



PrefSuite

the window maker's preference

Módulos de PrefUserVB

WhitePaper

Noviembre de 2006



Contenidos

1. Propósito	3
2. ApplicationModule	3
2.1. Eventos Principales de ApplicationModule	3
3. CustomerModule	3
3.1. Eventos Principales de CustomerModule	4
4. ModelModule	4
4.1. Eventos Principales de ModelModule	5
5. PricesModule	12
5.1. Eventos Principales de PricesModule	12
6. ProductionModule	12
6.1. Eventos Principales de ProductionModule	12
7. PurchasesModule	17
7.1. Eventos Principales de PurchasesModule	17
8. DataManagerModule	19
8.1. Eventos de DataManagerModule	19

1. Propósito

Este documento explica brevemente los módulos más relevantes que se encuentran dentro de un proyecto PrefUserVB.

Así mismo, se mencionan los eventos que intervienen en estos módulos, especificando donde se dispara cada uno de ellos.

2. ApplicationModule

Este módulo dispara eventos propios de las aplicaciones PrefSuite. Es decir, dispara eventos al cargar una aplicación, salir de ella,... veamos los mas relevantes.

2.1. Eventos Principales de ApplicationModule

```
Public Sub OnExit()
```

Este evento salta cuando salimos de las aplicaciones PrefCAD y PrefGest.

```
Public Sub OnInit()
```

Este evento se dispara cuando iniciamos las aplicaciones PrefCAD y PrefGest.

```
Public Sub OnMainUI()
```

Este evento se dispara cuando el usuario pulsa, dentro de la aplicación de PrefCAD, el botón de conexión industrial.

3. CustomerModule

Este módulo se encarga de disparar eventos, en función de los pasos que hagamos dentro de la pestaña clientes que se encuentra, en el grupo de ventas, dentro árbol de aplicaciones de PrefGest. Veamos los eventos principales de este módulo.

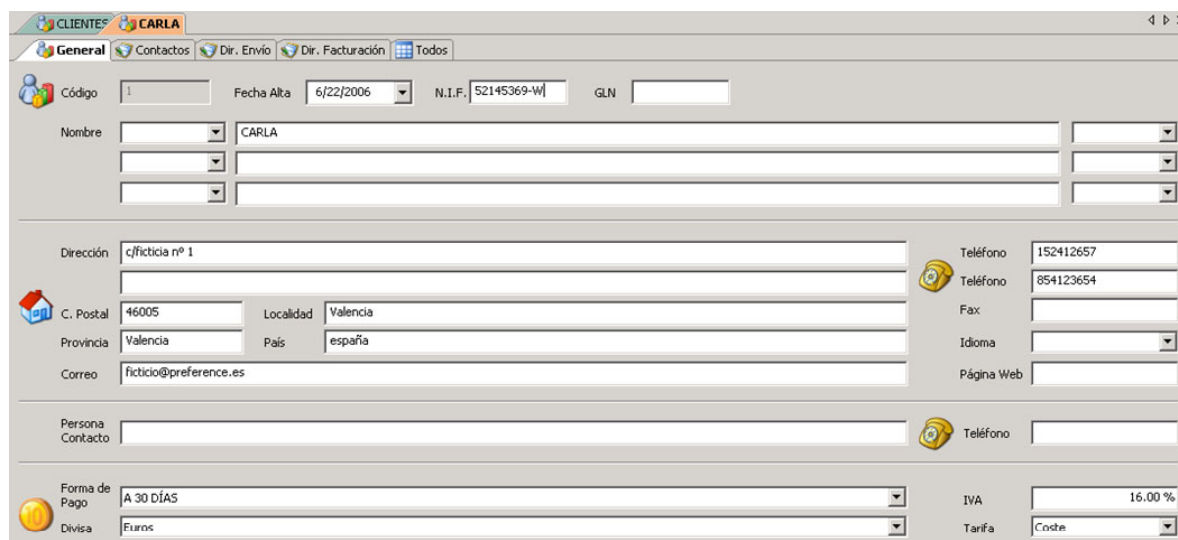


Figura 1. Pestaña de clientes en PrefGest.

3.1. Eventos Principales de CustomerModule

```
Public Sub Customer_AfterSave (ByVal Customer As PrefGest.ICustomer, Number As Long)
```

Este evento se dispara después de guardar los datos que hemos introducido de nuestro cliente.

```
Public Sub Customer_BeforeSave (ByVal Customer As PrefGest.ICustomer, ByVal Number As Long, Cancel As Boolean)
```

Este evento salta antes de haber guardado los datos que estamos introduciendo de un cliente.

```
Public Sub Customer_ExtraUI (ByVal Customer As PrefGest.ICustomer, ByVal Number As Long)
```

Este evento se dispara cuando el usuario pulsa, dentro de la pestaña del cliente en PrefGest, el botón de personalización.



Figura 2. Botón de personalización.

4. ModelModule

Este módulo se encarga de disparar eventos relacionados con el comportamiento / estado de los modelos. Estos eventos son llamados por la aplicación de PrefCAD.

