJSTVCPN"
Local aTipos:={"Adquisición","Reembolso","Actualización","Dividendo","Gastos de custodia";
"Canje derechos","Split de acciones","Traspaso entre entidades","Venta derechos"; "Compra
derechos","Prima asistencia","Devolución Capital"}
Return If(At(cTipo,cTipos)>0,aTipos[At(cTipo,cTipos)],"Error de tipo")
```

También podemos pedir al programa el valor de una variable a pesar de no haberla enviado previamente a FR, es algo tan sencillo como usar la función:

```
GetHbVar('Nombrevariable')
```

lo cual nos devuelve el contenido de cualquier variable salvo los arrays, aunque si deseamos consultar un valor que esté en un array podemos utilizar la función

```
GetHbArrayVar('NombreArray',nItem)
```

Igualmente se pueden alimentar variables del programa y valores de tablas desde FastReport con las funciones inversas

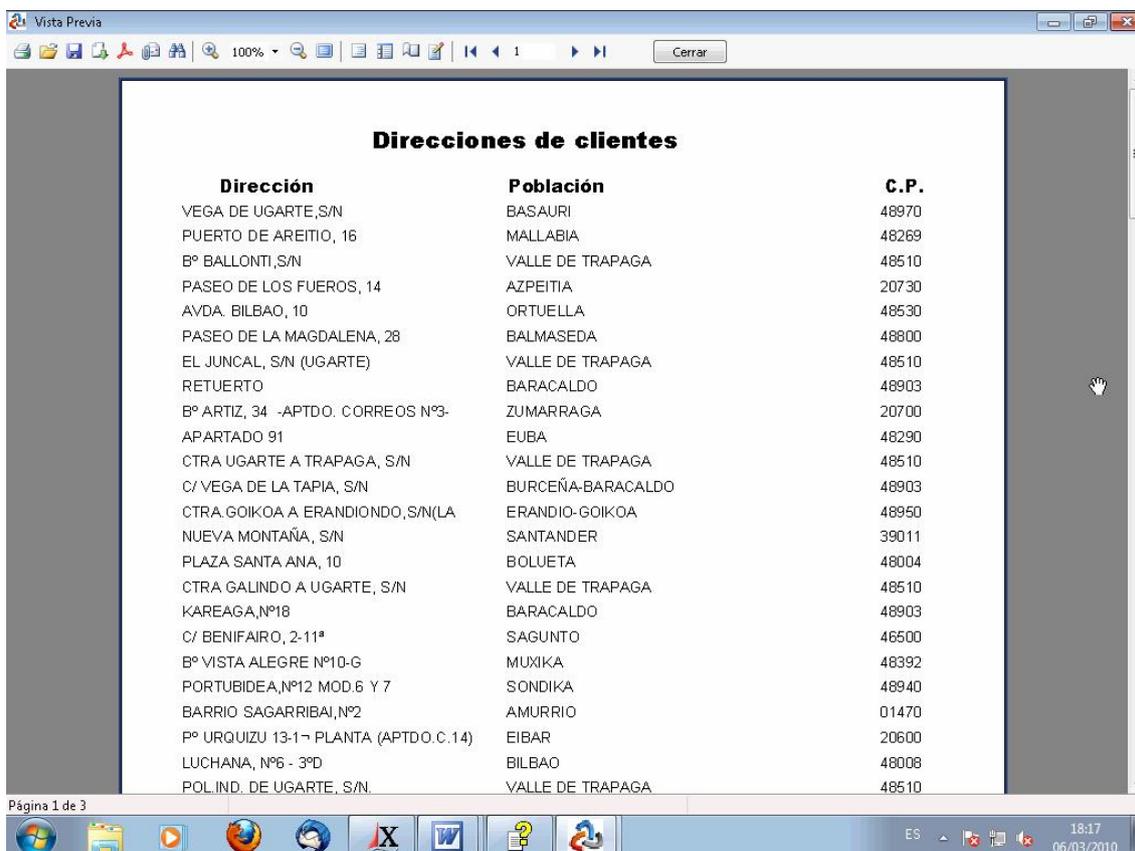
```
SetHbVar('Nombrevariable',xValor)
SetHbArrayVar('NombreArray',xValor,nItem)
```

Se pueden también crear codeblocks, evaluarlos y destruirlos.

Finalmente se pueden obtener imágenes grabadas en strings desde campos memo o blob con la función LoadImageFromHbVar

Mediante las macroexpresiones podremos hacer auténticas diabluras en nuestros informes pero todo tiene un límite, para llegar más allá a donde las funciones de FastReport no llegan deberemos de usar su lenguaje de scripts, lo cual veremos más adelante.

## Tema 11. Manipular la vista previa de FR



La vista previa de FR es potentísima, no solo se limita a mostrar el resultado del informe por pantalla sino que permite hacer maravillas desde la barra de herramientas mostrada.

Esta barra de herramientas puede ser limitada desde el código fuente de la aplicación ya que hay opciones que podemos querer que el usuario final no utilice como por ejemplo la edición en línea de la previsualización y que luego imprima algo que el programa no mostró.

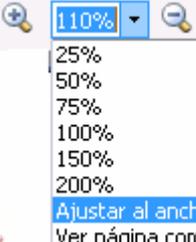
Esto se hace mediante el método `Fr:PreviewOptions:SetButtons(1535)`

Este número entre paréntesis indica que botones se quieren mostrar o no y se obtiene por la suma de los códigos de los botones que si se desean mostrar con la siguiente lista

Botón PRINT	1
Botón LOAD	2
Botón SAVE	4
Botón EXPORT	8
Botón ZOOM	16
Botón FIND	32
Botón OUTLINE	64
Botón PAGESETUP	128
Botón TOOLS	256
Botón EDIT	512
Botón NAVIGATOR	1024

Así por ejemplo se puede ocultar el botón EDIT y no dejar que el usuario final modifique los informes, o que no pueda exportarlos, etc.

**Botones de la barra de herramientas.**

	<p>Imprime el documento por impresora, si no se ha especificado nada se mostrará el dialogo de selección y configuración de impresora, pero si se ha indicado una impresora concreta en el diseño saldrá por ella, o si se ha forzado a no mostrar el diálogo de impresión saldrá por la impresora por defecto. ::PrintOptions:SetShowDialog( .F. )</p>
	<p>Abrir un informe previamente grabado en el disco.</p>
	<p>Grabar un informe en el disco. Aunque no parece muy lógico se puede grabar el informe previsualizado (no el diseño sino el resultado) en el disco en un archivo especial de tipo FP3 que luego puede ser abierto con la opción anterior de FR.</p>
	<p>Exportar permite convertir el informe en un archivo, de entre los múltiples tipos que se muestran en el menú que se ve a la derecha y grabarlo en el disco para utilizarlo después con otro programa como por ejemplo Acrobat Reader, Excel, Open Office, etc. o incluso enviarlo directamente por email aunque para esto haya que configurar los datos de SMTP del cliente.</p> <p>Cada formato de visualización muestra una ventana de opciones al efecto, que también puede ser manipulada por código fuente, incluso para establecer unos valores y no ser mostrada:</p> <pre> ::SetProperty("PDFExport", "FileName", cExportFileName) ::SetProperty("PDFExport", "EmbeddedFonts", .t.) ::SetProperty("PDFExport", "ShowDialog", .f.)                 </pre> <div data-bbox="959 577 1353 1003" style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> <li>Archivo PDF...</li> <li>Archivo HTML...</li> <li>Archivo RTF...</li> <li>Archivo CSV...</li> <li>Tabla Excel (OLE)...</li> <li>Imagen BMP...</li> <li>Imagen JPEG...</li> <li>Imagen TIFF...</li> <li>Imagen Gif...</li> <li>Archivo de Texto...</li> <li>E-mail...</li> <li>Tabla Excel (XML)...</li> <li>Texto (Impresora matricial)...</li> <li>Hoja de Cálculo Open Document ODS...</li> <li>Documento de Texto Open Document ODT...</li> <li>Fichero Excel 97/2000/XP...</li> </ul> </div>
	<p>Buscar es capaz de localizar un texto dentro de la previsualización allí donde se encuentre, cabeceras, detalle o pies.</p>
	<p>Zoom permite cambiar el tamaño de visualización del documento presentado.</p> <p>Aunque el zoom predeterminado en la previsualización se puede realizar desde código fuente ::PreviewOptions:SetZoomMode(2) (siendo los valores 0 por defecto, 1 página entera, 2 ancho de página, 3 multipágina) se puede cambiar desde este menú de tamaños.</p>
	<p>Mostrar en Pantalla completa oculta todo salvo el documento impreso. Pulsando ESC se vuelve al modo de vista normal.</p>
	<p>Índices muestra una lista lateral a la izquierda con una lista de los grupos que se muestran en el informe, pueden anidarse subgrupos a varios niveles.</p>
	<p>Diapositivas muestra una lista lateral a la izquierda con miniaturas de la páginas, permitiendo ir a la que se desee haciendo un clic sobre ella. Este opción puede ser suministrada por código de forma que aparezca directamente en la previsualización.</p>

	<p>Configurar página permite cambiar la configuración de las páginas como su tamaño, márgenes, etc. a la página actual o a todas las del documento.</p> <p>En la práctica no es muy útil por que se cambie lo que se cambie el contenido de las páginas no se adapta a la nueva situación, es decir, si por ejemplo se amplía el margen izquierdo demasiado, se come el margen derecho y si hace falta saca el contenido de la hoja.</p>	
	<p>Editar página es una opción muy arriesgada ya que permite que una vez generado un informe y presentado en pantalla, el usuario pueda modificar el texto que desee poniendo otra cosa en su lugar, un importe de ventas diferente, un nuevo total, etc.</p>	
	<p>Barra de navegación que permite moverse por las páginas del documento atrás, adelante o escribiendo directamente la página a visualizar.</p>	
	<p>Cierra la previsualización y vuelve al diseñador o al programa.</p>	

## **Tema 12. FASTREPH.PRГ y Xailer**

Este PRГ es suministrado para ser enlazado junto con los programas (x)Harbour y el hecho de que sea un PRГ y no una LIB por ejemplo, hace que estе en manos del desarrollador hacer cualquier modificaci3n, mejora o ampliaci3n de las funcionalidades de FR.

Como contrapartida en cada nueva revisi3n o actualizaci3n de FR vendr3 un nuevo PRГ junto con la DLL y el CH y entonces habr3 que revisar por completo el c3digo de este PRГ para adaptar las diferencias que suelen ser m3nimas y a veces incluso inexistentes ya que los cambios se han hecho a nivel de la DLL 3nicamente.

En este PRГ el desarrollador puede incorporar nuevas propiedades y m3todos, que aunque no mejorar3n lo que FR finalmente hace, es decir preparar un informe, presentarlo e imprimirlo, si pueden mejorar notablemente su uso, operativa y aspecto final.

En el caso de BiSoft este PRГ est3 pr3cticamente duplicado en tama3o, incluyendo en 3l todo el interfaz de uso del desarrollador y del cliente, para que el primero realice los informes en un tiempo record y el segundo los pueda utilizar de una forma ordenada y sencilla.

M3todos nuevos como SetDatSet() env3an todos los datasets de un form a FR sin tener que indicarlos uno por uno con oFr:SetWorkArea(,:oDbfClientes:Alias(),:oDbfClientes:Area()) o como SetAppdata() que pasa a FR todas las variables de AppData que tenga la aplicaci3n.

Adem3s se han incorporado funcionalidades que FR no tiene como integraci3n con MySQL y funciones nuevas para ejecuci3n de c3digo MySQL en tiempo de ejecuci3n o impresi3n de im3genes desde campos Blob de la base de datos.

Incluso se ha realizado una sobrecarga a la clase Browse para que se pueda imprimir un Browse completo simplemente llamando a ::oBrowse:FastReport()

## Tema 13. Métodos de FastReport.

El uso de FastReport se basa en la clase **frReportManager** que es la que engloba todo el motor de informes.

Vamos por lo tanto a desgranar los métodos necesarios para su uso.

### 13.1 Métodos constructor y destructor.

#### **:New()**

Es el método constructor de la clase, inicializa y carga el motor de informes de FastReport.

**Nota:** Se puede cargar FastReport al momento de iniciar la aplicación y descargarlo al finalizar, o cargarlo cada vez que haga falta y descargarlo cuando se quiera ya que la carga es rapidísima, el motor muy pequeño y no hay problema en cargarlo y descargarlo repetidamente siendo su consumo de recursos muy reducido.

Si se carga al inicio, el objeto creado se puede guardar en una variable pública o que tenga visibilidad durante todo el tiempo de ejecución del programa para poder generar informes en cualquier momento.

**Sintaxis:** frReportManager():New([<cRutaDll>]) -> <oFR>

#### **Parámetros:**

<cRutaDll>	Opcional, es una cadena con la ruta donde se encuentra la DLL de FastReport (FRSyst.dll). Si este parámetro se omite la DLL se buscará en la misma carpeta donde se encuentra el programa EXE que intenta cargarla, opción más habitual
------------	---

#### **Devuelve:**

<oFR>	Un objeto instanciado de la clase FastReport a partir del cual obtendremos los informes manipulando sus métodos y propiedades.
-------	--

#### **Ejemplo:**

```
oFr:= frReportManager():New("C:\Path\FrSystH.dll")
      o simplemente
oFr:= frReportManager():New()
```

#### **:DestroyFR()**

Es el método destructor de la clase.

Destruye todos el objeto instanciado de FastReport y descarga FRsyst.dll de la memoria.

Se puede llamar a este método al finalizar la ejecución del programa si se ha instanciado en el inicio o cada vez que se haya instanciado para un informe concreto.

**Sintaxis:** <oFr>:DestroyFR() -> NIL

**Parámetros:** Ninguno

**Devuelve:** Nada

#### **Ejemplo:**

```
oFr:DestroyFR()
```

## 13.2 Métodos para cargar diseños, diseñar informes y grabar estos diseños.

Una vez cargado FastReport por nuestro programa, los siguientes métodos son los más usados para la carga de diseños, el diseño de los informes y la presentación e impresión de los mismos.

### **:LoadFromFile()**

Carga un diseño de informe previamente guardado en disco, para su modificación o vista previa, impresión o exportación.

Los archivos se guardan en formato Xml y con extensión FR3, pudiendo ser editados si fuera necesario con un simple editor de textos o por la aplicación.

**Sintaxis:** <oFr>:LoadFromFile(<cNomArch>) -> <lCargado>

#### **Parámetros:**

<cNomArch>	Nombre del archivo del informe que deseamos cargar. El nombre debe incluir ruta completa de donde se encuentra el archivo, así como el nombre de dicho archivo. Lógicamente, este parámetro <b>es obligatorio</b> .
------------	---

#### **Devuelve:**

<lCargado>	Un valor lógico que es .T. si el informe pudo ser cargado, y .F. si hay algún problema con el archivo que contiene el informe, por ejemplo en caso de que el archivo no exista.
------------	---

#### **Ejemplo:**

```
IF .NOT. oFr:LoadFromFile(Application:cDirectory+" Reports\Cientes.fr3")
  MsgInfo("Problema cargando el informe de Clientes")
ENDIF
```

### **:LoadFromBlob()**

Carga un informe almacenado en un campo memo o un campo blob de una tabla de datos.

**Sintaxis:** <oFR>:LoadFromBlob(<nWorkArea>, <cNomCampo>)

#### **Parámetros:**

<nWorkArea>	Número del área de trabajo donde se encuentra abierta la tabla de datos por (x)Harbour, no por FastReport. Este número es el valor devuelto por la función SELECT().
<cNomCampo>	Nombre del campo memo o blob donde se encuentra almacenado el informe.

### **:SaveToBlob()**

Guarda un informe en un campo memo o blob de una tabla de datos, de esta manera no se encuentra el archivo FR3 en el disco, no pudiendo ser manipulado por ningún usuario.

**Sintaxis:** <oFR>:SaveToBlob(<nWorkArea>, <cNomCampo>)

#### **Parámetros:**

<nWorkArea>	Número del área de trabajo donde se encuentra abierta la tabla de datos por (x)Harbour, no por FastReport. Este número es el valor devuelto por la función SELECT().
<cNomCampo>	Nombre del campo memo o blob donde se va a guardar el informe.

### **:LoadFromResource()**

Carga un informe desde un recurso a partir de un nombre o un número identificador

**Sintaxis:** <oFR>:LoadFromResource( <xID> ) -> <lCargado>

**Parámetros:**

<xID>	Puede ser un valor numérico o una cadena de caracteres con el identificador del recurso (el recurso debe ser del tipo RCDATA).
-------	--

**Devuelve:**

<lCargado>	Valor lógico que indica si el informe pudo ser cargado desde el recurso cuando vale .T.. Si el informe no se pudo cargar, devuelve .F.
------------	--

### **:SaveToFile()**

Guarda un informe en un archivo con un nombre especificado. Normalmente no es necesario recurrir a este método puesto que los archivos son grabados desde el propio diseñador cuando se pulsa sobre la opción Guardar o Guardar como...

**Sintaxis:** <oFR>:SaveToFile( <cNomArchivo> )

**Parámetros:**

<cNomArchivo>	Cadena de caracteres con el nombre del archivo que va a contener el informe. Puede incluir la unidad de disco y la ruta de la carpeta donde se debe de guardar el informe.
---------------	--

### **:DesignReport()**

Lanza la ventana del diseñador visual de informes, la cual nos permitirá diseñar nuestros informes y guardarlos en disco como archivos FR3 para posteriormente presentarlos en pantalla e imprimirlos con el método :ShowReport().

Si antes de llamar a este método se llama al método :LoadFromFile(), el diseñador visual permitirá modificar el informe cargado. Si no se llama al método :LoadFromFile() antes de llamar a :DesignReport(), el diseñador se presentará en blanco para crear un nuevo informe.

**Sintaxis:** <oFr>:DesignReport() -> NIL

**Parámetros:** Ninguno

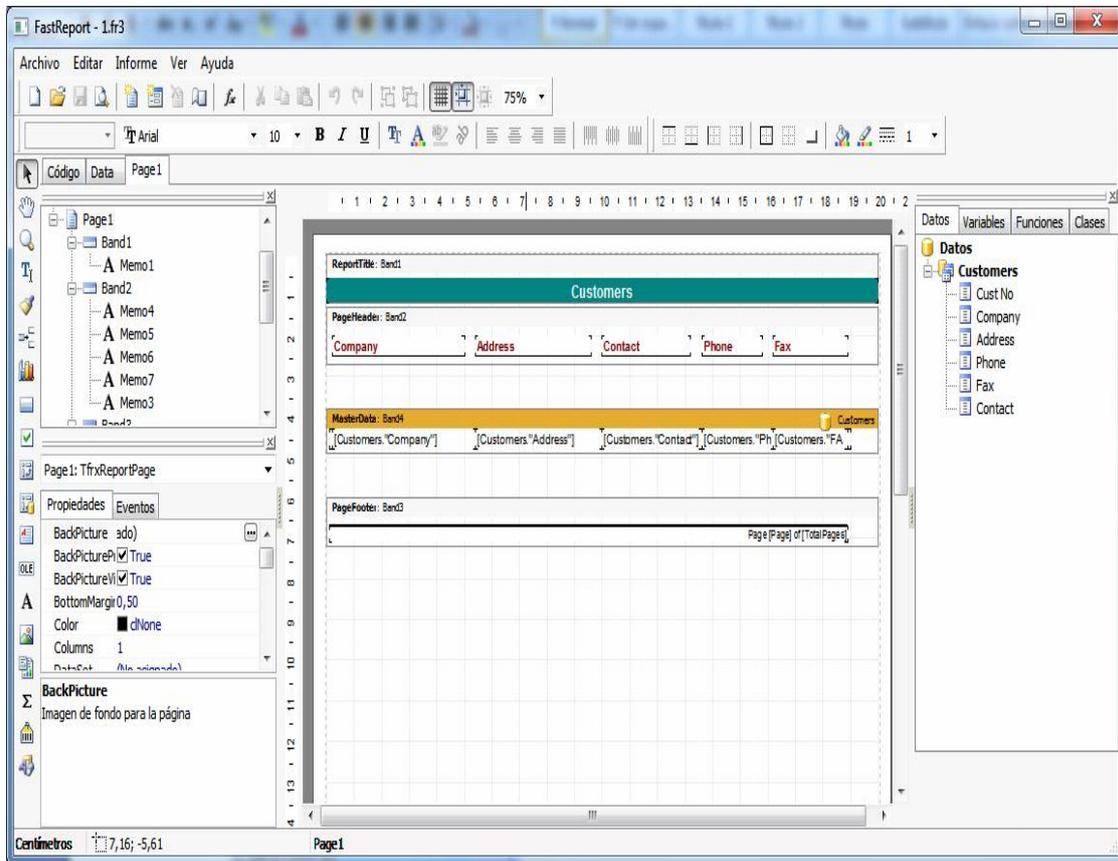
**Devuelve:** Nada

**Ejemplo:**

```

WITH OBJECT FrPrn
  IF .NOT. oFr:LoadFromFile("C:\midir\informes\clientes.fr3")
    MsgInfo("Problema cargando el informe de Clientes")
  ELSE
    :DesignReport()
  ENDIF
END

```



## :Clear()

Limpia (cierra) el informe actualmente en uso tanto para diseño con `:DesignReport()` como para presentarlo en pantalla e imprimirlo con `:ShowReport()`. Se usa este método para finalizar un informe, si se ha recurrido a una única instanciación de FR y por lo tanto no se puede invocar a `:Destroy()` salvo al final del programa. En caso de invocar a `:New()` para cada informe se debe de utilizar `:Destroy()`

**Sintaxis:** <oFr>:Clear() -> NIL

**Parámetros:** Ninguno

**Devuelve:** Nada

### 13.3 Métodos para presentar, imprimir y exportar informes.

#### **:ShowReport()**

Genera el informe que previamente fue cargado usando el método :LoadFromFile() o :LoadFromBlob() o :LoadFromResource() y lo muestra en la ventana de vista previa de impresión.

**Sintaxis:** <oFr>:ShowReport([<nNoBorrar>]) -> NIL

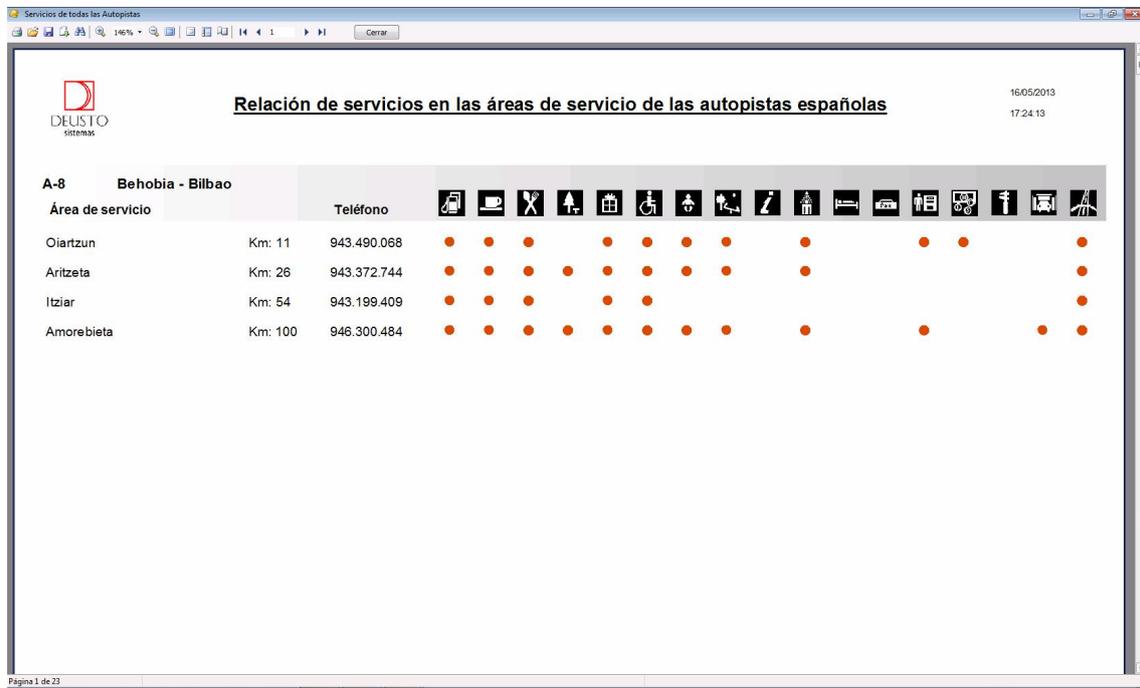
#### **Parámetros:**

<b>&lt;nNoBorrar&gt;</b>	Es un parámetro numérico opcional, que se utiliza para realizar un multipreview. Viene definida por la constante FR_NOCLEARLASTREPORT que es igual a 1. Indica que el nuevo informe se agregará al que está cargado actualmente en el motor de vista previa. Si este parámetro se omite, se borrará cualquier informe previamente mostrado en la ventana de vista previa y posteriormente presentar el nuevo informe.
--------------------------	---

**Devuelve:** Nada

#### **Ejemplo:**

```
WITH OBJECT FrPrm
  IF .NOT. oFr:LoadFromFile(Application:cDirectory+" Reports\Cientes.fr3")
    MsgInfo("Problema cargando el informe de Cientes")
  ELSE
    :ShowReport()
  ENDIF
END
```



**:PrepareReport()**

Genera un informe con todos sus datos sin presentarlo en la ventana de vista previa.

**Sintaxis:** <oFR>:PrepareReport( <nNoBorrar> )

**Parámetros:**

<nNoBorrar>	Es un parámetro numérico opcional, que se utiliza para realizar un multipreview. Viene definida por la constante FR_NOCLEARLASTREPORT que es igual a 1. Indica que el nuevo informe se agregará al que está cargado actualmente en el motor de vista previa. Si este parámetro se omite, se borrará cualquier informe previamente mostrado en la ventana de vista previa y posteriormente presentar el nuevo informe.
-------------	---

**:ShowPreparedReport()**

Muestra en pantalla el informe, el cual ha sido construido previamente llamando al método ::PrepareReport(). Este informe a su vez puede estar integrado por varios informes que se pueden unir antes de ser presentados.

**Sintaxis:** <oFR>:ShowPreparedReport()

**Parámetros:** Ninguno

**Ejemplo:**

