

1. En el vadmin se pueden crear diferentes usuarios-Permisos pero si se desea tener un módulo de usuarios para que se le puedan configurar permisos pero en funciones de la aplicación como tal hay algo que velneo ya tenga estructurado o se debe hacer desde 0. (Paula Andrea Uribe Vásquez)

- Uso de la variable de sistema sysUserName
- Comandos de instrucción para habilitar/deshabilitar y mostrar/ocultar acciones
- Condiciones de activo/visible
- Configuración de usuarios y permisos en vBase

1. Sistema de permisos integrado en vBase

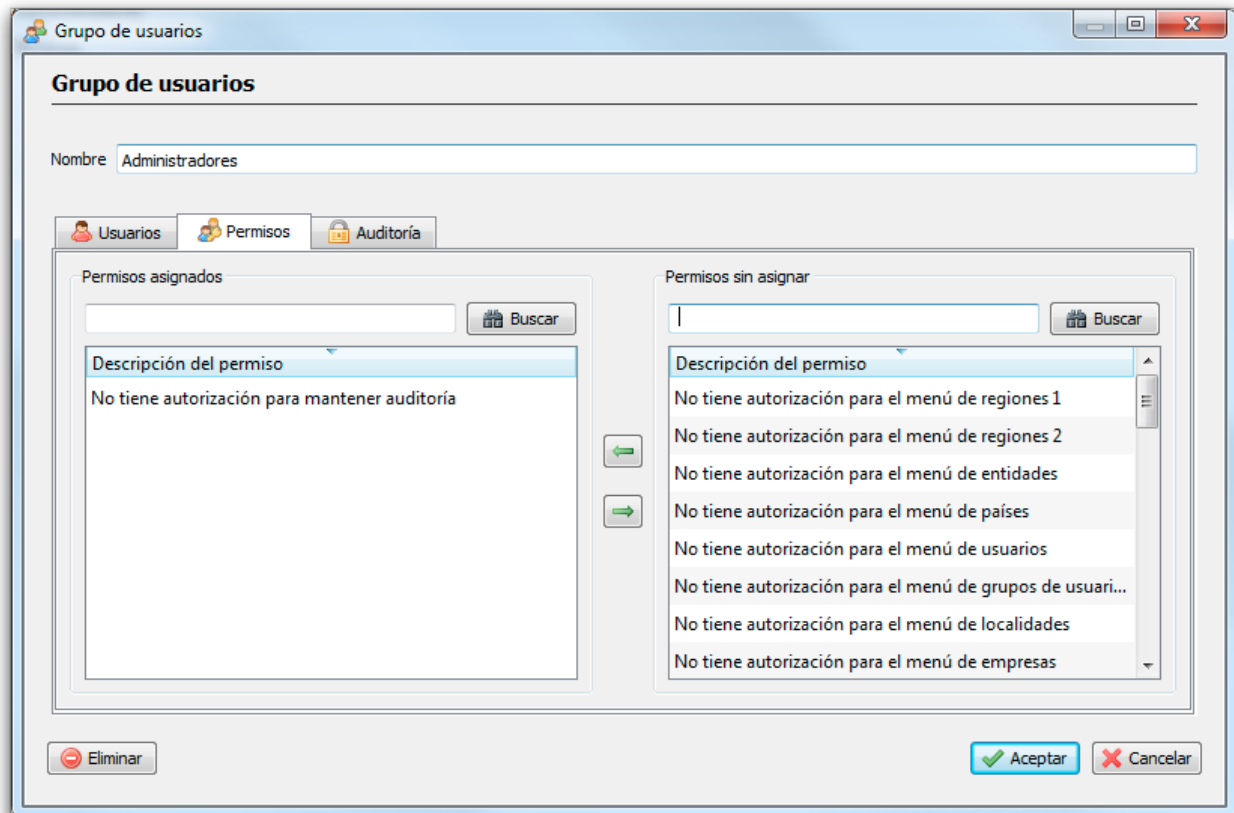
1.1 Objetivos

El sistema de permisos integrado en vBase persigue tres objetivos:

