



Velneo V7 7.15 Velocidad y Estabilidad

Índice

[Velneo V7 Velocidad y Estabilidad.](#)

[Desarrollo](#)

[Aplicaciones Rápidas.](#)

[Rejilla de base de datos.](#)

[Edición de datos: Data Catcher Edit](#)

[Localizador de registros](#)

[Usabilidad en la plataforma desarrollo](#)

[Administración del Servidor de Base de datos](#)

[Estabilidad en aplicaciones en ejecución](#)

[Estabilidad en la programación de aplicaciones](#)

[API de Programación](#)

[Base de datos y servidor](#)

[Generador de informes](#)

[Otras novedades](#)

[Open Apps](#)

[Gestión de contactos de manera sencilla.](#)

[Cómo personalizar aplicaciones en ejecución](#)

[Software Erp de código abierto pymes, versión 1.4](#)

[Programa TPV en código abierto](#)

[Cloud](#)

[Explorador Archivos Cloud](#)

[Panel de control cloud](#)

[Cloud Latinoamérica](#)

[TCP en Cloud](#)

[Utilidades](#)

[Exportación a Excel](#)

[Exportación a CSV](#)

[CSS en aplicaciones de escritorio.](#)

[Instalar entorno desarrollo rápidamente](#)

[Abandono del soporte NPAPI por Google](#)

[Formación y Web](#)

[Formación en Programación](#)

[Página web sencilla y rápida.](#)

[Evento de Software. Life is Soft 2015.](#)

[¿Cómo empezar a programar?](#)

[Aprender a programar rápido](#)

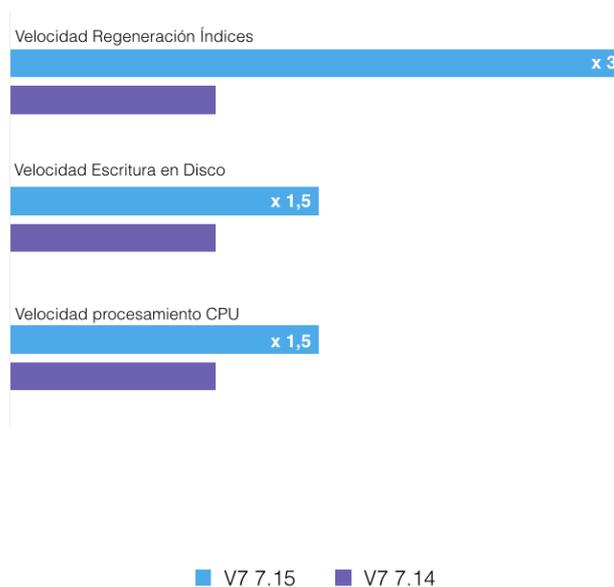
[Velneo V7 Velocidad y Estabilidad](#)

Velneo V7 Velocidad y Estabilidad.

Velneo V7 7.15, hasta tres veces más rápido.

Hemos trabajado intensamente en todas las áreas de la empresa mejorando nuestros productos y servicios, para que cada día Velneo sea más veloz y estable para desarrollar aplicaciones empresariales. Nuestro objetivo es conseguir que los programadores disfruten desarrollando aplicaciones empresariales de manera rápida y fácil. Queremos dar las gracias a la comunidad por la participación y la ayuda que nos presta cada día, esperamos que disfrutéis de esta nueva iteración.

Comparativa V7 7.15 vs V7 7.14



Velocidad en aplicaciones: La velocidad es algo fundamental para las aplicaciones, en la nueva versión 7.15 hemos incluido una serie de mejoras que implican una mejor integración con los sistemas operativos Microsoft Windows. Esta mejor integración se traduce en una mayor velocidad y rendimiento de todos los componentes de Velneo V7 en Windows y por tanto de nuestras aplicaciones. Se ha mejorado la **velocidad en aspectos como el alta y baja de registros, escritura de ficheros, regeneración de índices, regeneración de área de datos, entre otros.**

En procesos críticos, como son la regeneración de índices u otras operaciones en tabla, **se han obtenido mejoras que triplican la velocidad de versiones anteriores.** Las aplicaciones serán mucho más rápidas, y la mejora será más notable cuando se trabaje con tablas con gran número de registros y muchos índices.



Estabilidad: Uno de nuestros principales objetivos con la plataforma es mantener la estabilidad. Cada día se ponen en producción nuevas instalaciones desarrolladas en Velneo V7, por ello debemos poner foco en mantener y mejorar la estabilidad lograda hasta el momento. Para ello hemos trabajado en +100 incidencias que mejoran la estabilidad y usabilidad de las aplicaciones.



Personalización de aplicaciones: Gracias a la potencia y mejoras del nuevo API hemos conseguido que se pueda personalizar la interfaz de las aplicaciones en tiempo real. Esta funcionalidad aumentará nuestra velocidad de implementación en los clientes. Sin tocar nuestro código podremos introducir personalizaciones sorprendentes a nuestros clientes para cubrir sus necesidades. No te pierdas el vídeo de [Jesús Arboleya mostrando un caso en tiempo real](#).

Más y más novedades

Encontrarás muchas más novedades en Velneo V7 7.15, entre otras:

- Objeto rejilla renovado
- Nuevo control Data Catcher Edit
- Uso TCP en el cloud
- Localizador sobre índices complejos
- Login por Google en velneo.es
- Explorador de archivos Cloud
- Importantes novedades en el API de Velneo para Javascript
- Instalación de Open Apps desde vDevelop
- y muchas más novedades...

Puedes leer el listado completo y detallado de novedades en el siguiente archivo PDF

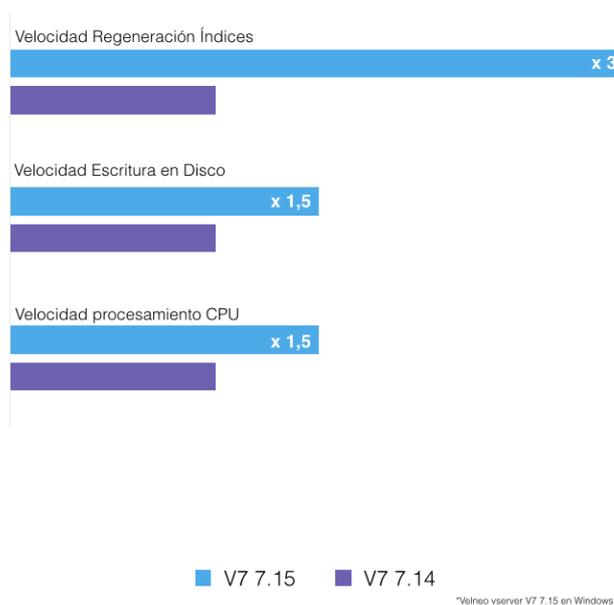
Durante las próximas semanas iremos publicando posts en el blog de velneo.es explicando en profundidad cada una de las novedades introducidas en esta versión Velneo V7 7.15. Mediante estos artículos os podremos mostrar la dimensión de cada una de las novedades.

En la página de descargas están disponibles las descargas de los distintos componentes de Velneo V7 para los diferentes sistemas operativos. En esta versión 7.15, la descarga para el componente Velneo vwebclient V7 se podrá realizar mediante una consulta a soporte donde se nos ofrecerá el enlace a la descarga del componente como se explica en el fichero completo de novedades.

Desarrollo

Aplicaciones Rápidas.

Comparativa V7 7.15 vs V7 7.14



En la nueva versión 7.15 hemos incluido una serie de mejoras que implican una mejor integración con los Sistemas operativos Microsoft Windows. Esta mejor integración se traduce en un mayor rendimiento de todos los componentes de Velneo V7 y por tanto de nuestras aplicaciones.

En procesos críticos, como son la regeneración de índices u otras operaciones en tabla, se han obtenido mejoras de velocidad en torno al 60%, lo que quiere decir que Velneo V7 7.15 es tres veces más rápido que la versión anterior en algunas áreas, mejorando con mucho nuestras mayores expectativas al respecto.

Esta novedad afecta a diferentes tareas que realizamos en nuestras aplicaciones a diario, la manipulación de cadenas, operaciones de base de datos como el alta o baja de registros, procesamiento de CPU, escritura de ficheros, regeneración de datos e índices, lo que se refleja en que nuestras aplicaciones serán aún más rápidas. Todo ello redundará en una mejor experiencia de usuario, sobre todo cuando se trate de operaciones con mucho volumen de datos.

No solo el cliente y el servidor se verán beneficiados por esta optimización, sino que todos los componentes de Velneo V7 se beneficiarán. Y en el que más lo notaremos será en Velneo vServer V7, y en concreto en operaciones de la base de datos. Esta es una de las áreas que mejor responde a las optimizaciones de la versión. Una tarea tan crítica como es la regeneración de índices supera la mejoría del 60% en muchos casos, reduciendo en tres veces los tiempos de espera.

Con mayor volumen de información, la mejoría de rendimiento es más notable y los tiempos se reducen proporcionalmente. Por lo tanto, nuestras aplicaciones serán mucho más rápidas, todavía más notable cuando manejan tablas con gran número de registros y muchos índices.

Rejilla de base de datos

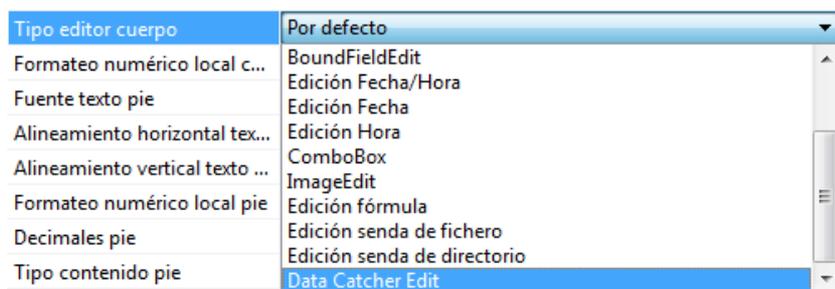
Importantes novedades en rejilla y rejilla editable

El objeto más usado en nuestras aplicaciones para mostrar listas de datos es, sin duda, la rejilla. Por este motivo cualquier mejora que se produce en este objeto cobra vital importancia para nuestros desarrollos.

En la versión 7.15 la rejilla ha recibido una profunda revisión donde la mayoría de las mejoras no tendrán una repercusión visual en la interfaz, pero sí las notarán los usuarios de nuestras aplicaciones cuando la usen o los desarrolladores cuando programen con el objeto. Las mejoras afectan tanto a la rejilla editable como en modo visualización. Vamos a repasar todos esos pequeños grandes detalles que nos ayudan en nuestro día a día:

Edita el contenido de las columnas con el editor que prefieras

Hasta ahora, al definir una columna editable no podíamos asignar cómo se representa visualmente su editor, asumiendo un valor por defecto. A partir de esta versión podemos seguir dejando el editor por defecto o declarar en la propiedad "tipo editor cuerpo" cuál queremos que utilice el usuario, como vemos en la imagen.



Gracias a este nueva propiedad, nuestras rejillas editables ganan en usabilidad, funcionalidad, velocidad y estabilidad.

Data catcher edit en rejillas

En esta versión ha aparecido un nuevo y potente control de edición llamado data catcher edit. Hemos incluido este control dentro de los que puedes seleccionar para usar en la edición de cualquier columna, añadiendo más funcionalidad en la edición de datos en rejilla.

| Artículo | Artículo en idioma cliente | Artículo en idioma del usuario |
|----------------------|----------------------------|--------------------------------|
| Coca Cola | Coke | Cocacolen |
| Nombre | gther | BicFuegofen |
| Cigarrillos Marlboro | oro Cigarettes | Cigarrofen Marlboro |
| Coca Cola | oro Cigarettes | Cigarrofen Marlboro |
| Encendedor Bic | | Cocacolen |
| | oro Cigarettes | Cigarrofen Marlboro |

Define el alto de las cabeceras de rejilla

Si en la propiedad "alto de cabecera" dejas el valor 0, se asume que la configuramos en modo automático y se creará con un alto de cabecera estándar preestablecido para el control. Si lo deseas, puedes aplicar el valor que te interese para hacerla más grande o pequeña, con el objetivo de ajustarla a tus intereses de interfaz, o incluso hacerla multilínea.

| Nº línea | Foto | Nombre Artículo | Cantidad Servida | Precio | Importe | Servido |
|----------|------|--------------------|------------------|-----------|-----------|---------|
| 1 | | Audi A8 | 1,00 | 46.320,00 | 46.320,00 | No |
| 2 | | Mercedes C 220 CDI | 2,00 | 41.143,00 | 82.286,00 | No |
| 3 | | BMW Serie 1 | 3,00 | 24.810,00 | 74.430,00 | No |

En la migración de versión la propiedad se pone a 0, es decir, en modo alto automático.

Las rejillas ahora también tienen propiedad y señal “timer”

Al igual que ocurre con otros objetos, ahora ya puedes declarar la propiedad “timer” en la rejilla y gestionar la señal para ejecutar manejadores de evento. Esto te permite programar automatismos directamente en la rejilla, tanto si está incrustada en un formulario con si está en modo vista.

| | |
|-----------------------|-------------------------------------|
| Color de fondo de pie | <input type="checkbox"/> Base |
| Toolbar | <input checked="" type="checkbox"/> |
| Longitud de columnas | 720 |
| Timer (Milisegundos) | 5000 |

Ya puedes usar el comando “interfaz: establecer foco” en columnas

Puedes usar el comando de instrucción “interfaz: establecer foco” con las columnas de la rejilla, lo que te permite gestionar saltos del foco entre columnas en función de las condiciones de edición que desees.

Nuevas señales de edición pre-aceptada y edición pre-cancelada

Es en estas nuevas señales donde puedes utilizar el comando de instrucción “Set retorno proceso = NO” para evitar que se acepte o cancele la edición. También es en estas señales donde puedes utilizar el comando de instrucción “interfaz: obtener la ficha en edición de la rejilla” ya que en este punto la edición sigue activa. La señales edición aceptada y edición cancelada se ejecutan con posterioridad a que se acepte o cancele la edición.

Las columnas de la rejilla editable emiten la señal “value changed”

Esta nueva funcionalidad nos permite, de forma sencilla, gestionar cuándo el usuario ha modificado el valor de alguna de las columnas de la rejilla.

Es funcional la variable EVENT_PARAMS para señales de ítem

Hemos incluido la actualización de valores de la variable EVENT_PARAMS para las señales item-activado, item-clicked, item-doble-clicked y itemSelChanged, devolviendo en el primer valor del array la línea de la rejilla en la que se produce el evento.

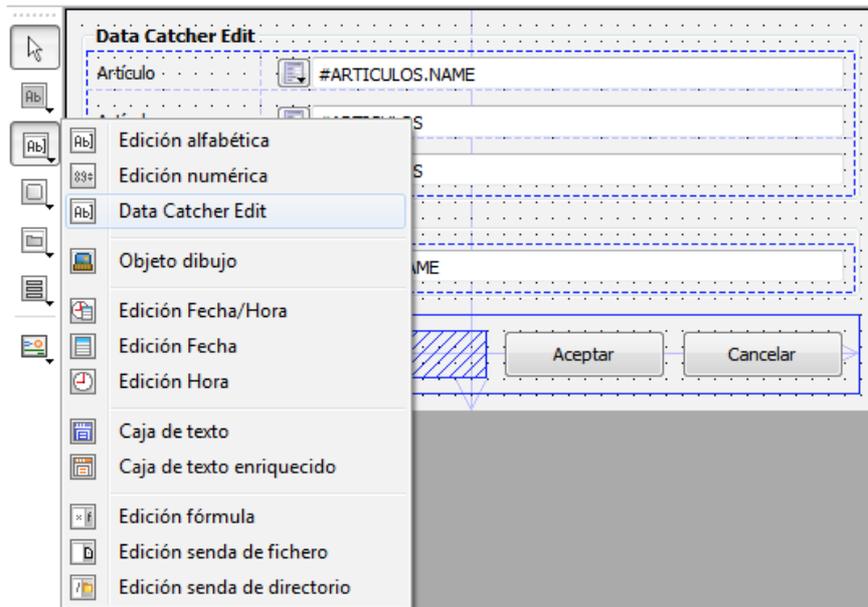
Mejorando lo ya existente

Además de las novedades incluidas en esta versión de la rejilla y rejilla editable hemos realizado un notable esfuerzo en pulir todos esos pequeños o grandes detalles de este objeto para que te ayuden a crear mejores interfaces. A continuación te detallamos algunas de las correcciones y mejoras realizadas:

- Hemos revisado por completo el sistema de señales del objeto y sus subobjetos.
- En edición "si se produce un error en el alta" ya no permanece una línea en blanco, evitando la necesidad de refrescar la rejilla.
- Se han resuelto los problemas que se producían en algunas rejillas editables que combinaban columnas editables con otras no editables, ahora ambos tipos de columnas se llevan bien durante la edición.
- Se han solucionado los problemas relacionados con las altas en rejilla editable.
- Se ha mejorado el uso de los iconos de tabla estática así como su combinación con texto en la misma columna.
- Se ha mejorado el uso de la tecla "Tab" en los controles que combinan ventana de edición y botones.
- Ahora en el editor de rejillas la función ancho de título (Alt+F7) calcula bien el ancho teniendo en cuenta que las cabeceras se ponen en negrita, y también lo hace correctamente con texto pequeños.
- La función setCurrentSelect(Int) de la clase VAbstractListDataView, pasándole un parámetro entero con la posición del registro, ya funciona correctamente.
- En la rejilla editable ahora se dispara correctamente la señal de edición aceptada si estando en edición hacemos clic con el ratón en una columna no editable, además ahora también se dispara correctamente si estando dentro de edición se cambia el foco fuera de la rejilla, por ejemplo, haciendo click en otro control.
- Se ha subsanado la incidencia que producía que no se ejecutase el subproceso del comando "interfaz: obtener la ficha en edición de la rejilla" en un evento disparado con las señales edición aceptada y edición cancelada.
- En vDataClient al visualizar la rejilla de una tabla ya se muestran las columnas correspondientes a punteros singulares o hermanos, aunque estén en las últimas posiciones de la lista de campos.
- Subsana la incidencia que producía que en una rejilla con totales de suma acumulada, con una columna que muestra un campo puntero a maestro, cuando hay registros que contienen códigos inexistentes en el maestro, la rejilla recalculaba constantemente totales. Ahora sólo se calculan los totales una vez.
- El pie de rejilla se ajusta correctamente cuando el pie activo es verdadero y con un modo de cabecera vertical, aunque cambie el ancho de la cabecera vertical.
- Subsana la incidencia que producía que al ordenar o invertir orden en rejillas, la multiselección no se ordenaba, con lo que quedaban seleccionados otros registros de la lista.
- Mejorado el inspector de errores para que no rompa vDevelop al ejecutar el inspector de errores aunque haya una rejilla que tenga asignada una tabla que no existe y tenga declaradas columnas con campos de dicha tabla.
- Aunque en los eventos de pérdida de foco no es recomendable mostrar mensajes, hemos evitado que, si se hace, rompa vClient al hacerlo en una columna de rejilla en edición.
- Hemos conseguido evitar que se dispare la señal de pérdida de foco de la rejilla cada vez que una columna entra en edición.
- Y otras muchas pequeñas correcciones de menor importancia... siempre pensando en la estabilidad y velocidad.

Edición de datos: Data Catcher Edit

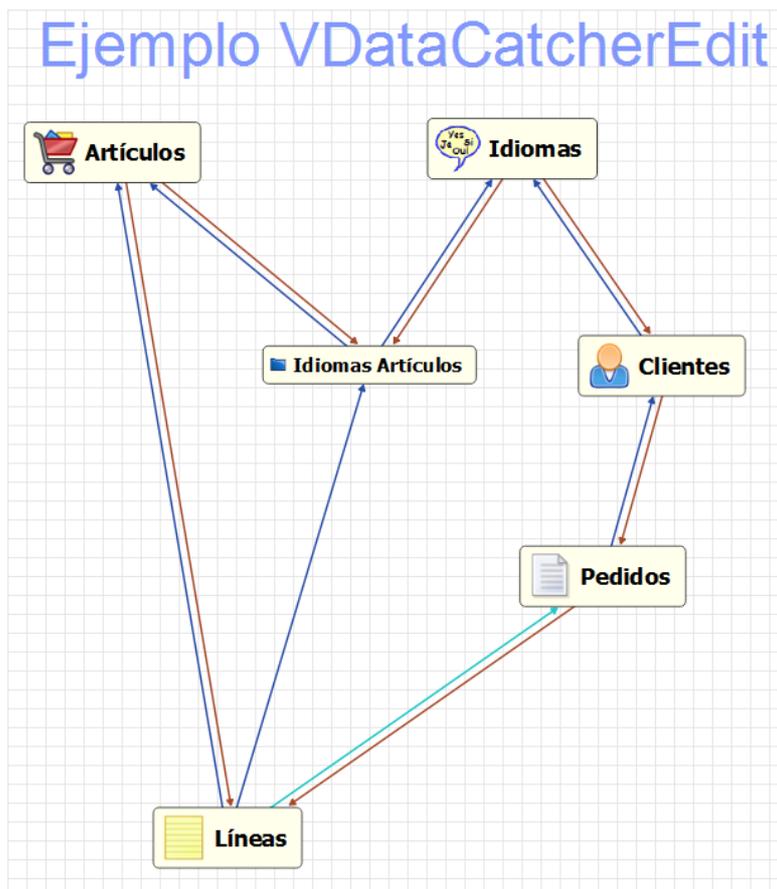
La galería de controles de Velneo V7 ha recibido un nuevo miembro denominado data catcher edit.



Como se puede apreciar en la imagen, el control se puede incluir en un formulario, aunque también podemos declarar columnas de rejilla que editen su contenido con este nuevo control.

Un ejemplo para comprender su funcionalidad

La mejor forma de explicar este nuevo control es viendo un ejemplo de uso. Supongamos que tenemos una aplicación de gestión donde contemplamos una tabla de artículos (ARTICULOS), una tabla de idiomas (IDIOMAS) y una tabla plural de ambas donde declaramos el nombre del artículo en sus diferentes idiomas (ARTIDIOS), veamos el esquema de esta estructura de tablas de la aplicación:



En el esquema podemos observar cómo la tabla de líneas de pedidos apunta tanto a la de artículos como a la de idiomas de los artículos. El problema que se plantea en este tipo de estructuras es ¿Cómo puedo seleccionar el artículo por el nombre, pero seleccionando por una lista de artículos que estén en el nombre del idioma del cliente o el idioma configurado por el usuario en su aplicación?

Sin este control, la localización y selección se vuelve compleja, hay que escribir bastante código para poder resolverlo, además de que también resulta complicado gestionar los diferentes controles a crear en el formulario para la selección del dato en el idioma correspondiente y mostrarlo en el idioma que deseemos. Si queremos hacerlo en una rejilla, aún resulta más complicado.

La potencia del control data catcher edit

Supongamos que pudiésemos tener un control que nos permitiese mostrar la lista de nombres de artículos subindexada para mostrar “sólo” el idioma que nosotros deseemos, pero que al seleccionar el registro de la tabla ARTIDIOS (idiomas de artículos) el control fuese capaz de actualizar directamente el artículo de la línea con el valor correspondiente al artículo seleccionado en el idioma.

Pues esto es precisamente lo que nos permite hacer el control data catcher edit. Podemos, indicarle al control por qué lista de registros de qué tabla nos queremos mover para realizar la selección y también en qué campo queremos dejar el resultado. En concreto, en nuestro ejemplo podremos localizar los artículos por la lista de nombres de artículos por idioma, pero al seleccionarlo realmente estaremos grabando en el campo artículo de la línea el código correspondiente al artículo seleccionado.

¿Cómo se define?

Vamos a repasar lo sencillo que es configurar las propiedades de este control utilizando como referencia la siguiente imagen:

| | |
|---------------------|---|
| Tabla apuntada |  ARTIDIOS@VDataCatcher Dat |
| Campo visible |  NAME |
| Asignar a contenido | #ARTICULOS |
| Contenido inicial | #ARTIDIO |

Sólo tenemos que cumplimentar estas 4 propiedades con los siguientes datos para nuestro ejemplo.

- **Tabla apuntada:** Será la tabla de artículo-idioma (ARTIDIOS) donde tenemos la lista de registros a mostrar para la selección.
- **Campo visible:** Será el campo que se visualice en el control de edición, en este caso el nombre del artículo en el idioma por el que estamos seleccionando.
- **Asignar a contenido:** Indicamos qué campo de la tabla de selección se asignará al campo especificado en la propiedad “contenido” del control, en nuestro ejemplo el campo #ARTICULOS de la tabla ARTIDIOS.
- **Contenido inicial:** Será el campo ARTIDIO, que es un campo puntero indirecto virtual declarado en la tabla de líneas que apunta a la tabla ARTIDIOS, resolviendo por el artículo de la línea y el idioma del cliente en este caso.

En el control de formulario el campo contenido de nuestro ejemplo tiene el valor #ARTICULOS.

Si declaramos en una columna editable de una rejilla que usaremos el tipo de editor data catcher edit, veremos que además de las propiedades comentadas para el control de formulario, tenemos una última propiedad “campo modificable” que es necesaria porque, a diferencia de lo que ocurre en el control de formulario, en la rejilla el dato que se ve no es el mismo que el que se edita, y la propiedad “contenido de la columna” de rejilla es nuestro ejemplo tiene el valor #ARTIDIO.NAME.

| | |
|---------------------|---|
| Tabla apuntada |  ARTIDIOS@VDataCatcher Dat |
| Campo visible |  NAME |
| Asignar a contenido | #ARTICULOS |
| Contenido inicial | #ARTIDIO |
| Campo modificable |  ARTICULOS |

- **Campo modificable:** Indicamos el campo que se modificará con el valor de la propiedad asignar a contenido.

Otro ejemplo

Podríamos con este control mostrar la lista de albaranes pendientes de facturar y, al seleccionar el albarán, atrapar el código del cliente del albarán que es lo que realmente queremos seleccionar.

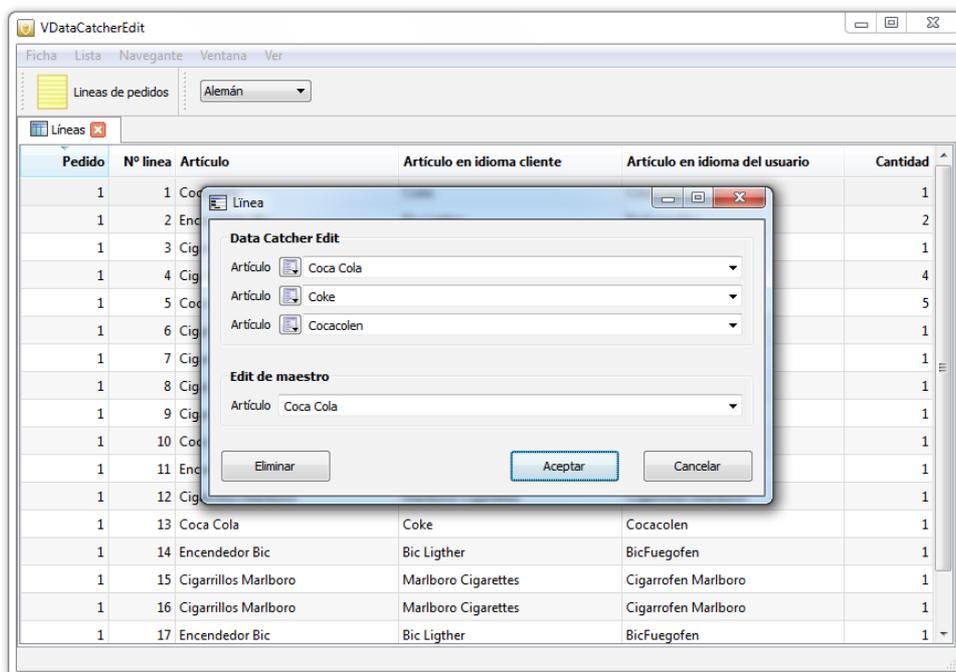
Pero la potencia de este control no termina en estos ejemplos, sino que además su versatilidad nos permite que en un formulario podamos alimentar a este control con una lista de registros que obtenemos en un proceso o script JavaScript y que puede ser el resultado de cualquier búsqueda, cruces de listas, filtrados, etc. Es decir, no sólo podemos utilizar la subindexación sino que además podemos enviarle al control cualquier lista de registros que deseemos, incluso cambiarla dinámicamente en cualquier momento, en tiempo de ejecución.

¿Cómo se comporta a nivel de interfaz?

Su comportamiento es similar al de un control de edición de tabla maestra, nos permite autocompletar el texto, subindexar la lista de registros, mostrar una vista de datos con la lista de registros seleccionables, podemos poner un botón menú para incluir opciones de funcionalidad adicional como localizar, añadir, editar, etc.

En definitiva se trata de un control realmente completo funcionalmente pero que además nos resuelve, de forma elegante y sencilla, una problemática realmente compleja de programar.

A continuación vemos un ejemplo de rejilla editable con columnas que usan este control y el formulario en el que visualmente no somos capaces de distinguir que el primer control es un control de edición de maestro, el segundo es un data catcher edit que nos permite seleccionar el artículo por los nombres de los artículos en el idioma del cliente y el tercero es similar al segundo, pero seleccionando por el idioma configurado en el combobox de la toolbar principal de la aplicación.

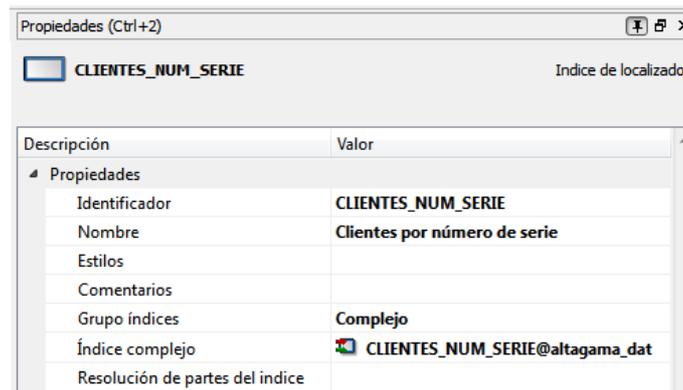


Para completar la información de este control a continuación tienes enlaces al manual de ayuda de Velneo V7 donde te explica una a una las propiedades para que sepas crear controles data catcher edit en tus aplicaciones, tanto en [formularios](#) como en [rejillas](#).

Localizador de registros

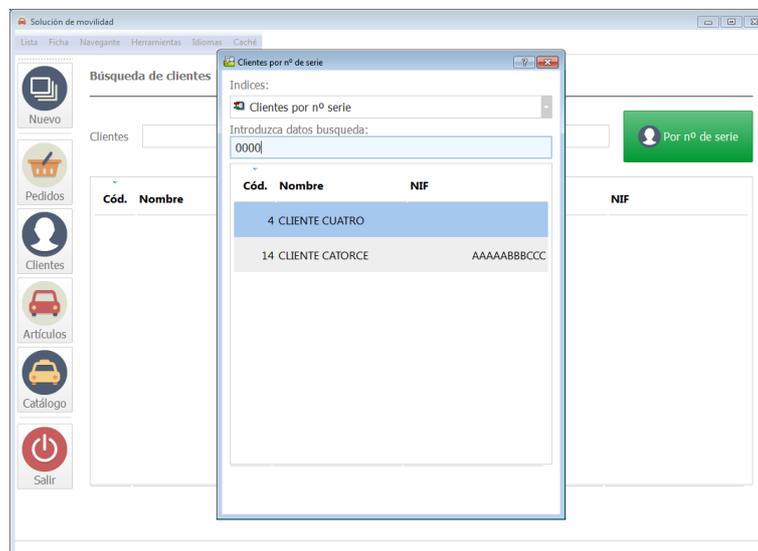
En esta nueva versión de Velneo V7 damos un repaso a este objeto tan utilizado en nuestros desarrollos. Entre las novedades que incluye esta nueva versión encontraremos:

Índices complejos



En la nueva versión podremos hacer uso de índices complejos como componentes del localizador, de tal forma que podremos encontrar un registro de forma directa buscando por otras tablas.

Estos localizadores podremos usarlos, por ejemplo, en una acción para localizar una ficha y mostrar el formulario, o para atrapar un maestro para un campo enlazado a maestro.

