Cambios en la plataforma para realizar la gestión de IVAS

Estos son los cambios realizados en agrolur.com (srv-web/agrolur). Es posible que en algún otro proyecto haya que hacer más / otros cambios. No os toméis esto al pie de la letra.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BBDD

generamos el nuevo campo para el IVA general de pedido

```
ALTER TABLE `tblregpedidos` ADD `pedIVA` FLOAT NOT NULL DEFAULT '16' AFTER `pedBaseImpo` ;
```

Comprobamos que los IVA de productos y familias están correctos Aquí tenemos que revisar la gestión de los IVA en productos y en familia, ya que por defecto en vez de relacionar el proIVA con el id de la tabla tblmaeivas, lo relaciona directamente con el valor, por lo que hay que cambiar todos los valores de los productos a mano (o con una update). Otra opción es lo que he hecho que consiste en cambiar la forma en la que metemos los valores de IVA en los productos, así como la forma en la que los recogemos.

Aquellos proyectos en los que tengamos Familias Comerciales (basterra, salica, etc…) hay que recordar que también debemos modificar esas familias (igual que con familias).

Cambios en BBDD (entiendo que los valores de ivaId no cambian respecto de los creados al generar el proyecto, en caso contrario poned los que os correspondan):

```
UPDATE tblcfgproductos SET proIVA = 1 WHERE proIVA = 16;
UPDATE tblcfgproductos SET proIVA = 2 WHERE proIVA = 7;
UPDATE tblcfgproductos SET proIVA = 3 WHERE proIVA = 4;
```

```
UPDATE tblcfgfamilias SET famIVA = 1 WHERE famIVA = 16;
UPDATE tblcfgfamilias SET famIVA = 2 WHERE famIVA = 7;
UPDATE tblcfgfamilias SET famIVA = 3 WHERE famIVA = 4;
```

FICHEROS

catalogo/produedit.inc

Sustituímos (sobre la línea 464 aprox.)

```
ListaDesplegable ($rs, "proIVA", "ivaIVA", " ivaIVA", $Campos["proIVA"],
"Textos", "", false, $manual);
```

Por:

```
ListaDesplegable ($rs, "proIVA", "ivaIVA", "ivaId", $Campos["proIVA"],
"Textos", "", false, $manual);
```

catalogo/familias.php

Sustituímos (sobre la línea 285 aprox.)

```
ListaDesplegable ($rs, "famIVA", "ivaIVA", " ivaIVA", $Campos["famIVA"],
"Textos", "", false, $manual);
```

Por:

```
ListaDesplegable ($rs, "famIVA", "ivaIVA", "ivaId", $Campos["famIVA"],
"Textos", "", false, $manual);
```

api/apicatalogo.inc

findProducto → Cambiamos la obtención del valor de IVA

```
$Campos["proIVA"] = $this->BuscarIVA ($Familia, $Campos["proIVA"]);
```

Sustituimos la función BuscarIVA actual por la siguiente:

```
function BuscarIVA($Familia, $producto) {
    // Configuración Ecomm
    $sqlTxt = "SELECT * FROM tblmaeivas WHERE ivaId = ".$producto;
    $rs = $this->Conn->Consulta($sqlTxt);
    $num_resultados = $rs->rowcount;
    $Campos = $rs->Reg();
    $rs->Close();
    // si falla el IVA por la razon que sea que lo coja de la familia
    if ($this->Conn->HayError() || $num_resultados == 0 || $Campos["ivaIVA"]
== 0) {
        if ($Familia == array()) {
            return ($this->IVA_General);
```

```
        } else  {
                                return $this->BuscarIVA($Familia[CAMPO_PADRE],
$Familia[CAMPO_IVA]);
        }
    } else {
    // tenemos que seleccionar el IVA
        return $Campos["ivaIVA"];
    }
}
```