```
oFr:LoadFromFile("1.fr3")
oFr:PrepareReport()
oFr:LoadFromFile("4.fr3")
oFr:PrepareReport(FR_NOCLEARLASTREPORT)
oFr:ShowPreparedReport()
```

**:Print()**

Envía directamente a imprimir el informe que se encuentre cargado.

Para mas información sobre las opciones de impresión revise la clase frPrintOptions que se detalla más adelante en este manual.

**Sintaxis:** <oFR>:Print( <IDialogoHijo> )

**Parámetros:**

<IDialogoHijo>	Valor lógico. Cuando es .T. indica que el diálogo de las opciones de impresión será hijo de la ventana o formulario que se encuentre activa (por omisión: .F.).
----------------	---

**:DoExport()**

Exporta directamente un informe a un formato específico de entre los varios formatos de exportación de FastReport. Cada formato es a su vez un objeto que se puede manipular para configurar la forma en que se deben de exportar los informes.

**Sintaxis:** <oFR>:DoExport( <cFormato> )

**Parámetros:**

<code>&lt;cFormato&gt;</code>	<p>Cadena de caracteres con el nombre del formato al cual deseamos exportar un informe. Los valores posibles pueden cualquiera de estos:</p> <table> <tr> <td>PDFExport</td> <td>Archivos PDF de Acrobat Reader</td> </tr> <tr> <td>HTMLEXport</td> <td>Página web para navegador de Internet</td> </tr> <tr> <td>RTFExport</td> <td>Archivo Rich Text Format para Microsoft Word</td> </tr> <tr> <td>CSVExport</td> <td>Texto con separadores por columna</td> </tr> <tr> <td>XLSExport</td> <td>Archivo XLS para Microsoft Excel</td> </tr> <tr> <td>DotMatrixExport</td> <td>Archivo texto para impresión en matriz de puntos</td> </tr> <tr> <td>BMPExport</td> <td>Archivo de imagen BMP</td> </tr> <tr> <td>JPEGEExport</td> <td>Archivo de imagen JPG</td> </tr> <tr> <td>TXTEExport</td> <td>Archivo de texto</td> </tr> <tr> <td>TIFFExport</td> <td>Archivo de imagen TIFF</td> </tr> <tr> <td>GIFExport</td> <td>Archivo de imagen GIF</td> </tr> <tr> <td>SimpleTextExport</td> <td>Archivo de texto simple</td> </tr> <tr> <td>MailExport</td> <td>Envío por correo electrónico del informe en alguno de los formatos de exportación</td> </tr> <tr> <td>XMLEExport</td> <td>Archivo XML</td> </tr> <tr> <td>ODSEExport</td> <td>Archivo de Calc de Open/Star Office</td> </tr> <tr> <td>ODTEExport</td> <td>Archivo de Write de Open/Star Office</td> </tr> </table>	PDFExport	Archivos PDF de Acrobat Reader	HTMLEXport	Página web para navegador de Internet	RTFExport	Archivo Rich Text Format para Microsoft Word	CSVExport	Texto con separadores por columna	XLSExport	Archivo XLS para Microsoft Excel	DotMatrixExport	Archivo texto para impresión en matriz de puntos	BMPExport	Archivo de imagen BMP	JPEGEExport	Archivo de imagen JPG	TXTEExport	Archivo de texto	TIFFExport	Archivo de imagen TIFF	GIFExport	Archivo de imagen GIF	SimpleTextExport	Archivo de texto simple	MailExport	Envío por correo electrónico del informe en alguno de los formatos de exportación	XMLEExport	Archivo XML	ODSEExport	Archivo de Calc de Open/Star Office	ODTEExport	Archivo de Write de Open/Star Office
PDFExport	Archivos PDF de Acrobat Reader																																
HTMLEXport	Página web para navegador de Internet																																
RTFExport	Archivo Rich Text Format para Microsoft Word																																
CSVExport	Texto con separadores por columna																																
XLSExport	Archivo XLS para Microsoft Excel																																
DotMatrixExport	Archivo texto para impresión en matriz de puntos																																
BMPExport	Archivo de imagen BMP																																
JPEGEExport	Archivo de imagen JPG																																
TXTEExport	Archivo de texto																																
TIFFExport	Archivo de imagen TIFF																																
GIFExport	Archivo de imagen GIF																																
SimpleTextExport	Archivo de texto simple																																
MailExport	Envío por correo electrónico del informe en alguno de los formatos de exportación																																
XMLEExport	Archivo XML																																
ODSEExport	Archivo de Calc de Open/Star Office																																
ODTEExport	Archivo de Write de Open/Star Office																																

**Propiedades de los Objetos para Exportación de Reportes**

Los objetos de exportación de informes de todos los formatos anteriores, tienen un conjunto común de propiedades que son:

Propiedad	Tipo	Descripción
ShowDialog	L	.T. indica que se debe mostrar el dialogo de configuración para la exportación
FileName	C	Nombre del archive al cual se va a exportar el informe
ExportNotPrintable	L	.T. indica que el archivo exportado no debe de ser idéntico al informe impreso
UseFileCache	L	.T. indica que se debe usar un archivo para almacenamiento temporal durante la generación del informe
DefaultPath	C	Ruta de acceso predeterminada donde se va a generar el informe exportado
ShowProgress	L	.T. indica que se debe mostrar los diálogos de progreso durante la generación del informe
OverwritePrompt	L	.T. indica que debe aparecer un dialogo de aviso en caso de que ya exista previamente otro archivo con el mismo nombre.

**Propiedades del objeto "PDFExport"**

Propiedad	Tipo	Descripción
Compressed	L	.T. indica que el archive PDF tiene que generarse comprimido
EmbeddedFonts	L	.T. indica que los tipos de letra usados en el informe deben incluirse en el archive PDF
OpenAfterExport	L	.T. indica que después de generar el archive PDF debe abrirse el Acrobat Reader para visualizarlo
PrintOptimized	L	.T. indica que se debe usar la impresión optimizada
Outline	L	.T. indica que se debe incluir el árbol con el resumen de datos (outline) dentro del archivo PDF, esta propiedad solo se puede usar cuando el informe tenga activo el resumen de datos.
Author	C	Nombre del autor del informe
Subject	C	Descripción del archive PDF
Background	L	.T. indica que la exportación debe realizarse en segundo plano
Creator	C	Nombre de la herramienta utilizada para generar el PDF

**Propiedades del objeto "HTMLExport"**

Propiedad	Tipo	Descripción
OpenAfterExport	L	.T. indica que se debe de abrir el archivo exportado con el navegador de Internet predeterminado
FixedWidth	L	.T. indica que el ancho de las tablas debe de permanecer fijo y no ajustarse al tamaño de la ventana del navegador si se cambia el ancho de la ventana.
ExportPictures	L	.T. indica que las imágenes del informe deben de ser exportadas también.
PicsInSameFolder	L	.T. indica que las imágenes del archivo html exportado sean colocadas en el mismo folder donde se genera el archivo exportado.
ExportStyles	L	.T. Indica que se deben de exportar los estilos y tipos de letra de la hoja html. Si se asigna a .F. el informe se exporta mas rápido pero la presentación no tiene la misma calidad que el informe en la vista previa.
Navigator	L	.T. indica que se debe crear un índice de links html al principio del archivo para navegar mas rápidamente por el archivo html donde se exporta el informe.
Multipage	L	.T. indica que cada hoja del informe debe generarse en un archivo html por separado
MozillaFrames	L	.T. indica que el archive html debe de exportarse con soporte para marcos de FireFox
UseJpeg	L	.T. indica que las imágenes deben de ser exportadas en formato .JPG. Si se usa esta propiedad no se puede usar la propiedad UseGif
UseGif	L	.T. indica que las imágenes deben de exportarse en formato .GIF. Si se usa esta propiedad no se puede usar la propiedad UseJpeg
AbsLinks	L	.T. indica que los links del informe exportado deben de indicarse con la ruta completa.
Background	L	.T. indica que se debe exportar un fondo para la página HTML, si este fue definido para el informe.
Centered	L	.T. indica que el informe deberá aparecer centrado en la ventana del navegador.
EmptyLines	L	.T. indica que las líneas vacías deben presentarse en el informe.

**Propiedades del objeto "RTFExport"**

Propiedad	Tipo	Descripción						
ExportPageBreaks	L	.T. indica que los saltos de hoja deben de ser incluidos en el archivo RTF exportado.						
ExportPictures	L	.T. indica que las imágenes del informe deban ser incluidas en el archivo RTF.						
OpenAfterExport	L	.T. indica que se abra el archivo RTF después de exportar el informe.						
Wysiwyg	L	.T. indica que el archivo RTF debe generarse exactamente igual a como se generó el informe en FastReport.						
Creator	C	Nombre del producto que generó el archivo RTF						
SuppresPageHeadersFooters	L	.T. indica que se deben suprimir los encabezados y los pies de página.						
HeaderFooterMode	C	Valor que indica como deben de exportarse los encabezados y los pies de página, se deben de indicar las constantes: <table border="1" data-bbox="762 1787 1366 1917"> <tbody> <tr> <td>hfText</td> <td>Se exportan como textos simples</td> </tr> <tr> <td>hfPrint</td> <td>Se exportan insertados en el documento</td> </tr> <tr> <td>hfNone</td> <td>No se exportan</td> </tr> </tbody> </table>	hfText	Se exportan como textos simples	hfPrint	Se exportan insertados en el documento	hfNone	No se exportan
hfText	Se exportan como textos simples							
hfPrint	Se exportan insertados en el documento							
hfNone	No se exportan							

**Propiedades del objeto “CSVExport”**

Propiedad	Tipo	Descripción
Separador	C	Indica cual es el carácter separador de los datos
OEMCodepage	L	.T. indica que los datos se exportan en formato OEM
OpenAfterExport	L	.T. indica que el archivo exportado debe de abrirse después de generarse.

**Propiedades del objeto “XLSExport”**

Propiedad	Tipo	Descripción
ExportStyles	L	.T. indica que los estilos, tipos de letra y otros atributos del texto deben de exportarse. Si se indica .F. el informe se exporta mas rápido pero pierde calidad visual.
ExportPictures	L	.T. indica que las imágenes deben de ser exportadas al archivo .XLS
OpenExcelAfterExport	L	.T. indica que el archivo exportado deberá abrirse con Excel después de ser generado.
Wysiwyg	L	.T. indica que el archivo exportado deberá ser visualmente idéntico al informe generado por FastReport, si esta opción se asigna a .F. se reduce considerablemente el número de renglones y columnas necesarios para generar el informe.
AsText	L	.T. indica que todos los datos, sin importar su tipo deben de ser explotados como textos. Esta opción es particularmente útil cuando el informe tiene fórmulas muy complejas.
Background	L	.T. indica que se deben exportar los colores de fondo del informe a las celdas de Excel.
FastExport	L	.T. indica que se debe de utilizar el algoritmo de FR para exportación rápida de datos a Excel . F. desactiva esta opción, la exportación se realiza mas lentamente, pero garantiza la compatibilidad con todas las versiones de Excel.
PageBreaks	L	.T. indica que los saltos de página también debe ser contemplados en la exportación a Excel.
EmptyLines	L	.T. indica que se deben exportar las líneas en blanco del informe como renglones vacíos de Excel.
SupperssPageHeadersFooters	L	.T. indica que se deben suprimir los encabezados y pies de página para las hojas de Excel.

**Propiedades de los objetos “BMPExport, TIFExport, JPEGExport, GIFExport”**

Propiedad	Tipo	Descripción
CropImages	L	.T. Indica que después de exportar el informe, las áreas en blanco alrededor del los datos se eliminan.
Monochrome	L	.T. indica que la imagen debe generarse en blanco y negro.
JPEGQuality	N	Valor numérico que indica el rango de compresión de la imagen JPG

**Propiedades del objeto “MailExport”**

Propiedad	Tipo	Descripción
Address	C	Dirección de correo electrónico del destinatario
Subject	C	Asunto del correo electrónico
Lines	C	Texto con el cuerpo del mensaje de correo

ShowExportDialog	L	.T. indica que se debe mostrar el dialogo de exportación el cual permitirá seleccionar el formato con el cual se enviará el informe por correo electrónico como archivo anexo
FromMail	C	Dirección de correo electrónico del remitente
FromCompany	C	Nombre del remitente o de su organización
Signature	C	Texto con la firma que deberá adjuntarse al correo
Smtphost	C	Nombre del servidor de correo saliente SMTP
Smtport	N	Puerto que utilice el servidor SMTP
Login	C	Nombre del usuario si el servidor SMTP requiere autenticación
Password	C	Password del usuarios si el servidor SMTP requiere autenticación
UseIniFile	L	.T. indica que la configuración del correo debe de ser tomada de un archivo .INI
Logfile	C	Nombre del archive que contendrá los mensajes que se generen durante el proceso de envío del correo electrónico

### Propiedades del objeto "XMLExport"

Propiedad	Tipo	Descripción
ExportStyles	L	.T. Indica que los estilos, tipos de letra y demás atributos de los textos deben ser exportados. Si se indica .F. el informe se genera mas rápido pero se pierde calidad visual
ExportPageBreaks	L	.T. indica que se deben exportar los saltos de página al archivo XML exportado.
OpenExcelExport	L	.T. Indica que el documento debe abrirse con Excel después de generarse.
ShowProgress	L	.T. indica que se debe mostrar el diálogo que muestra el avance del progreso en la generación del archivo.
Wysiwyg	L	.T. indica que el informe debe de exportarse exactamente igual a como lo presenta el motor de vista previa de FastReport.
Background	L	.T. indica que se debe exportar la imagen asignada a la página al archivo, esto aumenta considerablemente el tamaño del archivo.
Creator	C	Nombre del programa que generó el archivo XML
EmptyLines	L	.T. indica que se deben incluir las líneas en blanco en el archivo exportado.
SupressPageHeaderFooters	L	.T. indica que se deben de suprimir los encabezados y pies de página en el archivo a exportar.

### Propiedades del objeto "SimpleTextExport"

Propiedad	Tipo	Descripción
PageBreaks	L	.T. indica que se deben exportar al archivo los códigos de los saltos de página.
Frames	L	.T. indica que se deben exportar los marcos de los textos.
EmptyLines	L	.T. indica que se deben exportar las líneas en blanco.
OEMCodepage	L	.T. indica que se debe exportar los datos en formato OEM.
OpenAfterExport	L	.T. indica que se debe de abrir el archivo de texto después de exportarlo.

### Propiedades de los objetos "ODTExport" y "ODSExport"

Propiedad	Tipo	Descripción
ExportStyles	L	.T. Indica que los estilos, tipos de letra y demás atributos de los textos deben ser exportados. Si se indica .F. el informe se genera mas rápido pero se pierde calidad visual
ExportPageBreaks	L	.T. indica que se deben exportar los saltos de página al

		archivo de OpenOffice.
OpenAfterExport	L	.T. Indica que el documento debe abrirse con OpenOffice después de generarse.
ShowProgress	L	.T. indica que se debe mostrar el diálogo que muestra el avance del progreso en la generación del informe.
Wysiwyg	L	.T. indica que el informe debe de exportarse exactamente igual a como lo presenta el motor de vista previa de FastReport.
Background	L	.T. indica que se debe exportar la imagen asignada a la página al archivo, esto aumenta considerablemente el tamaño del archivo.
Creador	C	Nombre del programa que generó el archivo de OpenOffice
EmptyLines	L	.T. indica que se deben incluir las líneas en blanco en el archivo exportado.
SupressPageHeaderFooters	L	.T. indica que se deben de suprimir los encabezados y pies de página en el archivo a exportar.

### :SendMail()

Envía un correo electrónico directamente desde FastReport, este correo electrónico no necesariamente se utiliza para enviar informes como archivos anexos al correo, se puede utilizar para enviar mensajes simples.

NOTA: En la versión 4 de FastReport no se pueden utilizar servidores de correo con SSL, esto hace que en la mayoría de los casos no se pueda utilizar. Más adelante veremos un ejemplo de cómo puentear el sistema de envío de FastReport para que use un sistema de envío propio.

**Sintaxis:** <oFR>:SendMail( <cServidor>, <nPuerto>, <cUsuario>, <cPass>, <cDe>, <cPara>, <cAsunto>, <cEmpresa>, <cTexto>, <cArchivos>) -> <cError>

### Parámetros

<cServidor>	Cadena de caracteres con el nombre del servidor de correo saliente SMTP.
<nPuerto>	Valor numérico con el numero de puerto que utiliza el servidor de correo saliente (por omisión 25).
<cUsuario>	Cadena de caracteres con el nombre del usuario si el servidor de correo requiere autenticación
<cPass>	Cadena de caracteres con la clave de acceso al servidor si se requiere autenticación para enviar correo.
<cDe>	Cadena de caracteres con la dirección de correo electrónico del remitente.
<cPara>	Cadena de caracteres con la dirección de correo electrónico del destinatario
<cAsunto>	Cadena de caracteres con el asunto del correo electrónico
<cEmpresa>	Cadena de caracteres con el nombre de la empresa del remitente
<cTexto>	Cadena de caracteres con el texto del cuerpo del mensaje de correo
<cArchivos>	Cadena de caracteres con la lista de archivos a enviar como documentos anexos al correo.

### Devuelve:

<cError>	Una cadena vacía si no se ha generado ningún error, de lo contrario un texto con el error generado al momento del envío del correo electrónico.
----------	---

### Ejemplo:

```
#DEFINE CRLF CHR(13)+CHR(10)
cTextField := "Hola" + CRLF + "¿ Que tal ?"
cFileNames := "C:\MyAttach1.bin"+ CRLF + "C:\MyAttach2.txt"
```

```
S := oFr:SendMail("servidor.com",;  
                25,;  
                "Usuario",;  
                "Password",;  
                "mio@dominio.com",;  
                "para@dominio.com",;  
                "Asunto",;  
                "BiSoft",;  
                cTextField,;  
                cFileNames)  
  
IF S <> ""  
    MsgInfo(S)  
ELSE  
    MsgInfo("Correo enviado")  
ENDIF
```

## 13.4 Trabajando con variables y bases de datos.

Analicemos la manera en que nuestros informes trabajarán con bases de datos.

De manera general podemos dividir los datos para construir nuestros informes en 3 grupos lógicos:

1. Valores que pueden provenir de cualquier variable, sin importar de que tipo sea, también valores calculados como por ejemplo los que devuelvan las funciones contenidos en nuestro programa.
2. El segundo grupo lo llamaremos “WorkArea” (área de trabajo), para efectos de tablas DBF o ADT, aunque también podemos tener componentes como DataSets, RecordSets, etc. En general en esta categoría entraría toda la información que se puede obtener de una base de datos, sin importar su formato de archivo
3. El último grupo lo conforma aquella información que no es un WorkArea, pero puede ser presentada como tal, por ejemplo una matriz bidimensional.

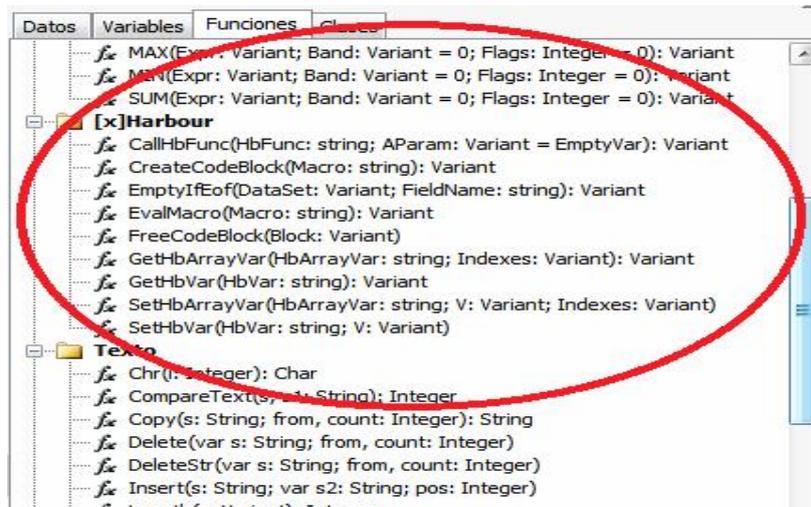
### 15.4.1 Variables

Analicemos la manera en que obtenemos datos del primer grupo.

Existen varias maneras de transferir datos de nuestro programa (x)Harbour al motor de FastReport.

La primera manera de obtener valores de las variables, o bien de obtener los valores retornados por funciones y/o procedimientos es usar las funciones que tiene FastReport incorporadas para tales tareas.

Dentro del diseñador de FastReport existe una pestaña llamada FUNCIONES y dentro de esa pestaña una sección con funciones especiales para obtener datos de un programa xHarbour:



Analicemos las funciones en detalle. La declaración de estas funciones se presenta en sintaxis de Object Pascal (FastReport está programado en Delphi), pero no hay nada malo en ello, baste con entender que en Pascal, una variable tipo “Variant” es una variable de cualquier tipo para (x)Harbour, de decir, cuando nos encontramos con un “variant”, este puede ser un valor numérico, carácter, lógico, fecha, o cualquier otro valor válido de (x)Harbour.

Estas funciones pueden ser arrastradas sobre un informe, a continuación se documentan y ejemplifica el uso de estas funciones en nuestro informe FastReports:

#### GetHbVar()

Obtiene el valor de cualquier variable definida en nuestro programa (x)Harbour, excepto los valores de objetos y matrices o vectores.

**Sintaxis:** GetHbVar(HbVar: Cadena de caracteres):Variant

**Parámetros:**

HbVar	Cadena de caracteres con el nombre de la variable definida en nuestro programa (x)Harbour. Ojo ha de ser pública.
-------	---

**Devuelve:**

Variant	El valor de la variable solicitado, de cualquier tipo: carácter, numérico, lógico o fecha.
---------	--

**Ejemplo:**

Si en nuestro código (x)Harbour tenemos algo como esto:

```
...
Public cNombre, dFecha
cNombre := "Pedro Perez"
dFecha := Date()
...
```

En nuestro informe FastReport usaremos esto para obtener los valores de dichas variables en tiempo de ejecución de nuestro programa:

```
[GetHbVar('cNombre')]
[GetHbVar('dFecha')]
```

## GetHbArrayVar()

Obtiene el valor de un solo elemento de un array (array).

**Sintaxis:** GetHbVar(HbArrayVar: Cadena de caracteres; Indexes:Variant):Variant

**Parámetros:**

HbArrayVar	Cadena de caracteres con el nombre del array definido en nuestro programa (x)Harbour.
Indexes	Valor numérico del índice del elemento dentro del array que queremos obtener

**Devuelve:**

Variant	El valor del elemento del array solicitado, es un valor de cualquier tipo: carácter, numérico, lógico o fecha.
---------	--

**Ejemplo:**

Si en nuestro código (x)Harbour tenemos algo como esto:

```
...
Public aDias, aMeses
aDias := { 'Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes' }
aMeses:= { { "Enero",31 }, { "Febrero",28 }, { "Marzo",31 }, { "Abril",30 }, { "Mayo,31 } }
...
```

En nuestro informe FastReport usaremos esto para obtener los valores de los distintos elementos de un array:

```
[GetHbArrayVar('aDias',[3])]
[GetHbArrayVar('aMeses',[3,2])]
```

**CallHbFunc()**

Obtiene el valor que resulta de la ejecución de una función contenida en nuestro programa (x)Harbour. Esta función puede ser tanto una función propia del lenguaje de programación, así como una función definida por el usuario.

**Sintaxis:** CallHbFunc(HbFunc: Cadena de caracteres;AParam: Variant=EmptyVar): Variant

**Parámetros:**

HbFunc	Cadena de caracteres con el nombre de la función de la cual queremos obtener su valor de retorno.
AParam	Parámetro opcional, es un array con los valores de los parámetros que recibe la función de la cual queremos obtener su valor de retorno. Si se omite, se asume que la función no recibe parámetros.

**Devuelve:**

Variant	El valor retornado por la ejecución de la función con o sin parámetros.
---------	---

**Ejemplo:**

```
[CallHbFunc('SubStr', ['HGFHGFHGFHG', 2, 5])]  
[CallHbFunc('MiFuncion', [NIL, 10])]  
[CallHbFunc('DbSkip')]
```

**NOTA IMPORTANTE:** Si por algún motivo nuestra función da un error el programa se cerrará y con él la instancia de FastReport perdiendo el diseño si no lo hemos grabado previamente. Mucho cuidado, antes de previsualizar un informe que llame a una función hay que grabarlo.

**EvalMacro()**

Ejecuta (macrosustituye) una cadena de caracteres que se puede interpretar como “macro” en (x)Harbour.

**Sintaxis:** EvalMacro(Macro: Cadena de caracteres): Variant

**Parámetros:**

Macro	Cadena de caracteres que contiene una macro expresión. Cualquier expresión que pueda ser ejecutada con el operador “&” de (x)Harbour es válida.
-------	---

**Devuelve:**

Variant	El valor retornado por la ejecución de la macro.
---------	--

**Ejemplo:**

```
[EvalMacro('Variable := Directory("Ruta\mascarilla")')]
```

**CreateCodeBlock()**

Crea un bloque de código a partir de una cadena de caracteres. Esta función se usa en para escribir scripts.

**Sintaxis:** CreateCodeBlock(Macro: Cadena de caracteres): Variant

**Parámetros:**

Macro	Cadena de caracteres que se desea convertir en bloque de código. Los bloques de código creados deben ser destruidos después de usarlos.
-------	---

**Devuelve:**

Variant	El valor retornado por la ejecución del bloque de código.
---------	---

**Ejemplo:**

```
var
  CodeBlock: Integer;
procedure ReportOnStartReport(Sender: TfrxComponent);
begin
  CodeBlock := CreateCodeBlock('{||ITEMS->PartNo}');
end;
```

**FreeCodeBlock()**

Destruye un bloque de código previamente creado.

**Sintaxis:** FreeCodeBlock(Block: Variant)

**Parámetros:**

Block	Variable que contiene un bloque de código previamente creado con la función CreateCodeBlock().
-------	--

**Devuelve:** Nada

**Ejemplo:**

```
procedure ReportOnStopReport(Sender: TfrxComponent);
begin
  FreeCodeBlock(CodeBlock);
end;
```

- **El evento OnGetValue**

Una segunda manera de obtener datos para el primer grupo, consiste en definir bloques de código para controlar el evento de FastReport **OnGetValue**. La clase **frReportManager** tiene un método universal para definir un bloque de código controlador para todos los eventos de FastReport:

**:SetEventHandler()**

Define un bloque de código para controlar un evento de FastReport. Este bloque de código se ejecutará cuando ocurra el evento al cual está asignado.

**Sintaxis:** <oFr>.SetEventHandler(<cObjeto>,<cEvento>,[<bControl>])-> Nil

**Parámetros:**

<cObjeto>	Cadena de caracteres con el nombre del objeto al cual queremos definir un bloque de código para controlar un evento. Solo puede tener 2 valores "Report" para motor de vista previa ó "Designer" para el diseñador de informes.
<cEvento>	Cadena de caracteres con el nombre del evento al cual queremos asignarle un bloque de código controlador.

<bControl>	Parámetro opcional, es un bloque de código que se ejecutará cuando el evento de FastReport al cual está asignado suceda, este bloque de código nos permitirá controlar lo que pasa durante los diversos eventos que acontecen durante el proceso de generación de un informe usando nuestro propio programa
------------	---

**Devuelve:** Nada

**Ejemplo:**

```
SetEventHandler("Report", "OnProgress" ,{|x,y|MiBarra(x, y)})
```

De momento consideremos únicamente el evento "OnGetValue". Durante el proceso de generación del informe este evento se ejecutará de tal forma que podemos aprovecharlo para devolver al informe algún valor para una variable DEFINIDA EN EL DISEÑO DEL MISMO.

**Ejemplo:**

En nuestro diseñador de informes insertaremos una variable, que es simplemente un texto con el siguiente formato:

**[MiVar1]**

Para sustituir [MiVar1] por un valor calculado, en nuestro código fuente (xHarbour) escribiríamos algo como esto:

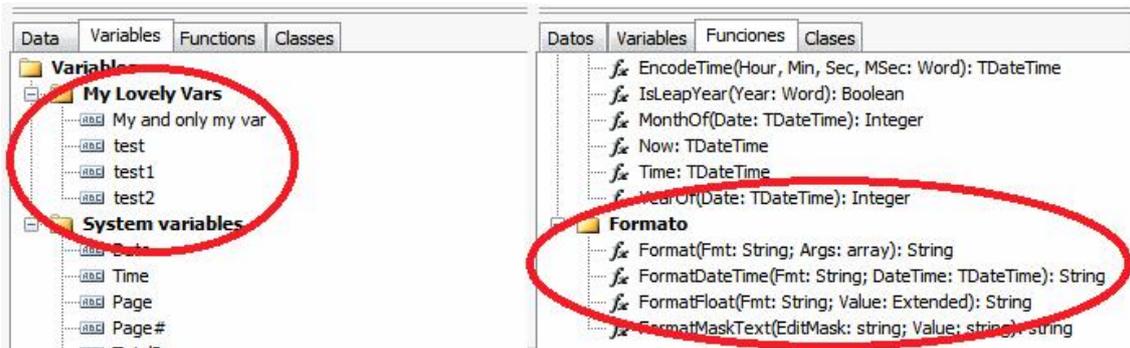
```
...
// definimos el evento:
oFr:SetEventHandler("Report", "OnGetValue",;
    {| VarName | GetValue(VarName)})
...

FUNCTION GetValue(VarName)
DO CASE
    CASE VarName == "MiVar1"
        RETURN 56
    CASE VarName == "MiVar2"
        RETURN Substr(&MiVar3, 1, 3)
    OTHERWISE
        RETURN NIL
    ENDCASE
RETURN NIL
```

De acuerdo a nuestro código anterior, al momento de visualizar nuestro informe FastReport [MiVar1] será sustituido por un "56".

***:AddFunction y :AddVariable:***

El uso de estos dos métodos resulta mas conveniente para obtener valores de nuestro programa, porque los nombres de las funciones y variables definidas por estos 2 métodos aparecen para ser utilizadas desde el diseñador visual, en la pestaña *Variables* y *Funciones* respectivamente:



## :AddFunction()

Agrega una función al diseñador de informes de FastReport.

**Sintaxis:** <oFR>:AddFunction(<cFuncion>,[<cCategoria>],[<cDescripcion>])

**Parámetros:**

<cFuncion>	<p>Cadena de caracteres con el nombre y sintaxis de la función EN SINTAXIS DE PASCAL, la sintaxis debe incluir los tipos de los parámetros enviados y el tipo de dato que la función devuelve, sin embargo, no es necesario tipificar estrictamente los tipos de variables ni el valor de retorno, ambos pueden ser declarados como <i>VARIANT</i>.</p> <p>“NombreFuncion(Aparam: Cadena de caracteres; Bparam: Lógica): Double”  “NombreFuncion(Aparam: Variant; Bparam: Variant): Variant”</p> <p>Si se usa la primer sintaxis, entonces debemos de ser cuidadosos al momento de llamar a las funciones y enviar los parámetros exactamente al tipo al que pertenecen. En la segunda sintaxis esto no es necesario.</p> <p>Las equivalencias de los tipos de datos (x)Harbour contra los tipos de pascal son:</p> <table border="1" style="width: 100%;"> <tr> <td>Carácter</td> <td>Cadena de caracteres – Cadena de caracteres</td> </tr> <tr> <td>Fecha</td> <td>TDate – TdateTime</td> </tr> <tr> <td>Lógico</td> <td>Lógica</td> </tr> <tr> <td>Numérico sin decimales</td> <td>Byte-Word-Integer-Longint-Cardinal</td> </tr> <tr> <td>Numérico con decimales</td> <td>Real-Single-Double-Extended-Currency</td> </tr> <tr> <td>Matrices o vectores</td> <td>Array-array</td> </tr> </table>	Carácter	Cadena de caracteres – Cadena de caracteres	Fecha	TDate – TdateTime	Lógico	Lógica	Numérico sin decimales	Byte-Word-Integer-Longint-Cardinal	Numérico con decimales	Real-Single-Double-Extended-Currency	Matrices o vectores	Array-array
Carácter	Cadena de caracteres – Cadena de caracteres												
Fecha	TDate – TdateTime												
Lógico	Lógica												
Numérico sin decimales	Byte-Word-Integer-Longint-Cardinal												
Numérico con decimales	Real-Single-Double-Extended-Currency												
Matrices o vectores	Array-array												
<cCategoria>	Parámetro opcional. Es una cadena de caracteres que indica el nombre de la categoría dentro del diseñador visual donde deberá agruparse la función recién definida												
<cDescripcion>	Es una pequeña descripción de lo que hace la función, esta descripción aparece en el diseñador visual cuando se selecciona la función, en la parte inferior.												

**Ejemplo:**

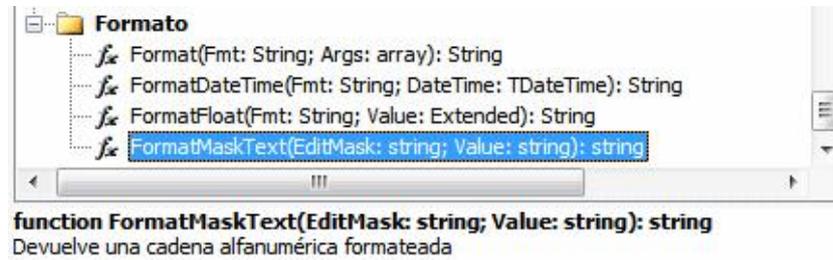
```
oFr:AddFunction("function Format(Fmt: Cadena de caracteres; Args: Array): Cadena de caracteres";
  "Formato", "Aplica una mascara a un array de argumentos y devuelve una cadena formateada")
```

```
oFr:AddFunction("function FormatDateTime(Fmt:Cadena de caracteres;DateTime:TDateTime):Cadena de caracteres";
  "Formato", "Aplica un formato a un valor Fecha/Hora")
```

```
oFr:AddFunction("function FormatFloat(Fmt: Cadena de caracteres; Value: Extended): Cadena de caracteres";
  "Formato", "Aplica un formato a un valor decimal")
```

```
oFr::AddFunction("function FormatMask(EditMask: Cadena de caracteres; Value: Cadena de caracteres): Cadena de caracteres";
    "Formato","Devuelve una cadena alfanumérica formateada")
```

Y en el diseñador visual veremos:



Otros ejemplos de funciones del usuario:

```
oFr::AddFunction("function DeleteFile(FileName: String ): Variant","Archivos", "Borra del disco un archivo")
```

```
oFr::AddFunction("function FileUnique(FilePath: String, Extension: String, RootName: String ): Variant","Archivos","Devuelve un nombre de archivo temporal")
```

```
oFr::AddFunction("function LinuxPath(FilePath: String ): Variant","Archivos","Devuelve un nombre de archivo tipo Linux convirtiendo barras \ en /")
```

```
oFr::AddFunction("function QueryValue(Select: String ): Variant","MySQL","Ejecuta una sentencia Select que devuelve un único valor")
```

```
oFr::AddFunction("function QueryArray(Select: String): Array","MySQL","Ejecuta una sentencia Select que devuelve un array de valores")
```

### ***El Evento OnUserFunction***

Las funciones añadidas al diseñador visual, se evalúan con el evento OnUserFunction, este evento, como todos los demás en FastReport, debe tener un bloque de código controlador, el cual recibe 2 parámetros: Una cadena de caracteres con el nombre de la función que se está ejecutando y un array con la lista de parámetros para la función.

#### **Ejemplo para las funciones antes descritas:**

```
Function UserFunctions(cNombre,aParams)
Local xVal, cDirectory

Do Case
Case Upper(cNombre)=="DELETEFILE"
xVal:=Ferase(aParams[1])
Case Upper(cNombre)=="FILEUNIQUE"
IF Empty(aParams[1]) .And. !ExistDir( Application:cDirectory+"Tmp" )
MakeDir( Application:cDirectory+"Tmp" )
Endif

xVal:=FileUnique(IF(Empty(aParams[1]),Application:cDirectory+"\Tmp",aParams[1]),IF(Empty(aParams[2]),"Tmp",aParams[2]),aParams[3])
LogDebug(xVal)
Case Upper(cNombre)=="LINUXPATH"
xVal:=StrTran(aParams[1],"\" ,"/")
Case Upper(cNombre)=="QUERYVALUE"
xVal:=Appdata:oSqlSource:QueryValue(aParams[1])
Case Upper(cNombre)=="QUERYARRAY"
xVal:=Appdata:oSqlSource:QueryArray(aParams[1])
End