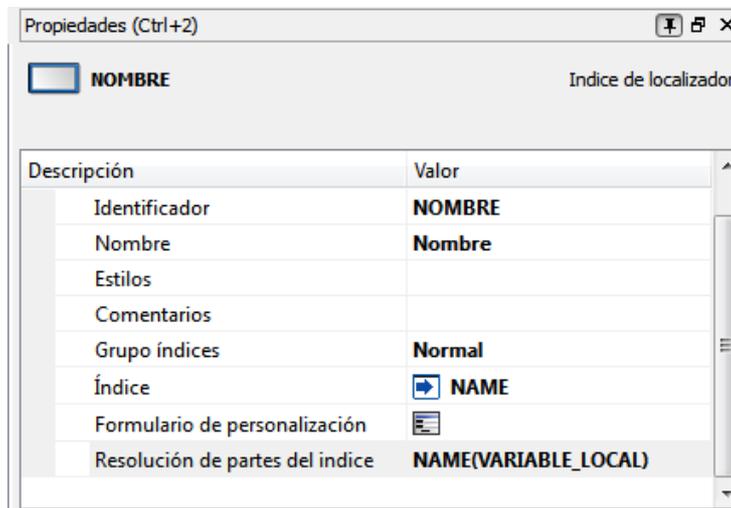


Por ejemplo, localizar pedidos por artículos de sus líneas, localizar clientes por artículos, series, etc., que provengan de líneas de sus pedidos, entre otros muchos ejemplos.

Con esta nueva opción podremos ampliar la funcionalidad de nuestras aplicaciones permitiendo al usuario final acceder a información de forma mucho más rápida y sencilla.

Uso de variables locales

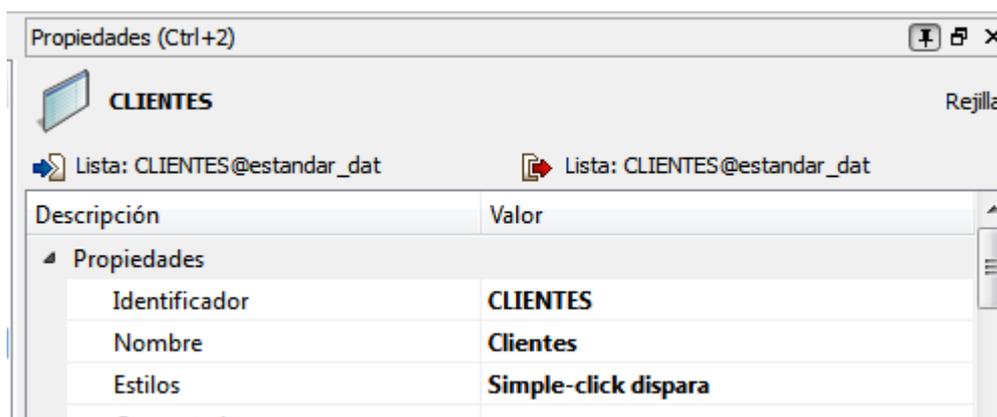
Otra de las novedades que encontraremos es el uso de variables locales en localizadores. Por medio de los comandos de instrucción de Objeto, o las nuevas clases en javascript (VObjectInstance), podremos alimentar variables que podremos usar para resolver los índices usados en el localizador.



Esta nueva funcionalidad nos será muy útil en la [subindexación de localizadores](#). Un ejemplo de uso es resolver la empresa en un localizador multi-empresa por medio de una variable local, evitando el uso de variables globales, definir la subindexación a partir del formulario en el que nos encontramos y un largo etcétera de usos, que nos permitirán añadir velocidad de uso a nuestras aplicaciones..

Simple clic en localizadores

Otra de las novedades incluidas en esta versión es el uso de simple clic en localizadores, especial para dispositivos táctiles. Cuando seleccionamos la rejilla que vamos a usar en el localizador, podemos escoger una en la que hayamos definido el estilo "Simple clic dispara".



De esta forma, con un simple clic el registro se selecciona, consiguiendo mayor velocidad a la hora de usar los localizadores, mejorando la experiencia del usuario.

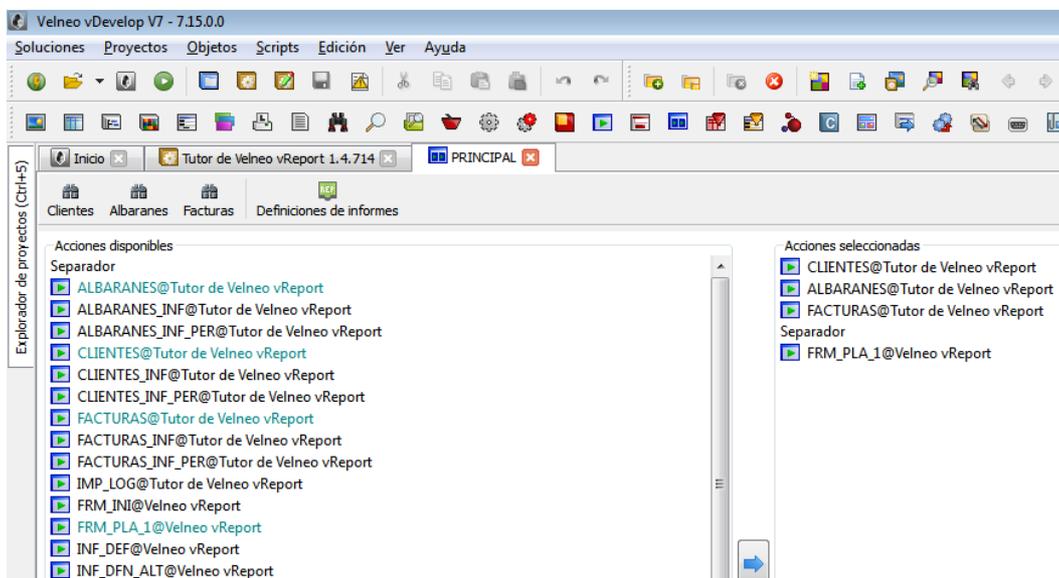
Usabilidad en la plataforma desarrollo

En todas las versiones de Velneo V7 trabajamos no sólo en mayor velocidad y rendimiento de la plataforma, también revisamos las funcionalidades ya existentes para darles más usabilidad y que la herramienta sea más Life is Soft.

En esta nueva versión de Velneo V7 hemos incorporado una serie de mejoras que redundan en la usabilidad de la plataforma, sobre todo cuando desarrollamos, pero también hemos incorporado una serie de novedades orientadas a facilitar la administración e incluso a que las aplicaciones que desarrollemos tengan una mejor experiencia de usuario.

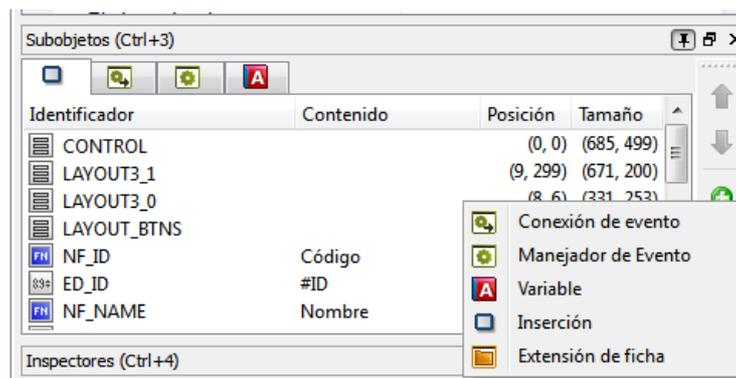
Desarrollo

Por ejemplo, en el área de desarrollo hemos mejorado los editores de menús y toolbars (VBM-0837), las acciones que hayamos incluido en nuestro menú aparecerán marcadas en otro color, de tal forma que nos permite volver a usarla, pero podemos saber que ya ha sido incluida anteriormente en el menú o toolbar que estamos editando. De esta forma se facilita la edición de menús y toolbars, sin limitar las acciones que podemos incluir en ellas.

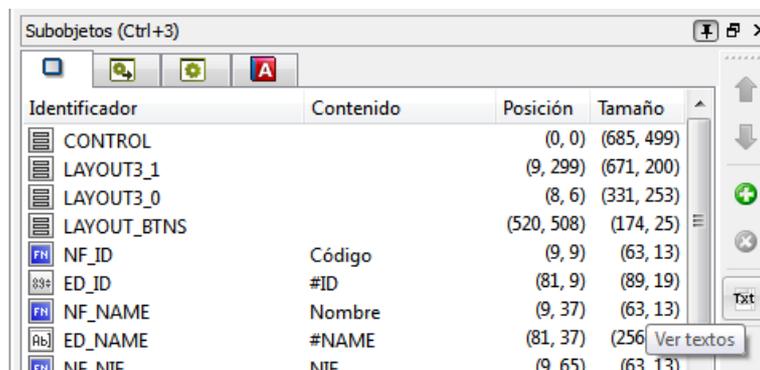


Al mejorar el editor de menús, hemos evitado la incidencia que se presentaban en Mac al realizar esta tarea (VBM-2382) y que hacía perder el menú del editor.

También nos hemos acordado de la ventana de subobjetos, una de las más usadas, y hemos revisado el orden de los subobjetos en todos los objetos (VBM-3273). Así, el orden de los subobjetos seguirán el mismo orden, siendo coherentes y facilitando al desarrollador la creación y mantenimiento de subobjetos.



Además hemos aprovechado a aplicar en esta misma ventana una funcionalidad que ya tenían las pestañas de los inspectores, por ejemplo, se implementa un botón en la toolbar del dock de subobjetos (VBM-3480) que permite configurar si se muestran texto e iconos o sólo iconos en las pestañas, lo que mejora la usabilidad del dock y facilita al desarrollador el acceso a los distintos subobjetos del objeto.



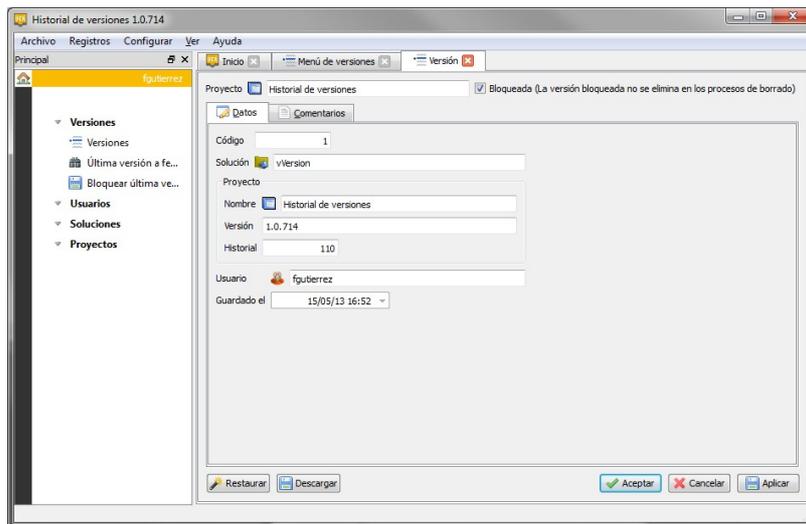
Hemos trabajado también en mejorar temas de conexión, por ejemplo, evitar que nos pongamos a importar componentes sin habernos conectado (VBM-3568) o a crear una nueva solución sin estar conectados al servidor de edición (VBM-3563) y mostrando la ventana de conexión para que comencemos con buen pie.

También hemos aprovechado a matizar algunos asistentes (VBM-3564) cuyo tamaño no se ajustaba bien al contenido que mostraban.

Pero no sólo hemos trabajado en cuanto al diseño o al aspecto en la usabilidad, hemos revisado también funcionalidades como el inspector de errores y el inspector dónde se usa (VBM-3605), que a partir de esta versión nos permitirán encontrar los objetos que se usan para los tipos de dato lista de fórmulas.

De forma sencilla podremos localizar errores que pueda haber en estos tipos de dato o qué objetos se usan, facilitando nuestra tarea de programación como desarrolladores.

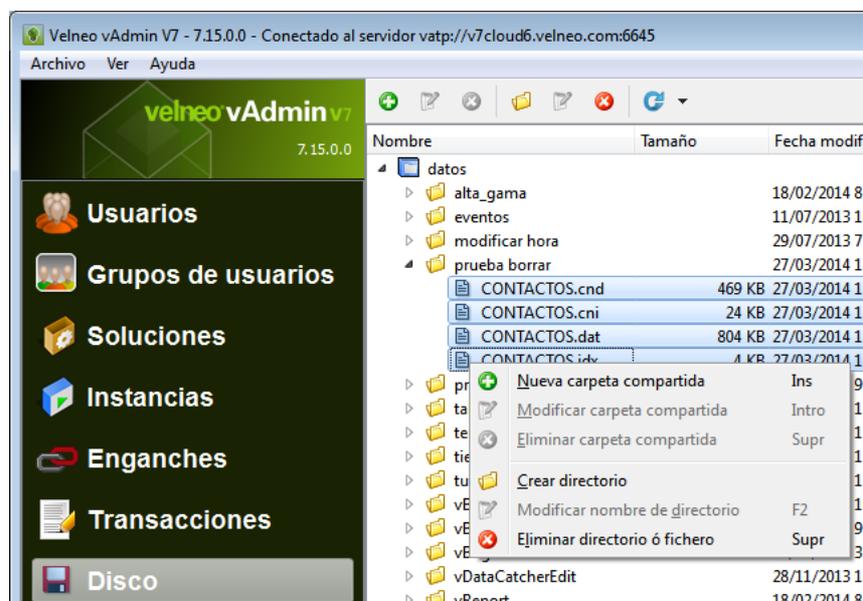
También hemos trabajado en los componentes que ayudan al desarrollo, y por ejemplo en la nueva versión de Velneo vVersion, el historial de versiones hemos mejorado el sistema de recuperación de ficheros (VBM-3719). Cuando recuperemos el proyecto se guardará con un número de historial superior al existente ya en el servidor, con lo que no se produzcan confusiones en cuanto a la versión en curso del proyecto, facilitando al desarrollador su gestión.



Administración

En el área de administración, para mejorar la gestión de instancias y ficheros de datos, hemos mejorado el gestor de disco, y por ello, a partir de esta versión, cuando borramos una instancia se eliminan del registro de ficheros y ya se encuentran disponibles para ser eliminados en disco (VBM-3376).

Así, de forma sencilla podemos deshacernos de información e instancias que hayan quedado obsoletas en nuestro servidor, facilitándonos el trabajo de desarrollo. Con el gestor de disco basta borrar todos los ficheros, y luego, si lo necesitamos, podremos borrar la carpeta correspondiente.

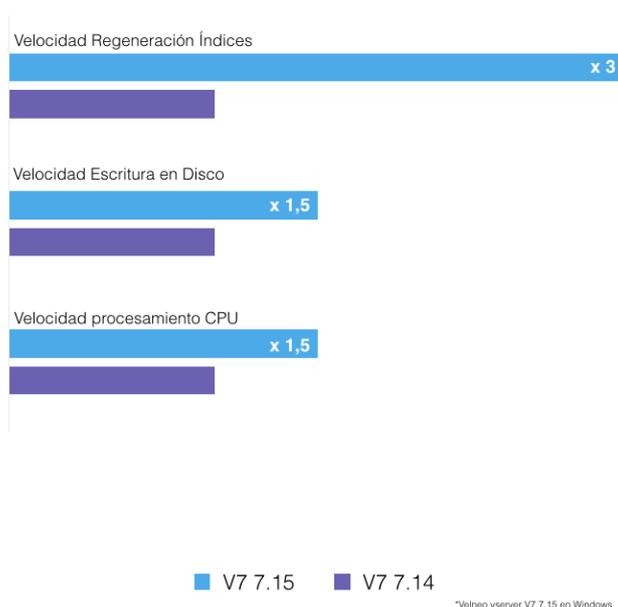


Aplicación de Velneo V7

En el ámbito de la aplicación y la ejecución hemos trabajado en varios frentes, por un lado optimizando la experiencia de usuario, por otro lado creando herramientas que os permitan programar funcionalidades en vuestras aplicaciones que ayuden a la usabilidad de éstas, y mejorando el aspecto visual y la funcionalidad de las aplicaciones creadas con Velneo V7.

Hemos realizado optimizaciones y mejoras en la ejecución de nuestras aplicaciones que redundan en una mejor experiencia de usuario, por ejemplo mejorando la integración de Velneo V7 con el sistema operativo Windows, aumentando la velocidad de ejecución de las aplicaciones en Windows (VBM-3739).

Comparativa V7 7.15 vs V7 7.14

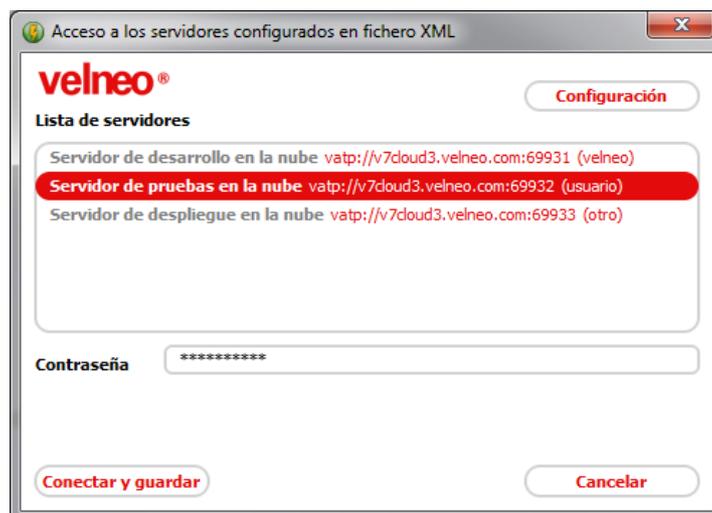


También hemos optimizado puntos clave de nuestras aplicaciones, como los formularios, en los que se ha revisado el sistema de condiciones de visible y activo de los controles (VBM-3614) para que se ejecuten únicamente las veces que sean necesarias y de esta forma se mejora el rendimiento de los formularios usados en nuestras aplicaciones, lo que redundará en una mejor experiencia de usuario.

Entre las funcionalidades que hemos incluido para que podáis mejorar la usabilidad de vuestras aplicaciones está la instrucción de proceso “Interfaz: establecer foco al control” (VBM-3644) que, a partir de esta versión permite seleccionar una columna de rejilla en la que posicionarse, abriendo la edición en el caso de que esté configurada para esa columna.

De esta forma podemos gestionar en manejadores de evento de la rejilla la columna en la que queremos el foco según nuestras necesidades, ampliando la funcionalidad de las rejillas en nuestras aplicaciones.

Nos hemos acordado también de la gestión de idiomas, tema muy importante en nuestras aplicaciones, y por ello se han añadido más opciones para especificar el idioma a las ya existentes (menús, registro, etc.), permitiendo especificarlo como un parámetro o con ayuda de una función en el caso de usamos login QML (VBM-3547). Todo ello con el objetivo de facilitar al desarrollador que pueda hacer más accesible al usuario su aplicación.



Además, hemos incluido ayudas al usuario que ejecuta las aplicaciones, como puede ser la de contemplar la inducción de las tablas de extensión (VBM-3696), así, a partir de la versión 7.15, la inducción se contempla en ambas direcciones, tanto si el plural es de la tabla padre como si es de la tabla de extensión.

Dentro del diseño y el aspecto visual, pero sin abandonar lo funcional, hemos incluido mejoras como que se mantenga la multi-selección (VBM-3695) cuando cambia el orden de los registros, o que los pies de rejilla se ajusten al tamaño de las columnas aunque se añada la barra de scroll (VBM-3693), cuando hay cambios en el contenido de la rejilla, mejorando con ello la experiencia de usuario.

| Nº pedido | Fecha | Hora | Cliente | Total Pedido |
|-----------|----------|-------|--------------------|------------------|
| 22117 | 19/02/14 | 09:26 | CLIENTE CUATRO | 1.424.731,00 |
| 22116 | 19/02/14 | 08:24 | CLIENTE CATORCE | 2.736.326,00 |
| 22112 | 19/02/14 | 08:14 | CLIENTE CUATRO | 221.991,00 |
| 22111 | 19/02/14 | 08:13 | CLIENTE CINCO | 1.031.305,00 |
| 22110 | 19/02/14 | 00:12 | CLIENTE DIECISESIS | 3.001.350,00 |
| 22108 | 18/02/14 | 20:08 | CLIENTE CUATRO | 1.057.235,00 |
| 22107 | 18/02/14 | 19:49 | CLIENTE CUATRO | 203.533,00 |
| 22106 | 18/02/14 | 19:47 | CINCUENTA | 610.599,00 |
| 22105 | 18/02/14 | 10:53 | CLIENTE CATORCE | 1.057.235,00 |
| 22104 | 18/02/14 | 10:29 | CINCUENTA | 42.176,00 |
| 22103 | 18/02/14 | 10:27 | CINCUENTA | 203.533,00 |
| 22101 | 17/02/14 | 20:19 | CLIENTE CINCO | 20.260,00 |
| 22.090 | | | | 4.449.090.163,34 |

Y también nos hemos preocupado de pequeños detalles como la homogeneización de cómo se muestran los booleanos (VBM-3405), (en todos los puntos donde se muestre un booleano por defecto se presentará con los valores sí o no), ayudan a hacer más comprensibles nuestras aplicaciones.

En definitiva, con el objetivo de hacer más Life is soft nuestra plataforma, hemos trabajado en muchos frentes, ayudando al desarrollador y al usuario de la aplicación a que sea más sencillo y rápido trabajar con Velneo V7, disfrutando de la experiencia.

Administración del Servidor de Base de datos

En la nueva versión de Velneo V7 7.15 hemos trabajado en novedades y funcionalidades muy interesantes relacionadas con el uso de Velneo vAdmin V7 y la administración del servidor, gestión de grupos, usuarios e instancias.

Todas ellas nos permitirán disfrutar aún más de la velocidad y estabilidad de Velneo V7 7.15.

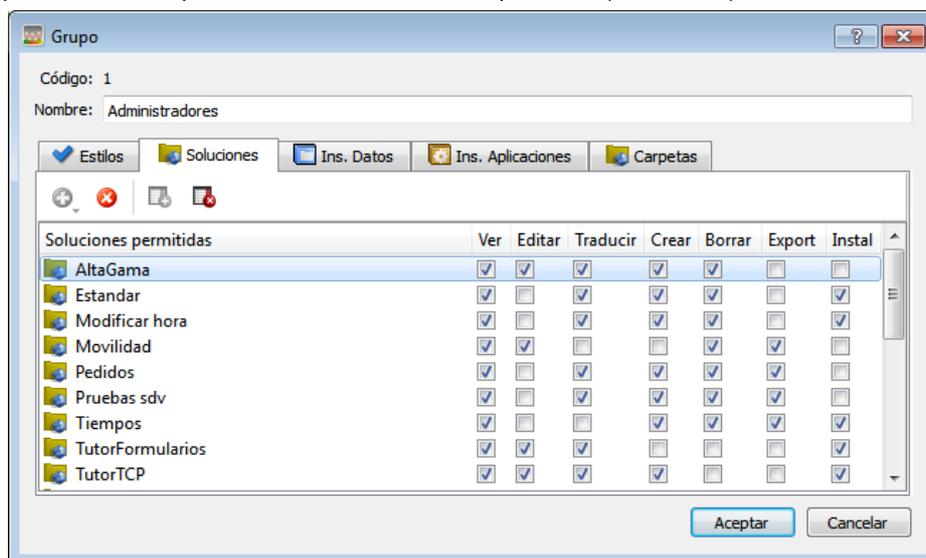
Implementación de permisos para controlar creación de instalables o importación de componentes

Cuando un grupo de usuarios tenía permiso para editar una solución, desde ese momento también podía crear un instalador con Velneo vInstallBuilder V7 o usar la opción de Importar componentes para llevarse el proyecto a otro servidor.

A partir de la versión 7.15, los grupos de usuarios tienen dos nuevas opciones de permisos para las soluciones: Instalar que determina si el usuario puede crear un instalable de la solución en curso. Export que determina si el usuario puede importar esta solución.

De esta forma, podemos configurar de forma sencilla los permisos de acceso y distribución de nuestras soluciones, adjudicando los roles según nuestras necesidades, y garantizando la seguridad de nuestro servidor.

Una vez que actualicemos a la versión 7.15 debemos activar estos permisos en función de la configuración que nos interese, para que los usuarios puedan hacer uso de estas opciones. (VBM-3625)



Eliminación de ficheros de datos correspondientes a una instancia en el gestor de disco de Velneo vAdmin V7

Cuando borrábamos una instancia, los ficheros de datos seguían estando en el registro de ficheros, y cuando queríamos borrar los ficheros y la carpeta en la que estaban, no era posible hasta reiniciar el servidor.

A partir de la versión 7.15, cuando borramos una instancia se eliminan a su vez del registro de ficheros y ya se encuentran disponibles para ser eliminados en disco.

Podemos eliminar los ficheros y la carpeta que los contiene usando la opción de gestión de Disco que tenemos en Velneo vAdmin V7, con lo que podremos limpiar de forma sencilla una instancia que no sea ya necesaria y los ficheros que haya generado. Para ello, debemos borrar primero todos los ficheros, y luego podremos borrar la carpeta correspondiente. (VBM-3376)

Mejora en la asignación de permisos a grupos en el uso de instancias de datos

Cuando en Velneo vAdmin V7 añadíamos a un grupo de usuarios todas las instancias de aplicación y aceptábamos el formulario, al volver a editar el grupo ya no podíamos asignarle instancias de datos.

A partir de la versión 7.15 se soluciona la incidencia y que los botones de añadir instancia o todas las instancias estén activos dependerán únicamente de que estén pendiente que haya instancias de datos que asignar, facilitando la gestión de permisos en Velneo vAdmin V7. (VBM-3720)

El comando de instrucción SDV

Conectar permite acceder al servidor de disco SDV en función de los permisos del grupo de usuarios para acceder a las carpetas compartidas existentes

Cuando usábamos este comando de instrucción para acceder al servidor de disco, requería que el usuario, además de pertenecer a un grupo con permisos, fuera de tipo supervisor.

A partir de la versión 7.15, el comando permite acceder al servidor de disco SDV sin requerir ser supervisor, únicamente comprueba que el grupo de usuarios tiene permisos en alguna de las carpetas compartidas. En el momento de acceder a la carpeta, bien para subir o para descargar ficheros, se comprobará si el usuario tiene permisos para acceder al grupo de usuarios.

Cuando ejecutamos comandos SDV se comprueba si estamos conectados a algún servidor, en caso contrario se ejecuta el comando contra el servidor al que estamos conectados ejecutando la aplicación.

El comando "SDV: Conectar" requiere que el usuario tenga al menos una carpeta compartida en el servidor. Todos los comandos incluidos en el subproceso del comando "SDV: Conectar" se ejecutan contra ese servidor. Todos los comandos de SDV comprueban que el usuario tenga permiso para la carpeta en la que se está ejecutando el comando.

El comando "SDV: Conectar" solo es necesario ejecutarlo si vamos a trabajar contra un servidor diferente al que ya estamos conectados o si queremos ejecutar los comandos con un usuario que tenga permisos diferentes al usuario en curso que está ejecutando la aplicación.

Si queremos ejecutar comandos SDV con el usuario en curso contra el mismo servidor al que estamos conectados ejecutando la aplicación no es necesario ejecutar el comando "SDV: Conectar".

Estabilidad en aplicaciones en ejecución

A continuación se detallan las novedades e incidencias solucionadas relativas a la ejecución de aplicaciones de Velneo V7, tanto con Velneo vClient V7 como con el resto de ejecutores (Velneo vDataClient V7, etc.).

Implementación del control Data catcher edit en rejilla y formulario

Se trata de un control de edición que se usará principalmente para la edición del contenido de campos enlazados a tablas de datos maestras. Permite buscar en una tabla para devolver un dato de otra tabla. Por ejemplo: recorrer un plural para seleccionar un maestro.

Este control tiene la particularidad de que se muestra un contenido y se edita otro. Además, admite subindexación.

Ejemplo: Artículos con plural de referencias. Permite seleccionar atrapar el artículo navegando por la tabla de referencias.

Ejemplo: Mezclando data catcher edit y subindexación podemos trabajar con artículos multi-idioma. Se trataría de tener una tabla de traducciones enlazada a una de artículos y a una de idiomas. De esta forma, con este control podemos atrapar un artículo concreto recorriendo la tabla de sus traducciones a un idioma concreto (VBM-3717).

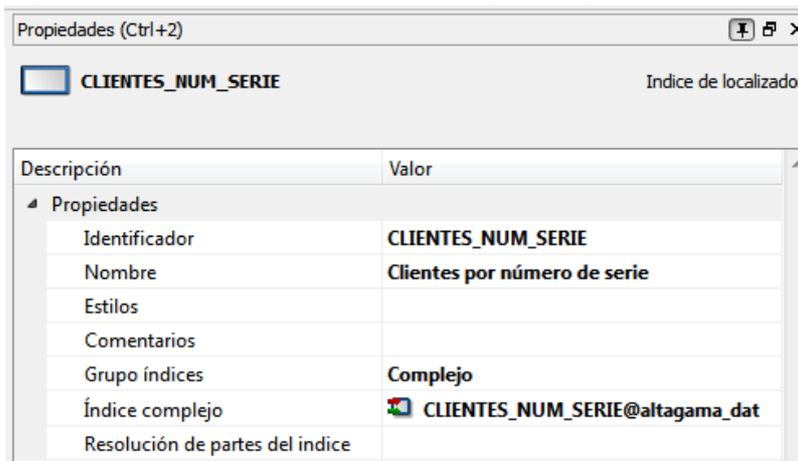
| Artículo | Artículo en idioma cliente | Artículo en idioma del usuario |
|----------------------|----------------------------|--------------------------------|
| Coca Cola | Coke | Cocacolen |
| Nombre | gther | BicFuegofen |
| Cigarrillos Marlboro | oro Cigarettes | Cigarrofen Marlboro |
| Coca Cola | oro Cigarettes | Cigarrofen Marlboro |
| Encendedor Bic | | Cocacolen |
| | oro Cigarettes | Cigarrofen Marlboro |

Implementación de índices complejos y variables locales en localizador

Se ha mejorado el funcionamiento del objeto Localizador, añadiéndole nuevas funcionalidades:

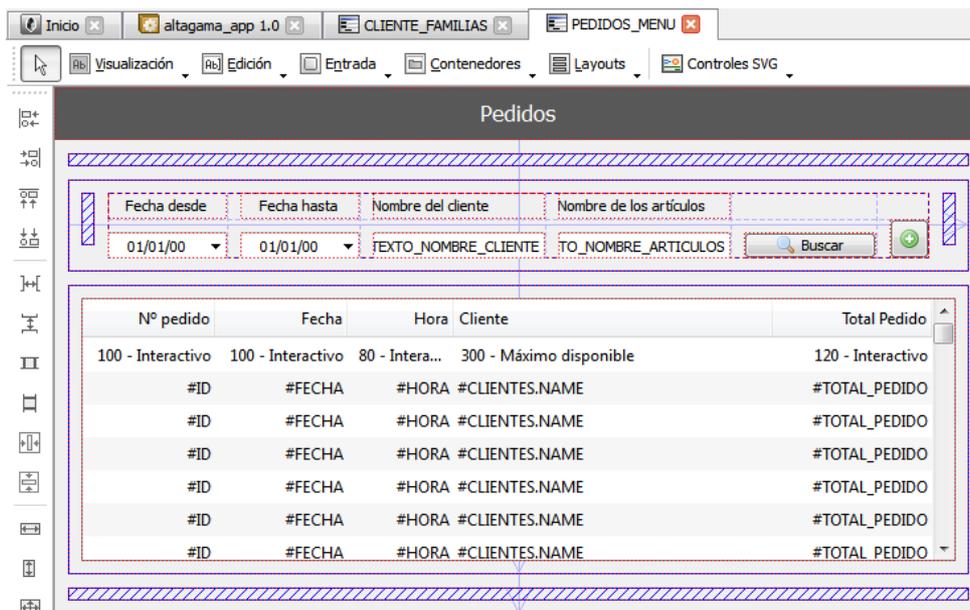
A partir de la versión 7.15 podremos hacer uso de índices complejos en el localizador, por lo que podremos buscar registros de una tabla introduciendo datos de otra, tal y como permiten los índices complejos.

