



**Velneo V7 7.16**

**+Life is Soft que nunca**

# Índice

Velneo 7.16 Informes, Android, Life is Soft	5
Velneo vReport	7
Velneo vClient Android	17
Velneo Driver ODBC V7	23
Usuarios y Grupos de usuario	26
Controla las rejillas como nunca	31
Nuevo súper control de formulario TreeWidget	32
Guarda y recupera listas de registros	39
Configura la barra de menú a tu gusto en ejecución	40
Parseador de documentos XML	41
Configura el control de selección de fichero y directorio	44
Consulta los estilos de los objetos	45
Cambia dinámicamente los objetos en ejecución	46
Cambia dinámicamente los controles en ejecución	48
Obtén directamente el VFileInfo de un fichero	49
Introduce una fecha de forma sencilla	50
Optimización del pintado de la Rejilla Avanzada	51
Optimización de la Edición de Scripts	52
Mejoras de la autodetección en librerías externas	53

Mejoras en procesamiento numérico	54
Mejora del soporte de sistemas operativos con Qt 4.8.6	55
Más novedades	58
Velneo vTools	62
Rebuscar, combinar y filtrar	63
Exportación CSV, Excel y Html	67
Listas virtuales	68
Gestión de usuarios	69
Sistema de activación	72
El ERP más personalizable	73
Una gran solución ERP	75
Un ERP diseñado por programadores para programadores	77
ERP estándar personalizable	80
Personalizar la barra de menú de tu ERP	81
Personaliza el menú general de tu ERP	82
Personaliza todos los formularios y grids de tu ERP	84
Personaliza todos los informes de tu ERP	86
Personaliza la tecla INTRO de tu ERP	88
Personaliza el diseño de tu ERP con CSS	89
Usabilidad en ciclos de compras y ventas de tu ERP	90
ERP con trazabilidad completa	92
Los tipos de entidad más fáciles de definir en tu ERP	93
Una versión de ERP cargada de mejoras	94

Seminarios On-Line

98

Life is Soft 2015

102

## Velneo 7.16 Informes, Android, Life is Soft

Después de unos intensos meses de trabajo os presentamos la iteración Velneo 7.16 donde hemos trabajado en hacer la vida más fácil al desarrollador de aplicaciones empresariales. Estamos seguros de que esta versión será **+Life is Soft que nunca**.

**Informes:** Somos conscientes de la importancia de la generación de informes en las aplicaciones empresariales. Hemos dado un gran paso en convertir a Velneo vReport en uno de los módulos referencia de la plataforma, para ello hemos integrado este componente para que cualquier programador lo pueda usar sin necesidad de tener que utilizar ninguna open app. Además vuestros usuarios finales pueden crear y editar informes en tiempo de ejecución, imprimir datos con texto enriquecido,... y muchas más mejoras para que **Velneo vReport +Life is Soft que nunca**.

**Android:** Sin duda la plataforma de Google es la de mayor crecimiento y uso en plataformas de movilidad empresarial. Hemos trabajado en un componente Velneo vClient 7.16 basado en QT5 que soluciona la mayoría de las incidencias de la versión anterior. Es un componente estable y que te permitirá trabajar en esta plataforma. Además a partir de este momento cualquier programador o usuario final se podrá descargar el Velneo vClient desde Google Play (Link)

**+Life is Soft que nunca:** Nuestro compromiso es haceros la vida más fácil como desarrolladores y hemos trabajado en muchos detalles para conseguirlo, hemos reescrito el control fecha y el driver ODBC para hacerlos más sencillos y rápidos. En esta versión además aparecen muchas utilidades que facilitarán el uso de vuestros programas, el Treewidget, alta de usuarios, listas virtuales, parseador XML, control de rejillas,...

Muchas más novedades:

- Nuevas instrucciones de API.
- Velneo vTools.
- Nuevo sistema de activación.
- Velneo vERP 7.16.
- Simplificación de Velneo.es
- ...

Durante las próximas semanas iremos publicando posts en el blog de velneo.es explicando en profundidad cada una de las novedades introducidas en esta versión Velneo V7 7.16. Mediante estos artículos os podremos mostrar la dimensión de cada una de las novedades.

Puedes leer el listado completo de post de novedades en el siguiente archivo PDF.

En la página de descargas están disponibles los distintos componentes de Velneo V7 para los diferentes sistemas operativos.

El próximo [16 de Octubre a las 16:00 horas](#) celebraremos un seminario On-Line donde explicaremos las principales novedades de esta iteración.

## Velneo vReport

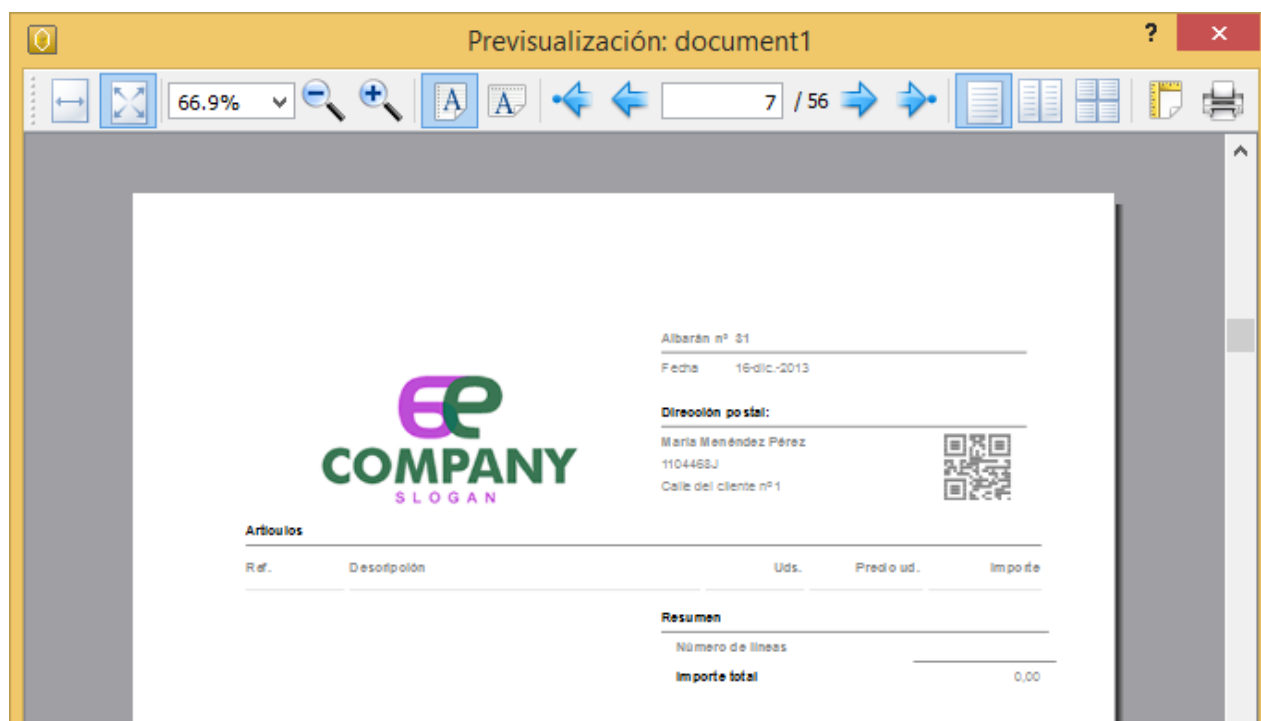
En esta nueva versión de Velneo V7 damos la bienvenida a un Velneo vReport diferente. Presentamos un Velneo vReport más Life is soft, más integrado, más dinámico, con más recursos. Con 30 nuevas funcionalidades disponibles en esta versión para simplificar la elaboración de informes.

### Usa Velneo vReport sin necesidad de heredar

A partir de esta versión de Velneo V7, Velneo vReport se encuentra integrada totalmente en Velneo V7, con lo que no será necesario el uso de la open app para trabajar con informes de Velneo vReport.

De todas formas, continuamos distribuyendo una versión de Velneo vReport con el fin de facilitar su uso, además del tutor correspondiente. En el Tutor de Velneo vReport encontrarás ejemplos en los que se aplican las principales novedades.

### Genera informes de varios niveles aunque no haya registros en todos los niveles



Se ha implementado el uso de subconsultas para generar informes de varios niveles: un maestro con varios plurales que a su vez tienen plurales.

Este tipo de informes permiten, por medio de la configuración en los orígenes de datos, imprimir maestros aunque no haya plurales. También podemos condicionar la impresión de secciones o controles a la existencia de registros.

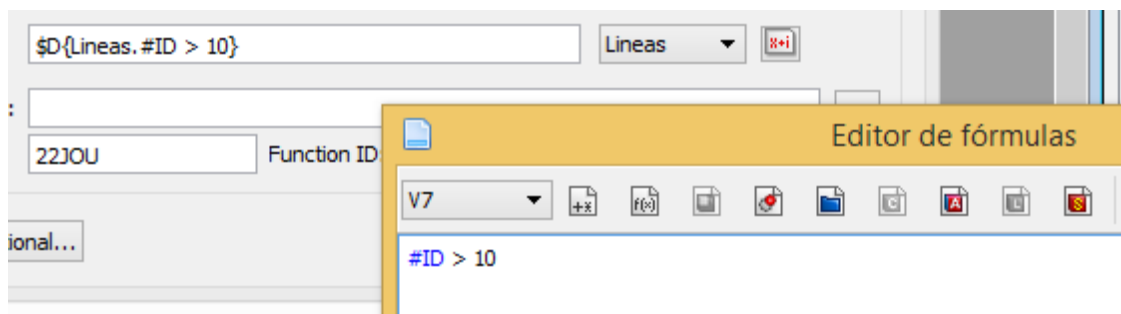
Con esta nueva funcionalidad podremos conseguir varios anidamientos en los informes de vReport, consiguiendo informes aún más complejos y dinámicos.

### Edita fórmulas de código Velneo con el nuevo asistente

Por medio del nuevo asistente de fórmulas para scripts, podremos incluir y editar fórmulas complejas de Velneo V7 con el asistente y aprovechar todas sus funcionalidades.

El asistente permitirá definir una fórmula del tipo `$D{""}` dentro de un script y editarla, seleccionando el origen adecuado.

Además de facilitar la escritura de fórmulas en scripts, nos permitirá seleccionar el origen de datos que usaremos en la fórmula, con lo que podremos ampliar las funcionalidades de nuestros informes de forma sencilla y ajustarlos a nuestras necesidades.



### Imprime HTML combinado con datos

En los controles HTML de un informe encontrarás el nuevo modo Plantilla. Este nuevo modo permite definir contenido de textos que incluyen fórmulas. Podemos incluir fórmulas de Velneo V7 o parámetros (`$P{}` ó `$D{}`).

En ejecución se realizará la sustitución del contenido de un control por el resultado de la fórmula



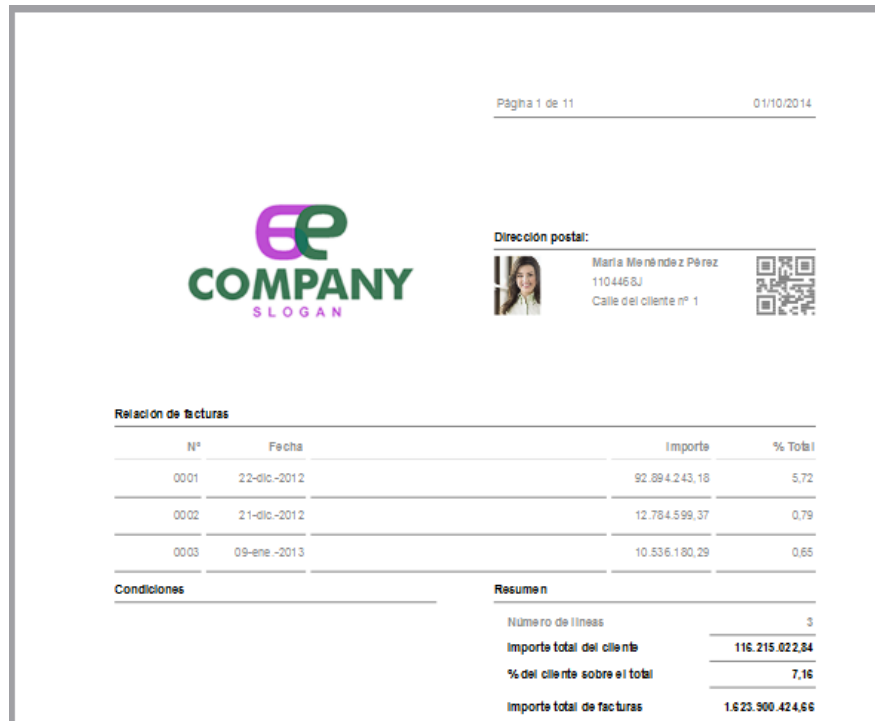
del tipo \$P{} ó \$D{}. De esta forma obtenemos texto continuo que incluye valores dinámicos, permitiéndonos definir informes del tipo combinar correspondencia para cartas, u otros documentos personalizados.



### Envía parámetros desde la aplicación a tus informes

Se ha implementado el paso de parámetros y sus valores correspondientes a informes de Velneo vReport en los comandos de instrucción de impresión de informes de Velneo vReport.

Los parámetros se podrán usar tanto en los controles como en las fórmulas script donde sean necesarios.



Para pasar los parámetros en el momento de la impresión, disponemos de una nueva opción en los comandos de instrucción de impresión de informes de Velneo vReport que nos permite definir, mediante una fórmula, parejas de nombres de parámetros y sus valores correspondientes, con el siguiente formato:

"nombre\_parametro\_1:valor\_parametro\_1|nombre\_parametro\_2:valor\_parametro\_2"

Estos parámetros podremos usarlos en el informe en cualquiera de las fórmulas scripts disponibles, lo que nos permite pasar datos independientes del origen de datos y nos permitirán configurar e imprimir el informe de Velneo vReport según nuestras necesidades.














### Nueva funcionalidad de subinformes en una misma sección en Velneo vReport

Cuando definimos subinformes a partir de un origen de datos generador de subinformes, podemos imprimir dos de los subinformes en una misma sección de detalle.

Para hacer uso de esta funcionalidad, en la configuración del detalle tenemos la propiedad Origen de datos secundario. Esto nos permitirá incluir controles en el detalle seleccionando un origen u otro.

Esta funcionalidad requiere que el origen de datos principal tenga un número mayor o igual de

líneas que el secundario. Cuando sucede que el origen de datos secundario tiene menos registros y, por tanto, uno de los detalles contiene más líneas que el otro, debemos hacer uso de la variable @DS2 para condicionar la impresión de los controles correspondientes al origen de datos secundario, ya que de otra forma, se imprimirían los datos del último registro existente en paralelo al detalle del origen de datos principal.

Factura n° 1		Página 1 de 62			
Fecha		22-dic-2012			
		<b>Dirección postal:</b>  María Menéndez Pérez 1104465J Calle del cliente n° 1			
					
Artículos				Vencimientos	
Ref.	Descripción	Uds.	Precio ud.	Importe	Fecha
REF0000010	Peugeot 5008	38	17.910,00	680.580,00	2013-01-01
	 Peugeot renueva su monovolumen 5008 con cambios				
REF0000008	Ford Mustang	12	45.900,00	550.800,00	2013-01-31
	 Es un deportivo de dos puertas, cuatro plazas.				
REF0000006	Porsche 918 Spyder	99	800.000,00	79.200.000,00	2013-02-05
	 El Porsche 918 Spyder tiene un precio cercano a				
REF0000004	Mercedes Benz Clase GLA	94	31.150,00	2.928.100,00	
	 Su carrocería se asemeja a las de los todoterrenos.				
REF0000003	Jaguar F-Type Coupe	11	77.150,00	848.650,00	
	 La versión con techo cerrado del F-Type está				

### Imprime etiquetas sin que se repitan las de la última fila

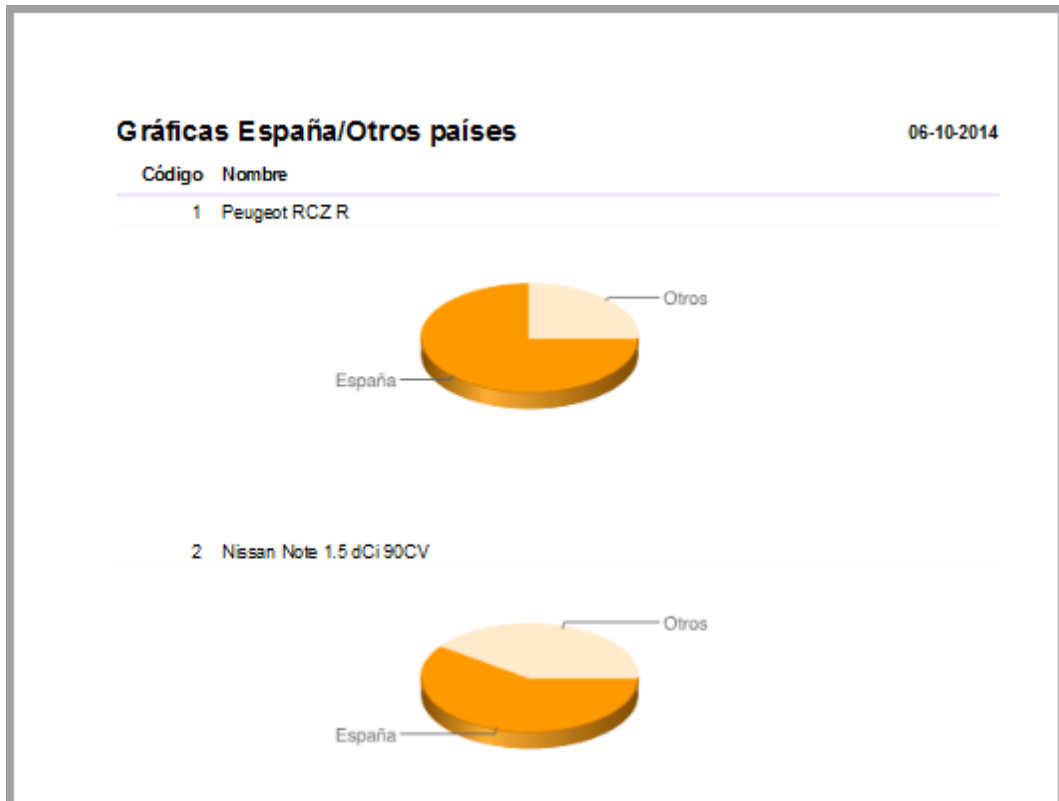
Cuando imprimimos etiquetas usamos la condición @SKIP\_NEXT para el salto de registro en el detalle y en la última línea, en el caso de que no se completen los registros, se repite el último registro en aquellas etiquetas que deberían quedar vacías al no tener más registros que imprimir.

Usando la variable de sistema @ISNOTFINISHED como condición para imprimir, evitaremos que se impriman estas etiquetas de más, mejorando el resultado final del informe.



### Imprime múltiples imágenes con origen URL en informes

Cuando incluíamos un control Imagen en el detalle de un informe de Velneo vReport cuyo tipo de origen del ítem era URL, en la impresión del informe únicamente se imprimía el último origen descargado en cada uno de las líneas de detalle.



Ahora podremos incluir imágenes como resultado de definir una URL obtenida de la base de datos, permitiéndonos que en cada línea de detalle imprimamos una imagen distinta en función de la URL, ampliando las funcionalidades de nuestros informes, por ejemplo, incluyendo imágenes como resultado de gráficos.

### Ajusta de forma dinámica un control en un informe de vReport cuando cambie el papel

Se han implementado dos nuevas propiedades: Ajustar a la izquierda, ajustar a la derecha para los controles Texto, Campo, Rectángulo y línea, que permiten configurar el ajuste de la posición con respecto a los márgenes del papel, cuando hay cambios en el tamaño o la orientación de éste. Ajustar a la izquierda o derecha llevarán el control al margen correspondiente, independientemente del tamaño u orientación del papel.

Esta nueva funcionalidad facilitará el desarrollo de informes dinámicos, en los que el usuario cambie el papel que se usará en la impresión.

### **Soportada la impresión de textos largos en multi-zona**

Si tenemos definido un informe que incluye objetos de longitud variable, aunque todos los objetos del informe tenga en sus propiedades la propiedad zona ID = 0, si no se activa la propiedad "Salto permitido entre zonas" en las propiedades del informe, éste da error al imprimirlo (texto demasiado largo).

Para evitar este mensaje, lo que debemos hacer es definir una zona con ID>=1 y activar el "Salto permitido entre zonas". Los controles con textos que requieren partirse han de estar en zonas independientes para que sean impresos correctamente.

### **Usa la función de campo *getName()* para informes multi-idioma**

Cuando queremos evitar etiquetas estáticas para el nombre de campos, lo que podemos hacer es crear un control campo, con origen de la tabla de detalle, que aplica la función de campo *getName()*, para obtener el nombre del campo:

Campo:

Origen: LíneasFactura

Fórmula: #ARTICULOS.NAME:getName()

Cuando hacíamos esto, en determinados casos ciertos controles del informe se imprimían de forma incorrecta.

Solucionada la incidencia podemos hacer uso de esta funcionalidad que nos facilitará el desarrollo de informes aún más dinámicos.

### **Usa variables calculadas en agrupamientos en los pies**

Cuando usamos una variable que se inicializa en el agrupamiento y que totaliza un campo del detalle, el valor que se presentaba en el pie de página no se ajustaba al valor que se mostraba en la agrupación si ésta última no se imprimía de forma completa en la página en curso.

Solucionada la incidencia, podremos hacer uso de las variables recalculadas en agrupamientos en el pie de página que mostrará el valor correspondiente, coincidiendo el del agrupamiento con el pie de página.

### **Elimina cabeceras o pies de agrupamiento en informes**

Cuando eliminamos una cabecera o pie de agrupamiento en un informe de Velneo vReport en determinadas circunstancias podía producirse un error que hacía que se cerrara el editor.

Solucionada la incidencia, podemos trabajar cómodamente con las cabeceras o pies de agrupamiento para la edición de nuestros informes.

### **Imprime imágenes en informes PDF desde procesos en 2º plano**

Podremos ampliar la funcionalidad de los informes de Velneo vReport incluyendo imágenes en nuestros informes también cuando queramos que la impresión se produzca en segundo plano en pdf.

### **Mejorada la impresión de subinformes cuando no tienen plurales**

En un informe con subinformes (por ejemplo: facturas con líneas de factura y vencimientos), cuando la factura no tenía ni líneas de factura ni líneas de vencimientos, al imprimir el informe éste no es generado, y se presentaba el mensaje de error:

Columna de agrupación o fórmula script no válida: \$D{#FACTURAS.ID}

ID de grupo: Factura

Valor: NULL

Solucionada la incidencia, los subinformes contemplan el caso de registros generadores de subinformes que en realidad no tienen plurales, permitiendo imprimir las zonas correspondientes al registro generador de subinforme aunque no se impriman los subinformes no existentes en realidad. De esta forma evitamos programación adicional, como el filtrado previo de registros sin plurales. (VBM-3773)

### **Mejoras en los cálculos agrupados por página**

Cuando una variable tenía como opción para el reinicio "Page" (página), este reinicio no se producía.

Solucionada la incidencia, podemos hacer uso de esta opción en nuestros informes cuando necesitemos que una variable se ponga a 0 al comienzo de cada página, con lo que ampliaremos la funcionalidad de nuestros informes.

## Retorno de la función toFixed en scripts de vReport

En vReport la función "toFixed" no devolvía el valor correspondiente. Solucionada la incidencia, ya podemos hacer uso de esta función y ampliar la funcionalidad de las fórmulas de nuestros informes.