Return xVal
```

Es necesario poner atención a la sintaxis de los parámetros opcionales, los cuales por omisión tienen valor vacío. Los parámetros siempre se deben de indicar del lado derecho de la sintaxis de la función.

Después de agregar una función, esta existirá hasta que se descargue de la memoria FastReport. A diferencia de las funciones, las variables (las definidas por el usuario) existen en el archivo que contiene el informe. Esto quiere decir que las variables son parte del informe y son guardados y cargadas cada vez que se usa el informe. Por lo general estas variables no son variables “reales” y pueden tener valores simples o pueden ser el resultado de la ejecución de una expresión.

**IMPORTANTE:** Cuando se usa una variable del informe, su valor es calculado como una cadena de caracteres. Esto quiere decir que la variable cuyo valor es “GetHbVar(xVariable)” retornará un valor de cualquier tipo definido en xVariable, pero la expresión ““GetHbVar(xVariable)”” (nótese la comilla anidada), que devolverá el valor de xVariable como una cadena de caracteres.

Las variables también pueden ser definidas desde el diseñador visual, en el menú principal opción Informe / Variables, que presenta una ventana como esta que nos permitirá definir visualmente las variables necesarias:



Es posible agregar, cambiar valores o borrar variables tanto en código como en el diseñador visual usando estos métodos:

**:AddVariable()**

Agrega una variable definida por el usuario a la lista de variables disponibles en el diseñador visual de FastReport.

**Sintaxis:** <oFR>:AddVariable( <cCategoria>, <cVariable>, <xValor> )

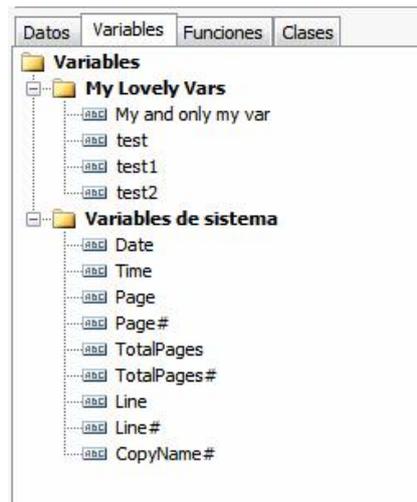
**Parámetros:**

<cCategoria>	Cadena de caracteres con el nombre de la categoría en la cual se debe agregar la nueva variable, si no existe se crea una nueva.
<cVariable>	Cadena de caracteres con el nombre de la variable que queremos crear.
<xValor>	Valor a asignar a la variable, puede ser de cualquier tipo

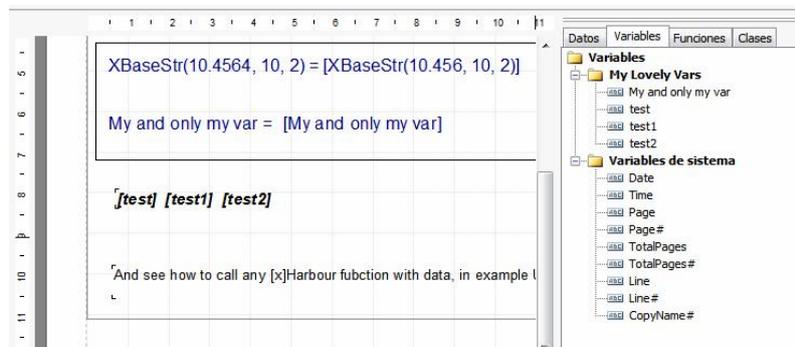
**Ejemplo:**

```
oFr:AddVariable("My Lovely Vars", "My and only my var", 10)
oFr:AddVariable("My Lovely Vars", "test", 100.25)
oFr:AddVariable("My Lovely Vars", "test1", "Test")
oFr:AddVariable("My Lovely Vars", "test2", ctod("01/01/2007"))
```

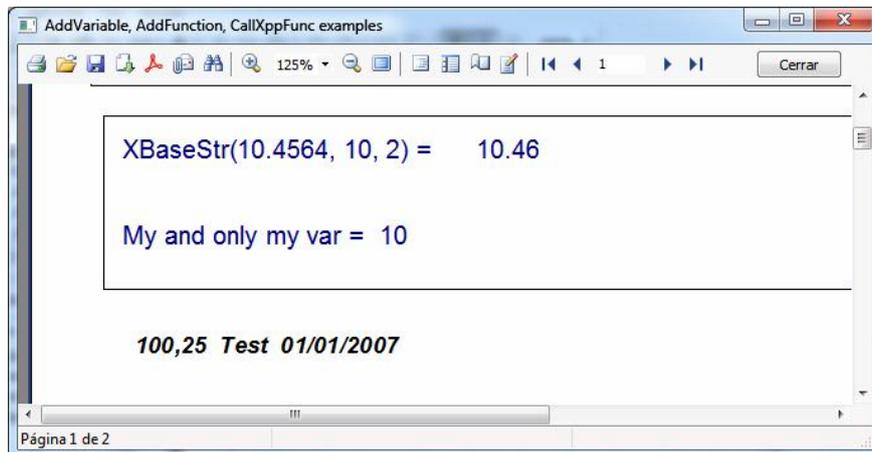
En el diseñador visual veremos:



Estas variables se pueden seleccionar y arrastrar con el ratón sobre el diseño del informe:



El valor de las variables será visible en la vista previa de informe:



## **:SetVariable()**

Modifica el valor actual de una variable definida por el usuario.

**Sintaxis:** <oFR>:SetVariable( <cVariable>, <xValor> )

**Parámetros:**

<cVariable>	Cadena de caracteres con el nombre de la variable de la cual queremos modificar su valor.
<xValor>	Valor a asignar a la variable, puede ser de cualquier tipo.

### **:GetVariable()**

Devuelve el valor de una variable definida por el usuario.

**Sintaxis:** <oFR>:GetVariable( <cVariable> )

**Parámetros:**

<cVariable>	Cadena de caracteres con el nombre de la variable de la cual queremos obtener su valor.
-------------	---

### **:DeleteVariable()**

Elimina una variable definida por el usuario del diseñador visual.

**Sintaxis:** <oFR>:DeleteVariable( <cVariable> )

**Parámetros:**

<cVariable>	Cadena de caracteres con el nombre de la variable a eliminar del diseñador visual.
-------------	--

### **:DeleteCategory()**

Elimina una categoría del diseñador visual y TODAS sus variables.

**Sintaxis:** <oFR>:DeleteCategory( <cCategoría> )

**Parámetros:**

<cCategoría>	Cadena de caracteres con el nombre de la categoría a eliminar del diseñador visual.
--------------	---

## 15.4.2 Trabajando con WorkAreas (Áreas de trabajo)

Esta es la segunda manera, y la más importante, de obtener datos para nuestros informes FastReport.

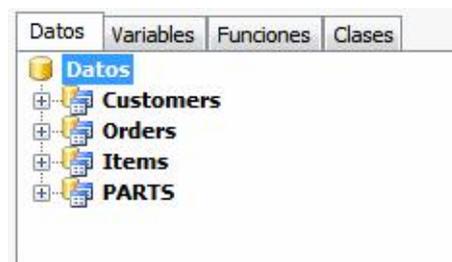
Esta forma de obtención de datos trabaja sobre tablas de datos y sus métodos nos permiten una amplia gama de complejos informes.

Los métodos más importantes, que nos permitirán crear informes básicos son:

### **:SetWorkArea()**

Añade un área de trabajo a lista de datos de FastReport, o modifica las áreas de trabajo existentes en un informe.

El uso de este método además de añadir un área de trabajo, la hace visible y disponible junto con sus campos generador visual de FastReport, en la pestaña *Datos*:



Este método debe ejecutarse SIEMPRE antes de entrar en modo de diseño con el método `::DesignReport()` o antes de imprimirlo con el método `::ShowReport()`.

**Sintaxis:** `<oFR>:SetWorkArea(<cAliasFr>, <nArea>, [<lOem>], [<aRango>])`

#### **Parámetros:**

<code>&lt;cAliasFr&gt;</code>	Cadena de caracteres con un nombre que identificará a una tabla de datos en el informe FastReport. Todas las futuras referencias a esta tabla deberán hacerse usando este nombre.
<code>&lt;nArea&gt;</code>	El número del área de trabajo de (x)Harbour donde se encuentra abierta la tabla de datos sobre la cual vamos a generar un informe, este valor se puede obtener con la función <code>SELECT()</code> de (x)Harbour
<code>&lt;lOem&gt;</code>	Parámetro opcional, valor lógico que indica si debe realizarse conversión OEM-ANSI de los datos del informe, por omisión: <code>.F.</code>
<code>&lt;aRango&gt;</code>	<p>Parámetro opcional. Define el rango de datos a imprimir, es un array de 3 posiciones que indica : Primer dato a imprimir, último dato a imprimir y el número de registros a imprimir. Estos 3 elementos del array son valores numéricos y están definidos por los siguientes valores:</p> <p>Primer elemento del array: <code>FR_RB_FIRST (0)</code> o <code>FR_RB_CURRENT(1)</code>, un valor 0 indica que se debe realizar un GOTOP y reportar los datos desde el primer registro de la tabla, un valor 1 indica que el informe debe generarse desde el registro donde se encuentre el puntero de la tabla de datos.</p> <p>Segundo elemento del array: <code>FR_RE_LAST(0)</code> ó <code>FR_RE_CURRENT(1)</code> ó <code>FR_RE_COUNT(2)</code>. El valor 0 indica que se tienen que reportar los datos hasta que se encuentre un fin de archivo (EOF), el valor 1 indica que se tienen que reportar los datos hasta donde se encuentre el puntero de la tabla de datos y un valor 2 indica que se deben de imprimir el numero total de registros que existan en la tabla.</p>

	<p>Tercer elemento del array: El número total de registros a imprimir. Si el segundo elemento del array se establece a FR_RE_COUNT (2), este tercer parámetro es ignorado y se puede pasar con un valor de 0.</p> <p>Por omisión este array es: {FR_RB_FIRST, FR_RE_LAST, 0}.</p>
--	---

**Ejemplos:**

```

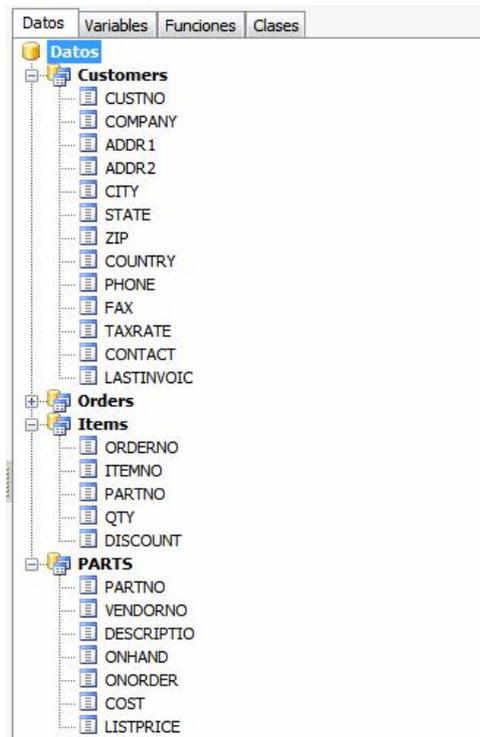
SELECT 1
USE CLIENTES SHARED
SELECT 2
USE ORDENES SHARED
oFr:SetWorkArea("Orders", 2)
SELECT 1
oFr:SetWorkArea(Alias(), Select(), .t.,{FR_RB_FIRST,FR_RE_COUNT,10})

```

**:SetFieldAliases()**

Establece un nombre simbólico para los campos de una tabla de datos y permite hacer visibles o no ciertos campos para generar un informe.

Así mismo hace visibles en el generador visual de informes, en la pestaña *DATOS*, los campos de una o varias tablas ya sea con un nombre simbólico o bien con el nombre real del campo.



Este método debe ejecutarse SIEMPRE antes de entrar en modo de diseño con el método `::DesignReport()` o antes de imprimirlo con el método `::ShowReport()`.

**Sintaxis:** `<oFR>:SetFieldAliases(<cAliasFr>, <cAliasCampos>)`

**Parámetros:**

<cAliasFr>	Cadena de caracteres con el nombre de la tabla de la cual queremos definir campos. Este nombre es establecido previamente con el método ::SetWorkArea().
<cAliasCampos>	<p>Es una cadena de caracteres con los nombres reales y simbólicos de los campos de la tabla, la cadena debe seguir la siguiente estructura:</p> <p>NombreRealCampo = NombreSimbolico;  Ó  NombrRealCampo;</p> <p>Se debe indicar esta estructura tantas veces como campos deseemos presentar en el diseñador visual, cada dupla va separada por un “;” y no es necesario que un elemento tenga siempre un nombre simbólico, pero si es importante indicar siempre el nombre real del campo.</p> <p>Si se omite un nombre real de campo, este no aparecerá en el diseñador visual y por consiguiente no podrá ser utilizado en el informe.</p> <p>Cuando en una dupla se indica nombre simbólico, es el nombre simbólico el que aparecerá en el diseñador visual, si no se indica nombre simbólico, aparecerá el nombre real.</p>

**Ejemplos:**

```
oFr:SetWorkArea("Customers", 1)
oFr:SetFieldAliases("Customers", "CUSTNO=Cust No;"+
"Company;ADDR1=Address;Phone;Fax;Contact")
```

**:RemoveDataSet()**

Elimina un área de trabajo de la lista de datos de FastReport.

**Sintaxis:** <oFR>:RemoveDataSet(<cAliasFr>)

**Parámetros:**

<cAliasFr>	Cadena de caracteres con un nombre que identificará a una tabla de datos en el informe FastReport. Este nombre se establece con el método ::SetWorkArea().
------------	--

**:ClearDataSets()**

Elimina todas las áreas de trabajo y todos los DataSets definidos por el usuario de la lista de datos de FastReport.

**Sintaxis:** <oFR>:ClearDataSets()

**Parámetros:** Ninguno

- *Reportes con varias tablas y maestros-detalle*

Este es el capítulo mas importante de esta documentación, entender los principios de cómo FastReport trabaja con las áreas de trabajo permitirá crear informes sumamente complejos.

Comúnmente, los informes mas complejos vinculan datos que pueden estar en distintas tablas y por ende en distintas áreas de trabajo, estas áreas de trabajo tienen dependencias entre ellas, y a esas dependencias se les llama *relaciones*.

Las relaciones en (x)Harbour se establecen usando el comando SET RELATION o la función DbSetRelation(), una relación entre 2 tablas, se establece a partir de un campo que tiene datos comunes en las tablas con las cuales se relaciona, dichos campos comunes se utilizan para que las tablas “dejen huellas” entre si, y que al mover el puntero en una tabla, provoque que también se muevan los punteros de las tablas relacionadas.

Ejemplo:

```
USE Items NEW
USE Parts NEW
SET INDEX TO Parts
SELECT Items
```

\*establecemos la relación entre PARTS e ITEMS usando el campo común ParNo  
DbSetRelation("Parts", {||PartNo})

```
// y ahora añadimos estas 2 áreas al informe FastReport. oFr:SetWorkArea("Items", Select("Items"))
oFr:SetWorkArea("Parts", Select("Parts"))
```

A nivel informe esto no genera un informe con campos relacionados. El método ::SetWorkArea() crea un objeto FastReport que opera cada área de trabajo de manera individual. Este objeto llama a funciones como DbGoTop(), DbSkip(), EOF(), etc. Para mejorar el rendimiento en la generación del informe, este objeto tiene una pequeña área de almacenamiento (cache) en el cual se almacena el registro actual del área de trabajo. Hemos agregado 2 objetos al informe, estos objetos son independientes al manejo que hace (x)Harbour de las tablas, vía el RDD y por lo tanto los objetos del informe no saben nada el uno del otro.

Para ejemplificar esto, supongamos que el primer objeto llama a la función DbSkip(), como esta función utiliza el mecanismo nativo de de (x)Harbour (RDD) para el manejo de tablas, la relación establecida provoca que el puntero en la segunda tabla se mueva, sin embargo, el objeto del informe que controla la segunda área, no se ha enterado que el registro ha cambiado de valor, porque en su cache tiene almacenado el valor original y no tiene forma de enterarse que se ha realizado un cambio en una tabla asignada al objeto.

FastReport provee de un mecanismo de notificación que permite informar al informe que dos o mas tablas tienen dependencia, para ello existe el método:

### **:SetResyncPair()**

Indica al informe FastReport que cuando cambie el puntero en el área de trabajo “padre”, se resincronice el cache del objeto que tiene asignada la tabla “hija”.

Un “padre” puede tener tantos “hijos” como necesite y un “hijo” puede tener tantos pares como sea requerido. Las referencias circulares no son soportadas. Es decir, un padre no puede ser al mismo tiempo hijo de sí mismo ni directa o indirectamente.

**Sintaxis:** <oFR>:SetResyncPair(<cAliasPadre>,<cAliasHijo>)

#### **Parámetros:**

<cAliasPadre>	Cadena de caracteres con el nombre del área de trabajo “padre” o principal. Este nombre tiene que ser el nombre que se establece con el método ::SetWorkArea().
<cAliasHijo>	Cadena de caracteres con el nombre del área de trabajo “hija” la cual debe de sincronizarse cuando cambie el puntero de la tabla “padre”. Este nombre tiene que ser el nombre que se establece con el método ::SetWorkArea().

De vuelta a nuestro ejemplo anterior, el siguiente código muestra los cambios necesarios para que el informe funcione adecuadamente:

```
USE Items NEW
USE Parts NEW
SET INDEX TO Parts
SELECT Items

DbSetRelation("Parts", {||PartNo})

oFr:SetWorkArea("Items", Select("Items"))
oFr:SetWorkArea("Parts", Select("Parts"))

oFr:SetResyncPair("Items", "Parts")
oFr:DesignReport() // for example
oFr:ClearResyncPair("Items", "Parts")
oFr:ClearDataSets()
```

### **:ClearResyncPair()**

Cancela la relación establecida por la llamada al método ::SetResyncPair().

**Sintaxis:** <oFR>:ClearResyncPair( <cAliasPadre>, <cAliasHijo> )

**Parámetros:**

<cAliasPadre>	Cadena de caracteres con el nombre en FastReport del área de trabajo padre de la relación.
<cAliasHijo>	Cadena de caracteres con el nombre en FastReport del área de trabajo hija.

### **:Resync()**

Resincroniza un área de trabajo de FastReport con la posición actual de los datos del programa (x)Harbour.

Esta función se debe utilizar si durante la construcción de un informe se hace una llamada a una función de (x)Harbour que altere la posición del puntero de una tabla.

**Sintaxis:** <oFR>:Resync( <cAlias> )

**Parámetros:**

<cAlias>	Cadena de caracteres con el nombre en FastReport del área de trabajo que se va a sincronizar.
----------	---

Ejemplo:

En FastReport:

```
procedure SomeBandOnAfterPrint(Sender: TfrxComponent);
begin
CallXppFunc('MyFunction');
end;
```

En (x)Harbour:

```
FUNCTION MyFunction()
SELECT Items
DbSkip(5)
oFr:Resync("Items")
RETURN NIL
```

### **:SetMasterDetail()**

Crea una relación maestro/detalle entre dos áreas de trabajo. Esto significa que cada vez que el registro en el área de trabajo maestra en FastReport cambia, este valor nuevo es usado para filtrar los registros en el área de trabajo detalle. El área de trabajo de detalle debe tener un índice activo.

Un maestro puede tener varios detalles, pero un detalle solo puede tener un maestro. No están permitidas las referencias circulares.

**Sintaxis:** <oFR>:SetMasterDetail(<cAliasMaestro>,<cAliasDeta>,<bScope>)

#### **Parámetros:**

<cAliasMaestro>	Cadena de caracteres con el nombre en FastReport del área de trabajo maestra.
<cAliasDeta>	Cadena de caracteres con el nombre en FastReport del área de trabajo detalle.
<bScope>	Bloque de código que debe retornar un valor que será usado por la función OrdScope() para filtrar el área de trabajo detalle

Ejemplo:

```
oFr:SetMasterDetail ("Clientes", "Orders",{|Clientes->Codigo})
```

### **:ClearMasterDetail()**

Cancela la relación entre dos áreas de trabajo establecida por la llamada al método ::SetMasterDetail().

**Sintaxis:** <oFR>:ClearMasterDetail( <cAliasDetalle> )

#### **Parámetros:**

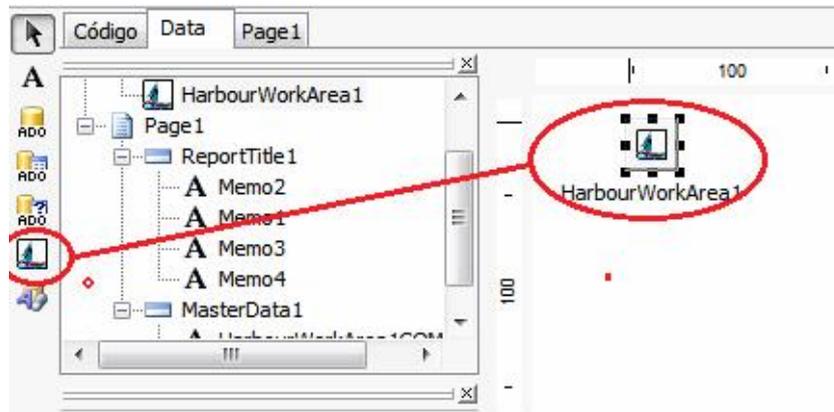
<cAliasDetalle>	Cadena de caracteres con el nombre en FastReport del área de trabajo detalle en una relación maestro/detalle..
-----------------	--

- *Áreas de trabajo desde el diseñador visual.*

Hasta este momento hemos venido trabajando con las áreas de trabajo desde código, sin embargo también es posible definir áreas de trabajo de manera visual, directamente desde el diseñador mediante los objetos TfxhHarbourWorkArea.

Estos objetos nos permiten crear diversos tipos de informes usando únicamente el ratón.

Para crear un objeto de este tipo, seleccionamos la pestaña DATA y escogemos el DataSet respectivo y lo arrastramos al cuerpo de informe:



Al arrastrar este objeto a diseño del informe hará aparecer su listado de propiedades en el inspector de propiedades:



Alias	Alias del área de trabajo. Contiene un combobox con una lista de todos los "alias" de todas las áreas de trabajo abiertas en este momento
AlreadyUse	Si es .T. indica que el área de trabajo esta abierta en este momento, si es .F. FastReport llamará a la función DbUseArea()
ClosedDataSource	No se utiliza
Description	Descripción del objeto
FieldAliases	Lista de los alias de los campos del área de trabajo. Similar al uso del método ::SetFieldAliases()
Filter	Condición de Filtro
Filtered	Indica si el área de trabajo se debe filtrar usando la condición establecida en la propiedad Filter
IsExclusive	Cuando la propiedad AlreadyUse está en .F. este valor es usado al llamar a la



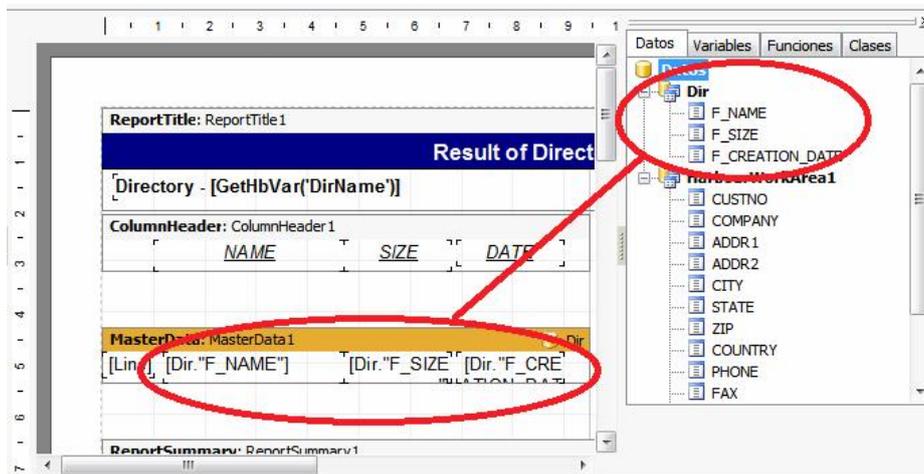
<bEOF>	Bloque de código que deberá retornar .T. cuando se haya alcanzado el final de los datos en el dataset.
<bObtenerValor>	Bloque de código que deberá retornar el dato a imprimir, recibe un parámetro: el nombre del campo definido en <aCampos> que se está procesando.

Ejemplo:

A continuación se presenta la forma de usar una matriz como DataSet definido por el usuario:

```
PRIVATE aDir, I := 1, DirName := "C:\*.*"
aDir := Directory(DirName)
oFr:SetUserDataSet("Dir", "F_NAME;F_SIZE;F_CREATION_DATE",;
    {||I := 1},;
    {||I := I + 1},;
    {||I := I - 1},;
    {||I > Len(aDir)},;
    { | cFieldz | xx:=IF(cField="F_NAME",1, IF(cField="F_SIZE",2,9)),;
    aDir[I, xx]})
```

Después de definir el DataSet definido por el usuario, en el diseñador visual veremos algo como esto:



FastReport no distingue entre datos reales de una base de datos y datos virtuales de un DataSet definido por el usuario. Para navegación en un área de trabajo FastReport usa las funciones usuales: DbGoTop(), DbSkip(), Eof(), etc. Para navegación en el DataSet definido por el usuario, FastReport ejecuta bloques de código definidos por el programador .

Para eliminar los DataSets definidos por el usuario usamos los métodos ::RemoveDataSet() y ::ClearDataSet().

## 13.5 Opciones de Impresión

El conjunto de los métodos, los cuales son responsables del ajuste de la impresión se localizan en una clase separada, llamada **frPrintOptions**.

Esta clase se instancia al momento de crear la clase frReportManager y se puede manipular a través de una propiedad de la clase principal llamada PrintOptions.

### **:SetCopies()**

Establece el número de copias que se debe imprimir del informe:

**Sintaxis:** <oFR>:PrintOptions:SetCopies([<nCopias>]) -> <nValorAnt>

#### **Parámetros**

<nCopias>	Valor numérico que indica el número de copias a imprimir del mismo informe, valor por omisión: 1
-----------	--

#### **Devuelve:**

<nValorAnt>	Valor anterior establecido para el número de copias.
-------------	--

### **:SetCollate()**

Define si se deben de agrupar en orden las copias del informe.

**Sintaxis:** <oFR>:PrintOptions:SetCollate([<IAgrupar>]) -> <IValorAnt>

#### **Parámetros**

<nCopias>	Valor lógico opcional. Si es .T. (por omisión) indica que las copias se deben ordenar.
-----------	--

#### **Devuelve:**

<IValorAnt>	Valor lógico del estado anterior del método.
-------------	--

### **:SetPageNumbers()**

Establece las hojas del informe que se desean imprimir.

**Sintaxis:** <oFR>:PrintOptions:SetPageNumbers([<cPags>]) -> <cValorAnt>

#### **Parámetros**

<cPags>	Parámetro opcional. Cadena de caracteres con la lista de hojas que se desean imprimir del informe, por ejemplo:  "1,3,5-12,17-"  Por omisión: "", es decir, todas las hojas.
---------	--

**Devuelve:**

<cValorAnt>	Cadena de caracteres con la lista establecida anteriormente.
-------------	--

**:SetPrinter()**

Indica la impresora a la cual se desea enviar a imprimir un informe.

**Sintaxis:** <oFR>:PrintOptions:SetPrinter( [<cImpr>] ) -> <cValorAnt>

**Parámetros**

<cImpr>	Parámetro opcional. Cadena de caracteres con el nombre de la impresora a la cual deseamos enviar a imprimir un informe. Este nombre es el que aparece en el listado de impresoras de Windows. Por omisión: la impresora definida como predeterminada.
---------	---

**Devuelve:**

<cValorAnt>	Cadena de caracteres con el nombre de la impresora definida anteriormente.
-------------	--

**:SetPrintPages()**

Establece que grupo de páginas se envían a imprimir.

**Sintaxis:** <oFR>:PrintOptions:SetPrintPage( [<nPags>] ) -> <nValorAnt>

**Parámetros**

<nPags>	Parámetro opcional. Valor numérico indicando el grupo de páginas que se van a imprimir, los valores posibles son:  FR_PP_ALL – Todas las páginas FR_PP_ODD – Páginas nones FR_PP_EVEN – Páginas pares  Por omisión: FR_PP_ALL
---------	---

**Devuelve:**

<nValorAnt>	Valor numérico con el grupo anterior de páginas a imprimir.
-------------	---

**:SetShowDialog()**

Indica si se debe de mostrar el dialogo de impresión antes de imprimir el informe.

**Sintaxis:** <oFR>:PrintOptions:SetShowDialog([<lMostrar>]) -> <lValorAnt>

**Parámetros**

<lMostrar>	Parámetro opcional. Valor lógico, cuando es .T. indica que el diálogo de impresión debe ser mostrado.
------------	---

**Devuelve:**

<lValorAnt>	Valor lógico con el valor anterior del método.
-------------	--

**:SetReverse()**

Indica si el informe debe imprimirse comenzando por la última hoja.

**Sintaxis:** <oFR>:PrintOptions:SetReverse( [<lAtras>] ) -> <lValorAnt>

**Parámetros**

<lAtras>	Parámetro opcional. Valor lógico cuando es .T. indica que el informe deberá comenzar a imprimirse por la última hoja. Por omisión: .F.
----------	--

**Devuelve:**

<lAtras>	Valor lógico con el estado anterior del método.
----------	---

**:ClearOptions()**

Elimina todos los valores establecidos y los devuelve a sus valores predeterminados.

**Sintaxis:** <oFR>:PrintOptions:ClearOptions()

**Parámetros:** Ninguno

## 13.6 Opciones del diseñador visual

### **:SetIcon()**

Establece el icono que deberá mostrarse en las ventanas de FastReport, diseñador y motor de vista previa.

**Sintaxis:** <oFR>:SetIcon( <xValor> )

**Parámetros:**

<xValor>	Cadena de caracteres ó valor numérico que identifica al icono dentro de la lista de recursos. Si se especifica un icono este aparecerá del lado superior izquierdo de la ventana . Solo es posible utilizar recursos ya sea integrados al EXE o en una DLL, no archivos .ICO por separado
----------	--

### **:SetTitle()**

Establece el texto que deberá mostrarse en el título de la ventana y en la barra de tareas cuando se minimiza la aplicación.

**Sintaxis:** <oFR>:SetTitle( <cTitulo> )

**Parámetros:**

<cTitulo>	Cadena de caracteres con el título de la ventana.
-----------	---

### **:LoadLangRes()**

Carga el archivo con el idioma para mostrar en la interfaz tanto del diseñador visual, como del motor de vista previa y de los diálogos de configuración de FastReport.

FastReport soporta 32 idiomas distintos para su interface.

**Sintaxis:** <oFR>:LoadLangRes( <cArchivo> )

**Parámetros:**

<cArchivo>	Cadena de caracteres con el nombre del archivo de idioma a utilizar, puede incluir la unidad de disco y la ruta del directorio donde se encuentre el archivo.
------------	---

## 13.7 Opciones de la Vista Previa

El conjunto de los métodos, los cuales son responsables del ajuste de la ventana de vista previa se localizan en una clase separada, llamada **frPreviewOptions**.

Esta clase se instancia al momento de crear la clase frReportManager y se puede manipular a través de una propiedad de la clase principal llamada PreviewOptions.

### **:SetAllowEdit()**

Activa o desactiva la capacidad de editar los datos de un informe después de haberlo generado.

**Sintaxis:** <oFr>:PrintOptions:SetAllowEdit( <lEditar> )

#### **Parámetros**

<lEditar>	Valor lógico que indica si los datos presentados en el informe se pueden editar, por omisión: .T.
-----------	---

### **:SetButtons()**

Permite definir que botones deben de aparecer en la barra de botones de la ventana de vista previa.

**Sintaxis:** <oFr>:PrintOptions:SetButtons( <nBotones> )

#### **Parámetros**

<nBotones>	<p>Valor numérico que es la suma aritmética del valor de cada botón que queremos mostrar. Los valores posibles son:</p> <pre> FR_PB_PRINT 1 FR_PB_LOAD 2 FR_PB_SAVE 4 FR_PB_EXPORT 8 FR_PB_ZOOM 16 FR_PB_FIND 32 FR_PB_OUTLINE 64 FR_PB_PAGESETUP 128 FR_PB_TOOLS 256 FR_PB_EDIT 512 FR_PB_NAVIGATOR 1024 </pre> <p>Por omisión: Todos los botones (2047)</p>
------------	---

### **:SetDoubleBuffered()**

Activa el pintado de doble buffer para la ventana de vista previa. Esto provoca que al momento de pintarse, la ventana no parpadee, sin embargo la velocidad de procesamiento del informe se ve reducida.

**Sintaxis:** <oFr>:PrintOptions:SetDoubleBuffered( <lDoble> )

#### **Parámetros**

<lDoble>	Valor lógico, por omisión .T. indica que se desea activar el pintado de doble
----------	---

	buffer.
--	---------

### **:SetMaximized()**

Define si la ventana de vista previa debe presentarse maximizada.

**Sintaxis:** <oFr>:PrintOptions:SetMaximized( <IMax> )

#### **Parámetros**

<IMax>	Valor lógico, .T. la ventana debe presentarse maximizada., .F. la ventana debe presentarse normal.
--------	--

### **:PreviewOptions()**

Establece la posición y el tamaño de la ventana de vista previa.

**Sintaxis:** <oFr>:PrintOptions:PreviewOptions( <nIzq>, <nArr>, <nAncho>, <nAlto> )

#### **Parámetros**

<nIzq> <nArr> <nAncho> <nAlto>	Valores numéricos que indican la posición y tamaño de la ventana. Los valores se deben indicar en el sistema de coordenadas estándar de Windows.
---	--

### **:SetOutlineVisible()**

Establece si el panel de presentación de resumen de datos del informe se debe mostrar.

**Sintaxis:** <oFr>:PrintOptions:SetOutlineVisible( <IVisible> )

#### **Parámetros**

<IVisible>	Valor lógico. .T. indica que se debe mostrar el panel con el resumen de datos. Por omisión: .F.
------------	---

### **:SetOutlineExpand()**

Indica si el árbol con el resumen de datos del informe debe aparecer expandido.

**Sintaxis:** <oFr>:PrintOptions:SetOutlineExpand( <IExpandido> )

#### **Parámetros**

<IExpandido>	Valores lógico, cuando es .T. el árbol del resumen aparece expandido.
--------------	---

### **:SetOutlineWidth()**

Establece el ancho en pixeles del panel con el resumen de los datos del informe.

**Sintaxis:** <oFr>:PrintOptions:SetOutlineWidth( <nAncho> )

**Parámetros**

<nAncho>	Valores numéricos que indica el ancho del panel del resumen de datos en la posición y tamaño de la ventana.
----------	---

**:SetShowCaptions()**

Define si se deben mostrar los botones de la barra de herramientas con texto. Si se usa esta propiedad será necesario limitar el número de botones en la barra ya que es posible que no todos los botones cuenten con espacio suficiente.

**Sintaxis:** <oFr>:PrintOptions:SetShowCaptions( <ITexto> )

**Parámetros**

<ITexto>	Valor lógico que indica si se deben de mostrar texto en los botones de la barra de herramientas.
----------	--

**:SetZoom()**

Establece el valor por omisión de zoom para el informe.

**Sintaxis:** <oFr>:PrintOptions:SetZoom( <nZoom> ) -> <nValAnt>

**Parámetros**

<nZoom>	Valor numérico con el valor de zoom para el informe. Por omisión 1.0.
---------	---

**Devuelve**

<nValAnt>	Valor numérico con el valor anterior del factor de zoom del informe.
-----------	--

**:SetZoomMode()**

Establece la modalidad de zoom con la cual queremos presentar el informe.

**Sintaxis:** <oFr>:PrintOptions:SetZoomMode( <nModoZoom> ) -> <nValAnt>

**Parámetros**

<nModoZoom>	<p>Valor numérico con la modalidad de zoom que queremos usar. Los valores posibles:</p> <p>FR_ZM_DEFAULT 0  FR_ZM_WHOLEPAGE 1  FR_ZM_PAGEWIDTH 2  FR_ZM_MANYPAGES 3</p> <p>Por omisión: 0.</p>
-------------	--

**Devuelve**

<nValAnt>	Valor numérico con el valor anterior de la modalidad de zoom del informe.
-----------	---

**:SetPictureCacheInFile()**

Establece si se deben de almacenar en memoria temporalmente las imágenes para el informe.

**Sintaxis:** <oFr>:PrintOptions:SetPictureCacheInFile( <lAlmac> )

**Parámetros**

<lAlmac>	Valor lógico. Cuando es .T. se almacenarán en memoria las imágenes.
----------	---

**Opciones del informe actualmente en uso**

El conjunto de los métodos, los cuales son responsables del ajuste de informe actualmente en uso se localizan en una clase separada, llamada **frReportOptions**.

Esta clase se instancia al momento de crear la clase frReportManager y se puede manipular a través de una propiedad de la clase principal llamada ReportOptions.

**:SetAuthor()**

Establece el nombre del autor del informe.

**Sintaxis:** <oFR>:ReportOptions:SetAuthor( [<cAutor>] ) -> <cValorAnt>

**Parámetros**

<cAutor>	Parámetro opcional. Cadena de caracteres con el nombre del autor del informe.
----------	---

**Devuelve:**

<cValorAnt>	Cadena de caracteres con el valor anterior del método.
-------------	--

**:SetCompressed()**

Define si el informe debe comprimirse en ZIP.

**Sintaxis:** <oFR>:ReportOptions:SetCompressed( [<lComp>] ) -> <lValorAnt>

**Parámetros**

<lComp>	Parámetro opcional. Valor lógico que cuando es .T. indica que el informe debe comprimirse.
---------	--

**Devuelve:**

<lValorAnt>	Valor lógico con el valor anterior del método.
-------------	--

**:SetCreateDate()**

Establece la fecha y la hora de creación del informe.

**Sintaxis:** <oFR>:ReportOptions:SetCreateDate( [<cFech>] ) -> <cValorAnt>

**Parámetros**

<cFech>	Parámetro opcional. Cadena de caracteres con la fecha y hora de creación del informe en formato de tipo de dato DATETIME:  “MM/DD/AA HH:MM:SS”  Opcionalmente puede indicar AM o PM para la hora.
---------	---

**Devuelve:**

<cValorAnt>	Cadena de caracteres con el valor anterior del método.
-------------	--

**:SetLastChange()**

Establece la fecha y la hora del último cambio al informe.

**Sintaxis:** <oFR>:ReportOptions:SetLastChange( [<cFech>] ) -> <cValorAnt>

**Parámetros**

<cFech>	Parámetro opcional. Cadena de caracteres con la fecha y hora de creación del informe en formato de tipo de dato DATETIME:  “MM/DD/AA HH:MM:SS”  Opcionalmente puede indicar AM o PM para la hora.
---------	---

**Devuelve:**

<cValorAnt>	Cadena de caracteres con el valor anterior del método.
-------------	--

**:SetDescription()**

Crea una descripción para el informe.

**Sintaxis:** <oFR>:ReportOptions:SetDescription( [<cDesc>] )-> <cValorAnt>

**Parámetros**

<cDesc>	Parámetro opcional. Cadena de caracteres con la descripción del informe.
---------	--

**Devuelve:**

<cValorAnt>	Cadena de caracteres con el valor anterior del método.
-------------	--

**:SetInitString()**

Establece una cadena de caracteres con los códigos de escape para inicializar impresoras de matriz de puntos.

**Sintaxis:** <oFR>:ReportOptions:InitString( [<cInit>] ) -> <cValorAnt>

**Parámetros**

<cInit>	Parámetro opcional. Cadena de caracteres con los códigos de escape para inicializar impresoras de matriz de puntos.
---------	---

**Devuelve:**

<cValorAnt>	Cadena de caracteres con el valor anterior del método.
-------------	--

**:SetName()**

Establece el nombre del informe.

**Sintaxis:** <oFR>:ReportOptions:SetName( [<cNombre>] ) -> <cValorAnt>

**Parámetros**

<cNombrer>	Parámetro opcional. Cadena de caracteres con el nombre del informe.
------------	---

**Devuelve:**

<cValorAnt>	Cadena de caracteres con el valor anterior del método.
-------------	--

**:SetPassword()**

Establece una clave de acceso para el informe, la cual se solicitará al momento de usar el método ::LoadFromFile().

**Sintaxis:** <oFR>:ReportOptions:SetPassword( [<cCve>] ) -> <cValorAnt>

**Parámetros**

<cCve>	Parámetro opcional. Cadena de caracteres con la clave de acceso al informe, el nombre del autor del informe.
--------	--

**Devuelve:**

<cValorAnt>	Cadena de caracteres con el valor anterior del método.
-------------	--

**:SetVersion()**

Establece la versión del informe.

**Sintaxis:** <oFR>:ReportOptions:SetVersion( [<aVersion>] ) -> <cValorAnt>

**Parámetros**

<aVersion>	Parámetro opcional. Un vector de 4 elementos cadenas de caracteres que son: {<cVerMayor>,<cVerMenor>,<cVerLiberada>,<cVerCompilacion>}
------------	---

**Devuelve:**

<aValorAnt>	Vector de 4 posiciones con el valor anterior del método.
-------------	--

## 13.8 Otros métodos de la clase frReportManager

### **:LoadStyleSheet()**

Carga una hoja de estilos para el informe actualmente en uso.

**Sintaxis:** <oFR>:LoadStyleSheet( <cNomArchivo> )

**Parámetros:**

<cNomArchivo>	Cadena de caracteres con el nombre del archivo que contiene los estilos a aplicar al informe. Puede incluir la unidad de disco y la ruta de directorio donde se encuentra el archivo.
---------------	---

### **:GetErrors()**

Devuelve la lista de los errores generados durante el proceso de generación de un informe.

**Sintaxis:** <oFR>:GetErrors() -> <cErrors>

**Parámetros:** Ninguno

**Devuelve**

<cErrors>	Cadena de caracteres con la lista de los errores generados, cada línea está separada por chr(13)+chr(10).
-----------	---



**Sintaxis:** <oFR>:MemoAt( <cTexto>, <nIzq>, <nArriba>, <nAncho>,,  
<nAlto> ) -> <cMemo>

**Parámetros:**

<cTexto>	Texto a dibujar en el cuerpo del informe.
<nIzq> <nArriba> <nAncho> <nAlto>	Coordenadas donde se va a colocar el texto en las unidades de medida establecidas en el método ::StartManualBuild().

**Devuelve:**

<cMemo>	Cadena de caracteres con el nombre de un objeto memo que posteriormente se podrá manipular usando el método ::SetManualObjProperty().
---------	---

Ejemplo:

```
tmp_Name := :MemoAt("Some memo with bottom frame...",30,30,300,50)
```

### **:LineAt()**

Dibuja un objeto línea en unas coordenadas dadas.

**Sintaxis:** <oFR>:LineAt( <nIzq>, <nArriba>, <nAncho>,,  
<nAlto> ) -> <cLinea>

**Parámetros:**

<nIzq> <nArriba> <nAncho> <nAlto>	Coordenadas donde se va a colocar el texto en las unidades de medida establecidas en el método ::StartManualBuild().
--	--

**Devuelve:**

<cLinea>	Cadena de caracteres con el nombre de un objeto línea que posteriormente se podrá manipular usando el método ::SetManualObjProperty().
----------	--

Ejemplo:

```
oFr:LineAt(30, 200, 100, 100)
```

### **:PictureAt()**

Dibuja un objeto imagen en unas coordenadas dadas.

**Sintaxis:** <oFR>:PictureAt( <cImagen>, <nIzq>, <nArriba>, <nAncho>,,  
<nAlto> ) -> <cImagen>

**Parámetros:**

<cImagen>	Cadena de caracteres con el nombre del archivo que contiene la imagen a mostrar en el informe. Puede incluir la unidad de disco y la ruta al directorio donde se encuentra el archivo.
<nIzq>	Coordenadas donde se va a colocar la imagen en las unidades de medida

<nArriba> <nAncho> <nAlto>	establecidas en el método ::StartManualBuild().
----------------------------------	---

**Devuelve:**

<cImagen>	Cadena de caracteres con el nombre de un objeto imagen que posteriormente se podrá manipular usando el método ::SetManualObjProperty().
-----------	---

Ejemplo:

```
oFr:PictureAt(DataDir + "logo.bmp" , 30, 400, 300, 300)
```

**:NewPage()**

Agrega una página nueva al informe actual.

**Sintaxis:** <oFR>:NewPage()

**Parámetros:** Ninguno

Ejemplo:

```
oFr:NewPage()
```

**:SetDefaultFontProperty()**

Un informe generado manualmente solo puede utilizar un tipo de letra (Font) al mismo tiempo, este método se encarga de establecer las características de dicho tipo de letra.

**Sintaxis:** <oFR>:SetDefaultFontProperty( <cProp>, <xValor> )

**Parámetros:**

<cProp>	Cadena de caracteres con el nombre de la propiedad del tipo de letra que queremos modificar.
<xValor>	Valor (de cualquier tipo) que debe tomar la propiedad del Font.

Ejemplo:

```
oFr:SetDefaultFontProperty("Name", "Times New Roman")
oFr:SetDefaultFontProperty("Size", 16)
```

**:SetManualObjProperty()**

Establece un nuevo valor para una propiedad de un objeto previamente creado con los métodos ::MemoAt(), ::LineAt(), ::PictureAt().

**Sintaxis:** <oFR>:SetManualObjProperty( <cNombre>, <cProp>, <xValor> )

**Parámetros:**

<cNombre>	Cadena de caracteres con el nombre del objeto del cual queremos modificar su propiedad.
-----------	---

<cProp>	Cadena de caracteres con el nombre de la propiedad que queremos modificar.
<xValor>	Valor (de cualquier tipo) que debe asignarse a la propiedad.

Ejemplo:

```
oFr:SetManualObjProperty(tmp_Name+".Frame", "Typ", "[ftBottom]")
```

## 13.10 Opciones del motor de impresión

El conjunto de los métodos, los cuales son responsables del ajuste del motor de impresión se localizan en una clase separada, llamada **frEngineOptions**.

Esta clase se instancia al momento de crear la clase frReportManager y se puede manipular a través de una propiedad de la clase principal llamada EngineOptions.

### :SetConvertNulls()

Se utiliza para imprimir valores NULL de una tabla de datos. Activando esta opción, los valores NULL serán convertidos a 0, .F. o "" dependiendo del tipo de campo.

**Sintaxis:** <oFR>:EngineOptions:SetConvertNulls( [<IConvert>] )

#### Parámetros

<IConvert>	Parámetro opcional. Valor lógico, cuando es .T. se realiza la conversión de los valores NULL, por omisión .T..
------------	--

### :SetDoublePass()

Activa la opción doble pasada para el informe.

**Sintaxis:** <oFR>:EngineOptions:SetDoublePass( [<IDoble>] )

#### Parámetros

<IDoble>	Parámetro opcional. Valor lógico, cuando es .T. indica que el informe debe realizar doble pasada de impresión, por omisión .F..
----------	---

### :SetPrintIfEmpty()

Define si el informe se imprime en blanco, es decir, solo el diseño del informe, pero sin datos.

**Sintaxis:** <oFR>:EngineOptions:SetPrintIfEmpty( <IImprime> )

#### Parámetros

<IImprime>	Valor lógico, cuando es .T. el informe se imprime en blanco.
------------	--

### :SetSilentMode()

Activa el "modo silencioso" (silent mode). Cuando esta funcionalidad se activa no se presentan mensajes de error en pantalla, los mensajes de error se almacenarán en memoria y se pueden recuperar posteriormente con el método ::GetErrors().

**Sintaxis:** <oFR>:EngineOptions:SetSilentMode( <ISilencioso> )

#### Parámetros

<ISilencioso>	Valor lógico, cuando es .T. se activa el modo silencioso del informe.
---------------	---

## :SetNewSilentMode()

Es una variante mas poderosa del modo silencioso ya que permite definir que mensajes de error se deben omitir

**Sintaxis:** <oFR>:EngineOptions:SetNewSilentMode( <nModo> )

### Parámetros

<nModo>	Valor numérico que indica los mensajes a omitir. Los valores posibles son: FR_SIM_MESSAGEBOXES (0) FR_SIM_RETHROW (1) FR_SIM_SILENT (2)
---------	--

## :SetMaxMemSize()

Establece el tamaño máximo en MBytes que se deben reservar en memoria para almacenamiento del informe (cache).

Es particularmente útil cuando se utiliza el método :UseFileCache().

Si un informe comienza a ocupar mas memoria de la asignada durante su construcción se realiza un almacenamiento temporal a un archivo.

Esta propiedad es inexacta y solo determina un limite aproximado de memoria.

**Sintaxis:** <oFR>:EngineOptions:SetMaxMemSize( <nTam> )

### Parámetros

<nTam>	Valor numérico con el tamaño en Megabytes a reservar en memoria. Por omisión: 10.
--------	---

## :SetTempDir()

Define la ubicación de los archivos temporales de FastReport.

**Sintaxis:** <oFR>:EngineOptions:SetTempDir( <cDir> ) -> <cDirAnt>

### Parámetros

<cDir>	Cadena de caracteres con la ruta donde se deben de almacenar los archivos temporales generados por FastReport.
--------	--

### Devuelve

<cDirAnt>	Cadena de caracteres con el valor de la ruta anterior.
-----------	--

**:SetUseFileCache()**

Define si el almacenamiento temporal (cache) de los informes generados debe realizarse a un archivo.

**Sintaxis:** <oFR>:EngineOptions:SetUseFileCache( <IUsar> )

**Parámetros**

<IUsar>	Valor lógico que indica si se tiene que usar almacenamiento temporal en archivos.
---------	---

## 13.11 Métodos universales del objeto frReportManager.

Dado el alto nivel de complejidad de FastReport, con todos sus objetos, métodos y propiedades, crear métodos específicos para cada objeto aumentaría la complejidad en su aprendizaje, en vez de eso, se han creado una serie de métodos universales que pueden utilizarse sobre cualquier objeto de FastReport.

### Controladores de eventos

#### :SetEventHandler()

Los eventos de de todos lo objetos de FastReport de controlan usando este método.

**Sintaxis:** <ofr>:SetEventHandler( <cNomObj>, <cNomEven>, [<bControl>] )

#### Parámetros

<cNomObj>	Cadena de caracteres con el nombre del objeto FastReport.
<cNomEven>	Cadena de caracteres con el nombre del evento del objeto que se desea controla
<bControl>	Parámetro opcional, es un bloque de código que se ejecutará cuando ocurra el evento, por omisión {   NIL}

A continuación se muestran los eventos soportados por los distintos objetos de FastReport.

Para entender la sintaxis de los eventos veamos el siguiente ejemplo:

Supongamos que tenemos:

MiEvento (<parámetro>) para el objeto "REPORT"

Para definir su controlador haremos:

```
ofr:SetEventHander("REPORT","MIEVENTO",{||parámetro| ....}
```

Listado de los eventos soportados por el objeto "REPORT":

Evento	Parámetros	Se ejecuta...
OnAfterPrint(<cNomObj>)	<cNomObj> Nombre del objeto informe	Cuando se inicia el informe y después de que se crean todos los objetos.
OnAfterPrintReport()		Después de la impresión del informe
OnBeforeConnect(<cConObj>)	<cConObj> Nombre del objeto conexión	Antes de conectar con un origen de datos externo (ADO, IBX, DBX, etc).
OnBeforePrint(<cNomObj>)		Cuando se inicia el informe y antes de que se creen todos los objetos
OnBeginDoc()		Antes de la construcción del informe
OnEndDoc()		Después de la construcción del informe
OnClickObject(<cNomObj>; <nBoton>)	<cNomObj> Nombre del objeto informe <nBoton> Botón del ratón: 0 izquierdo 1 derecho 2 medio	Cuando se hace click sobre un objeto durante la presentación de un informe en la ventana de vista previa
OnGetValue(<cNomVar>)	<cNomVar>	Cuando inicia un informe y solo

	Nombre de variable	si se encuentra una variable desconocida. El bloque de código de control deberá retornar un valor para esta variable.
OnPreview()		Antes de que se muestre la ventana de vista previa.
OnPrintPage(<nNumCopia>)	<nNumCopia> Número de copia	Antes de que se imprima una hoja de informe.
OnPrintReport()		Antes de la impresión del informe
OnProgress(<nActividad>; <nProgreso>)	<nActividad> Tipo de actividad: 0 Construyendo 1 Exportando 2 Imprimiendo <nProgreso> Número de la página en proceso.	Sirve para indicar el progreso de la actividad que se está ejecutando. Se puede utilizar para mostrar nuestro propio mensaje de progreso.
OnProgressStop(<nActividad>; <nProgreso>)	Mismos parámetros que OnProgress	Cuando se detiene la actividad
onProgressStart(<nActividad>; <nProgreso>)	Mismos parámetros que OnProgress	Cuando la actividad se inicia
OnUserFunction(<cNomFuncion>; <aParams>)	<cNomFuncion> Nombre de la función <aParams> Array con lista de parámetros.	Cuando se llama a una función definida usando el método ::AddFunction()

Listado de eventos soportados por le objeto "Designer":

<b>Evento</b>	<b>Parámetros</b>	<b>Se ejecuta...</b>
OnInsertObject()		Cuando un objeto nuevo es insertado en diseño del informe
OnLoadReport()		Cuando se carga un informe. Nos permite controlar la carga de los informes. El bloque de código controlador deber retornar .T. si el informe pudo ser cargado
OnSaveReport(<lGuardarComo>)	<lGuardarComo> Es .T. si se ha seleccionado la opción de Guardar como... del menú del diseñador	Cuando se guarda el informe. Nos permite controlar el proceso de guardado. El bloque de código controlador debe retornar .T. si el informe pudo ser guardado
OnShow()		Cuando el diseñador se inicia
OnShowStartUpScreen()		Cuando el diseñador se muestra.

- *Asignación y recuperación de valores de Propiedades (GET / SET)*

## :GetProperty()

Método universal que devuelve el valor de cualquier propiedad del objeto FastReport.

**Sintaxis:** <oFR>.GetProperty( <cNomObj>, <cNomProp> ) -> <xValor>

### Parámetros

<cNomObj>	Cadena de caracteres con el nombre del objeto. Si el nombre tiene un objeto anidado, se usa "." Como separador ("Designer.DefaultFont").
<cNomProp>	Cadena de caracteres con el nombre de la propiedad de la cual queremos obtener su valor.

### Devuelve

<xValor>	Valor de cualquier tipo asignado a la propiedad.
----------	--

## :SetProperty()

Método universal para asignar un valor a cualquier propiedad de FastReport.

**Sintaxis:** <oFR>.SetProperty( <cNomObj>, <cNomProp>, <xValor>)

### Parámetros

<cNomObj>	Cadena de caracteres con el nombre del objeto. Si el nombre tiene un objeto anidado, se usa "." Como separador ("Designer.DefaultFont").
<cNomProp>	Cadena de caracteres con el nombre de la propiedad de la cual queremos obtener su valor.
<xValor>	Valor de cualquier tipo asignado a la propiedad

Compatibilidad de tipos de datos:

FastReport	(xHarbour)	oFr:SetProperty...
Cadena de caracteres	Cadena de caracteres	("Report", "ScriptLanguage",; "C++Script")
Cualquier tipo numérico	Numérico	("Designer", "DefaultLeftMargin",; 15)
Lógica	Lógica	("PDFExport", "ShowDialog", .f.)
Enumerated	Cadena de caracteres	("RTFExport", "HeaderFooterMode",; "hfText")
Set	Cadena de caracteres, delimitada por [] y cada elemento separado por ";"	("Designer", "Restrictions",; "[drDontCreateReport,; drDontLoadReport,; drDontSaveReport]")
TString	Cadena de caracteres en formato texto separado por CHR(13)+CHR(10)	(MailExport.Lines, "Text",; "Hello!" + Chr(13)+ Chr(10) + "Look attachment!")

- *Objeto "REPORT"*

Propiedad	Tipo	Descripción
VERSION	C	Versión de FastReport. Solo lectura
DotMatrixReport	L	Indica si un informe es para impresora de matriz de puntos. Si esta propiedad es .T. el informe puede contener páginas y objetos para matriz de puntos. No use esta propiedad directamente, utilice la opción "Archivo   Nuevo" del menú del diseñador para crear informes para matriz de puntos.

IniFile	C	Nombre del archivo o de la llave del registro donde se almacena el estado de FastReport. Por omisión: “\Software\Fast Reports for [x]Harbour”
OldStyleProgress	L	Muestra el progreso de los informes como en la versión 2.xx de FastReport (en una ventana separada). Por omisión .F.
ScriptLanguage	C	Nombre del lenguaje script para usar en el informe, los valores posibles son: “PascalScript”, “C++Script”, “BasicScript”, “Jscript”
ScriptText	C	Código del script a ser utilizado en el informe
ShowProgress	L	Indica si se deben de mostrar los diálogos de progreso cuando se genera el informe
OnStartReport	C	Nombre del script que debe ejecutarse cuando el informe se inicia
OnStopReport	C	Nombre del script que debe ejecutarse cuando el informe termine.
OnRunDialogs	C	Nombre del script que debe ejecutarse cuando se muestre algún diálogo.

- **Objeto “DESIGNER”**

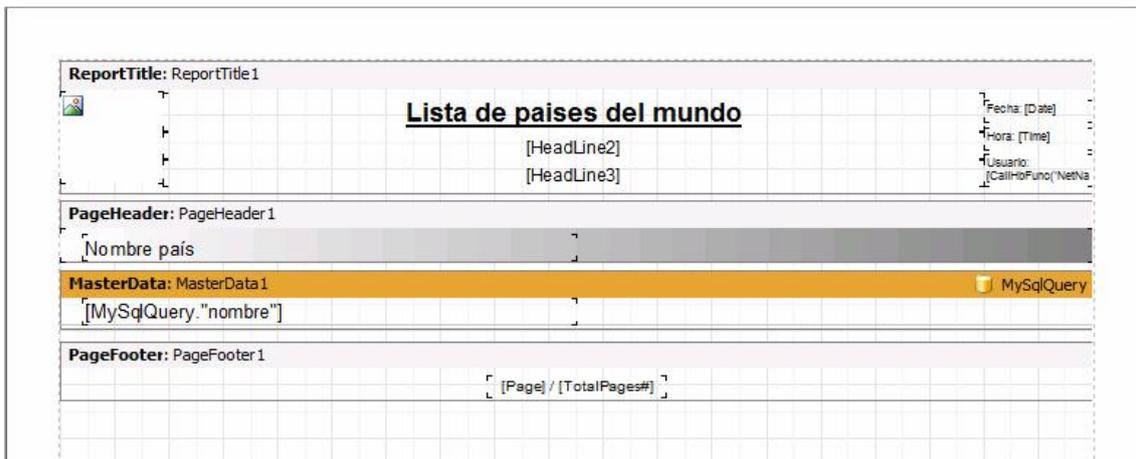
Propiedad	Tipo	Descripción
CloseQuery	L	Sirve para activar la pregunta de guardar el informe cuando se cierra el diseñador
DefaultScriptLanguage	C	Lenguaje script usado por omisión en el diseñador, los valores validos son: PascalScript, C++Script, BasicScript, JScript
DefaultFont	Objeto	Tipo de letra predeterminado para usar en los objetos texto del diseñador. Esta propiedad nunca se utiliza sola, se deben acceder a sus sub-propiedades para configurarla.
DefaultFont.Charset	N	Identifica el juego de caracteres del tipo de letra. Cada tipo (especificado por la propiedad Name) soporta uno o mas juegos de caracteres. Refiérase a la documentación del Font para determinar cuales juegos de caracteres son válidos
DefaultFont.Color	C	Color de los caracteres de texto expresado en hexadecimal. (\$00FF0000) para azul por ejemplo
DefaultFont.Height	N	Altura del tipo de letra, en pixeles
DefaultFont.Name	C	Nombre del tipo de letra
DefaultFont.Pitch	C	Especifica si los caracteres en el tipo de letra tienen todos el mismo ancho, su valor es: (fpDefault, fpVariable, fpFixed)
DefaultFont.Size	N	Altura del tipo de letra, en puntos.
DefaultFont.Style	C	Añade características especiales a usar con el tipo de letra, su valor es: [fsBold, fsItalic, fsUnderline, fsStrikeOut]
DefaultLeftMargin	N	Margen izquierdo por omisión
DefaultRightMargin	N	Margen derecho por omisión
DefaultTopMargin	N	Margen superior por omisión
DefaultBottomMargin	N	Margen inferior por omisión
DefaultPaperSize	N	Tamaño de la hoja de papel por omisión
DefaultOrientation	C	Orientación de la hoja de papel, valores posibles: poPortrati poLandscape
OpenDir	C	Nombre de una carpeta desde donde deben de abrirse los archivos por omisión
SaveDir	C	Nombre de una carpeta donde se guardarán los archivos por omisión
Restrictions	C	Sirve para indicar que operaciones deben de ser prohibidas en el diseñador, sus valores posibles son: “[drDontInsertObject, drDontDeletePage, drDontCreatePage, drDontChangePageOptions, drDontCreateReport, drDontLoadReport, drDontSaveReport, drDontPreviewReport, drDontEditVariables, drDontChangeReportOptions, drDontEditReportData]”

## Tema 14. Ejemplos de informes hechos con FastReport.

### Informe columnar simple

Vamos a empezar con los ejemplos viendo el informe más simple posible pero con algunos añadidos.

Cualquier listado estándar tendrá básicamente una serie de bandas mínima como:



ReportTitle, donde podemos poner un icono a la izquierda, un título en el centro, algún dato más a la derecha, que solo saldrá en la primera página.

PageHeader, para los títulos de las columnas, aparecerá en cada página.

Masterdata para los datos de los registros a imprimir, aparece una línea por cada registro de la tabla o consulta.

PageFooter, para poner el nº de página, aparece el pie de cada página.

Si nos fijamos en el ReportTitle veremos que a su izquierda hay un cuadro vacío.

Es un cuadro de imagen TFrxFigureview  para el logo, cuyo contenido se puede poner de tres maneras. Se podría haber puesto en el diseño directamente y se vería, pero en este caso se carga del disco en ejecución ya que hemos indicado su propiedad

`FileLink` `.\Reports\ReportLogo.jpg` y por último podría provenir de una base de datos si

`DataField` `logoreport`  
`DataSet` `DatosEmpresa`

indicamos las propiedades `Dataset` y `DataField`, además podemos asignar otras propiedades como el autoajuste, el centrado en su contenedor, la calidad, mantener el ratio de aspecto a pesar del tamaño, etc Una propiedad `IPrintable` indica si además se imprime, útil para presentar objetos en previsualización pero no imprimirlos en impresora si el papel que se usa ya tiene el logo preimpreso o si la factura viene ya preimpresa y solo se van a imprimir los datos, pero en pantalla se quiere mostrar completa.

La alineación.

Es importante destacar que la alineación de este objeto es `Align` `baLeft` y de esta manera siempre estará a la izquierda de su banda. Si además a la derecha la fecha, hora,

usuario están con `Align` `baRight` siempre estarán a la derecha y los títulos podrían estar centrados pero es más lógico que sean `Align` `baWidth` ya que de esta manera ocuparán el resto del contenedor y a su vez saldrán centrados si usamos el centrado de texto 

Usar la alineación es muy interesante para tener un diseño flexible que permita después por ejemplo pasar a un informe apaisado sin tener que modificar absolutamente nada.

### Mostrar Variables en campos Memo.

Como se puede ver a la derecha se muestran textos y variables entre corchetes `[Date]` y `[Time]` son variables propias de FR para mostrar la fecha y hora de la impresión.