4.1. Eventos Principales de ModelModule

```
Public Sub OnAfterCalculate(ByVal model As IDualModelo)
```

Este evento se dispara justo después de que en PrefCAD o en PrefGest se calculen o generen los materiales del modelo.

Un ejemplo de uso de este evento es para asociar un tipo de producto a un modelo en concreto.

```
Public Sub OnAfterDimensionChange(ByVal model As IDualModelo, ByVal DimensionName As String, ByVal NewValue As Double)
```

Este evento se dispara después de que el usuario cambia las dimensiones de un modelo, ya sea en PrefCAD o PrefGest.

Cota	SubCota	Valor	Variable
L		1,000.00	
A		1,000.00	

Figura 3. Dimensiones de un modelo.

```
Public Sub OnAfterGenerateMaterials(ByVal model As IDualModelo)
```

Este evento salta justo después de generar materiales de un modelo en concreto ya sea en PrefCAD o PrefGest. El uso de este evento sería, por ejemplo, para añadir nuevos materiales a nuestro modelo por código VB.

Referencia	Nº	Longitud	Altura	A	B	Coste	Venta	Total
SCC831500	1	904.00	0.00	45.00	45.00	0.00	0.00	0.00
SCC831500	1	904.00	0.00	45.00	45.00	0.00	0.00	0.00
SCC831500	1	904.00	0.00	45.00	45.00	0.00	0.00	0.00
SCC831500	1	904.00	0.00	45.00	45.00	0.00	0.00	0.00
SCC860116	1	1,006.00	0.00	45.00	45.00	0.00	0.00	0.00
SCC860116	1	1,006.00	0.00	45.00	45.00	0.00	0.00	0.00
SCC860116	1	1,006.00	0.00	45.00	45.00	0.00	0.00	0.00
SCC860116	1	1,006.00	0.00	45.00	45.00	0.00	0.00	0.00
VIDRIO	1	894.00	894.00	0.00	0.00	0.00	0.00	0.00
SCC224510	1	3,456.00	0.00	0.00	0.00	0.00	0.00	0.00

Figura 4. Pantalla de materiales generados.

```
Public Sub OnAfterPropertiesDialog(ByVal model As IDualModelo, ByVal OkPressed As Boolean)
```

Este evento se dispara al pulsar salir de la pantalla de propiedades de un modelo, ya sea en PrefCAD o PrefGest. Desde este evento podríamos programar, por ejemplo, que saliera un mensaje al aceptar las propiedades de un modelo, ya sea un mensaje de error tipo “La hoja 123 no es compatible para este modelo”, de tipo advertencia como “Por la altura de la ventana sería conveniente poner un herraje de seguridad nivel 2”, etc.... No obstante, esto también se puede configurar o codificar como plausibilidades.

```
Public Sub OnAfterSubDimensionChange(ByVal model As IDualModelo, ByVal DimensionName As String, ByVal SubDimension As Long, ByVal NewValue As Double)
```

Este evento se dispara, ya sea en PrefCAD o PrefGest, después de cambiar las subcotas de un modelo en concreto.

```
Public Sub OnAfterUserChangedOption(ByVal model As IDualModelo, ByVal OptionName As String, ByVal OptionValue As String, ByVal element As IDualElemento)
```

Este evento salta después de que el usuario cambia el valor de una opción. Desde este, podríamos programar, por ejemplo, si el cambio de esta opción es correcto, o dar un valor específico a una opción en concreto cuando el modelo cumpla una serie de requisitos.

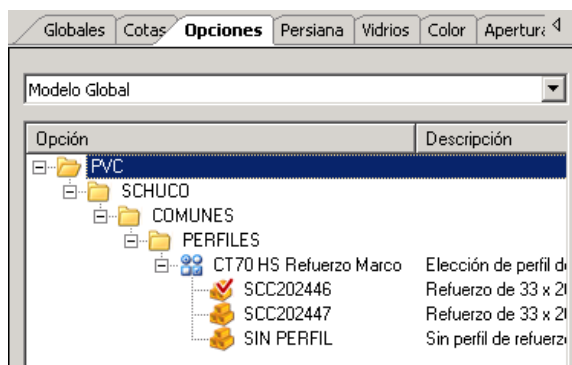


Figura 5. Árbol de opciones de un modelo.

```
Public Sub OnBeforeCalculate(ByVal model As IDualModelo)
```

Este evento se dispara justo antes de que en PrefCAD o en PrefGest se calculen o generen los materiales del modelo.

```
Public Sub OnBeforeDimensionChange(ByVal model As IDualModelo, ByVal DimensionName As String, ByRef NewValue As Double, Apply As Boolean)
```

Este evento se dispara, ya sea en PrefCAD o PrefGest, antes de cambiar las cotas de un modelo en concreto.

```
Public Sub OnBeforeGenerateMaterials(ByVal model As IDualModelo)
```

Este evento salta justo antes de generar materiales de un modelo en concreto ya sea en PrefCAD o PrefGest.

```
Public Sub OnBeforeLoad(ByVal model As IDualModelo)
```

Este evento se dispara antes de cargar un modelo en concreto, ya sea en PrefCAD o PrefGest.

```
Public Sub OnBeforePropertiesDialog(ByVal model As IDualModelo, ShowDialog As Boolean)
```

Este evento se ejecuta antes de abrir la ventana de propiedades del modelo, ya sea desde PrefGest o desde PrefCAD.