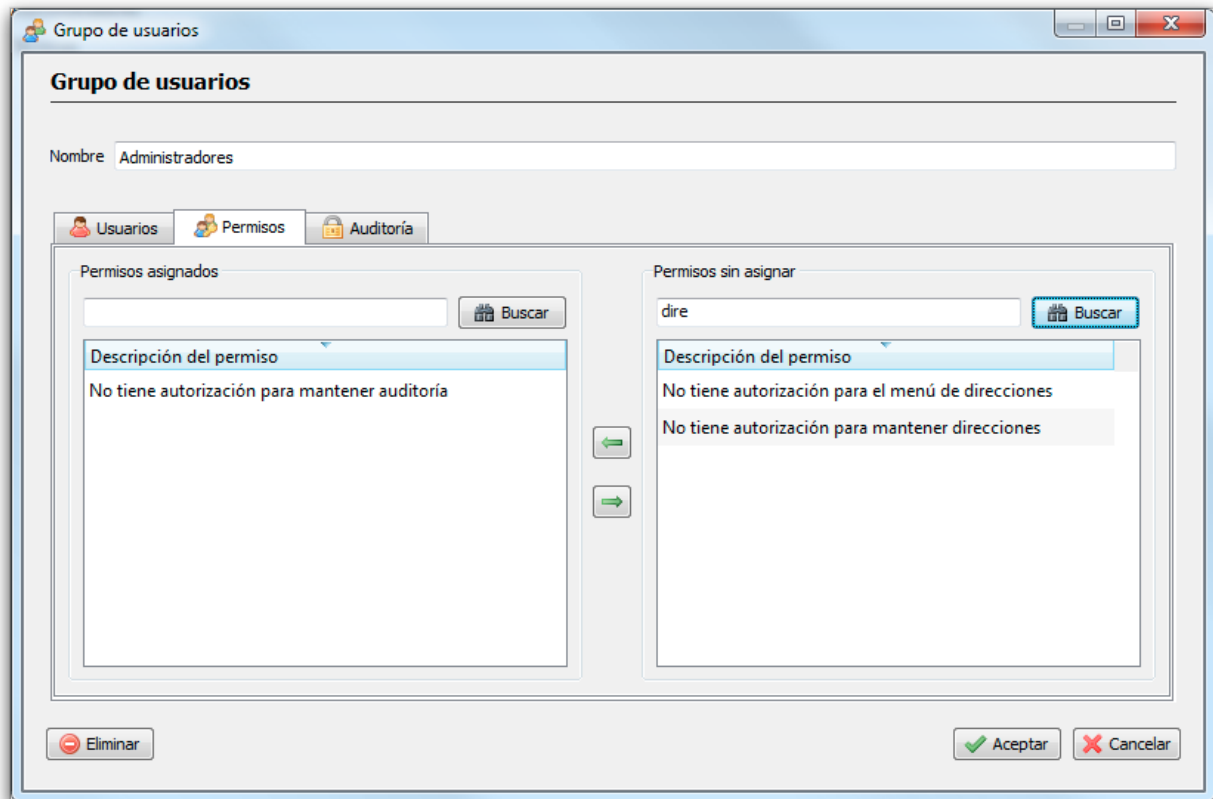
1. Abstracto para que pueda ser utilizado en una gran variedad de funcionalidades.
2. Flexible para facilitar la integración de permisos de las aplicaciones que heredan vBase.
3. Sencillo de implementar por parte del programador.
4. Sencillo de utilizar por parte del usuario. Que sólo sean necesarios definir las excepciones para que requiera la menor configuración posible de permisos.

1.2 Permisos por grupo de usuarios

Los permisos se gestionan a nivel de grupos de usuario, no a nivel de usuario individual. Un usuario puede pertenecer a múltiples grupos de usuarios y asumirá los permisos de todos sus grupos.



Como se puede observar en la siguiente imagen, para cada lista de asignados o sin asignar disponemos de sendos buscadores para facilitar la selección de los permisos.