## Otras novedades y mejoras del generador de informes vReport

- Implementado en el control código de barras la entrada de tipo script, de forma que podemos definir el contenido no sólo por medio de un valor o un campo, sino también mediante un script que se calcula en el momento de imprimir.
- Nueva función rowCount que permite obtener el número de registros que tiene un determinado origen de datos (rowCount(\$D{myDataSource})
- Implementado tamaños por defecto de los controles. En preferencias encontraremos nuevas propiedades en preferencias que nos permiten definir el tamaño por defecto de los controles y el texto que incluyen.
- Implementado autocompletado en la ventana de propiedades del control código de barras.
- Implementada codificación de fichero en función de la configuración.
- Implementada configuración en el informe de la cabecera en la salida html de informes
- Implementado reinicio de variable en modo página.
- Implementada edición de id de un control en cuadros de diálogo.
- Implementado soporte multilínea en scripts.
- Mejora del soporte de CSS en controles y secciones en la salida html de informes
- Optimizado el rendimiento del sistema de previsualización de informes para un menor consumo de memoria
- Evitada la migración incorrecta a UTF-16 de los informes en el editor.
- Solucionada incidencia que provocaba el error "Pie de informe demasiado largo".
- Evitada la necesidad de resetear las variables cuando usamos funciones en el cálculo de variables en informes con múltiples detalles y presentando los datos después del primer detalle.
- Se evita que rompa el editor de informes al añadir secciones de cabecera o pie.



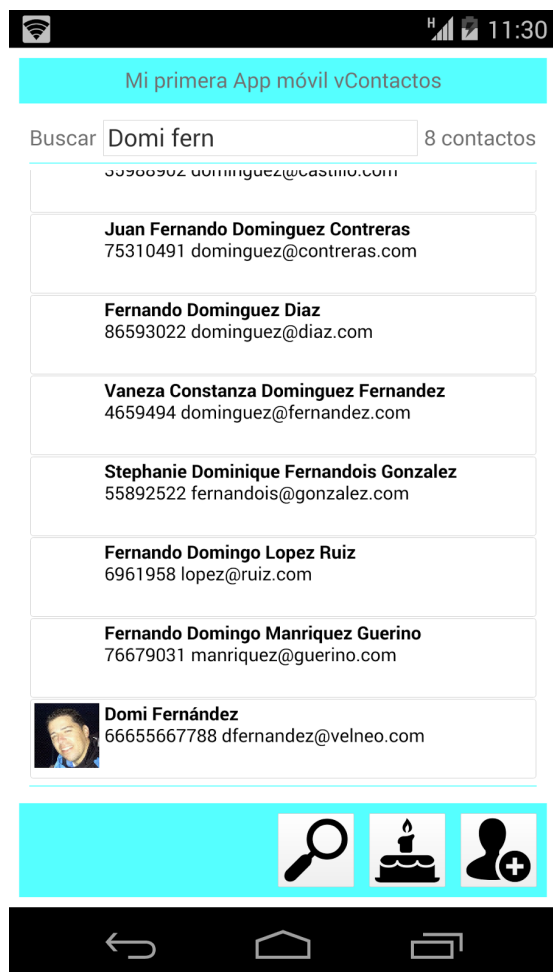
## Velneo vClient Android

En Velneo seguimos apostando por dotar a la plataforma de una solución de movilidad que permita dar a nuestras aplicaciones un acceso remoto desde el mayor número de dispositivos móviles posibles. En esta iteración 7.16 hemos publicado una nueva versión del componente Velneo V7 vClient Android.

Como primera mejora hemos implementado la nueva versión 5.3.1 de las librerías Qt. Hemos incorporado una serie de novedades que podremos disfrutar en las nuevas versiones, como la integración con el teclado y los controles, ya no es necesaria la instalación de librerías adicionales (Ministro), simplificando el proceso de instalación y un mejor funcionamiento en general de las aplicaciones.

Hemos desarrollado una sencilla Open App de ejemplo, [vContactos Móvil](#), que se complementa con una serie de [recomendaciones para la programación orientada a móvil con Velneo](#). Esto servirá de guía práctica y teórica, de forma que se simplifica y agiliza enormemente el desarrollo de interfaz para su ejecución en dispositivos móviles.

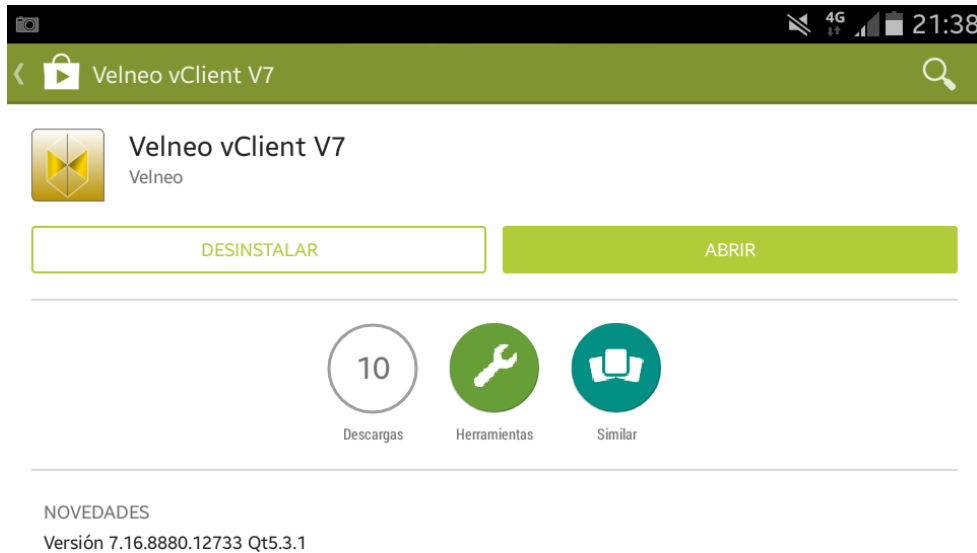
[Enlace a video](#)



### Disponible en Play Store

Para facilitar el uso de Velneo vClient V7 en tu Android, lo podrás obtener a través de Play Store de Google donde podrás descargar e instalar Velneo vClient V7. Además, disfrutarás del servicio de actualización de Play Store de Google que puedes configurar para que te avise de nuevas versiones y te actualice de forma automática.

Desde luego, seguirás disfrutando de Velneo vClient V7 haciendo uso del enlace que suministramos en la página de descargas.



## Más fácil de instalar

Hemos simplificado el sistema de instalación de Velneo vClient V7 para que te sea más fácil disfrutar de Velneo en tu tableta o móvil. Ya no es necesaria la instalación del software Ministro de Android para ejecutar Velneo vClient V7 en Android.

Para disfrutar de la nueva versión, desinstala Ministro e instala Velneo vClient V7 de la forma que prefieras, desde Play Store de Google o con el enlace que suministramos en la página de descargas. Velneo vClient V7 se instalará en un único paso.

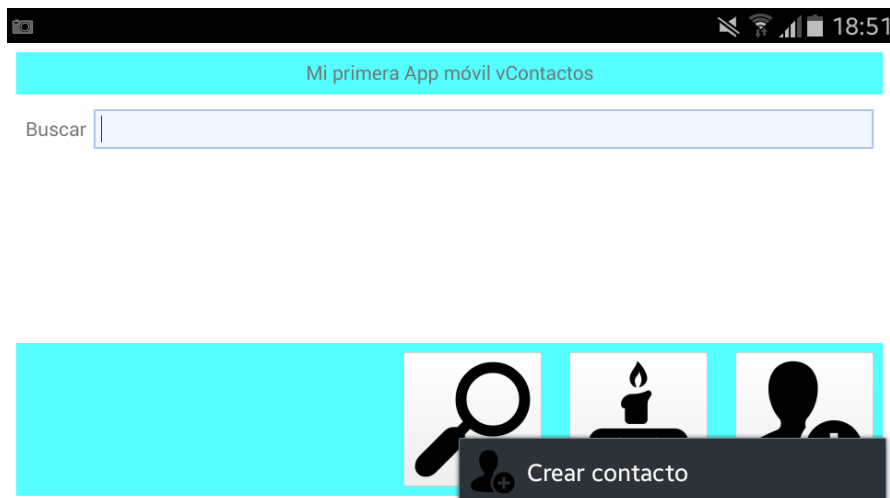


## Incluye el botón estándar de opciones para tus menús en Android

Cuando incluyas un objeto menú en el marco AUTOEXEC, este menú se integrará de forma automática con Android, mostrando una barra de título con el nombre de la aplicación y un botón estándar de Android para el despliegue de las opciones.

En función del dispositivo móvil, la opción de menú se mostrará en pantalla, si éste no tiene botón físico de Menú. O al pulsar el botón físico del dispositivo, si es que lo tiene.

El menú soporta además varios niveles, por lo que podrás incluir no sólo acciones sino también submenús que se desplegarán a medida que el usuario seleccione opciones.

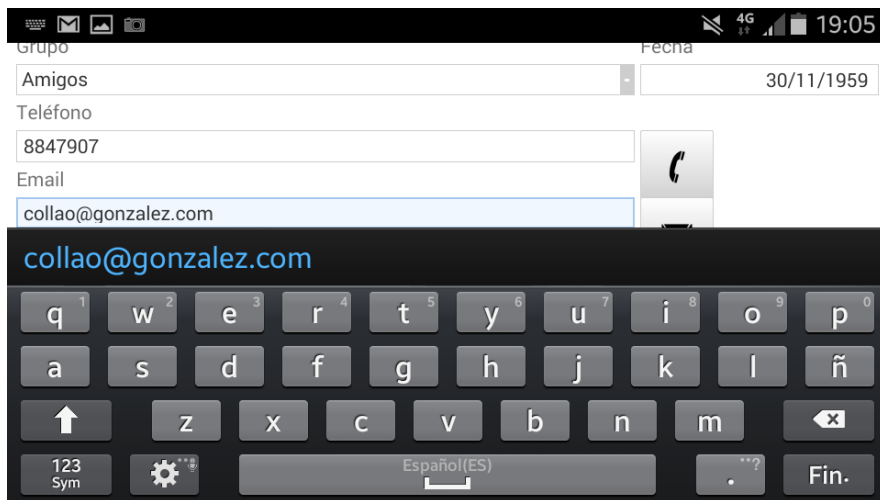


## Mejora en la integración del teclado con tus aplicaciones

El uso del teclado estándar de Android y de las funciones de autocompletado y opciones de corrección ortográfica podían dificultar la introducción de texto en la ventana de login o en los distintos controles incluidos en la aplicación.

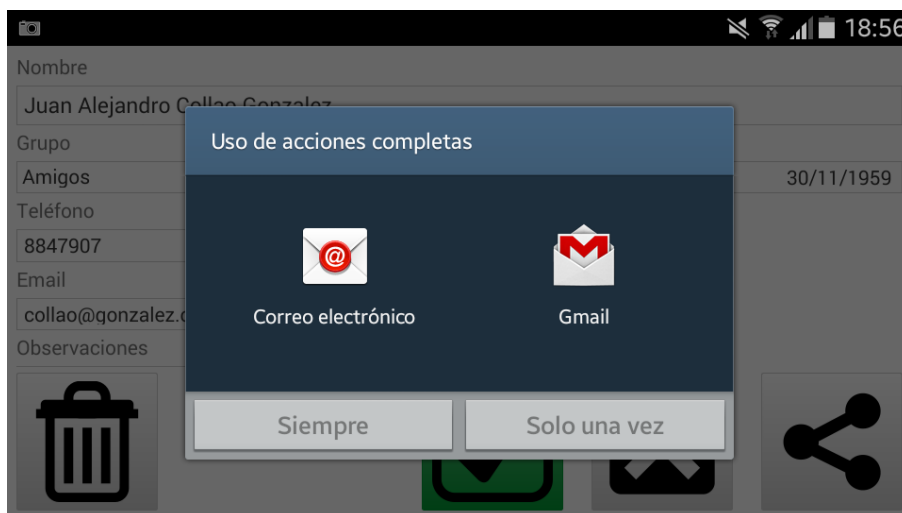
En esta nueva versión se mejora el soporte del teclado estándar de Android lo que nos permite hacer uso de estas funcionalidades, mejorando también la integración con otros teclados que responderán de mejor forma a la introducción de datos.

Problemas como el existente a la hora de introducir una VRL la primera vez, se ven solventados en esta nueva versión.



## Integra tu aplicación con Android para enviar correos, llamar por teléfono, etc.

Con ayuda del comando de instrucción “Ejecutar documento” puedes enviar correos electrónicos desde el móvil, marcar un número de teléfono, abrir mapas, enviar un whatsapp, etc.



## Uso de las fuentes del sistema para tus aplicaciones

Las aplicaciones de Velneo V7 reconocerán los tipos de letra existentes en el sistema y adoptará la fuente del sistema adecuada, teniendo en cuenta también el idioma en que se encuentre configurado el sistema.

De esta forma mostrará correctamente todos los caracteres especiales (con tildes, etc.)

correspondientes al idioma en curso.

### **Tus aplicaciones en el idioma de Android**

Las aplicaciones de Velneo V7 reconocerán el idioma configurado en tu móvil y configurará el idioma por defecto de la aplicación que ejecutemos.

### **Presenta tus aplicaciones en pantalla completa**

Cuando necesites el mayor espacio en pantalla que te proporcione tu tableta o móvil, podrás usar la opción de pantalla completa.

Recuerda que podrás acceder de nuevo al resto de aplicaciones de tu tableta o móvil usando los gestos para abrir las notificaciones.

### **Usa comandos de acción y menús en stock en tus aplicaciones**

Disfruta de todas las opciones en stock en tus aplicaciones y que te permiten usar funcionalidades sin tener que programarlas.

En versiones anteriores estaba limitado el uso de acciones y menús en stock, ahora tendrás disponibles más opciones para usarlas en tus aplicaciones.

### **Sal de una aplicación con el botón Cancelar**

Para cerrar nuestras aplicaciones podemos hacer uso de la opción de menú en stock Salir, pero también podremos salir de una aplicación con el botón de Android Cancelar, lo que no sólo hará que vuelvas al punto anterior, si no que realmente cerrará la aplicación, dejando de consumir recursos del sistema.

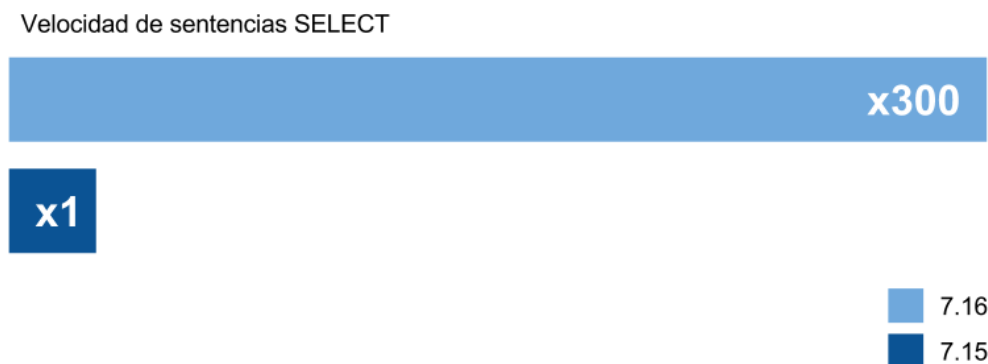
Para ello, debemos cerrar previamente las vistas de datos que se encuentren abiertas.

## Velneo Driver ODBC V7

Las mejoras que hemos implementado nos permitirán integrar Velneo V7 en otros sistemas de forma rápida y sencilla. Hemos conseguido optimizar su rendimiento obteniendo resultados de hasta 300 veces la velocidad original, y también hemos simplificado la administración y configuración del driver de Velneo V7.

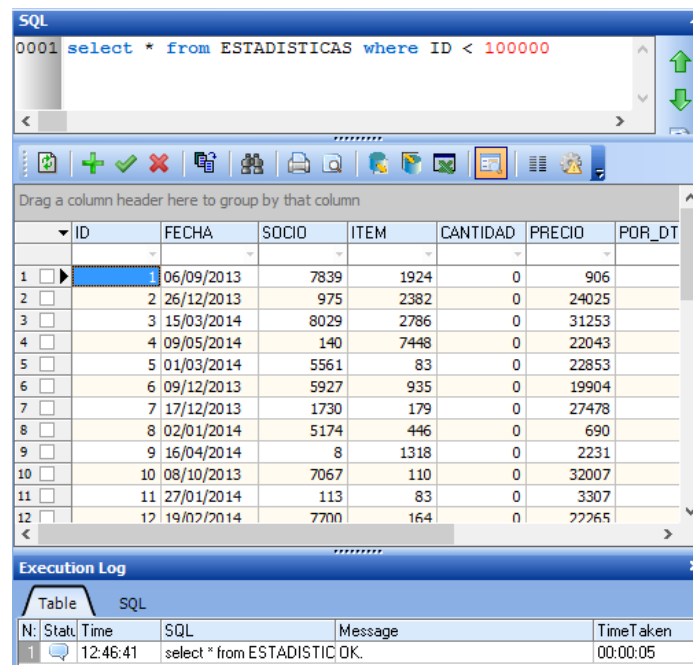
---

### Comparativa Velneo V7 7.15 vs Velneo V7 7.16



### Sentencias SELECT hasta 300x más rápidas

Hemos trabajado en la mejora de velocidad cuando retornamos listas con muchos registros. Hemos conseguido mejoras de hasta 300 veces la velocidad de partida cuando usamos SELECT que devuelven millones de registros.

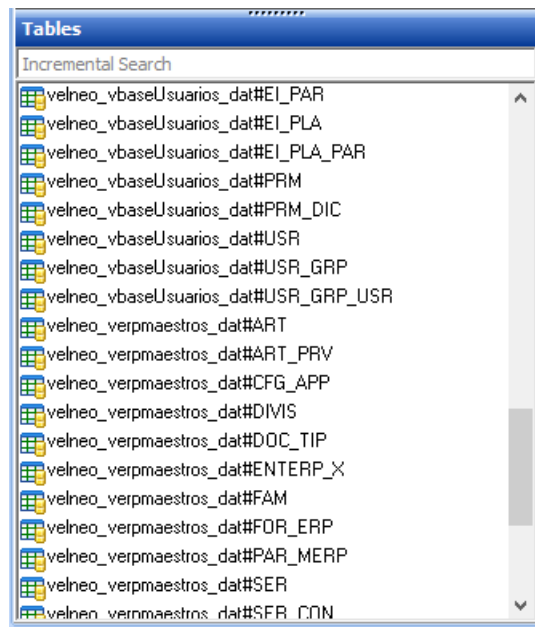


Esta mejora de velocidad permite obtener información usando el driver ODBC mucho más rápido desde otras aplicaciones, lo que mejorará la integración de Velneo V7 con otros sistemas, facilitando el tráfico de datos y la interoperación de aplicaciones.

### Accede a todas las instancias con una única conexión

Cuando definimos una conexión a una instancia además de las tablas del proyecto en curso, tendrás disponibles las instancias de datos de todos los proyectos heredados, y por tanto, podrás acceder a la información de todas las tablas que herede.





De esta forma será más sencillo acceder a la información y facilita la configuración de conexiones del driver, minimizando las tareas de administración. Con una única conexión, accederás a toda la información de tu aplicación.

Para acceder a las tablas heredadas basta con anteponer el alias de la aplicación al nombre de la tabla, usando el separador #. Por ejemplo: aliasAplicacion#NOMBRE\_TABLA.

## Usuarios y grupos de usuarios, el servidor a tu alcance

Hemos creado un nuevo subAPI de Velneo V7 llamado “AdminApi”. Está formado por 5 nuevas clases con las que podrás leer, crear, modificar y eliminar usuarios, grupos de usuarios y sus permisos desde tus aplicaciones, igual que hasta ahora lo hacías con vAdmin.

### subAPIs

Esta versión incorpora 32 clases nuevas o mejoradas con más de 310 nuevas funciones. Para mejorar la gestión del API hemos organizado las clases en módulos o subAPIs. De esta forma podemos evitar que todas las clases se carguen por defecto en el motor de ejecución de JavaScript.

Cuando queramos el nuevo grupo de función de administración (Módulo o subAPI Admin) lo haremos con la nueva función *importApi()*.

### Nuevo objeto theServerAdmin

Este nuevo objeto representa el servidor de aplicaciones (Velneo vServer V7) al que estamos conectados. Este objeto juega un papel importante en el nuevo subAPI AdminAPI ya que nos permite acceder a toda la configuración de usuarios, grupos, carpetas compartidas, soluciones y permisos configurados en el servidor.

A continuación vamos a repasar las 5 nuevas clases y sus funciones:

### VServerAdmin

Esta clase se usa para realizar labores administrativas con el servidor. Estas son las funciones incluidas en la nueva clase.

- addUser
- addUserGroup
- checkUserPassword
- delUser
- delUserGroup

- getAllUserGroups
- getAllUsers
- getLastErrorCode
- getSolutions
- getUserById
- getUserByName
- getUserGroupById
- getUserGroupByName
- getUsersByGroup
- modUser
- modUserGroup

## **VUser**

Representa a un usuario declarado en el servidor de aplicaciones. Estas son las funciones incluidas en la nueva clase.

- addGroup
- changePassword
- changePasswordNextConnect
- comments
- fullName
- groupCodeAt
- groupCount
- id
- isAccountBlocked
- isAccountDisabled
- isAdministrator
- isGroupMember
- name
- passwordNotExpire
- removeGroup
- setAccountBlocked
- setAccountDisabled
- setAdministrator
- setChangePasswordNextConnect
- setComments
- setFullName
- setName
- setPasswordNotExpire

- VUser

## VUserGroup

Representa a un grupo de usuarios declarado en el servidor de aplicaciones. Estas son las funciones incluidas en la nueva clase.

- addAppPermission
- addAuthorizedSharedFolder
- addDataPermission
- addPermissionList
- authorizedSharedFolders
- canCreateSolutions
- canDeleteSolutions
- canRunDataClient
- dataPermissionList
- editPermission
- editPermissionList
- hasAppPermission
- hasDataPermission
- hasEditPermission
- id
- isAuthorizedSharedFolder
- name
- removeAllAppPermission
- removeAllDataPermission
- removeAllEditPermission
- removeAppPermission
- removeAuthorizedSharedFolder
- removeDataPermission
- removeEditPermission
- setCanCreateSolutions
- setCanDeleteSolutions
- setCanRunDataClient
- setEditPermission
- setName
- VUserGroup

## VUserGroupList

Representa a una lista de grupos de usuarios declarados en el servidor de aplicaciones. Estas son las funciones incluidas en la nueva clase.

- append
- getAt
- removeAll
- removeAt
- size
- sortById
- sortByName
- VUserGroupList

### **VUserList**

Representa a una lista de usuarios declarados en el servidor de aplicaciones. Estas son las funciones incluidas en la nueva clase.