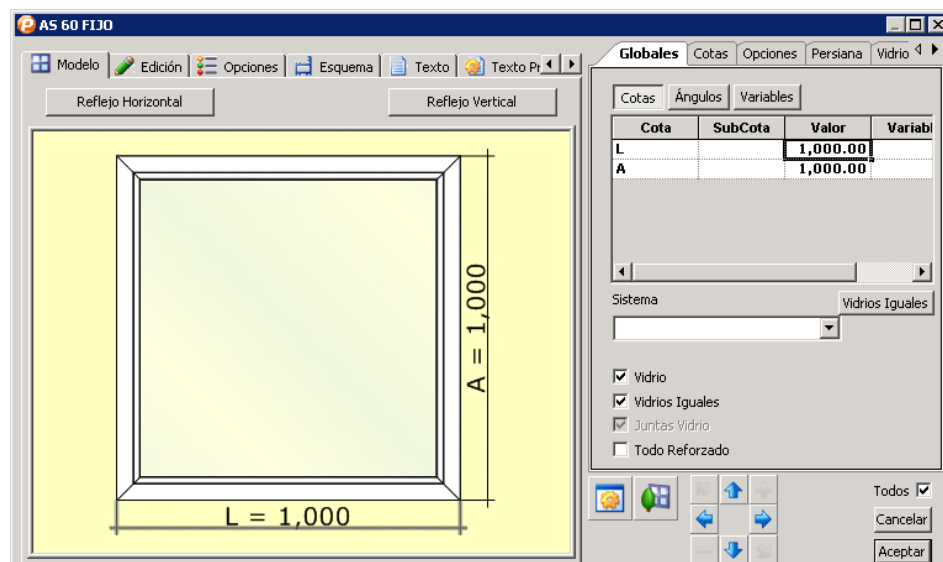


Figura 6. Ventana de propiedades de un modelo.

```
Public Sub OnBeforeSubDimensionChange(ByVal model As IDualModelo, ByVal DimensionName As String, ByVal SubDimension As Long, ByVal NewValue As Double, Apply As Boolean)
```

Este evento salta antes de cambiar las subcotas de un modelo específico, ya sea en PrefCAD o PrefGest.

```
Public Sub OnCalculateHandleHeight(ByVal model As IDualModelo, ByVal Hueco As IHueco, HandleHeight As Single, Apply As Boolean)
```

Se dispara al calcular la altura de la maneta, por si se quiere modificar esta altura.

```
Public Sub OnCalculateHandleHeightList(ByVal model As IDualModelo, ByVal Hueco As IHueco, HandleHeights As Variant)
```

También se ejecuta al calcular la altura de la maneta. Es útil si se quiere dar al usuario una lista de alturas entre las que puede elegir para esa maneta, en lugar de introducir cualquier valor.

```
Public Sub OnDrawHandle(ByVal model As IDualModelo, ByVal Hueco As IHueco, ByVal Imagen As IModelImage, Drawn As Boolean)
```

Se ejecuta cuando se dibuja la maneta en una hoja. Útil para cuando se quiere modificar el dibujo de la maneta para una hoja en concreto, por ejemplo en hojas abatibles, en las que el programa no dibuja maneta por defecto.

```
Public Sub OnDrawHinges(ByVal model As IDualModelo, ByVal Hueco As IHueco, ByVal Imagen As IModelImage, Drawn As Boolean)
```

Se ejecuta cuando se dibujan las bisagras en una hoja.

```
Public Sub OnLoad(ByVal model As IDualModelo)
```

Este evento se ejecuta al buscar cargar un modelo en concreto, ya sea en PrefCAD o PrefGest.

```
Public Sub OnModelIncluded(ByVal model As IDualModelo, ByVal NewModelCode As String, ByVal NewModelFamily As String, ByVal NewModelDescription As String, ByVal NewModelText As String)
```

Este evento al agregar a nuestro modelo en construcción una plantilla con la opción “Nuevo contorno” ó “Insertar modelo” (con la funcionalidad del Fast Offering o Presupuestado rápido), ya sea desde PrefCAD ó desde PrefGest.

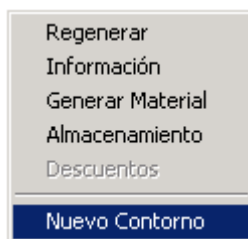


Figura 7. Opción “Nuevo Contorno”.

```
Public Sub OnNew(ByVal model As IDualModelo)
```

Este evento salta al crear un modelo nuevo en PrefCAD.

```
Public Sub OnSave(ByVal model As IDualModelo, ByVal XMLCode As String)
```

Este evento se ejecuta al salvar un modelo en PrefCAD.


```
Public Sub OnSpecialUserInterface(ByVal model As Modelo, ByVal SelectedHole As Hueco)
```

Este evento se dispara cuando el usuario pulsa, dentro de la aplicación de PrefCAD, el botón de conexión industrial.