```
[Fecha: [Date]
[
[Hora: [Time]
]
```

En el pie también pueden verse variables del propio FR `[Page] / [TotalPages#]` que mostrarán el nº de página actual más un texto / y el total de páginas del documento.

Junto al título también hay 2 variables pero en este caso son propias de nuestra aplicación y las usamos para complementar el título con 2 líneas donde poder indicar por ejemplo “Desde 01-01 Hasta 31-12” o “Solo cobros pendientes”

```
[HeadLine2]
[HeadLine3]
```

### Mostrar resultados de Funciones en campos Memo.

FR dispone de un buen nº de funciones propias pero además permite llamar a las funciones que queramos de (x)Harbour o incluso de nuestra aplicación e imprimir directamente aquello que devuelve su Return

Por ejemplo `[Usuario: [CallHbFunc('NetName',[True]])]` devuelve directamente el resultado de la función `NetName(.T.)` de clipper, o sea el nombre del usuario de la estación de trabajo. Hay que tener en cuenta que el primer parámetro es el nombre de la función entre “ ” y el segundo es un array de parámetros que la función necesita entre [ ] y separados por comas. Además deben utilizarse los valores en formato Pascal donde .T. es TRUE.

### Mostrar campos de los dataset y sus nombres en la cabecera

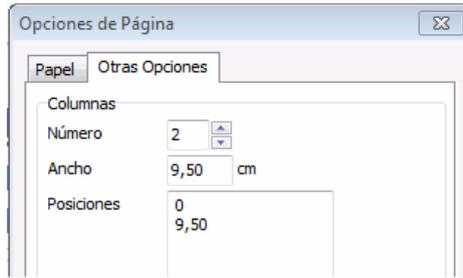
En el PageHeader mostraremos normalmente campos memo con los nombres escritos a mano de los campos que se muestran debajo en el Masterdataset. Además hemos incorporado al PageHeader un fondo degradado `tFrXGradientView`  que hemos alineado a `baClient` para que ocupe el espacio completo de la banda.

**NOTA:** si se añade este degradado después de poner textos se pondrá encima de estos, para ordenar las capas de objetos hay que pulsar botón derecho en uno de ellos e indicar el orden con las opciones

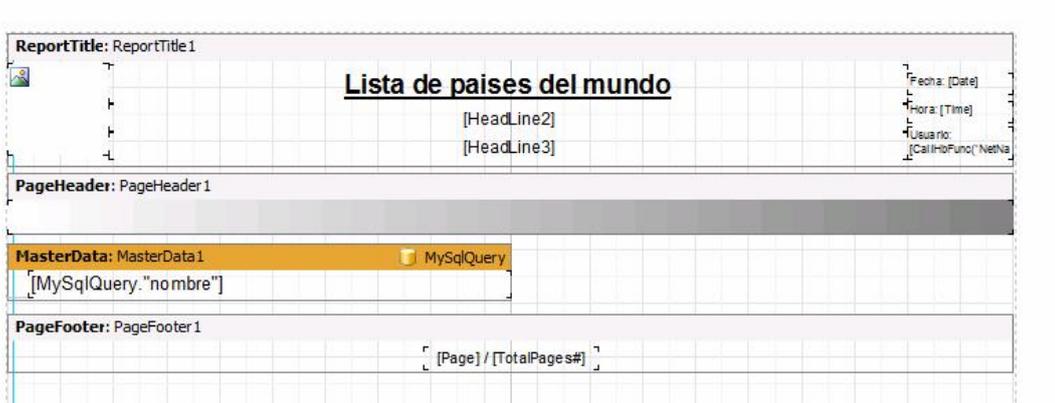
```
Traer al frente
Enviar Atrás
```

Para poder añadir campos de un dataset o un query primeramente tenemos que habérselo suministrado a FR, segundo tenemos que haberlo seleccionado de la lista de dataset desde el menú Informe – Datos y tercero haberlo asignado a la banda MasterData haciendo doble clic sobre la banda naranja. Después podemos arrastrar campos desde la lista que se muestra a la derecha. Se ajusta el tamaño de letra deseado, los siguientes saldrán con este tamaño y se le da la proporción deseada en anchura. Si la anchura proporcionada no es suficiente el campo se trunca, pero si se le da más altura se mostrará en varias líneas.

Como en el caso planteado solo hemos puesto una columna, ocuparemos solo media hoja de ancho, entonces podemos aprovechar el resto de la hoja indicando con doble clic sobre la hoja la propiedad de columnas como 2 columnas.



Quedando el diseño



Y pasando a imprimirse como



La línea de separación que se observa entre las líneas, está añadida con un tFrLineView  Línea en el MasterDataSet debajo de los campos y alineada a baWidth, para que vaya de lado a lado.

### Otra manera de hacer columnas.

Hemos visto que hacer 2 columnas ha sido muy sencillo, pero este sistema tiene una pega, primero escribe la primera columna de arriba abajo y luego la siguiente como en un procesador de textos.

Si queremos que las columnas sean escritas en el orden de lectura normal de izquierda a derecha dejémoslo en una sola columna y a continuación seleccionando la banda MasterData indicamos en sus propiedades:

ColumnGap	0,50
Columns	2
ColumnWidth	9,50

ColumnGap será la separación entre columnas o canal, las columnas son 2 y la anchura de cada una 9,50. Entre estas dos mas la separación no deben de exceder del ancho de la página o si no quedarán truncadas.

El resultado parece el mismo pero está ordenado de otra manera, que quizás sea a la estamos más acostumbrados y a la vista más logica también para el usuario.

## Informe columnar con grupos anidados

En este informe vamos a utilizar grupos de información para mostrar la información más clara y poder totalizarla.

Se trata de un informe de riesgos de pagos de clientes.

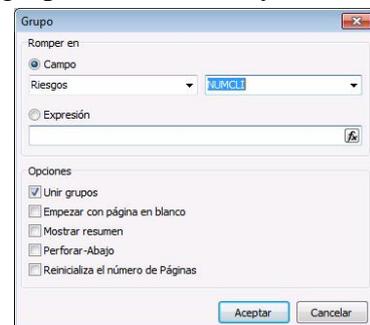
ReportTitle: ReportTitle1									
<b>Informe de riesgos con clientes</b>									
GroupHeader: GroupHeader1									
[Riesgos."N"]	[Riesgos."NOMBRE"]								[Date]
[Fecha Vto.]	[N° Factura]	[Fecha Factura]	[Doc. Recibido]	[Entidad]	[Total Factura]	[Documento de cobro]			[Time]
GroupHeader: GroupHeader2									
Riesgos."FEHAVTO"									
MasterData: MasterData1									
[Riesgos."Fe"]	[Riesgos."N"]	[Riesgos."Fe"]	✓	[Riesgos.""]	[Riesgos."To"]	[Riesgos."MedioCobro"]			[Riesgos."Im"]
GroupFooter: GroupFooter2									
								[Total Vtos.]	[SUM(<Riesg
GroupFooter: GroupFooter1									
								[Total Vtos. cliente]	[SUM(<Riesgos
ReportSummary: ReportSummary1									
[Periodo medio de Vto:]				[<nPMVto]	[Total vencimientos . . . . .]				[SUM(<Riesgos."I
PageFooter: PageFooter1									
[Page] / [TotalPages#]									

Se observa que además de las bandas “normales” hay dos GroupHeader y dos GroupFooter.

En el GroupHeader 1 hemos aprovechado para incluir los títulos de las columnas en vez de en un PageHeader, para que se muestren en cada nuevo grupo, además incluye también dos campos de la tabla para indicar que cliente es en cada caso.

Lo más importante es indicar cual es el campo que hace el salto del grupo, que puede ser un campo concreto de la tabla o una expresión suma de varios campos o de una función. Cada Grupo tendrá su salto correspondiente y en este caso la primera agrupación es por el numero del cliente y la segunda es por la fecha del cobro.

**NOTA:** Si queremos usar agrupaciones los datos del dataset han de venir forzosamente ordenados por estos campos ya que FR no los ordena.



En el GroupHeader 2 solo hemos indicado el campo de agrupamiento y no contiene ningún elemento por lo que lo hemos dejado al mínimo posible de altura para que no ocupe.

En el GroupFooter 2, que es el primero en orden de vista para ir cerrando los grupos, se ha indicado una operación de sumatorio [SUM(<Riesgos."IMPORTE">,MasterData1)] que solo suma este campo para los miembros del grupo 2 de la misma fecha

En el GroupFooter 1, se ha indicado una operación de sumatorio que es la misma [SUM(<Riesgos."IMPORTE">,MasterData1)] pero suma este campo para los miembros del grupo 1 de todas las fechas del mismo cliente.

Así se obtiene el siguiente resultado:

 <span style="float: right;">30/05/2013 8:43:37</span>							
<b>Informe de riesgos con clientes</b>							
<b>43000015 AMANCO ESPINA PEDRUEZA</b>							
Fecha Vto.	Nº Factura	Fecha Factura	Doc. Recibido	Entidad	Total Factura	Documento de cobro	Importe Vto.
20/05/2013	FV130077	30/04/2013			52.750,50	TALÓN	52.750,50
<b>Total Vtos.</b>							<b>52.750,50 €</b>
<b>Total Vtos. cliente</b>							<b>52.750,50 €</b>
<b>43000123 CORRALGADOS ATPEYIA, S.L.</b>							
Fecha Vto.	Nº Factura	Fecha Factura	Doc. Recibido	Entidad	Total Factura	Documento de cobro	Importe Vto.
16/05/2013	FV130067	26/04/2013	✓		57.155,52	PAGARÉ	57.155,52
<b>Total Vtos.</b>							<b>57.155,52 €</b>
23/05/2013	FV130087	03/05/2013	✓		3.283,65	PAGARÉ	3.283,65
23/05/2013	FV130086	03/05/2013	✓		69.157,41	PAGARÉ	69.157,41
23/05/2013	FV130082	30/04/2013	✓		73.326,35	PAGARÉ	73.326,35
<b>Total Vtos.</b>							<b>145.767,41 €</b>
<b>Total Vtos. cliente</b>							<b>202.922,93 €</b>

**Banda ReportSummary.**

Se muestra al final del informe después de terminar el MasterData y se usa p.e. para hacer cálculos de todo el informe.

```
ReportSummary: ReportSummary1
[Periodo medio de Vto: ] [

```

Como se ve la formula de la derecha es la misma que hemos indicado en GroupFooter 1 y 2, pero en este caso nos dará la suma de todo el informe por no pertenecer a un grupo. Y a la izquierda tenemos un campo calculado [

**Eventos en FastReport y el editor de código.**

Para realizar el cálculo tendremos que teclear un poco de código dentro de FR pero antes veamos en que momento.

Cada uno de los objetos del diseño tiene una serie de eventos que podemos utilizar a nuestro gusto.

Por ejemplo en este caso quiero computar los días de vencimiento de cada línea del MasterData para hacer una media y por lo tanto voy a utilizar el evento OnBeforePrint de MasterData para hacerlo en cada línea que va al informe. Los eventos están a la izquierda en una solapa junto a las propiedades del objeto. Haciendo doble clic sobre ellos nos lleva al editor de código de FR y en el podemos escribir código en varios lenguajes Pascal, C++, Basic y Java, aunque el primero es el sistema nativo de FR y usando los otros me he encontrado problemas de funcionamiento.

Este es el código necesario para este cálculo:

```
var nPMVto1 : double; nPMVto2 : double;

procedure MasterData1OnBeforePrint(Sender: TfrxComponent);
begin
  nPMVto1:=nPMVto1+(<Riesgos."IMPORTE">*( <Riesgos."FECHAVTO">-<Riesgos."FECHAFAC">)) ;
  nPMVto2:=nPMVto2+<Riesgos."IMPORTE"> ;
end;

begin
end.
```

Se definen 2 variables nPMVto1 y nPMVto2 como numéricas de tipo doble y para cada línea de masterdata antes de ser impresa se hacen unos acumulados que luego podemos utilizar en nuestro informe para dar este resultado al final del último cliente:

Fecha Vto.	Nº Factura	Fecha Factura	Doc. Recibido	Entidad	Total Factura	Documento de cobro	Importe Vto.
28/04/2013	FVR130005	31/03/2013			225,54	TALÓN	225,54
<b>Total Vtos.</b>							<b>225,54 €</b>
31/05/2013	FVR130002	28/02/2013	✓		110,35	PAGARÉ	110,35
<b>Total Vtos.</b>							<b>110,35 €</b>
<b>Total Vtos. cliente</b>							<b>335,89 €</b>
<b>Periodo medio de Vto: 91,60 Dias</b>				<b>Total vencimientos . . . . .</b>		<b>10.128.713,02 €</b>	

Otro ejemplo de código. Si por ejemplo queremos que en el GroupFooter2 aparezca el nombre del cliente podemos poner este campo dentro

```
GroupFooter: GroupFooter1
[Total Vtos. [Riesgos."Nombre"] [SUM(<Riesgos
```

Y funciona perfectamente salvo para el último cliente, ya que terminados los datos de MasterData ya no tiene el campo para el pie final que queda vacío.

<b>Total Vtos.</b>	<b>1.096,55 €</b>
<b>Total Vtos. EXCAVACIONES</b>	<b>3.588,18 €</b>

Entidad	Total Factura	Documento de cobro	Importe Vto.
	225,54	TALÓN	225,54
<b>Total Vtos.</b>			<b>225,54 €</b>
	110,35	PAGARÉ	110,35
<b>Total Vtos.</b>			<b>110,35 €</b>
<b>Total Vtos. 0,00</b>			<b>335,89 €</b>

La solución es inventarnos una variable cNombre que en cada cambio de GroupHeader almacene el valor del cliente, es decir, en el evento OnBeforePrint del GroupHeader1 almacenamos cNombre:= <Riesgos."Nombre">;

```
var nPMVto1 : double; nPMVto2 : double; cNombre : Char ; |

procedure MasterData1OnBeforePrint(Sender: TfrxComponent);
begin
  nPMVto1:=nPMVto1+(<Riesgos."IMPORTE">*( <Riesgos."FECHAVTO">-<Riesgos."FECHAFAC">)) ;
  nPMVto2:=nPMVto2+<Riesgos."IMPORTE"> ;
end;

procedure GroupHeader1OnBeforePrint(Sender: TfrxComponent);
begin
  cNombre:= <Riesgos."Nombre">;
end;

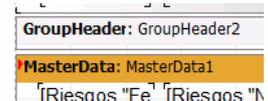
begin
end.
```

Y así usamos este valor en el GroupFooter1 en lugar del campo del dataset.

GroupFooter: GroupFooter1									
								Total Vtos. [cNombre]	[SUM(<Riesgos

Y saldrá perfectamente hasta en el último grupo.

**NOTA:** Cuando un objeto tiene código asociado a uno de sus eventos aparece al lado izquierdo de éste un punto rojo.



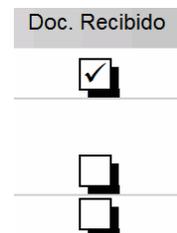
### Campos lógicos con el componente tFrxCkView.

Un detalle de este informe es que la columna **Doc. Recibido** muestra un símbolo de check en lugar de un True o un Sí.

Esto se hace tomando de la paleta izquierda el componente  e indicando entre sus

propiedades **DataField** `ReciPagare` **DataSet** `Riesgos` con lo que el resultado será un check a elegir de entre varios tipos **CheckStyle** `csCheck` o un elemento vacío.

Es recomendable añadir un cuadro alrededor de este componente para que sea más visual en caso de estar vacío para ver más claramente que indica .F.



## Informe columnar maestro - detalle

Trabajando con DBF's podemos utilizar la posibilidad de hacer informes Maestro / Detalle-s de forma que se relacionan dos tablas mediante un índice.

Para ello debemos de indicarlo en nuestro código fuente al instanciar FR de la siguiente manera

```
With object frReportManager():New()
  If !:lError
    :oBrowse:=:oDbBrowse
    :SetDataSet(Self)
    :SetMasterDetail("Vehiculos", "VehiculosRepostajes", {| |Vehiculos->Matricula})
    :SetMasterDetail("Vehiculos", "VehiculosMantenimientos", {| |Vehiculos->Matricula})

    :ReportList(:cFormName,!Appdata:lUser)
  Endif
End With
```

Como se observa hay tres DBF, Vehiculos, VehiculosRepostajes con la lista de veces que ha repostado este vehiculo y VehiculosMantenimineto con la lista de veces que ha ido al taller. Estos dos últimos tienen un índice principal que es la matrícula que es el TAG activo y se relacionan con el primero a través del campo Vehiculos->Matricula como se indica en

```
:SetMasterDetail("Vehiculos", "VehiculosRepostajes", {| |Vehiculos->Matricula})
:SetMasterDetail("Vehiculos", "VehiculosMantenimientos", {| |Vehiculos->Matricula})
```

De esta manera es FR el que se encarga de hacer las Set Relation entre las tablas para que con el siguiente diseño

ReportTitle: ReportTitle1			
<b>Informe de costos de vehiculos</b>		[Date]	[User]
[HeadLine2]		[Time]	
[HeadLine3]			
GroupHeader: GroupHeader1 Vehiculos."MATRICULA"			
MasterData: MasterData1 Vehiculos			
Vehículo [<Vehiculos."MATRICULA">+ ' +<Vehiculos."MARCA">+]			
Header: Header1			
<b>Repostajes</b>			
DetailData: DetailData1 VehiculosRepostajes			
[VehiculosRe	[VehiculosRe	[VehiculosRe	[VehiculosRep
Footer: Footer1			
Total repostajes		[SUM(<Vehicu]	[SUM(<Vehicu]
Header: Header2			
<b>Mantenimiento</b>			
DetailData: DetailData2 VehiculosMantenimientos			
[VehiculosMa	[VehiculosMa	[VehiculosMantenimientos."CONCEPT	[VehiculosMant
Footer: Footer2			
Total mantenimiento		[SUM(<Vehicu]	
ReportSummary: ReportSummary1			
Costo total		[SUM(<Vehicu]	
PageFooter: PageFooter1			
		[Page]	[TotalPa]

De lugar al siguiente informe



### Informe de costos de vehículos

Desde 01-01-2013 Hasta 30-05-2013

30/05/2013

10:27:49

Administrador

#### Vehículo 2314DND FORD TRANSIT

##### Repostajes

01/02/2013	250 Km	50 Lt	60,00 €
14/02/2013	1.600 Km	65 Lt	72,00 €
19/03/2013	2.250 Km	50 Lt	65,00 €
<b>Total repostajes</b>		<b>165 Lt</b>	<b>197,00 €</b>

##### Mantenimiento

30/05/2013	2.000 Km	Revision inicial	120,00 €
30/05/2013	3.500 Km	Cambios de filtros	231,00 €
<b>Total mantenimiento</b>			<b>351,00 €</b>

**Costo total 548,00 €**

Se puede observar que para un MasterData hay dos DetailData cada uno con su Header y Footer correspondiente, e incluso con una configuración de columnas y valores independiente y un ReportSummary final con la suma de todo ello

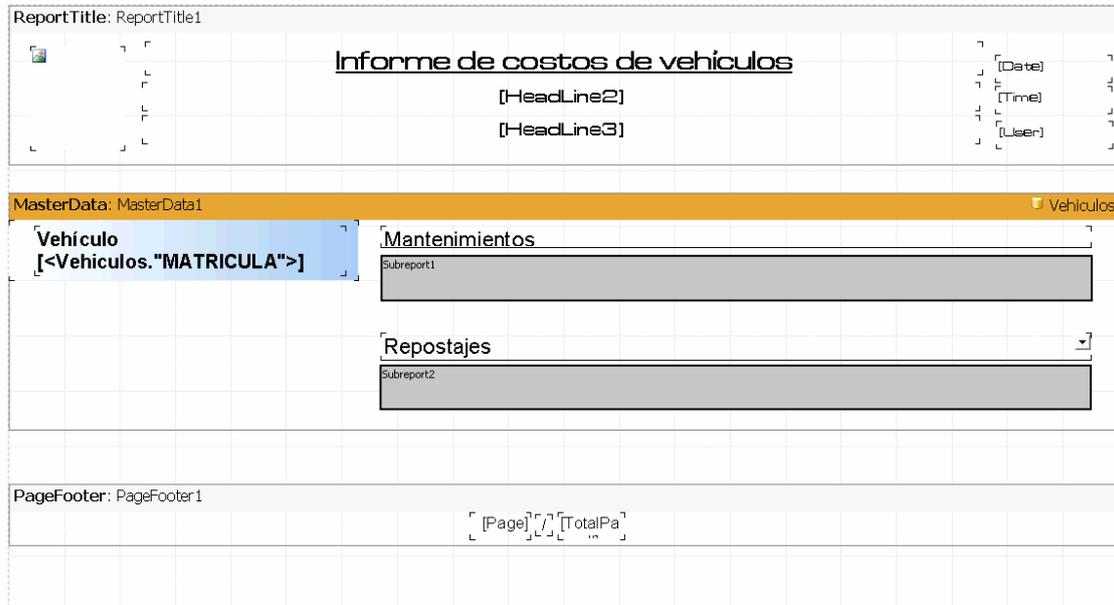
```
[SUM(<VehiculosRepostajes."IMPORTE">,DetailData1)+SUM(<VehiculosMantenimientos."IMPORTE">,DetailData2)]
```

El problema es que este esquema maestro–detalle **no funciona para queries de MySql** con lo que hay que optar por otros sistemas como traer en un único Query todos los datos de la cabecera y de las líneas, duplicando os datos de la cabecera para cada línea y utilizar Groups para mostrar los resultados de forma parecida, pero no se pueden usar esquemas maestro–detalle sincronizados.

## Informe columnar con subinforme

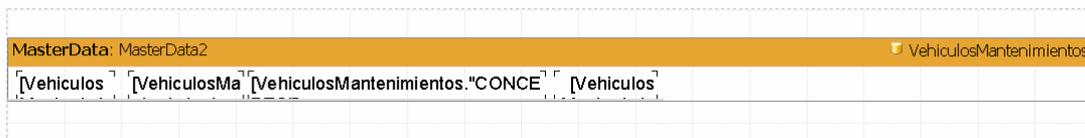
Una variante de un informe maestro-detalle es el informe con subinforme, en el cual las líneas del DetailData van insertas dentro del MasterData como subinformes.

Para ello hacemos un diseño como el siguiente:

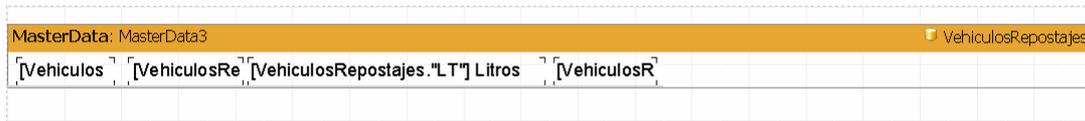


En principio solo tiene un MasterData y dentro de el vamos a meter dos Subinformes, para ello tomamos el objeto  y lo soltamos sobre el Masterdata y de repente aparece una nueva página de diseño llamada SubReport1 y repitiendo la operación otra llamada SubReport2. Código | Data | Page1 | Subreport1 | Subreport2

En cada una de ellas habrá que realizar un nuevo diseño, sobre un MasterData por ejemplo en la primera:



Y en la segunda de ellas:



Como se puede ver se ha diseñado a la izquierda intencionadamente, por que después será impreso en el espacio que queda a la derecha del MasterData del diseño Página 1 y lo que no quepa se truncará.

De esta manera la presentación final será la siguiente

		<b>Informe de costos de vehículos</b>		<small>05/05/2013</small> <small>18:57:39</small> <small>Administrador</small>
		<small>Desde 01-01-2013 Hasta 31-05-2013</small>		
<b>Vehículo 2312DGH</b> <b>FIAT DUCATO</b>	<u>Mantenimientos</u>			
	10/04/2013	1.500 Km Revision inicial		150,00 €
	10/05/2013	15.000 Km Revision 15000 km		345,00 €
	Repostajes			
<b>Vehículo 2314DND</b> <b>FORD TRANSIT</b>	<u>Mantenimientos</u>			
	30/05/2013	2.000 Km Revision inicial		120,00 €
	30/05/2013	3.500 Km Cambios de filtros		231,00 €
	Repostajes			
	01/02/2013	250 Km. 50 Litros		60,00 €
	14/02/2013	1.600 Km. 65 Litros		72,00 €
19/03/2013	2.250 Km. 50 Litros		65,00 €	

**NOTA:** Este esquema tampoco sirve para bases de datos SQL, solo sirve para DBF relacionados.

## Informes con índices o miniaturas

Podemos incorporar fácilmente a un informe un índice sobre todo si está agrupado.

Para ello basta que indiquemos en la cabecera/s de grupo que genere un índice con un campo de ellas

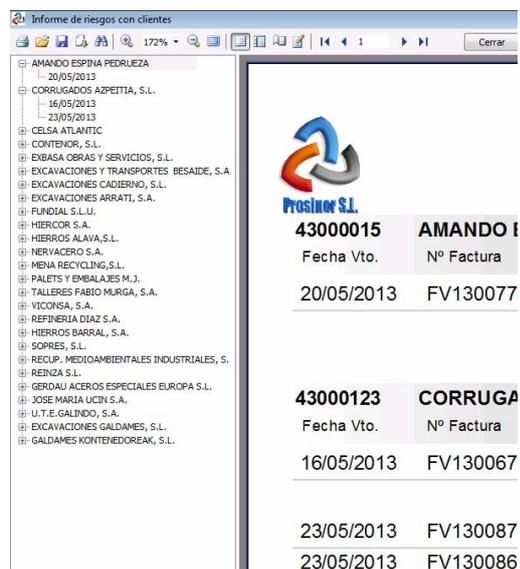
Por ejemplo en este informe

ReportTitle: ReportTitle1		[Date]
<b>Informe de riesgos con clientes</b>		[Time]
GroupHeader: GroupHeader1		Riesgos."NUMCLI"
[Riesgos."N"]	[Riesgos."NOMBRE"]	
[Fecha Vto.]	[Nº Factura]	[Fecha Factura]
[Doc. Recibido]	[Entidad]	[Total Factura]
[Documento de cobro]	[Importe Vto.]	
GroupHeader: GroupHeader2		Riesgos."FECHAVTO"
MasterData: MasterData1		Riesgos
[Riesgos."Fe"]	[Riesgos."N"]	[Riesgos."Fe"]
<input checked="" type="checkbox"/>	[Riesgos."To"]	[Riesgos."MedioCobro"]
[Riesgos."Im"]		
GroupFooter: GroupFooter2		
	[Total Vtos.]	[SUM(<Riesg
GroupFooter: GroupFooter1		
	[Total Vtos. [cNombre]	[SUM(<Riesgos
ReportSummary: ReportSummary1		
[Periodo medio de Vto.]	[<nPMVto1>/<nPMVto2>] Dias	[Total vencimientos . . . . .]
		[SUM(<Riesgos."I
PageFooter: PageFooter1		
	[Page] / [TotalPages#]	

En el GroupHeader1 introduzco en su propiedad `OutlineText Riesgos."NOMBRE"` y en el GroupHeader 2 `OutlineText Riesgos."Fechavto"`.

Con esto estoy indicando que se genere un índice a dos niveles que se mostrará en la parte izquierda del documento, el primer nivel estará visible y el segundo aparece en forma de árbol cuando se pulse 

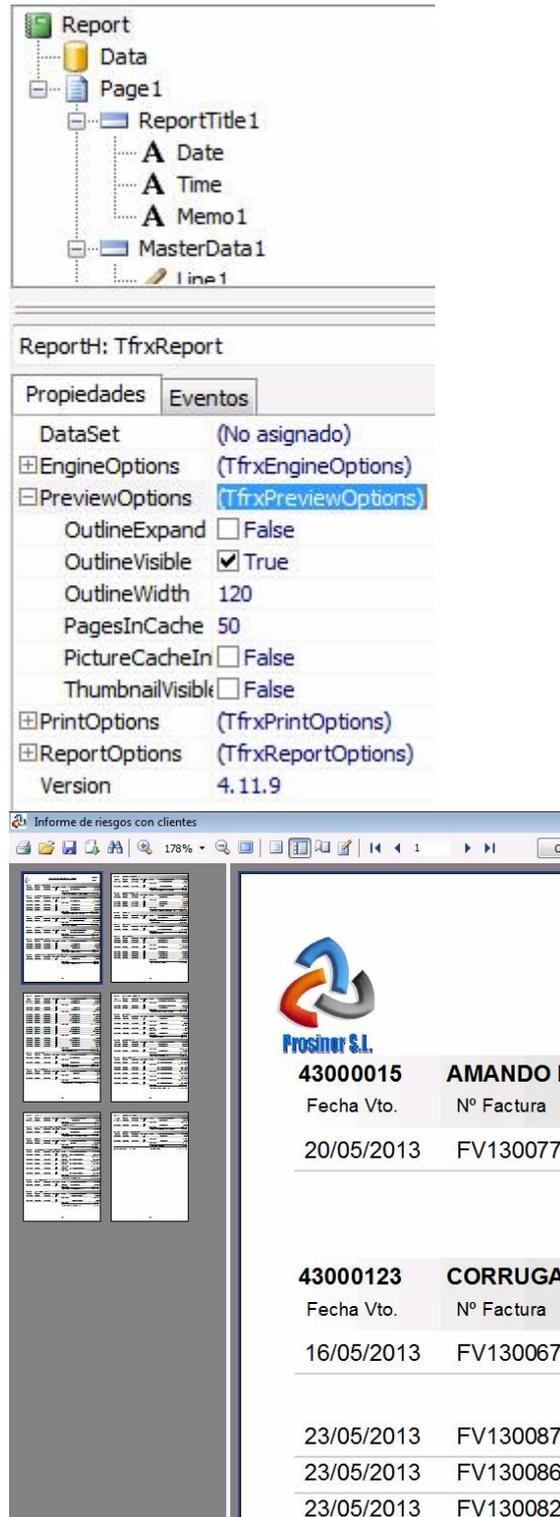
Cada vez que se pulse en cualquiera de los niveles el documento se situará en la página y línea a la que hace referencia el índice.



The screenshot shows a report viewer window titled "Informe de riesgos con clientes". On the left, there is a tree view listing various companies and their invoice dates. On the right, there is a data table with columns for "Fecha Vto." and "Nº Factura".

Fecha Vto.	Nº Factura
20/05/2013	FV130077
16/05/2013	FV130067
23/05/2013	FV130087
23/05/2013	FV130086

Para que el índice sea visible o bien se hace clic en la previsualización sobre el botón  o si se quiere presentar ya abierto se pueden manipular las propiedades del Report en su sección PreviewOptions. Aquí podemos indicar si queremos que el índice aparezca expandido y visible con la propiedad "OutlineVisible" a True y su anchura con la propiedad "OutlineWidth"



The image shows the FastReport 4 interface. At the top, a tree view displays the report structure: Report -> Data -> Page 1 -> ReportTitle 1 (containing Date, Time, Memo 1) and MasterData 1 (containing Line 1). Below this is the 'ReportH: TfrxReport' properties window, which has two tabs: 'Propiedades' and 'Eventos'. The 'Propiedades' tab is active, showing various settings for the report. The 'PreviewOptions' section is expanded, showing the following properties:

Propiedad	Valor
DataSet	(No asignado)
EngineOptions	(TfrxEngineOptions)
PreviewOptions	(TfrxPreviewOptions)
OutlineExpand	<input type="checkbox"/> False
OutlineVisible	<input checked="" type="checkbox"/> True
OutlineWidth	120
PagesInCache	50
PictureCacheIn	<input type="checkbox"/> False
ThumbnailVisible	<input type="checkbox"/> False
PrintOptions	(TfrxPrintOptions)
ReportOptions	(TfrxReportOptions)
Version	4.11.9

Below the properties window is a preview window titled 'Informe de riesgos con clientes'. It shows a grid of thumbnails on the left and a detailed view of a report page on the right. The detailed view includes the Prosinar S.L. logo and a table of invoice data:

Nº Factura	Fecha Vto.	Cliente
43000015	20/05/2013	AMANDO E
43000123	16/05/2013	CORRUGAI
	23/05/2013	
	23/05/2013	
	23/05/2013	

Aunque es menos útil, también se pueden mostrar miniaturas de las páginas a la izquierda del documento. Para esto no hay que preparar nada y basta con pulsar el botón  de la previsualización y se quiere mostrar ya abierto se pueden manipular las propiedades del Report en su sección PreviewOptions. Aquí podemos indicar si queremos que el "ThumbnailVisible" sea True para que aparezca visible.

Pulsando sobre cada miniatura nos lleva a la página correspondiente.

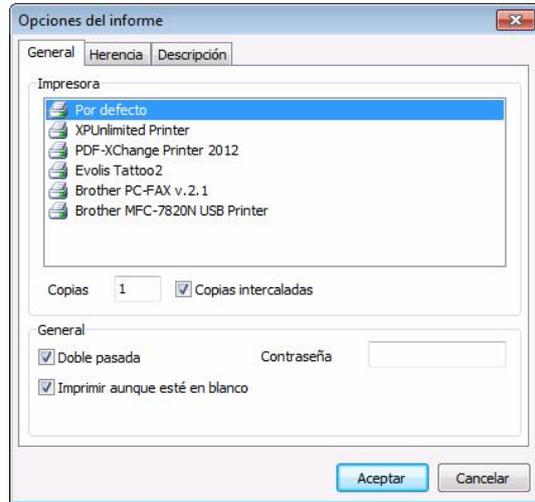
### Informe columnar con multicopia y un texto identificativo por copia.

A menudo hay que imprimir un documento con varias copias, incluso un número de copias fijas siendo una copia para la empresa, otra para el interesado, otra para la administración, etc.

Para esto primero se puede incorporar directamente el número de copias a imprimir directamente al diseño desde el menú Informe – Opciones – Copias. Pero de esta manera o solicitando el usuario n copias al imprimir, todas ellas serán iguales.

Para diferenciarlas se puede programar un texto diferente a imprimir en cada una de ellas.

Tendremos que realizar el diseño y dentro de este incluir una variable de FR llamada [CopyName#] que en este caso pondré en el fondo de la página a 45° y en letra grande y gris.



PageHeader: PageHeader1

CIF. B-95300398

[Clientes."clirazonsoc"]  
[Clientes."clinombre2"]  
[Clientes."clinombre3"]  
[Clientes."clidireccion"] [Clientes."clidireccion2"]  
[Clientes."clipostal"] - [Clientes."clipoblacion"]  
[Clientes."cliprovincia"]  
[Clientes."nombrepais"]

Factura N°	Fecha	Pagina N°	Cliente N°	DNI / NIF / VAT	Referencia Cliente		
[Facturas."CO"]	[Facturas."F"]	[Page]	[Clientes."clidni"]	[Facturas."FACREFCLENT"]			
Fecha	Contenedores	Ref.Pedido	Origen / Destino	Concepto	Unds	Bruto	Neto

GroupHeader: GroupHeader1

[Lines."CodigoAlb"]

[Lines."F"] [Lines."NumConte"] [Lines."PedRe fPed"] [Lines."ORIDES"] [Lines."POBLACIONES"]

MasterData: MasterData1

[Lines."PEDTIP"] [Lines."Text"] [ds] [to]

GroupFooter: GroupFooter1

ReportSummary: ReportSummary1

[>,MasterData1]> > 0, 'Total']

PageFooter: PageFooter1

[Forma de Pago]	[Total Importes]	[Facturas."FAC"]
[Facturas."nombrefpa"]	[Fact."NIF(<Fact"] [Facturas."FAC"]	[Facturas."FA"]
[Vencimiento] [Facturas."facvto1"]	[Base Imponible]	[Facturas."FAC"]
	[Facturas."FAC"]	[Facturas."FAC"]
	[TOTAL FACTURA EUROS]	[Facturas."FAC"]

En cumplimiento de lo dispuesto en la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal se informa que los datos recogidos en este documento y los que pudiésemos necesitar serán incorporados en un fichero informatizado, del cual es responsable COTRANSON S.L. En el caso de producirse alguna modificación de sus datos, pedimos nos lo comuniquen debidamente por escrito. Podrá ejercer sus derechos de acceso, rectificación, cancelación y oposición, en los términos que establece la Ley, a la dirección: COTRANSON S.L. C/ CAPITAN MENDEZABAL Nº 12 LONJA TRASERA 48960 SANTURCE/VIZCAYA

Para que funcione deberemos de ir a un objeto como por ejemplo el PageHeader y utilizar su evento OnBeforePrint para añadir el siguiente código:

```

procedure PageHeader1OnBeforePrint(Sender: TfrxComponent);
begin
    frxGlobalVariables['CopyName0'] := 'Previsualización'; // Texto del preview
    frxGlobalVariables['CopyName1'] := 'ORIGINAL'; // Texto de la primera copia
    frxGlobalVariables['CopyName2'] := 'COPIA'; // Texto de la segunda copia
    frxGlobalVariables['CopyName3'] := 'Administración'; // Texto de la tercera copia

    if <GeneraMail> = True
    then
        frxGlobalVariables['CopyName1'] := 'COPIA'; // Texto que aparece en el emmail
    end;
end;
    
```

De esta manera al previsualizar aparecerá el texto “Previsualización” o nada si ponemos “”, pero en la primera copia impresa pondrá “Original” en la segunda “Copia” en la tercera “Administración”

CIF. B-33555553

**EUROPA S.A.**

CANAL DE SALES 8-1º PLANTA

28000 - MADRID

MADRID

SPAIN

Factura Nº	Fecha	Página Nº	Cliente Nº	DNI / NIF / VAT	Referencia Cliente
2013-000001	15/04/2013	1 / 1	4	A84430107	

Fecha	Contenedores	Ref. Pedido	Origen / Destino	Concepto	Unds	Bruto	Neto
04.04.13	TTNU5827983--		CANDINA CANDINA	Iruztun / Iruztun			
	8014CJM	SAPREM SA DE PREFORMADOS METALICOS	IRURTZUN / IRURTZUN		1,00	528,84	
	8014CJM			FUEL SURCHARGE (BAF)	1,00		37,02
10.04.13	CLHU4665705--		CANDINA CANDINA	Legutiano			
	5198-CNY			ULMA   LEGUTIANO	1,00	345,18	
	5198-CNY			FUEL SURCHARGE (BAF)	1,00		24,16
11.04.13	TRIU0510630--		CANDINA CANDINA	Cubás De La Sagra			
	6657DBW	AGADIN (TALLER DE PINTURA)	CUBAS DE LA SAGRA		1,00	1172,50	
	6657DBW			FUEL SURCHARGE (BAF)	1,00		82,08
12.04.13	AMFU4984319--		CANDINA CANDINA	Cubás De La Sagra			
	5127-CNY	AGADIN (TALLER DE PINTURA)	CUBAS DE LA SAGRA		1,00	1172,50	
	5127-CNY	aplicado técnicas del embalaje (Grifon)			1,00		56,65
	5127-CNY			FUEL SURCHARGE (BAF)	1,00		82,08
15.04.13	CLHU3344043--		CANDINA CANDINA	Reinosa			
	8014CJM	VICORGILTA CORTA PERU INTERNATIONAL SL	REINOSA		1,00	558,20	
	8014CJM			FUEL SURCHARGE (BAF)	1,00		38,93
<b>Total Recargo de gasoil BAF</b>						<b>264,27</b>	

<b>Forma de Pago</b>	<b>Total Importes</b>	4.096,14
Transferencia Bancaria 30 Dias	10,00 %Dto \$/	3.775,22
	<b>Base Imponible</b>	3.718,62
<b>Vencimiento</b> 15/05/2013	<b>TOTAL FACTURA EUROS</b>	<b>3.718,62</b>

En cumplimiento de lo dispuesto en la Ley Orgánica 15/1999, de Protección de Datos de Carácter Personal se le informa que los datos recogidos en este documento, los que pudieran ser necesarios para ser incorporados en un fichero informatizado, del cual se reserva por parte de CONTRANSCON, S.L., el derecho de producirle alguna modificación de uno de los datos, siempre que lo comunique de buena fe por escrito. Podrá ejercer sus derechos de acceso, rectificación, cancelación y oposición en los términos que establece la Ley, a la dirección: CONTRANSCON S.L., C/ CAPTAN, 18001 LA SAGRA, N.º 1 O LA SAGRA (TRAJED) A 489 80 SA. NUTRIC 8-1992 A 18

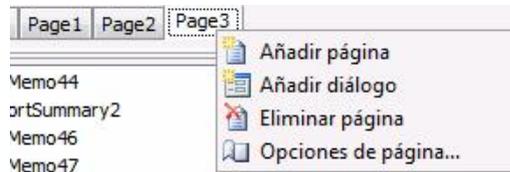
## Informe con múltiples diseños de página

FastReport normalmente contiene un único diseño de página para todo el documento a imprimir, pero si se desea se puede crear un diseño multipágina.

Ojo, no confundir el concepto de página de FR con una página de papel, para FR una página es un diseño con sus bandas de donde podrá resultar n hojas de papel cuyo número dependerá de los datos del dataset a imprimir. Pero se pueden utilizar más páginas de diseño para imprimir diferentes informes consecutivos como si fueran uno solo.

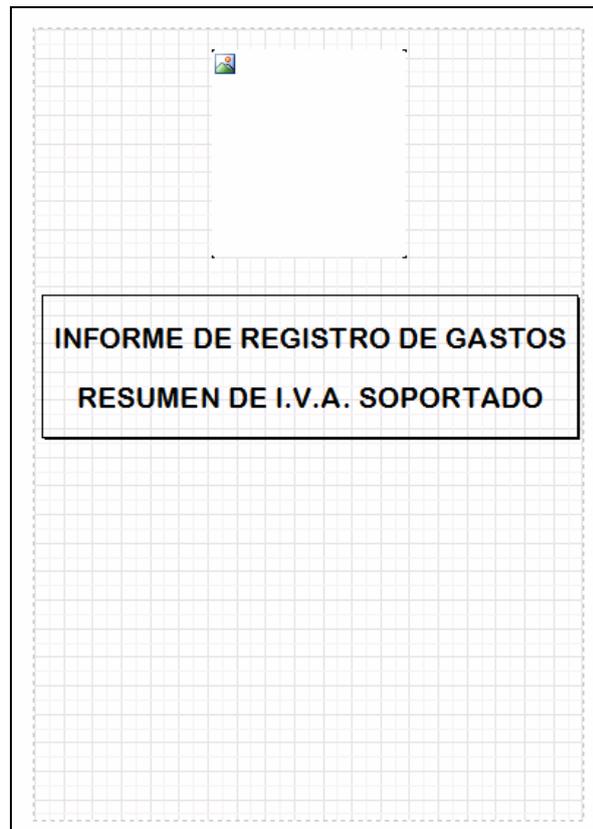
Veamos el siguiente ejemplo, se trata de un informe de IVA de gastos en el cual hemos diseñado una primera página que es una portada, después tendremos una segunda página que es el informe columnar de los gastos en hoja apaisada y finalmente en una página vertical se presenta un resumen de los IVAs de los gastos anteriores.

Para crear páginas de diseño hemos de situarnos en la parte superior sobre la página 1 y pulsar el botón derecho del ratón para obtener este menú y añadir páginas, que luego se pueden mover a izquierda o derecha.



En la primera página que será la portada simplemente introducimos un gráfico para el logotipo de la empresa y un texto enmarcado.

Una de las ventajas de usar un informe multipágina es que las n hojas resultantes de la fusión de todas ellas formarán una unidad y saldrán numeradas de forma consecutiva, claro que en la portada no queremos número y además en la primera hoja del siguiente diseño de página queremos que empiece por 1.



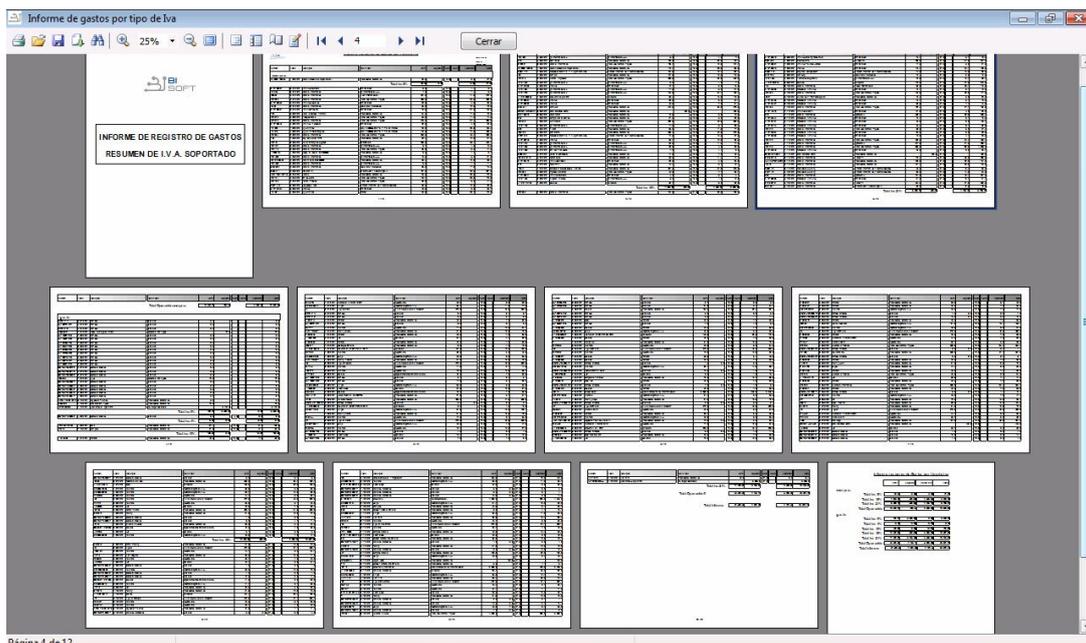
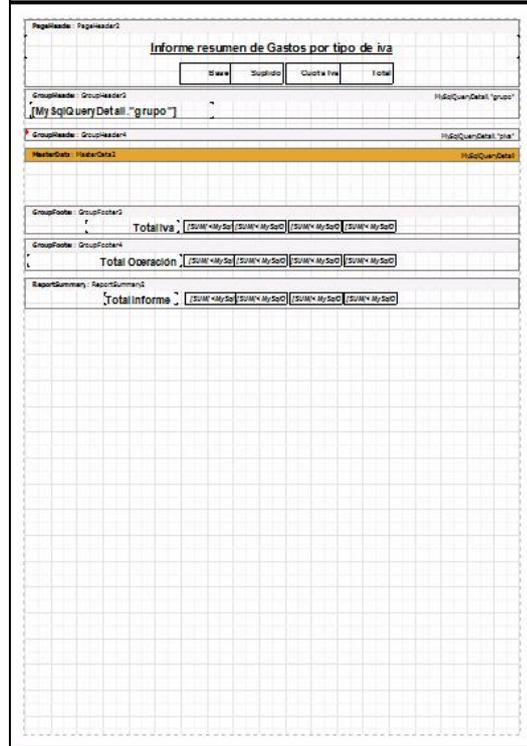
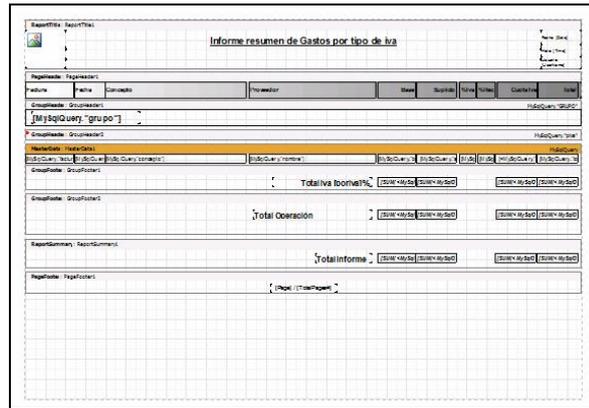
En la segunda página realizamos el diseño del informe columnar de gastos, con sus grupos, totales, etc.

Para que en la primera hoja de este diseño no ponga página 2, tenemos que seleccionar la hoja y cambiar la propiedad

```
ResetPageNumber  True
```

Y por último preparamos el resumen en un diseño de página separado que es totalmente diferente de lo anterior aunque usa el mismo dataset para obtener solo totales de grupo ya que el MasterData se deja intencionadamente vacío.

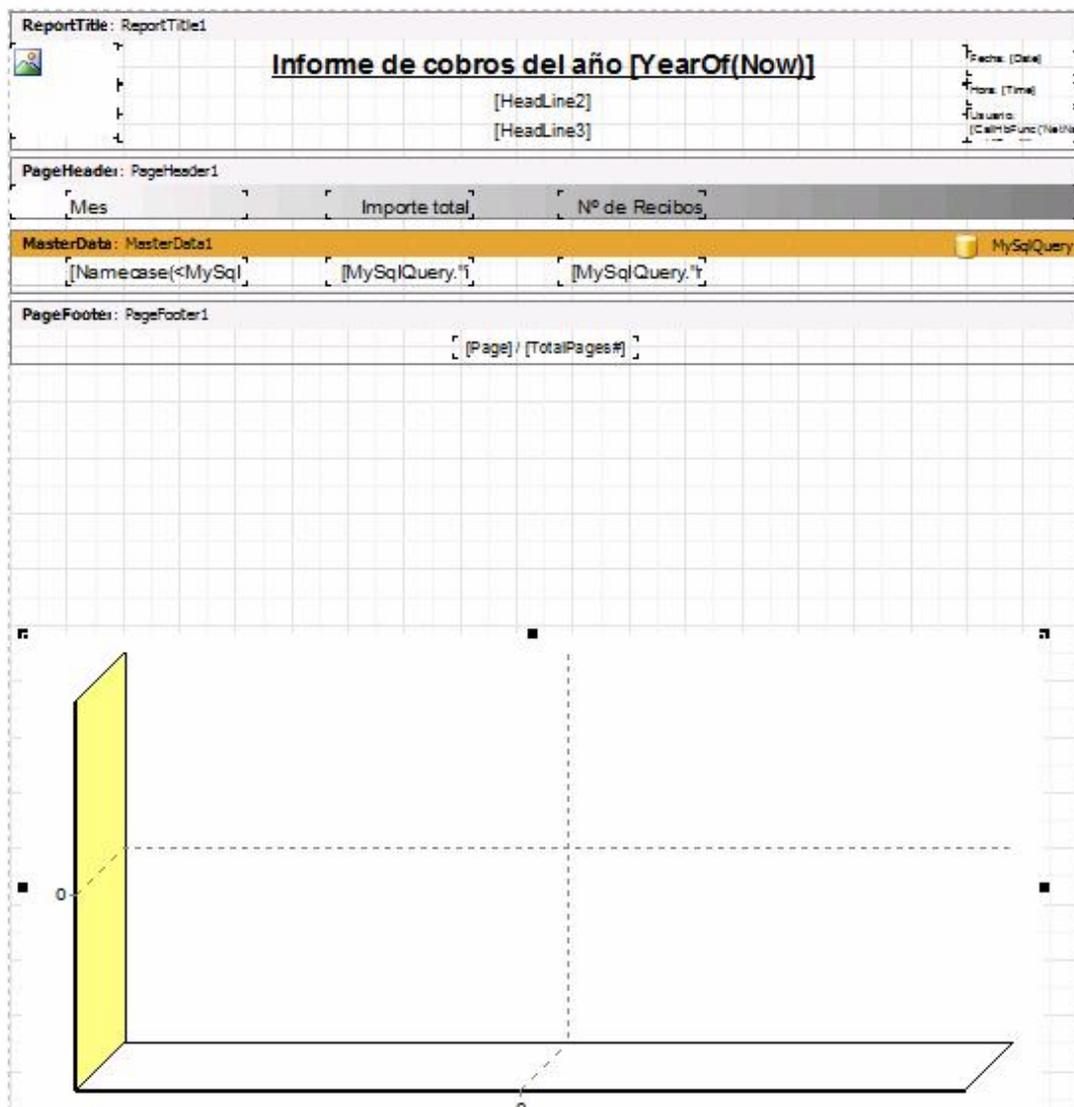
El resultado final será un documento completo listo para encuadernar.



## Informe con gráficas de datos

FastReport no solo es un generador de listados, también es capaz de generar documentos con otro tipo de elementos como por ejemplo graficación de datos que además se pueden mezclar o no con los informes.

Como puede verse en este diseño se puede añadir a un informe una gráfica que en este caso es de barras



Se ha añadido en la parte inferior de la misma página, por que se sabe que un listado mensual de cobros no supera las 12 líneas, si no sería mejor haberlo añadido en una banda de sumario para que aparezca bajo los datos de la última página, o en una página separada dentro de un informe multipágina.

Para confeccionar una gráfica disponemos del componente Chart



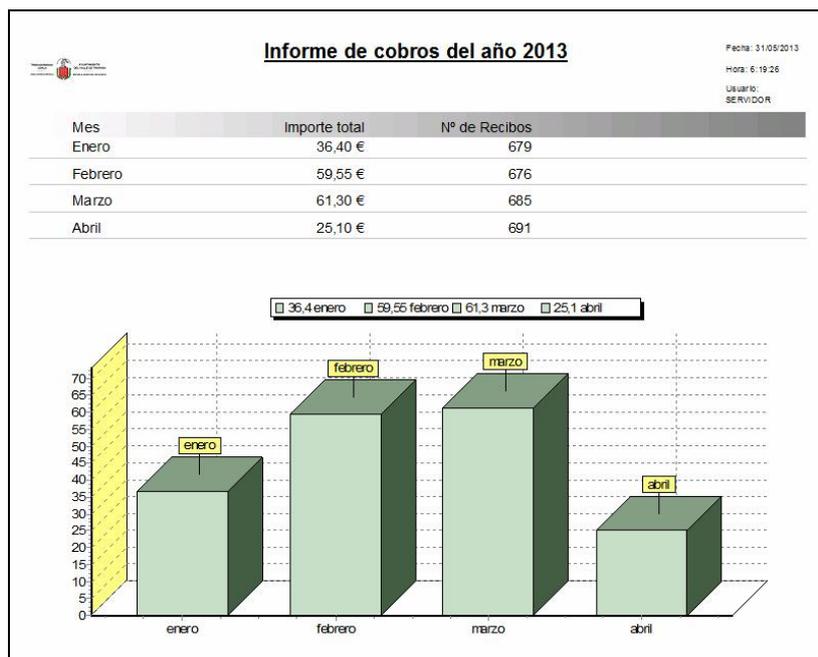
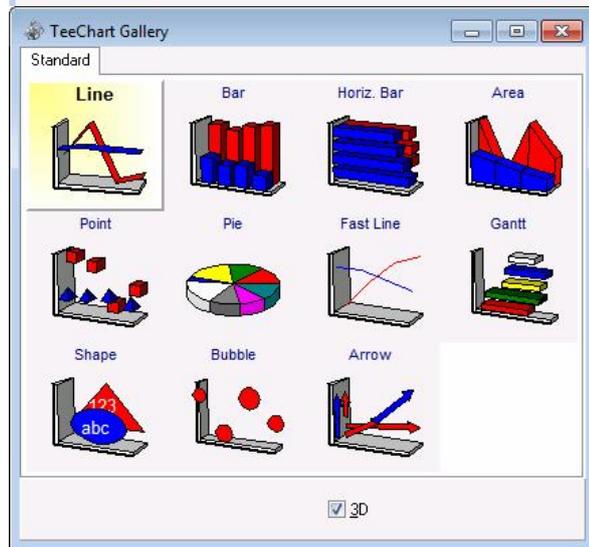
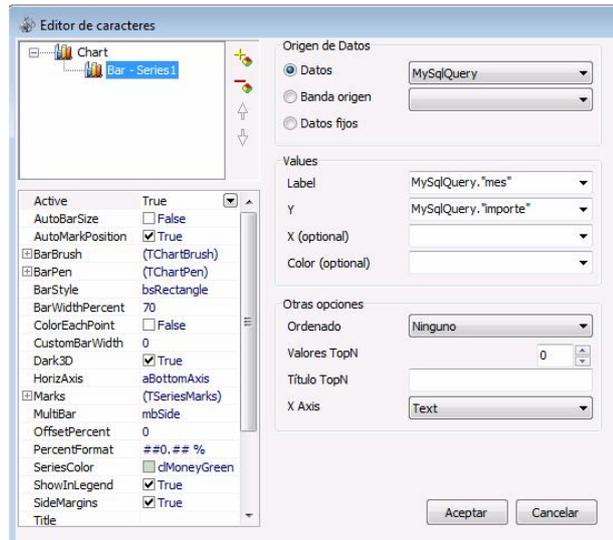
que puesto sobre el diseño nos mostrará el diálogo de edición de gráficas, dentro del cual podremos seleccionar el tipo de gráfica deseado, pulsando sobre el botón  en modo 3D o 2D.

A continuación debemos alimentarla con labels para el eje X y valores numéricos, en este caso para el eje Y

Se pueden cambiar características como la profundidad del efecto 3D, el color, la situación de la leyenda, etc.

**NOTA:** No se pueden utilizar funciones para los datos a representar, han de venir calculados previamente.

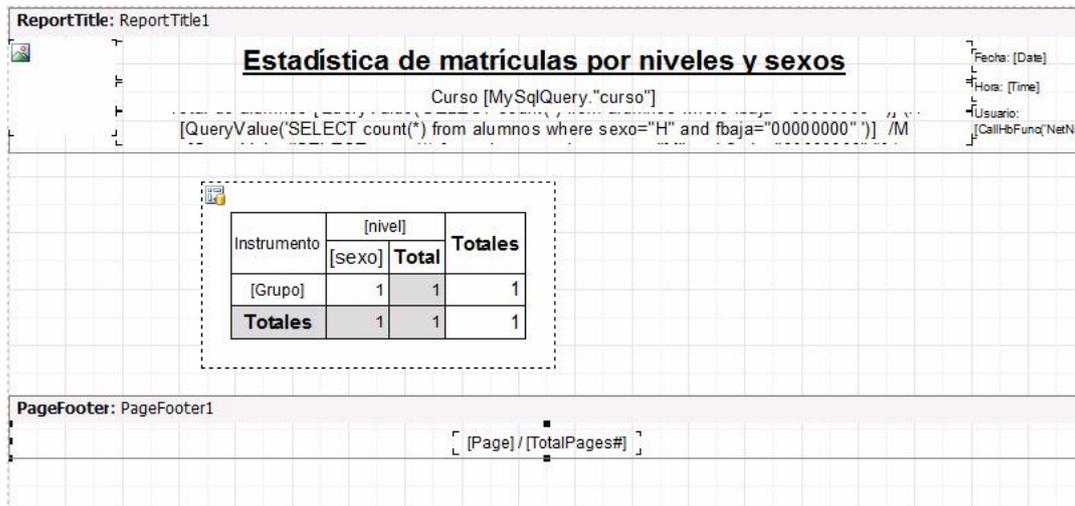
Así el resultado se mostrara como:



## Informe de tablas cruzadas

Otro tipo de elemento a insertar en un informe es la tabla cruzada de datos. Este elemento es extremadamente útil, ya que permite tomar una serie de datos de un dataset y cruzarlos para obtener una tabla con los resultados muy elaborados y con un mínimo esfuerzo.

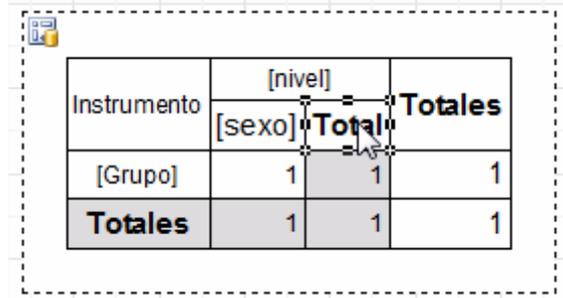
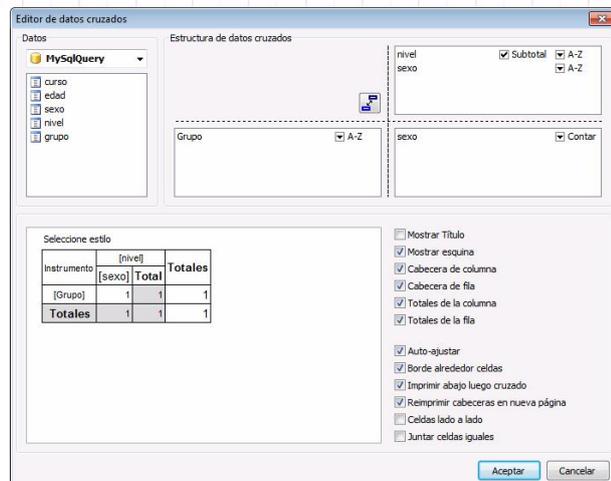
Veamos el siguiente diseño, se trata de obtener para los alumnos de un colegio el total de alumnos por nivel escolar y dentro de cada nivel separarlos por sexos:



Para ello además del título y el pie habituales insertamos un componente  de datos cruzados, dentro del cual indicamos que campo/s ira al eje vertical (en este caso grupo), que campo/s al horizontal (en este caso nivel y sexo) y en el centro que operación queremos hacer con que campo, en este caso contar personas.

Podemos indicar un orden tanto en vertical como en horizontal.

Una vez aceptado también se pueden cambiar textos de totales, fuentes, sombreado de celdas, formateado de números, etc desde el exterior, seleccionando cada celda concreta o en grupos de celdas.



Y el resultado es instantáneo tan sencillo como previsualizar y FR se encarga de todos los cálculos y de la presentación:

**Estadística de matrículas por niveles y sexos**

Curso 0  
Total de alumnos 677 (H 268 /M 409 )

Fecha: 31/05/2013  
Hora: 7:15:03  
Usuario:  
SERVIDOR

Instrumento	0			2			3			Totales
	H	M	Total	H	M	Total	H	M	Total	
BATERIA			9			9				9
BIOLINA			13	23	36	1	7	8		44
BIOLONTXELOA			4	8	12	1	1			13
FLAUTA			6	18	24	2	2			26
KITARRA			22	19	41					41
KITARRA ELEKTRIKA				12	12					12
KLARINETEA			10	9	19	1	2	3		22
PIANOA			26	52	78	2	14	16		94
SAXOFOIA			8	1	9	2	1	3		12
SOINU TXIKIA			9	28	37	1	2	3		40
SOINUA			7	21	28	3	3	3		31
TRONPETA			3		3	1		1		4
TXISTUA			3	3	6	2	1	3		9
autoc-tonos	2	12	14							14
contacto	89	83	172							172
danza	83	198	281							281
solo conj. instrumental	2	2	4							4
<b>Totales</b>	<b>176</b>	<b>295</b>	<b>471</b>	<b>132</b>	<b>182</b>	<b>314</b>	<b>10</b>	<b>33</b>	<b>43</b>	<b>828</b>

Esta herramienta es muy potente ya que obtener con ella algo a priori tan complejo como un informe de los viajes realizados por una empresa de transportes, por cada camión y chofer a cada obra de cada cliente, es cosa de un momento.

**Informe de servicios por clientes y obras**

31/05/2013  
7:22:06

Desde 01-04-2013 Hasta 30-04-2013  
(Solo son mostrados cambios y retiradas)

	0019CVN Angel Hernan	0059B-GV Jon Imanol S	0804FLB Fer Perez Me	0863HDL Arioste Lope	1424GFV Ester Perez	1447HCY Rubin Perez	3110GJN Fernando Per	3240BZL Jon Pérez Sá	8340GNC Luis Dicampo	B60598F Guzas Goite
AFESAMEDIO AMBIENTE, S.A	MORGAN*98 BILBAO				2	1				
	Total	0	0	0	2	1	0	0	0	0
ALRA S.L. CONSTRUCCIONES Y REFORMAS	BALLONTI PORTUGALETE				1					
	Total	0	0	0	1	0	0	0	0	0
ALDECO CONSTRUCCIONES MODULARES, S.A.	POL. ARIANDI DURANGO	1					1			
	Total	1	0	0	0	0	1	0	0	0
ALDUILERES GALINDO, S.L. (GALIN)	B* GALINDO TRAPAGARAN									1
	Total	0	0	0	0	0	0	0	0	1
AMANTO Y GESTION, S.L.U.	SIMÓN BOLIVAR 27 BILBAO	1			1	1			3	
	Total	1	0	0	1	1	0	0	3	0
ARQUIMIDER CONSTRUCCION EN MADERA S.L.	B*MARTIO UGAO	1								
	Total	1	0	0	0	0	0	0	0	0
ASTAR	SANGRONZ SONDIKA	2								
	Total	2	0	0	0	0	0	0	0	0
	B*MALLOAPE SIN MURUETA	2								
ASTILLEROS DEMURUETA S.A.	JOSE LUIS GOIOAGA 33 ERANDIO	6	4				1	1	3	
	Total	8	4	0	0	0	1	1	3	0
ASTILLEROS ZAMACONA S.A.	PUERTO SANTURTZI	4					7		2	
	Total	4	0	0	0	0	7	0	2	0
ATEA DISEÑO INTERIOR SL	MICARREDO N*35 BILBAO	1					2			
	Total	1	0	0	0	0	2	0	0	0
AYUNTAMIENTO DE TRAPAGA	CEMENTERIO TRAPAGA	1								
	Total	1	0	0	0	0	0	0	0	0
BARACONS S.L.	CARAGÓN N*4 BARAKALDO									1
	LURKEZAGA BARAKALDO									2
	PÉREZ GALDÓS 21 BILBAO									1
	Total	0	0	0	0	0	0	0	0	4
BIOK, OBRAS Y PROYECTOS S.L.	KASUNE N*1 ALGORITA				2					1
	LOZOÑO OTZARKOAGA				2	1				
	UGARTE 6 - OTZARKOAGABILBO									1
	Total	0	0	0	4	1	0	0	0	2

Página 1 de 20

### Impresión de documentos sobre plantilla escaneada

A veces se plantea la necesidad de utilizar FR para rellenar documentos oficiales que usan papel preimpreso, entonces la mejor solución pasa por escanear estos documentos de una o varias páginas (cada página será una página de diseño diferente de FR) y ponerlos como fondo de diseño en FR utilizando la propiedad BackPicture de la página, de forma que después se puedan situar encima los datos a imprimir en su lugar correcto,

En el siguiente diseño se puede observar el fondo escaneado y puesto como BackPicture sobre el que se han situado variables enviadas desde nuestro programa.

ACTIVO		NOTAS DE LA MEMORIA	EJERCICIO [NEje] (2)	EJERCICIO [NEje] (3)
Espacio destinado para las firmas de los administradores				
A) ACTIVO NO CORRIENTE	11000		[AA_2]	[AA_2]
I. Inmovilizado intangible	11100		[AA1]	[AA1_2]
II. Inmovilizado material	11200		[AA2]	[AA2_2]
III. Inversiones inmobiliarias	11300		[AA3]	[AA3_2]
IV. Inversiones en empresas del grupo y asociadas a largo plazo	11400		[AA4]	[AA4_2]
V. Inversiones financieras a largo plazo	11500		[AA5]	[AA5_2]
VI. Activos por impuesto diferido	11600		[AA6]	[AA6_2]
VII. Deudores comerciales no corrientes	11700			
B) ACTIVO CORRIENTE	12000		[AB]	[AB_2]
I. Existencias	12200		[AB2]	[AB2_2]
II. Deudores comerciales y otras cuentas a cobrar	12300		[AB3]	[AB3_2]

A la hora de imprimir se puede imprimir tal cual con fondo o si se va a utilizar el papel preimpreso simplemente se indica False en la propiedad BackPicturePrintable y solo se imprimirán los datos, aunque en pantalla quedará perfecto si se indica True en BackPictureVisible.

ACTIVO		NOTAS DE LA MEMORIA	EJERCICIO 2013 (2)	EJERCICIO 2012 (3)
Espacio destinado para las firmas de los administradores				
A) ACTIVO NO CORRIENTE	11000		19.200,00 €	19.200,00 €
I. Inmovilizado intangible	11100		0,00 €	0,00 €
II. Inmovilizado material	11200		-0,01 €	-0,01 €
III. Inversiones inmobiliarias	11300		0,00 €	0,00 €
IV. Inversiones en empresas del grupo y asociadas a largo plazo	11400		0,00 €	0,00 €
V. Inversiones financieras a largo plazo	11500		0,00 €	0,00 €
VI. Activos por impuesto diferido	11600		0,00 €	0,00 €
VII. Deudores comerciales no corrientes	11700			
B) ACTIVO CORRIENTE	12000		22.400,00 €	22.400,00 €
I. Existencias	12200		0,00 €	0,00 €
II. Deudores comerciales y otras cuentas a cobrar	12300		9.600,00 €	9.600,00 €
1. Clientes por ventas y prestaciones de servicios	12380		0,00 €	0,00 €
a) Clientes por ventas y prestaciones de servicios a largo plazo	12381			
b) Clientes por ventas y prestaciones de servicios a corto plazo	12382		0,00 €	0,00 €
2. Accionistas (socios) por desembolsos exigidos	12370		0,00 €	0,00 €
	12300		19.208,32 €	19.208,32 €

## Informe sin DataSets, impresión de un recibo.

Como se ha visto antes en este manual se puede utilizar FR sin que haya banda alguna, ya que aquellos campos del dataset que se dispongan sobre la hoja de imprimir para el registro en curso, pero y si además ni siquiera tenemos un registro o no queremos utilizar un dataset.

Veamos este caso, desde un Form queremos imprimir un recibo con unos datos concretos que incluso han sido calculados en el momento de generar el form y no están en ningún registro, pues simplemente se los enviamos a FR a modo de variables y listo.