- append
- getAt
- removeAll
- removeAt
- size
- sorterByFullName
- sorterById
- sorterByName
- VUserList

Ejemplo de como leer los usuarios y sus grupos:

```
// -----  
// Leer los usuarios y sus grupos  
// -----  
importApi("AdminApi");  
  
var listaUsuarios = new VUserList();  
if (theServerAdmin.getAllUsers(listaUsuarios))  
{  
    var usuario = new VUser();  
    var grupo = new VUserGroup();  
    var numUsuarios = listaUsuarios.size();  
  
    for (var numUsuario = 0; numUsuario < numUsuarios; numUsuario++)  
    {  
        usuario = listaUsuarios.getAt(numUsuario);  
        alert("Usuario: " + usuario.id() + " " + usuario.name());  
  
        numGrupos = usuario.groupCount();  
  
        for (var numGrupo = 0; numGrupo < numGrupos; numGrupo++)  
        {  
            theServerAdmin.getUserGroupById(usuario.grupoCodeAt(numGrupo), grupo);  
            alert("Grupo: " + grupo.id() + " " + grupo.name());  
        };  
    };  
};
```

Ejemplo de como crear un nuevo usuario:

```
// -----  
// Crear un nuevo usuario  
// -----  
var usuarioNuevo = new VUser();  
usuarioNuevo.setName("prueba");  
usuarioNuevo.setFullName("Nombre Apellidos");  
theServerAdmin.addUser(usuarioNuevo);
```

## Controla las rejillas como nunca

Las nuevas funciones incorporadas en la clase *VGridListView* nos permiten controlar como nunca la visualización de las filas, de las columnas, del grid, la edición sobre la rejilla e incluso leer directamente la información visualizada en las celdas de la rejilla.

A continuación se detallan las nuevas funciones de la clase *VGridListView*:

### Controla las filas

Cambia el alto de cada fila independientemente con la función *resizeRowToContents()* o de todas las filas de la rejilla en función de su contenido con la función *resizeRowsToContents()*. Además puedes ocultar filas de la rejilla con la función *setRowVisible()* y conocer si una fila está o no oculta con la función *isRowHidden()*.

### Controla las columnas

Analiza cuantas columnas hay en la rejilla con *columnCount()* o cuantas columnas están visibles en cada momento con *visibleColumnCount()*. Interroga la rejilla por el ancho en píxeles de una columna con *columnWidth()* y cambia el ancho de cada columna a un tamaño fijo con *setColumnWidth()* o de todas las columnas con *resizeColumnToContents()*.

### Controla la edición

Ahora puedes de forma programada editar una fila o una celda concreta determinada por una fila y columna, o una fila y el identificador de una columna con la nueva función *editItem()*.

### Controla el grid

Ahora puedes dinámicamente saber si el grid está visible con *showGrid()* y ocultarlo o hacerlo visible con la función *setShowGrid()*.

### Controla la ordenación

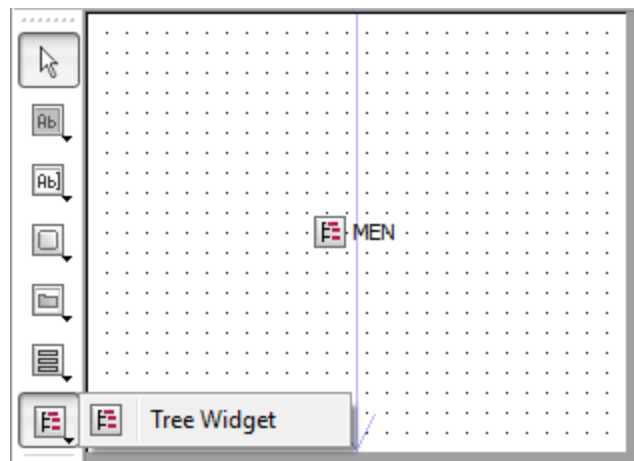
Activa o desactiva que el usuario final pueda ordenar los registros de la rejilla haciendo clic en la cabecera de columna con la función *setSortingEnabled()*, e interroga a la rejilla para saber si el usuario puede o no hacer ordenación con la función *isSortingEnabled()*.

### Controla los datos

Aunque las rejillas muestran listas de registros de una tabla de origen en sus celdas se pueden visualizar datos de tablas maestras, fórmulas, variables, etc. Ahora tienes la posibilidad de saber que tipo de información se visualiza en cada columna *dataType()* y obtener el dato visualizado en cada celda con *data()* y con *dataImage()* en caso de imágenes o iconos.

## Nuevo súper control de formulario TreeWidget

Este nuevo control de formulario nos aporta una gran potencia y flexibilidad ya que nos permite pintar información con una estructura arbolada con múltiples columnas, lo que también nos permite simular un grid.

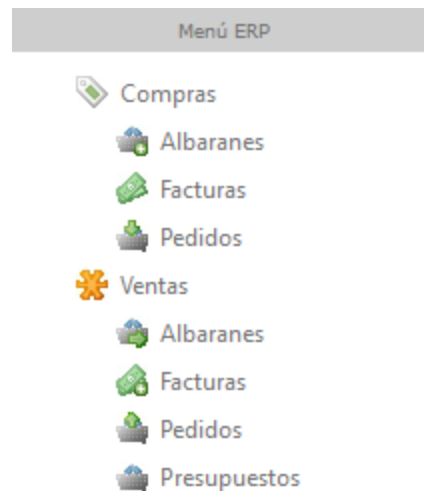


Este control no tiene origen de datos por lo que podremos visualizar información procedente de cualquier origen de información, tanto tablas de Velneo V7, como de cualquier base de datos, ficheros XML, etc.

La gestión de la funcionalidad del control se realiza desde 2 nuevas clases del API lo que nos aporta la flexibilidad necesaria para que el control sea totalmente configurable y dinámico en tiempo de ejecución.



Un ejemplo de la potencia de este control es el nuevo menú dinámico programado en Velneo vERP que genera el menú a partir de una tabla arbolada que utiliza como base de la información a integrar dentro del menú arbolado creado con este nuevo control.



El control se gestiona mediante 2 clases VTreeWidget para configurar y gestionar los aspectos generales del control y la clase VTreeltem con la que podremos gestionar los ítems que forman el árbol.

No debemos olvidar que el control no exige una estructura arbolada por la que es fácil convertirlo en una rejilla sin origen de datos.

A continuación vemos un ejemplo de código JavaScript que rellena el control con un menú arbolado.

El proceso comienza preparando el control, las variables y cargando los registros de la tabla arbolado que servirá de base para generar el menú.

```
// Preparar el control de menú
var formulario = theRoot.dataView();
var menu = formulario.control("MEN_APP");

// Configuración general del menú
menu.setHeaderLabel(0, "Menú General");
menu.setHeaderLabel(1, "Tipo");
menu.setHeaderLabel(2, "idRef");
menu.hideColumn(1);
menu.hideColumn(2);

// Preparar variables de trabajo
importClass("VImage");
var opcionPadre = "";

// Cargar todas las opciones del menú
var listaMenu = new VRegisterList(theRoot);
listaMenu.setTable("velneo_verp_2_dat/PRS_MEN_W");
listaMenu.load("ID", []);
```

Con este bucle recorreremos los registros de la tabla arbolada y vamos generando los ítems del árbol.

```
// Leer las opciones del menú y cargar el control del árbol
for (var numRegistro = 0; numRegistro < listaMenu.size(); numRegistro++ ) {

    // Lectura de los datos de la opción de menú
    var registro = listaMenu.readAt(numRegistro);
    var opcionId = registro.fieldToString("ID");
    var nivel = opcionId.length;
    var opcionTexto = registro.fieldToString("NAME");
    var opcionTipo = registro.fieldToString("MEN_OBJ");
    var opcionAccion = registro.fieldToString("OBJ_ID_REF");
    var icono = registro.fieldToImage("ICO");

    // Control de nivel principal
    if (nivel == 3) {
        item = menu.addTopLevelItem();
        item.setText(0, opcionTexto);
        item.setIcon(0, icono);
    }
    else {
        if (nivel == 6) {
            opcion = item.addChild();
            var opcionPadre = opcion;
        } else {
            opcion = opcionPadre.addChild();
        };
        opcion.setText(0, opcionTexto);
        opcion.setText(1, opcionTipo);
        opcion.setText(2, opcionAccion);
        opcion.setIcon(0, icono);
    };
};
```

Finalmente con este código hacemos los últimos ajustes visuales del menú

```
// Preparar visualización del menú por defecto
menu.headerHidden = true;
menu.rootIsDecorated = true;
menu.expandAll();
```

A continuación se detallan las 2 nuevas clases del API y las 50 funciones que hacen de este control uno de los más potentes y versátiles de Velneo V7.

## **VTreeWidget**

Esta clase representa el control *TreeWidget*. Con las funciones de esta clase tendremos acceso a la interfaz de un control de árbol basado en ítems, podremos crear vistas de datos jerarquizados. El control no tiene origen de datos por lo que podemos utilizarlo para visualizar datos tanto de tablas de origen Velneo como de cualquier origen de información.

Antes de añadir ítems al control se debe fijar el número de columnas. Esto permite que cada ítem pueda tener una o más etiquetas. El número de columnas en uso puede obtenerse con la función *columnCount()*.

El árbol puede tener una cabecera que contenga una sección por cada columna en el control. Esto facilita la configuración de las etiquetas de cada sección aplicando una lista de textos con la función *setHeaderLabels()*, además una cabecera personalizada puede configurarse con la función *setHeaderItem()* de la clase *VTreeItem*.

Los ítems en el árbol pueden clasificarse por una columna en función de un orden predefinido. Si la ordenación está habilitada, el usuario podrá clasificar los ítems haciendo clic en la cabecera de columna. La ordenación puede activarse o desactivarse con la función *setSortingEnabled()*. Con la función *isSortingEnabled()* podremos saber si la ordenación está o no habilitada.

- *addTopLevelItem*
- *clear*
- *collapseAll*
- *collapseItem*
- *columnWidth*
- *currentColumn*
- *currentItem*
- *deleteTopLevelItem*
- *editItem*
- *expandAll*
- *expandItem*
- *expandToDepth*
- *hideColumn*
- *insertTopLevelItem*
- *resizeColumnToContents*
- *scrollToItem*

- `setColumnWidth`
- `setCurrentItem`
- `setHeaderLabel`
- `setIconSize`
- `showColumn`
- `sortColumn`
- `sortItems`
- `topLevelItem`

## **VTreeltem**

Esta clase es usada para gestionar cada una de los ítems que forman un control de la clase *VTreeWidget*.

Cada ítem puede contener una o varias columnas de datos en la que podemos incluir textos, etiquetas y un icono.

Los ítems se construyen con un padre que se corresponde con un ítem de primer nivel en el objeto de la clase *VTreeWidget* u otro ítem de la clase *VTreeltem* de nivel superior.

Cada columna en un ítem puede tener su propio color de fondo fijado con la función *setBackgroundColor()* y su color de texto fijado con la función *setForegroundColor()*.

La principal diferencia entre los ítems de nivel principal y el resto de ítems inferiores del árbol es que los ítems de nivel superior o principal no tiene ítem padre, esta información puede ser útil a la hora de insertar o borrar ítems del árbol. Los ítem hijos pueden ser eliminados con la función *deleteChild()* o añadidos con la función *addChild()*.

Cada ítem, por defecto es seleccionable, activable, etc. Esta configuración en el ítem se realiza a través de flags que pueden ser alterados con la función *setFlags()* aplicando los valores que se encuentran en el enum *ItemFlag* de esta clase.

- `addChild`
- `backgroundColor`
- `child`
- `childCount`
- `columnCount`
- `deleteChild`
- `flags`

- foregroundColor
- isExpanded
- isHidden
- isSelected
- setBackgroundColor
- setExpanded
- setFlags
- setForegroundColor
- setHidden
- setIcon
- setSelected
- setStatusTip
- setText
- setTooltip
- setWhatsThis
- statusTip
- text
- toolTip
- whatsThis

## Guarda y recupera listas de registros

Hasta ahora hemos manejado listas de registros en tablas en disco y en memoria, mediante búsquedas, localizadores, incluso almacenando las listas en cestas en memoria, pero a partir de ahora tienes el poder de guardar listas de registros en ficheros en disco, en variables en memoria e incluso en campos objeto.

Gracias a las sencillas y potentes nuevas funciones de serialización de la clase *VRegisterList*, podrás almacenar listas de registros en disco con *saveToFile()* en una variable de tipo *ByteArray* con la función *saveToData()* y recuperar dichas listas con las funciones *loadFromFile()* y *loadFromData()* para obtener un objeto de la *VRegisterList* cuando quieras.

Un ejemplo práctico de la potencia de estas nuevas funciones es la funcionalidad de listas virtuales incorporada en las nuevas *vTools* que aparecen también en esta versión integradas en *Velneo vERP* que te permiten sin programar guardar y recuperar listas virtuales desde el menú contextual de cualquier objeto de vista de datos.

## Configura la barra de menú a tu gusto en ejecución

Con las nuevas funciones incorporadas en la clase VMainWindow ahora ya puedes limpiar la barra completamente y añadir e insertar acciones, acciones en stock, menús y separadores para configurar la barra de menú a tu gusto en tiempo de ejecución. Esto te permite configurar las opciones dinámicamente, por ejemplo en función del usuario. A continuación se detalla la lista de nuevas funciones de la clase destinadas a gestionar la barra de menú.

### VMainWindow

- clearMenuBar
- addMenuToMenuBar
- addActionToMenuBar
- addStockMenuToMenuBar
- addSeparatorToMenuBar
- insertStockMenuToMenuBar
- insertSeparatorToMenuBar
- insertMenuToMenuBar
- insertActionToMenuBar



## Parseador de documentos XML

Hasta ahora disponíamos de una clase para la generación de documentos XML *VXmlWriter*. A partir de esta versión también podremos parsear documentos XML gracias a la nueva clase *VXmlReader*, que contiene un conjunto de funciones que nos facilitan la lectura de documentos XML con gran flexibilidad.

### VXmlReader

Esta clase nos da acceso a un parseador rápido de archivos well-formed XML 1.0 para lectura de manera secuencial. Una vez construido el objeto *VXmlReader* hay que indicarle uno de los siguientes orígenes:

- Disco, con la función *setFile()*
- Memoria, añadiendo uno o varios buffers o strings con las funciones *addDataBuffer()* y *addDataString()*.

Un bucle típico con *VXmlReader*:

```
importClass ("VXmlReader");

VXmlReader xml;
xml.setFile (myVFile);

while (!xml.atEnd())
{
    xml.readNext ();
    ... // procesar
};

if (xml.hasError ())
{
    ... // manejar el error
};
```

La clase contiene 44 funciones que te permitirán procesar documentos XML leyendo la

información del documento, del DTD, los tokens, controlar los errores que se produzcan, etc. A continuación se detallan la lista de funciones de esta nueva clase.

- addDataBuffer
- addDataString
- atEnd
- attributeValue
- characterOffset
- clear
- columnNumber
- documentEncoding
- documentVersion
- dtdName
- dtdPublicId
- dtdSystemId
- error
- errorString
- hasError
- isCDATA
- isCharacters
- isComment
- isDTD
- isEndDocument
- isEndElement
- isEntityReference
- isProcessingInstruction
- isStandaloneDocument
- isStartDocument
- isStartElement
- isWhitespace
- lineNumber
- name
- namespaceUri
- prefix
- processingInstructionData
- processingInstructionTarget
- qualifiedName
- raiseError
- readElementText
- readNext

- readNextStartElement
- setFile
- skipCurrentElement
- text
- tokenString
- tokenType
- VXmlReader

## Configura el control de selección de fichero y directorio

En la clase *VFilePathEditBrowser* se han añadido nuevas funciones que nos permiten configurar el control para que se puedan seleccionar más fácilmente ficheros o directorios.

Ahora podemos especificar el directorio por defecto, los filtros posibles, el filtro seleccionado y además de podemos saber si el control es de selección de directorio únicamente.

### VFilePathEditBrowser

- `isShowDirsOnly`
- `setDialogCaption`
- `setDialogDir`
- `setDialogFilter`
- `setDialogSelectorFilter`

Ejemplos de las nuevas funciones:

```
var pathEdit = theRoot.dataView().control("FILE");
pathEdit.setDialogCaption( "Seleccione fichero xbel 1.0" );
pathEdit.setDir( "C:\rss" );
pathEdit.setDialogFilter( "XBEL files (*.xbel);;XML files (*.xml)" );
pathEdit.setDialogSelectedFilter( "XBEL files (*.xbel)" );
```

## Consulta los estilos de los objetos

Con las nuevas funciones de la clase *VObjectInfo* podemos conocer toda la configuración de estilos de los diferentes tipos de objetos aportando información muy valiosa a la hora de procesar dichos objetos.

Ahora ya puedes saber:

- Si un proceso es accesible web con *isAccessibleFromWeb()*.
  - Si un control de edición tiene retardo para disparar *valueChanged* con *isDelayedEdition()*.
- 
- *isDragCopyEnabled*
  - *isDragMoveEnabled*
  - *isHardLock*
  - *isInserPoint*
  - *isOnlyOneExpanded*
  - *isPrivate*
  - *isRemoteFunction*
  - *isRunOnSimpleClic*
  - *isVertical*
  - *style*

## Cambia dinámicamente los objetos en ejecución

A continuación repasamos todas las propiedades que ahora puedes conocer y configurar, dinámicamente y en tiempo de ejecución, en los diferentes objetos de la plataforma con las nuevas clases y funciones.

### En todos los objetos de vista de datos

Puedes obtener el tamaño de la lista de registros que contiene con la función *listSize()* de la clase *VAbstractListView*.

### Rejilla avanzada

Usando la clase *VAdvancedGridListView* podemos imprimir el contenido de la rejilla con *print()* o generar un fichero PDF con contenido con *printToFile()*.

### Alternador

Con la clase *VAlternatorListView* podemos activar cualquiera de sus vistas con la nueva función *activateDataView()*.

### ComboView

La clase *VComboListView* nos da acceso a los objetos *ComboView* y con sus funciones podemos mostrar la lista de datos con *showPopup()* u ocultar la lista de datos con *hidePopup()*.

### CoverFlow

La clase *VCoverFlowListView* nos da acceso a los objetos *CoverFlow* y con sus funciones podemos saber si está visible el título con la información del registro usando la función *isTitleVisible()* y mostrar, u ocultar, el título del registro con *setTitleVisible()*.

### ListView

La clase *VListView* nos da acceso a los objetos *ListView* y con sus funciones podemos seleccionar todos los elementos con *selectAll()*, posicionarnos al principio de la lista con *scrollToTop()*, al final de la lista con *scrollToBottom()* o en una ítem concreto con *scrollToItem()*, además podemos configurar el tamaño del grid con *setGridSize()*.

### Informe

La clase *VReportListView* nos da acceso a los objetos *Informe* y con sus funciones podemos conocer la página en curso con *currentPage()*, conocer el número total de páginas con *pageCount()*, cambiar la página en curso con *setCurrentPage()*, conocer la orientación del informe con *orientation()* o cambiar la orientación dinámicamente con *setOrientation()*, controlar

el modo de visualización de la presentación preliminar ajustando la vista a la página con *fitInView()*, al ancho con *fitToWidth()*, saber si el informe está ajustado con *isFitting()*, imprimir el informe a papel con *print()* o generar un fichero PDF con *printToFile()*, preguntar por el modo de vista que se está aplicando con *viewMode()*, o cambiarlo *setViewMode()*, conocer el factor de zoom que se está aplicando con *zoomFactor()*, incluso cambiar el factor de zoom con *setZoomFactor()*.

### Casillero

La clase *VSlotListView* nos da acceso al objeto Casillero del que podemos conocer si el cálculo de filas y columnas es automático con *isAutoRowsOrCols()*, conocer si está configurado horizontal o verticalmente con *isHorizontalScroll()*, cambiar la orientación a horizontal y vertical con *setHorizontalScroll()* y *setVerticalScroll()*, conocer el número de filas o columnas con *getRowsOrCols()* y cambiar dinámicamente la configuración del número de filas y columnas con *setRowsOrCols()*.

### Árbol visor de tablas y Multivista en modo árbol

Con las clases *VTreeListView* y *VTreeMultiTableListView* ahora tenemos la posibilidad de contraer todo el árbol con *collapseAll()* o expandir el árbol completo con *expandAll()*, expandir el árbol hasta un nivel de profundidad específico con *expandToDepth()*.

Otras funciones nos permiten configurar dinámicamente si queremos que el doble clic permite expandir y contraer ramas del árbol con *expandsOnDoubleClick()* o cambiar este comportamiento con *setExpandsOnDoubleClick()*, conocer los píxeles de indentación que se usa para construir las ramas con *indentation()* o cambiar el número de píxeles con *setIndentation()*.

Otro grupo de funciones nos permite conocer si la cabecera del árbol está oculta con *isHeaderHidden()* y cambiar el estado para mostrar u ocultar la cabecera con *setHeaderHidden()*, conocer si los ítems son expandibles con *itemsExpandable()* o configurar dinámicamente si los ítems serán expandibles o no con *setItemsExpandable()*, conocer si el árbol muestra un icono flecha apuntando hacia la derecha si está cerrada la rama o hacia abajo si está abierto con *rootIsDecorated()* y hacer visible u ocultar el icono con *setRootIsDecorated()*, y conocer si el alto de cada ítem será uniforme en todo el árbol con *uniformRowHeights()* o cambiar esta configuración con *setUniformRowHeights()*.

## Cambia dinámicamente los controles en ejecución

Las nuevas clases y funciones te permiten un mayor control

### Combobox y Combobox de maestro

Con las clases *VComboBox* y *VBoundFieldComboBox* ahora puedes dinámicamente y en tiempo de ejecución abrir la lista de ítem con *showPopup()* o cerrarla con *hidePopup()*.

### Edit con y sin botón de menú

Usando las clases *VBoundFieldEdit* y *VBoundFieldEditBrowser* ahora puedes dinámicamente y en tiempo de ejecución abrir la vista de datos del control con *showPopup()* o cerrarla con *hidePopup()*, además de conocer si el control tiene registros en la lista de datos con *hasListDataView()*.

### Data Catcher Edit con o sin botón de menú

Usando las clases *VDataCatcherEdit* y *VDataCatcherEditBrowser* ahora puedes dinámicamente y en tiempo de ejecución abrir la vista de datos del control con *showPopup()* o cerrarla con *hidePopup()*, puedes conocer si el control tiene registros en la lista de datos con *hasListDataView()*, y cuando la lista de registros que muestra el control se carga en modo automático puedes forzar el recálculo de la lista de registros con *updateListDataView()*.

### ListBox

Con la clase *VListBox* y la nueva función *setGridSize()* puedes cambiar el tamaño en píxeles del grid.



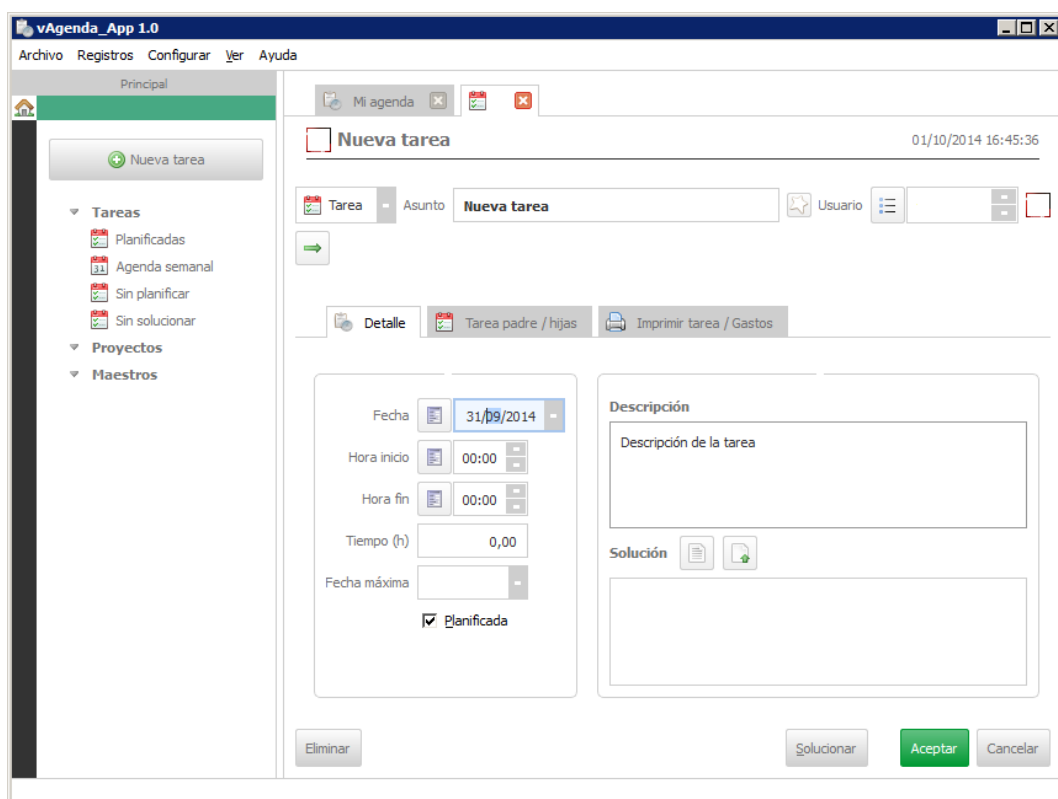
## Obtén directamente el `VFileInfo` de un fichero

Hemos añadido a la clase `VFile` la nueva función `info()` que te permite obtener el `VFileInfo` de cualquier fichero de la clase `VFile`, esto te facilita la programación al no tener que generar un fichero de la clase `VDir` para obtener el `VFileInfo` de un fichero.