Figura 8. Botón de conexión industrial.

```
Public Sub OnValidatePropertiesDialog(ByVal model As IDualModelo, Apply As Boolean)
```

Este evento se dispara al pulsar el botón aceptar dentro de la pantalla de propiedades del modelo, bien sea desde PrefCAD o desde PrefGest donde la hayamos presentado.

```
Public Sub OnBeforeCalculateMachining(ByVal model As IDualModelo)
```

Este evento salta antes de que el programa calcule las operaciones necesarias para que las maquinas realice los cortes necesarios para producir el modelo.

Este evento solo se ejecuta cuando:

- El modelo este dentro de un pedido.
- Que este pedido tenga activada la fecha “Preparado para producción”.

The screenshot shows a window titled "PEDIDO" with a sub-header "Número 2/2" and "Nº Pedido 1". It features a tabbed interface with tabs for "Documento", "Cliente", "Dir. Envío", "Dir. Factura", "Estado", "Subtotales", and "Planificación". The "Estado" tab is active, displaying a "Seguimiento" section with date pickers for "Revisión", "Necesidades", "Actualización Stock", and "Preparado producción" (which is checked). Below this is a "Facturación" section with date pickers for "Pagado", "Contabilizado", and "Facturado", along with checkboxes for "A Procesar" and "Precio Cerrado". A "Cancelación" section includes a date picker for "Cancelado" and a dropdown for "Motivo".

Figura 9. Pedido con el check “Preparado para producción” activado.

Dentro de las variables globales de PrefWise, la opción “Calcular Mecanizados” esté activada.

```
Public Sub OnAfterCalculateMachining(ByVal model As IDualModelo)
```

Este evento se ejecuta después de que el programa calcule las operaciones para las maquinas, pero sin desdoblarse las operaciones específicas para la geometría de los perfiles, es decir, las suboperaciones correspondientes a cada una de las operaciones.

Este evento solo se ejecuta cuando:

- El modelo este dentro de un pedido.
- Que este pedido tenga activada la fecha “Preparado para producción”.
- Dentro de las variables globales de PrefWise, la opción “Calcular Mecanizados” esté activada.

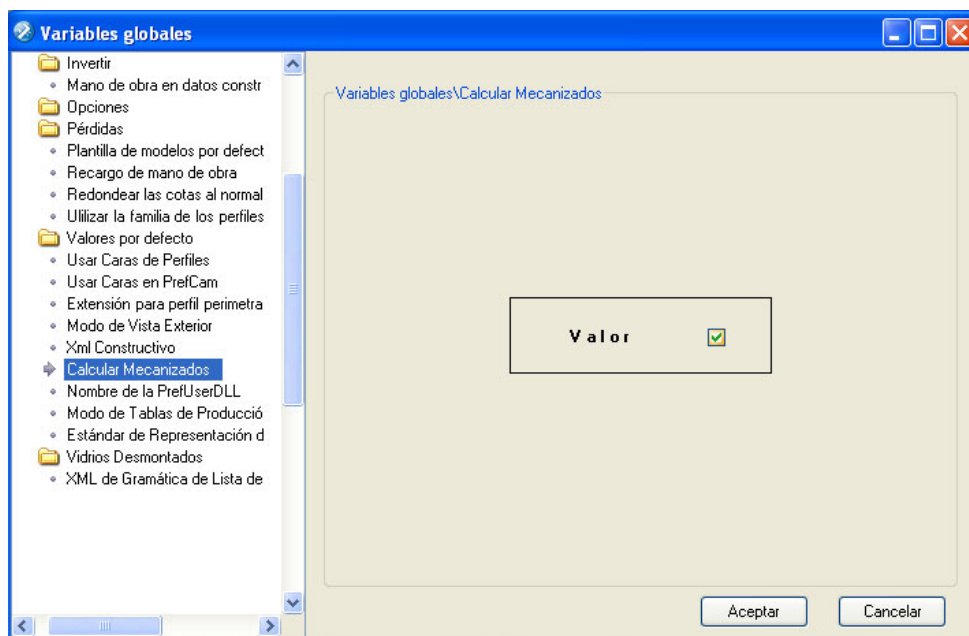


Figura 10. “Calcular Mecanizados” activado.

```
Public Sub OnAfterOrderMachining(ByVal model As IDualModelo)
```

Este evento se dispara cuando el programa ha calculado todas las operaciones y suboperaciones necesarias para que las máquinas realicen los cortes necesarios para producir el modelo.

Este evento solo se ejecuta cuando:

- El modelo este dentro de un pedido.
- Que este pedido tenga activada la fecha “Preparado para producción”.
- Dentro de las variables globales de PrefWise, la opción “Calcular Mecanizados” esté activada.

```
Public Sub OnColorChanged(ByVal model As IDualModelo)
```

Este evento salta al cambiar el color del modelo en concreto, bien sea desde PrefGest ó desde PrefCAD.

```
Public Sub OnCustomConstructiveLink(ByVal model As IDualModelo, ByVal element As IDualElemento, customScript As String)
```

Este evento sirve para asociar a cualquier elemento del modelo un escandallo por vínculo constructivo. Este evento, no devuelve los vínculos constructivos del modelo. Solo sirve para añadirlos.

```
Public Sub OnCustomCommonLink(ByVal model As IDualModelo, ByVal element As IDualElemento, customScript As String)
```

Este evento sirve para asociar a cualquier elemento del modelo un escandallo por vínculo común. No devuelve los vínculos comunes del modelo, por lo que no es informativo. Solo sirve para añadirlos.

```
Public Sub OnPDistance(ByVal model As IDualModelo, ByVal master As ITramo, ByVal slave As ITramo, ByVal typeDist As TipoDescuento, ByRef dist As Double, ByRef zDist As Double, ByRef zAngle As Double)
```

Este evento se ejecuta cuando, en un descuento en PrefWise, tenemos activada la opción “Lanza Evento”.