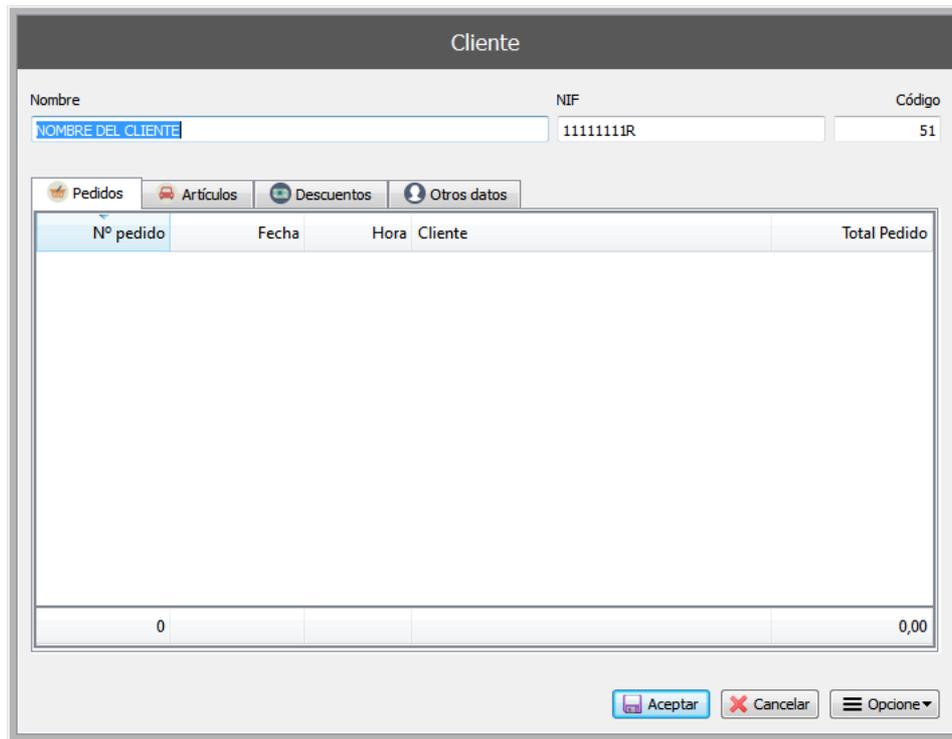
Además, podremos resolver las partes de un índice por medio de variables locales, que podremos alimentar disparando el objeto localizador con ayuda de los comandos de instrucción del grupo Objeto. (VBM-3718).



Optimización de la ejecución de las condiciones de visible y activo de los controles de un formulario

A partir de la versión 7.15 se ha optimizado el funcionamiento de las condiciones de visible y activo, con lo que se reduce el número de veces que se disparan, haciéndolo únicamente las veces necesarias. De esta forma se mejora el rendimiento de los formularios usados en nuestras aplicaciones, lo que redundará en una mejor experiencia de usuario. (VBM-3614).





Implementada la carga de plurales correspondientes a las tablas maestras en las tablas de extensión

En las tablas de extensión, el navegante a plurales mostraba únicamente los plurales de la tabla maestra (padre) pero no funcionaban. Tampoco funcionaba el comando cargar plurales, aunque en el combo del editor si aparecían correctamente los plurales de la tabla maestra (padre).

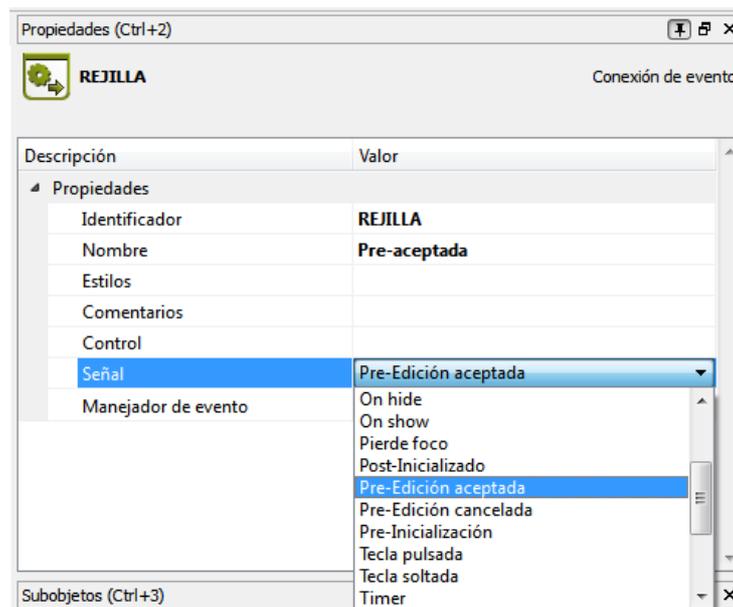
A partir de la versión 7.15 se han implementado las cargas de plurales correspondientes a las tablas maestras de las tablas de extensión, con lo que podremos hacer uso de esta opción en nuestras aplicaciones para ampliar la funcionalidad de las tablas de extensión. (VBM-3639).

Nuevas señales Pre-Edición aceptada y Pre-Edición cancelada

Cuando teníamos una rejilla editable con una conexión de evento asociada a la señal Edición aceptada o Edición cancelada, y en el evento correspondiente usamos el comando "Interfaz: Obtener la ficha en edición de la rejilla()" para acceder a la ficha, en ejecución, pese a aceptar la ficha, el subproceso de este comando no se ejecuta, no ejecutaba.

Este es el comportamiento esperado, ya que cuando se produce el lanzamiento de esa señal, en ese punto ya no está disponible la ficha que estamos editando.

A partir de la versión 7.15 se han implementado dos nuevas señales Pre-Edición aceptada y Pre-Edición cancelada, que podemos asociar a manejador de evento. Cuando estas señales se disparen, la ficha en edición estará viva y podremos acceder a ella a través del comando de instrucción "Interfaz: Obtener la ficha en edición de la rejilla", para realizar las tareas que sean necesarias. (VBM-3641, VBM-3642).



La instrucción "Interfaz: establecer foco al control" permite seleccionar una columna de rejilla en la que posicionarse

A partir de la versión 7.15 se implementa la instrucción "Interfaz: establecer foco al control" que permite seleccionar una columna de rejilla en la que posicionarse, abriendo la edición en el caso de que esté configurada para esa columna.

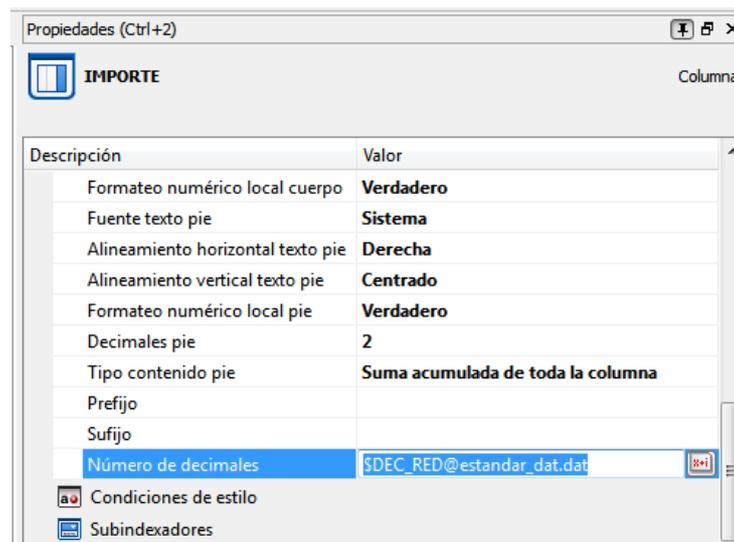
De esta forma podemos gestionar en manejadores de evento de la rejilla la columna en la que queremos el foco según nuestras necesidades, ampliando la funcionalidad de las rejillas en nuestras aplicaciones. (VBM-3644).

Recálculo del valor de la propiedad "Número de decimales" para aplicarlo en cada fila de una rejilla editable

Cuando configuramos el número de decimales en función de una fórmula que toma valor distinto para cada fila, sin entrar a editar, se mostraba correctamente el valor correspondiente con el número de decimales configurado en cada fila, en ocasiones al editar, se mostraba el número de decimales correspondiente a una fila editada anteriormente.

Este comportamiento se debía a que no se estaba tomando en cada momento el valor de decimales por cada fila.

A partir de la versión 7.15, para cada fila se recalcula el valor del número de decimales a aplicar, con lo que podremos editar los números con el número de decimales dependiendo de la fila, facilitando la experiencia de usuario de las rejillas editables de nuestras aplicaciones. (VBM-3665).



Implementación de la inducción en tablas de extensión

A partir de la versión 7.15 se contempla la inducción en tablas de extensión. La inducción se contempla en ambas direcciones, tanto si el plural es de la tabla padre como si es de la tabla de extensión.

- Se induce desde cualquier padre cargados en el mismo objeto u objetos padre, por ejemplo un formulario que tiene un separador de pestañas que dentro tiene otro separador y dentro una rejilla, podrían inducirse maestros de todos los formularios que intervienen en el ejemplo.
- En caso de que en una tabla haya varios punteros al mismo maestro (típico ejemplo de artículos con puntero a los 3 proveedores más habituales) siempre se resuelve el primero y sólo el primero, el resto ya no se resuelven.
- Los maestros y maestros de extensión se consideran como una misma tabla a la hora de hacer la inducción, por lo que si en la misma tabla tienes puntero al maestro y al maestro de extensión se resolverá el primero de ellos (en el orden físico en que estén declarados los campos en la tabla) y sólo el primero. En este caso recomendamos poner primero el maestro y que los campos punteros a maestros de extensión tengan como contenido inicial el campo puntero al maestro. (VBM-3696).

Configuración de idioma por parámetro de Velneo vClient V7 y dentro del qml

Para la configuración del idioma teníamos la opción de modificar una clave de registro que Velneo vClient V7 tomaba como referencia para establecer el idioma al inicio. Pero esto no era posible establecerlo para la personalización del login (por parámetros o Login QML).

A partir de la versión 7.15 se añade la opción de establecer el idioma de ejecución de Velneo vClient V7 por medio del parámetro /loginLocale o /llo, que nos permite establecer el idioma que tomará Velneo vClient V7. Para ello, debemos hacer uso del código de dos caracteres definido por la norma ISO-639, al que podemos añadir la variante del país según la norma ISO 3166-1 alfa-2 en el formato id_PA (Ejemplo: Español de España = "es_ES").

También desde login QML podremos establecer el idioma de inicio de Velneo vClient V7. Por medio de la nueva función setLocale de la clase VLogin, podremos establecer el idioma de ejecución de Velneo vClient V7, indicando el idioma de igual forma que para el parámetro: theLogin.setLocale(id_PA). (VBM-3547).

Revisión de las señales en todos los objetos

En la nueva versión 7.15 se han revisado todas las señales, tanto en ejecución como en edición, aquellas que estaban pendientes de implementar se han incluido en esta versión, y se han quitado del combo de selección cuando esa señal no está permitida, para evitar confusiones. (VBM-3653).

Aquí está el listado completo de las señales que se han revisado:

- Multivista: Grupo FocusKey, Grupo Mouse, Grupo Widget, Timer (No hay propiedad timer en Multivista)
Se han quitado en edición las señales no funcionales y se ha implementado el grupo Widget para este objeto.
- Caja de texto: Funciona la señal Ratón: botón pulsado pero no el resto de señales del Grupo Mouse
Se han quitado en edición las señales del grupo Mouse no funcionales.
- Caja de texto enriquecido: Funciona la señal Ratón: botón pulsado pero no el resto de señales del Grupo Mouse
Se han quitado en edición las señales del grupo Mouse no funcionales.
- Formulario: Grupo Widget
Se ha implementado el grupo Widget para este objeto.
- Bloc: Grupo Widget
Se han quitado en edición las señales del grupo Widget.
- Listbox: Sólo funciona Ítem: cambio seleccionado pero no el resto de señales del grupo Ítem que tiene asignadas
Se han quitado en edición las señales del grupo no funcionales.
- Combobox: Sólo funciona Ítem: cambio seleccionado pero no el resto de señales del grupo que tiene asignadas, que no incluye Ítem: Doble clic.
Se han quitado en edición las señales del grupo Mouse no funcionales.
- Árbol: Timer (El objeto no tiene la propiedad)
Se han quitado en edición la señal ya que no es funcional.
- Rejilla avanzada: Drop finalizado (El objeto no tiene programado drop)
Se han quitado en edición la señal ya que no es funcional.
- Visor HTML:
 - No se disparan la señales Ganar foco y Perder foco
 - No se disparan las señales del grupo mouse excepto Ratón: Movimiento.Se han implementado las señales
- Combobox: Sólo funciona Ítem: cambio seleccionado pero no el resto de señales del grupo Ítem que tiene asignadas
Se han quitado en edición las señales no funcionales excepto Ítem: Activado que ha sido implementada.
- Árbol: Grupo Mouse
Se han quitado en edición las señales del grupo Mouse ya que no son funcionales.
- ListaQML: Ítem simple clic, Ítem doble clic
Se han quitado en edición las señales ya que no son funcionales
- Alternador: ¿Señales show/hide se lanzan dos veces?
Corregido
- Informe: Señal disparada
Se ha implementado la señal
- Bloc: Post-inicializado
Se ha implementado la señal
- Marco: Grupo Show
Se ha implementado el grupo de señales
- Combobox: Ratón doble clic, Ratón movimiento
Se han quitado en edición la señales ya que no son funcionales
- Listbox: Grupo Mouse
Se ha quitado en edición el grupo ya que no son funcionales
- SVG Toggle: Value changed

Se ha implementado la señal

- SVG Switch: Value changed

Se ha implementado la señal

- Objeto dibujo: Value changed

Se ha implementado la señal

- Edición objeto fórmula: Value changed

Se ha implementado la señal

- Selección de fichero: Value changed

Se ha implementado la señal

- Selección de directorio: Value changed

Se ha implementado la señal

- Separador de formularios: Grupo FocusKey

Se ha quitado en edición el grupo ya que no son funcionales

- Caja de formularios: Grupo FocusKey

Se ha quitado en edición el grupo ya que no son funcionales

- Pila de formularios: Grupo FocusKey, Ítem: Cambio de seleccionado (Para cambiar el formulario se usa un manejador de evento)

Se ha quitado en edición el grupo ya que no son funcionales

- Splitter: Grupo FocusKey

Se ha quitado en edición el grupo ya que no son funcionales

- Rejilla avanzada: Grupo Mouse

Se ha quitado en edición el grupo ya que no son funcionales

- Formulario QML: Tiene la propiedad timer pero no tiene la señal. Funciona la señal Ratón: Botón pulsado pero no el resto de señales del grupo Mouse.

Se ha quitado en edición el grupo ya que no son funcionales, y se ha implementado la señal Timer

Los datos se adaptan a la configuración del tipo de campo, por ejemplo Alfa64 o Alfa40, en la pérdida de foco de un campo alfabético

Cuando en ejecución editamos el contenido de un control correspondiente a un campo Alfa64 o Alfa40, al terminar la edición el control no mostraba el contenido actualizado según el tipo de dato hasta que no editábamos otro control.

Por ejemplo:

Si en una tabla declaramos un campo de tipo Alfa64, por ejemplo el campo NIF. En un formulario de origen de la tabla ponemos un control de edición alfabética y en el contenido declaramos el campo #NIF

En ejecución, cuando grabamos datos en el control de edición las letras que escribimos se ven en minúsculas. Al perder el foco el control de edición del campo #NIF los caracteres grabados se siguen viendo en minúsculas.

El contenido de ese campo no lo veremos en mayúsculas hasta que aceptemos el formulario y volver a abrirlo o modifiquemos otro campo, en cuyo caso vemos como el datos grabado en el campo #NIF se pone en mayúsculas.

A partir de la versión 7.15, cuando modifiquemos un contenido de esta forma, al perder el foco del control se actualizará el contenido de acuerdo al campo utilizado. De esta forma el usuario sabe cómo queda el dato nada más introducirlo, sin esperar a grabarlo en disco o editar otro campo. (VBM-1042).

Los cuadros de diálogo de resolución se cierran al aceptar con la tecla intro

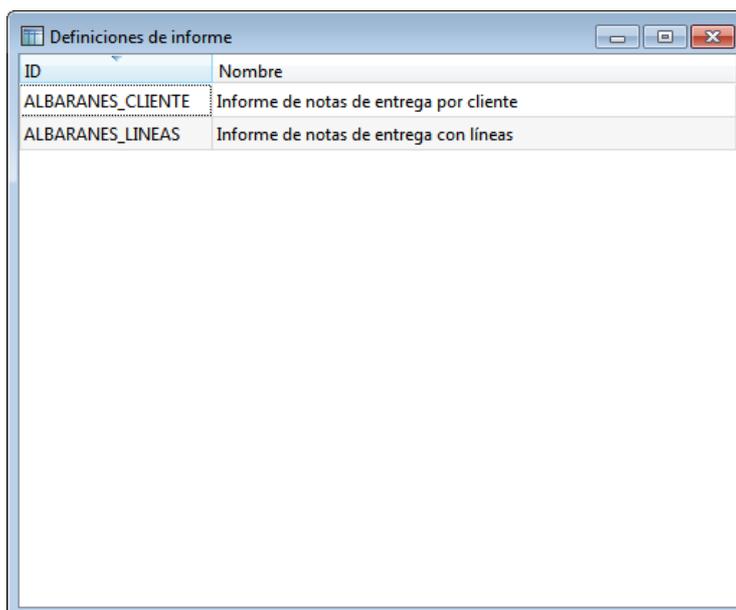
Cuando tratábamos de resolver índices, campos en búsquedas, capilares de tubos, punteros indirectos, etc., nos aparecía un cuadro de diálogo de resolución de fórmulas que no permitía que se cerrara la edición aceptando con el botón Intro del teclado.

El cuadro de diálogo de resolución de fórmulas ha sido reprogramado, mejorándolo y optimizándolo y se le ha añadido la opción de atrapar la tecla Intro para cerrar el cuadro de diálogo. De esta forma mejora la usabilidad de este cuadro de diálogo que usamos tan a menudo. (VBM-1034).

El localizador al construir la rejilla que muestra la lista, toma como ancho de partida el ancho definido en la columna de la rejilla, con lo que podemos definir un ancho mínimo

Cuando presentábamos un localizador con una rejilla que incluía columnas con la propiedad Tipo de ancho con el valor "Máximo disponible", sucedía que la rejilla podría presentarse muy estrecha, ya que se ajustaban al poco espacio disponible.

A partir de la versión 7.15, el localizador al construir la rejilla que muestra la lista, toma como ancho de partida el ancho definido en la columna de la rejilla, con lo que podemos definir un ancho mínimo que permita ver la información que queremos presentar al usuario con suficiente amplitud, mejorando la experiencia de usuario y facilitando al usuario final la selección. (VBM-2088).



Los estilos css se aplican en todos los objetos y controles de formulario

En determinados objetos y controles no se aplicaban los estilos CSS de forma correcta:

Edit de maestro

Rejillas (tanto como objeto como en controles objeto en formulario)

Control edición fecha: Si tiene activa la propiedad "Con calendario".

Control calendario de edit

Calendario

A partir de la versión 7.15, todos ellos aplican los estilos CSS en todos los objetos y controles, con lo que podremos personalizar de forma completa nuestras aplicaciones para adaptarse a las necesidades de nuestros clientes. (VBM-1754).

El texto de la barra de estado se refresca cuando se sale de un menú asociado a un botón

En un formulario principal (asociado al marco) cuando tenemos un control "Botón", con un comando "Menú", al pasar por las opciones de menú aparece el texto en la barra de estado.

Cuando abandonábamos el menú (haciendo click fuera o lanzando la acción del menú), el texto de la barra de estado no se refrescaba.

A partir de la versión 7.15, este texto no permanece, con lo que se evitan errores y confusiones. (VBM-1875).

| Código | Nombre | NIF | Total pedidos |
|--------|--------------------|------------|------------------|
| 14 | CLIENTE CATORCE | 00000000T | 152.094.085,86 |
| 5 | CLIENTE CINCO | 1111111111 | 138.811.282,40 |
| 4 | CLIENTE CUATRO | 00000000T | 148.233.427,52 |
| 19 | CLIENTE DIECINUEVE | 00000000T | 161.510.991,86 |
| 18 | CLIENTE DIECIOCHO | 00000000T | 156.397.524,40 |
| 16 | CLIENTE DIECISESIS | 00000000T | 160.234.953,53 |
| 17 | CLIENTE DIECISIETE | 00000000T | 149.684.426,75 |
| 10 | CLIENTE DIEZ | 00000000T | 138.778.745,95 |
| 12 | CLIENTE DOCE | 00000000T | 151.478.403,85 |
| 9 | CLIENTE NUEVE | 00000000T | 152.193.489,07 |
| 8 | CLIENTE OCHO | 00000000T | 154.159.826,51 |
| 11 | CLIENTE ONCE | 00000000T | 144.636.501,29 |
| | | | 4.434.833.768,04 |

La modificación mediante manejador de evento en un formulario se realiza para todos los tipos de campo

Cuando en un formulario tenía un campo de tipo booleano, fecha o tabla estática, y modificaba los campos usando un evento, al ejecutar el comando de instrucción "Interfaz: Aceptar", los cambios no se reflejaban en la tabla.

A partir de la 7.15, hemos mejorados los manejadores de evento disparados desde formularios para que los comandos de instrucción Interfaz: Aceptar e Interfaz: Guardar la ficha en alta o modificación atrapen los cambios realizados en campos con el comando de instrucción Modificar campo aunque dichos campos hayan sido modificados previamente por el usuario en los controles del formulario. (VBM-1895).

La edición en rejilla con el ratón (doble clic) se conserva en la siguiente línea (on changed) y se mejora la velocidad de cambio de línea

Cuando hacíamos doble clic con el ratón para iniciar edición en una rejilla editable, no se mantenía la edición más que en la siguiente línea, y en ocasiones tardaba más en continuar la edición.

Hemos mejorado la rejilla editable y a partir de la versión 7.15 se soluciona ésta y otras incidencias que hacen de la rejilla un objeto más usable y que proporciona una mejor experiencia de usuario, facilitando la entrada de datos haciendo uso de la rejilla editable. (VBM-2005).

| | |
|----|----------|
| 29 | Prueba 1 |
| 30 | Prueba 1 |
| 31 | Prueba 1 |
| 32 | Prueba 1 |

La tabulación en la rejilla editable salta los campos no editables

En la rejilla editable, cuando tabulábamos para pasar de una columna a otra, cuando se pasaba a uno que no es editable se cerraba la edición.

A partir de la versión 7.15, cuando pasamos a una columna que no es editable se salta a la siguiente y se mantiene la edición. (VBM-0098).

La propiedad "Alto de cabecera" de la rejilla determina el alto de la cabecera cuando tiene un valor mayor que 0 y se autoajusta al contenido con un valor igual a 0

Aunque definiéramos un valor en la propiedad Alto de cabecera, en ejecución no se hacía caso del valor establecido en dicha propiedad y la cabecera de la rejilla tomaba el alto ajustando al contenido de la cabecera.

A partir de la versión 7.15, el valor definido en esta propiedad se tendrá en cuenta a la hora de construir la rejilla. Un valor 0 en esta propiedad hará que el alto de cabecera se comporte como hasta la versión anterior, es decir, se autoajustará al contenido de la cabecera. Un valor distinto de 0, se aplicará al alto de la cabecera independientemente del tamaño del contenido.

Para que no suponga una modificación visual de las aplicaciones que están actualmente en producción, el valor que se tuviera en esta propiedad pasa a ser 0, de tal forma que continuará autoajustándose al alto de la cabecera, y podremos modificarlo si deseamos establecer un alto concreto independiente de su contenido. (VBM-2173).

Cuando se recalcula el contenido de una rejilla, las columnas se ajustan al ancho de la rejilla cuando se muestra la barra de scroll o se quita

Cuando cambiaba el contenido de una rejilla, por ejemplo con Interfaz: Procesar rejilla, o Quitar ficha de la lista, la barra de scroll dejaba un espacio en blanco si ya no era necesaria y no se mostraba, o tapaba los registros si se mostraba al tener más registros de los que podía mostrar la rejilla.

A partir de la versión 7.15, las columnas se ajustarán al ancho de la rejilla cuando se muestre la barra de scroll o se quite, evitando mostrar un espacio innecesario o tapar parte de la última columna de la rejilla, mejorando la experiencia de usuario. (VBM-2505).

| Nº pedido | Fecha | Hora | Cliente | Total Pedido |
|-----------|----------|-------|--------------------|------------------|
| 22117 | 19/02/14 | 09:26 | CLIENTE CUATRO | 1.424.731,00 |
| 22116 | 19/02/14 | 08:24 | CLIENTE CATORCE | 2.736.326,00 |
| 22112 | 19/02/14 | 08:14 | CLIENTE CUATRO | 221.991,00 |
| 22111 | 19/02/14 | 08:13 | CLIENTE CINCO | 1.031.305,00 |
| 22110 | 19/02/14 | 00:12 | CLIENTE DIECISESIS | 3.001.350,00 |
| 22108 | 18/02/14 | 20:08 | CLIENTE CUATRO | 1.057.235,00 |
| 22107 | 18/02/14 | 19:49 | CLIENTE CUATRO | 203.533,00 |
| 22106 | 18/02/14 | 19:47 | CINCUENTA | 610.599,00 |
| 22105 | 18/02/14 | 10:53 | CLIENTE CATORCE | 1.057.235,00 |
| 22104 | 18/02/14 | 10:29 | CINCUENTA | 42.176,00 |
| 22103 | 18/02/14 | 10:27 | CINCUENTA | 203.533,00 |
| 22101 | 17/02/14 | 20:19 | CLIENTE CINCO | 20.260,00 |
| 22.090 | | | | 4.449.090.163,34 |

Se elimina un registro que se muestra con un formulario abierto desde una rejilla con el estilo simple clic

Cuando usábamos el estilo simple clic para que se muestre el formulario correspondiente a una rejilla, por ejemplo, y en el formulario pulsábamos el botón con la función eliminar, aunque mostraba el mensaje de confirmación de la eliminación del registro, al pulsar el botón sí, realmente no se eliminaba el registro y seguía apareciendo en la rejilla.

A partir de la versión 7.15, la eliminación se produce correctamente, quitando de la rejilla la fila correspondiente y evitando la confusión que generaba el uso de simple clic para mostrar formularios. (VBM-2621).

Mejorados editores automáticos en Velneo vDataClient V7

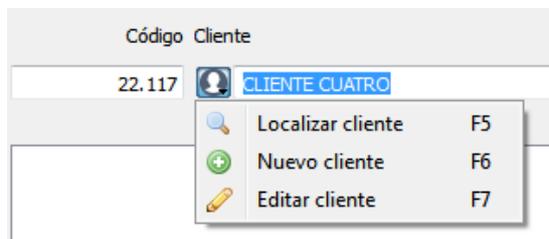
- En Velneo vDataClient V7 al editar la columna de un campo fecha aunque el campo no tuviera contenido inicial, asumía por defecto la fecha de hoy, y una vez asumida no se podía eliminar.
- En edición de fechas, horas y tiempo no se crea el control adecuado y falla el tabulador.
- Campos punteros a tabla estática no podían modificarse
- Los campos booleanos se mostraban como uno y cero.
- No se producía el salto de columnas no editables.

A partir de la versión 7.15 se mejora y optimiza la rejilla editable de Velneo vDataClient V7, y se solucionan éstas y otras incidencias con lo que se mejora la experiencia de usuarios, facilitando el mantenimiento de datos de las tablas. (VBM-2626).

La propiedad "Alto" se aplica tanto al botón y como al alto del control en un control de edición de maestro que tiene declarado tipo de menú de botón

Cuando en un formulario incluíamos un control de edición de maestro que tenía declarado tipo de menú de botón, el valor de la propiedad "Alto" se aplicaba únicamente al botón y no al alto del control.

A partir de la versión 7.15, el valor que establezcamos en la propiedad "Alto" se aplica a todo el control, incluido el botón, con lo que podemos definir el alto que tendrá el control completo y adaptando la presentación del control en nuestro formulario a nuestras necesidades. (VBM-2688).



A partir de la versión 7.15, cuando nos movemos de un control a otro una vez iniciada la edición, se selecciona el contenido, facilitando al usuario final la edición de los datos, mejorando la experiencia de usuario. (VBM-2740).

El botón del comando de instrucción mensaje aplica los estilos css configurados

Pese a definir un estilo CSS específico para aplicarlo en la ventana que genera el comando de instrucción Mensaje, éste no se aplicaba.

A partir de la versión 7.15, solucionada la incidencia, la ventana de mensajes aplicará el CSS correspondiente, permitiéndonos personalizar por completo según nuestras necesidades. (VBM-2745).

En rejilla editable, la edición de campo fecha continúa en la siguiente columna editable

Al llegar a la edición de un campo fecha con la tecla de tabulación se pasaba por el día, el mes y el año y al tabular finalizaba la edición de la fila en lugar de continuar editando el siguiente campo. El cursor seguía avanzado con la tecla tabulación pero sin entrar en modo edición.

A partir de la versión 7.15, una vez finalizada la edición de cada una de las partes de la fecha, se pasa al siguiente control sin cerrar la edición, con lo que se facilita la introducción de datos por parte del usuario final, mejorando su experiencia de usuario. (VBM-2766).

En una rejilla editable, si hay error en el alta, la línea en blanco correspondiente se limpia

Si en una rejilla editable hacíamos un alta de nuevo registro e inmediatamente pulsábamos la tecla Enter, el sistema intentaba hacer el alta, pero no la completaba por ser inválida, y la línea en blanco (sin ID) permanecía en la rejilla hasta que la refrescábamos. Además, no podíamos reutilizar esa línea para una nueva entrada.

A partir de la versión 7.15 esta incidencia se resuelve y en el caso de introducir una línea sin los datos necesarios, la fila no se limpia y podremos seguir introduciendo registros. Con esto facilitamos el mantenimiento de datos a través de rejillas editables, con lo que se mejora la experiencia de usuario. (VBM-2825).

Le ejecución del comando de instrucción set retorno proceso = NO en un manejador de evento lanzado por la señal edición iniciada limpia la línea en blanco de la rejilla

Para evitar la posibilidad de crear fichas desde una rejilla editable programábamos un evento de interfaz que, en caso de alta, retorne NO, disparado mediante una señal del tipo "edición iniciada".

En ejecución, al hacer el usuario doble clic en cualquier área vacía de la rejilla, se iniciaba la edición, se dispara el evento y se cancelaba la creación del registro, pero se pintaba una nueva línea en blanco al final de la rejilla.

A partir de la versión 7.15, cuando cancelemos la edición no se pintará una línea en blanco al final de la rejilla, mejorando la experiencia de usuario en las aplicaciones en que usemos la rejilla y esta funcionalidad. (VBM-2981).

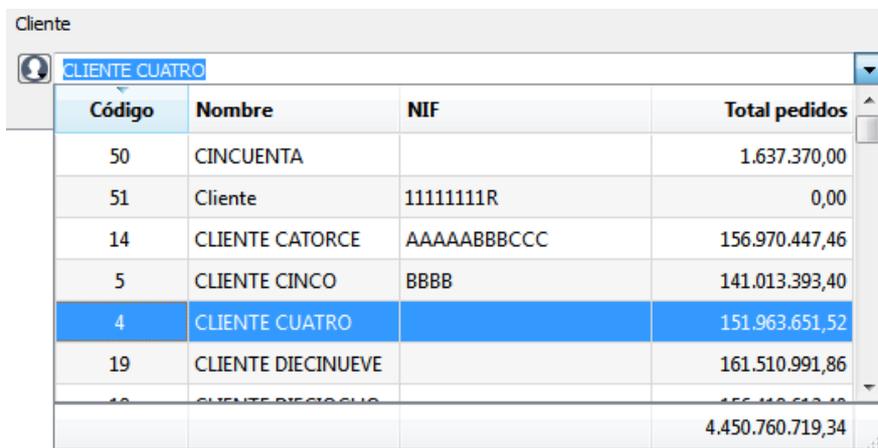
En linux se implementa la selección de contenido en rejillas editables con campos maestro y numéricos cuando nos movemos por los registros en la misma columna

En Linux en una rejilla editable que editaba un maestro o un dato numérico, al movemos arriba y abajo por los distintos registros, haciendo uso de las teclas de dirección, no quedaba seleccionado todo el contenido del campo, como en un inicio de edición.

El registro dado de alta a partir de una acción asociada a un control de edición de un maestro se incluye en la vista de datos asociada al edit de maestro

En un formulario con un control de edición de maestro con vista de datos de lista, cuando creábamos un maestro, aunque se inducía a la ficha en curso, al desplegar la vista de datos no aparecía.