Comprobamos los IVA generales

api/apicompra.inc

Cambiamos las funciones MKWCompra → inicializamos la variable $this→IVA_General = 0; Cargar → Le asignamos el valor general $this→IVA_General = $Campos['ecommIVA']; formatPedido → ponemos $Pedido[“pedIVA”] sustituyendo a los IVA generales que tengamos Añadimos un campo nuevo con los demás campos. $Compra[CAMPO_IVA_GENERAL] = (float)$Pedido[“pedIVA”]; formatCompra → Añadimos En este caso a veces la llamamos desde el Carrito y otras desde el gestor. Tenemos que diferenciarlas, por lo que le pasamos a getCompra como parámetro la info del pedido (le pasamos el $_POST de pedidoedit.inc). En mi caso en $DetallePedido le paso el pedido como parámetro a la función (en forma de Array) para que coja los datos de ahí en vez de cogerlos del Carrito si es un pedido cogemos IVA_General desde el pedido. Si no lo cogemos de la variable IVA_General

```
if ($DetallePedido == array()) {
    $iva_general = $this->IVA_General;
} else {
    $iva_general = $DetallePedido["pedIVA"];
}


$Compra[CAMPO_IVA_GENERAL] = $iva_general; (esta variable es la que luego
usaremos en la template de pago.tpl)
```

RegistrarCompra → Cambiamos el INSERT INTO. Añadimos pedIVA con valor $this→compraDatos[CAMPO_IVA_GENERAL]

ecomm/pedidoedit.inc

Añadimos los valores generales de ecomm al principio del fichero (después de los include)

```
// Configuración Ecomm
$rs = $Conn->Consulta("Select * from tblcfgecomm");
$Campos = $rs->Reg();
$rs->Close();
$Dto_General = $Campos["ecommDto"];
$Margen_General = $Campos["ecommMargen"];
$Iva_General = $Campos["ecommIVA"];
```

En el <form> añadimos la fila del IVA del pedido (para usarlo en subsiguientes operaciones)

```
<input type='hidden' name='pedIVA' id='pedIVA'
value='<?=$Pedido["pedIVA"]?>' />
```

En opcion Guardar/Actualizar → modificar el INSERT INTO para añadir el valor del IVA_general ($Iva_General)

En las opciones en las que aparezca el 16 lo sustituimos por el valor de $Pedido[“pedIVA”] (que es el que obtenemos de la tabla tblregpedidos)

TEMPLATES

compra/pago.tpl

sustituimos los 16 por el campo CAMPO_IVA_GENERAL que obtenemos de la función formatCompra en apicompra.inc

varios/pedidocom.tpl

sustituimos los 16 por el campo CAMPO_IVA_GENERAL que obtenemos de la función formatPedido en apicompra.inc

Comprobamos las operaciones de compra y generación de pedidos

Tenemos que llevar a cabo las siguientes operaciones:

Comprobaciones al 7% y 16 %:

1. - Compra al 16 % para comprobar el correcto funcionamiento de los cambios que hemos realizado.
1. Comprobación de que el pedido resultante lo podemos ver desde menú de usuario –> gestión de pedidos y lo vemos correctamente.
1. Acceso al gestor y modificación del pedido para ver que se hace correctamente.
1. Generación de nuevo pedido desde el gestor.

Comprobaciones al 8% y 18 %:

1. Realizar los cambios pertinentes en ecoomerce → valores generales y en e-commerce → Gestión de IVAS para cambiar los valores por el 18% y el 8%.
1. Compra al 18 % para comprobar el correcto funcionamiento de los cambios que hemos

realizado.

1. Comprobación de que el pedido resultante lo podemos ver desde menú de usuario –> gestión de pedidos y lo vemos correctamente.
1. Comprobación de que un pedido anterior generado al 16% sigue mostrándose al 16%.
1. Acceso al gestor y modificación del pedido para ver que se hace correctamente.
1. Cambio desde gestor de un pedido realizado al 16% para comprobar que los valores se cogen correctamente.
1. Generación de nuevo pedido desde el gestor.
1. Volvemos a comprobar el pedid generado.

From:
https://wiki.merkatu.info/ - **Wiki de merkatu**

Permanent link:
https://wiki.merkatu.info/cambio_iva?rev=1276240891

Last update: **2017/03/27 17:43**