## Introduce una fecha de forma sencilla

Se han realizado una serie de mejoras en el control de edición de fecha con el fin hacer más sencilla y rápida la introducción de datos.

Para ello, se ha reescrito el sistema de validación con el fin de permitir que ésta se realice tras la introducción del día y del mes correspondientes, con lo que podemos introducir fechas de forma más sencilla, sobre todo para los casos en los que comenzamos a editar una fecha de un mes que no contiene los días que queremos introducir.



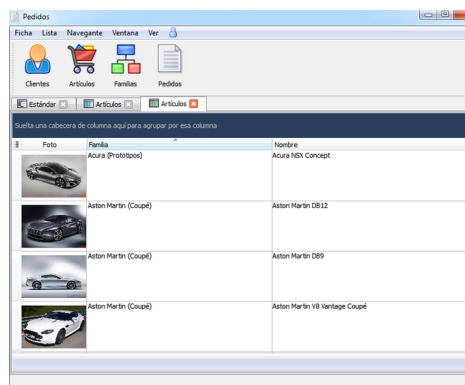
Un ejemplo: Editamos la fecha 30-abril-2014 para introducir 31 de marzo. A priori, abril no tiene 31 días, sin embargo, la validación no se producirá hasta introducir el mes, por lo que podremos escribir el día 31 y el mes de marzo a continuación, y en ese momento será en el que se validará la fecha.

Si la validación fuese dato a dato, como en versiones anteriores, no nos permitiría introducir el día, pero ya nos cambiaría el dato del día antes de editar el mes. De esta forma facilitamos la introducción rápida de datos, tan importante en muchas de vuestras aplicaciones.

## Optimización del pintado de la Rejilla Avanzada

En esta nueva versión de Velneo V7 7.16 hemos realizado mejoras y solucionado incidencias en la rejilla avanzada que nos van a permitir disfrutar más de este objeto, sobre todo gracias a su mejora de rendimiento. Entre las novedades más importantes encontrarás:

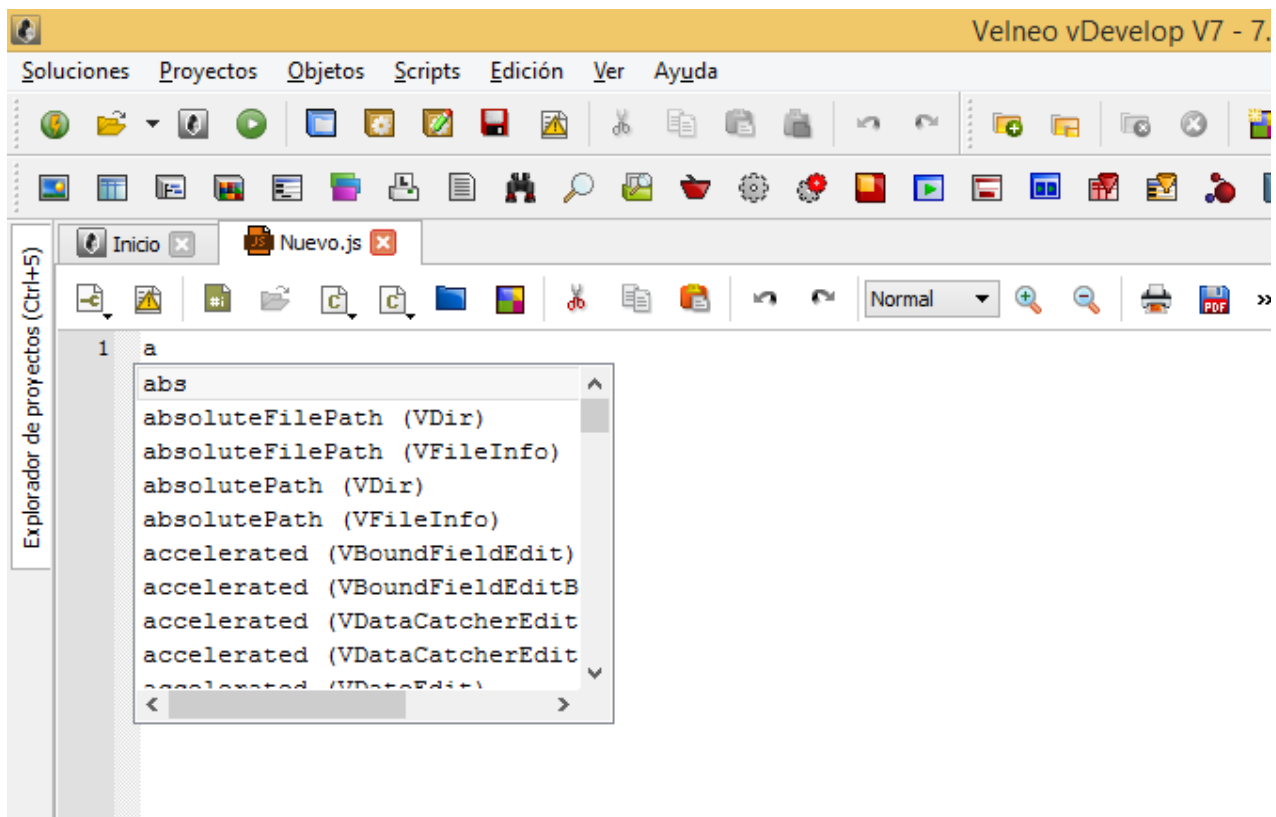
- Actualización de la librería y compatibilizado el código con Qt 5.3
- Mejorado el rendimiento de pintado evitando pintados innecesarios y bajando el uso de CPU para el repintado.
- Mejorada la presentación de información en la rejilla añadiendo espacio suficiente para que las letras no aparezcan pegadas a las líneas.
- Implementada la impresión de todas las cabeceras agrupadas de una rejilla.
- Se permite agrupar por una columna tras un cambio en la información que se muestra.
- Se evita que el diálogo de filtro se muestre de forma automática sin llegar a hacer uso de él en determinadas circunstancias.
- Se solucionan incidencias relativas a la fila que recibe el foco.
- Se soluciona una incidencia que hacía romper vClient debido a la gestión incorrecta de las columnas visibles.
- Añadidas nuevas opciones de comparación (mayor y menor que) en el cuadro de diálogo de filtro para columnas numéricas, de tiempo o fecha.
- Solucionadas incidencias en el filtro automático que hacían que no funcionara correctamente la comparación Menor, menor o igual, mayor y mayor o igual.
- Solucionada incidencia que hacía romper vClient al cerrar una rejilla avanzada en Mac.
- Solucionada incidencia que hacía romper vClient en Mac después de imprimir contenido agrupado.



## Optimización de la Edición de Scripts

En esta nueva versión de Velneo V7 7.16 hemos realizado mejoras y solucionado incidencias en este control, integrando la última versión de las librerías, lo que nos ha permitido optimizar el funcionamiento de este editor. A continuación detallamos las principales novedades e incidencias resueltas.

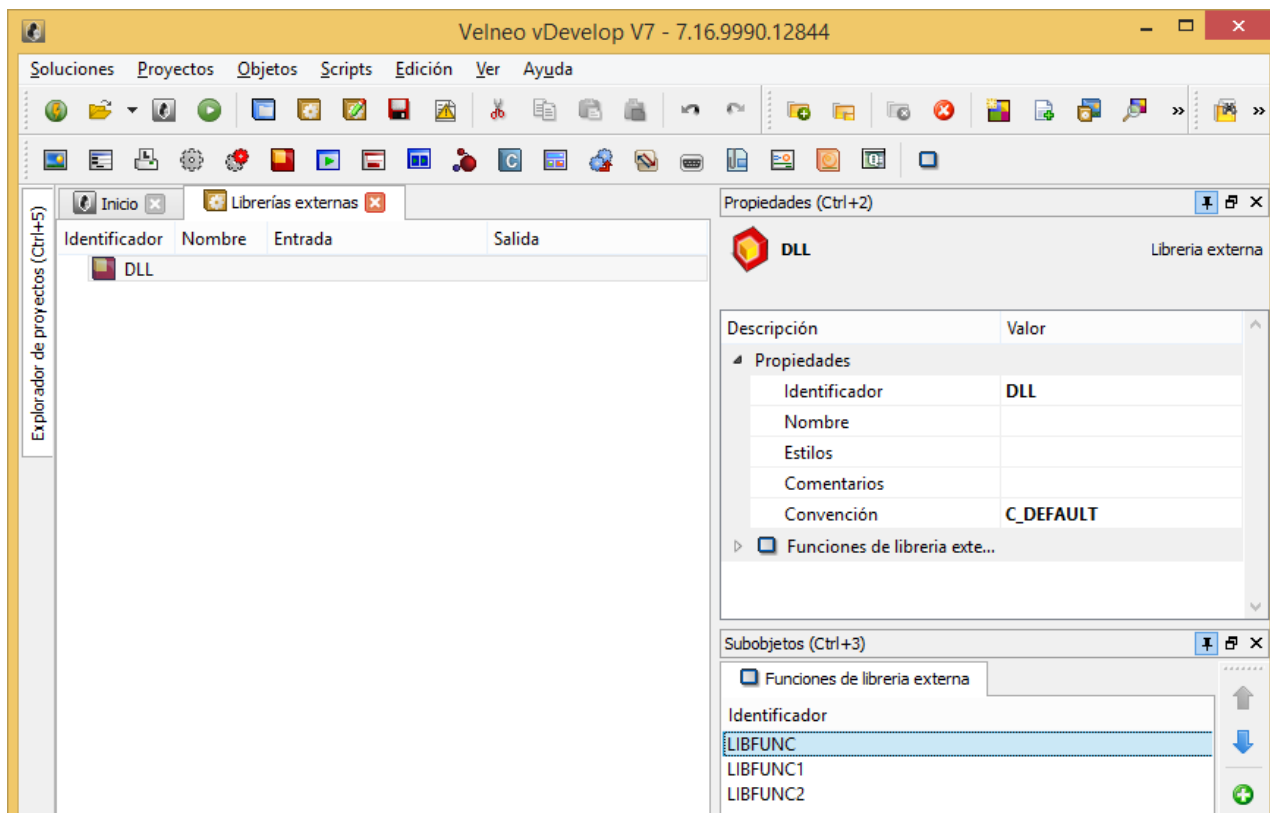
- Actualización de la librería y compatibilizado el código con Qt 5.3
- Actualización de la traducción.
- Mejora del rendimiento para líneas de gran longitud soportando centenares de miles de caracteres por línea.
- Mejora en la selección de contenido en sistemas operativos OSX.
- Evitadas incidencias en el autocompletado para entradas del tipo Texto.\* y TextoMás
- Solucionadas incidencias con barras de scroll.



## Mejoras de la autodetección en librerías externas

En la nueva versión de Velneo V7 hemos incluido mejoras y solucionado incidencias en el acceso a librerías externas que nos facilitarán su uso. A continuación detallamos las principales novedades e incidencias resueltas.

- Mejora en la autodetección de arquitectura vía defines de preprocesador.
- Uso de portasm (GNU, Apple, Microsoft) para ensamblador en x86, ppc32.
- Mejora de la convención de llamada cdecl x86 por medio de la cuál el stack siempre está alineado a 16 bytes.

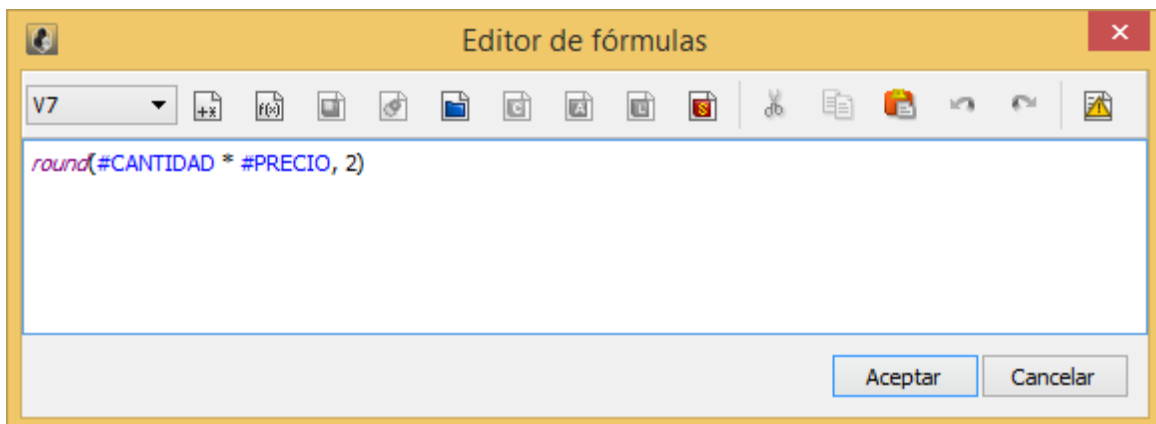


[Para más información accede a esta página.](#)

## Mejoras en procesamiento numérico

Hemos integrado la última versión del sistema de procesamiento numérico, lo que nos ha permitido introducir mejoras que solucionan incidencias en el mismo. A continuación detallamos las principales incidencias resueltas:

- Solucionada incidencia con funciones de signo en números decimales de precisión doble o quad.
- Solucionada incidencia en redondeo para operaciones logarítmicas cuando la precisión de la operación es insuficiente para el cálculo.
- Solucionada incidencia con retorno de exponentes en funciones de escalado que podía desbordarse al retornar un valor excesivo cuando el exponente del primer operador y la magnitud del segundo tenían al menos 9 dígitos.
- Solucionada incidencia con macro en reordenación que podía provocar un resultado incorrecto cuando la optimización hiciera uso de las reglas estrictas c99 de anti-aliasing.
- Solucionada incidencia con tamaño de buffer incorrecto en decimales tipo float.



[Para más información en detalle accede a esta página.](#)

## Mejora del soporte de sistemas operativos con Qt 4.8.6

En la nueva versión de Velneo V7 7.16 hemos integrado nuestro código con la última versión de las librerías Qt de la rama 4.8 en todos los sistemas operativos, lo que mejora la ejecución en las últimas versiones de los sistemas operativos.

Esta última versión aporta mucha estabilidad ya que nos permite solucionar muchas incidencias, además, se han aprovechado nuevas funcionalidades para mejorar la integración con los sistemas. La actualización de librerías aporta también mejoras de rendimiento en algunas funcionalidades que usamos más habitualmente.

A continuación detallamos las principales novedades e incidencias resueltas.

### General

- Mejorado el soporte de Windows 8.1
- Mejorado el soporte de Mac OS X 10.9 Mavericks
- Parche de seguridad para evitar LA incidencia con XML Entity Expansion Denial of Service (the 'Billion Laughs' attack)



### Interfaz

- Solucionada regresión en el color de un texto con selección del tabulador en OS X 10.6
- Implementado que el dock mantenga la posición cuando se despegan de la ventana.
- Reconocimiento del ratón del separador de ítem en menús para provocar el cerrado automático de sub-menús al moverse a un nivel superior.
- Solucionado problema de pintado de polígonos cuando el ancho es 0 ó 1.
- Solucionada fuga de memoria en componente del control de selección de fuente.
- Se tiene en cuenta la parte virtual del escritorio a la hora de determinar la visibilidad de un dock.
- Solucionada incidencia en control de edición con botones arriba y abajo que provocaba un salto a izquierda y derecha del cursor cuando se usaban los botones.
- Evitada desalineación de la región seleccionada cuando se centra el texto.
- Mejorado la gestión de multilínea de Listview en modo Icono.
- Restauración de ventanas maximizadas o en modo pantalla completa a la pantalla

correcta.

- Se aplica fuente definida en CSS a la etiqueta de botones.
- Evitados problemas en la presentación del cuadro de diálogo de impresión en Mac.
- Implementado modo duplex en impresoras CUPS.
- Evitado desbordamiento de memoria en lectura en el escalado de imágenes.

### **Network**

- Mejorada la gestión multi-thread de conexiones tcp.
- Mejorada la descarga de URL en modo multiparte.
- SSL: Actualizada la lista negra ANSSI de certificados intermedios.
- Evitados errores espúreos de certificados locales para SSL.

### **Interno**

- Mejoras en gestión de hilos: creación, seguridad, limpieza y destrucción.
- Implementado el soporte de apóstrofo en argumentos de comandos de Windows.
- Mejoras en la gestión de las cadenas url en las funciones de url.
- Mejora de llamadas recursivas a scripts.
- Mejora del acceso a bases de datos externas, evitando errores de memoria.

### **Linux**

- Mejorado el sistema de pintados usando el sistema nativo.
- Mejorado el uso de conexiones tcp.

### **Windows**

- Uso del estilo vista para asistentes.
- Mejorada apariencia de barra de scroll en Windows 8 adecuándose al estilo nativo.
- Mejora del mapeado de teclas virtuales evitando problemas de perder caracteres al escribir muy rápido, por ejemplo, ^e para obtener ê.
- Inicialización de color de texto en los ítems de menú.

### **Mac OS X**

- En el cuadro de diálogo de selección de fichero se muestran las aplicaciones .app como carpetas.
- Mejora en drag and drop solucionando memory leaks y el uso del botón derecho del ratón.
- Implementado manejo del maximizado de una ventana sin marco.
- Implementado soporte de formato nativo de fecha y hora de Mac OS X 10.7 y 10.8.
- Solucionada incidencia que provocaba no poder restaurar una ventana en Mac OS X 10.7 y posteriores.
- Mejora de ajuste de las toolbar en pantallas retina.
- Mejora en la integración con Mac OS X 10.9 Mavericks, haciendo uso de la fuente del



sistema de forma correcta.

- Implementado soporte de tamaño de papel configurable en Mac.
- Eliminado soporte de video de webkit en OS X debido a los problemas de estabilidad que generaba.
- Solucionado error tipográfico en el mapeado de la tecla F9 que implicaba que no se mostrará la tecla aceleradora.
- Mejorado el filtro del diálogo de selección de fichero cuando no comienza por \*.

[Para más información accede a esta página.](#)

## Más novedades

En la nueva versión de Velneo V7 hemos trabajado en mejorar la estabilidad y funcionalidades de la plataforma, sobre todo aquellas que repercuten en la mejora de experiencia de uso de las aplicaciones creadas con Velneo.

A continuación detallamos las más importantes:

### **Reintenta introducir la contraseña correcta cuando te equivoques al ejecutar una aplicación por medio de una acceso directo**

Cuando en un acceso directo a una instancia usamos la opción para forzar la petición de contraseña (\*), si el usuario se equivoca al introducir la contraseña, se presentará de nuevo la petición de contraseña, en lugar de mostrar la ventana de conexión.

De esta manera mejoramos la usabilidad del proceso de login, facilitando al usuario la conexión al servidor y ejecución de aplicaciones de forma más transparente para él.

### **Suprime el contenido de un control numérico para que su contenido pase a ser 0**

En un control de edición numérica basta eliminar su contenido para que pase a ser 0, sin obligarte para ello a introducir el valor 0 para ello. Esto sirve para cualquier contenido que tenga el control de edición numérica, sin importar el número de decimales, etc.

El control de edición numérica valida el contenido del control en todo momento, por lo que no te permite introducir caracteres que no sean numéricos, cualquiera intento de introducir un valor erróneo, llevará a retornar al valor anterior.

En el caso de eliminar el contenido, el control espera que contenga al menos un dígito, por lo que de forma automática se asignará un valor 0, lo que facilitará al usuario final la usabilidad de este control en nuestras aplicaciones y evitará confusiones.

### **Uso de validación y eventos con edit de maestro**

Cuando en un formulario en un edit de maestro se requería validación (validar permitiendo vacío o validar no permitiendo vacío), y se ejecutaba algún evento de interfaz, bajo determinadas circunstancias, al ejecutarse el evento, aunque no hubiese cambiado el campo enlazado al maestro, se recalculaban los contenidos iniciales basados en el mismo.

Solucionada la incidencia, la ejecución del manejador de evento no supone el recálculo de los contenidos iniciales, manteniendo los valores hasta que se producen cambios.

### **Encuentra de forma sencilla un objeto en los combos de selección de objeto**

En los combos de selección de objeto encontrarás los objetos ordenados por identificador en orden alfabético ascendente, aunque pertenezcan a proyectos distintos que por herencia tengas acceso.

De esta forma te será más sencillo encontrar el objeto que quieres usar. Esto se aplica a todas las selecciones de objeto: Acciones, alternadores, árboles, blocs, búsquedas, casilleros, cestas, etc.

### **Trabaja con filtros en la rejilla avanzada**

Cuando usábamos una rejilla avanzada y usábamos la opción de filtrado rápido, al pulsar escape la rejilla daba un error y vClient se cerraba.

Solucionada la incidencia, se evita el error y se configura un opción en el cuadro de diálogo para su cierre.

### **Renombra variables locales sin afectar al código javascript**

En un proceso donde tenía definida una variable local y donde usaba fórmulas JavaScript en las propiedades de los comandos de instrucción, al renombrar la variable hacía también la sustitución en el código JavaScript coincidente.

Solucionada la incidencia, cuando cambiemos el identificador a la variable local, el código javascript no se verá afectado, y únicamente se realizarán cambios dónde se use la variable local, lo que nos facilitará nuestro trabajo como desarrolladores.

### **Eliminado el botón de ayuda (F1) donde no tiene uso**

En el asistente de fórmulas aparecía un botón ayuda (F1) que no tenía utilidad alguna y producía confusión.

Solucionada la incidencia, el botón se ha quitado para no generar confusión.

### **Usa la señal valueChanged con los editores de objeto Texto y Texto enriquecido en una rejilla editable**

Los controles de edición Objeto texto y Texto enriquecido en rejilla editable no enviaban la señal valueChanged si la modificación se realizaba usando el botón y el cuadro de diálogo correspondiente para modificar el dato.

Solucionada la incidencia, cuando modificas el contenido de estos controles haciendo uso de los editores correspondientes, al aceptar el cuadro de diálogo se dispara la señal, que puedes aprovechar para disparar un manejador de evento para ampliar la funcionalidad de tus aplicaciones.

### **Convierte un control de edición de maestro con vista de datos en un control de edición estática**

Cuando copiamos un control de edición de maestro al que le asignamos una Vista de datos y a continuación cambiamos el contenido del control para apuntar a una tabla estática, aunque no se ve, la propiedad Vista de datos sigue teniendo asignado el objeto elegido.