A partir de la versión 7.15, cuando demos de alta un registro maestro, el contenido de la vista de datos se refresca y se muestra en la vista. (VBM-2925).



| Código | Nombre | NIF | Total pedidos |
|--------|--------------------|-------------|------------------|
| 50 | CINCUENTA | | 1.637.370,00 |
| 51 | Cliente | 11111111R | 0,00 |
| 14 | CLIENTE CATORCE | AAAAABBBCCC | 156.970.447,46 |
| 5 | CLIENTE CINCO | BBBB | 141.013.393,40 |
| 4 | CLIENTE CUATRO | | 151.963.651,52 |
| 19 | CLIENTE DIECINUEVE | | 161.510.991,86 |
| | | | 4.450.760.719,34 |

Alta de registro en una rejilla editable asignada a la propiedad Vista de datos de lista

Estando en un control enlazado a maestro, con la propiedad de Vista de datos de lista activa y asignada una rejilla editable, si pulsábamos doble clic iniciando la edición de la rejilla, Velneo vClient daba un error y se cerraba.

A partir de la versión 7.15 este error no se produce, permitiendo el alta por medio de formulario de un nuevo registro maestro, facilitando la gestión de maestros al usuario final. (VBM-2961).

Las vistas desplegadas cogen el ancho del control base, permitiendo que en ejecución se modifique el ancho del control

El cálculo de ancho de la rejilla asociada a una vista de datos de un edit variaba en función del número de registros. Si el maestro contenía pocos registros, se aplicaba el tamaño por defecto del objeto (la rejilla) y cuando había muchos registros, se ajustaban el ancho y el alto de la rejilla, dando a la rejilla el ancho del control.

A partir de la versión 7.15 este control ha sido mejorado y ahora las vistas desplegadas cogen el ancho del control base, permitiendo que en ejecución se modifique el ancho del control con lo que se facilita la usabilidad del control, mejorando la experiencia de usuario. (VBM-2969).

Mejora en la ordenación de listas

A partir de la versión 7.15 disfrutaremos de una optimización en los procesos que requieran la ordenación de listas.

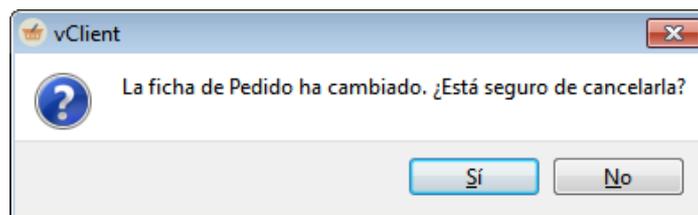
El ordenamiento se realiza en bloques mayores, más fáciles de gestionar y destruir con lo que mejoran todos los procesos que requieren ordenación y que trabajan con gran volumen de datos: Ordenar listas de gran tamaño en procesos, rejillas, salidas de búsquedas, etc., se verán beneficiadas de esta optimización, reduciendo los tiempos de trabajo mejorando la experiencia de usuario, que deberá esperar menos tiempo por la finalización de estas tareas. (VBM-3565).

El comando de botones "deshacer controlado" pide confirmación cuando se modifican líneas y no el maestro

Si activamos el bloqueo duro al un formulario clásico de cabecera con líneas e incluimos un botón que dispara el comando "deshacer controlado", cuando en el formulario de cabecera añadíamos alguna línea, al pulsar el botón, el sistema no pedía confirmación y deshacía la transacción directamente. Esto sucedía en alta, modificación y baja.

A partir de la versión 7.15, en los tres casos se agrupa la transacción y se incluyen las operaciones correspondientes, solicitando al usuario confirmación en el caso de cancelar el formulario de cabecera, tal y como espera, mejorando la usabilidad de las aplicaciones en las que hagamos uso de bloqueo duro.

Esta incidencia está relacionada con la incidencia VBM-3622 - En alta de formulario con bloqueo duro emular el bloqueo duro aunque no exista la ficha, que cierra el ciclo de esta incidencia, que se refiere al proceso de alta, ya que ahí se ha de simular el bloqueo duro al no haber realmente una ficha todavía. (VBM-3162).



Unificación del criterio de visualización de los campos booleanos en la rejilla

Cuando se visualiza un campo booleano en la columna de una rejilla se mostraban los valores Sí y No. Sin embargo, si la rejilla es editable se mostraba en la columna un combo con los valores Verdadero y Falso.

A partir de la versión 7.15 se unifica el criterio de visualización de los campos booleanos en la rejilla, usándose el criterio Sí/No en todos los casos. (VBM-3405).

Se dispara el evento gana foco de un control de un subformulario incluido en un separador de pestañas que tiene delante otro subformulario que no es visible

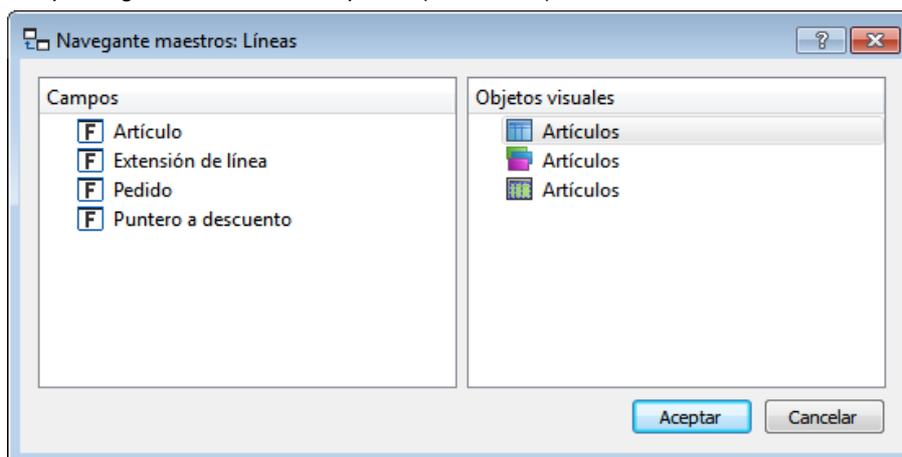
Cuando en un separador de pestañas incluimos un subformulario que no es visible porque la condición de visibilidad no se cumple, no se dispara el evento gana foco de un subformulario que esté a continuación.

Solucionada la incidencia, la señal se dispara de forma correcta, por lo que podremos usarla para cubrir necesidades de nuestra aplicación. (VBM-3426).

La opción de navegante cargar maestros muestra los campos punteros a las tablas maestras de extensión

La opción Cargar maestros del Navegante no mostraba los campos enlazados a tablas maestras de extensión, con lo que no era posible realizar esta navegación a maestro.

A partir de la versión 7.15, la opción Cargar maestros presentará no sólo los campos enlazados a maestro si no que también podrá el usuario seleccionar de entre los campos enlazados a tablas maestras de extensión, con lo que se amplía la funcionalidad de esta opción de menú en stock, ampliando de forma automática su funcionalidad en todas las aplicaciones en las que hagamos uso de esta opción. (VBM-3489).



Se aplican las mismas configuraciones de fuente a las rejillas, tanto si se muestran sin edición, como cuando el usuario inicie la edición de la rejilla

En una rejilla editable, a la que le habíamos personalizado la fuente del cuerpo de las columnas, cuando una columna de la rejilla pasaba a modo editable, se cambiaba la fuente pasando a utilizar para la edición la fuente del sistema.

Esto sucedía también en los siguientes escenarios:

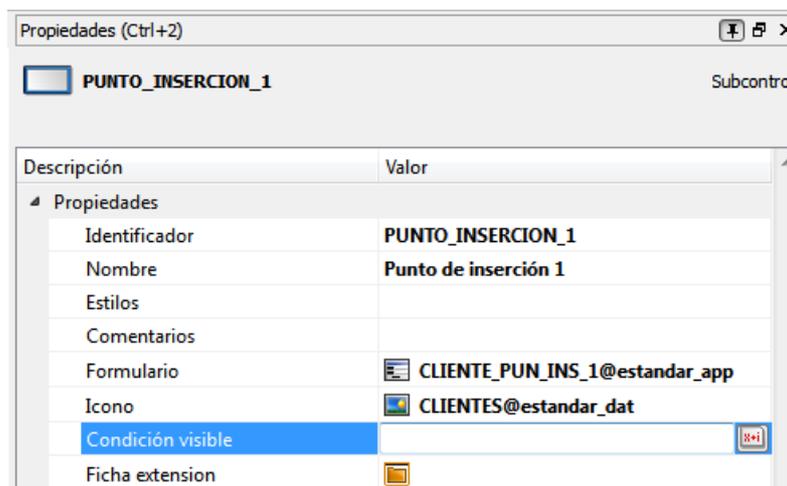
- Con alteración de la de sistema, incrementando su tamaño.
- Asignando una fuente personalizada distinta de la del sistema.
- Personalizando la fuente mediante la aplicación de hojas de estilo CSS.

A partir de la versión 7.15, se aplican las mismas configuraciones de fuente a las rejillas, tanto si se muestran sin edición, como cuando el usuario inicie la edición de la rejilla, de esta forma podemos personalizar también la visualización de las rejillas editables, de acuerdo a nuestras necesidades, lo que redundará en una mejor experiencia de usuario y usabilidad de nuestras aplicaciones. (VBM-3521).

Implementada evaluación de una condición visible de un subobjeto inserción de un formulario de una tabla de extensión

Una condición visible de un subobjeto Inserción de un formulario de una tabla de extensión el sistema en ejecución devolvía campo inexistente al evaluarla cuando contenía campos de la tabla padre. Esto era debido a que la fórmula se calculaba con origen la ficha del formulario principal en lugar de la ficha de extensión.

Solucionada la incidencia, ahora el cálculo de la fórmula se realiza con origen la ficha de extensión, podremos usar sin problema campos de la tabla padre para las condiciones permitiéndonos aumentar las posibilidades para el condicionamiento de los puntos de inserción. (VBM-3550).



Disparo de los contenidos iniciales de todos los campos en las tablas de extensión

Un campo de la tabla de extensión que tuviera como contenido inicial una fórmula que no dependiera de campos no se disparaba en el alta de la ficha.

Solucionada la incidencia, los contenidos iniciales de fichas de tablas de extensión se disparan tanto si usan campos dependientes como si no, permitiendo inicializar los campos que requieran ese tipo de fórmulas. (VBM-3562).

Se muestran en el formulario las modificaciones realizadas con "Interfaz: Obtener ficha en edición" en un manejador de evento asociado a señal disparada que se lanza desde una acción

En una rejilla editable, teníamos una columna con un menú que tiene una acción que dispara una señal.

El manejador de evento asociado a esta señal realizaba modificaciones en el campo correspondiente a la columna en curso con el comando de instrucción "Interfaz: Obtener ficha en edición".

En ese caso la modificación no se mostraba una vez realizada, sin embargo la modificación estaba hecha realmente: si hay columnas que obtienen su valor a partir de ese campo (bien por contenidos iniciales, bien porque sea un maestro), se ve refrescada esa información y al aceptar se guarda.

A partir de la versión 7.15 los datos modificados mediante "Interfaz: Obtener ficha en edición" en un manejador de evento asociado a señal disparada, lanzada desde una acción, se refrescan correctamente. (VBM-3590).

En una rejilla editable cuando la propia edición de una columna hace que no se cumpla la condición de edición de columna se puede continuar la navegación por la rejilla

Es una rejilla editable que tenía una única columna editable (campo NAME) de la tabla y como condición para editar columna usaba la fórmula siguiente: #NAME:isEmpty()

En ejecución, si el último registro de la rejilla tenía el campo NAME vacío, editábamos la celda y le dábamos un valor, al tabular Velneo vClient V7 se quedaba bloqueado.

A partir de la versión 7.15, este caso es considerado en la tabulación y permite navegar de forma coherente con las condiciones establecidas, permitiendo al usuario final continuar trabajando en ese punto. (VBM-3606)

En una columna de rejilla editable puntero se muestra el menú contextual de la vista de datos a un edit de maestro

Si en una columna de una rejilla editable que editaba un campo puntero a maestro le declarábamos la propiedad vista de datos, en ejecución usando el menú contextual de la vista de datos en esa columna dábamos de alta el maestro, rompía Velneo vClient V7 al mostrarse el formulario correspondiente en vista de datos.

A partir de la versión 7.15, en ese caso el formulario se mostrará en cuadro de diálogo, ya que una vista de datos que proviene de un popup, ha de mostrarse también como modal. De esta forma, además, evitamos el error y el usuario final podrá dar de alta un registro sin provocar incidencia alguna en Velneo vClient V7. (VBM-3608).

En rejilla editable se dispara la señal edición aceptada si estando en edición hacemos clic con el ratón en una columna no editable

En una rejilla editable no se disparaba la señal edición aceptada cuando en edición hacíamos clic con el ratón en una columna no editable. Aunque sí se disparaba cuando cambiamos de celda usando teclado.

A partir de la versión 7.15, se implementa que acceder a una columna no editable implique el final de edición y por tanto se dispare la señal de edición aceptada. (VBM-3612).

El control edit de maestro se mejora incluyendo la opción de alta de registros en la vista de datos asociada

En una edit de maestro cuando le declaramos una vista de datos con una rejilla editable, al dar alta en la rejilla Velneo vClient V7 daba error y se cerraba si no había datos en la rejilla editable.

A partir de la versión 7.15 se ha mejorado y optimizado el funcionamiento del control edit de maestro, por lo que podremos declarar una vista de datos con una rejilla editable, pudiendo dar altas en esa misma rejilla, ampliando las opciones para dar funcionalidad a nuestras aplicaciones. (VBM-3628).

Mejora de la presentación de mensajes en rejillas editables

En una rejilla editable que tenía una conexión de evento asociada a la pérdida de foco de la rejilla, el manejador de evento que se ejecuta mostraba un mensaje con el comando de instrucción correspondiente Mensaje.

Cuando entraba a editar una celda, se disparaba la pérdida de foco, se ejecutaba el manejador de evento, se presentaba la ventana de mensaje, y al aceptar pulsando el botón correspondiente vClient daba error y se cerraba..

A partir de la versión 7.15, se soluciona esta incidencia. Sin embargo, hemos de tener en cuenta que en los manejadores de evento de foco no se recomienda hacer uso de mensajes, debemos mostrar la información de otra manera. (VBM-3645).

Implementación de los comandos Ficha: Localizar, Alta y Editar Maestro aplicadas en un menú de botón de un edit de maestro en una extensión de ficha

A partir de la versión 7.15 podremos hacer uso de las opciones de localizar, alta y editar maestro en controles de edición de maestro en extensiones de ficha, por lo que podremos aumentar la funcionalidad de nuestras aplicaciones, facilitando al usuario final la selección del maestro para la ficha en curso. (VBM-3651).

Se despliega la vista de datos de un control edit de maestro al pulsar el botón con el ratón, en un formulario abierto con un botón en modo popup desde otro

Cuando invocábamos un formulario desde el comando de botones mostrar subformulario popup y dicho formulario incluía un edit con la propiedad "vista de datos de lista" activada, no se desplegaba la lista haciendo clic sobre el botón para desplegar la lista.

Mientras que si usábamos la tecla aceleradora F4 sí se desplegaba.

A partir de la versión 7.15 podremos desplegar con el ratón la vista de datos una vez solucionada la incidencia, mejorando la experiencia de usuario y facilitando al usuario final el uso de esta funcionalidad en nuestras aplicaciones. (VBM-3655).

Alineamiento de los pie de rejilla con las columnas correspondientes

En una rejilla con pie activo verdadero y con un modo de cabecera que incluya el vertical, el pie quedaba desplazado de sus columnas al recargar registros en la rejilla cuando cambiaba el ancho de la cabecera vertical.

En la carga inicial, cuando cargábamos algunos registros, el pie aparecía bien alineado, pero cuando luego cargábamos más registros, esto producía que aumentara el ancho de la primera columna "Cabecera vertical", desalineándose el pie.

A partir de la versión 7.15 se ha implementado el alineamiento del pie de la rejilla con las columnas, de tal forma que mejorará el aspecto visual de las rejillas con pie de nuestras aplicaciones, mejorando por tanto la experiencia de usuario. (VBM-3693).

| Nº pedido | Fecha | Hora | Cliente | Total Pedido |
|-----------|----------|-------|--------------------|------------------|
| 22117 | 19/02/14 | 09:26 | CLIENTE CUATRO | 1.424.731,00 |
| 22116 | 19/02/14 | 08:24 | CLIENTE CATORCE | 2.736.326,00 |
| 22112 | 19/02/14 | 08:14 | CLIENTE CUATRO | 221.991,00 |
| 22111 | 19/02/14 | 08:13 | CLIENTE CINCO | 1.031.305,00 |
| 22110 | 19/02/14 | 00:12 | CLIENTE DIECISESIS | 3.001.350,00 |
| 22108 | 18/02/14 | 20:08 | CLIENTE CUATRO | 1.057.235,00 |
| 22107 | 18/02/14 | 19:49 | CLIENTE CUATRO | 203.533,00 |
| 22106 | 18/02/14 | 19:47 | CINCuenta | 610.599,00 |
| 22105 | 18/02/14 | 10:53 | CLIENTE CATORCE | 1.057.235,00 |
| 22104 | 18/02/14 | 10:29 | CINCuenta | 42.176,00 |
| 22103 | 18/02/14 | 10:27 | CINCuenta | 203.533,00 |
| 22101 | 17/02/14 | 20:19 | CLIENTE CINCO | 20.260,00 |
| 22.090 | | | | 4.449.090.163,34 |

Mejora de la ejecución de las señales del grupo Ítem en conexiones de evento de formulario, de un alternador incrustado en formulario

En un formulario en el que incluíamos un alternador y en el formulario creábamos conexiones de evento con señales del tipo: Ítem: cambio de seleccionado, Ítem: simple clic asociadas al alternador, en ejecución, estas señales se disparaban dos veces cuando realizábamos la acción asociada.

A partir de la versión 7.15 este comportamiento no esperado se evita, con lo que las señales se dispararán una única vez en cada ocasión, evitando de esta forma disparos innecesarios de manejadores de evento. (VBM-3660).

Al borrar el primer o último registro de una rejilla se ejecuta el manejador de evento asociado a la señal Ítem: cambio de seleccionado al posicionarse en otro registro

En una rejilla definíamos un manejador de evento que mostraba información en un mensaje del registro seleccionado.

Al eliminar el primer registro o el último no se ejecutaba el manejador de evento asociado a la señal Ítem: cambio de seleccionado, pese a que se posicionaba en otro registro una vez eliminado y quitado de la rejilla el que se había borrado.

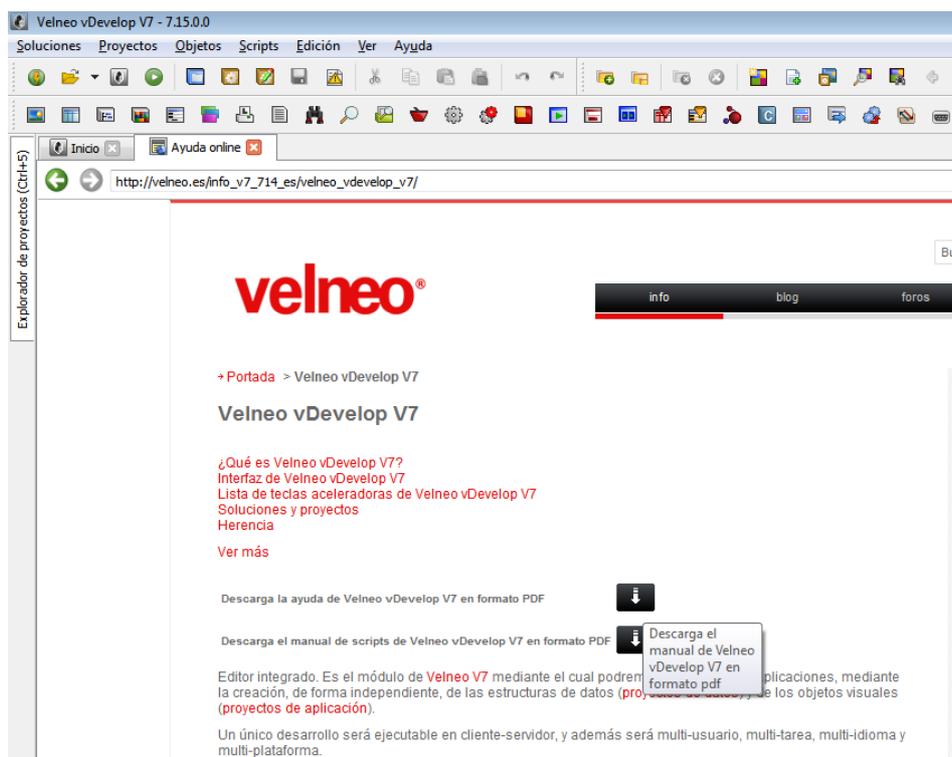
A partir de la versión 7.15, tanto si eliminamos cualquier otro registro como el primero o el último, se ejecuta el manejador

de evento asociado a la señal de cambio de seleccionado, con lo que podremos usar esta señal para ejecutar un manejador de evento con las operaciones que nos interese realizar, aumentando la funcionalidad de nuestras aplicaciones. (VBM-3666).

En una página html mostrada en un control visor HTML se muestran los tooltips definidos con la etiqueta "alt"

En una página html que incluía una imagen con un atributo "alt", al pasar con el ratón no se mostraba en tooltip el texto correspondiente al atributo.

A partir de la versión 7.15, cuando las imágenes con el atributo "alt" mostrarán un tooltip con el texto definido, con lo que podemos facilitar información al usuario final de esta forma, mejorando la experiencia de usuario. (VBM-2513).



En un manejador de evento después de ejecutar otro con el comando Interfaz: Ejecutar manejador de evento, se ejecutan los comandos de cesta sobre las cestas locales creadas previamente

En un manejador de evento creamos una cesta local. A continuación ejecutábamos un manejador de evento de V7 o javascript con el comando de instrucción Interfaz: Ejecutar manejador de evento, y a partir de ese momento, los comandos de cesta aplicados a la cesta local anteriormente creada no tenían efecto.

Por ejemplo, un mensaje en el subproceso del comando de instrucción Cesta: Procesar(CES) no se mostraba.

Esto se debía a que la ejecución de un manejador de evento hacía como si fuese el inicial, con lo que al finalizar limpiaba las cestas.

A partir de la versión 7.15 se controla que no se limpien las cestas correspondientes al manejador de evento llamador, con lo que la ejecución de las cestas en éste será la correcta y podremos hacer uso de cestas en cualquier punto facilitando el desarrollo de nuestros manejadores. (VBM-3672).

Un formulario es lanzado en cuadro de diálogo en lugar de en vista si alguno de sus ascendientes es popup

A partir de la versión 7.15, los formularios que se muestren en cuadro de diálogo, cuando alguno de sus ascendientes es popup, lo harán en cuadro de diálogo, en modal, en vez de mostrarse en vista, debido a que esto último podría producir incidencias. De esta forma mejora la experiencia de usuario de nuestras aplicaciones. (VBM-3679).

Las modificaciones realizadas en formularios incluidos por herencia inversa de tablas de extensión se realizan en la ficha principal, evitando redundancias

Cuando modificábamos un campo de la ficha principal al aceptar daba error porque antes de modificar la ficha principal en su sitio normal la modificaba en el pre de fichas de extensión por cada subformulario de extensión ya que la ficha principal es para estos una ficha de extensión con modificación en pre.

Solucionada la incidencia, las modificaciones realizadas de esta forma se realizan únicamente en la ficha del formulario principal, con lo que se evita la redundancia y el error. (VBM-3681).

Implementados los edit de maestro sobre fichas de extensión en formularios sin origen

Trabajando en un formulario sin origen, cuando en una ficha de extensión incluíamos un edit de maestro, éste no era funcional.

A partir de la versión 7.15 se implementa el funcionamiento de los controles de edición de fichas de extensión en formularios sin origen, con lo que podremos hacer uso de estos controles en nuestras aplicaciones, ampliando la funcionalidad de las tabla de extensión. (VBM-3682).

Mejorada la indexación de índices complejos

Tenemos tres tablas: Clientes, Facturas e Líneas.

Tenemos un índice configurado como:

Tabla asociada: Líneas

Tabla de búsqueda: Líneas

Tabla para indexar en el índice: Clientes, campo Nombre del cliente.

Tenemos un cliente con código 1, con una factura con código 2, y al menos una línea.

Tenemos un cliente con código 2, que no tiene ningún plural, ni facturas como plural ni por tanto, ninguna línea.

Cuando modificamos el nombre del cliente con código 2 que no tiene plurales, y su código se corresponde con el código de una factura, en este caso la factura código 2 del cliente 1, no se modifica el registro.

A partir de la versión 7.15, solucionada esta incidencia, es posible indexar en este tipo de índices de forma normal, con lo que podremos aprovechar sin límites en nuestras aplicaciones la potencia de los índices complejos. (VBM-3689).

En una rejilla editable, en una columna editable que tiene vista de datos abierta si pulsamos la tecla Esc se cierra la edición pero se evita cerrar la vista de datos

En una rejilla editable, en una columna editable que tenía una vista de datos abierta si pulsábamos la tecla Esc se cerraba la vista de datos y también cerraba la edición en línea, en vez de quedar el foco en el campo en edición, como en los formularios.

Esto sucedía también con las teclas Enter y Return

A partir de la versión 7.15, cuando usemos cualquiera de estas teclas, se cerrará la vista de datos pero se mantiene la edición en la rejilla, con lo que mejoramos la usabilidad y la experiencia de usuario. (VBM-3690).

Velneo vDataClient V7 muestra todas las columnas correspondientes a los campos de la tabla, incluyendo punteros singulares o hermanos cuando se encuentran en las últimas posiciones de la lista de campos

Con Velneo vDataClient V7 cuando visualizábamos una tabla en la que los últimos campos por orden eran punteros singulares o hermanos, éstos no se mostraban en la rejilla, no pudiendo ver los valores correspondientes.

A partir de la versión 7.15 se ha mejorado la rejilla de edición de Velneo vDataClient V7, solucionando la incidencia, y permitiendo visualizar las columnas correspondientes a los campos punteros aunque estos se encuentren al final de la tabla, facilitando al administrador la gestión y mantenimiento de registros con esta herramienta. (VBM-3703).

| | Código #27696 | Pedido | Artículo | Cantidad | Precio | Puntero a descuento |
|----|---------------|--------|------------------------|----------|---------------|---------------------|
| 1 | 1 | 1 | Porsche 911 | 2.000000 | 99393.000000 | |
| 2 | 2 | 1 | Mercedes C 220 CDI | 1.000000 | 41083.000000 | |
| 3 | 1 | 2 | Ferrari 458 Italia | 1.000000 | 219939.000000 | |
| 4 | 1 | 3 | Mercedes E 350 CDI | 1.000000 | 13825.000000 | |
| 5 | 1 | 4 | Porsche 911 | 1.000000 | 24849.000000 | |
| 6 | 2 | 4 | Mercedes C 220 CDI | 2.000000 | 10271.000000 | |
| 7 | 3 | 1 | Ferrari 458 Italia | 2.000000 | 219939.000000 | |
| 8 | 3 | 4 | Porsche 911 | 1.000000 | 24849.000000 | |
| 9 | 4 | 4 | Mercedes C 220 CDI | 2.000000 | 10271.000000 | |
| 10 | 5 | 4 | Ferrari 458 Italia | 1.000000 | 54985.000000 | |
| 11 | 1 | 5 | Audi A8 | 1.000000 | 46260.000000 | |
| 12 | 2 | 5 | Audi A8 3.0 TDI | 1.000000 | 73320.000000 | |
| 13 | 1 | 6 | Citroen C4 Picasso | 1.000000 | 17100.000000 | |
| 14 | 1 | 7 | Ferrari 458 Italia | 1.000000 | 219939.000000 | |
| 15 | 1 | 8 | Ferrari F12 Berlinetta | 1.000000 | 279809.000000 | |

Mejora del comando de instrucción Cargar plurales para la resolución de instancias en tercer plano

La instrucción "Cargar plurales" en tercer plano devolvía los registros correspondientes a las posiciones de los plurales de la última instancia creada en instancias que reutilizan una instancia con la tabla maestra

Tenemos un proyecto de datos que contiene la tabla MAESTROS.

Tenemos otro proyecto de datos que hereda del anterior y contiene la tabla HISTORICOS.

La tabla HISTÓRICOS es plural de MAESTROS.

Instanciamos el proyecto de datos que contiene la tabla MAESTROS una única vez.

Instanciamos el proyecto de datos que contiene la tabla HISTORICOS dos veces, reutilizando en ambos caso la misma instancia del proyecto que contiene la tabla MAESTROS.

De esta forma, ambas instancias, compartirán la tabla MAESTROS, conteniendo los mismos maestros, pero los plurales serán diferentes para cada instancia.

Un proceso con origen ficha en MAESTROS ejecuta un proceso en tercer plano que usa el comando de instrucción Cargar plurales para devolver los correspondientes al maestro.

Si ejecutamos el proceso en la segunda instancia de datos (la última que hemos creado), nos devuelve los registros plurales correspondientes.

Si ejecutamos el proceso en la primera de las instancias de datos (la primera que hemos instanciado), nos devuelve los registros existentes en las posiciones correspondientes a los plurales de la instancia anterior. Incluso, si las posiciones no está ocupadas por plurales, devuelve posiciones vacías.

Si creamos más instancias de datos de plurales, heredando la instancia común de maestros, en todas las instancias anteriores se devolverán los plurales de acuerdo a las posiciones de los plurales en la última instancia creada.

A partir de la versión 7.15 se mejoran las funciones que usa el servidor para resolver la instancia que debe usar, incluyendo el soporte para este caso, lo que nos permitirá incrementar las estructuras de ejecución de nuestras instancias. (VBM-3692).

La reordenación de registros de una rejilla mantiene la multi-selección

En una rejilla con multiselección, cuando teníamos varios registros seleccionados, al reordenar el contenido de la rejilla, la multiselección mantenía sus posiciones, la selección no se movía con el registro y dejaba por tanto seleccionados otros registros.

A partir de la versión 7.15 se mejora el sistema de multiselección, permitiendo reordenar el contenido de la rejilla manteniendo la multiselección, mejorando la experiencia de usuario de las rejillas de nuestras aplicaciones y evitando confusiones. (VBM-3695).

| Nº pedido | Fecha | Hora | Cliente | Total Pedido |
|-----------|----------|-------|--------------------|------------------|
| 22117 | 19/02/14 | 09:26 | CLIENTE CUATRO | 1.424.731,00 |
| 22116 | 19/02/14 | 08:24 | CLIENTE CATORCE | 2.736.326,00 |
| 22112 | 19/02/14 | 08:14 | CLIENTE CUATRO | 221.991,00 |
| 22111 | 19/02/14 | 08:13 | CLIENTE CINCO | 1.031.305,00 |
| 22110 | 19/02/14 | 00:12 | CLIENTE DIECISESIS | 3.001.350,00 |
| 22108 | 18/02/14 | 20:08 | CLIENTE CUATRO | 1.057.235,00 |
| 22107 | 18/02/14 | 19:49 | CLIENTE CUATRO | 203.533,00 |
| 22106 | 18/02/14 | 19:47 | CINCUESTA | 610.599,00 |
| 22105 | 18/02/14 | 10:53 | CLIENTE CATORCE | 1.057.235,00 |
| 22104 | 18/02/14 | 10:29 | CINCUESTA | 42.176,00 |
| 22103 | 18/02/14 | 10:27 | CINCUESTA | 203.533,00 |
| 22101 | 17/02/14 | 20:19 | CLIENTE CINCO | 20.260,00 |
| 22.090 | | | | 4.449.090.163,34 |

Mejora del cruce de componentes cuando una de las lista resultantes tiene un número mucho mayor de registros que el resto

Cuando en una búsqueda teníamos dos componentes que se cruzaban, y la lista resultante de la segunda componente tenía un número muchísimo mayor de elementos que la primera, se producía un error en la búsqueda que hacía que Velneo vClient V7 se cerrara.

A partir de la versión 7.15 se mejora el funcionamiento de las búsquedas y la gestión de las listas en los cruces, con lo que se evita el problema y podremos despreocuparnos del tamaño de las listas resultantes de las componentes de las búsquedas. (VBM-3700).

En una rejilla con totales suma acumulada, con una columna que muestra un campo puntero a maestro, cuando hay registros que contienen códigos inexistentes en el maestro, la rejilla detecta que no existe el registro y evita recalcular constantemente totales

En una rejilla con totales en los que se muestra la suma acumulada de columna donde una columna mostraba un campo puntero a maestro, y había un registro contenía códigos inexistentes en el maestro, la rejilla recalculaba constantemente los totales.