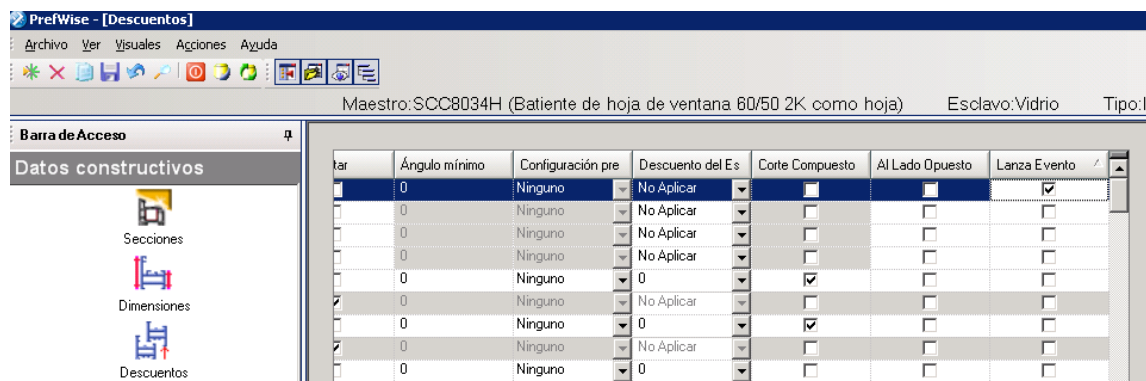


Figura 11. Opción “Lanza Evento” activada en un descuento.

```
Public Sub OnGlazingRule(ByVal model As IDualModelo, ByVal glazedPiece As ITramo, ByVal glassThickness As Double, ByRef HGlazingBead As String, ByRef VGlazingBead As String, ByRef profileGasket As String, ByRef glazingbeadGasket As String, ByRef glazingCutType As glazingCutType)
```

Este evento se ejecuta cuando, en las reglas de acristalamiento en PrefWise, tenemos activada la opción “Lanza Evento”.

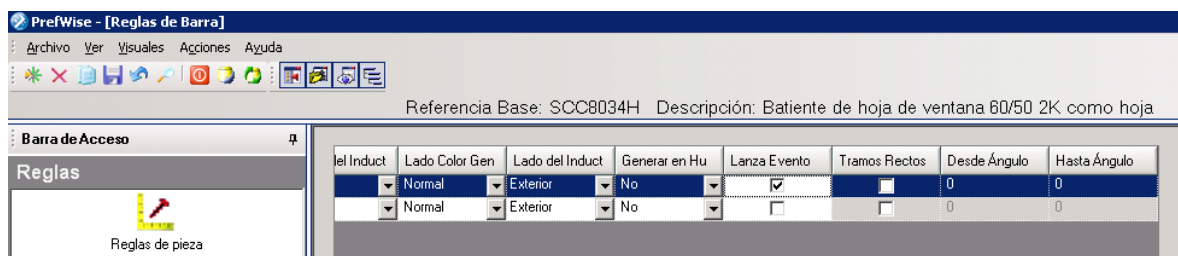


Figura 12. Opción “Lanza Evento” activada en una regla de barra.

```
Public Sub OnRodRule(ByVal model As IDualModelo, ByVal AInductor As ITramo, ByVal BInductor As ITramo, ByVal Inducted As ITramo)
```

Este evento se ejecuta cuando, en una regla de barra en PrefWise, tenemos activada la opción “Lanza Evento”.

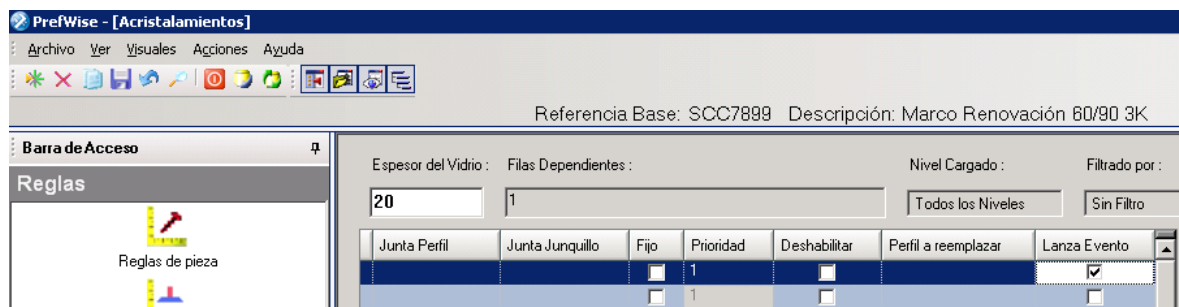


Figura 13. Opción “Lanza Evento” activada en un acristalamiento.