En ejecución, esto produce un error y vClient rompe.

Solucionada la incidencia, en la conversión se tiene en cuenta este punto y se evita, por lo que en ejecución podremos usarlo sin preocuparnos de realizar manualmente esta operación.

### **Pasa valores a variables en procesos procesos javascript en 2º y 3º plano por medio de VProcess**

Hasta ahora sólo era posible pasar valores a variables a procesos javascript en 1º plano por medio de VProcess.

En esta nueva versión podrás pasar valores a procesos en 2º y 3º plano, y también retornar al proceso llamador valores obtenidos (recuerda que únicamente en el caso de 3º plano, ya que en 2º plano no se espera por la finalización del proceso).

### **Configura en una columna de rejilla el número de decimales con que se muestra el resultado de una fórmula**

Cuando el contenido de una columna de una rejilla es una fórmula numérica, al visualizar el resultado se ajustará el número de decimales al que hayamos configurado para la columna.

### **Mejora de la ejecución del objeto Protocolo TCP en 3º plano**

Se ha implementado un enganche virtual al objeto Protocolo TCP ejecutado en tercer plano con el fin de que se gestionen por medio de ese enganche todas las ejecuciones que se realizan en el objeto TCP.

Esto hace que estén disponibles además objetos como theApp. Ahora podremos hacer uso del objeto theApp en nuestros procesos javascript, esté donde esté ejecutado el objeto Protocolo TCP.

### **Actualización de librerías de Open SSL para navegadores (HTTPS)**

Actualización de las librerías para soporte de SSL en navegadores para incluir revisiones de seguridad, que incluye soluciones para las siguientes incidencias:

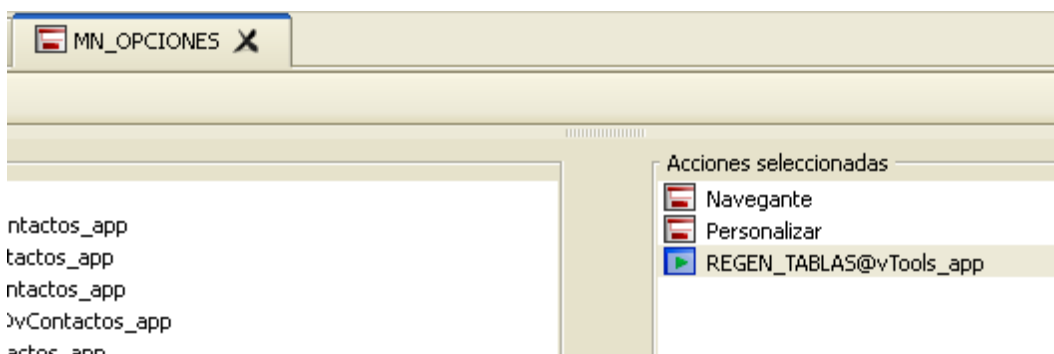
- SSL/TLS MITM vulnerability (CVE-2014-0224)
- DTLS recursion flaw (CVE-2014-0221)
- DTLS invalid fragment vulnerability (CVE-2014-0195)
- SSL\_MODE\_RELEASE\_BUFFERS NULL pointer dereference (CVE-2014-0198)
- SSL\_MODE\_RELEASE\_BUFFERS session injection or denial of service (CVE-2010-5298)
- Anonymous ECDH denial of service (CVE-2014-3470)
- CVE-2014-0076: Fix for the attack described in the paper "Recovering
- OpenSSL ECDSA Nonces Using the FLUSH+RELOAD Cache Side-channel Attack"

Más información: [http://www.openssl.org/news/secadv\\_20140605.txt](http://www.openssl.org/news/secadv_20140605.txt)

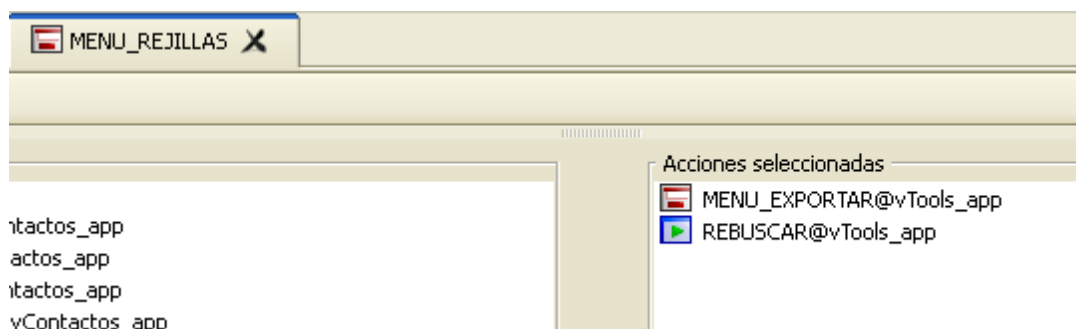
## Velneo vTools

vTools es una solución desarrollada en Velneo V7 que incorpora una serie de funcionalidades útiles para el programador y que puede incorporar de forma sencilla dentro de sus propias aplicaciones, haciendo el desarrollo con Velneo V7 más Life is Soft que nunca.

La forma más sencilla de utilizar las funcionalidades incluidas en vTools es usar las acciones o menús que incorpora vTools. Por ejemplo, si queremos incluir una opción de menú para regenerar los índices y datos de todas las tablas, tan sólo tenemos que añadir a nuestro menú la acción REGEN\_TABLAS, tal y como se muestra en la siguiente imagen:



*Incluida la acción dentro de nuestro menú*



*Incluido el menú exportar dentro de nuestro menú*

## Acciones sobre lista

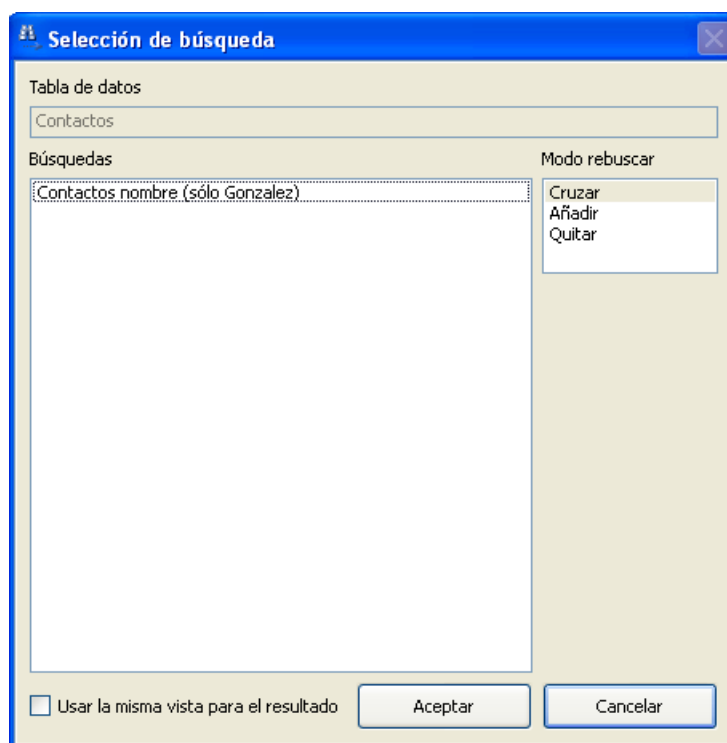
Se incorporan una serie de acciones sobre lista, que además agrupamos en el menú “Utilidades de lista” (LST\_TO0). De esta manera, podemos fácilmente agregar ese menú a nuestro menú contextual asociado a los objetos de lista o a una toolbar.

Dentro de esas acciones de lista tenemos, entre otras:

### Rebuscar

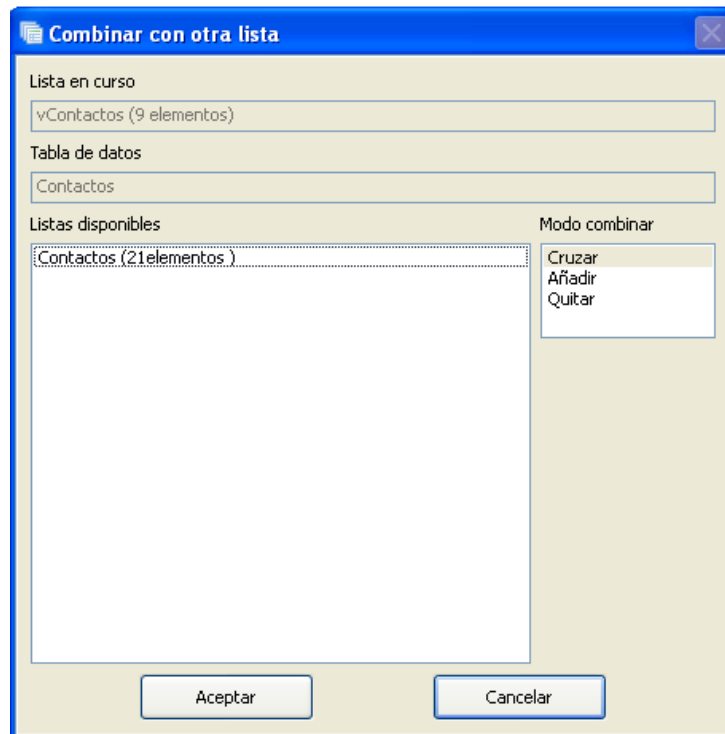
Esta acción (RBS) nos permite, a partir de una lista, realizar operaciones (cruzar, añadir o quitar) de sus elementos contra el resultado de otra búsqueda de la misma tabla.

En la siguiente imagen, vemos como cuando lanzamos la acción rebuscar desde una lista de la tabla contactos, nos muestra todas las búsquedas públicas de la tabla contactos y nos da la opción de cruzar, añadir o quitar el resultado de la búsqueda seleccionada a la lista de origen. Además, podemos hacer que el resultado aparezca en la misma vista de origen o en una nueva vista.



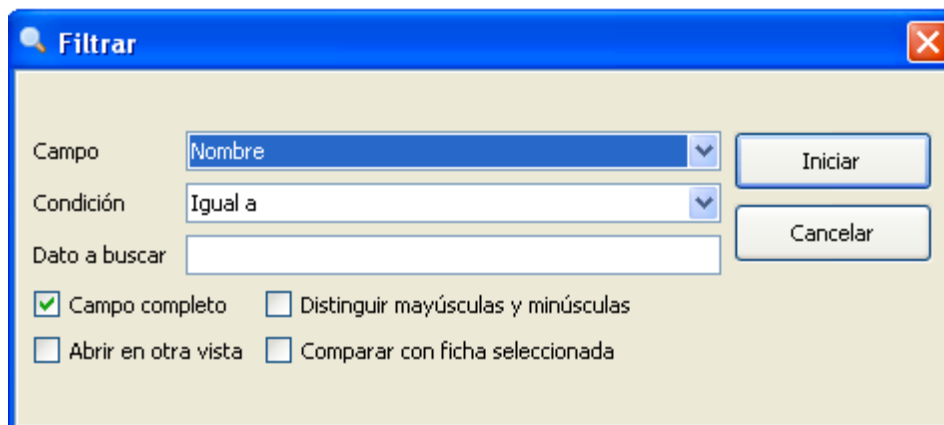
### Combinar

Esta acción (CMB) nos permite combinar la lista actual con otra de las listas abiertas. Por ejemplo, si tenemos una lista de contactos con 9 elementos y otra con 21, podemos añadir, cruzar o quitar los elementos entre sí.



## Filtrar

Esta acción (FLT) nos permite mostrar un cuadro de diálogo de filtrado tal y como aparece en la siguiente imagen:



Los campos que aparecen en el primer combobox son todos los de la tabla asociada a la lista de la vista desde que se lanza la acción, que no sean privados, y que sean de los siguientes tipos:

- Alfabético
- Numérico
- Fecha



- Hora
- Tiempo
- Booleano
- Fórmula numérica
- Fórmula alfabética
- Fórmula fecha
- Fórmula tiempo

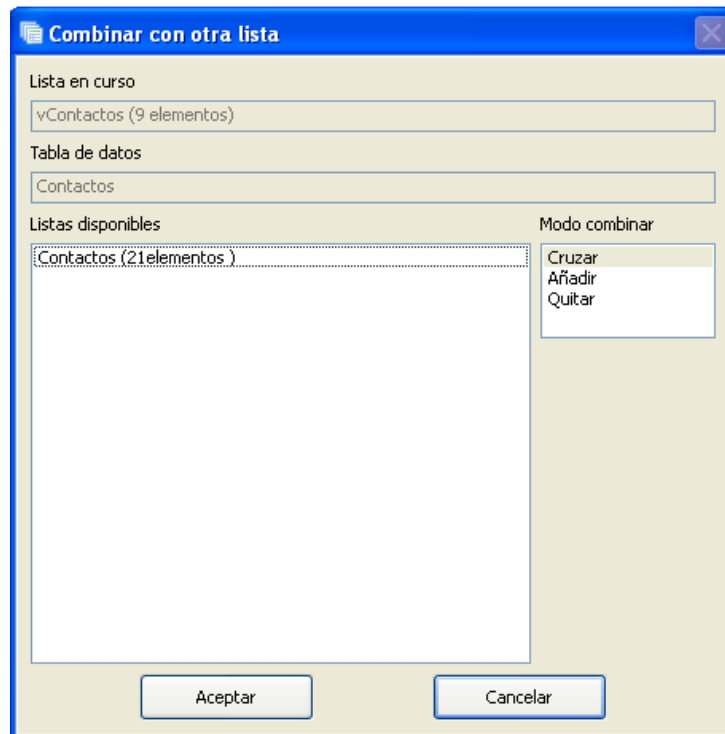
En función del tipo del campo seleccionado en el primer combobox, aparecen unas determinadas condiciones en el segundo combobox y el control de introducción del valor es del tipo acorde al del campo. Por ejemplo, si el campo elegido es de tipo fecha, tendremos las condiciones igual, distinto, menor y mayor y el control para la introducción del valor será de tipo fecha.

En el caso de que seleccionemos un tipo de campo alfabético, podremos indicar si queremos comparar con el campo completo o por partes. Por ejemplo, si tenemos desmarcado “Campo completo” y buscamos *poli*, nos encontrará tanto *poliglota* como *polígono*.

Además, en el caso de campos alfabéticos, podemos indicar si queremos distinguir entre mayúsculas y minúsculas. Si le indicamos que no, en caso de buscar *Ana*, nos encontrará *Ana* y *ana*.

Si marcamos la opción “Abrir en otra vista”, el resultado del filtrado aparecerá en una vista nueva. En caso contrario, el resultado se mostrará usando la misma vista de origen.

Si marcamos “Comparar con la ficha seleccionada”, podremos comparar el campo elegido con el registro que tengamos seleccionado en la vista desde la que se lanza la acción.



## Exportaciones

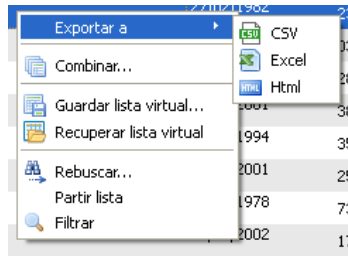
En vTools contamos con una serie de exportaciones automáticas de rejillas, de tal manera que exporta todos los campos que se visualizan en la rejilla a determinados formatos.

Podemos, o bien incluir el menú `Exportar` que está en vTools, el cual incluye todas las opciones de `Exportar` disponibles, o añadir la acción correspondiente a la exportación que nos interese incluir en nuestra aplicación. Muy Life is Soft.

### Exportar a CSV

Esta funcionalidad nos permite exportar todos los datos que tengamos en una rejilla (excepto las imágenes) a un documento de tipo csv.

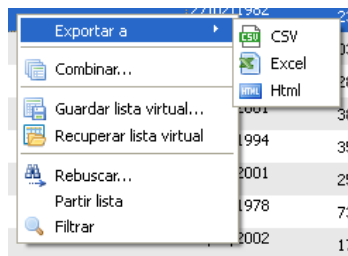
Podemos, o bien incluir el menú `Exportar` que está en vTools, el cual incluye todas las opciones de `Exportar` disponibles, o añadir a nuestro menú o cualquier otro disparador la acción `EXP_CSV`.



### Exportar a Excel

Esta acción nos permite llevar todos los datos (excepto las imágenes) de una rejilla a una hoja excel. Esta funcionalidad sólo está operativa para sistemas operativos Windows que tengan Microsoft Excel instalado.

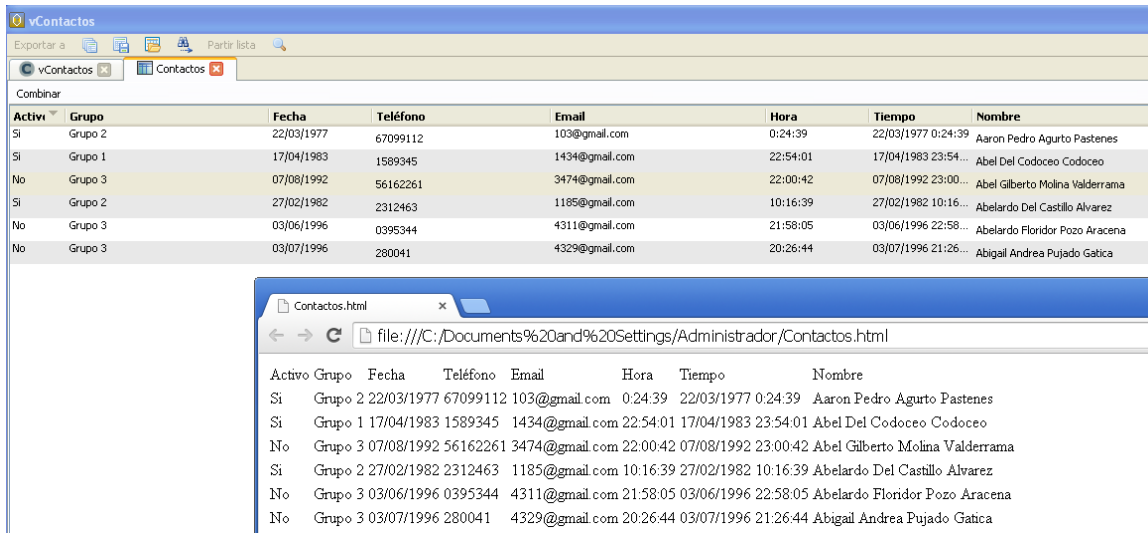
Podemos, o bien incluir el menú Exportar que está en vTools, el cual incluye todas las opciones de Exportar disponibles, o bien añadir a nuestro menú o cualquier otro disparador la acción EXP\_XLS.



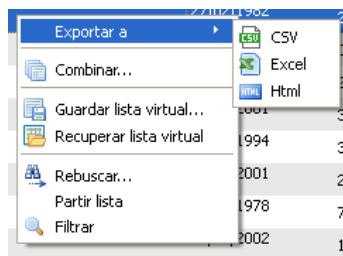
### Exportar a Html

Esta acción nos permite exportar todas las columnas (excepto las imágenes) de la rejilla mostrada a una tabla html sencilla, sin formato.

Como vemos en la siguiente imagen, la lista de contactos que aparece en segundo plano se ha exportado al html que aparece en primer plano.



Podemos, o bien incluir el menú Exportar que está en vTools, el cual incluye todas las opciones de Exportar disponibles, o bien añadir a nuestro menú o cualquier otro disparador la acción EXP\_HTML.



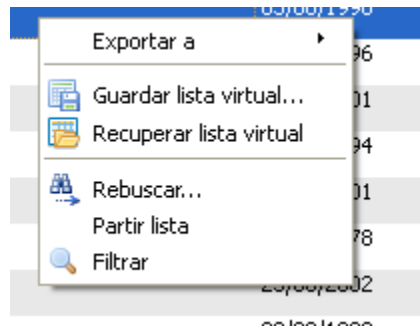
## Listas virtuales

Una lista virtual es una lista de registros que queremos almacenar en disco para posteriormente volverla a usar. Por ejemplo, obtenemos una lista de clientes después de haber hecho varias operaciones de filtrado, cruces, etc., y quiero guardar esa lista o compartirla con un compañero. Es una funcionalidad orientada principalmente al usuario final y que hace tu aplicación más Life is Soft.

Disponemos de acciones para guardar una lista virtual en disco y recuperarla en cualquier otro momento.

Estas acciones se incluyen de forma predeterminada en el menú “Utilidades de lista”, aunque podemos usar las acciones correspondientes que se suministran para incluirlo en cualquiera de nuestros menús.

Para guardar una lista, tan sólo tendremos que utilizar la acción LST\_VRT\_GRB, asociada a una toolbar de lista o a un menú derecho de listas. Para la acción de cargar una lista virtual usamos LST\_VRT\_CRG. Por ejemplo, aquí las tenemos asociadas al menú derecho de las rejillas:

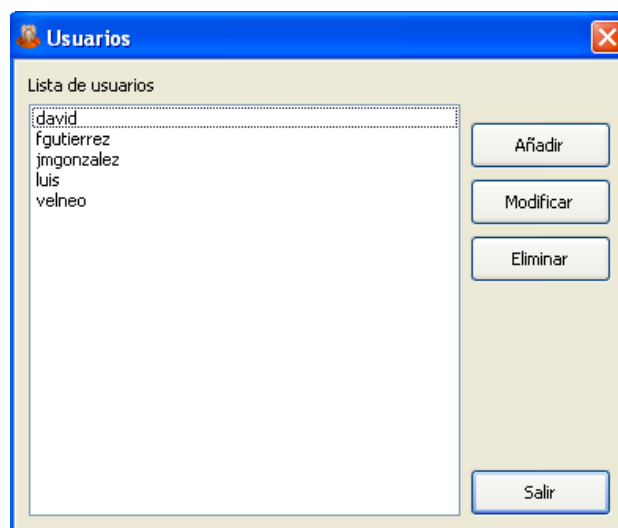


## Gestión de usuarios

Con la versión Velneo V7 7.16, aparece un conjunto completo de funcionalidades para gestionar los usuarios y grupos de nuestro vServer desde el vClient.

En vTools, hemos utilizado una parte de estas nuevas funciones para que puedas incorporar en tu aplicación la opción de que un usuario final pueda agregar otro usuario a tu instancia, sin necesidad de tu intervención o de tener acceso con Velneo vAdmin, haciendo tu aplicación más Life is Soft.

Con esta acción (USRS), lanzamos un formulario con la lista de usuarios del vServer que tiene el siguiente aspecto:





Con el fin de garantizar la seguridad en entornos multiinstancia, sólo se muestran los usuarios de los grupos a los que pertenece el usuario que lanza la acción.



Sólo los usuarios marcados como Supervisor en el vServer pueden hacer cambios en los usuarios.

Si añadimos o modificamos un usuario, nos muestra un formulario como el siguiente donde podemos indicar las propiedades de dicho usuario.



En la lista de grupos disponibles sólo se muestran los grupos a los que pertenece el usuario que lanza la acción.

Una configuración típica para un cliente final sería de este tipo:

Usuario	Administradores	Usuarios avanzados	Usuarios
administrador	x		
luis@empresa.com		x	x
carmen@empresa.com			x
rafael@empresa.com			x

En este caso, nos tenemos que encargar en nuestra aplicación usando algún tipo de seguridad que sólo el usuario dentro del grupo “Usuarios avanzados” pueda lanzar la acción USRS.

De esta manera los grupos disponibles donde luis podrá incluir a un usuario serán “Usuarios avanzados” y “Usuarios”.