A partir de la versión 7.15, se evita esta circunstancia y la rejilla una vez realizado el cálculo de los totales no fuerza el recálculo ya que detecta que hay maestros inexistentes. (VBM-3705).

Se realiza el cierre de edición cuando editamos un registro en una rejilla editable y hacemos clic con el ratón en la zona vacía de registros de la rejilla

En una rejilla editable con pocos registros y que tuviera en la parte inferior espacio vacío (en blanco), bastaba editar un registro, mientras estábamos en edición hacíamos clic con el ratón en la zona vacía de la rejilla y veíamos como el registro en edición había perdido el foco y visualmente parece que se había cerrado la edición, pero la edición continuaba activa.

A partir de la versión 7.15 se mejora la rejilla editable, con nuevas funcionalidades y entre ellas veremos que cuando pulsemos en la zona vacía de registros se finalizará la edición, y se dispararán las señales correspondientes. Esto mejora la experiencia de usuario de las rejillas editables de nuestras aplicaciones y evita confusiones al usuario final. (VBM-3706).

La rejilla editable únicamente emite la señal de pierde foco cuando se pierde el foco y evitando su emisión cuando se entra en edición

En una rejilla editable, en el momento de entrar en edición, se emitía la señal de pérdida de foco, ya que realmente la rejilla perdía el foco.

A partir de la versión 7.15, se evita la emisión de la señal cada vez que entra en edición, emitiéndose únicamente cuando el control rejilla pierde el foco. Esto facilita la gestión de la pérdida de foco de la rejilla y el desarrollo de funcionalidades en que requieran del uso de esta señal. (VBM-3710).

La validación tomará el valor 0 como nulo en el caso de controles de edición de maestro numérico permitiendo continuar al siguiente control

La validación no funciona cuando un control de edición de maestro cuyo contenido es el campo código de un maestro y no apunta a ningún registro no permitiendo continuar al siguiente control

En un control de edición de maestro en que editábamos el maestro de una ficha por el campo código del maestro (Contenido: #MAESTRO o #MAESTRO.ID) se mostraba un 0 cuando no se apuntaba a ningún registro, pero la validación no lo tomaba como nulo y no permitía continuar al siguiente control.

A partir de la versión 7.15 el valor 0 será tomado como nulo en el caso de controles de edición de maestro numéricos (como en un enlace a maestro), de tal forma que el funcionamiento sea homogéneo, dando igual que contenga 0 o esté vacío para ser considerado como nulo y facilitando al usuario final la navegación por los controles del formulario. (VBM-3728).

Estabilidad en la programación de aplicaciones

La nueva versión Velneo V7 7.15 trae consigo muchas novedades relacionadas con el desarrollo de aplicaciones.

En este apartado encontraremos aquellas novedades relacionadas con usabilidad, nuevas funcionalidades, mejoras en inspectores y nuevos editores que encontraremos en el entorno de desarrollo Velneo vDevelop V7 y que nos permitirán disfrutar de la velocidad y estabilidad de la nueva versión de Velneo V7.

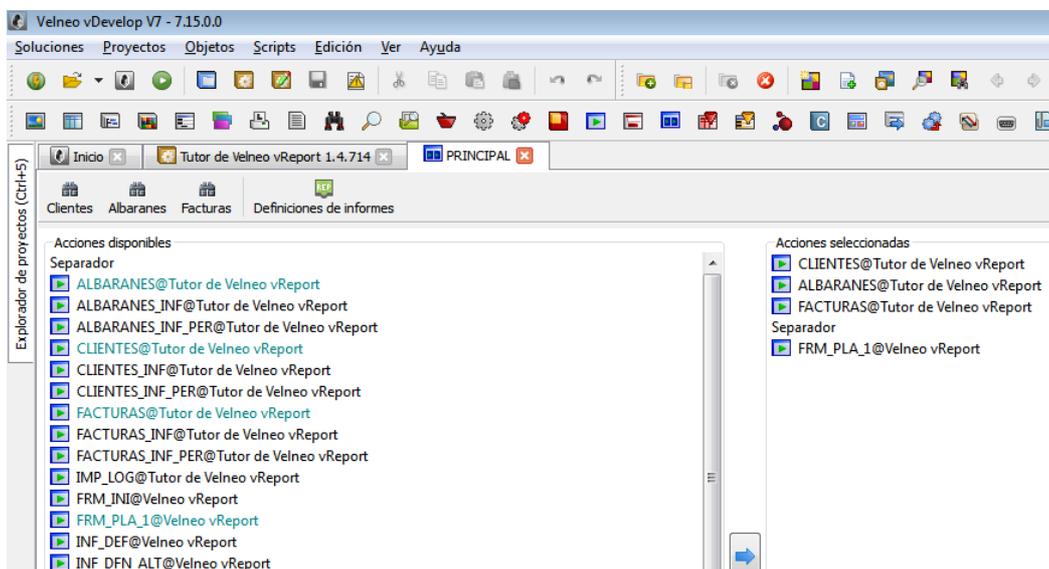
A continuación detallamos las mejoras más importantes que encontrarás:

Nuevos editores de menús y toolbars

En el editor de menús o toolbars tenemos una ventana en la que seleccionamos las acciones que queremos incluir en nuestro menú o toolbar y otra ventana en la que aparecen las acciones incluidas y el orden.

Cuando incluíamos una acción en el menú o toolbar en curso no teníamos forma de saber si esa opción ya había sido incluida o no, debíamos revisar manualmente las acciones incluidas.

A partir de la versión 7.15, las acciones que hayamos incluido en nuestro menú aparecerán marcadas en otro color, de tal forma que nos permite volver a usarla, pero podemos saber que ya ha sido incluida anteriormente en el menú o toolbar que estamos editando. De esta forma se facilita la edición de menús y toolbars, sin limitar las acciones que podemos incluir en ellas. (VBM-0837).



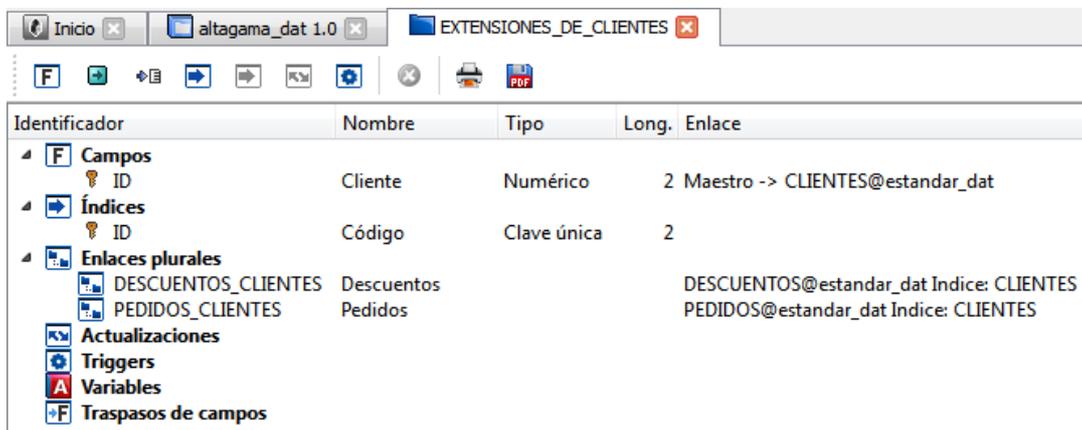
Creación de plurales heredados a tablas extendidas al cargar herencia

Creación de plurales heredados a tablas extendidas al cargar herencia

En una tabla maestra que habíamos extendido por medio de una tabla de extensión, cuando necesitábamos cargar los plurales correspondientes desde la tabla de extensión, teníamos primero que leer la ficha del maestro y, desde esta ficha, cargar los plurales correspondientes.

A partir de la versión 7.15 se implementa la creación de plurales heredados de tablas extendidas al cargar la herencia, de tal forma que tendremos disponibles estos plurales para su uso en nuestras aplicaciones.

Esto quiere decir que de forma automática se crean los plurales correspondientes al maestro en las tablas de extensión, con lo que podremos usarlos para navegar desde la tabla de extensión a los plurales de la tabla que hemos extendido sin tener que cargar previamente el registro de la tabla correspondiente, lo que facilitará nuestra tarea como desarrolladores. (VBM-3674).



| Identificador | Nombre | Tipo | Long. | Enlace |
|---|--|---|--|--|
| <ul style="list-style-type: none"> <ul style="list-style-type: none"> Campos <ul style="list-style-type: none"> ID Índices <ul style="list-style-type: none"> ID Enlaces plurales <ul style="list-style-type: none"> DESCUENTOS_CLIENTES PEDIDOS_CLIENTES Actualizaciones Triggers Variables Trasposos de campos | <ul style="list-style-type: none"> Cliente Código Descuentos Pedidos | <ul style="list-style-type: none"> Numérico Clave única | <ul style="list-style-type: none"> 2 2 | <ul style="list-style-type: none"> Maestro -> CLIENTES@estandar_dat DESCUENTOS@estandar_dat Indice: CLIENTES PEDIDOS@estandar_dat Indice: CLIENTES |

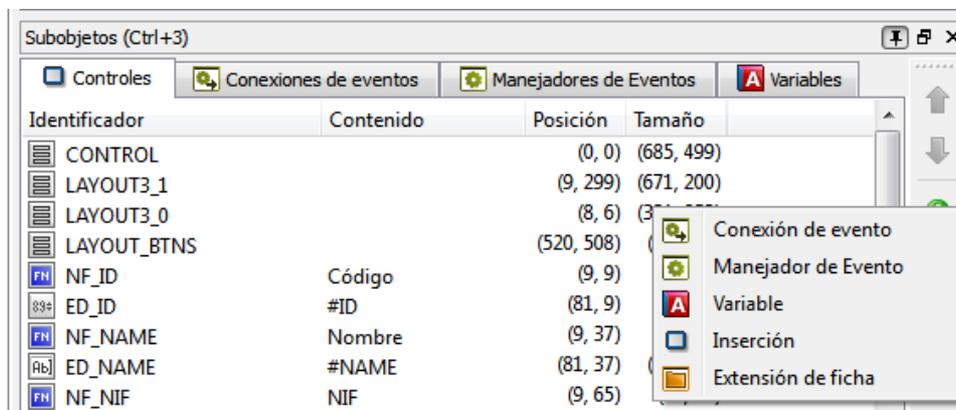
Nueva opción configurar en la toolbar del dock de subobjetos si se muestran texto e iconos o sólo los iconos en las pestañas

En las pestañas correspondientes a los subobjetos se presentaba tanto el icono como el nombre correspondiente al subobjeto que se incluye en la pestaña. Esto hacía que en los casos en los que hay gran número de tipos de subobjetos, pocos de ellos pudieran presentarse en un primer vistazo, haciendo necesario usar los botones para moverse entre pestañas del pestañador.

A partir de la versión 7.15 se implementa un botón en la toolbar del dock de subobjetos que permite configurar si se muestran texto e iconos o sólo iconos en las pestañas, lo que mejora la usabilidad del dock y facilita al desarrollador el acceso a los distintos subobjetos que tiene el objeto. (VBM-3480).



Se ordenan y homogeneiza la presentación de los subobjetos en la pestaña de subobjetos de todos los objetos, tanto en las pestañas como en el botón de añadir



Dependiendo del objeto, las pestañas y opciones para crear subobjetos en el mismo, aparecían en distintos órdenes.

A partir de la versión 7.15, el orden de los subobjetos seguirán el mismo orden, tanto en las pestañas como en botón para añadir subobjetos, siendo coherentes unos objetos con otros y facilitando al desarrollador la creación y mantenimiento de subobjetos. (VBM-3273).

El inspector de errores y el inspector dónde se usa controlan datos de tipo lista de fórmulas

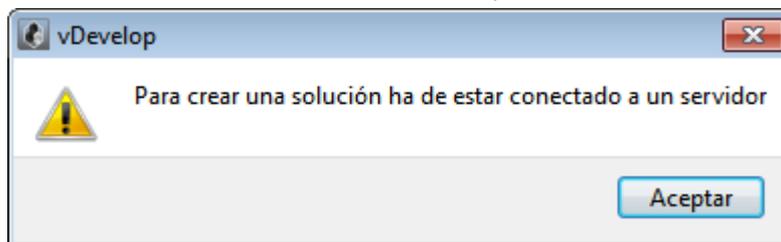
El inspector de errores y el inspector dónde se usa no controlaban datos de tipo lista de fórmulas como los usados al resolver un índice, un puntero indirecto, un componente de búsqueda, etc.

A partir de la versión 7.15 este tipo de dato se controla, con lo que ambos inspectores nos permitirán encontrar los objetos que se usan en este tipo de dato o errores que pueda haber, facilitando al desarrollador la programación. (VBM-3605).

El asistente de nueva solución solicita los datos de conexión si no nos hemos conectado previamente al servidor de edición

Cuando creábamos una solución con Velneo vDevelop V7, nos mostraba un asistente para la creación, pero hasta el último paso de creación, finalizado el asistente, no nos indicaba que no estábamos conectados al servidor, finalizando entonces el asistente.

A partir de la versión 7.15, antes de comenzar, el asistente ya comprueba si estamos conectados a un servidor de edición, con lo que no nos permitirá continuar antes de que nos hayamos conectado. (VBM-3563).



Se desactiva importar componentes hasta que no hay enganche a un servidor de edición

Cuando pulsábamos la opción de importar componentes no se comprobaba previamente que estuviésemos conectados a un servidor de desarrollo.

A partir de la versión 7.15, hasta el momento en el que nos conectemos a un servidor de desarrollo, no estará permitido usar la opción de importar componentes. (VBM-3568).

Más ancho inicial a los asistentes de nueva: Solución, Tabla, Rejilla y Formulario

Los asistentes de creación de Solución, Tabla, Rejilla y Formulario tenían un ancho inicial por defecto escaso, que no permitía ver la información adecuadamente y con un tamaño amplio de controles.

A partir de la versión 7.15 se han mejorado los asistentes, incrementando el ancho inicial por defecto con el fin de que se puedan ver de forma contempla los controles y facilite la introducción y modificación de datos, facilitando el proceso de creación de nuevos objetos al desarrollador. (VBM-3564).

El inspector de errores detecta si hay una rejilla que tenga asignada una tabla que no existe y tiene declaradas columnas con contenido de campos de la tabla

Velneo vDevelop V7 da error y se cierra al ejecutar el inspector de errores cuando tenemos una rejilla que tiene asignada una tabla que no existe y tiene declaradas columnas con contenido de campos de la tabla.

A partir de la version 7.15 se soluciona esta incidencia y se indica en el inspector de errores que la tabla no existe, informando al desarrollador del objeto afectado, lo que facilitará el desarrollo de nuestras aplicaciones y evitará el uso de rejillas sin tabla existente. (VBM-3694).

Al modificar el identificador de un control se actualiza la propiedad Control de EventoConnect cuando es el último o el único del identificador compuesto



Cuando modificábamos el identificador de un control que está siendo usado en la propiedad control de una conexión de evento, la propiedad control seguía mostrando el identificador antiguo, cuando este era el último o el único del identificador compuesto.

A partir de la versión 7.15 se controla el cambio de identificador en este caso, de esta forma no tenemos que preocuparnos por cambios en los identificadores, ya que en todos los casos se actualizarán de forma correcta, facilitando el desarrollo de nuestras aplicaciones. (VBM-3725).

Mejora en la visualización de la barra de progreso de la ventana de Importar solución En MacOSX

En sistemas operativos MacOSX, al importar una solución tras aceptar la solución a importar, cambiaba la ventana donde se muestra la barra de progreso, que dejaba de mostrar el avance.

A partir de la versión 7.15, estos problemas visuales se solucionan y la ventana con la barra de progreso se muestra correctamente, mejorando la experiencia de usuario del desarrollador. (VBM-1661).

El cambio del identificador de un índice que interviene en un enlace plural actualiza el identificador del plural

Al cambiar el identificador de un índice cambiaba bien la propiedad Índice de un objeto Plural, pero no cambiaba el identificador del propio plural que es la suma del identificador de la tabla y el del índice.

Esto mismo sucedía con las tablas, al cambiar el identificador de la tabla no se cambiaban los identificadores del plural.

A partir de la versión 7.15, cuando modifiquemos el identificador de un índice o una tabla que intervienen en un plural, su identificador será cambiado de forma automática, lo que facilitará al desarrollador el uso de los plurales evitando confusiones. (VBM-1826).

Mejora del editor de menús y toolbars en MacOSX

En MacOSX, cuando editábamos un objeto menú se producían problemas con la barra principal de menú de Velneo vDevelop V7

A partir de la versión 7.15 se ha rehecho todo el editor, simplificándolo, evitando el problema en MacOSX y además se han añadido funcionalidades como marcar en color distinto las acciones que ya se hayan incorporado al menú, con lo que se facilita el desarrollo de estos objetos. (VBM-2382).

Mejora de la edición de layouts en el editor de formularios en Mac

En sistemas operativos MacOSX, cuando en un formulario usábamos un dibujo como layout, tapaba los controles que contenía, lo que dificultaba la selección de los mismos.

A partir de la versión 7.15 mejora la integración con Mac, con lo que podremos usar un dibujo como layout, permitiendo la edición del resto de controles, facilitando el desarrollo de formularios. (VBM-2968).

API de Programación

En la nueva versión de Velneo V7 7.15 encontraremos nuevas clases y funciones en el API de Velneo para scripts, con las que podremos lanzar procesos, disparar objetos, etc., y que nos permitirán ampliar la funcionalidad de nuestras aplicaciones.

A continuación tenemos un detalle de las novedades más importantes que encontrarás en el API de Velneo y que te permitirán disfrutar de la velocidad y estabilidad de esta nueva versión:

Documentación mejorada

Hemos mejorado la documentación del API, incluyendo un árbol de las clases que te ayudarán a conocer la relaciones entre las clases, esto te ayuda a la hora de saber que funciones y propiedades tienes disponibles para ejecutar desde una clase, ya que se pueden ejecutar tanto las suyas propias como todas las de las clases heredadas. Hemos incluido también un enlace por si quieres ver la lista de clases por orden alfabético. También hemos incluido ejemplos en todas las nuevas clases y funciones incorporadas. Esta mejoras están disponibles tanto desde la ayuda online de vDevelop como desde el manual de la página web.

Velneo V7: JavaScript Class Library

Ver lista de clases

- **Funciones globales**
- **Objetos**
- **VApp**
- **VByteArray**
- **VDir**
- **VFile**
 - **VTextFile**
- **VFileInfo**
- **VImage**
- **VMainWindow**
- **VMdiView**
- **VObjectInfo**
- **VObjectInstance**
 - **VDataViewDialog**
 - **VFinder**
 - **VProcess**
 - **VQuery**
- **VProjectInfo**
- **VQmlListModel**
- **VRegister**
- **VRegisterList**
- **VRoot**
- **VSysInfo**
- **VTableInfo**
- **VXmlWriter**
- **Widget**
 - **VAbstractBrowser**
 - **VBoundFieldEditBrowser**
 - **VDataCatcherEditBrowser**
 - **VLineEditBrowser**
 - **VFilePathEditBrowser**
 - **VFormulaEditBrowser**

Nueva clase **VObjectInstance**

Esta nueva clase representa una instancia de un objeto en ejecución, como por ejemplo una búsqueda (VQuery) o un proceso (VProcess). Esta clase implementa las funciones comunes a todos los objetos como es el uso de las variables locales.

Las funciones de esta clase se usan desde los objetos creados por las clases que la heredan VDataViewDialog, VFinder, VProcess, y VQuery, en los ejemplos de estas clases podemos ver el uso de las funciones de VObjectInstance.

El API de Velneo V7 y los procesos JavaScript son ahora mucho más potentes y flexibles a la hora de integrarse con el resto de procesos y objetos de la aplicación. (VBM-3716)

Nueva clase VDataViewDialog

La clase VDataViewDialog representa un objeto en ejecución de ficha o lista con interfaz. Nos permite ejecutar objetos de lista o ficha como rejillas, formularios, casilleros, etc. Construimos un objeto VDataViewDialog pasándole el objeto VRoot en curso y le indicamos el objeto a ejecutar con la función setDataView.

Una vez inicializado le pasamos un registro (setRegister) o una lista de registros (setRegisterList) en función del tipo de entrada del objeto y también el contenido a las variables locales del proceso con setVarLocal (si lo requiere) y la ejecutamos con la función exec.

El registro que se pasa al objeto si exec devuelve true contendrá la información actualizada tras haber sido procesado en el objeto. No se genera un nuevo registro de salida (result), sino que se trabaja sobre el mismo VRegister o VRegisterList de entrada.

Esta clase nos abre la posibilidad de interactuar con objetos de interfaz de vista de datos de ficha y lista y gestionarlos directamente desde nuestro proceso JavaScript. Esto supone una mayor sencillez en nuestra programación a la vez que nos otorga una mayor potencia a la hora de crear soluciones más flexibles. (VBM-3722).

```
// -----
// Abrimos un formulario al que le pasamos parámetros y del que recuperamos valores de retorno
// -----

//Manejamos una lista de registros
var Registros = new VRegisterList(theRoot);
Registros.setTable("49z6dcke.vcd/DUMMY");
Registros.load("ID", [1]);

if (Registros.listSize())
{
    //Nos quedamos con un registro
    Registro = Registros.readAt(0);

    //Creamos el formulario
    var formulario = new VDataViewDialog(theRoot);
    formulario.setDataView(VObjectInfo.TypeForm, "49z6dckf.vca/DUMMY")

    //Le pasamos un valor a la variable local del formulario (como manejador de objetos)
    formulario.setVar("TEXTO", "Hola mundo");
    formulario.setRegister(Registro);

    //Mostramos el formulario
    if(formulario.exec())
    {
        //Leemos el valor de una variable del formulario y lo mostramos
        alert(formulario.varToString("RESULTADO"));
    };
};
```

Nueva clase VFinder

La clase VFinder representa un localizador en ejecución. Nos permite ejecutar el objeto de proyecto localizar. Construimos un objeto VFinder pasándole el objeto VRoot en curso y le indicamos el localizador a ejecutar con la función setFinder. Tras ejecutar la función exec podemos recoger la ficha localizada con la función result, que nos devuelve un objeto de la clase VRegister con el cual podemos operar. (VBM-3712).

```
// -----
// Ejemplo de localizador multiempresa
// Localizador subindexado a un parámetro que le pasemos a la variable local que lo subindexa
// -----

//Importamos la clase
importClass("VFinder");

//Creamos el localizador
var Finder = new VFinder(theRoot);
Finder.setFinder("49z6dckf.vca/VFINDER");

//Le indicamos la subindexación(por ejemplo para multiempresa) sin el uso de variables globales
Finder.setVar("TIP", 1);

//Mostramos el localizador
if(Finder.exec())
{
    //Si se obtienen resultados
    if(Finder.result()){

        //Mostramos el formulario seleccionado
        var formulario = new VDataViewDialog(theRoot);
        formulario.setDataView(VObjectInfo.TypeForm, "49z6dckf.vca/DUMMY");
        formulario.setRegister(Finder.result());
        formulario.exec();
    };
};
```

Nueva clase VProcess

Hasta esta versión era posible ejecutar procesos JavaScript lanzados desde código V7 pasándole parámetros a través de variables locales. Ahora ya es posible lanzar procesos V7 y JavaScript desde un proceso JavaScript pasando parámetros a través de variables locales, al igual que hacemos en V7 con disparar objeto contra un manejador de objeto. Además, esta clase nos permite ejecutar los procesos en 1º, 2º y 3º plano lo que facilita la optimización de nuestras aplicaciones para conseguir mayor velocidad de ejecución.

La clase VProcess representa un proceso en ejecución. Nos permite ejecutar procesos y recoger el resultado.

Esta nueva clase nos permitirá ejecutar procesos desde procesos javascript de forma análoga a como hacemos con los comandos del grupo de Objeto, de Velneo V7. De esta forma podremos encadenar procesos javascript de forma sencilla, optimizando y reutilizando nuestro código. (VBM-3715).

```

// -----
// Proceso único que calcula el precio del artículo en base a los parámetros recibidos
// -----

//Se crea el manejador de objeto del proceso
var procesoCalculoPrecio = new VProcess(theRoot);

//Se alimenta con las variables necesarias
procesoCalculoPrecio.setProcess("gestion.dat/CAL_PRE_ART");
procesoCalculoPrecio.setVar("PRM_TRF_TIP", 2);
procesoCalculoPrecio.setVar("PRM_FCH", currentDate());
procesoCalculoPrecio.setVar("PRM_CLF", CLF);
procesoCalculoPrecio.setVar("PRM_ENT_GES", ENT_GES);
procesoCalculoPrecio.setVar("PRM_ART", ART);
procesoCalculoPrecio.setVar("PRM_CNT", 1);

//Se dispara el proceso con los parametros indicados
if (procesoCalculoPrecio.exec())
{
    //Leemos las variables de los resultados de la ejecución de l proceso
    var precioPRE = procesoCalculoPrecio.varToDouble("PRE");
    var precioDTO_1 = procesoCalculoPrecio.varToDouble("DTO_1");
    var precioDTO_2 = procesoCalculoPrecio.varToDouble("DTO_2");
    var precioDTO_3 = procesoCalculoPrecio.varToDouble("DTO_3");
    var precioDTO_4 = procesoCalculoPrecio.varToDouble("DTO_4");
    var precioDTO_5 = procesoCalculoPrecio.varToDouble("DTO_5");
}
else
{
    var precioPRE = 0;
    var precioDTO_1 = 0;
    var precioDTO_2 = 0;
    var precioDTO_3 = 0;
    var precioDTO_4 = 0;
    var precioDTO_5 = 0;
};

```

Clase VQuery revisada

Aunque la clase VQuery ya existía, se ha revisado para que herede de la clase VObjectInstance y tenga un funcionamiento homogéneo con las nuevas clases que la heredan. Este cambio no afecta al código JavaScript existente que use la clase, ya que las funciones se siguen llamando igual.

```
// -----
// Lanzar una búsqueda con los datos solicitados desde un formulario
// -----

importClass("VQuery");

//Creamos el formulario
var formulario = new VDataViewDialog(theRoot);
formulario.setDataView(VObjectInfo.TypeForm, "49z6dckf.vca/DUMMY_PEDIR_DATOS")

//Mostramos el formulario
if(formulario.exec())
{
    //Leemos el valor de las fechas que introducimos en el formulario
    fechaDesde = formulario.varToDate("FCH_DES");
    fechaHasta = formulario.varToDate("FCH_HAS");

    //Si los datos no son vacíos
    if ((fechaDesde != null) && (fechaHasta != null))
    {
        //Creamos el manejador de la búsqueda
        var busqueda = new VQuery(theRoot);

        //Asignamos la búsqueda a utilizar
        busqueda.setQuery("49z6dckf.vca/VQUERY");

        //Asignamos los valores adquiridos previamente
        busqueda.setVar("FCH_DES", fechaDesde);
        busqueda.setVar("FCH_HAS", fechaHasta);

        //Ejecutamos la búsqueda
        if(busqueda.exec())
        {
            //Retornamos la lista de resultados
            theRegisterListOut.append(busqueda.result());
        };
    };
};
```

Nueva clases para controles de tipo browser

En esta versión hemos añadido nuevas clases que nos permite gestionar los controles de edición de tipo browser, que son aquellos a los que les hemos declarado un menú de botón. Las nuevas clases son:

- VBoundFieldEditBrowser : Control de edición de campo enlazado a tabla maestra con botón de menú. Esta clase implementa la misma funcionalidad que la clase VBoundFieldEdit más las propiedades y funciones heredadas de la clase VAbstractBrowser.
- VDataCatcherEditBrowser : Control de edición data catcher edit con botón de menú. Esta clase implementa la misma funcionalidad que la clase VDataCatcherEdit más las propiedades y funciones heredadas de la clase VAbstractBrowser.
- VDateEditBrowser : Control de edición de fecha con doble botón arriba/abajo y botón de menú. Esta clase implementa la misma funcionalidad que la clase VDateEdit más las propiedades y funciones heredadas de la

clase VAbstractBrowser.

- VDateTimeEditBrowser : Control de edición de fecha/hora con doble botón arriba/abajo y botón de menú. Esta clase implementa la misma funcionalidad que la clase VDateTmeEdit más las propiedades y funciones heredadas de la clase VAbstractBrowser.
- VNumberSpinBoxBrowser : Control de edición de números con doble boton arriba/abajo y botón de menú. Esta clase implementa la misma funcionalidad que la clase VNumberSpinBox más las propiedades y funciones heredadas de la clase VAbstractBrowser.
- VTimeEditBrowser : Control de edición de hora/minutos/segundos con doble botón arriba/abajo y botón de menú. Esta clase implementa la misma funcionalidad que la clase VTimeEdit más las propiedades y funciones heredadas de la clase VAbstractBrowser.

Mejorado el API de controles browser para hacer funcionales todas sus propiedades y funciones

Hemos mejorado el API para que sean funcionales todas las propiedades y funciones de los controles tipo browser, con botón de menú. (VBM-3704, VBM-3711).

Los clases mejoradas han sido:

- VBoundFieldEditBrowser
- VCWebView
- VDataCatcherEditBrowser
- VDateEditBrowser
- VDateTimeEditBrowser
- VImageEdit
- VLineEditBrowser
- VNumberSpinBoxBrowser
- VTextEdit
- VTimeEditBrowser

Nueva función parentDataView() de la clase VAbstractDataView

Hasta ahora podíamos acceder a los objetos en cascada navegando por ellos de arriba hacia abajo, es decir, desde el principal o padre hacia sus hijos. También disponemos de la función mainForm() de la clase VFormDataView que nos permitía situarnos en el objeto padre o principal de la cadena de objetos amalgamos donde nos encontramos. Pero con esta nueva función tenemos ya la posibilidad de navegar hacia arriba desde el objeto en el que estamos hacia su padre y así recursivamente hasta llegar al padre u objeto principal.

Si nuestro objeto no se encuentra amalgamado dentro de otro objeto esta función devolverá null.

Si nuestro objeto está amalgamado esta función devolverá un objeto de la clase VAbstractDataView que representa al objeto dentro del que se encuentra amalgamado.

Esta función puede aplicarse recursivamente para movernos hacia arriba en la cadena de objetos amalgamados hasta encontrar un null que nos indique que ya estamos en el objeto de nivel superior de la cadena.

```
// -----
// Analizar la vista actual para saber si está amalgamada dentro de otras vistas
// -----

// Se busca una vista cuyo padre sea nulo, ya nos indica que es la padre o principal
var sendaVistas = "";
var vista = theRoot.dataView();

while(vista != null)
{
    sendaVistas += vista + "\n";
    vista = vista.parentDataView();
};

// Se muestra la lista de vistas en orden ascendente: el 1º es el objeto de partida el último el principal
alert(sendaVistas);
```

Nueva función deleteRegisterWhithoutDeupdating() de la clase VRegister

Gracias a esta nueva función ahora podemos eliminar un registro con y sin desactualización. (VBM-3576).

Implementación de la función root() en la clase VAbstractDataView

Se implementa la función root() en la clase VAbstractDataView que nos permite obtener un objeto de la clase VRoot correspondiente al objeto en ejecución.

A partir de la versión 7.15 podremos usarlo en nuestros procesos javascript de forma análoga a como hacíamos a cuando obteníamos el root en la clase VMdiView, ampliando las funcionalidades de nuestra aplicación. (VBM-3611).

Mejorada la función control() de la clase VFormDataView

La función control() mejorada de la clase VFormDataView ahora es capaz de devolver el control incluso cuando tipo un botón de menú (browser). (VBM-3701).

Optimizada la gestión de memoria de la clase VImage

Hemos optimizado el funcionamiento de la clase VImage. Podrás trabajar con múltiples objetos imagen sin que esto afecte al rendimiento y consumo de memoria del equipo. (VBM-3680).

Implementada la función staticTableItemImage() de la clase VApp

Hemos resuelto una incidencia que producía que la función staticTableItemImage() de la clase VApp se mostrase el error:

Error en script: TypeError: cannot call staticTableImage():Unknown return type `const QObject**` (register de the type with qScriptRegistermetaType()) (VBM-3671).

Base de datos y servidor

En esta nueva versión de Velneo V7 7.15 hemos mejorado la velocidad y estabilidad del servidor y la base de datos, trabajando para ello en muchas áreas: aumentando capacidad, dotándolo de nuevas funcionalidades, simplificando la gestión de instancias, mejorando la gestión de recursos y la integración con el sistema, etc.