```
With object frReportManager():New()
  If !:lError
    :aVariables:={ { "nRecibo", ::oRS:Id },;
                  { "nImporte", nImporte },;
                  { "cFormaCobro", ::oTipoCobro:Value },;
                  { "cNumCta", ::oNcta:Value },;
                  { "cTitular", ::oRS:Titular },;
                  { "cAlumno", Appdata:oSqlSource:QueryValue("SELECT '<b>',alumnos. ....
                  { "cDireccion", Appdata:oSqlSource:QueryValue("SELECT calles.nombre .....
                  { "cPoblacion", Appdata:oSqlSource:QueryValue("SELECT callejero.cp,'...'
                  { "cFechafactura", dToT(::oRs:FechaRemesa) },;
                  { "cFechaVencimiento", dToT(::oRs:FCobro) } }
    :LoadFromFile(Appdata:cDataReport+"Recibo.fr3")
    :Run("P",!Appdata:lUser)
  Endif
End With
```

De esta forma realizamos un diseño como el siguiente

Donde cada valor a imprimir nRecibo, nImporte, etc. es una variable sin necesidad de un dataset.

Fijémonos en un detalle, hay un área sombreada que contiene un texto “Recibo por importe de:” y un campo calculado:  
 [NameCase(CallHbFunc('NumToLetras',[<nImporte>,NULL,'Euros','Centimos'])))

Primero NameCase es una función de FR que pone los primeros caracteres de cada letra en Mayúsculas y el resto en minúsculas, va entre [ ] ya que ha de ser evaluada y escrito su resultado. Dentro hay otra función de FR CallHbFunc que se usa para llamar a una

función de Harbour o de nuestro programa que es NumToLetras que en realidad es una función de nuestro fuente que sirve para convertir números a un texto que expresa ese número. Nótese que el nombre de esta función es el primer parámetro entre ‘ de CallHBFunc.

**FUNCTION NumToLetras(nValue, lMale, cDescUnidad, cDescDecimal, lRecursive)**

Como se ve que hay que pasarle unos parámetros y estos van desde FR a nuestro fuente como un array de valores entre [ ] siendo [**nImporte**],**NULL**,**'Euros'**,**'Centimos'**] El primero de estos parámetros es a su vez uno de los valores recibidos como variables, el importe del recibo que es un valor numérico y que hemos de indicar por lo tanto con < > ya que estamos usando a su vez [ ] para el array de parámetros. El segundo parámetro es el género masculino o femenino del texto resultante que puede ser TRUE o FALSE (sintaxis pascal) o si no por defecto lo tomará la función si indicamos NULL por que **no podemos dejarlo vacío**.

**NOTA:** antes de previsualizar con F9, es mejor grabar el diseño por que si hay cualquier error al llamar a CallHBFunc en el código fuente de la función Harbour, el programa dará un error como es lógico y se cerrará, perdiendo el diseño al completo.

Otro detalle interesante está en el cuadro “Localidad de expedición” dentro del cual puede verse:  
 [QueryValue('SELECT nombre FROM m\_codigos.municipios WHERE id=(SELECT idmunicipio FROM configuracion LIMIT 1)')]

La función Queryvalue no es una función original de FR, pero tampoco vemos un CallHBFunc. En realidad es una función externa que incorporamos a FR al ejecutarlo con el método **AddFunction** descrito antes en la lista de métodos de FR.  
 oFr::AddFunction("function QueryValue(Select: String ): Variant","MySql","Ejecuta una sentencia Select que devuelve un único valor")

Se encarga de hacer un QueryValue sobre la base de datos MySql en curso, captar el valor que necesitamos e imprimirlo es esa posición.

RECIBO NÚMERO 69550	LOCALIDAD DE EXPEDICIÓN MUNGIA	IMPORTE ***50,15 €***
FECHA DE EXPEDICIÓN 4 de abril de 2.013	VENCIMIENTO 4 de abril de 2.013	

Recibo por importe de:  
 Cincuenta Euros Con Quince Centimos\*\*\*\*\*

Nº Cta 3084-0013-06 073629 Titular JAIONE

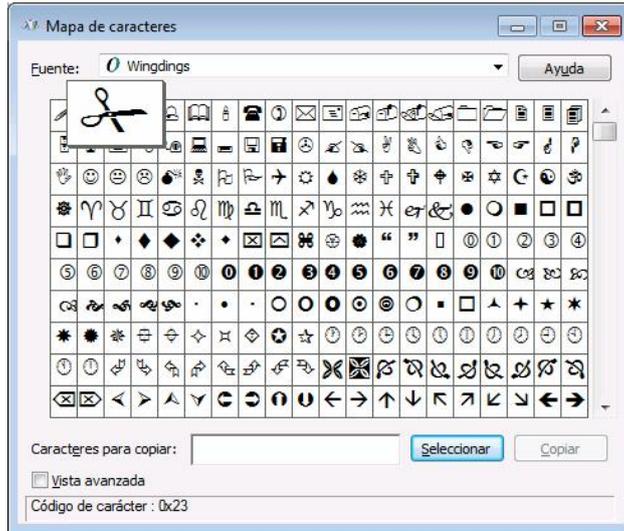
NOMBRE Y DOMICILIO DEL PAGADOR

**Conchita**  
 ELORDUIGOITIA, -  
 48100 MUNGIA - BIZKAIA



Y para terminar ponemos una línea de puntos con una tijera ya que al ser impreso en un DIN A4 hay que recortarlo. La línea no es problema por que la pinta FR pero la tijera, que en otros programas no sería un problema puesto que basta con usar un fuente Windings nativo de Windows, en FR si lo es por que los fuentes de símbolos no aparecen en la lista de fuentes.

A pesar de esto están ocultos y si escribimos su nombre en el cuadro de fuentes aparecen y se pueden utilizar, otra cosa es que sepamos que carácter poner y para esto tenemos la herramienta de Windows llamada “Mapa de caracteres” donde podemos seleccionar y copiar para después pegar en un objeto Memo de FR.



## Informe hoja de calendario

Para imprimir un calendario con un contenido en cada celda y con colores en algunos números como los domingos o las festividades, o incluso coloreando los textos:



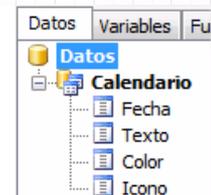
**Calendario del planning Junio de 2013**

03/06/2013  
17:21:45

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
					1	2
3	4	5	6 Reunión de Madrid Controles de Xailer	7 Reunión de Madrid Fast Report	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Tenemos que realizar el siguiente diseño

Y lo que le enviamos es un MemDataSet con los campos que se monta en base a un filtro de un dataset desde el 01 de un mes hasta fin de ese mes.



El problema es que para cuadrar esos días habrá que añadir por delante n registros vacíos teniendo en cuenta en que día comienza el mes, en este caso en sábado y rellenar con registros vacíos los días restantes al final hasta completar  $7 \times 5 = 35$  celdas.

```

If MsgEdit("Seleccione una fecha para obtener una hoja de calendario",
"Calendario del planning", @dDia, "PLANNING")

oFr:cHeadline2:=cMonth(dDia)+" de "+Allstring(Year(dDia))

dIni:=FirstOfMonth(dDia)
dDia:=dIni
nVacios:=If(Dow(dIni)=1,6,Dow(dIni)-2)
dFin:=LastOfMonth(dDia)

//Primeros días vacios
For nItem:=1 to nVacios
  Aadd(aData,{,"","black",0})
Next
//Días rellenos
For nItem:=0 To dFin-dIni
  Aadd(aData,{dDia,TextoDia(oDbfPlanning,dDia), oDbfPlanning:Color ,0})
  dDia++
Next
//Días sobrantes
For nItem:=Len(aData)+1 to 35
  Aadd(aData,{,"","black",0})
Next

oCalendar:lOpen:=.F.
oCalendar:Open(aData,{"Fecha","Texto","Color","Icono"})

Endif

```

De esta manera al poner este MemDataset sobre un MasterData pintamos 7 cuadros en la misma línea, ponemos contenido al primero e indicamos que queremos 7 columnas

Columns	7
ColumnWidth	4,00

de datos

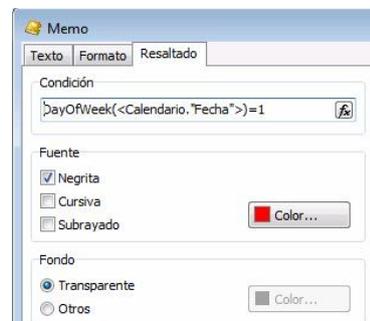
El contenido de cada cuadro se compone de dos memos uno para el texto sin más y otro para el día que tiene como particularidad que cambia de color al color que hemos indicado que queremos en el campo Color del MemDataset.

Su contenido es

**[IIF(ValidDate(<Calendario."Fecha">),DayOf(<Calendario."Fecha">),0)]**

Como se ve se ha usado una función IIF(validDate es decir si es válida la fecha imprime el día de la fecha y si no imprime un cero 0 que luego se oculta indicando que no queremos ver ceros  HideZeros  True

Para que el color varíe los domingos editamos el memo y en el fólder de resaltado indicamos un color rojo y añadimos una función de FR DayOfWeek() que dirá que la condición de color sea que el día de la semana sea el 1 (Domingo)



Otra forma de variar el color es usando Tags de HTML  AllowHTMLTags  True  
 <font color> - color del fuente - </font>

De manera que podemos usar el color recibido del MemDataSet en código HTML escrito dentro del memo <font color=[Calendario."Color"]> [Calendario."Texto"]</font>

## Informe columnar con líneas de altura variable

Una de las funcionalidades más apreciadas de FR es el Stretched, es decir que los campos de las bandas o de la hoja puedan crecer de altura a conveniencia.

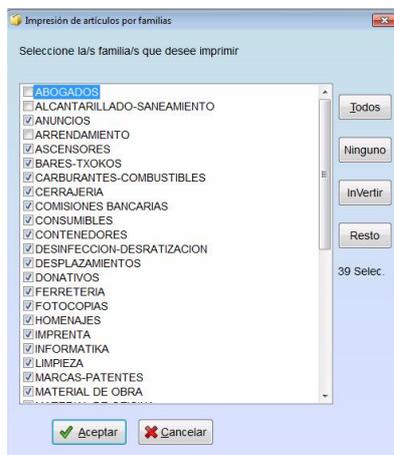
Es muy frecuente añadir columnas que contienen campos Memo o BLOB de gran extensión y por lo tanto un problema a la hora de imprimirlos ya que no se sabe que altura tendrán para cada registro del MasterData y es entonces cuando podemos dar un efecto Stretched.

Veamos el siguiente diseño:

The screenshot shows a report design window with the following bands and fields:

- ReportTitle: ReportTitle1**: Contains the title "Informe de artículos por familias" and fields for Fecha: [Date], Hora: [Time], and Usuario: [User Name].
- GroupHeader: GroupHeader1**: Contains the field "Familia [MySqlQuery."nomfamilia"]" and columns: Código, Descripción, Proveedores, Ult. Precio, Ult. Plazo, Observaciones.
- MasterData: MasterData1**: Contains the main data rows with fields like [MySqlQuery], [NameCase(<MySqlQuery)], [NameCase(IF(<MySqlQuery...)], [MySqlQu], [MySqlQu], [MySqlQuery."observaci].
- GroupFooter: GroupFooter1**: Contains the field "Total de artículos familia [cNomGrupo]:" and "asterData1".
- ReportSummary: ReportSummary1**: Contains the field "Total general de artículos:" and "asterData1".
- PageFooter: PageFooter1**: Contains the field "[Page] / [TotalPages#]".

Este es un informe de artículos por familias que hemos forzado al máximo para que quepa en un DIN A4 vertical.



Además permite seleccionar múltiples familias antes de comenzar y queremos que el nombre de todas ellas se muestren en el subtítulo de la banda ReportTitle1 con lo que este título podrá ser también más largo o más corto dependiendo de las familias que se seleccionen.

Por lo tanto andaremos escasos de sitio y es entonces cuando hay que utilizar Stretched y NameCase().

El resultado impreso de este diseño quedará como sigue:

<b>Informe de artículos por familias</b>					
 Anuncios - Ascensores - Bares-bohios - Carburantes-combustibles - Cerrajería - Comisiones Bancarias - Consumibles - Contenedores - Desinfeccion-desratizacion - Desplazamientos - Donativos - Ferreteria - Fotocopias - Homenajes - Imprenta - Infomatika - Limpieza - Marcas-patentes - Material De Obra - Material De Oficina - Mensajería - Organizaciones - Pintura - Procurador - Productos Quimicos - Publicaciones - Publicidad-propaganda - Registro Propiedad - Reparacion-mantenimiento - Restaurantes - Seguros - Señalización - Subvenciones - Suministros - Tributos - Trofeos - Utiles-herramientas - Varios - Vestuario					Fecha: 06/06/2013 Hora: 7:28:25 Usuario: Administrador
<b>Familia ANUNCIOS</b>					
Código	Descripción	Proveedores	Ult. Precio	Ult. Plazo	Observaciones
ANUNC0001	Periodico-boletines	Diputacion Foral De Bizkaia			publicación de anuncios en periodicos, boletines de bizkaia, pais vasco, estatal
ANUNC0001	Periodico-boletines	Deia Editorial Iparagirre S.a.			publicación de anuncios en periodicos, boletines de bizkaia, pais vasco, estatal
ANUNC0001	Periodico-boletines	Gabinorte R.p. Asesores En Comunicacion S.I.			publicación de anuncios en periodicos, boletines de bizkaia, pais vasco, estatal
Total de artículos familia ANUNCIOS:					3
<b>Familia ASCENSORES</b>					
Código	Descripción	Proveedores	Ult. Precio	Ult. Plazo	Observaciones
ASCEN0001	Ascensor	Zardoya Otis S.a.			rae 23650- Olentzeoren etxea rae 25398- Laukariz ik. rae 18863- udaletxea entrada urbanismo-economica rae 24554- udaletxea entrada principal rae 24060- udaletxea entrada udaltzaingoa
ASCEN0001	Ascensor	Eca, Entidad Colaboradora De La Administracion, S.			rae 23650- Olentzeoren etxea

Como se puede apreciar el subtítulo es grande pero ha entrado por que hemos habilitado su propiedad `StretchMode` `smMaxHeight`. Su valor normal `smActualHeight` es que no varía de altura al igual que `smDontStretch` que no le permite crecer, pero si se usa `smMaxHeight` podrá crecer todo lo necesario. Pero para que esto funcione de verdad hay que seleccionar la banda contenedora, en este caso `ReportTitle1` e indicar también en su propiedad `Stretched`  `True` para que crezca a la par que el campo Memo. Así cabrá sin problemas pero si queremos ahorrar aún mas espacio podemos indicar que aquello que se escriba en este memo sea `[NameCase(<HeadLine2>)]`. La función `NameCase()` hace que el texto sea convertido a minúsculas con la primera letra de cada palabra en mayúsculas, de esta manera aunque en la base de datos las familias están escritas en mayúsculas, quedan mucho mas presentables, legibles y sobre todo ocupan mucho menos espacio en el papel. Esta función es muy útil para imprimir nombres y apellidos entre otras cosas.

Veamos ahora la MasterData. Los campos descripción, nombre del proveedor y observaciones se han indicado como `StretchMode` `smMaxHeight` y la banda en si misma como `Stretched`  `True` con lo que conseguimos que la altura de cada línea o registro sea variable y por lo tanto la altura máxima sea la del campo más largo, habitualmente las observaciones, además hemos aplicado a la descripción y al nombre del proveedor la función `NameCase()` para que ocupen menos, no así a las observaciones que están en minúsculas en la base de datos y que con `NameCase` quedarían bastante mal.

### Impresión de documentos de tipo factura

Cuando salimos del informe columnar estándar lo más habitual es hacer documentos de tipo factura, albarán, etc.

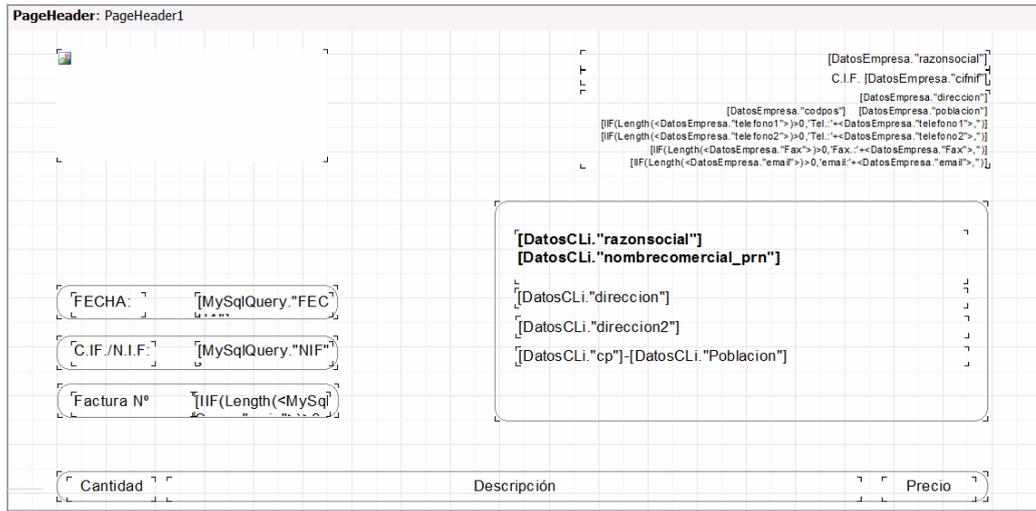
En este caso lo más frecuente es que no se impriman más de un documento, si con múltiples hojas pero solo un documento y por lo tanto vamos a ver como se imprime un documento partiendo de una tabla de cabecera que aportará un único registro y una tabla de detalle que aportará n registros o líneas al cuerpo de la factura.

La tabla maestra se llama en este caso MySQLQuery y la de detalle MySQLQueryDetail y además tenemos otras 2 tablas DatosCli y DatosEmpresa que solo aportan datos de un registro, el cliente de la factura y los datos de la empresa que realiza la factura. Con ellas vamos a componer el siguiente diseño:

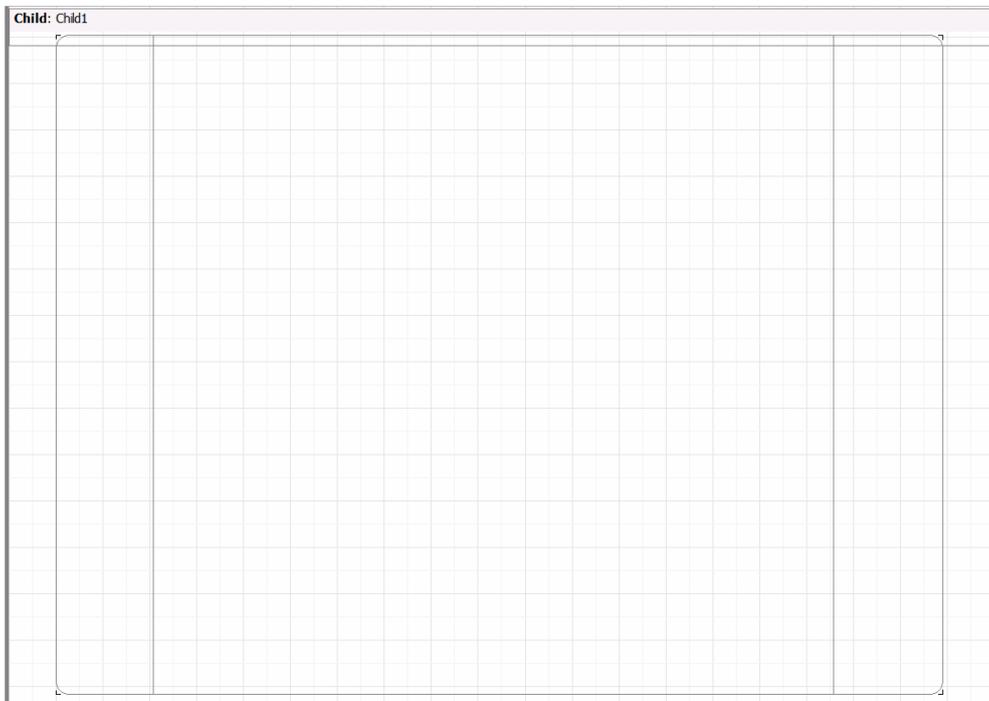
The screenshot shows a FastReport design grid for an invoice layout. The components and their contents are as follows:

- PageHeader: PageHeader1**: Contains a logo and a block of data from `[DatosEmpresa]` including `razonsocial`, `C.I.F.`, `direccion`, `codigos`, `telefono1`, `telefono2`, `telefono3`, `fax`, and `email`. It also includes a block of data from `[DatosCli]` including `razonsocial`, `nombrecomercial_pm`, `direccion1`, `direccion2`, and `cp` concatenated with `Poblacion`.
- Child: Child1**: Contains a table with columns `Cantidad`, `Descripción`, and `Precio`.
- MasterData: MasterData1**: Contains a table with a single column `MySQLQueryDetail`.
- PageFooter: PageFooter1**: Contains a summary table with columns: `Suma importes`, `Dts.`, `Base Imponible`, `% IVA`, `%Rec`, `Importe I.V.A.`, and `TOTAL FACTURA`. The `Base Imponible` column has sub-rows for `p0`, `mp1`, `mp2`, and `mp3`. The `Importe I.V.A.` column has sub-rows for `dva1`, `dva2`, and `dva3`. The `TOTAL FACTURA` column contains the formula `doc">,"`. Below the summary table is a section for `FORMA DE PAGO` and `VENCIMIENTOS` with sub-rows for `FechaVto1`, `FechaVto2`, and `FechaVto3`, each with corresponding `ImpVto` values. A SQL query is also visible: `FROM fomaspago WHERE`.

Se puede observar que la mayor parte de los campos están situados dentro de un PageHeader, es decir el logotipo de la empresa y los campos de DatosEmpresa de la cabecera, los datos del cliente para la ventanilla del sobre de DatosCli y algunos de la cabecera de factura como la fecha, el nif y el número de factura de MySQLQuery.



A continuación tenemos una banda hija de la anterior Child1, una banda hija siempre irá a continuación de la banda padre a la que pertenece y se crea desde la banda padre pulsando clic derecho del ratón y seleccionando “Añadir child Band”. Sobre esta ponemos el dibujo del contorno de los datos de MySQLQueryDetail con la altura que estimemos conveniente, aunque se puede ver que incluso se salen por debajo de la banda solo cuenta donde esté situado el principio del dibujo y de hecho si la hacemos mas alta la banda de detalle empezará cuando termine esta child y no quedará bien.



Sobre ella a una altura conveniente para el inicio de la impresión de los datos, se sitúa la banda MasterData, que será la encargada de mostrar el detalle de MySQLQueryDetail.

Child: Child1	
MasterData: MasterData1	MySqlQueryDetail
[IIF(<MySql]	[IIF(<MySqlQ]

La banda que usamos es una Masterdata y no una Detaildata que puede parecer más lógico, pero al no usar una Masterdata para los datos principales, ya que si no saldrían las n cabeceras de la tabla Maestra, usamos una MasterData para el detalle para que salgan en columnas los n registros de la tabla de Detalle asociados a esta cabecera.

El campo del medio que parece estar vacío no lo está por que en realidad es un RichText que viene como tal de la BBDD pero esto no se puede indicar sobre el diseño sino en las propiedades del campo.

DataField	Descripcion
DataSet	MySqlQueryDetail

Aunque lo más importante de este campo es la propiedad **StretchMode** `smMaxHeight` que indica que el campo puede variar su altura a conveniencia ya que al ser un texto RTF de “longitud ilimitada” puede ser bastante largo, pero ojo cuando se usa un memo Stretched hay que indicar a la banda anfitriona que ha de ser **Stretched**  `True` para que crezca también con sus campos Stretched.

Y por último estarán las totalizaciones de la factura que se sitúan sobre una banda de tipo PageFooter y así saldrán forzosamente al final de cada página.

PageFooter: PageFooter1						
[Suma importes]	[Dtos.	[Base Imponible]	[% IVA	[%Rec	[Importe I.V.A.	[TOTAL FACTURA]
[Query."bruto">,")]	[p">,"),")]	[p0">,")]	[,")]	[,")]	[civa1">,")]	[doc">,")]
		[mp1">,")]	[,")]	[,")]	[civa2">,")]	
		[mp2">,")]	[,")]	[,")]	[civa3">,")]	
		[mp3">,")]	[,")]	[,")]		
[FORMA DE PAGO]		[VENCIMIENTOS]		[FechaVto1]	[ImpVto1]	
[Query."bruto">,")]		[Query."bruto">,")]		[FechaVto2]	[ImpVto2]	
[Query."bruto">,")]		[Query."bruto">,")]		[FechaVto3]	[ImpVto3]	

Y el resultado será este:

La altura real de las líneas de MySQLQueryDetail está limitada por el inicio de la banda MasterData y la altura hasta el PageFooter y si no hay suficiente espacio para las líneas, se pasará a una nueva hoja el resto, pero por supuesto en la nueva hoja también se imprimirán de nuevo el PageHeader y el Page Footer.

Esto puede ser un problema no para la cabecera que se repite sino para el pie que daría totales parciales e incorrectos en todas las páginas salvo en la última. Para evitarlo se introduce una condición

`[IIF(<Page>=<TotalPages>, x, ‘’)]`

en todos los campos del pagefooter para que sean impresos solo en la última página.

The screenshot shows a BISOFT invoice report. At the top left is the BISOFT logo. At the top right is the company information: Juan Jose Garcia Diaz, J. Ingen. Udaldekoa Erosketa C.B., C.I.F. B36673038, C/ Arrieta Nazkaletxe 47, 2º. 48001 Bilbao, Tel. 944232921, www.bisoft.es.

Below the logo are three input fields: FECHA: 03.06.2013, C.F./N.I.F.: F. 102, and Factura Nº: F/102. To the right is a box for 'AYUNTAMIENTO DE' with some redacted text.

The main part of the report is a table with columns 'Cantidad', 'Descripción', and 'Precio'. It contains two rows of items:

Cantidad	Descripción	Precio
1	Software de gestión Muska Eskola - Instalación - Conversión de datos program a antiguo - Curso de utilización - Mantenimiento hasta el 31-12-2014	8.750,00
1	Paquete de mensajes 5000 SMS	440,00

At the bottom of the report is a summary table:

Suma importes	Dibos.	Base Imponible	% IVA	%Rec	Importe I.V.A.	TOTAL FACTURA
9.190,00		9.190,00	21,00		1.929,90	11.119,90

Below the summary table is a section for 'FORMA DE PAGO' and 'VENCIMIENTOS'. The form type is 'AL CONTADO' and the due date is '03.06.2013' for a total amount of '11.119,90'.

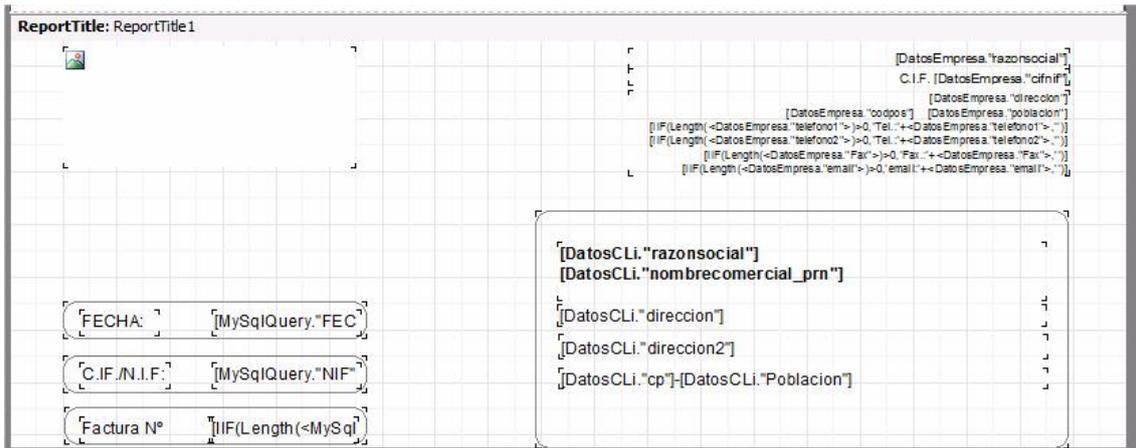
Quizás esta no sea la manera más eficiente de generar una factura ya que lo ideal sería que la cabecera solo apareciera en la primera hoja, que los totales solo estuvieran en la última y que entre tanto el detalle ocupe todo el espacio disponible. Para esto hay que entrar en un diseño algo más complejo como el siguiente:

The screenshot displays the FastReport 4 design tool interface for an invoice report. The design is organized into several distinct bands:

- ReportTitle:** Contains a logo and a large text area for company information, including fields for [DatosEmpresa."razonsocial"], C.I.F., [DatosEmpresa."dirci"], and various phone numbers with conditional logic like [IF (Length(<DatosEmpresa."telefono1">)>0, Tel. <=>DatosEmpresa."telefono1">)].
- Client Information:** A separate box containing fields for [DatosCLI."razonsocial"], [DatosCLI."nombrecomercial\_prn"], [DatosCLI."direccion"], [DatosCLI."direccion2"], and [DatosCLI."cp"]-[DatosCLI."Poblacion"].
- Form Fields:** Three input fields for 'FECHA:', 'C.I.F./N.I.F.', and 'Factura N°' are linked to [MySqlQuery."FEC"], [MySqlQuery."NIF"], and [IF (Length(<MySql)] respectively.
- PageHeader:** A simple header band.
- Child Bands:** Two table-like bands (Child1 and Child2) with columns for 'Cantidad', 'Descripción', and 'Precio'. Child2 also includes 'Fecha:', [MySqlQuery.""], and 'Factura N°'.
- MasterData:** A band with a yellow background containing [IF(<MySql] and [IF(<MySqlQ].
- PageFooter:** Includes 'Pág. [Page]' and a summary box with 'Suma y sigue ...' and [SUM(<MySqlQuery].
- Footer:** A detailed summary table with columns: 'Suma importes', 'Dtos.', 'Base Imponible', '% IVA', '%Rec', 'Importe I.V.A.', and 'TOTAL FACTURA'. It lists 'seimp0' through 'seimp3' and 'arciva1' through 'arciva3'. Below this is a 'FORMA DE PAGO' field linked to a query and a 'VENCIMIENTOS' section with fields for [FechaVto1], [ImpVto1], [FechaVto2], [ImpVto2], and [FechaVto3], [ImpVto3].

Como se puede ver a simple vista, es un diseño de una complejidad mayor ya que contiene un gran número de bandas diferentes. Para esto lo primero que hemos de hacer es no limitar la altura de diseño de la página a un DIN A4, pulsamos doble clic en el fondo de la hoja y en las Opciones de página – Otras opciones seleccionamos  Gran altura en modo diseño lo cual nos da una amplitud casi ilimitada para diseñar sin afectar a la impresión final del documento que seguirá siendo un DIN A4.

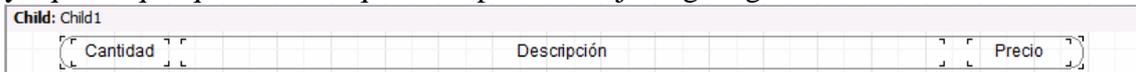
En primer término tenemos una banda ReportTitle que tiene como característica principal que solo se muestra en la primera página y con esto ya nos quitamos el primer problema de encima.



A continuación tenemos una banda PageHeader que aparecerá en cada página por encima de los datos de la MasterData, pero está vacía:



Por que no queremos que sea ella la que se imprima sino una de sus dos bandas Child, ya que lo que queremos es que en la primera hoja salga algo breve como:



Y en la segunda página y sucesivas, algo más complejo con el logo en miniatura, la fecha y número de factura, como:



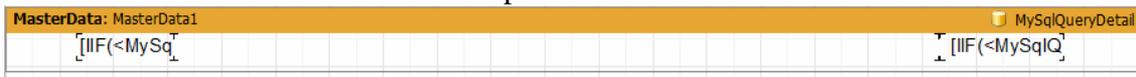
Para esto tenemos que hacer varias cosas, la primera aplastar la altura del PageHeader para que no ocupe espacio por si mismo ya que si no saldría en la hoja pero vacío (aunque podría contener datos y se imprimiría además de una de las Child), la segunda es recurrir a un poco de código Pascal, como puede deducirse por el punto rojo del PageHeader.