En este módulo encontraremos el evento “OnDrawViewControlHandle”. Debemos de tener en cuenta que este evento esta obsoleto, y que actualmente no se ejecuta.

5. PricesModule

Este módulo sirve para modificar el precio de un material. En el encontramos un evento capaz de modificar el precio de un material. Veamos este evento.

5.1. Eventos Principales de PricesModule

```
Public Sub OnTariffPartialAmountsModification(ByRef matPriceItem As
IMaterialPriceItem, ByRef modified As Boolean)
```

Este evento se lanza siempre y cuando en la tarifa y en el material está activado el campo “Habilitar el cálculo personalizado del importe”.

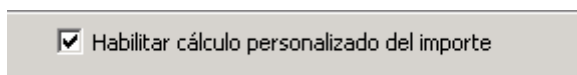


Figura 14. Opción “Habilitar el cálculo personalizado del importe” activada.

6. ProductionModule

Este módulo especifica el código y eventos para el módulo de producción en PrefGest. Se encarga básicamente de crear/modificar la información de los lotes. Se considera un módulo relevante dentro de nuestro proyecto VB. Veamos sus eventos más representativos.

6.1. Eventos Principales de ProductionModule

```
Public Sub Production_AfterCalculateMaterialNeeds(ByVal Production As
PrefGest.IProduction, ByVal ProductionLot As Long, ByVal MNSet As Long)
```

Este evento se dispara después de calcular las necesidades del material de un lote.

```
Public Sub Production_AfterCreateProductionLot (ByVal Production As PrefProduction.IPrefProduction, ByVal ProductionLot As Long)
```

Este evento se lanza justo después de crear un lote de producción.

Figura 15. Datos de optimización.

```
Public Sub Production_AfterDeleteProductionLot (ByVal Production As PrefProduction.IPrefProduction, ByVal ProductionLot As Long)
```

Este evento se dispara justo después de eliminar un lote de producción.

```
Public Sub Production_AfterRemoveCalculationDate (ByVal Production As PrefGest.IProduction, ByVal ProductionLot As Long)
```

Este evento salta cuando desmarcamos la fecha en la que se ha calculado el lote. En este evento se puede programar, por ejemplo, que cuando esto suceda, elimine los ficheros que se han creado para la maquina al calcular dicho lote, para asegurarnos que efectivamente no existen y no los tenemos guardados en disco.

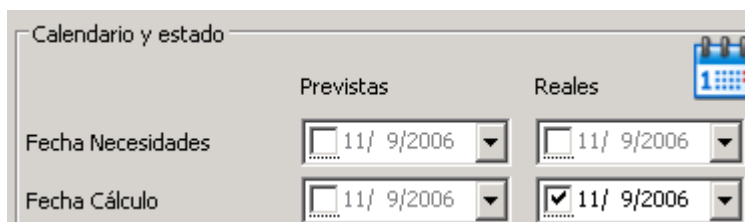


Figura 16. Check "Fecha de cálculo".

```
Public Sub Production_AfterSave (ByVal Production As PrefGest.IProduction, ByVal ProductionLot As Long)
```

Este evento se dispara después de calcular un lote en concreto.

```
Public Sub Production_AfterSaveLotContent (ByVal Production As PrefGest.IProduction, ByVal ProductionLot As Long, ByRef xml As msxml2.IXMLDOMDocument2)
```

Este evento se dispara después de guardar (que no calcular) el contenido de un lote, es decir, al pulsar el botón guardar. Este evento pasa como parámetro el contenido del lote en un XML, pero dicho lote es puramente informativo, es decir, no se podrá modificar dicho XML.



Figura 17. Botón guardar.

```
Public Sub Production_BeforeCreateProductionLot (ByVal Production As PrefProduction.IPrefProduction, ByRef xml As msxml2.IXMLDOMDocument2, ByRef Cancel As Boolean)
```

Este evento se dispara antes de crear el lote de producción.

```
Public Sub Production_BeforeDeleteProductionLot (ByVal Production As PrefProduction.IPrefProduction, ByVal ProductionLot As Long, ByRef Cancel As Boolean)
```

Este evento se lanza antes de eliminar un lote de producción en concreto.

```
Public Sub Production_BeforeOptimizeProductionLot (ByVal Production As PrefProduction.IPrefProduction, ByVal ProductionLot As Long, ByRef xml As msxml2.IXMLDOMDocument2, ByRef XMLModified As Boolean)
```

Este evento se dispara justo antes de optimizar el lote de producción. Desde este evento podemos controlar, por ejemplo, que efectivamente todas las longitudes de corte que tenemos en nuestro lote (el evento pasa un xml con estas longitudes) no están fuera del rango de longitudes de corte que es capaz de cortar nuestra maquina (ya que este xml es modificable desde este evento).

```
Public Sub Production_BeforeSaveLotContent(ByVal Production As  
PrefGest.IProduction, ByVal ProductionLot As Long, ByRef xml As  
msxml2.IXMLDOMDocument2, ByRef Cancel As Boolean)  
End Sub
```