Con ello hemos conseguido que Velneo vServer V7 nos permita disfrutar de la edición y ejecución de nuestras aplicaciones. A continuación vamos a ver las novedades más importantes:

Aumento del tamaño máximo de un objeto en el contenedor

El tamaño máximo de un objeto en el contenedor era de 256 MB.

A partir de la versión 7.15 el tamaño máximo de un objeto en el contenedor se amplía, llegando este límite a 512 MB. Debemos tener en cuenta que tanto para introducir el objeto como para exportarlo, el servidor requerirá 512 MB libres de memoria hasta los 2GB que tiene como máximo para su uso en procesadores de 32 bits windows/linux o 4GB en Linux 64 bits. (VBM-3570).

Implementación del control vía línea de comandos de un rango de puertos TCP válidos para iniciar la escucha en tercer plano

Se ha implementado un sistema de control de puertos abiertos en el servidor, que limita el rango de puertos de escucha en que podemos iniciar un objeto protocolo en el servidor. (VBM-3583).

Cuando iniciamos el servidor con los siguientes parámetros configurados, éste será capaz de iniciar protocolos que usen como puerto de escucha, el rango especificado en los parámetros:

apFirst: Número de puerto mínimo
apLast: Número de puerto máximo

Ej:

/apFirst=6900 /apLast=7000

El entretenedor en procesos ha sido ampliado

El entretenedor de procesos es el sistema que permite mantener viva la ejecución de un proceso por el que se está esperando la finalización y del proceso llamador que recibirá la respuesta o salida del proceso.

A partir de la versión 7.15, Se ha ampliado la aplicación de este entretenedor a los comandos de alta, baja o modificación, y estará operativo en triggers, subprocessos, etc., con lo que podremos ejecutar procesos sin límite de tiempo para la ejecución. (VBM-3585).

Mejora de la finalización y parada de Velneo vServer V7

Cuando parábamos un servidor, al eliminar instancias de aplicación no se eliminaba el ejecutor interno de la instancia aunque se quitaba de la lista de instancias.

A partir de la versión 7.15, se incluye el proceso de eliminar el ejecutor de la instancia en el momento en que eliminamos una instancia, con lo que al finalizar el servicio, el servidor realiza siempre la parada de forma correcta. (VBM-3573).

Mejora en reinicio de solución: ahora reinicia los proyectos por orden de herencia, de abajo a arriba

Cuando reiniciábamos una solución, el orden de reinicio dependía del orden físico de creación y no del orden jerárquico de herencia.

A partir de la versión 7.15, el orden de reinicio será el que determine herencia, con lo que será más eficiente. (VBM-3569).

Mejorada la gestión de ficheros abiertos en Velneo vServer V7 para Linux

En sistemas operativos Linux, el uso de la librería libc de GLib para la gestión de ficheros determina que no se puede superar el límite de 1024 ficheros abiertos debido a un bug que corrompe la memoria de stack de los sockets cuando se supera este límite.

<https://bugreports.qt-project.org/browse/QTBUG-24247>

<https://bugs.launchpad.net/ubuntu/+source/glibc/+bug/386558>

Para evitar esta problemática, el servidor cerraba ficheros a medida que no se usaban, pero en servidores con mucha carga este límite se podía llegar a alcanzar en determinadas circunstancias.

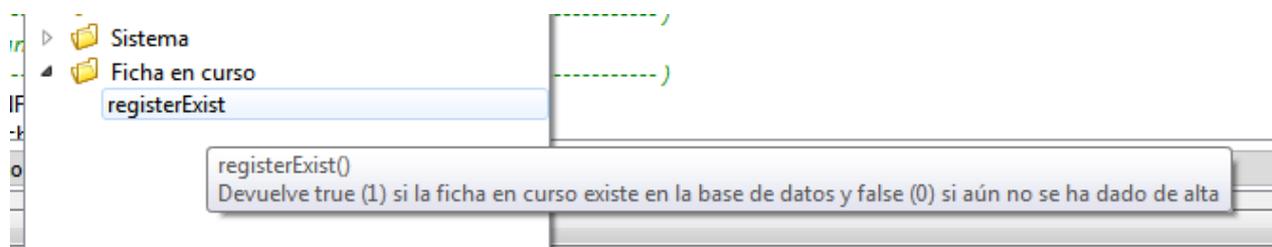
A partir de la versión 7.15 disfrutaremos de una serie de mejoras y optimizaciones de tal forma que el cierre de ficheros se produce en mayor número y a mayor velocidad, con esto conseguimos que el servidor haya de sufrir una carga extrema de trabajo para que pueda llegar a ese límite. Además, en el caso de que el servidor detecte que no puede abrir más ficheros, realiza un cierre de ficheros reiterativo hasta que le es posible continuar operando.

Conseguimos de esta manera evitar el bug existente en la librería libc de GLib y permitir que el servidor trabaje de forma eficiente y rápida, aún con gran número de ficheros en disco. (VBM-3709).

Nueva función para consultar si un registro está dado de alta o no

A partir de la versión 7.15 disponemos de la nueva función de calculador registerExist() del grupo Ficha en curso que nos indicará si el registro en curso ya está dado de alta y que podremos usar en cualquier fórmula de Velneo V7.

Con ayuda de esta función podremos ampliar la funcionalidad de nuestros procesos y formularios, teniendo información en cada momento de si la ficha existe. (VBM-3402).



Generador de informes

En esta nueva versión de Velneo vReport hemos trabajado en simplificar su uso con el fin, por un lado, hacer más sencilla su implementación, y por otro lado, evitar modificaciones incorrectas de informes o perder modificaciones realizadas.

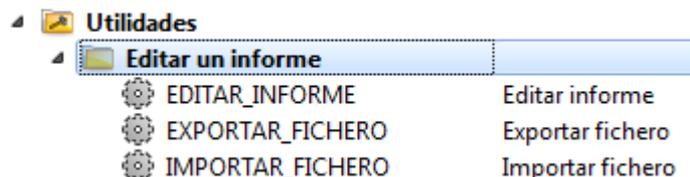
Las mejoras incluidas han sido también añadidas en el tutor de Velneo vReport, que incluye ejemplos de éstas en acción.

Las principales novedades de esta versión son:

- Novedades en formulario de Definición de informes (INF_DFN)
 - Mejora de los procesos de guardado y modificación de informes (Formulario INF_DFN, Manejadores INF_EDIT_BTN). De esta forma evitamos modificaciones incorrectas o perder modificaciones realizadas.
 - Nuevas opciones en definiciones de informe:
 - Duplicar informe (INF_DUP)
- Nuevos objetos para utilidades

Editar un informe

En la carpeta correspondiente, encontramos una serie de procesos con origen INF_DFN, la tabla de Definiciones de informes, que nos permiten de forma sencilla lanzar el editor de informes correspondiente, exportar el fichero de informes o importarlo.

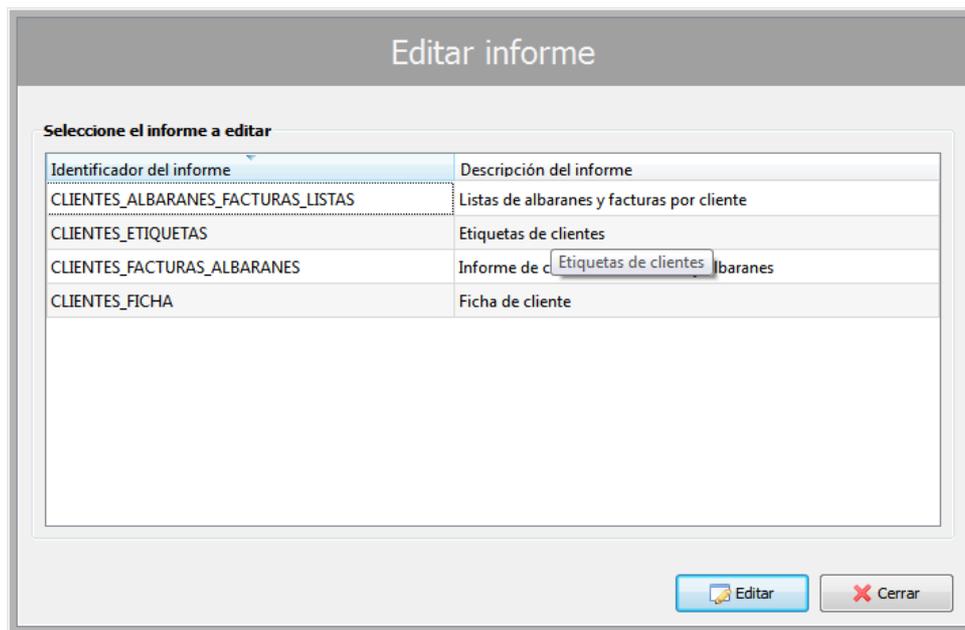


Seleccionar informe a editar

El formulario incluido permite seleccionar de forma sencilla un informe de entre los existentes en el repositorio.

Está pensado para utilizar los identificadores únicos de los informes, buscando los que se correspondan con cierto prefijo.

Por ejemplo, los informes de la tabla clientes, pueden llevar el prefijo CLIENTES, con la que la búsqueda se realizará indicando éste al proceso, que se encargará de mostrar al usuario los resultados coincidentes y mostrar el editor de informes con el informe que hayamos seleccionado.

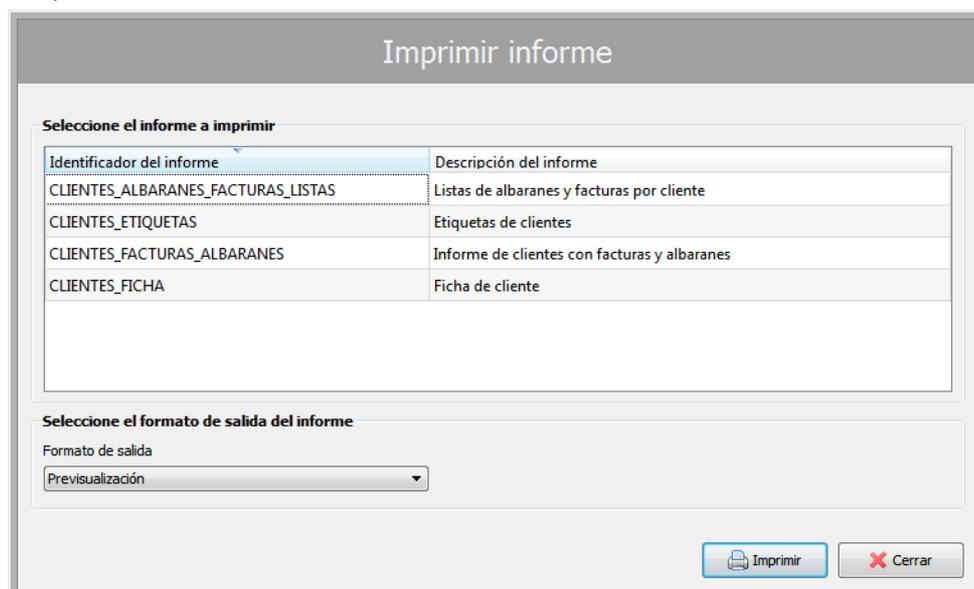


Seleccionar informe a imprimir

Los objetos incluidos permiten seleccionar de forma sencilla un informe de entre los existentes en el repositorio y solicitar los datos necesarios para realizar esta tarea.

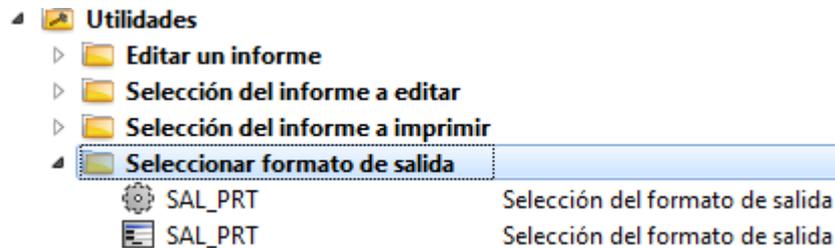
Debemos hacer uso del proceso SELECCION_INFORME_IMPRIMIR, pasando los valores adecuados para que el formulario que presenta permita realizar la selección de informe.

Está pensado para utilizar los identificadores únicos de los informes, buscando los que se correspondan con cierto prefijo. Por ejemplo, los informes de la tabla clientes, pueden llevar el prefijo CLIENTES, con la que la búsqueda se realizará indicando éste al proceso para realizar la búsqueda, que se encargará de mostrar al usuario los resultados coincidentes e imprimir los datos usando el informe seleccionado.



Seleccionar salida de impresión

Esta pequeña utilidad permite mostrar al usuario las salidas de impresión existentes y retorna el número de la salida seleccionada, según una tabla estática.



Llamando al proceso SAL_PRT nos mostrará un formulario para seleccionar el tipo de salida (y la senda en el caso de que la salida sea una exportación a disco), que guardará en variables que podremos leer después de usar el proceso con los comandos de objeto.

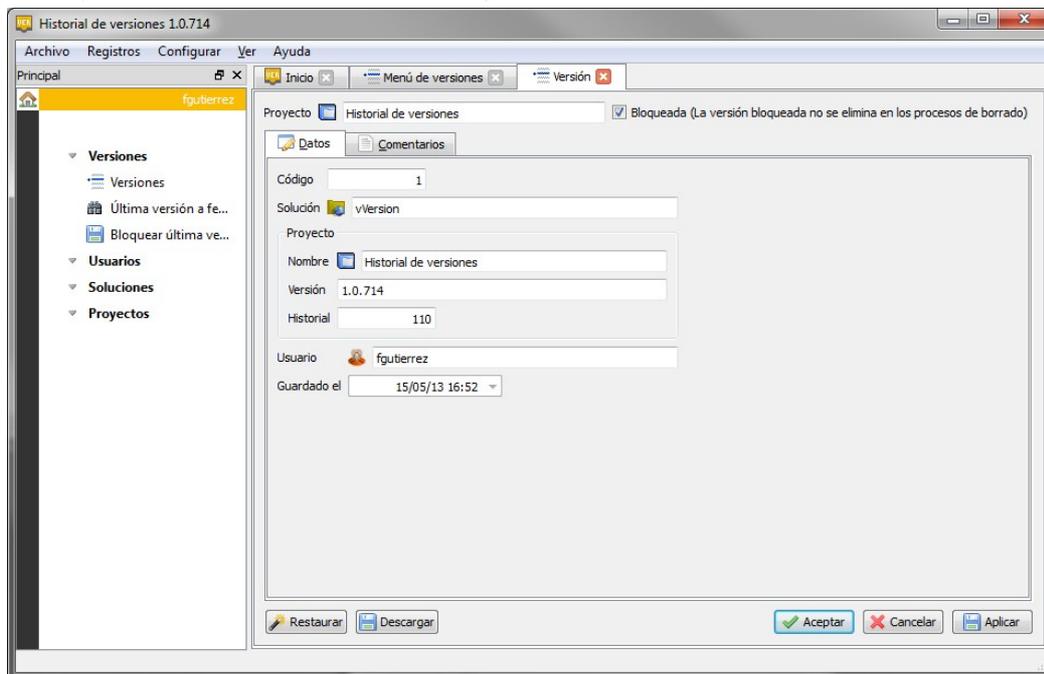
Otras novedades

En la nueva versión de Velneo V7 7.15 nos hemos ocupado también de mejorar los elementos que acompañan a la plataforma: la aplicación de historial de versiones, los instaladores, y otros elementos que son necesarios para que disfrutes de Velneo V7..

Mejora en la numeración de historial de un proyecto en la recuperación de éste con vVersion, Historial de Versiones

Cuando recuperábamos un proyecto con Velneo vVersion en el servidor, se recuperaba con el mismo número de historial con el que se había guardado, pudiendo producir confusiones.

A partir de la versión 7.15, con Velneo vVersion 1.4.715, cuando recuperamos el proyecto se guardará con un número de historial superior al existente ya en el servidor, con lo que no se producirán confusiones en cuanto a la versión en curso del proyecto, facilitando al desarrollador su gestión. (VBM-3719).



Actualización de certificado de firma digital para instalaciones de Windows

La firma realizada con este certificado indica y garantiza la procedencia del instalador descargado y de los ficheros ejecutables que se incluyen. Además, certifica que no han sido alterados, con lo el usuario puede confiar en su contenido. (Más información: <http://support.microsoft.com/kb/206637/es>)

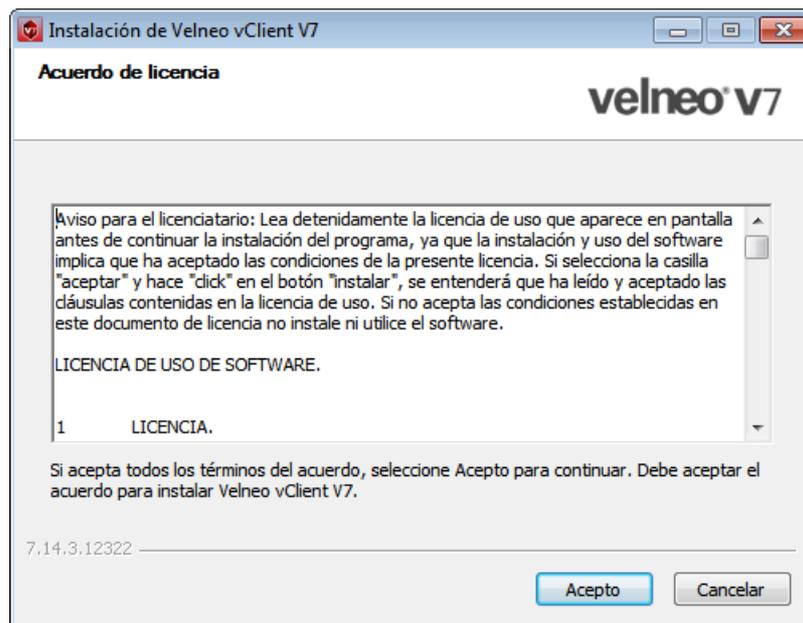
Las firmas digitales y los certificados de autenticidad se aplican a todos los instaladores y a todos los programas ejecutables y a partir de la versión 7.15 se actualiza para que sea funcional en más sistemas operativos Windows.

Gracias a este certificado el usuario puede confiar en que los ficheros de instalación descargados provienen de Velneo y no han sido alterados. (VBM-3733)

Incorporación de la traducción al inglés de los términos de licencia en el instalable de Velneo vClient V7 para Windows

El instalador en Windows mostraba los términos de la licencia en español, aunque la instalación se realizara en inglés.

A partir de la versión 7.15, los términos de licencia de la versión 7.15 se mostrarán en el idioma correspondiente a la instalación, que depende en último término del idioma del sistema operativo, facilitando al usuario final la instalación de los componentes en Windows. (VBM-3269).



Open Apps

Gestión de contactos de manera sencilla.

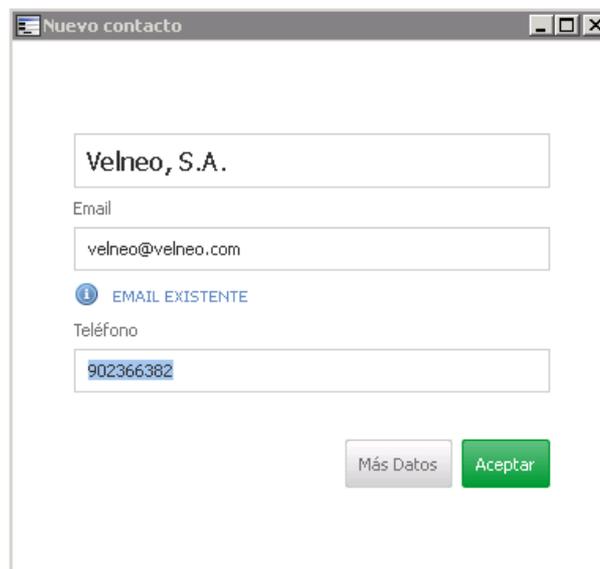
Tras el feedback de bastantes usuarios finales de la anterior versión de vBase y diversos test de usabilidad, llegamos a la conclusión de que era imprescindible llevar a cabo una revisión de la experiencia de usuario en la gestión de contactos empresariales, principal funcionalidad en esta Open App.

Por ese motivo se lanzó durante la iteración 7.15 de Velneo una nueva versión de vBase en la cual, sin variar la estructura de tablas y herencias, para facilitar la actualización a la nueva versión, sí se ha reenfocado desde cero la interfaz para la gestión de lo que antes llamábamos “Entidades” y ahora llamamos “Contactos” (Esta es la primera gran novedad de la versión, en la investigación y pruebas con usuarios hemos verificado que el término “contacto” es más estándar y se comprende mejor).

Queremos agradecer también a los miembros de la comunidad que participasteis como betatesters y nos enviasteis comentarios de mejora, gracias a ellos se han corregido o modificado varios aspectos de la Open App.

La principal novedad y en la que más se ha trabajado es el proceso de alta y modificación de contacto, la interfaz se ejecuta por niveles, desde los datos más básicos de un contacto hasta el máximo nivel de detalle.

Alta simple



Nuevo contacto

Velneo, S.A.

Email

velneo@velneo.com

EMAIL EXISTENTE

Teléfono

902366382

Más Datos Aceptar

Edición básica

Nombre Fiscal

NIF

Email

Teléfono

Dirección

País

Contacto Relacionado

Código **14365**

Edición "Más datos"

Datos Principales

Emails (2) Teléfonos (1) Direcciones (4) Relaciones (28) Otros

+ - * ✉

| Email | Descripción |
|-------------------|----------------------|
| velneo@velneo.com | * Email |
| admin@velneo.com | Email administración |

+ - *

| Teléfono | Descripción |
|-----------|-------------|
| 902366382 | * Teléfono |

Aceptar Cancelar

La nueva versión de vBase ya está disponible para su descarga en la zona de [Open Apps](#) de la web, así como un documento más detallado de las principales novedades de la versión e indicaciones sobre su instalación.

Cómo personalizar aplicaciones en ejecución

Las aplicaciones estándar tienen la ventaja de que se programan una vez para todos los clientes y el inconveniente de que no están ajustadas a las necesidades específicas de cada uno. Las aplicaciones a medida, justo lo contrario, se ajustan a las necesidades específicas pero eso obliga a mantener un programa por cliente.

El sueño de cualquier desarrollador o empresa de programación es disponer de un programa estándar que pueda mantener una única vez y que sin tocar ese código fuente pueda tener las adaptaciones específicas para cada uno de sus clientes. De esta forma podrías ofrecer unos productos y servicios de gran calidad a un precio ajustado.

¿Te gusta la idea pero crees que es un utopía?

Velneo V7 en cada versión ha mejorado añadiendo nuevas posibilidades para acercarte al sueño de conseguir alcanzar esa utopía.

Para conseguir que tus aplicaciones sean altamente personalizables y configurables Velneo V7 nos provee de las siguientes herramientas:

- Herencia de proyectos para reutilizar el código.
- Puntos de inserción para añadir opciones de menús y subformularios sin tocar el código fuente.
- Tablas de extensión para añadir campos a tablas del núcleo estándar sin modificar su estructura.
- Índices complejos para generar índices de tablas sin tocar su estructura o combinando campos de diferentes tablas incluyendo las que están en el estándar.
- API de Velneo que nos permite acceder a todos los objetos y subobjetos de la aplicación tanto para conocer su definición como para controlarlos en tiempo de ejecución.

La combinación de todas estas herramientas nos abre la puerta al desarrollo de un núcleo estándar que pueda ser personalizado específicamente para cada cliente.

Además, de las funcionalidades programadas específicamente para cliente existen una serie de configuraciones y personalizaciones que se hacen a nivel de interfaz para cada cliente o usuario. Gracias a las nuevas funciones disponibles en el API de Velneo podemos de forma rápida y sencilla conseguir que los programadores, implantadores e incluso el usuario pueda personalizar sus rejillas, formularios y menús.

El ejemplo

Te recomiendo que veas el [vídeo de la presentación realizada por Jesús Arboleya en Life is Soft 2014](#) donde realiza la personalización de la aplicación en tiempo de ejecución sin tocar el código fuente del estándar.

Formularios y rejillas personalizadas

A nivel de personalización de la interfaz podemos configurar diferentes acciones: ocultar y desactivar controles, añadir, quitar y sustituir subformulario.

Personalización

Nombre: 08.1. Ocultar código en rejilla de artículos

Tipo de personalización: **Ocultar controles**

Proyecto: [dropdown]

Tipo de objeto: [dropdown]

Objeto: [dropdown]

Subobjetos:

| Id | Nombre |
|--|--------------|
| <input checked="" type="checkbox"/> ID | Código |
| FOTO | Foto |
| NOMBRE | Nombre |
| FAMILIA | Familia |
| PRECIO | Precio |
| PDTE_SERVIR | Pdte. servir |

Todos los usuarios
Sólo a estos usuarios: [text area]

Buttons: Aceptar, Cancelar, Opciones

Incluso ejecutar un script, lo que nos abre las puertas a hacer prácticamente de todo en la personalización de un formulario o rejilla, sin tener que tocar su código fuente o cambiar sus propiedades. Y todo en tiempo de ejecución.

Personalización

Nombre: 11.1. Al crear un nuevo pedido generarlo con el cliente y artículo seleccionado

Tipo de personalización: **Ejecutar script**

Proyecto: estandar_app

Objeto: PEDIDO (Pedido)

Script:

```
// Al crear un nuevo pedido generarlo con el cliente y artículo seleccionado
importClass("\VFinder");
importClass("\VProcess");

// -----
// Controlar que sea un alta
// -----
if( theRegisterIn.exist() == false)
{
    var formulario = theRoot.dataView();

    // Crea localizador de cliente
    var localizadorCliente = new VFinder(theRoot);
    localizadorCliente.setFinder("estandar_app/CLIENTES");
}
```

Todos los usuarios
Sólo a estos usuarios: @invitado

Buttons: Aceptar, Cancelar, Opciones

Programar la personalización

El otro aspecto importante de la personalización de aplicaciones en ejecución con Velneo V7 es lo realmente sencillo que resulta, ya que solo necesitamos incluir un manejador de evento JavaScript con estas 2 líneas en cada objeto que queramos que sea personalizable.

```
1 #include "48c9d5e5.vca/js/personalizar.js"  
2 personalizar();
```

Y ejecutarlo en el evento post-inicializado de nuestros objetos formulario y rejilla, o en el momento que nos interese.

Este script es siempre igual para todos los objetos y no requiere cambiar nada, ya que la función no tiene parámetros, lo que permite programarlo a toda velocidad incluso en las aplicaciones que ya tengas programadas.

Hay que destacar que el uso de esta personalización no supone pérdida de rendimiento en la ejecución de tus aplicaciones ya que el código script de la personalización lo ejecuta Velneo V7 a toda velocidad.

Seminario sobre personalización de aplicaciones en ejecución y vERP

Tras la salida de la versión 7.15 realizaremos un seminario online el próximo 24 de abril en el que se explicará con detalle cómo puedes aplicar esta personalización en ejecución a tus aplicaciones y a los suscriptores de vERP les entregaremos el script para que puedan aplicarlo y disfrutarlo.

Ahorra muchas líneas de código y programa a toda velocidad aplicando la personalización de interfaz y menú de Velneo V7 en tus aplicaciones.

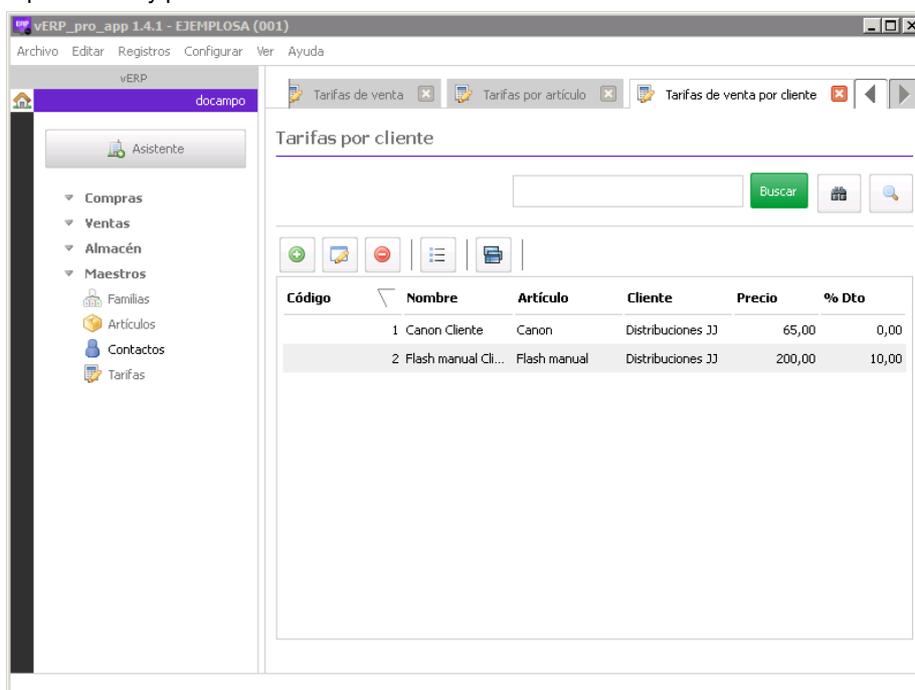
Software Erp de código abierto pymes, versión 1.4

Velneo vERP quiere ser la referencia en aplicaciones estables y de código abierto para la comunidad, por ello la apuesta total por este ambicioso proyecto.

A lo largo de 2014 seguiremos liberando versiones del producto, dotando cada vez de más velocidad y optimización, con las que obtener el mayor rendimiento de la base de datos Velneo. Para lo cual vamos a hacer un repaso por la trayectoria de los últimos meses:

[Versión 1.4](#)

- Tarifas de venta por cliente y por artículo



Tarifas de venta fijas para cliente, y configuración de múltiples tarifas de venta por artículo, siendo aplicables a lo largo de todos los documentos de venta.

- Formas de pago

Configuración en clientes de la forma de pago por defecto, con días fijos de pago, periodos de no pago y datos bancarios para la emisión de documentación.

En el maestro de formas de pago, inclusión del tipo de documento y recibos.

Generación de los recibos en las facturas de venta, con seguimiento de cobros.

Inicio
Facturas de venta
Factura de venta 11

Factura de venta

Artículos
Impuestos
Observaciones
Documentos Anteriores
Informe
Recibos

| Nº de factura de venta | Nombre | Fecha de recibo | Fecha de emisió | Importe | Liquidado |
|------------------------|--------------------------|-----------------|-----------------|---------|-----------|
| V/14-000016 | V/13-000000 - 05/01/2014 | 05/01/2014 | 24/01/2014 | 120,60 | No |
| V/14-000016 | V/13-000000 - 05/03/2014 | 05/03/2014 | 24/01/2014 | 120,60 | No |
| V/14-000016 | V/13-000000 - 05/05/2014 | 05/05/2014 | 24/01/2014 | 120,60 | No |

La factura está contabilizada
Aceptar
Cancelar
Opciones

Además, hemos seguido y seguiremos liberando revisiones del producto con mejoras de estabilidad del producto:

- 000393 Integración diseño vBase: formularios de baja independientes.
- 000416 Ordenación en documentos de clientes y proveedores desde el contacto.
- 000420 Localizar entidades directamente, sin filtro por tipo de entidad.
- 000400 Mejora en búsqueda de movimientos.
- 000417 Etiquetas de permisos en vERP.
- 000421 Configuración decimales importes y precios.
- 000432 Corrección en valor de constante personalizar informes.
- 000426 Proceso alta de tipos de documento, corrección.
- 000427 IVA artículos.

Para este 2014 estamos trabajando en una versión con una estructura de herencia mucho más sencilla y más documentación para programadores.

Programa TPV en código abierto

Durante las próximas semanas saldrá a la luz la versión 1.0 de Velneo vTPV con muchas novedades y plenamente funcional, con un diseño de interfaz limpio y potente.

Entregaremos todos los proyectos abiertos y pondremos a disposición de la comunidad de Velneo el mejor código abierto de QML, para un TPV con la funcionalidad básica necesaria en un terminal de punto de venta.

Entre las novedades más importantes tenemos:

- Varias formas de pago
- Emisión de facturas desde tickets
- Gestión de cambio
- y muchas más

Te mostramos algunas capturas de pantalla donde puedes ver la evolución del producto:



Apertura caja...

Fecha 01/04/2014

Turno Turno 1

Depósito...