```
procedure PageHeader1OnBeforePrint(Sender: TfrxComponent);
begin
    If <Page>=1 then PageHeader1.Child:=Child1 else PageHeader1.Child:=Child2
end;
```

Lo que indicamos es que si la página es la 1 el Child del PageHeader es Child1 y si no es el Child2. Así de simple aparece el uno o el otro en función de un **If**

**NOTA:** Puede haber múltiples child para casi cualquier banda de FR, algunas no lo soportan.

A continuación nos encontramos con la banda MasterData que al igual que antes tiene tres columnas siendo la central un componente RichText.



Debajo de esta tenemos una banda PageFooter que por definición sale en la parte de debajo de todas las páginas del documento, pero mediante programación y preguntando por el número de página podríamos hacer que saliera o no, aunque hay algo más sencillo

que es manipular su propiedad PrintOnLastPage como False  PrintOnFirstPage  True  PrintOnLastPage  False y saldrá en todas las páginas menos en la última.

Para terminar tendremos una banda Footer que sale en la última página, pero pegada a la última línea de MasterData y queremos que vaya también al final de la hoja como el PageFooter, queda mas elegante.

Suma importes	Dtos.	Base Imponible	% IVA	%Rec	Importe I.V.A.	TOTAL FACTURA
	p">,"))	seimp0"			areciva1">	otaldoc"
		seimp1"			areciva2">	
		seimp2"			areciva3">	
		seimp3"				
FORMA DE PAGO		VENCIMIENTOS		[FechaVto1]	[ImpVto1]	
[QueryValue("SELECT nombre FROM formas_pago WHERE id="+VarToStr(<MySqlQuery.Idformapago>))]				[FechaVto2]	[ImpVto2]	
				[FechaVto3]	[ImpVto3]	

Para esto recurrimos de nuevo al código Pascal para indicar que:

```
procedure Footer1OnBeforePrint(Sender: TfrxComponent);
begin
  Engine.CurY := engine.PageHeight-Footer1.Height-50 ;
end;
```

Engine.CurY es una variable de FR que dice cual es la coordenada Y (vertical) en la que está imprimiendo la impresora y lo que le decimos es que se desplace hasta la altura total de la hoja Engine.PageHeight menos la altura del propio Footer y le restamos unos 50 mm más. Esto último parece no tener sentido pero si no hacemos saltará a la siguiente página por que en la última página aunque no se imprime, por que así lo hemos pedido al PageFooter, guarda espacio para el y no puede pintar el Footer en ese espacio. En realidad estamos pintando el Footer un poco más arriba.

Pero hay que hacer un último ajuste, que pasa si al restar a Engine.CurY nos vamos para arriba y machacamos el texto de la MasterData, pues que será mejor calcular si nos estamos metiendo en el área escrita y en ese caso saltar de página.

```
procedure Footer1OnBeforePrint(Sender: TfrxComponent);
begin
  If Engine.CurY>engine.PageHeight-Footer1.Height-50 Then Engine.NewPage;
  Engine.CurY := engine.PageHeight-Footer1.Height-50;
end;
```

Así el resultado será el siguiente en caso de tener solo una hoja:

**BISOFIT**  
 Juan José García Glez. y J. Ingen. Ugararte Etxebarria C.B.  
 C.I.F. 69670336  
 C/ Alameda Ibañeta 47, 3º Dto 5  
 48004 - BILBAO  
 Tel: 94422020  
 email: bisofit@bisofit.es www.bisofit.es

**PROSINOR, S.L.**  
 POLIGONO INDUSTRIAL DE UGARTE, S/N  
 48510-Trapagaran

FECHA: 03/06/2013  
 C.I.F./N.I.F.: B48789655  
 Factura Nº F / 104

Cantidad	Descripción	Precio
2,5	06-05 Modificaciones informe de pagos y cobros Prosinor y Hierros Barral	45,00
5,5	07-05 Modificaciones Prosinor	45,00
1,5	07-05 (Tarde) Modificaciones Prosinor	45,00
5,08	05 Modificaciones WProsinor	45,00
5,14	05 Modificaciones Prosinor	45,00
5,21	05 Modificaciones Prosinor Bascula	45,00
3,22	05 Modificaciones Prosinor Bascula	45,00
5,28	05 Modificaciones Prosinor Bascula y Conta+	45,00
5,30	05 Modificaciones Prosinor Bascula	45,00

Suma importes	Dtos.	Base imponible	% IVA	%Rec	Importe I.V.A.	TOTAL FACTURA
1.687,50		1.687,50	21,00		354,38	<b>2.041,88</b>

FORMA DE PAGO: RECIBO A 30 DIAS  
 VENCIMIENTOS: 05/07/2013 2.041,88

Y este otro en el caso de tener más de una hoja:

**BISOFIT**  
 Juan José García Glez. y J. Ingen. Ugararte Etxebarria C.B.  
 C.I.F. 69670336  
 C/ Alameda Ibañeta 47, 3º Dto 5  
 48004 - BILBAO  
 Tel: 94422020  
 email: bisofit@bisofit.es www.bisofit.es

**PROSINOR, S.L.**  
 POLIGONO INDUSTRIAL DE UGARTE, S/N  
 48510-Trapagaran

FECHA: 03/06/2013  
 C.I.F./N.I.F.: B48789655  
 Factura Nº F / 104

Cantidad	Descripción	Precio
2,5	06-05 Modificaciones informe de pagos y cobros Prosinor y Hierros Barral	45,00
2,506	05 Modificaciones informe de pagos y cobros Prosinor y Hierros Barral	45,00
2,506	05 Modificaciones informe de pagos y cobros Prosinor y Hierros Barral	45,00
2,506	05 Modificaciones informe de pagos y cobros Prosinor y Hierros Barral	45,00
2,506	05 Modificaciones informe de pagos y cobros Prosinor y Hierros Barral	45,00
5,5	07-05 Modificaciones Prosinor	45,00
5,5	07-05 Modificaciones Prosinor	45,00
5,5	07-05 Modificaciones Prosinor	45,00
5,5	07-05 Modificaciones Prosinor	45,00
5,5	07-05 Modificaciones Prosinor	45,00
5,5	07-05 Modificaciones Prosinor	45,00
1,5	07-05 (Tarde) Modificaciones Prosinor	45,00
1,5	07-05 (Tarde) Modificaciones Prosinor	45,00
1,5	07-05 (Tarde) Modificaciones Prosinor	45,00
1,5	07-05 (Tarde) Modificaciones Prosinor	45,00
1,5	07-05 (Tarde) Modificaciones Prosinor	45,00
5,08	05 Modificaciones WProsinor	45,00
5,08	05 Modificaciones WProsinor	45,00
5,08	05 Modificaciones WProsinor	45,00
5,08	05 Modificaciones WProsinor	45,00
5,14	05 Modificaciones Prosinor	45,00
5,14	05 Modificaciones Prosinor	45,00
5,14	05 Modificaciones Prosinor	45,00
5,14	05 Modificaciones Prosinor	45,00
5,21	05 Modificaciones Prosinor Bascula	45,00
5,21	05 Modificaciones Prosinor Bascula	45,00
5,21	05 Modificaciones Prosinor Bascula	45,00
5,21	05 Modificaciones Prosinor Bascula	45,00
5,21	05 Modificaciones Prosinor Bascula	45,00
5,21	05 Modificaciones Prosinor Bascula	45,00

Suma importes	Dtos.	Base imponible	% IVA	%Rec	Importe I.V.A.	TOTAL FACTURA
1.687,50		1.687,50	21,00		354,38	<b>2.041,88</b>

FORMA DE PAGO: RECIBO A 30 DIAS  
 VENCIMIENTOS: 05/07/2013 2.041,88

Página 1/2 Suma y sigue 1.102,50 €

En este caso ya no hay que estar preguntando para cada componente memo del Footer la condición de impresión [IIF(<Page>=<TotalPages>, x, ‘’)] ya que este Footer solo se imprime al final.

En el PageFooter hemos incluido el número de página y lo hemos hecho dentro de un memo que se llamará Page  que combina un texto y una variable de FR.

Le damos un fondo gris, pero para que no resulte de tamaño excesivo o escaso le indicamos la propiedad `AutoWidth`  `True`

Además vamos a programar un poco para que solo aparezca en caso de tener el documento tres o más páginas, ya que para una sola no saldrá por que no aparece su banda contenedora, pero para dos páginas tampoco tiene mucho sentido.

Vamos a los eventos de este memo y hacemos doble clic en el `OnBeforePage` para pasar a teclear el siguiente código en Pascal:

```
procedure PageOnBeforePrint(Sender: TfrxComponent);
begin
  If <Totalpages>=3 then Page.Visible:=True else Page.Visible:=False;
end;
```

Como se observa podemos manipular la propiedad `Visible` del memo llamado "Page" y como ésta propiedad, cualquiera de las propiedades de las que dispone cualquier objeto o banda son manipulables desde código sabiendo cuales son sus valores posibles, lo cual es fácil de ver si los inspeccionamos desde el editor de propiedades de FR.

## Informe columnar de tarjetas con código de barras y fotografía

FastReport permite imprimir también imágenes y fotografías bien desde una base de datos o bien desde archivos de imagen en el disco.

También permite imprimir valores numéricos o de texto conteniendo números como códigos de barras.

Para ello confeccionamos el siguiente diseño:



Como se puede ver hemos recurrido a un sistema de 2 columnas con separación en el medio, con una anchura y altura estándar de tarjeta de crédito para adaptarnos a formatos de tarjetas comerciales tipo DecaDry ©.

ColumnGap	2,00
Columns	2
ColumnWidth	8,50

Además de los campos memo Nombre y Apellidos, hemos insertado un campo TfrxPictureBox al que se le alimentan las propiedades DataSet y DataField con un campo BLOB de la tabla de MySql

DataField	foto
DataSet	MySqlQuery

Si en lugar de esto

FileLink `[IIF(<Unica.\"SERV1\">,'.\\Imágenes\Punto naranja.jpg', '')]`

pusiéramos un nombre de

archivo en la propiedad FileLink el archivo a imprimir sería el indicado pero no se vería en diseño y solo se cargaría y mostraría en previsualización o impresión.

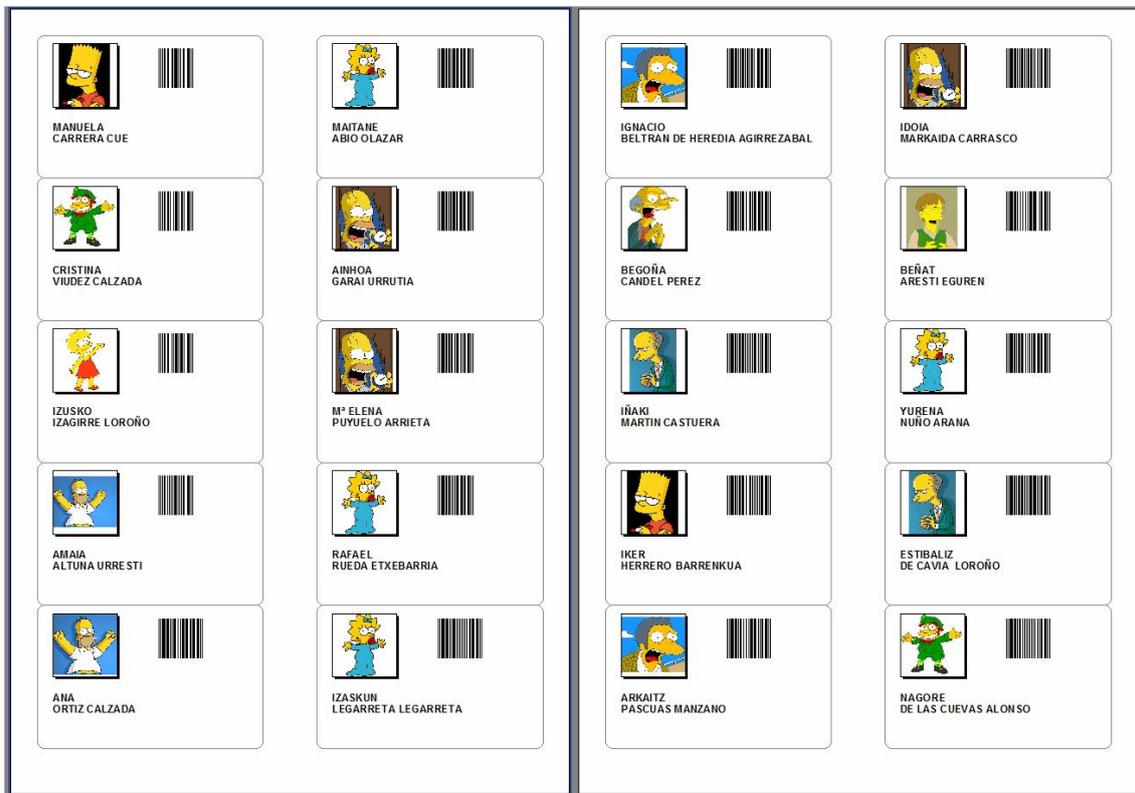
Otra posibilidad es cargar directamente un archivo desde la propiedad Picture y quedaría insertado en el propio FR3 como binario viéndose también en el diseño.

Picture (No asignado)

Por último añadimos un objeto de código de barras TfrxBarcodeView eligiendo el tipo de código a utilizar, indicando que campo de la base de datos lo alimenta y si queremos ver el texto del número debajo o no.



Si previsualizamos el diseño el resultado será el siguiente:



De la misma manera se pueden utilizar imágenes en un informe de columnas mezclando columnas de texto y de imagen.

## Apéndice 1: Atajos del teclado y del ratón en el editor.

### Funciones del teclado

ctrl.+O	Abrir...
ctrl.+S	Salvar
ctrl.+P [F9]	Previsualizar
ctrl.+Z	Deshacer
ctrl.+Y	Rehacer
ctrl.+C	Copiar
ctrl.+V	Pegar
ctrl.+X	Cortar
ctrl.+A	Seleccionar todo, si se encuentra dentro de una banda todo su contenido, si esta sobre la hoja selecciona absolutamente todo.
Ctrl + F	Busca un texto en el diseño y lo selecciona
Flechas del cursor Tabulador	Saltan entre los objetos
Supr	Anula el/los objetos seleccionados
Enter	Llama al editor del objeto seleccionado
Mayúscula+flechas del cursor	Modifica el tamaño de los objetos seleccionados, sirve para cualquier objeto incluso las bandas
Ctrl+ flechas del cursor	Mueve los objetos seleccionados
Ctrl+Mayúscula+ flechas del cursor	Mueve los objetos seleccionados más rápidamente.
Alt+flechas del cursor	El objeto seleccionado se adhiere al más cercano en la dirección especificada.

### Funciones del ratón

Botón izquierdo	Seleccionar objetos y mover objetos. Si pulsa el botón izquierdo mientras mueve el ratón, aparece un rectángulo que va 'capturando' los objetos que quedan dentro, seleccionándolos. Cuando hay varios objetos seleccionado aparece un punto rojo en la zona inferior derecha del último objeto seleccionado que permite escalarlos todos a la vez.
Doble clic del botón izquierdo	Llamada al editor de objetos del objeto sobre el que se pulsa para poder modificarlo. Cuando se hace doble clic en el espacio blanco de la página se llama al diálogo de Configuración de página, para modificar características de la página.
Botón derecho	El menú contextual de objeto seleccionado
Ctrl+Rueda del ratón	Amplia o reduce el zoom sobre el diseño.
Alt+botón izquierdo	Permite editar el texto de un control de texto directamente en su lugar del diseño sin llamar a su editor.

## Apéndice 2: Propiedades de los objetos.

### Propiedades y métodos comunes.

Las siguientes propiedades son comunes a todos los objetos de FastReport:

Propiedad	Tipo	Descripción
BandAlign	Integer	Especifica la alineación de los objetos dentro de la banda. Posibles valores: baNone, baLeft, baRight, baCenter, baWidth, baBottom.
Enabled	Boolean	Determina si un objeto puede responder o no a un evento. Posibles valores: True, False
FillColor	Integer	Determina el color de fondo de un objeto. El color puede especificarse con una constante: clXXX
FrameColor	Integer	Color del marco de un objeto
FrameStyle	Integer	Especifica el estilo de línea del marco de un objeto . Posibles valores: psSolid, psDash, psDot, psDashDot, psDashDotDot, psDouble
FrameTyp	Integer	Tipo de borde de un objeto. Posibles valores: frftTop, frftBottom, frftLeft, frftRight
FrameWidth	Double	Ancho del marco de un objeto
Height	Integer	Altura del objeto
Left	Integer	Determina en píxeles la coordenada x de un objeto, informando de la posición del mismo dentro del informe
Memo	String	Texto del objeto. Se puede acceder a la propiedad haciendo referencia a su índice, por ejemplo: Memo[1]
Memo Count	Integer	Devuelve el número de líneas de la propiedad Memo
Name	String	Nombre del objeto seleccionado
Stretched	Boolean	Determina si el objeto se ajustará ocupando el mínimo espacio imprescindible, por ejemplo: un cuadro de texto respecto a la banda de datos que lo contiene, o si por el contrario mantendrá sus dimensiones
Top	Integer	Determina en píxeles la coordenada y de un objeto
Visible	Boolean	Determina si el objeto será visible o no. Posibles valores: True, False
Width	Integer	Ancho del objeto

Adicionalmente a las propiedades comunes a todos los objetos, cada objeto tiene unas propiedades específicas.

### Objeto "Cuadro de Texto"

Propiedad	Tipo	Descripción
Alignment	Integer	Especifica la alineación del texto dentro del objeto. Posibles valores: frtaLeft, frtaRight, frtaCenter, frtaVertical, frtaMiddle, frtaDown
AutoWidth	Boolean	Determina si el objeto automáticamente modifica sus dimensiones ajustando su tamaño al texto que contiene
CharSpacing	Integer	Determina el espacio entre caracteres
Font.Name	String	Permite establecer una fuente específica para mostrar el texto del objeto
Font.Size	Integer	Tamaño de la fuente en píxeles
Font.Style	Integer	Determina si la fuente tiene alguno de los atributos: negrita, cursiva o subrayado. Posibles valores: fsBold, fsItalic, fsUnderline
Font.Color	Integer	Determina el color de la fuente
GapX	Integer	Determina el margen entre el borde izquierdo del objeto y el texto que contiene.
GapY	Integer	Determina el margen entre el borde superior del objeto y el texto que contiene.
HideZeros	Boolean	Si el valor es True entonces los valores igual a 0 de las variables son ignorados. Posibles valores: True, False
LineSpacing	Integer	Espacio entre las líneas de texto
Suppress Repeated	Boolean	Especifica si los valores repetidos serán mostrados o no. Posibles valores: True, False
TextOnly	Boolean	Determina si las variables que contiene el objeto serán procesadas o no. Posibles valores: True, False
WordBreak	Boolean	Particionado de palabras. Posibles valores: True, False. Ajuste de palabras
WordWrap	Boolean	Posibles valores: True, False

### Objeto "Bandas de datos"

Propiedad	Tipo	Descripción
Breaked	Boolean	Banda partida. Posibles valores: True, False
ChildBand	String	Banda Hija
ColumnGap	Integer	Distancia horizontal entre columnas
Columns	Integer	Número de columnas en la banda
ColumnWidth	Integer	Ancho de la columna
Condition	String	Especifica la condición de agrupación, se aplica a la banda Cabecera de Grupo
DataSource	String	Origen de datos
EOF	Boolean	Determina si se ha alcanzado o no el final del conjunto de datos. Posibles valores: True, False
FormNewPage	Boolean	Fuerza la impresión en una nueva página después de imprimir esta banda y todas sus bandas de detalle. Posibles valores: True, False
Master	String	Especifica qué banda de datos maestros del informe se usará para mostrar los datos agrupados según la condición especificada en la banda Cabecera de Grupo
OnFirstPage	Boolean	Imprimir en la primera página. Posibles valores: True, False
OnLastPage	Boolean	Imprimir en la última página. Posibles valores: True, False
PrintChildIfInvisi	Boolean	Especifica si las bandas de datos hijas se imprimirán o no

ble		si la banda de datos madre es invisible. Posibles valores: True, False
PrintfSubsetEmpt	Boolean	Especifica si se imprimirá o no la banda madre si su banda hija está vacía. Posibles valores: True, False
RepeatHeader	Boolean	Especifica si se repetirá la banda en todas las páginas del informe. Posibles valores: True, False. Aplicable a las bandas: Cabecera de Grupo, Cabecera Maestra, Cabecera de detalle, etc

### Objeto "Imagen".

Propiedad	Tipo	Descripción
BlobType	Integer	Especifica el formato de la imagen contenida en un campo Blob. Posibles valores: btBMP, btJPG, btICO, btWMF.
Center	Boolean	Especifica si se centrará o no la imagen dentro del objeto. Posibles valores: True, False
DataField	String	Especifica el nombre del campo de la base de datos que contiene la imagen.
Keep Aspect	Boolean	Especifica si se mantendrán las proporciones originales de la imagen cuando se modifiquen sus dimensiones. Posibles valores: True, False

### Objeto "Codebar".

Propiedad	Tipo	Descripción
DataField	String	Especifica el nombre del campo de la base de datos que contiene los datos.

### Objeto "CheckBox".

Propiedad	Tipo	Descripción
CheckColor	Integer	Especifica el color de la marca que se muestra si el CheckBox está seleccionado.
CheckStyle	Integer	Especifica el estilo del CheckBox. Posibles valores: csCross, csCheck.
DataField	String	Nombre del campo de la base de datos que contiene los datos.

### Objeto "RichText"

Propiedad	Tipo	Descripción
GapX	Integer	Margen entre el borde izquierdo del objeto y el texto que contiene.
GapY	Integer	Margen entre el borde superior del objeto y el texto que contiene.
TextOnly	Boolean	Determina si las variables que contiene el objeto serán procesadas o no. Posibles valores: True, False
DataField	String	Nombre del campo Blob de la base de datos que contiene los datos.

**Objeto "Rectángulo redondeado con sombra".**

Como este objeto desciende del objeto "Cuadro de texto", posee el mismo conjunto de propiedades que aquel y además las siguientes:

Propiedad	Tipo	Descripción
BeginColor	Integer	Especifica el color inicial usado en el gradiente.
EndColor	Integer	Especifica el color final usado en el gradiente
Gradient	Boolean	Especifica si se utilizará o no un gradiente. Posibles valores: True, False
RoundRect	Boolean	Especifica si las esquinas del rectángulo serán redondeadas. Posibles valores: True, False.
RoundSize	Integer	Si se establece que las esquinas del rectángulo sean redondeadas esta propiedad especifica el grado de curvatura
ShadowColor	Integer	Color de la sombra
ShadowWidth	Integer	Ancho de la sombra
Style	Integer	Si se utiliza el efecto gradiente esta propiedad especifica el estilo de dicho gradiente. Posibles valores: gsVertical, gsHorizontal, gsElliptic, gsRectangle, gsHorizCenter, gsVertCenter

Object "Dibujo".

Propiedad	Tipo	Descripción
Shape	Integer	Determina la forma del objeto. Posibles valores: skRectangle, skRoundRectangle, skEllipse, skTriangle, skDiagonal1, skDiagonal2