## Sistema de activación

En Velneo nos gusta escuchar a nuestra comunidad y en la iteración 7.16 hemos hecho realidad una petición que muchos de vosotros nos habíais transmitido: hacer más **life is soft que nunca** nuestro sistema de activaciones.

Velneo cuenta con un sistema de validación de las licencias de Velneo vServer para cerciorarse de que éstas se utilizan acorde a las normas de uso establecidas en las mismas.

Pues bien, a partir de ahora, ya no tendrás que solicitar las molestas activaciones por soporte o por correo electrónico: el sistema simplemente funcionará, siempre y cuando no trates de usar la misma licencia en dos máquinas diferentes de forma simultánea.

Por lo tanto, cuando necesites cambiar de máquina un servidor o reactivar una licencia porque ha cambiado el hardware de la máquina donde está instalada, podrás instalar y activar la licencia sin ningún problema.

¡Ahora, las reactivaciones, son más life is soft que nunca!



## El ERP más personalizable

La aparición de nuevas versiones es el proceso natural de evolución del software, sin embargo, no todas las versiones son iguales, en algunas ocasiones las versiones suponen un punto de inflexión en la evolución de un producto.



Sin duda, la versión 7.16 de Velneo vERP es un punto de inflexión ya que es el resultado de un intenso trabajo realizado durante muchos meses con un claro objetivo, convertirlo en un producto más fácil de entender, implantar y personalizar por parte de los desarrolladores y más fácil de ejecutar por los usuarios finales de las aplicaciones desarrolladas con Velneo vERP como base. En definitiva, hacer que Velneo vERP sea más Life is Soft que nunca.

Para conseguirlo teníamos en mente que Velneo vERP es un componente más de la plataforma completa de desarrollo y que para un importante número de desarrolladores Velneo vERP es la pieza estratégica que les permite abordar sus proyectos basados en un estándar personalizable que les facilita la creación de su propio ERP para abordar soluciones estándar y a medida. Por este motivo, a partir de esta versión Velneo vERP tendrá la misma numeración que la versión de la plataforma con la que ha sido desarrollada, por lo tanto la versión 7.16 de Velneo vERP es la sucesora de su anterior versión la 1.5.

Velneo vERP 1.5 era un punto de partida fantástico ya que su funcionalidad, su estructura de base de datos y su diseño eran una sólida base para abordar la nueva versión. Por lo tanto nuestro trabajo se focalizó en la filosofía Life is Soft manteniendo el alma de Velneo vERP 1.5 que tanto éxito ha estado cosechando entre los desarrolladores de la plataforma.

Velneo vERP es un producto diseñado por programadores para programadores, en esta versión simplificar ha sido una palabra clave a todos los niveles. Hemos podido comprobar como la frase “menos es más” se cumplía en cada una de las novedades abordadas, vERP cuenta ahora con una única solución que unifica en 2 proyectos todos los objetos con el fin de hacerlo más fácil de implantar a la vez que se convierte en un código más sencillo de entender y por lo tanto de personalizar.

Te invitamos a que conozcas todo lo que Velneo vERP puede aportar a tu estrategia de negocio. Te recomendamos que veas el seminario [Gana más dinero con tu propio ERP](#) para conocer a fondo las ventajas del concepto de software estándar personalizable. En la [ficha de Velneo vERP](#) encontrarás toda la documentación sobre las novedades de la versión 7.16, la documentación para los programadores con el documento de anexos de cambios realizados en la versión, el

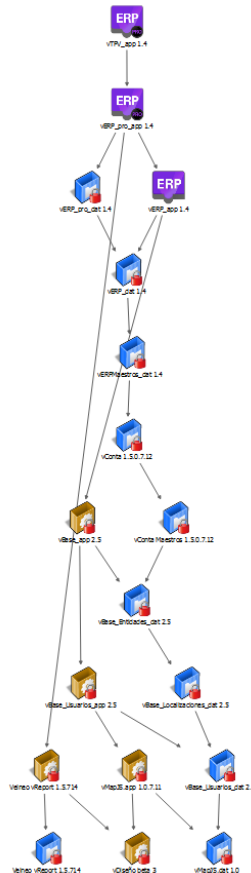
manual del usuario de Velneo vERP diseñado para que lo puedas personalizar con tu imagen, el manual de migración de la versión 1.5 a la versión 7.16 y el manual del programador.

Bienvenido a la nueva versión Velneo vERP 7.16 ¡Qué la disfrutes!

## Una gran solución ERP

Siempre se dice que una imagen vale más que mil palabras...

Esta es la estructura de soluciones y proyectos de la anterior versión 1.5 de Velneo vERP.



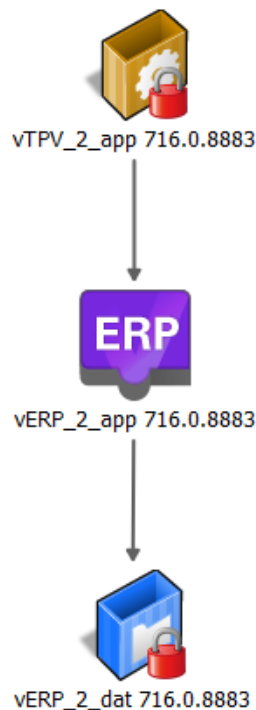
La hemos reducido para que pueda entrar en la página. Si te estás preguntando si esta estructura es funcional, la respuesta es sí, funciona correctamente. Si la pregunta es ¿Dónde está cada objeto? ¿Esto no es complicado de conocer y comprender antes de poder trabajar con este código?

La respuesta es más que evidente, necesitarás tiempo para conocerla, dominarla y saber dónde está cada uno de los objetos que quieras personalizar, además cuanto mayor es la estructura de herencias mayor es el nivel de programación requerido para comprenderla y dominarla.

¿Podíamos hacer algo para facilitar su comprensión, dominio y personalización?

La respuesta es SÍ, con mayúsculas.

Lo primero que hicimos fue simplificar al máximo la estructura de proyectos y soluciones. Pasamos de varias soluciones a una sola y de la anterior estructura de 18 proyectos a una nueva con realmente un proyecto de datos y otro de aplicación, más un proyecto adicional requerido para el marco de arranque del módulo de TPV que solo tiene un objeto marco. El resultado de esta simplificación es el siguiente.



Hemos invertido cientos de horas en esta simplificación, pero ha merecido la pena, estamos convencidos que todos los programadores que usen Velneo vERP se van a sentir mucho más cómodos sabiendo que todos los objetos de datos están en un proyecto y que todos los objetos de aplicación también los podrán localizar en un solo proyecto.

Gracias al uso de los inspectores y buscadores del editor vDevelop, localizar cualquier objeto de Velneo vERP 7.16 es cuestión de segundos.

## Un ERP diseñado por programadores para programadores

Para que Velneo vERP se convierta en una plantilla de trabajo fácil de usar tenía que aplicar estrictos criterios de programación y normalización de todos sus objetos. Aquí jugábamos con la ventaja de que la versión 1.5 de Velneo vERP ya contaba con bastantes criterios aplicados, sin embargo, le hemos dado una vuelta de tuerca para aplicar más buenas prácticas y normalizaciones, pensando siempre en los programadores que lo van a usar.

Todos los programadores que han participado en el desarrollo de esta versión 7.16 han programado “igual” para que a tí te resulte más sencillo entender y encontrar cualquier objeto o código.

Uno de los mayores esfuerzos realizados consistió en añadir un sufijo a todas las tablas de Velneo vERP con el claro objetivo de evitar futuros problemas de coincidencia de identificadores, tanto en las tablas de los futuros módulos de vERP como con las tablas creadas por los programadores que ya usan y personalizan Velneo vERP.










Este cambio nos obligó a repasar todos los identificadores de objetos visuales con origen, ya que su prefijo es el nombre de la tabla.

Pero no nos hemos quedado ahí, no hemos escatimado esfuerzo y hemos revisado también todos los campos e índices de todas las tablas con el fin de aplicar la normalización de abreviaturas.







Sin embargo, no estábamos dispuestos a que estas mejoras tuviesen la contrapartida de obligar a los suscriptores de vERP a realizar labores de programación para la migración de datos. Por este motivo, hemos creado en todas las tablas los trasposos de campos correspondientes, además hemos creado un manual y unos procesos de migración automática para que conozcas los pasos a realizar para la migración de la versión 1.5 a la versión 7.16 y usar las herramientas que ponemos a tu disposición. El objetivo era muy claro, todos estos cambios tenían que ser Life is Soft para nuestros suscriptores.

En el documento de anexos de cambios realizados en la versión hemos documentado todas las tablas y objetos que han cambiado, identificando cada uno de los cambios. Todo documentado al máximo detalle, porque sabemos lo importante que es para los programadores contar con con esa información.





















El resultado visible de todo este trabajo lo podemos apreciar si vemos la estructura de carpetas de los proyectos de aplicación y datos que han quedado muy reducidos, tanto la estructura de carpetas del proyecto de datos:

- ▷  **Esquemas**
- ▷  **Tablas**
- ▷  **Tablas estáticas**
- ▷  **Índices complejos**
- ▷  **Variables**
- ▷  **Constantes**
- ▷  **Procesos**
- ▷  **Funciones**
- ▷  **Dibujos**

Como la estructura de carpeta de proyectos de aplicación:

- ▷  **Marco**
- ▷  **Gestión**
- ▷  **Maestros**
- ▷  **Configuración**
- ▷  **TPV**
- ▷  **Recursos**

Además de quedar muy reducidos, la organización de las carpetas está en el mismo orden del menú general para que sean más fáciles de localizar todos los objetos:

- ▷  **Marco**
- ▾  **Gestión**
  - ▾  **Ventas**
    - ▷  **Presupuestos de venta**
    - ▷  **Pedidos de venta**
    - ▷  **Albaranes de venta**
    - ▷  **Facturas de venta**
  - ▾  **Compras**
    - ▷  **Pedidos de compra**
    - ▷  **Albaranes de compra**
    - ▷  **Facturas de compra**
  - ▾  **Almacén**
    - ▷  **Existencias**
    - ▷  **Trasposos entre almacenes**
    - ▷  **Movimientos de almacén**
    - ▷  **Stock a fecha**
- ▷  **Maestros**
- ▷  **Configuración**
- ▷  **TPV**
- ▷  **Recursos**

## ERP estándar personalizable

Una vez cumplido el objetivo de simplificar el código, Velneo vERP 7.16 nos planteaba un excitante reto tecnológico.



¿Cómo conseguir que los suscriptores puedan personalizar Velneo vERP para convertirlo en su propio ERP sin necesidad de tocar el código de Velneo vERP para poder seguir actualizando las nuevas versiones?

En esta versión hemos dado grandes pasos para conseguir convertir Velneo vERP en un estándar personalizable. Ahora podrás heredar Velneo vERP 7.16 y sin tocar su código podrás:

1. Personalizar la barra de menús en función de permisos de grupos de usuarios.
2. Personalizar el menú general, añadiendo, sustituyendo y eliminar las opciones que quieras, además de asignar a cada opción los permisos por grupos de usuarios.
3. Personalizar todas las rejillas y formularios. Ocultar y mostrar controles y columnas de rejilla; añadir, quitar o sustituir subformularios de cualquier contenedor de pestañas, e incluso ejecutar código personalizado al arrancar cualquier rejilla o formulario.
4. Personalizar los informes de compras, ventas y maestros en ejecución.
5. Configurar por usuario si la tecla INTRO avanza control o cierra formulario.
6. Personaliza el diseño de la aplicación ajustando las CSS a tu gusto.

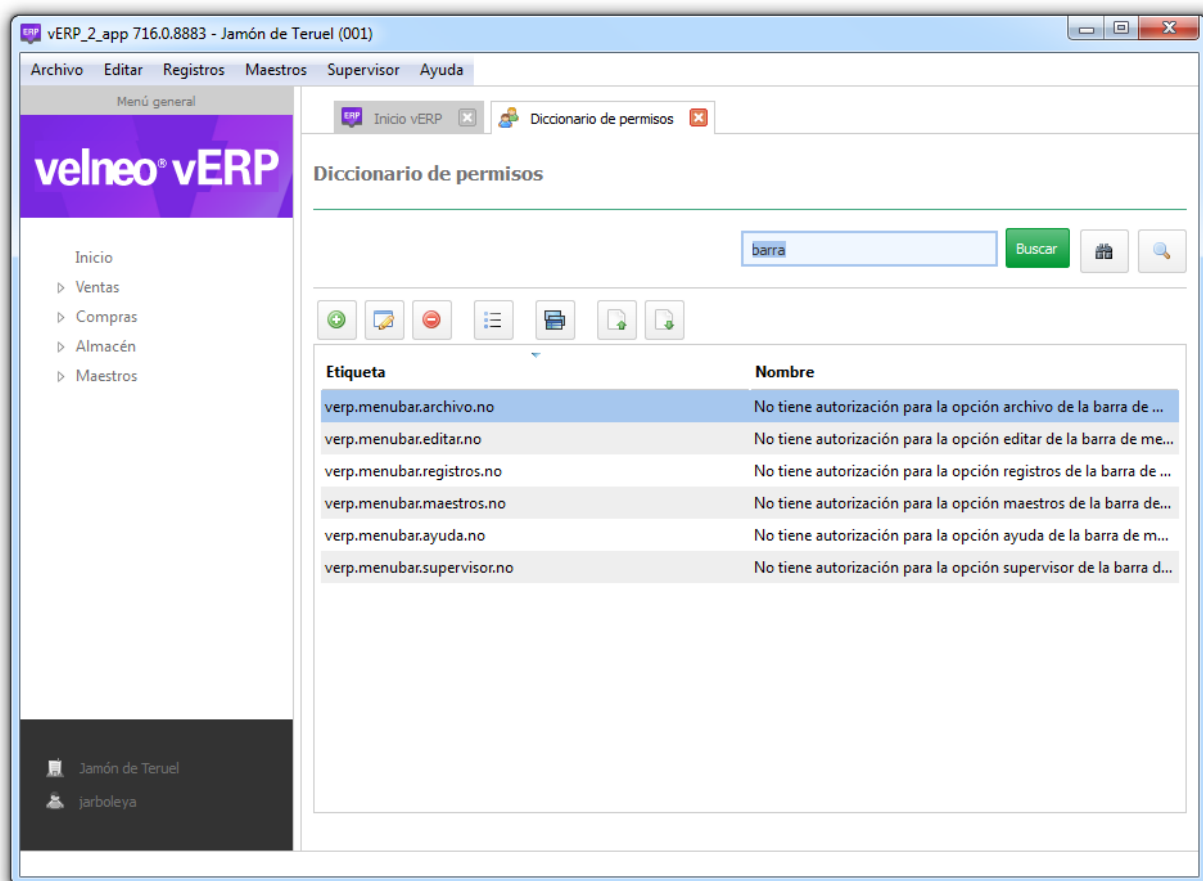
Esta ha sido una primera fase de personalización que ha cubierto las necesidades principales, pero el viaje no termina aquí y ya estamos trabajando en nuevas opciones de personalización que te permitirán personalizar más objetos de Velneo vERP sin tocar su código en próximas versiones.



## Personalizar la barra de menú de tu ERP

Velneo vERP incluye un sistema de personalización de la barra de menú que permite ocultar las opciones que no deben ver determinados usuarios. Esta configuración de permisos se aplica a nivel de grupos de usuarios.

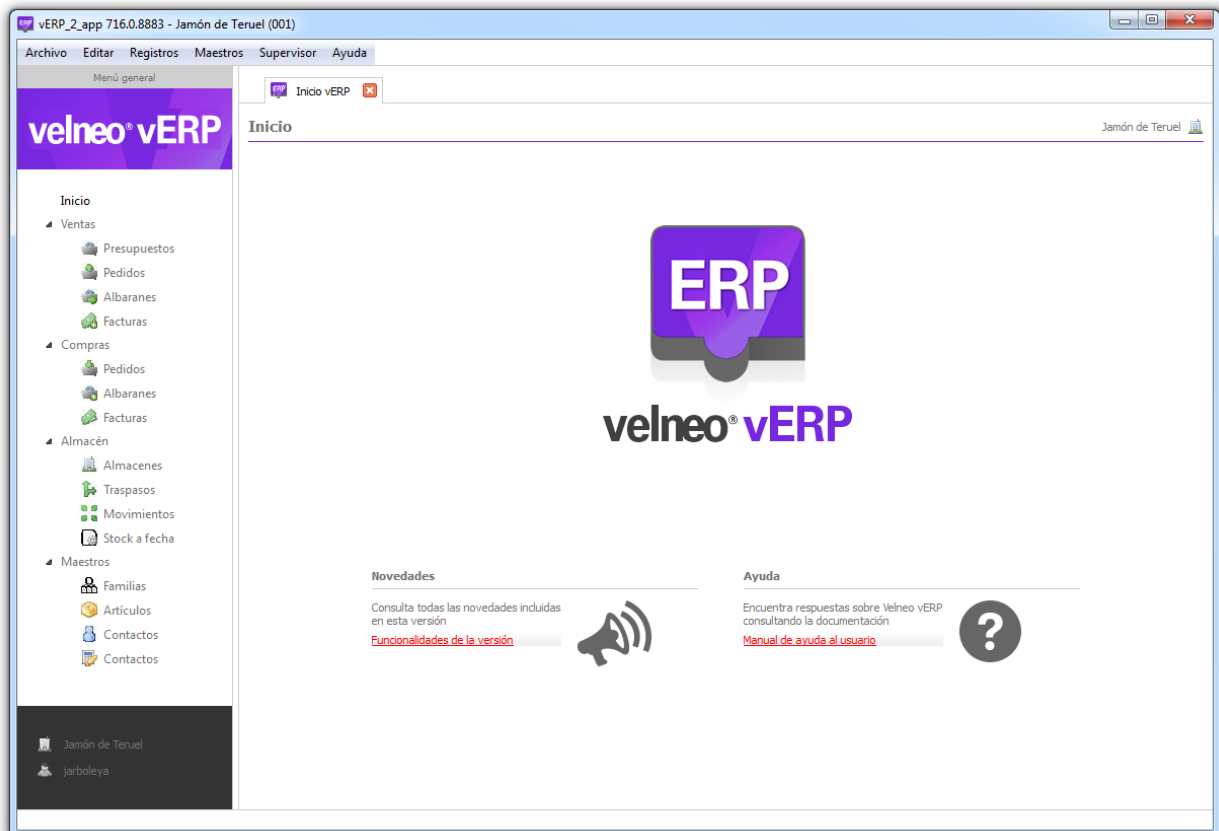
Además puedes añadir más opciones utilizando los puntos de inserción o las nuevas funciones del API de Velneo.



Hemos aprovechado el nuevo sistema para reestructurar las opciones existentes en los submenús de la barra de menú. En esta reestructuración hemos creado las opciones Maestros y Supervisor para facilitar la gestión de permisos globales de esas opciones. Además, el diccionario de permisos incluye la posibilidad de configurar las autorizaciones para cada una de las opciones individuales de cada menú.

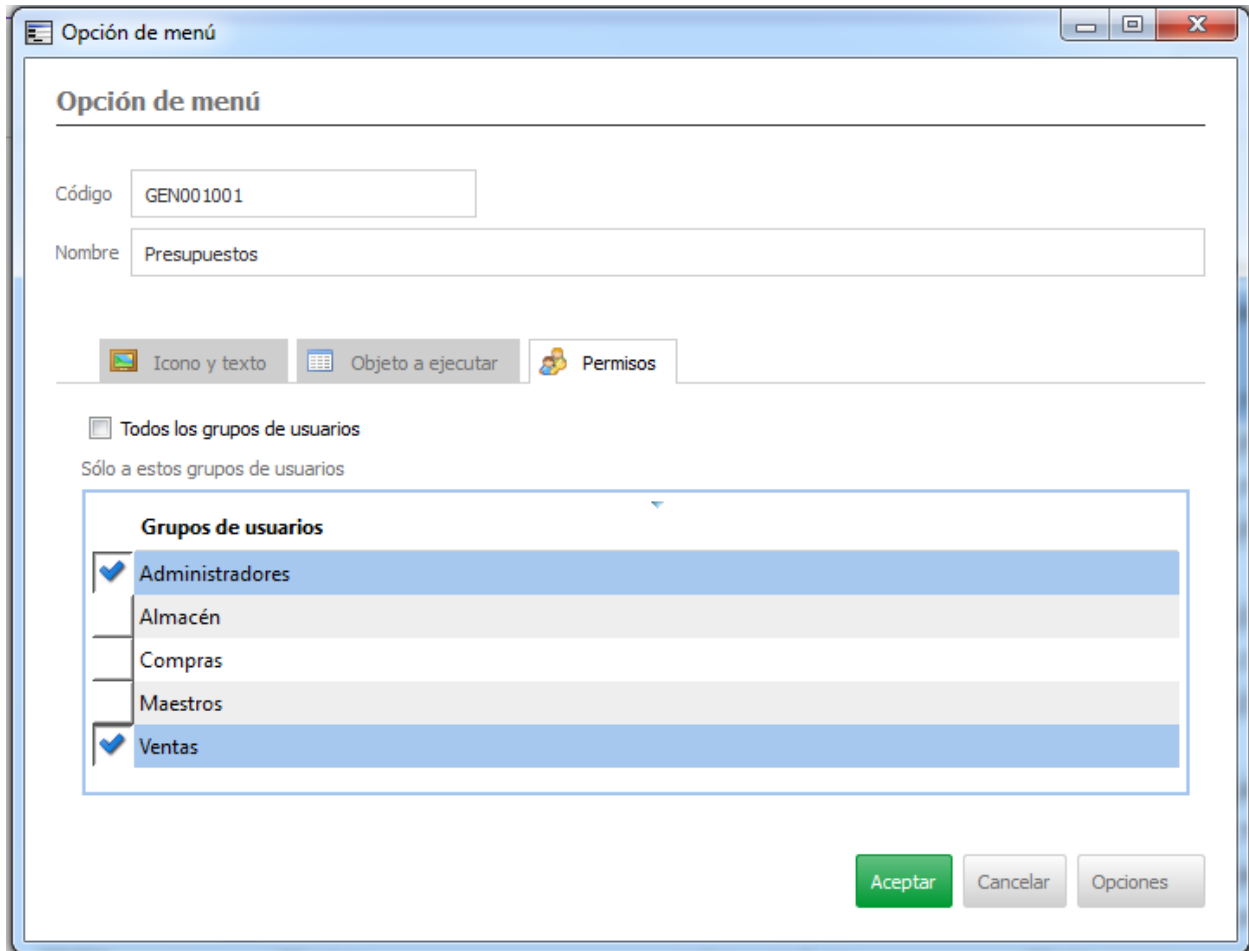
## Personaliza el menú general de tu ERP

Velneo vERP incluye un nuevo sistema de menú dinámico el árbol del menú general incluye opciones que pueden ser configuradas dinámicamente en tiempo de ejecución.



El nuevo sistema de configuración del menú permite añadir, eliminar o sustituir la opciones que por defecto incluye Velneo vERP.

Además, podemos configurar para cada opción qué grupos de usuario verán la opción o la tendrán oculta.