7 8 9

4 5 6

1 2 3

. 0 C

Aceptar Cancelar

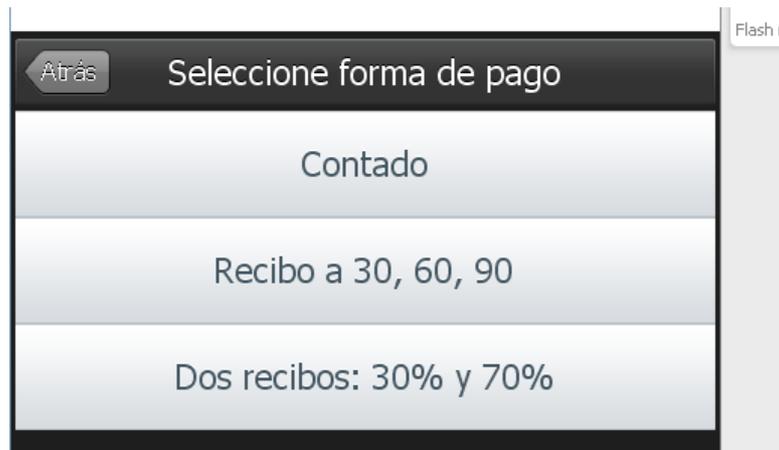
The screenshot shows a POS application window titled 'vTPV'. The top bar includes a user icon, a share icon, and a clock showing 09:50. Below the bar, the user 'docampo' is logged in. A shopping cart on the left lists items with their quantities and prices:

| Item | Quantity | Price |
|-----------------------------------|----------|--------|
| Objetivo | 1 | 200.00 |
| Canon | 1 | 375.00 |
| Libro: Fotografia digital para du | 1 | 24.95 |

The total amount is 599.95€. To the right, a grid of product tiles is displayed under the 'Accesorios' category. The tiles include: 'Cámaras de fotos', 'Accesorios', 'Funda pequeña' (30.00€), 'Tripode' (95.00€), 'Objetivo' (200.00€), 'Funda grande' (50.00€), 'Flash manual' (100.00€), 'Libro: Fotografia digital para dummies' (24.95€), and 'test.pmp' (0.00€). A search bar 'Buscar productos' is located at the top right of the grid.

At the bottom left, a numeric keypad is visible with buttons for digits 0-9, a decimal point, and a 'C' button. To the right of the keypad are buttons for 'Precio', 'Desc', and 'Pagar'.

The screenshot shows a virtual keyboard application window titled 'Teclado'. At the top, there is a search bar containing the text 'lente'. Below the search bar is a standard QWERTY keyboard layout with keys for letters, numbers, and symbols. The 'Alt' key is visible on the bottom left, and a search icon is on the bottom right.



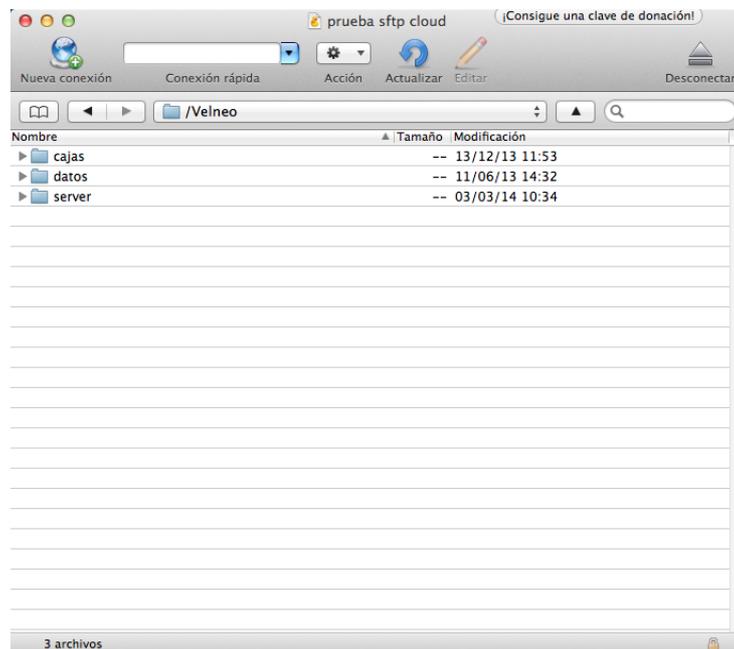
Os recordamos que los suscriptores de vERP y todos los suscriptores de nivel 4, recibirán la actualización del producto en cuanto esté concluido.

A partir de ese momento nos centraremos en la velocidad y estabilidad de ese componente.

Cloud

Explorador Archivos Cloud

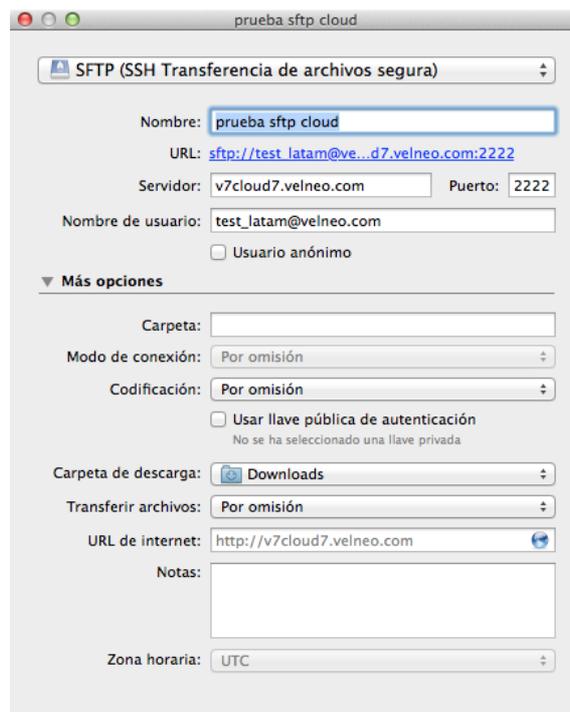
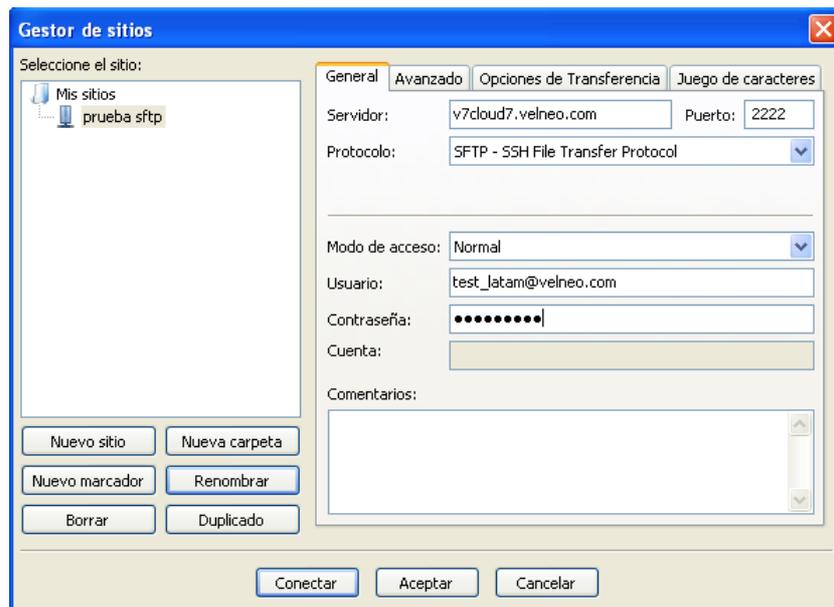
En Velneo seguimos apostando fuerte por el cloud y en esta iteración 7.15 hemos implementado un nuevo servicio que estamos seguros de que te será de gran utilidad: la exploración de archivos [sFTP](#).



Con este nuevo servicio, podrás acceder vía sFTP a tu servidor cloud. Esto te permitirá subir, bajar, reemplazar o borrar ficheros, directorios del mismo, descargar copia de tus datos, subir los datos de vserver local al cloud y todo ello de forma completamente autónoma y con la velocidad de tu conexión. Podrás usarlo a la hora que más te convenga, en función de las necesidades de tu cliente y siguiendo tus propios procedimientos de trabajo.

Y como siempre, siguiendo nuestra filosofía life is soft: basta con instalar un cliente sFTP y conectarse al host donde está ubicado tu Velneo vServer.

En Windows podemos usar por ejemplo [FileZilla](#) y en Mac [Cyberduck](#). Aquí te mostramos como quedarían las propiedades de conexión tanto en una aplicación como en otra.



Importante:

Como podemos apreciar en las capturas de pantalla, el protocolo a usar es SFTP y el puerto **siempre** es el 2222. Este servicio está activo para los vServer de desarrollo de suscriptores de nivel 3 y 4 y para todos los vServers de producción. Si no te encuentras dentro de este grupo y estás interesado en usar el servicio [contacta](#) con nuestro equipo comercial.

¿Tengo que tener mi Velneo vServer arrancado?

Este es un servicio independiente de tu Velneo vServer. No es necesario que esté iniciado. Esta característica permite que puedas interactuar con ficheros específicos de tu Velneo vServer como son los ficheros de usuarios o instancias. Recuerda que manipular esos ficheros puede provocar que el sistema se vuelva inestable, así que te recomendamos hacerlo sólo si sabes lo que haces.

Me devuelve Network error: Connection refused. ¿Qué puedo hacer?

Posiblemente no hayas indicado correctamente el host o el puerto de conexión. Recuerda que debes utilizar como servidor el mismo que usas con tu Velneo vServer (por ejemplo, v7cloud6.velneo.com) y como puerto el **2222**, **independientemente del puerto que tenga tu Velneo vServer.**

Me devuelve error de autenticación. ¿Qué estoy haciendo mal?

Este error puede producirse por dos causas:

1. Estás introduciendo una password errónea: Recuerda que debes utilizar el email y la password que usas para loguearte en velneo.es y que está asociada al vServer que quieres gestionar.
2. No tienes acceso al servicio: Este servicio está reservado para los vServers asociados a cuentas de desarrollo de niveles 3 y 4 y a todas las cuentas asociadas a vServers de producción.

Estoy usando login Social con Google. ¿Qué password debo usar?

Cuando para entrar en velneo.es usas login social, necesitarás establecer una password para nuestros servicios internos de velneo.es independiente de la que tengas en Google o el servicio social que hayas usado. Para ello, accede a tu página de perfil y establece la password de tu elección. Esa será la password que debes usar en el cliente SFTP.

Panel de control cloud

Si todavía te quedaban dudas de como conectarte a tu Velneo vServer en el Cloud una vez creado, con la nueva presentación de Mi Cloud se te disiparán todas las dudas. Ahora se representa claramente que dato se necesita en cada momento y además, cuando se obtiene el Velneo vServer Cloud por primera vez, te mostraremos la clave que necesitas usar además de enviártela por email. Cada vez más fácil.

Mi cloud

Estos son los datos que necesitas para conectarte a tu Velneo vServer usando Velneo vDevelop, vClient o cualquiera de los componentes que forman la plataforma Velneo V7:



| | |
|--|---|
| Estado actual | Detenido |
| Acciones | ▶ Iniciar Iniciar en modo seguro |
| Versión | 7.14.0 Actualizar a versión 7.14.3 |
| Otras acciones | |
| Ver estado del servicio Restaurar Velneo vServer Forzar parada | Obtener backup Resetear password |
| Más información | |
| Cómo instalar Velneo V7 Desarrolla tu primera aplicación | |

Cloud Latinoamérica

Ahora Velneo Cloud cuenta con la zona Latam para que puedas desarrollar más rápido y cómodo si eres de los que viven en el otro lado del charco. Si eres suscriptor de Velneo y estás interesado en este servicio no dudes en solicitarlo a través de nuestro centro de soporte. Notarás un gran aumento de la velocidad de ejecución de tus aplicaciones al disminuir significativamente la latencia de la conexión.

Además aprovechamos para recordarte que contamos con servicio de Velneo Cloud en Latam para que puedas poner tu aplicación en producción de forma muy económica y con un óptimo rendimiento. No dudes en contactarnos para solicitar más información.

TCP en Cloud

Si necesitas crear un servicio TCP en Velneo Cloud o comunicarte desde tu vServer con un servicio TCP, ahora puedes. Con la iteración 7.15 de Velneo V7, hemos habilitado el servicio TCP en la parte del servidor en Velneo Cloud.

Esto te permitirá por ejemplo poner uno o varios puertos TCP a la escucha y así poder comunicarte con otros servicios, sean de Velneo o no a la velocidad del rayo y con la estabilidad que proporciona el multihilo de Velneo vServer. Puedes también hacer uso de [Cirrus](#) para crear tu propio servicio REST.

¿Puedo usar este servicio?

Si eres nivel 4 o dispones de un vServer de producción en Velneo Cloud ya puedes solicitar el servicio de TCP en Cloud.

¿Cómo solicitar el servicio?

Si quieres hacer uso de este servicio, solicítalo a través del centro de soporte. Te proporcionaremos un rango de puertos válido solo para tu Velneo vServer.

¿Cómo sé que tengo el servicio habilitado?

Cuando te comuniquemos que tienes el servicio habilitado, tendrás que reiniciar tu Velneo vServer para que se habilite la nueva configuración. Si una vez hecho eso te conectas con

Velneo vAdmin, verás un mensaje similar al siguiente:

| Mensaje | Fecha-Hora | Descripción |
|---|---------------------|---|
|  Inicio | 2014-03-25 12:56:49 | Inicialización de bases de datos finalizada |
|  vServer | 2014-03-25 12:56:49 | Puerto TCP iniciado con éxito |
|  Inicio | 2014-03-25 12:56:49 | Puertos disponibles para protocolos desde 12010 hasta 12019 |
|  Inicio | 2014-03-25 12:56:49 | Servidor vatp iniciado con licencia Cloud. Versión: 7.15.0.12436, Puerto: 6 |

Donde te está indicando los puertos disponibles para tu Velneo vServer.

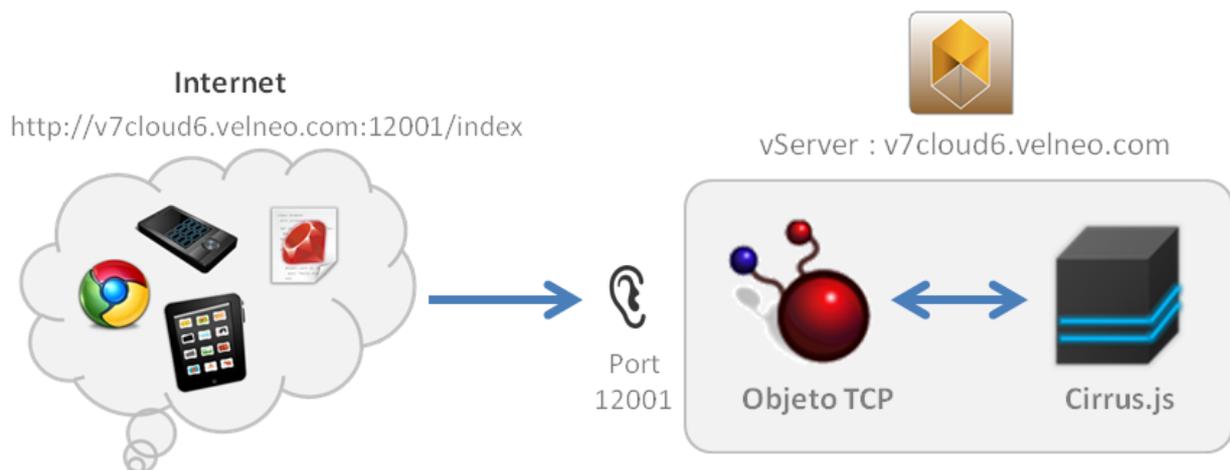
Para mostrar el uso del TCP en el lado del servidor en Velneo Cloud de una forma práctica, nuestro cliente y colaborador Cristian Vázquez ha hecho un proyecto usando su utilidad Cirrus:

Creando servicios web services con v7

Aprovechando al máximo el objeto TCP de velneo podemos crear servicios web que expongan contenido de nuestras aplicaciones de gestión de tal forma que podamos entregar datos a cualquier cliente (Navegador, librería, dispositivo, etc.) que sea capaz de realizar peticiones web, para ello es necesario el uso de 2 componentes:

- (1) Un TCP Handler, en este caso el conjunto de instrucciones TCP que permiten en un puerto específico enviar y recibir strings que respeten el protocolo HTTP.
- (2) Un conjunto de instrucciones que conviertan la petición del protocolo HTTP en propiedades y métodos útiles sobre los cuales podamos actuar y devolver información, en este caso [Cirrus.js](#).

¿Cómo funciona Cirrus?



El funcionamiento de Cirrus es el siguiente; desde Internet mediante browser, dispositivo, navegador o cliente http, se realiza una petición a (<http://v7cloud6.velneo.com:12001/>), la cual es recibida por el vserver que se encuentra escuchando en el puerto 12001 y le pasa la petición a Cirrus.js para que sea procesada, cirrus enruta la petición, la procesa y entrega el response, el cual será devuelto a quien haya hecho la petición.

Si deseas ver este concepto funcionando puedes ver la aplicación de prueba en la siguiente página web <http://v7cloud6.velneo.com:12001>

Si quieres ver Cirrus funcionando en el Cloud de Velneo lo puedes [hacer en el siguiente vídeo](#).

¿Que significa esto para mis aplicaciones?

Significa que usando tecnología 100% velneo v7 puedes permitir que tu aplicaciones sean consumidas por aplicaciones móviles o que sirvan de back-end para aplicaciones web dinámicas, permitiendo todo tipo de interacciones e integraciones de servicios.

¿Es posible usar estas funcionalidades en el cloud de Velneo?

Si, desde la version v7.15 velneo dispone en su cloud la posibilidad de abrir puertos TCP adicionales al que te es asignado por defecto, por lo que será perfectamente posible que tu vServer cloud escuche peticiones web en "v7Cloud6.velneo.com:12001" por ejemplo.

No existe diferencia alguna entre usar Cirrus.js en tu servidor local o en la nube, como tal el objeto TCP solo requiere un puerto por el que pueda escuchar peticiones.

¿Cual es el uso recomendado?

Hay que respetar roles, aunque con Cirrus.js + TCP se puede incluso servir una web completa, lo ideal es crear API's que sirvan de punto de intercambio de datos a otros clientes, dejándole toda la carga del tráfico web, creación UI, manejo de imágenes y librerías a tecnologías que fueron creadas para ello y que por ende son mucho más eficientes, es decir, lenguajes orientados a web (Php, Python, Ruby, Java, etc) usando servidores web potentes (Apache, Nginx, IIS, etc).

Aunque para proyectos pequeños de poco tráfico es viable servir incluso el contenido web directamente desde el vServer.

Utilidades

Exportación a Excel

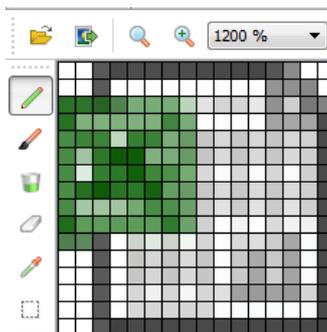
Los usuarios que usan hojas de cálculo Excel ¿Qué desean cuando están ejecutando una aplicación?

La respuesta es poder pulsar un botón y que la información que está viendo en pantalla se vuelque automáticamente a Excel. Fácil, rápido y sencillo.

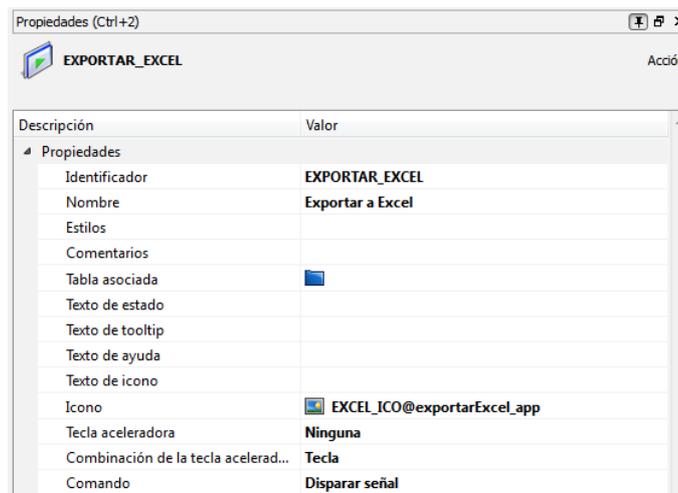
¿Cómo podemos conseguir añadir esta funcionalidad en nuestras aplicaciones?

Realmente podemos hacerlo con una programación realmente sencilla. Solo necesitamos crear 4 objetos y usarlos en las rejillas en las que deseemos que los usuarios puedan disponer de esta funcionalidad. Vamos a ver el sencillo proceso de la programación que requiere.

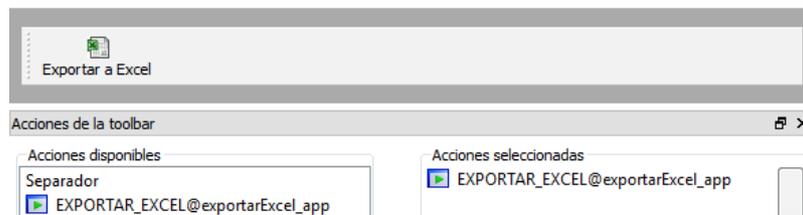
1º Creamos un objeto dibujo (EXCEL_ICO) en el que pondremos el icono a mostrar en el botón de exportación.



2º) Creamos una acción que será ejecutada desde el botón de la toolbar de la rejilla; en esta acción asignamos el icono anteriormente creado. En la imagen siguiente podemos observar cómo el comando a ejecutar en la acción será el de "Disparar acción".

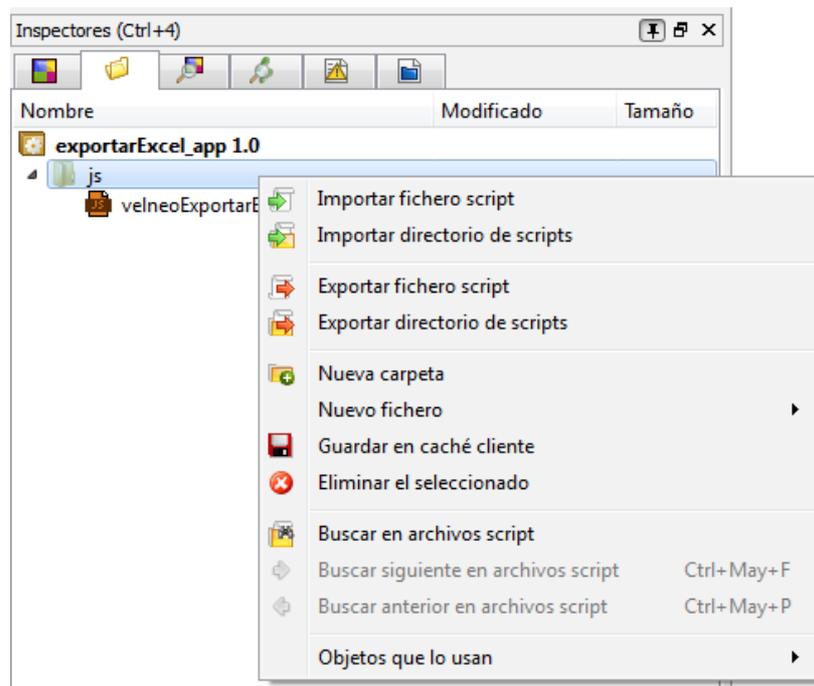


3º) Ahora tenemos que añadir esta acción en la toolbar o toolbars que utilicen nuestras rejillas.



4º) Ahora vamos a incluir en nuestra aplicación el código JavaScript que generará el fichero de VBScript que se encarga de abrir la hoja Excel con la información exportada. Para hacerlo realizamos los siguientes pasos:

1. Crear en el directorio de scripts de la aplicación una carpeta con nombre "js", por ejemplo.
2. ["Descargar el fichero JavaScript"](#) a importar en la carpeta de scripts de tu aplicación.
3. También podemos [descargar el JavaScript en formato PDF](#).
4. Importar el fichero en la carpeta js con la opción "importar fichero script" del menú contextual, como podemos ver en la siguiente imagen:



Una vez completados estos pasos, ya estamos listos para añadir la funcionalidad a nuestras rejillas. Lo primero que tenemos que hacer es asignar la toolbar que corresponda a la rejilla.

| | |
|----------------------------|------------------------------------|
| Color de líneas de rejilla | ■ Texto ventana |
| Pie activo | Falso |
| Toolbar | ■ EXPORTAR_EXCEL@exportarExcel_app |
| Posición de la toolbar | Arriba |
| Longitud de columnas | 360 |

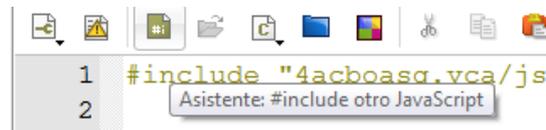
Ya estamos terminando, sólo nos quedan los 3 pasos finales a realizar sobre el objeto rejilla:

- Creamos un manejador de evento de tipo JavaScript que llamaremos EXPORTAR_EXCEL_JS en el que incluiremos el siguiente código:

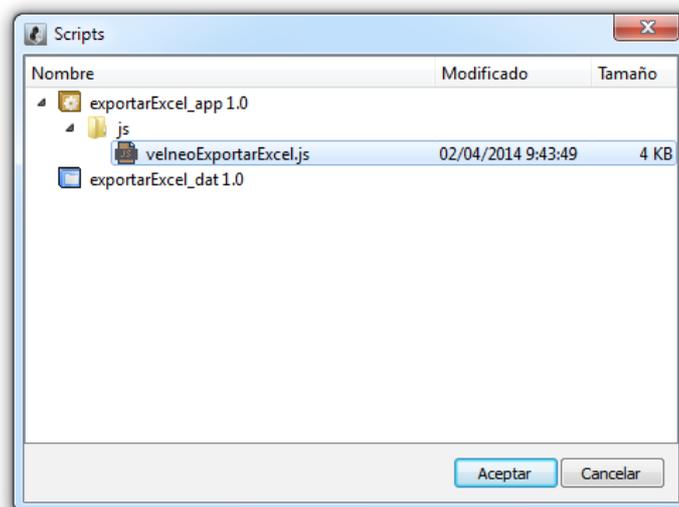
```
#include "4acboasg.vca/js/velneoExportarExcel.js"

// -----
// Exportar contenido de una rejilla a Excel
// -----
var rejilla = theRoot.dataView();
var sendaFicheroExcel = exportarExcel( rejilla );
```

MUY IMPORTANTE. La primera línea que se incluye debemos sustituirla por la que corresponda a nuestro proyecto. En el paso 4º importamos a nuestro proyecto un fichero JavaScript, como la instrucción include hace referencia al proyecto, debemos eliminar esa línea y añadir la que corresponda a nuestro proyecto haciendo uso del botón “Asistente #include para JavaScript”.



Se trata del tercer botón de la imagen que al pulsarlo nos visualizará una ventana donde tendremos que seleccionar nuestro script, como se puede ver en la siguiente imagen:



- Ahora creamos un manejador de evento de tipo V7 que llamaremos EXPORTAR_EXCEL que contendrá el código que podemos ver en la imagen siguiente:

```

Rem ( ----- )
Rem ( Generar fichero scripVB en disco con los datos de la rejilla )
Rem ( ----- )
Interfaz: Ejecutar manejador de evento ( EXPORTAR_EXCEL_JS, )
Libre
Rem ( ----- )
Rem ( Ejecutar fichero scripVB para generar la Excel )
Rem ( ----- )
Ejecutar documento ( sysCacheClientPath + "ExcelScript.vbs", )
    
```

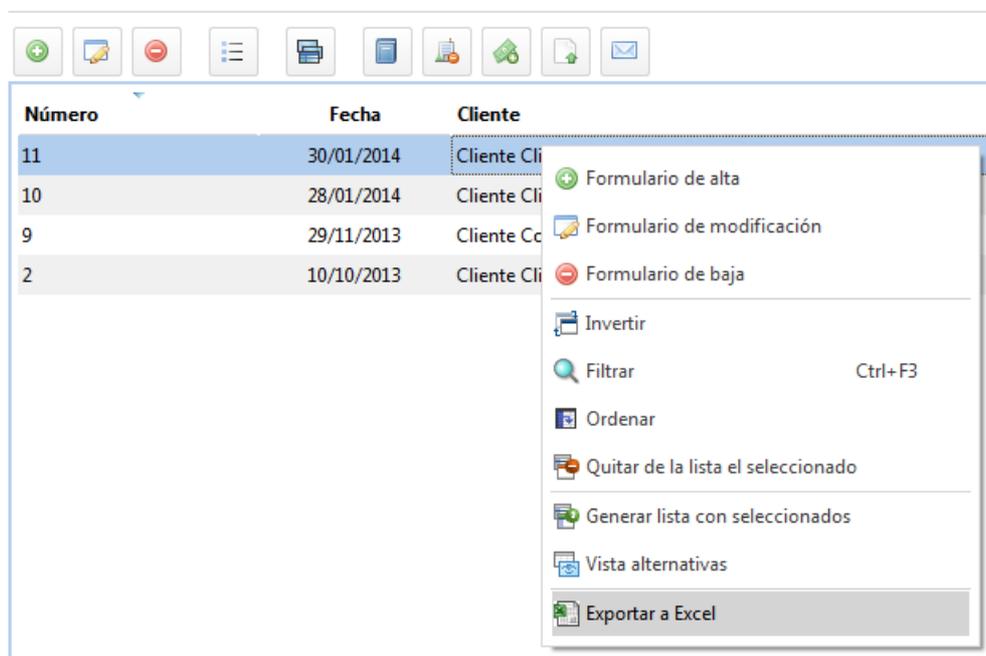
Como vemos la primera línea ejecuta el manejador de evento de tipo JavaScript creado en el paso anterior, que se encargará de generar el fichero VBScript. La segunda línea ejecuta dicho VBScript para que Excel se muestre al usuario con los datos exportados desde la rejilla.

- Creamos una conexión de evento que atraparé la señal emitida por la acción cuando el usuario pulse el botón de la toolbar, y le declaramos el manejador de evento EXPORTAR_EXCEL creado en el paso anterior.



Lo más interesante es que todo este trabajo que hemos realizado ya no tendremos que volver a hacerlo y nos servirá para todas las rejillas de nuestra aplicación. Lo único que tendremos que hacer es copiar la conexión de evento y los 2 manejadores en aquella rejilla a la que queramos añadirle esta funcionalidad. No es necesario tocar nada de código ni en la conexión, ni en los manejadores.

También podemos poner esta opción en el menú contextual de la rejilla como alternativa a la toolbar, quedando igual de operativo y cómodo para el usuario final, sin ocupar espacio físico en la interfaz.



A tener en cuenta:

- Como esta solución está basada en un script de VBScript, sólo es funcional para sistemas operativos Windows.
- Con la versión actual del API de Velneo no es posible exportar el contenido de las columnas cuya fórmula de contenido no se corresponda con un campo, es decir, no se evaluarán las fórmulas. Como solución alternativa podrás usar campo fórmula que devuelvan el mismo contenido y que sí podrán ser exportados por el script. Es posible que en futuras versiones desaparezca esta limitación.
- La velocidad de generación del fichero VBScript es muy alta aunque el tiempo de ejecución del script dependerá del número de registros que contenga la rejilla.

Exportación a CSV

Esta exportación es una versión adaptada de la exportación del contenido de la rejilla a Excel, pero en este caso generando un fichero en disco con formato CSV usando el punto y coma como separador de los datos. Además en la primera línea se genera automáticamente con los títulos de las columnas de la rejilla.

La programación de este método de exportación es similar al explicado en Excel así que indicaré únicamente las diferencias a tener en cuenta.