Este evento se dispara antes de guardar (que no calcular) el contenido de un lote.

```
Public Sub Production_CreatedXMLProductionLot(ByVal Production As  
PrefProduction.IPrefProduction, ByVal ProductionLot As Long, ByRef xml As  
msxml2.IXMLDOMDocument2)
```

Se lanza cuando se crea el XML con la optimización del lote. El evento también tiene el XML descriptivo de cada modelo. Desde este evento podemos, por ejemplo, especificar una asignación especial de carros y/o estantes, si es que para un caso específico queremos que se realice de forma especial, no utilizando la asignación que da por defecto el programa.

```
Public Sub Production_ExtraUI(ByVal Production As PrefGest.IProduction, ByVal  
ProductionLot As Long)
```

Este evento se dispara cuando el usuario pulsa, dentro de la pestaña del módulo de producción en PrefGest, el botón de conexión industrial.



Figura 18. Botón de conexión industrial.

```
Public Sub Production_PrintCustomizedReport(ByVal Production As  
PrefGest.IProduction, ByVal ProductionLot As Long, ByVal Layout As String,  
XMLDoc As String)
```

Este evento se lanza cuando se va a imprimir un listado del apartado personalizado en el módulo de producción.

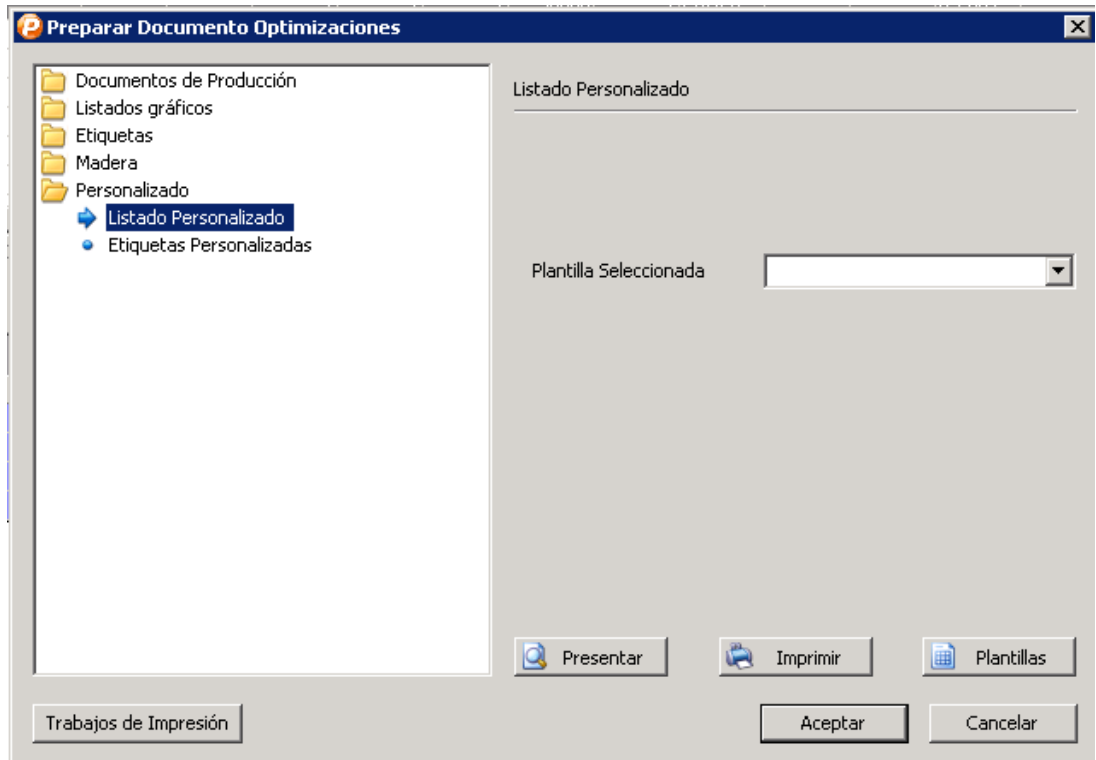


Figura 19. Pantalla para preparar un documento de optimizaciones.

```
Public Sub Production_ValidateSubModels(ByVal Production As
PrefGest.IProduction, ByRef xml As msxml2.IXMLDOMDocument2)
```

Se ejecuta cuando pasamos los submodelos desde la grid o tabla superior (donde se encuentran todos los submodelos a producir) a la grid o tabla donde vamos almacenando los submodelos que compondrán el lote que estamos realizando, dentro de la pantalla del módulo de producción de PrefGest. El xml que pasa el evento, contiene información del submodelo que se esta seleccionado en ese momento, así como el del resto de submodelos que tenemos seleccionados en la grid o tabla inferior, que ya componen nuestro lote.