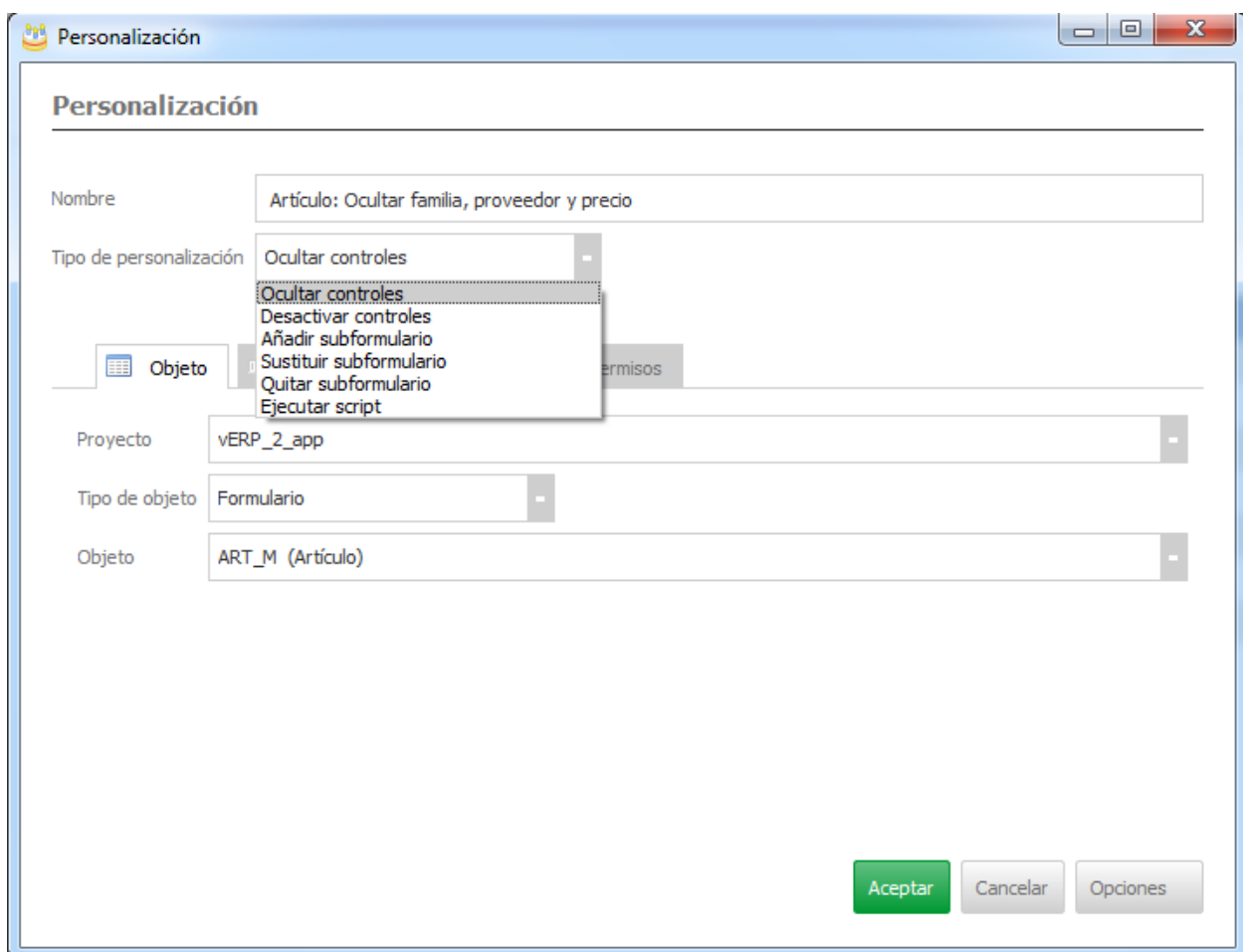


La personalización de opciones del menú general permiten sustituir las opciones de vERP por aquellas que queremos personalizar específicamente para nuestros clientes finales. Esta funcionalidad es fundamental para poder sustituir cualquier funcionalidad de Velneo vERP por otra personalizada.

## Personaliza todos los formularios y grids de tu ERP

Velneo vERP incluye un nuevo sistema de personalización dinámica, en tiempo de ejecución, que permite configurar todos los formularios y rejillas.

En cualquier formulario podremos ocultar o desactivar los controles que nos interese. Además, podremos añadir, quitar y sustituir subformularios incrustados en controles separadores o pilas de formularios.



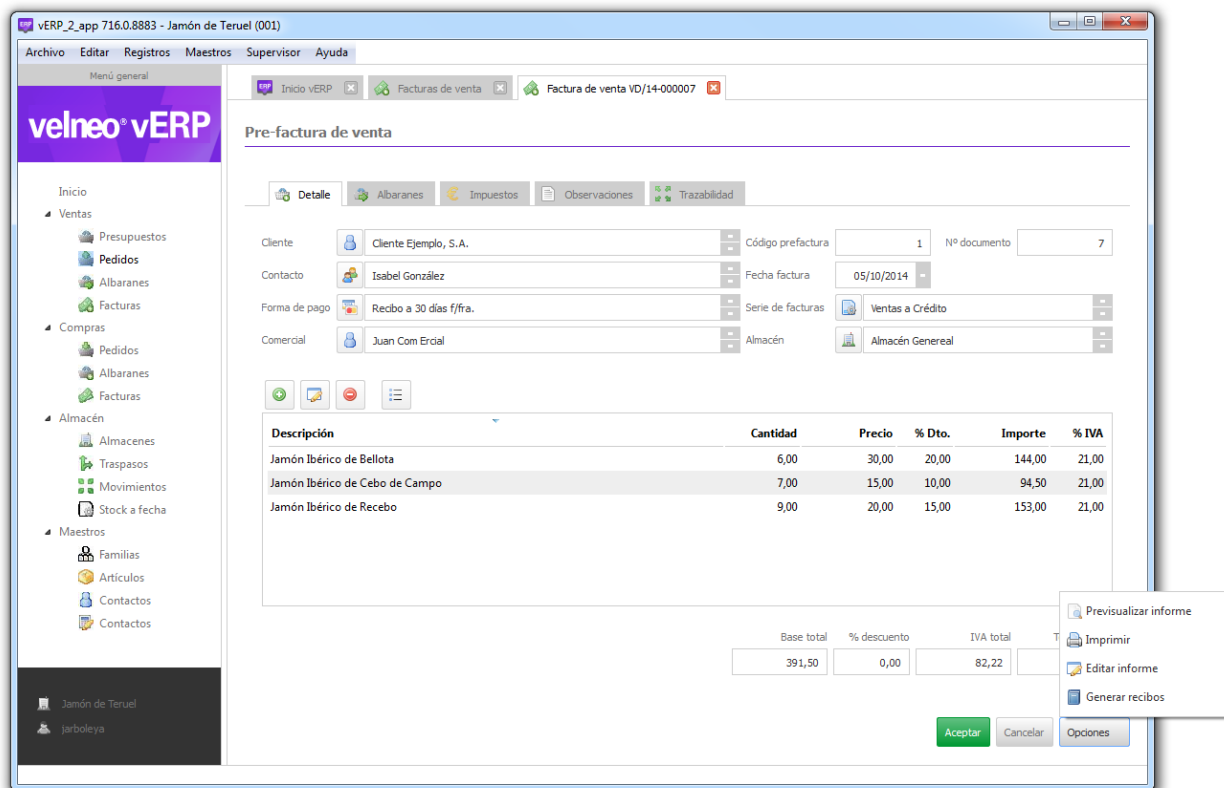
Al igual que en la configuración de las opciones del menú general, la personalización que apliquemos a las rejillas y formularios se puede configurar a nivel de grupos de usuarios.

La opción ejecutar script permite que aquellos programadores más avanzados puedan ejecutar el código JavaScript que deseen al iniciarse una rejilla o formulario, una opción realmente poderosa.

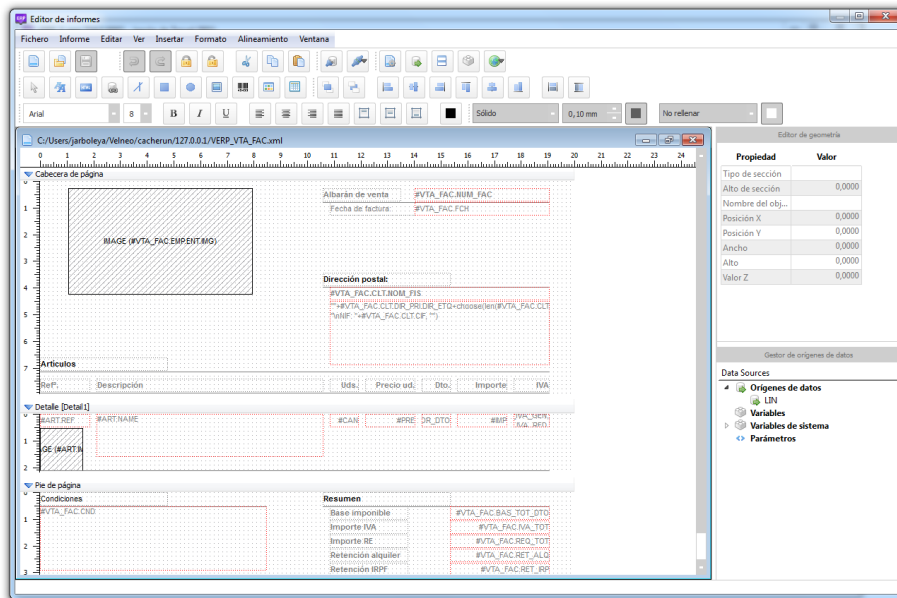
## Personaliza todos los informes de tu ERP

Velneo vERP incluye el sistema de informes personalizables Velneo vReport. Gracias a la inclusión de este sistema los informes de compras, ventas y algunos maestros pueden ser personalizados tanto por los programadores como los usuarios finales de las aplicaciones desarrolladas con Velneo vERP.

Por ejemplo, en el botón opciones de los formularios de compras y ventas se permite tanto previsualizar, imprimir o editar el informe si disponemos de los permisos necesarios para ver esta opción de menú.

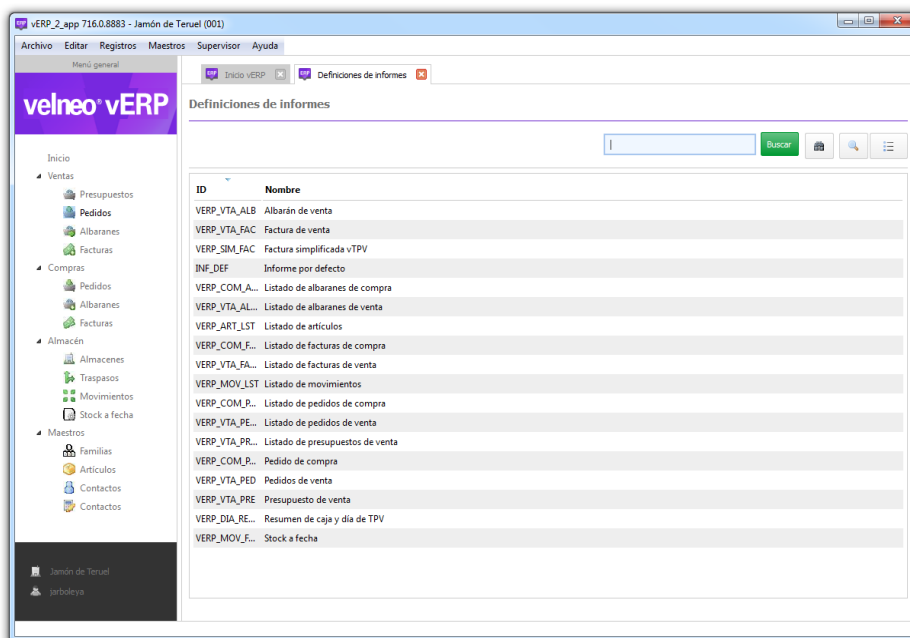


Al utilizar la opción editar informe se visualizará el editor vReport para configurar el informe según nuestras necesidades.



Los cambios serán guardados automáticamente en la tabla de definición de informes a la que podremos acudir en cualquier momento para editar, duplicar o eliminar cualquier informe.

El sistema de informes personalizados permite que los programadores e incluso los usuarios finales puedan añadir nuevos informes.



## Personaliza la tecla INTRO de tu ERP

El equipo de desarrollo de vERP ha considerado que sería muy interesante que cada usuario pueda configurar el funcionamiento de la tecla INTRO en la aplicación, pues hay usuarios que prefieren que la aplicación funcione como la web, usan la tecla tabulador para avanzar de campo y la tecla INTRO para aceptar el formulario. Sin embargo, otros usuarios prefieren la tecla INTRO avance de campo, y en este caso hemos incluido la tecla de función F12 para aceptar el formulario, además de poder pulsar INTRO sobre el botón aceptar.

Cada usuario podrá configurar la funcionalidad de la tecla INTRO que más le guste.

### Usuario

---

Código	<input type="text" value="1"/>	
Fecha de alta	<input type="text" value="08/07/2014"/>	
Nombre	<input type="text" value="docampo"/>	
	<input checked="" type="checkbox"/> Tecla Intro acepta el formulario	

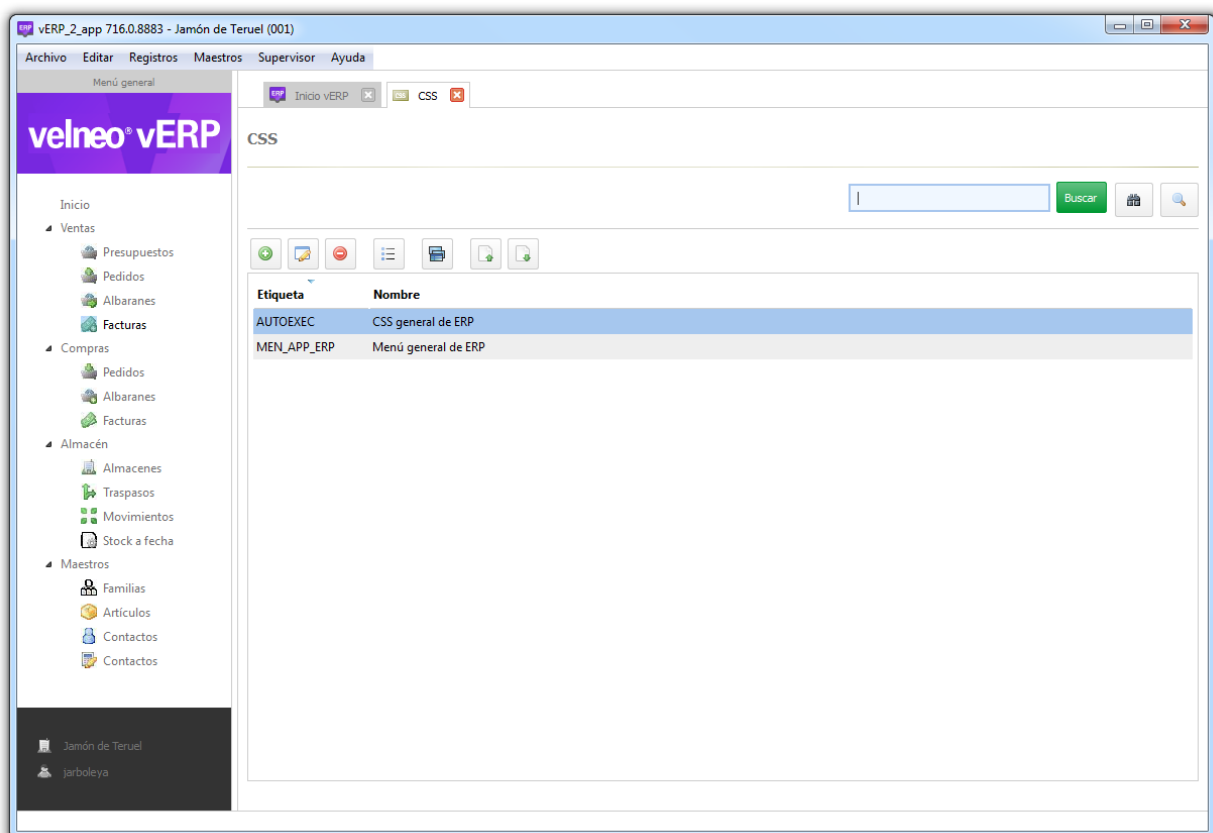


## Personaliza el diseño de tu ERP con CSS

El diseño minimalista de Velneo vERP es fruto del trabajo del equipo de diseño de Velneo. La configuración visual de la aplicación está basado en la aplicación de estilos en cascada (CSS).

En esta versión estos estilos que se aplican al arrancar la aplicación no están configurados de forma estática en el código del marco, sino que se entregan en una tabla diseñada para contener tanto las CSS que entregamos con Velneo vERP como cualquier otra CSS que desees aplicar en tus personalizaciones.

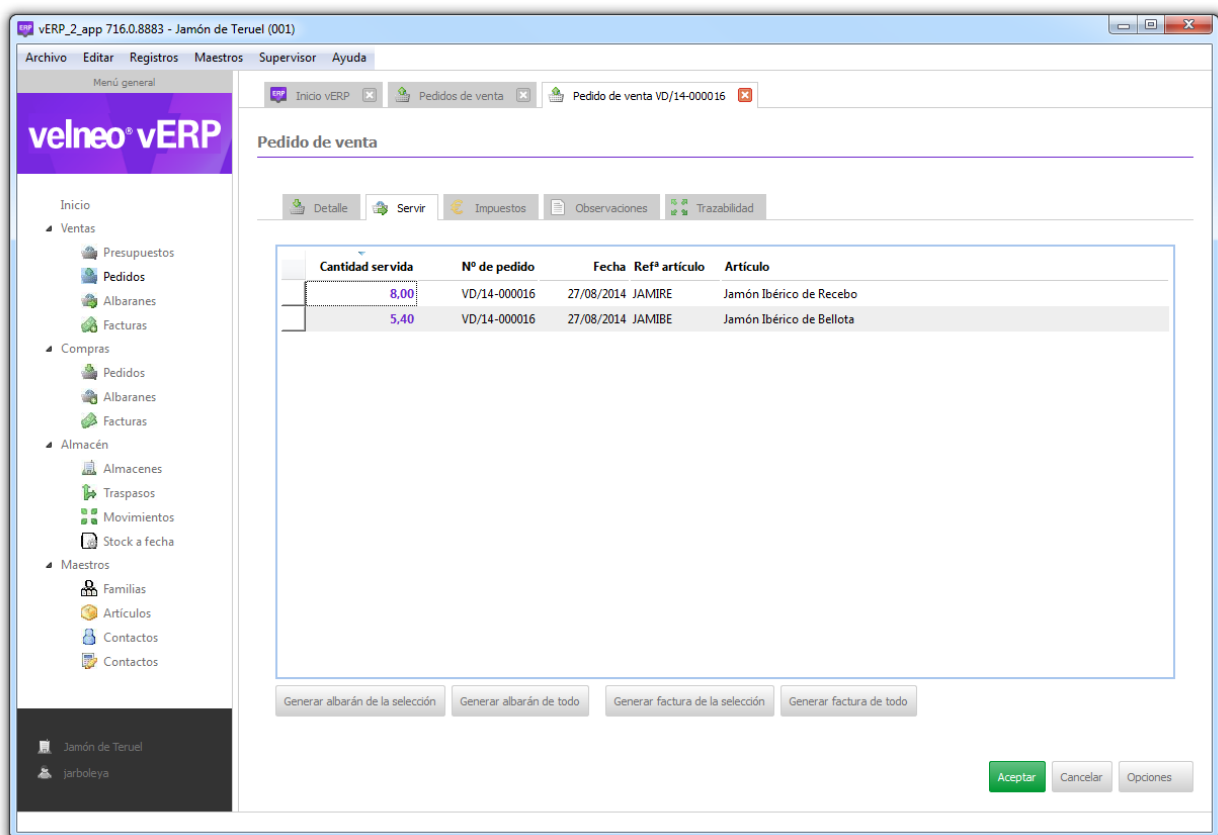
En el menú supervisor se encuentra la opción que accede al menú de configuración de las CSS.



## Usabilidad en ciclos de compras y ventas de tu ERP

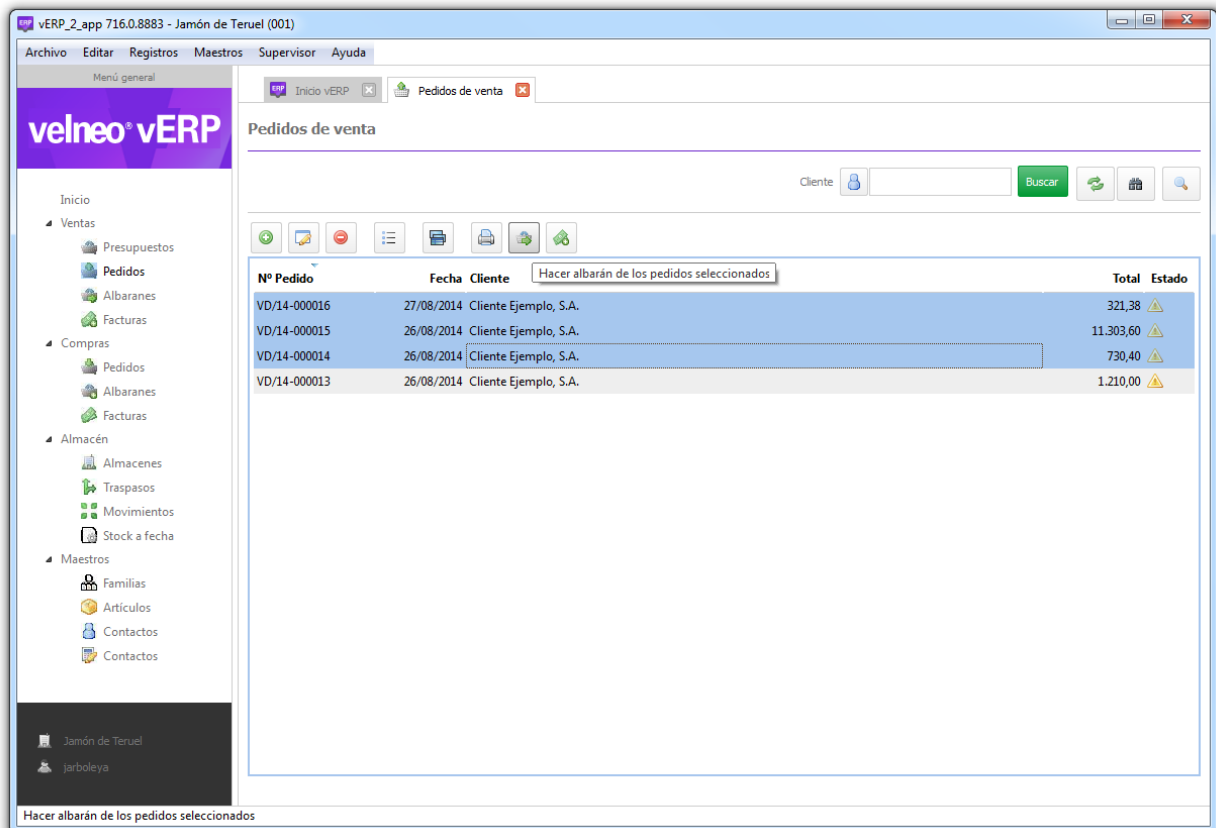
Velneo vERP trata de cubrir el 80% de la funcionalidad estándar que todas las empresas necesitan. En esta versión nos hemos concretado en mejorar la usabilidad del sistema de gestión de los documentos que conforman los ciclos de compras y ventas.

Hemos incluido en cada documento la pestaña necesaria para generar el siguiente documento de forma rápida y sencilla. Ahora esta funcionalidad está en el lugar donde el usuario final la necesita y no requiere cerrar el documento en curso para realizar la operación que desea realizar.



Ahora puedes generar pedidos desde el presupuesto, generar albaranes y facturas desde el pedido, añadir líneas a servir desde el pedido y facturar el albarán desde el formulario, tanto en los documentos de compras como de ventas.