1.3 Objetos incluidos

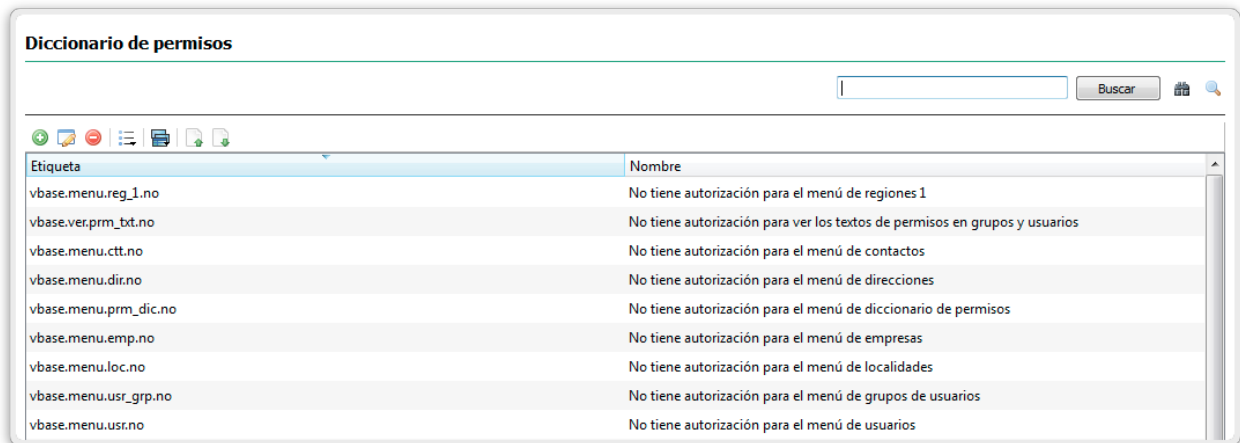
Para implementar el sistema se han incluido los siguientes objetos en vBase:

- Tabla PRM_DIC (Diccionario de permisos) y sus objetos visuales. Su objetivo es almacenar las etiquetas de los permisos a gestionar en la aplicación.
- Tabla PRM (Permisos) y sus objetos visuales. Que almacenará la asignación de permisos a los grupos de usuario.
- Opción del menú Maestros "Diccionario de permisos".
- Subformulario USR_GRP_PRM de "Permisos de un grupo".
- Variable global en memoria PRM_USR. Que almacenará todos los permisos del usuario. Esta variable se usa en la función de validación de permisos cuando es ejecutada en el cliente.
- Función PRM_USR que permite consultar un permiso para el usuario en curso.

1.4 El diccionario de permisos

El diccionario de permisos es muy sencillo de definir. Tan sólo requiere una etiqueta y una

descripción.



Etiqueta	Nombre
vbase.menu.reg_1.no	No tiene autorización para el menú de regiones 1
vbase.ver.prm_txt.no	No tiene autorización para ver los textos de permisos en grupos y usuarios
vbase.menu.ctt.no	No tiene autorización para el menú de contactos
vbase.menu.dir.no	No tiene autorización para el menú de direcciones
vbase.menu.prm_dic.no	No tiene autorización para el menú de diccionario de permisos
vbase.menu.emp.no	No tiene autorización para el menú de empresas
vbase.menu.loc.no	No tiene autorización para el menú de localidades
vbase.menu.usr_grp.no	No tiene autorización para el menú de grupos de usuarios
vbase.menu.usr.no	No tiene autorización para el menú de usuarios

Es importante tener en cuenta que lo mejor es definir la etiqueta en base al criterio de excepción.

Por ejemplo, si todos los usuarios van a tener acceso al menú de países, y sólo a unos determinados usuarios no queremos darles acceso al mismo, lo lógico es definir la etiqueta con el valor de la excepción, es decir, [vbase.menu.pai.no](#)

De esta forma conseguimos que la configuración de los permisos en los grupos de usuario sea realmente sencilla.

Además hay que tener en cuenta que se pueden crear tantos grupos de usuario como nos interese y cada usuario puede pertenecer a múltiples grupos de usuario, siendo sus permisos la suma de todos los permisos (de autorización o de negación de autorización) asignados a los grupos de usuarios que tenga asignados.

1.5 Exportación e Importación del diccionario de permisos

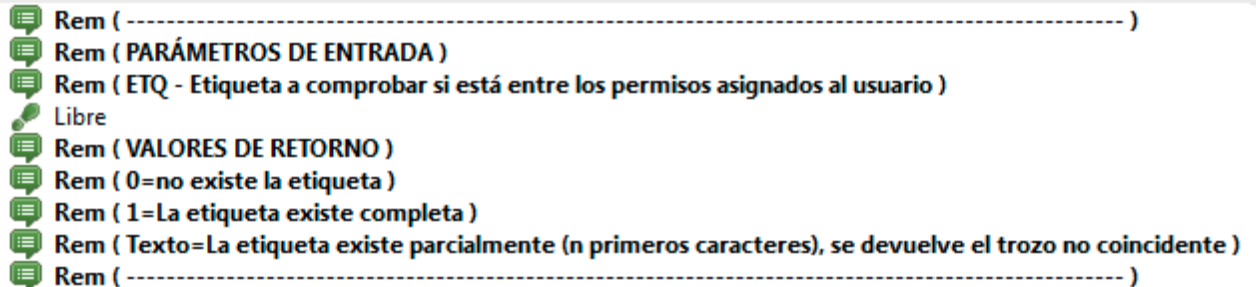
En la ficha de la Open App se encuentra para su descarga el fichero “Diccionario de permisos.txt”. Este fichero permite importar el diccionario en una instalación nueva.