- La acción a ejecutar en la toolbar o en el menú contextual de la rejilla podemos llamarla EXPORTAR_CSV.
- Aquí puedes [descargar el fichero JavaScript](#) para importar en el directorio de scripts.
- Aquí puedes [descargar el fichero PDF](#) con el código del script.
- En la rejilla sólo es necesario crear un manejador de evento que podemos llamar EXPORTAR_CSV_JS que contiene el siguiente código:

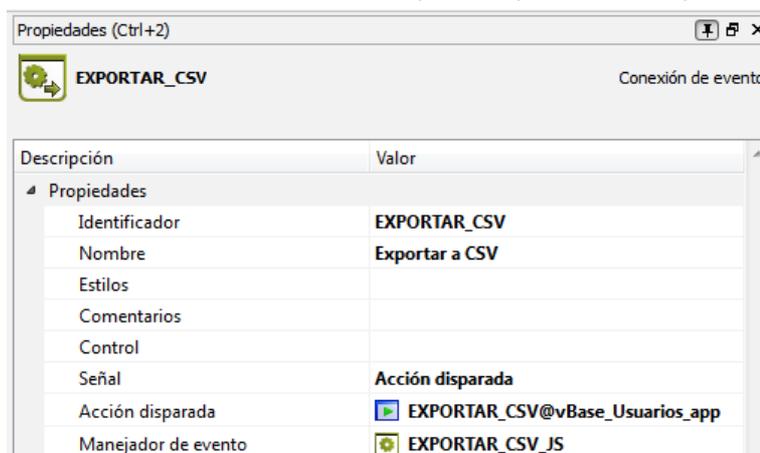
```
#include "21fb5lom.vcd/js/vTools/velneoExportarRejillaCsv.js"
```

```
// -----  
// Exportar contenido de una rejilla a Excel  
// -----
```

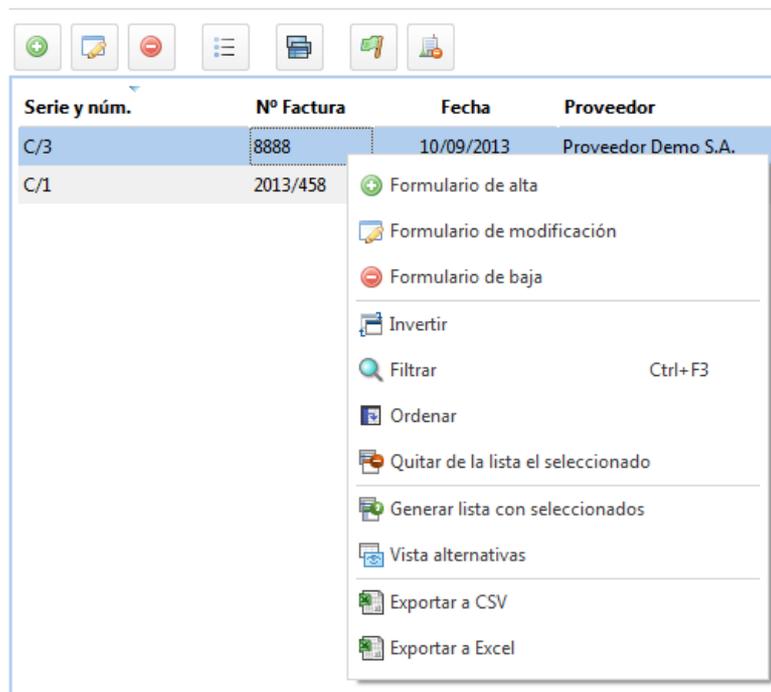
```
var rejilla = theRoot.dataView();  
var sendaFicheroExcel = exportarRejillaCsv(rejilla);
```

MUY IMPORTANTE. La primera línea que se incluye debemos sustituirla por la que corresponda a nuestro proyecto. En el paso 4º importamos a nuestro proyecto un fichero JavaScript, como la instrucción include hace referencia al proyecto, debemos eliminar esa línea y añadir la que corresponda a nuestro proyecto haciendo uso del botón “Asistente #include para JavaScript”.

- En la rejilla declaramos una conexión de evento para atrapar la señal disparada desde la acción.



Una vez programado, los usuarios finales de la aplicación podrá ejecutar directamente la exportación de la rejilla a un fichero con formato CSV.



El fichero se genera automáticamente en el carpeta del usuario y con el nombre que tenga el control que muestra la rejilla. A continuación podemos ver un ejemplo de una exportación de facturas.

```

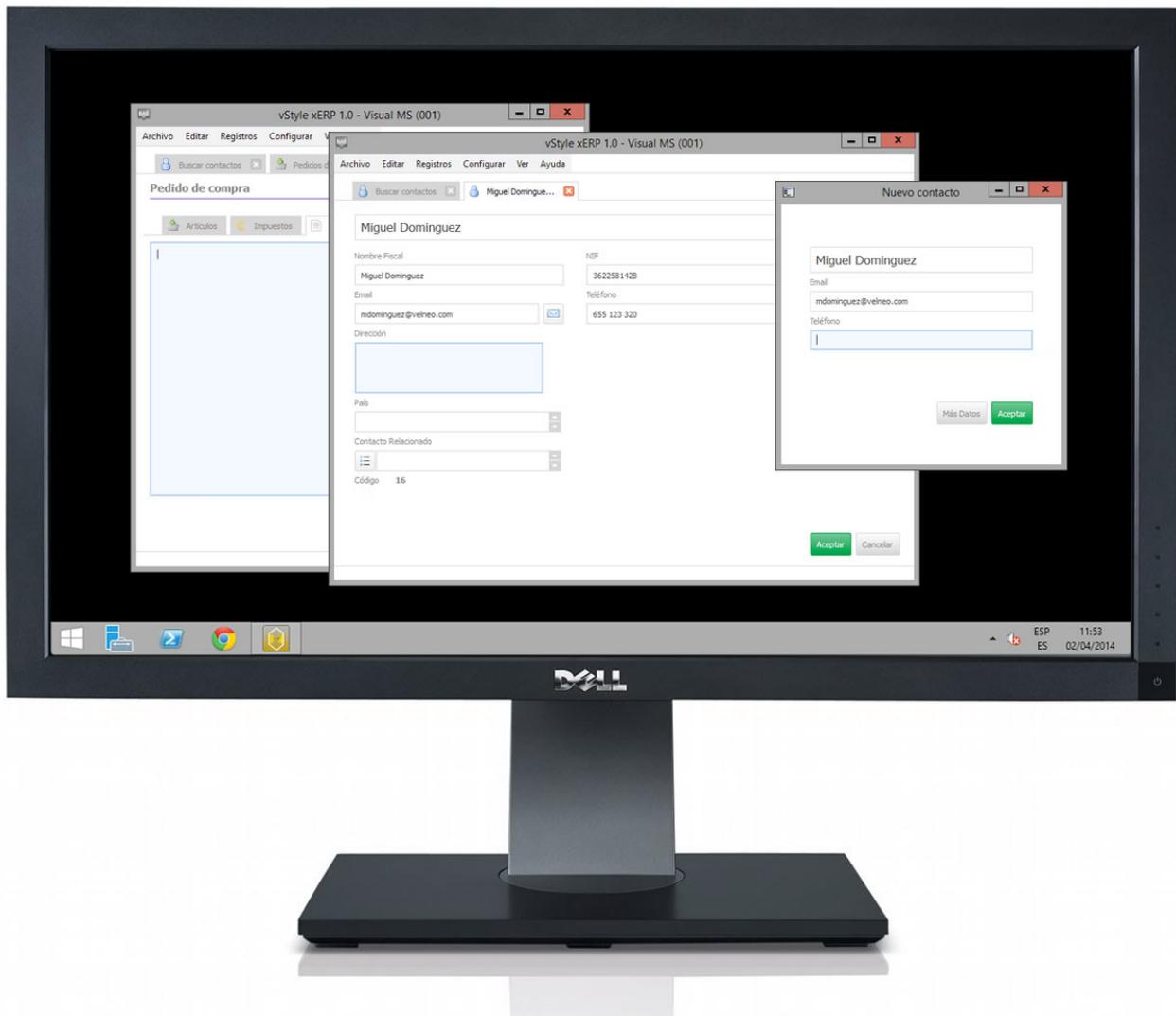
Lista.csv
1 Número;Fecha;Cliente;Total;Facturada;
2 11;30/01/2014;Cliente Clidesa;2902.72;Sin facturar;
3 10;28/01/2014;Cliente Clidesa;-29854.24;Sin facturar;
4 9;29/11/2013;Cliente Contado;-907.5;Sin facturar;
5 2;10/10/2013;Cliente Clidesa;1028.5;Sin facturar;
    
```

A tener en cuenta:

- A diferencia de como ocurría con la exportación a Excel, esta es una solución multiplataforma.
- Con la versión actual del API de Velneo no es posible exportar el contenido de las columnas cuya fórmula de contenido no se corresponda con un campo, es decir, no se evaluarán las fórmulas. Como solución alternativa podrás crear en la tabla un campo fórmula que devuelva el mismo contenido y que sí podrán ser exportados por el script.
- La velocidad de generación del fichero CSV es muy alta, aunque el tiempo de ejecución del script dependerá del número de registros que contenga la rejilla.

CSS en aplicaciones de escritorio.

Una interfaz con [un look and feel](#) atractivo para el usuario, redundará en la calidad percibida del producto así como la experiencia de uso. Para ello debemos asegurarnos que los elementos de la interfaz de nuestro producto sean coherentes, transmiten modernidad y dispongan de un estilo sencillo y agradable.



Desde sus primeras versiones, Velneo V7 nos ofrece la posibilidad de usar [hojas de estilo en cascada o CSS](#) (Cascading Style Sheets), permitiéndonos separar contenido de aspecto, unificar en un mismo lugar todo lo relacionado con el estilo de nuestra App y realizar cambios con mayor velocidad. Las Open Apps Oficiales de Velneo incluyen una CSS personalizada y estable, que hemos mejorado en esta última versión 1.0, y que ahora os ofrecemos para que la uséis en vuestras App's.

- [Documento CSS Open Apps Oficiales versión 1.0](#)
- [Documentación de referencia](#)

Instalar entorno desarrollo rápidamente

Ponemos a tu disposición en un solo ejecutable los componentes básicos de Velneo: vDevelop, vClient y vAdmin.

Descargando este archivo, podrás realizar la instalación de los componentes a toda velocidad y de una sola vez.

Bienvenido
para empezar, descarga la plataforma



Descarga Velneo V7

Versión para Windows

Instalándolos juntos podrás aprovechar todo el rendimiento del actualizador de versiones de la plataforma, de modo que cuando el servidor sea actualizado a una versión más reciente de Velneo los componentes previamente instalados se actualizarán de forma automática, ahorrando tiempo.

En caso de que dispongas además de otros componentes de la plataforma aparte de estos tres (por ejemplo vDataClient) te recomendamos que los extras los instales en una carpeta aparte, porque esos componentes quedan fuera del actualizador automático.

Por todas estas características, es una herramienta útil para todos: programadores, administradores de sistemas y usuarios finales.

Abandono del soporte NPAPI por Google

El equipo de desarrollo del [navegador Chrome](#) de Google [ha anunciado recientemente](#) el abandono del soporte del estándar [NPAPI](#) en su navegador Chrome.

Esto implica que nuestro componente Velneo vWebClient V7 también dejará de estar soportado por este navegador al hacer uso del estándar NPAPI para su inclusión como plugin.

Velneo vWebClient V7 continuará funcionando en navegadores [Internet Explorer](#) de Microsoft y [Firefox](#) de Mozilla, y otros que soporten el estándar NPAPI.

[Durante el año 2014 Chrome procederá al abandono progresivo del soporte](#), requiriendo el uso de configuración específica para su uso en el navegador de Chrome a partir de mayo de 2014, aunque se mantendrá su uso al menos este año.

Como podemos observar por el ejemplo de Google, el uso de NPAPI está discontinuándose, lo que implica que este componente no será evolucionado por Velneo en el futuro y se retirará su descarga de la página principal estando disponible a partir de ahora a través de solicitud al Centro de Soporte de Velneo. No recomendamos su uso a partir de esta versión.

F.A.Q.

¿En qué navegadores continuará funcionando Velneo vWebClient V7?

[NPAPI](#) continúa siendo soportado en IE Explorer y Firefox de Mozilla, por lo que Velneo vWebClient V7 podrá ejecutarse en ambos navegadores.

¿Habrá nueva versión para Chrome?

El abandono de estándares compartidos por los distintos navegadores (IE, Firefox, Chrome) hace muy complicada la evolución de este tipo de productos, ya que requieren desarrollo específico, por lo tanto no está en nuestros planes que evolucionemos este componente.

Formación y Web

Formación en Programación



Hemos mejorado la velocidad de aprendizaje de Velneo con la realización y publicación de múltiple material formativo, desde la realización de cursos on-line básicos y avanzados, hasta artículos para tratar un tema concreto de forma directa, sencilla y sobre todo práctica.

Curso básico

En este curso realizamos una aplicación de pedidos para así sumergirnos en diversas características de Velneo necesarias para el desarrollo de cualquier aplicación de gestión. El temario fue el siguiente:

- Conceptos básicos
- Arquitectura
- Estructura de datos
- Objetos visuales básicos
- Objetos visuales para tablas
- Métodos de acceso a la información
- Procesos
- Eventos y señales

Se puede acceder al mismo desde la opción de "Videotutoriales/02 Formación básica". Se componen de 2 vídeos y aquí os dejamos los links:

1. [Curso de desarrollo básico \(1\)](#)
2. [Curso de desarrollo básico \(2\)](#)

Curso avanzado

En este curso realizamos una aplicación de gestión basándonos en Velneo vERP y así cubrir características avanzadas en el desarrollo con Velneo. El temario fue el siguiente:

- Herencia
- Tablas y fichas de extensión
- Índices complejos
- Procesos, funciones y funciones remotas
- Manejadores, eventos y señales
- Personalización de informes
- Flujo y amalgama
- Traducción

A estos cursos se puede acceder desde la opción de “Videotutoriales/14 Seminarios avanzados”.

14 Seminario avanzado

- [Curso de desarrollo. Avanzado \(1\) 2:21:49](#)
- [Curso de desarrollo. Avanzado \(2\) 2:40:26](#)
- [Curso de desarrollo. Avanzado \(3\) 2:32:17](#)

Consultorías on-line

Durante estos meses también hemos realizado 5 consultorías on-line. Se tratan de talleres de consultoría on-line en grupo, donde cada asistente nos indica previamente una pregunta la cual será resuelta por un consultor durante el evento. Este tipo de talleres ofrecen grandes ventajas ya que te beneficias del conocimiento que obtienes de las preguntas de los demás participantes.

Estos son los temarios de las consultorías on-line realizadas y sus links a los vídeos:

1. Consultoría on-line del 09 de octubre - [Temario](#)
 - a. [Video 1](#)
 - b. [Video 2](#)
2. Consultoría on-line del 30 de octubre - [Temario](#)
 - a. [Video](#)
3. Consultoría on-line del 27 de noviembre - [Temario](#)
 - a. [Video](#)
4. Consultoría on-line del 12 de febrero - [Temario](#)
 - a. [Video](#)
5. Consultoría on-line del 12 de marzo - [Temario](#)
 - a. [Video](#)

Talleres específicos

También hemos publicado una serie de talleres específicos a los que se pueden acceder desde “Videotutoriales/13 Seminarios específicos”. Se tratan de videotutoriales que abordan componentes que nos permiten sacar el máximo partido a Velneo V7. Actualmente están disponibles los siguientes seminarios:

13 Seminarios específicos

-  [Informes con Velneo vReport 2:23:56](#)
-  [Web con Velneo vModApache 1:47:42](#)
-  [Optimización de aplicaciones 2:27:47](#)
-  [Css con Velneo V7 2:04:03](#)
-  [Api de Velneo V7 para Javascript 2:47:13](#)
-  [Desarrollo basado en Open Apps 1:45:19](#)
-  [Web con vModApache 4:06](#)

Buenas prácticas

No sólo nos hemos centrado en formar en el conocimiento de Velneo, sino que también ha sido objetivo nuestro el explicar conceptos, técnicas de programación mediante la publicación de una serie de artículos de buenas prácticas de desarrollo que nos faciliten el mantenimiento y optimización del rendimiento y estabilidad de nuestros desarrollos. A continuación una lista de artículos de buenas prácticas ya publicados:

1. [Conocer la plataforma en ejecución con Velneo V7](#)
2. [Optimizar los formularios para el cloud con Velneo V7](#)
3. [Usar índices condicionados en Velneo V7](#)
4. [Poner buenos nombres, alias e identificadores en Velneo V7](#)
5. [Planos de ejecución en Velneo V7](#)

Vitaminas

Pero también ha sido y es nuestro objetivo aportar código, un recurso o una técnica de programación que nos aporta funcionalidad a implementar en nuestras aplicaciones. Esto son las vitaminas, ejemplos prácticos donde podremos aprender a programar algo con Velneo V7 de una forma práctica con un ejemplo y con el código fuente disponible. Os indicamos a continuación los artículos de vitaminas ya publicados:

1. [Aceptar y seguir en cuadros de diálogo con Velneo V7](#)
2. [Hacer dinámico el título de la ventana con Velneo V7](#)
3. [Titular formulario con nombre de tabla origen en Velneo V7](#)
4. [Exprimir la caja de grupo con CSS y JavaScript en Velneo V7](#)
5. [Optimizar altas con muchas fórmulas o contenidos iniciales en Velneo V7](#)
6. [Actualizaciones y arrastrados con condición dinámica en Velneo V7](#)
7. [Desactivar ordenación en rejillas con CSS en Velneo V7](#)
8. [Deshabilitar controles de formulario con el API de Veleo V7](#)
9. [Deshabilitar controles de formulario con CSS de Velneo V7](#)
10. [Importación automática de datos con Velneo V7](#)
11. [Control de edición de etiquetas en Velneo V7](#)
12. [Aceptar y seguir en formularios en vista y cuadro de diálogo con Velneo V7](#)

Página web sencilla y rápida.

Base de conocimiento / Base de conocimiento comercial ✉ suscribirse

Comparativa entre niveles

Consultor publicó esto el 18 feb. 16:38



En Velneo comercializamos componentes y servicios en packs llamados niveles.

| NIVELES | NIVEL 1 | NIVEL 2 | NIVEL 3 | NIVEL 4 |
|---|---------|---------|---------|---------|
| Desarrollo en Cloud | ✓ | ✓ | ✓ | ✓ |
| Despliegue en Cloud | - | ✓ | ✓ | ✓ |
| Copias de Seguridad Cloud | - | ✓ | ✓ | ✓ |
| Desarrollo Propietario | - | ✓ | ✓ | ✓ |
| Despliegue propietario monousuario | - | ✓ | ✓ | ✓ |
| Despliegue propietario sin límite de usuarios | - | ✓ | ✓ | ✓ |
| Creador de instalaciones | - | ✓ | ✓ | ✓ |
| Editor de informes | - | ✓ | ✓ | ✓ |

Continuando con las mejoras y la simplificación de nuestra web, la hemos orientado de forma clara y efectiva a los recién llegados a Velneo, pasando la información más técnica (tipos de licencias, cloud, despliegue, FAQ's, ...) a la [Base de Conocimiento Comercial](#) para que todos los niveles, a partir del nivel 1, puedan acceder a ella aumentando así la velocidad de carga de la web.

velneo | Centro de soporte

INICIO | BASE DE CONOCIMIENTO | ENVIAR UNA SOLICITUD | REVISAR SUS SOLICITUDES EXISTENTES

Base de conocimiento / Base de conocimiento comercial

Información general | Recientes

- Tarifa de despliegue cloud**
Consultor 24 feb.
- Velneo Nivel 4**
Consultor 19 feb.
- Velneo Nivel 3**
Consultor 19 feb.
- Velneo Nivel 2**
Consultor 19 feb.
- Velneo Nivel 1**
Consultor 19 feb.
- Licencia Velneo vServer V7 Enterprise**
Consultor 19 feb.
- Licencia Velneo vServer V7 Propietaria Standard**
Consultor 19 feb.
- Licencia Velneo vServer V7 Express**
Consultor 19 feb.
- Tarifa de despliegue propietario**
Consultor 19 feb.
- FAQ Tarifas**
Consultor 19 feb.
- Comparativa entre niveles**
Consultor 18 feb.

Podrás acceder a la [Base de Conocimiento Comercial](http://soporte.velneo.es) directamente escribiendo la URL <http://soporte.velneo.es> como hasta ahora.



Ahora las comunicaciones referentes al mantenimiento de nuestro cloud, también estarán presentes en la Base de Conocimiento recogidas directamente desde la cuenta de twitter [@velneocloud](https://twitter.com/velneocloud) para mejorar la comunicación y la estabilidad de la información.

A registration form titled 'Regístrate gratis' with the subtitle 'Crea fácilmente una aplicación hoy mismo'. It features a prominent orange button with the Google logo and the text 'Accede con Google'. Below this, it says 'O con tu email'. The form includes input fields for 'Email', 'Usuario (minúsculas, max. 20 caracteres)', and 'Contraseña' (with a note: 'Se crea una automática que podrás editar'). A green 'Regístrate' button is at the bottom. A small disclaimer at the very bottom reads: 'Haciendo clic en el botón Regístrate estás aceptando las políticas de uso y privacidad de esta web'.

Para facilitar aún más el registro a Velneo, hemos añadido la posibilidad de realizarlo **mediante cuentas de Google**, que sin duda aumentarán la velocidad del registro y acceso a la web. La gran ventaja que tanto si eres nuevo como un usuario registrado podrás usar tu cuenta de google para validarte en velneo.es, consiguiendo que el trabajo en la web sea más *Life is Soft* que nunca.

Todo incluido

Desarrollo de aplicaciones empresariales

| | | |
|--|--|---|
| Nivel 1 Desarrolladores que quieren descubrir Velneo | Nivel 2 Desarrolladores que quieren iniciarse profesionalmente | Nivel 3 Pequeñas empresas con grandes proyectos |
| 0€ Alta gratis | 60€ / mes Alta de 299€ | 153€ / mes Alta de 1.999€ |

Contacta con nosotros

La **página de tarifas** se ha simplificado al máximo orientándole a encaminar los niveles en función del tipo de suscriptor. Para ello, mantenemos los tres niveles iniciales y eliminamos de la tabla el nivel 4, orientado a grandes empresas y que siempre suelen ser gestionados directamente por nuestros consultores.

¿Cómo empezar?

- Videotutoriales
- Formación
- Tutores

Información comercial

- Tarifas
- Casos de éxito

Documentación

- Características
- Ayuda

Hemos **eliminado secciones** que ya no eran utilizadas (servicios, cloud, ...) para que cada vez tengamos una web más limpia y fácil.

Sabemos que aún nos queda mucho por mejorar (Características, Casos de éxito, Documentación, Ayuda, ...) pero estamos convencidos que seguiremos avanzando en la línea de conseguir que la web de Velneo sea una herramienta válida tanto comercial como técnicamente.

Evento de Software. Life is Soft 2015.

Lo primero daros las **GRACIAS** por vuestra asistencia a Life is Soft 2014 . Sin vosotros no habría sido posible cada una de las ediciones.

Todos los que no habéis podido asistir o queráis revisar las ponencias, ya tenéis disponibles todos los videos de la edición 2014. Podéis consultarlos de manera individual en el [siguiente enlace](#). Además también tienes los videos de las ediciones anteriores [aquí](#).

Ya estamos trabajando en la edición de 2015 basándonos en vuestras sugerencias de mejora y aspectos a mantener, recogidos en la encuesta de asistencia life is soft 2014. Importante que os anotéis la fecha en vuestras agendas **12 de Marzo de 2015**. Podéis consultar la información en nuestra página web [lifeissoft](#) .

En esta edición habrá novedades importantes que esperamos sean de vuestro agrado. Os detallamos un resumen de las principales características:

La asistencia presencial a life is soft 2015 será gratuita

¿Qué tal la experiencia de volverte a encontrar con miembros de la comunidad de Velneo? ¿quieres saber de primera mano las novedades de la versión de 2015? ¿Por qué no hacer un break en tu rutina diaria y darte ese respiro? ¿te apetece compartir tu proyecto con el resto de la comunidad?.

No queremos que el tema económico sea un impedimento para que no puedas asistir de manera presencial. La asistencia será gratuita y, aunque el aforo será mayor, las plazas presenciales siguen siendo limitadas.

Por supuesto que siempre puedes asistir por streaming, pero ¿la experiencia sería la misma?. Creemos que no.

No te quedes sin tu plaza y [regístrate ya](#).

Novedad estructura y organización del networking

Aumentamos el horario dedicado al networking – (será de 10.00 a 14.00) para que puedas disfrutar compartiendo tus experiencias y hablando con los miembros del equipo de Velneo.

Habéis valorado positivamente la realización de la mesa redonda en el evento presencial, pues bien ¿Qué te parece presentar tu proyecto al resto de la comunidad?. Vamos a dejar un espacio en el networking para compartir estas experiencias (serán de 5 minutos por empresa) Si estás interesado solicita tu reserva aquí.

También dejaremos un espacio para compartir dudas e inquietudes y organizar círculos con los departamentos de comercial, conocimiento y desarrollo de Velneo.

Esto no podremos emitirlo por streaming, así que, **si no asistes, te lo perderás**. [Regístrate ya](#).

Mantenemos la estructura y duración del evento

La duración será de un día (mantenemos el Jueves) y la estructura similar: mañana networking y tarde ponencias.

Mantenemos la ubicación pero cambiamos la sala de celebración del evento

Somos conscientes de lo que supone edición tras edición cambiar la ubicación, no solo la ciudad sino el sitio donde se celebra el evento. Madrid nos ofrece buena comunicación con el resto de ciudades así que mantendremos la ubicación de la anterior edición.

Realizaremos el evento en la **Universidad Politécnica de Madrid**. Es importante que las nuevas generaciones de programadores conozcan la herramienta y su potencial y valoren su rendimiento y estabilidad.

Cambiaremos la sala donde se realizarán las ponencias y lo realizaremos en el salón de actos. Esta sala mejorará la parte técnica de sonido e iluminación y el aforo (aspectos de mejora que nos comentábais en la encuesta de satisfacción). Esto también permitirá que personas que no han podido asistir en la edición anterior por falta de aforo puedan hacerlo en la edición life is soft 2015.



Mantenemos su emisión en streaming

La emisión en streaming fue uno de los aspectos más valorados de la edición 2014. Para todos aquellos que no podáis asistir al evento presencial de la tarde disponéis de esta alternativa.

También podrás realizar preguntas en directo por twitter a los diferentes ponentes y para esta edición nos anotamos la pantalla con la emisión de los twits.

¿A qué esperas para registrarte? Puedes hacerlo desde [aquí](#)

Empezamos la cuenta atrás. Contamos contigo.

Nos vemos en Life is Soft 2015. 12 Marzo de 2015. Madrid

¿Cómo empezar a programar?

Somos conscientes que para iniciarse en cualquier nueva actividad, es mucho más productivo disponer de una guía sistemática que nos oriente sobre cómo empezar. Para ello, hemos preparado una página que te ayudará en el proceso de instalación de los componentes, configuración del servidor en la nube y creación de tu primera App, con un código fuente muy completo y estable, fácil de comprender y que te permitirá asimilar los conceptos básicos de la plataforma de una manera sencilla y a mayor velocidad.

¿Cómo empezar?
para empezar, descarga la plataforma

Descarga Velneo V7
Windows

- 1 Instala componentes**
Ver información
- 2 Crea tu servidor en cloud**
Ver información
- 3 Ejecuta tu primera aplicación**
Ver información

Instala componentes
Paso 1 del asistente: Descargar e instalar los componentes de la plataforma: El ejecutor de aplicaciones Velneo vClient V7, el administrador de servidor de aplicaciones Velneo vAdmin V7 y el editor Velneo vDevelop V7.

Crea tu servidor en cloud
Paso 2 del asistente: Para completar la configuración, el asistente te ayudará a crear tu servidor cloud, imprescindible para ejecutar tus aplicaciones.

Ejecuta tu primera aplicación
Finalizados los dos pasos anteriores, accede al Velneo vDevelop e instala la aplicación vContactos. Podrás ejecutarla, estudiar cómo está desarrollada y personalizarla.

¿Versión Linux o Mac?
En la página de descargas dispones de los componentes de las versiones para Mac OS X y Linux.

[Página de descargas](#)

© 2014, Velneo S.A. Todos los derechos reservados. Empresa | Contacto | Aviso legal

life is soft

Aprender a programar rápido

Me lo contaron y lo olvidé, lo vi y lo entendí, lo hice y lo aprendí.

(Confucio, filósofo chino)

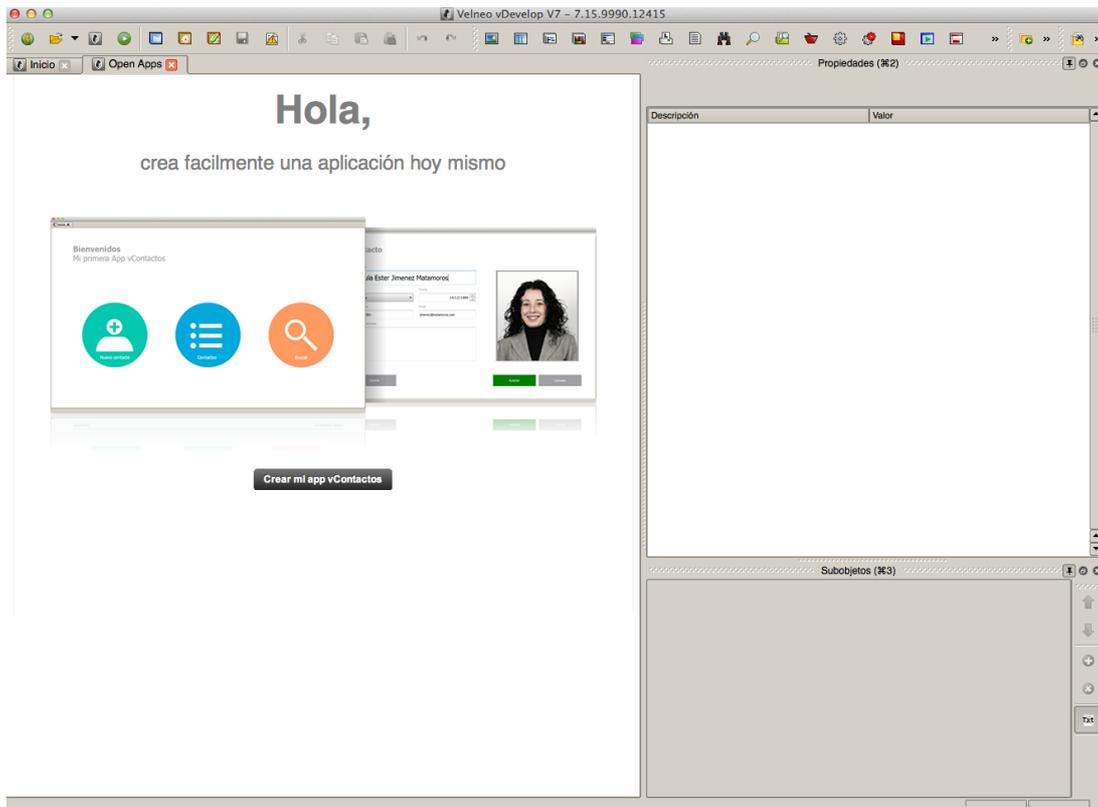
En esta versión de Velneo, hemos trabajado para facilitar y acortar a los nuevos usuarios su aprendizaje con Velneo. La experiencia nos demuestra que aquello que vivimos y experimentamos queda interiorizado y es la mejor forma de aprender.

Es por ello, que nuestro trabajo se ha centrado en acercar Velneo al usuario inmediatamente después del registro, facilitando desde el minuto uno, el acceso a la herramienta con la instalación de sus componentes y lo más importante, con la posibilidad de instalar una aplicación para ponerla en ejecución y probar Velneo de una forma práctica y real.

Se ha creado una página de Bienvenida, que asiste al usuario en este proceso, además con la descarga de Velneo V7, se lanzará un asistente que realizará los dos pasos necesarios para dejar Velneo listo para trabajar

- Instalar los componentes de Velneo
- Configurar el servidor cloud

El tercer paso a realizar, es instalar la aplicación vContactos. Trabajamos en mejorar este proceso, y ahora, desde el vDevelop, pulsando el botón instala, se lanzará un asistente que instalará de forma transparente la aplicación, dejándola preparada para su ejecución.



Consideramos que la aplicación vContactos es la forma más rápida de iniciarse con Velneo, por lo que además de facilitar su instalación y ejecución, incorporamos el vídeo de [“Desarrolla mi primera aplicación”](#) en la zona de Videotutoriales. La finalidad de este vídeo es mostrar cómo se puede desarrollar desde cero esta aplicación, con lo que cualquier usuario siguiendo los pasos del vídeo, en una hora tendrá desarrollada su primera aplicación en Velneo, sin olvidarnos que al tenerla instalada en el vDevelop, también podrá consultar su código fuente para aprender cómo está desarrollada.

La mejor forma de aprender rápido a programar es programando.

Velneo V7 Velocidad y Estabilidad

¿A qué esperas?