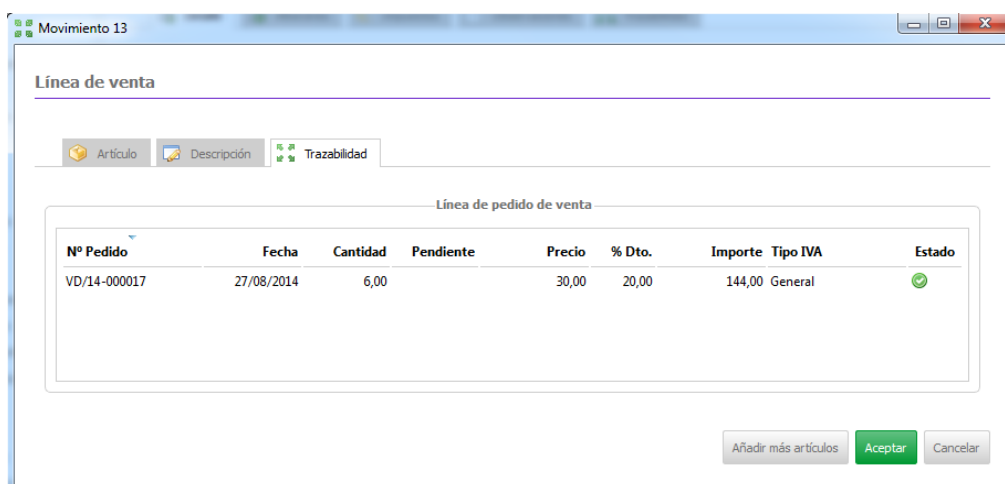
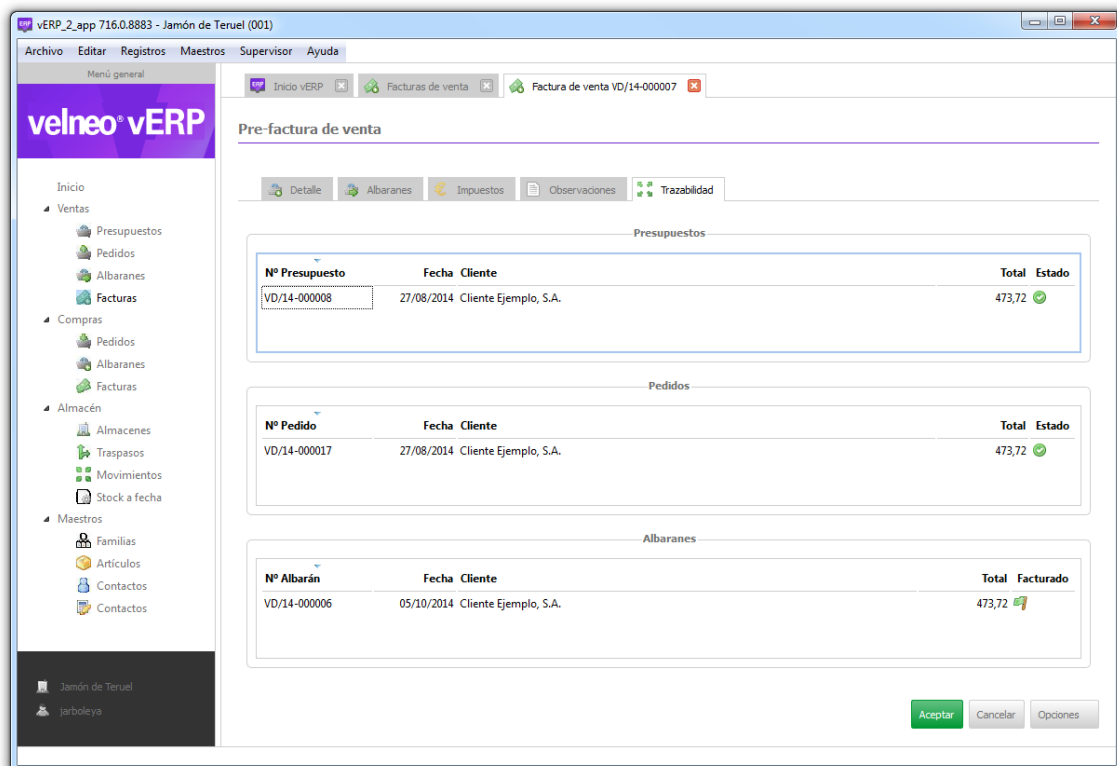
Además, existen opciones para generar documentos desde el menú de cada documento, esto permite realizar estas operaciones sobre un grupo de documentos de forma sencilla.



Otro aspecto a destacar es que estos ciclos son bidireccionales. Es decir que podemos tanto generar documentos hacia adelante en el ciclo como eliminar los documentos, siempre que cumplan las condiciones adecuadas. Por ejemplo, se controla que una factura ya contabilizada no puede ser eliminada, ni un albarán ya facturado, etc. Este sistema permite que si se cumplen las condiciones adecuadas podamos eliminar los documentos generados regresando los documentos al estado anterior de forma automática.

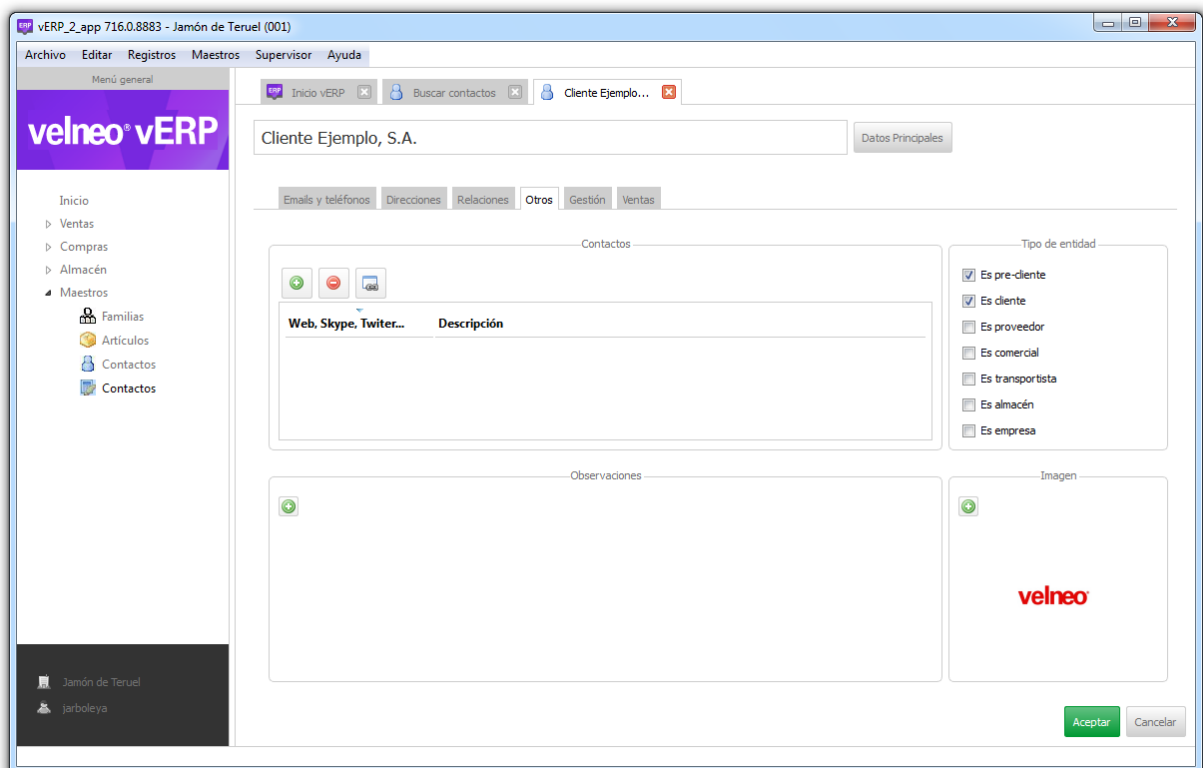
## ERP con trazabilidad completa

A la vez que hemos mejorado la usabilidad de los ciclos de compras y ventas hemos incluido en los documentos, tanto en cabecera como en las líneas, las pestañas de trazabilidad que permiten al usuario navegar entre los documentos relacionados de forma sencilla y directa.



## Los tipos de entidad más fáciles de definir en tu ERP

Otra importante mejora para los suscriptores de Velneo vERP ha sido la simplificación de la configuración de los contactos. A la hora de definir el tipo de entidad se ha eliminado la antigua estructura de tablas y en su lugar se han creado campos booleanos que permiten visualmente marcar con un clic los tipos de entidad de cada contacto.



Además de que es más sencillo para el usuario final definir el tipo de entidad, el programador cuenta con índices específicos que le facilitan en cada formulario la selección de contactos de un tipo o tipos de entidad deseados.

## Una versión de ERP cargada de mejoras

Adicionalmente a todas las novedades comentadas hasta ahora, esta versión llega cargada de otras muchas mejoras:

### **Nuevo diseño del menú general**

El formulario incluye en la cabecera el espacio para el logotipo de la aplicación, desaparece el botón del asistente, en el centro se muestra el nuevo menú general de opciones dinámicas y en el pie se ha incluido la información de la empresa en curso y el usuario.

### **Búsquedas de menús optimizadas**

Hemos optimizado el rendimiento de las búsquedas de documentos de compras y ventas en sus respectivos menús, ejecutando dicha búsqueda en el servidor para conseguir un gran rendimiento en cloud independientemente del número de registros que tenga la base de datos. Estas búsquedas ya tienen en cuenta la empresa en curso a la hora de visualizar sólo los registros adecuados.

### **Configuración de la serie de compra y venta en las entidades**

Se han añadido los campos serie de factura compra y serie de factura de venta en la tabla de Contactos. Esto permite que cada entidad (Cliente/Proveedor) tenga configurada una serie de compra o venta por defecto.

Además, en la configuración de la empresa también existe una configuración de serie de compra y venta por defecto. Esta configuración es general para todos los documentos de compras y ventas respectivamente.

Ahora la aplicación en la serie de cualquier documento de compra, asume por defecto la serie del proveedor si la tiene configurada y en su defecto asume la serie de compra configurada para la empresa. Para los documentos de venta es similar, asume por defecto la serie del cliente si la tiene configurada y en su defecto asume la serie de venta configurada para la empresa. En cualquier caso, el usuario una vez grabado el proveedor o el cliente podrá modificar la serie del documento.

### **Configuración de regímenes de IVA por contacto y serie**

Se podrá configurar la facturación de exención de IVA en la configuración del cliente o proveedor o bien por serie, en dicho caso la factura tendrá todos los artículos exentos de IVA. En caso contrario, cada línea de factura asumirá el tipo de IVA que corresponde a la configuración de Artículo.

**Contadores por serie para todos los documentos de compras y ventas**

Todos los documentos de compras y ventas se numeran usando un contador basado en la serie. Este contador puede configurarse para que se inicie a 1 en cada año o que sea correlativo a lo largo de los años.

**Formato del número de documento automático o por fórmula dinámica**

En la configuración de parámetros por empresa se han añadido unas fórmulas dinámicas para cada documentos de compra y venta donde podemos configurar el formato del número de documento. En caso de no configurarlo vERP utiliza un formato automático basado en la estructura "Serie /Año-Número".

**Opción para duplicar pedidos y presupuestos mejorada**

Cuando se realiza la duplicación de un pedido de compra, de un presupuesto o de un pedido de venta, se pide el formulario de cabecera del registro duplicado como confirmación de la duplicación. Además se podrán cambiar en ese momento los datos de cabecera del documento nuevo.

**Ordenación de líneas en documentos de compra**

Se ha añadido la opción de ordenar las líneas de documentos de compra (albarán y factura) arrastrando y soltando la línea a mover se posicionará en el último lugar, para así mejorar la presentación y punteo de documentos.

**Se eliminan las inserciones de formulario y se sustituyen por la personalización**

Ya no necesitarás los puntos de inserción de formulario para añadir tus propias pestañas de tu solución dentro de vERP, sino que podrás hacerlo siempre por configuración sin necesidad de que en la solución vERP esté previsto el punto de inserción de formulario. Esto permite además que insertes la nueva pestaña en la posición que desees.

**Unificación de datos de configuración de contactos en una tabla (ENT\_M)**

Se unifican las tablas de configuración del contacto ENT\_CLI (tarifa del cliente) y ENT\_ERP (datos de configuración de gestión) en una sola.

**Mejoras al desactivar controles de campos según estado**

Cuando un documento está pasado al siguiente estado del ciclo de compras o ventas, el originario se bloquea para mantener coherencia de los datos. Antes se hacía por CSS, ahora se ha cambiado para pasar a realizarlo por JavaScript, para evitar el efecto en cascada al aplicar CSS.

**Mejora en formularios de configuración de vERP**

Se han eliminado del formulario de parámetros de vERP aquellos controles que no existen en las tablas de configuración. Además se mejora maquetación y ordenación de los campos booleanos de la pestaña “Generales” y se ha modificado el título de la pestaña del formulario de parámetros básicos que antes salía en blanco. (461 y 463)

#### **Mejora en menú de búsqueda de contactos**

Se han añadido funcionalidades en el menú de búsqueda de contactos: salir con tecla aceleradora ESC y botón de formulario para la modificación del contacto. (464)

#### **Revisados textos estáticos de títulos de menú**

Corregido el texto estático de título del menú de turnos que hasta ahora mostraba terminales. (462)

#### **Mejora de usabilidad en menús de configuración**

Las opciones de menús de formas de pago, series, terminales, turnos y dependientes sólo abren la pestaña una vez si ya está abierta se posiciona en ella como hacen el resto de opciones del menú. Además, se renombran las acciones normalizando los nombres de acuerdo al estándar de los demás objetos del proyecto. (459)

#### **Mejora de usabilidad en barra de herramientas de movimientos de almacén**

En la toolbar de las rejillas de movimientos de almacén se quitan los botones de baja y modificación, puesto que no son funcionales. Cuando se hace doble click sobre un movimiento de almacén, en pantalla se muestra el formulario del documento que lo originó. (465)

#### **Mejoras de acceso en forma de consulta a documentos**

En las facturas contabilizadas, albaranes facturados y pedidos servidos/recibidos los controles de los formularios de cabecera y líneas saldrán desactivados para que no sean modificables. (460)

#### **Mejoras en formularios de líneas de factura**

En el formulario de líneas de factura de venta aparecía el texto de coste cuando no debería. (466)

#### **Mejoras en seguridad de acceso a toda la aplicación**

Se amplía el diccionario de permisos para acceso a todos los menús y mantenimientos de la aplicación, pudiendo limitar la consulta, alta y modificación de cualquier tipo de registro. Podrá importar el fichero plano del diccionario de permisos actualizado que le proporcionamos para aplicar esta configuración. (414)



**Mejoras en gestión de datos de los contactos**

Se tiene en cuenta cuando la dirección sólo tiene asignado un teléfono al copiar teléfono y dirección. (467)

**Se revisan y unifican los contenidos iniciales de los campos de totales de impuestos en los documentos de compra y venta**

Se revisan y unifican los contenidos iniciales de los campos de total recargo y total IVA, quitando el redondeo en aquellos que lo tienen, ya que son sumas de campos redondeados previamente al ser calculados. (458)

**Correcciones en enlace a Almacén, en parámetros y en facturas simplificadas**

Se muestra en edición de formulario de parámetros el almacén para TPV. Se realiza corrección en el contenido inicial de facturas simplificadas para que lo capture del parámetro indicado. (468 y 469)

**Mejora en la pestaña de impuestos de documentos**

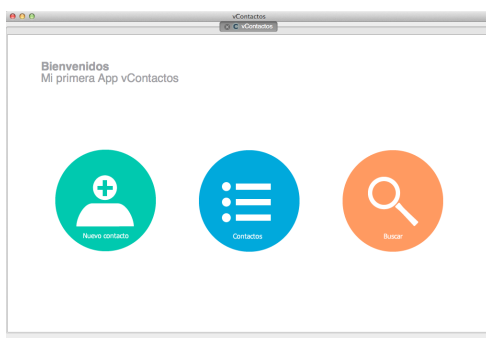
Se optimiza la carga de variables para que se realice al mostrar esta pestaña y no al inicio del formulario, de modo que captura la información en el manejador de evento onShow del formulario, leyéndola del contacto indicado en el documento de compra o venta. (473)

## Seminarios On-Line



Para Velneo siempre ha sido muy importante el facilitar al máximo las posibilidades de formación y la posibilidad de disponer de un canal de comunicación tanto para nuestros clientes, como para los que todavía no lo sois.

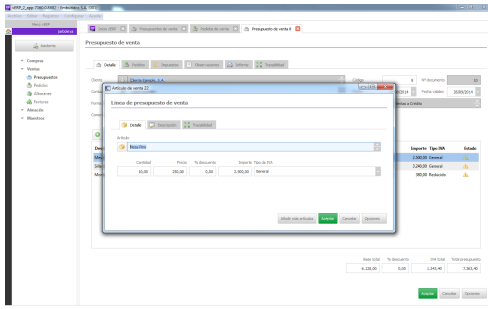
Escuchando las necesidades de la comunidad de desarrolladores, empezamos un proyecto muy ilusionante para nosotros como es el de ofrecer seminarios online mensuales para mostrar de una forma ágil qué es Velneo y las posibilidades de ganar más dinero con tu ERP.



Para ello, hemos creado e impartido un seminario denominado **“Tu primera aplicación empresarial desde cero”** donde mostramos cómo en menos de una hora es posible crear una aplicación de Contactos.

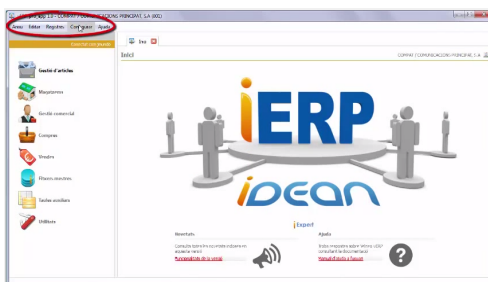
Se explica de forma práctica cómo crear tu servidor de desarrollo y a utilizar los objetos más elementales de la plataforma.

Hemos añadido también el ejemplo de la misma aplicación para sistemas de movilidad en Android como muestra de la potencia de la plataforma en los multidispositivos.



Otro de los seminarios que hemos impartido regularmente ha sido el denominado **“Gana más dinero con tu propio ERP”** donde se muestran las posibilidades de desarrollo y parametrización de aplicaciones empresariales usando como punto de partida el vERP de Velneo.

Orientado tanto a desarrolladores noveles, implantadores y por supuesto a los más avanzados ya que es donde aplicamos muchas de las nuevas técnicas de desarrollo y nuevas funcionalidades de la plataforma, como por ejemplo las muchas novedades aparecidas en los informes de la mano de vReport. Todo para que sea lo más Life is soft posible.



Otro punto de mejora muy importante que hemos introducido en los seminarios, es la inclusión de casos de éxito de nuestros clientes lo que ayuda a reforzar aún más la percepción de sinceridad y rentabilidad ya que son ellos mismos quienes explican sus necesidades iniciales, cómo han conseguido llegar a cumplir su objetivo de la mano de Velneo y las posibilidades que ofrecemos.



Para mejorar la información de la comunidad de desarrolladores, tras cada seminario realizado publicamos un post en el blog de Velneo y se notifica en las redes sociales para conseguir una mayor difusión.

Esto ha permitido avanzar en la mejora de las comunicaciones tanto a los clientes como a los muchos seguidores de Velneo en las diferentes redes sociales.



**2.366** registrados



**82** asistentes de media

Sin duda estamos muy contentos de la aceptación del proyecto de ofrecer formación online de forma continuada a nuestra comunidad y esto se refleja en los **2.366 registrados** en el periodo que va desde mayo a agosto con una media de **82 asistentes** por seminario.

Esto nos ayuda a seguir mejorando día a día en el proceso de formación y comunicación con vosotros y nos comprometemos a seguir avanzando en esta línea y prueba de ello es la mejora del nuevo vídeo de registro y descarga de la plataforma y los nuevos seminarios “Velneo: de 6x a V7” que empezamos a impartir el pasado 1 de octubre.

Gracias a todos por seguirnos en estos seminarios.  
Life is soft!

## Life is Soft 2015

### ¿Qué es?

Es un evento organizado por Velneo para hablar de los que nos gusta, de lo que sabemos, de **software**. Dirigido a desarrolladores y sobre todo a la **comunidad** de Velneo, es **nuestro evento** anual. En él tendrás la oportunidad de ponerte al día en temas técnicos y de negocio, intercambiar y aprender de diferentes experiencias de desarrolladores, y conocer distintos proyectos de software. Este año será +Life is Soft que nunca.

### ¿Cuándo?

Se celebrará el **12 de marzo** de 2015.

### ¿Cuanto cuesta?

No queremos que el tema económico sea un impedimento para que puedas asistir, con lo que el evento es **gratuito**. El aforo es limitado con lo que debes [registrarte](#) si quieres asegurar tu plaza.

### ¿Dónde?

En la [Escuela Universitaria de Informática de la Universidad Politécnica de Madrid](#). Mantenemos la misma ubicación que la edición 2014, ya que Madrid ofrece buena comunicación con el resto de ciudades, si bien en esta ocasión, atendiendo a vuestras sugerencias de mejora, utilizaremos una sala con mayor capacidad para dar respuesta a más solicitudes de asistencia.

### ¿Formato?

La duración es de un día completo. Habrá una jornada de mañana de **networking** y por la tarde **ponencias** con contenido técnico y de negocio de software, así como una mesa redonda de casos de éxito. Puedes consultar la [agenda](#) del evento aquí.

### ¿Cómo puedes asistir?

Puedes hacerlo de manera **presencial**, y cómo en la edición anterior pensando en la comunidad latinoamericana y para todos aquellos que no se puedan desplazar retransmitiremos en

**streaming** las ponencias de la jornada de tarde.

Para asistir al evento debes de [registrarte aquí](#)

### ¿Por qué asistir?

Date un motivo para compartir tiempo, dudas, ideas y ven a conocernos. Tendrás oportunidad de **conocer** de primera mano a otros **desarrolladores** de la comunidad ,con los que compartir experiencias, y al **equipo de Velneo** al que podrás plantear en persona tus inquietudes.

*“Cercanía del equipo Velneo. Es un evento de y para la **comunidad**”.*

*“La **inspiración** que se obtiene para seguir creciendo como **desarrollador de aplicaciones empresariales**”.*

*“El **networking** muy buena opción para poder comunicar con el **personal de Velneo** de forma directa y **cara a cara**, poder intercambiar información y exponer los problemas que podamos tener con el desarrollo. **Muy recomendable**”.*

*“Me ha gustado la elección de **temas** para las intervenciones. Un **amplio abanico** podría decirse y **todo muy interesante**”.*

*“El **networking** muy buena opción para poder comunicar con el personal de Velneo de **forma directa y cara a cara**, poder intercambiar información y exponer los problemas que podamos tener con el desarrollo. **Muy recomendable**”.*

*“Sinceramente me siento orgulloso de pertenecer a la comunidad y de estar de alguna manera **involucrado** en esta **evolución de Velneo**”.*

*“Por mi perfil de **no-desarrollador** me ha parecido muy interesante conocer de primera mano las impresiones y la información dirigida a la comunidad de desarrolladores”*

*“Sinceramente creo que ha sido muy interesante todo y un acierto dar cabida a tantos contenidos sin extenderse demasiado en cada uno de ellos. **Perfecto el evento**”.*

Comentarios de asistentes de la edición anterior.

**Velneo V7 +Life is Soft que nunca**

**¿A qué esperas?**