De la misma forma es posible exportar el diccionario de permisos de cualquier instalación a un fichero ASCII para su importación en otra instalación.

1.6 La función PRM_USR

El objeto que más uso tendrá en el nuevo sistema de permisos es la función PRM_USR que se

utiliza. La función incluye documentación de los parámetros de entrada y de los posibles valores de retorno.



```
Rem ( ----- )
Rem ( PARÁMETROS DE ENTRADA )
Rem ( ETQ - Etiqueta a comprobar si está entre los permisos asignados al usuario )
Libre
Rem ( VALORES DE RETORNO )
Rem ( 0=no existe la etiqueta )
Rem ( 1=La etiqueta existe completa )
Rem ( Texto=La etiqueta existe parcialmente (n primeros caracteres), se devuelve el trozo no coincidente )
Rem ( ----- )
```

Si por ejemplo existe en el diccionario la etiqueta [vbase.mant.auditoria.no](#) y ejecutamos la función `fun:PRM_USR@Usuarios.dat("vbase.mant.auditoria.no")`

- Si alguno de los grupos del usuario tiene asignado el permiso nos devolverá un "1"
- Si ninguno de los grupos del usuario tiene asignado el permiso nos devolverá un "0"

Si por ejemplo existe en el diccionario la etiqueta [vconta.cta.clt:430,431,435](#) y ejecutamos la función `fun:PRM_USR@Usuarios.dat("vconta.cta.clt:")`

- Si alguno de los grupos del usuario tiene asignado el permiso nos devolverá "430,431,435"
- Si ninguno de los grupos del usuario tiene asignado el permiso nos devolverá un "0"

1.7 Aplicación de la gestión de permisos en vBase



En vBase se ha implementado el uso del sistema de permisos a diferentes niveles. Esta implementación además de ser funcionalidad sirve de ejemplo para el uso de este sistema en las aplicaciones que hereden vBase.

1.7.1 Ejemplo de control de opciones de menú

```

Rem ( ----- )
Rem ( Aplicar permisos del usuario al menú general de la aplicación y las opciones de alta )
Rem ( ----- )
If ( fun:PRM_USR@Usuarios.dat("vbase.menu.ent.no") = "1" )
  Interfaz: Ocultar acción ( ENT_MEN@vBase )
  Interfaz: Ocultar acción ( ENT_ALT@vBase )
If ( fun:PRM_USR@Usuarios.dat("vbase.menu.dir.no") = "1" )
  Interfaz: Ocultar acción ( DIR_MEN@vBase )
If ( fun:PRM_USR@Usuarios.dat("vbase.menu.ctt.no") = "1" )
  Interfaz: Ocultar acción ( CTT_MEN@vBase )
If ( fun:PRM_USR@Usuarios.dat("vbase.menu.pai.no") = "1" )
  Interfaz: Ocultar acción ( PAI_MEN@vBase )
  Interfaz: Ocultar acción ( PAI_ALT@vBase )
  
```

1.7.2 Ejemplo de control en separador de pestaña

Propiedades	
Identificador	AUD
Nombre	Auditoría
Estilos	
Comentarios	
Formulario	 USR_GRP_AUD@Usuarios
Icono	 BTN_OFF_ICO@vDiseño
Condición visible	fun:PRM_USR@Usuarios.dat("vbase.mant.auditoria.no") = "0"

2. Tengo un software para varias empresas, se tiene claro que se crea una instancia por empresa, si del proyecto de datos algunas Tablas (No estáticas) son alimentadas por nosotros. ¿Como se añadiría un registro que quede en los datos de todas las empresas? Jose Antonio Olaya Restrepo

Por medio de la división de las tablas en diferentes proyectos de datos, podemos definir en la instanciación si un proyecto de datos se reutiliza o se crea nuevo por cada instanciación principal.

Esta instanciación es manual y debe hacerse de abajo a arriba. En el momento de instanciar, normalmente seleccionamos una carpeta distinta para cada instalación. Sin embargo, para compartir datos, lo que hacemos es seleccionar una instancia ya existente, de tal forma que la instancia final resultante, comparte la instancia de datos que hayamos decidido.

Ejemplo con vERP y vBase común

3. Funcionamiento de Herencias (Carlos Jose Llorente Arche)

- Herencia
- Herencia inversa
- Tablas de extensión
- Fichas de extensión

Herencia y herencia inversa

http://velneo.es/info_v7_714_es/velneo_vdevelop_v7/#herencia

Tabla de extensión

http://velneo.es/info_v7_714_es/velneo_vdevelop_v7/proyectos_objetos_y_editores/proyecto_de_datos/tabla/tipos_de_tabla/maestro_de_extension/

Ficha de extensión

http://velneo.es/info_v7_714_es/velneo_vdevelop_v7/proyectos_objetos_y_editores/proyecto_de_aplicacion/formulario/fichas_de_extension/

4. Los errores más frecuentes con javascript/qml (Armando Moreno)

El más habitual un error de parseado (Parse error), porque nos falten o sobren paréntesis. Esto en muchos casos se detecta pulsando el botón de la toolbar de **Verificar** Script (para errores de sintaxis).

Otro bastante habitual es el de errores en las nomenclaturas, por ejemplo usamos una variable en un punto del Script llamada “número” y más abajo usamos “números”, eso lo detectaremos en ejecución. Debemos ayudarnos de los asistentes para incluir información de campos, tablas, etc., de Velneo con el fin de evitar en los posibles errores de ese tipo.

También hay que tener cuidado al copiar código JavaScript que haga #include de otros scripts ya que si lo cambiamos de proyecto cambiará el alias y la senda del fichero de script no será válida.

En cuanto a JavaScript es muy recomendable hacer algún curso básico y leer manuales sobre buenas prácticas. Ya que al ser código puro y duro debemos mimar al máximo su claridad.

También hay que tener cuidado con el ámbito de las variables ya que es importante tener claro dónde deben declararse para no entrar en conflicto con otras del mismo identificador, o que no tengamos visibilidad a su valor.

En general estos son comentados son básicos, luego están los errores derivados de no conocer bien el API a la hora de llamar a funciones de clases o no controlar bien las herencias.

Todo esto y mucho más lo veremos en el curso de JavaScript de final de mes.

Relación de cursos de Formación e inscripción

<http://velneo.es/info/velneo-v7/como-empezar/formacion/>

Curso API de Velneo V7 para JavaScript

<http://velneo.es/info/velneo-v7/como-empezar/formacion/curso-de-api-de-velneo-v7-para-javascript/>

5. Estoy estudiando la manipulación de multiempresas en diferentes bases de datos y con 1 sola base de datos sea multiempresa. Así como ejecutar Webservice a diferentes aplicaciones para conectarse con velneo y poder consultar información de Bancos, Gobierno (Mauricio Oliva)

Respuestas:

Ver primera en punto 2

Ver segunda en punto 10

6. No tengo problema con un red local pero favor de explicar el procedimiento de instalación de vServer para despliegue propietario en un servicio de hosting u otras alternativas si es que las hay de manera que se pueda tener acceso a través de Internet. (Gustavo Camarena)

La mejor forma es hacer uso de Velneo Cloud para un despliegue que requiera acceso a través de internet. No tienes más que solicitarlo a Velneo y tendrás disponible un servidor de Velneo en la nube para el despliegue de aplicaciones. Se te asignará una vrl y un puerto para acceder y comenzar ya a instalar aplicaciones.

Además, desplegando en Velneo Cloud tendrás la posibilidad de usar Velneo Cloud API, lo que te permitirá gestionar usuarios, grupos y permisos de forma dinámica, muy útil para SaaS.

Por otro lado, desde los equipos o redes desde los que vas a acceder, y en el caso de que haya medidas de seguridad establecidas por un equipo de sistemas, tendrás que asegurarte de que el dominio del servidor y el puerto están accesibles, para lo que en ocasiones necesitarás que configuren el firewall o el router.

De otra forma tendrás que solicitar el acceso necesario para la instalación y configuración, normalmente con algún tipo de acceso remoto.

Además, el proveedor de hosting tendrá que proporcionarte información del dominio externo para conectarte (DNS). También tendrás que solicitarle la apertura del puerto que configures para el servidor, por lo que el proveedor deberá configurar su firewall para permitirlo. Si deseas usar el puerto reservado 690, en sistemas linux por ejemplo, el proceso deberá tener permisos especiales, por lo que deberás incluirlo en la información que necesitará el proveedor para la configuración.

También tendrás que estudiar con el proveedor de hosting el sistema de realización de copias de seguridad y la configuración necesaria para proteger el servidor de otros accesos, la configuración del sistema operativo para que funcione de forma óptima, etc.

Todo ello, en cualquier caso, dependerá del proveedor, del servicio que te suministre (servidor virtual, virtual dedicado, compartido) y del sistema operativo.

Después deberás configurar tu firewall o router para que permita acceder a ese servidor, etc.

Componente

<http://velneo.es/velneo-open-app/componente-velneo-vcloudapi/>

Tutor

<http://velneo.es/velneo-open-app/tutor-velneo-vcloudapi/>

Documentación de API

<http://velneo.es/documentacion-velneo-cloud-api-v1-2/>

7. Transacciones en Velneo. Bloqueo Blando/Duro. Concurrencia. Controlar Transacciones sin usar JS. ¿Un proceso es una unidad transaccional? (José Francisco Satué Porta)

- Explicación de qué es una transacción.
- Explicación de bloqueo blando/duro.
- Concurrencia: Lectura posible, Escritura: bloqueos y reintentos.
- Un proceso no es una unidad transaccional. Un proceso que inicia transacción, automáticamente todas las operaciones realizadas en ese mismo hilo serán agrupadas en esa transacción, por lo que las llamadas a otros procesos distintos generarán operaciones dentro del mismo proceso. La única forma de generar transacciones independientes en ese punto es el lanzamiento de Funciones de proceso, que generan transacción independiente.
- Controlar transacciones sin usar JS: Las transacciones son automáticas
 - Función de proceso: Genera transacciones independientes
 - Forzar transacción: Agrupa transacciones en un proceso que no inicia transacción.

Sistema transaccional

http://velneo.es/info_v7_714_es/velneo_vserver_v7/funcionalidades/sistema_transaccional/

Bloqueo Blando / Bloqueo duro

http://velneo.es/info_v7_714_es/velneo_vserver_v7/funcionalidades/sistema_de_bloqueos/

8. Integración de velneo con ISeries (AS400), todas las posibilidades y limitaciones. (Antonio Jose Lorca Zambudio)

En general las posibilidades de integración de datos con AS400 son similares a las de cualquier base de datos, es decir, podríamos vía ODBC atacar su base de datos DB2, esto es válido para soluciones con poco volumen de datos o donde el rendimiento no es crítico.

Hay experiencia de diferentes aplicativos para exportar datos de AS400 a PC para importarlos posteriormente en la base de datos de Velneo. La solución más eficaz que encontraron fue la exportación de datos a ASCII utilizando las funciones de exportación de Client Access. Mediante una tarea programada en AS400 preparaba los datos a exportar que luego mediante Client Access enviaba al disco del PC en un directorio en el que una tarea programada en Velneo se encarga de leer e importar a la base de datos de Velneo. Lógicamente este sistema también funciona en la dirección contraria, aunque para pocos datos, sí que puede ser más cómodo usar ODBC en lugar de utilizar ficheros ASCII de migración de datos.

En uno de los casos de un gran cliente en el que se exportaba diariamente una base de datos de 1 millón de registros que pasábamos a Velneo para su procesado, ya que esos datos se utilizaban en un call center y ahí Velneo permitía búsquedas muy rápidas y más potentes (trozos y palabras). Este sistema también se usó en todas las migraciones de datos de AS400 a PC para el cambio de aplicaciones de RPG a Velneo.

Aquí tienes un artículo que explica como hacerlo si dispone de un sistema de archivos integrado IFS.

<http://www.teknodatips.com.ar/as400/206-como-exportar-e-importar-datos-de-tablas-de-as400-mediante-iserie-navigator.html>

Para la importación y exportación mediante ficheros de texto:

<http://velneo.es/velneo-open-app/tutor-de-importacionexportacion-de-ficheros-ascii/>

Tutor

<http://velneo.es/velneo-open-app/tutor-de-acceso-a-base-de-datos-externas-10/>

Ejemplo

<http://velneo.es/acceso-a-bases-de-datos-externas/>

Tutor usando javascript

<http://velneo.es/velneo-open-app/importacion-y-exportacion-dinamica-con-javascript/>

En el seminario de API de Velneo para Javascript que habrá próximamente, se mostrará un ejemplo de importación/exportación dinámica con API de Velneo.

Todo esto y mucho más lo veremos en el curso de JavaScript de final de mes.

Relación de cursos de Formación e inscripción

<http://velneo.es/info/velneo-v7/como-empezar/formacion/>

Curso API de Velneo V7 para JavaScript

<http://velneo.es/info/velneo-v7/como-empezar/formacion/curso-de-api-de-velneo-v7-para-javascript/>

9. Debo hacer una migración de una antigua base de datos Access (100.000 registros aprox.) a un programa que estoy desarrollando en V7.

Me gustaría poder ver el desarrollo paso a paso de esa migración desde 0, para entenderlo mejor. (Juan Sans)

Comandos de instrucción para el acceso a Bases de datos externas.

El tutor de acceso a Bases de Datos externas usa como ejemplo precisamente el acceso a una base de datos Access, y muestra la importación de registros en la base de datos de Velneo a partir de los datos obtenidos con una sentencia SQL de tipo SELECT.

Tutor

<http://velneo.es/velneo-open-app/tutor-de-acceso-a-base-de-datos-externas-10/>

Ejemplo

<http://velneo.es/acceso-a-bases-de-datos-externas/>

Para la importación y exportación mediante ficheros de texto:

<http://velneo.es/velneo-open-app/tutor-de-importacionexportacion-de-ficheros-ascii/>

Tutor usando javascript

<http://velneo.es/velneo-open-app/importacion-y-exportacion-dinamica-con-javascript/>

En el seminario de API de Velneo para Javascript que habrá próximamente, se mostrará un ejemplo de importación/exportación dinámica con API de Velneo.

Curso API de Velneo V7 para JavaScript

<http://velneo.es/info/velneo-v7/como-empezar/formacion/curso-de-api-de-velneo-v7-para-javascript/>

10. En una de mis aplicaciones debo hacer una conexión con un WEB-SERVICE con el cual debo enviar y recibir información de tipo xml ¿como debo abordar este tema? ¿es posible tener un ejemplo práctico? (José Camino)

Existen dos partes en este tema. Enviar la información y obtener la respuesta y por otro lado procesar/parsear la respuesta.

Para enviar y obtener la información tenemos:

- De forma nativa dentro de Velneo podemos usar “Get fichero from url”, aunque está limitado a llamadas GET.
- Existen multitud de herramientas en línea de comandos que podemos invocar desde Velneo para enviar/recibir contenido desde una url. Por ejemplo curl (<http://curl.haxx.se/>). Curl es una librería que está se distribuye compilada para muchos sistemas operativos y que además suministra una dll para usar desde tus propias aplicaciones.
- Para cosas sencillas recomendamos usar vCurl (<http://velneo.es/velneo-open-app/vcurl/>). Es una librería intermedia entre Velneo y libcurl.dll.
 - Ejemplo: Obtener el xml de <http://feeds.bbc.co.uk/news/england/london/rss.xml> en una variable local de Velneo V7 para poder parsearla
- Para cosas más complicadas, por ejemplo comunicación soap+xml, recomiendo el uso de curl directamente en línea de comandos (ejecutar comando de sistema) indicando a curl que grabe la respuesta en un fichero. Después leer ese fichero con Velneo V7
- Tener en cuenta que cuando se llama a webservice con soap hay que enviar un xml según se indica en WSDL de ese webservice. Aquí explican como: <http://blog.rampartfaq.com/2010/06/how-to-invoke-web-service-call-with.html>
- Importante lo que pone sobre servidores con ssl. Si ese servidor no tiene un certificado emitido por una empresa certificadora de confianza hay que añadir el parámetro -k a la llamada a curl.
- Recomiendo que cuando os funcione la llamada desde línea de comandos la implementéis en Velneo V7.
- Aquí hay una captura del proceso que usamos para lanzar el curl y obtener el resultado:

```

Rem ( ----- )
Rem ( Preparamos los nombres de fichero. El xml de entrada a curl y el txt de salida )
Rem ( ----- )
Set ( FILENAME, homePath()+"\"+fun:GETUUID@vTools.dat() )
Set ( XML_FILENAME, FILENAME + ".xml" )
Set ( FILENAME, FILENAME + ".txt" )
Libre
Rem ( ----- )
Rem ( Escribimos la petición al fichero xml )
Rem ( ----- )
Fichero: Abrir ( h, XML_FILENAME, Lectura/escritura (Crea o limpia), OK, UTF-8 )
Fichero: Grabar línea ( h, DATA, )
Set ( COMANDO, "curl --connect-timeout 300 --silent -H {c} + HEADERS + {c} -d {c}@" + XML_FILENAME + "{c} -o {c}" + FILENAME + "{c}" + HOST )
Libre
Ejecutar comando del sistema ( COMANDO, EXITSTATUS )
Libre
Rem ( <--- Para depuración ---> )
Fichero: Abrir ( h, homePath()+"comandos.txt", Lectura/escritura (Crea o agrega), OK, UTF-8 )
Fichero: Grabar línea ( h, COMANDO+"r\n", )
Libre
Set ( RESULTADO, "" )
If ( 0=EXITSTATUS )
  Fichero: Abrir ( h, FILENAME, Sólo lectura, OK, UTF-8 )
  For ( K, 0, OK, 1 )
    Fichero: Leer línea ( h, LINEA, OK )
    If ( OK )
      Set ( RESULTADO, RESULTADO+"r\n"+LINEA )
Else
  Set ( RESULTADO, ~XML_ERROR@vTools.dat + ": Ejecución del comando fallida" )
Eliminar fichero de disco ( XML_FILENAME, )
Eliminar fichero de disco ( FILENAME, )
  
```

- Ese proceso hay que ejecutarlo en 3P y por eso tengo una función a la que realmente llamo para lanzar el comando

```

Crear manejador de objeto ( p, Proceso CURL_RUN@ )
Set variable local de objeto ( p, HEADERS, HEADERS )
Set variable local de objeto ( p, DATA, DATA )
Set variable local de objeto ( p, HOST, HOST )
Disparar objeto ( p, 1º plano: Local (síncrono), OK )
Libre
Get variable local de objeto ( p, RESULTADO, RESULTADO )
Set dato de retorno ( RESULTADO )
  
```

- Importante: Los datos que se pasan a curl en el parámetro -d (XML_FILENAME), el contenido de ese fichero, debe estar url_encode.

Para procesar el resultado en caso de que se trate de XML podemos hacer uso de algunas

funciones que se incluyen en la open app vTools (<http://velneo.es/velneo-open-app/vtools/>):

- getChildNode(node, pos): Retorna la primera ocurrencia si pos=0 del nodo pasado por parámetro
- nChildNode(node): Retorna el número de ocurrencias de node
- getAttribute(attribute): Retorna el valor del atributo pasado como parámetro

En el caso de JSON podemos procesar el resultado directamente con vJavascript.

Uso de QML para webservice

<http://velneo.es/velneo-open-app/qml-ejemplo-bus-gijon/>

11. Cómo implementar jQuery o jQuery Mobile (Joel Cabrera)

jQuery y el jQuery Mobile tiene, sobre todo, sentido implementarlo para aplicarlo con el controlHTML.

No es posible usar jQuery para la ejecución de procesos y eventos javascript de Velneo, ya que está hecho para Web, trabaja con DOM (Document Object Model) y BOM (Browser Object Model) y otros elementos y funciones que proveen los navegadores web. De ahí, que deba usarse en el control HTML. JavaScript puede usar el API de Velneo para scripts.

En cualquier caso, a efectos de implementar sería como en cualquier otro navegador, basado en WebKit en cuanto a funcionalidad (Safari, Firefox).

Un ejemplo de integración es la Open Apps vHighcharts.

<http://velneo.es/velneo-open-app/vhighcharts/>

Los ficheros de jQuery pueden tanto meterse en un fichero JavaScript en el directorio de scripts o también como ficheros adjuntos para luego indicar en el HTML que la senda base del controlHTML buscará los ficheros jQuery (incluidos) en el directorio del caché del cliente.

Curso API de Velneo V7 para JavaScript

<http://velneo.es/info/velneo-v7/como-empezar/formacion/curso-de-api-de-velneo-v7-para-javascript/>