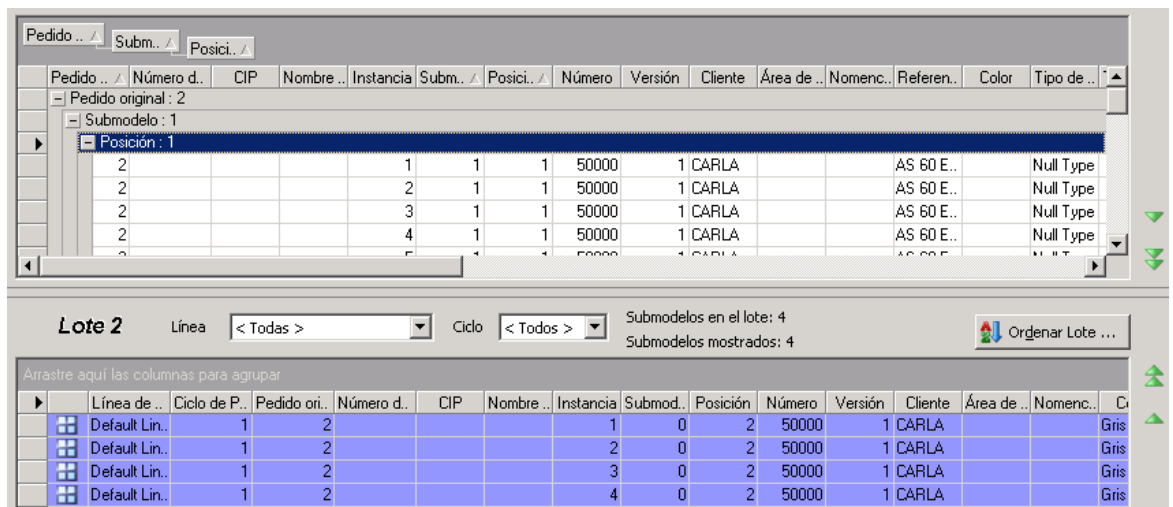


Figura 20. Agrupación de modelos para un lote.

7. PurchasesModule

Este módulo engloba los eventos específicos para el apartado de compras de PrefGest. Dentro de los eventos que comprende este módulo, destacamos.

7.1. Eventos Principales de PurchasesModule

```
Public Sub Purchases_AfterSave(ByVal Purchases As PrefGest.IPurchases, ByVal Number As Long, ByVal Numeration As Long)
```

Este evento se lanza después de guardar un PAF de compras en PrefGest.

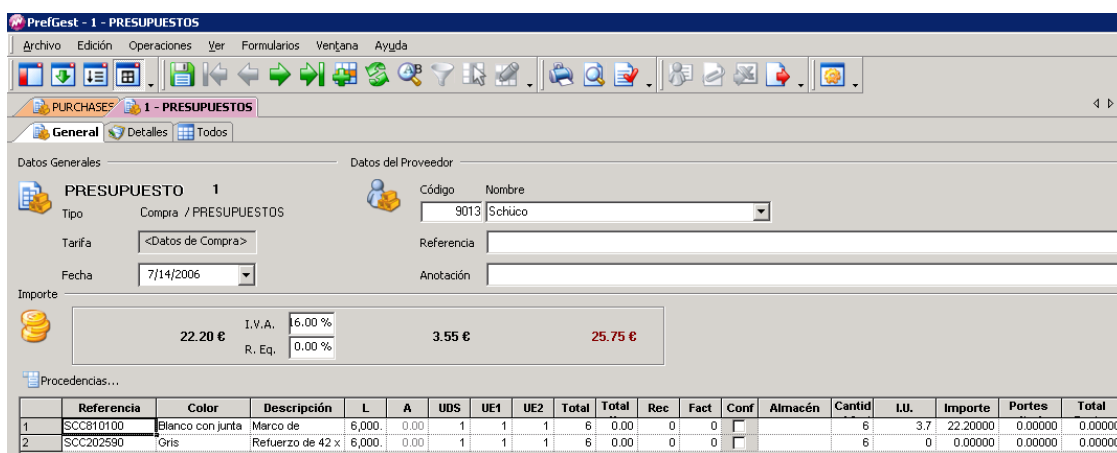


Figura 21. PAF de compras en PrefGest.

```
Public Sub Purchases_BeforeSave(ByVal Purchases As PrefGest.IPurchases, ByVal Number As Long, ByVal Numeration As Long, Cancel As Boolean)
```

Este evento se dispara antes de guardar un documento de compras dentro de PrefGest.

```
Public Sub Purchases_DefineVariablesAndFields(ByVal Purchases As PrefGest.IPurchases, ByVal hJob As Long, ByVal DocType As PurchasesPrintDocumentType, ByVal Layout As String)
```

Se ejecuta cuando se quieren añadir variable y/o campos a la plantilla del pedido de compras.

```
Public Sub Purchases_ExtraUI(ByVal Purchases As PrefGest.IPurchases, ByVal Number As Long, ByVal Numeration As Long)
```

Este evento se dispara cuando el usuario pulsa, dentro de un PAF de compras en PrefGest, el botón de personalización.



Figura 22. Botón de personalización.

```
Public Sub Purchases_PrintCustomizedReport (ByVal Purchases As
PrefGest.IPurchases, ByVal Number As Long, ByVal Numeration As Long, ByVal
Layout As String, XMLDoc As String)
```

Este evento se lanza cuando se va a imprimir un listado del apartado personalizado en el módulo de compras.

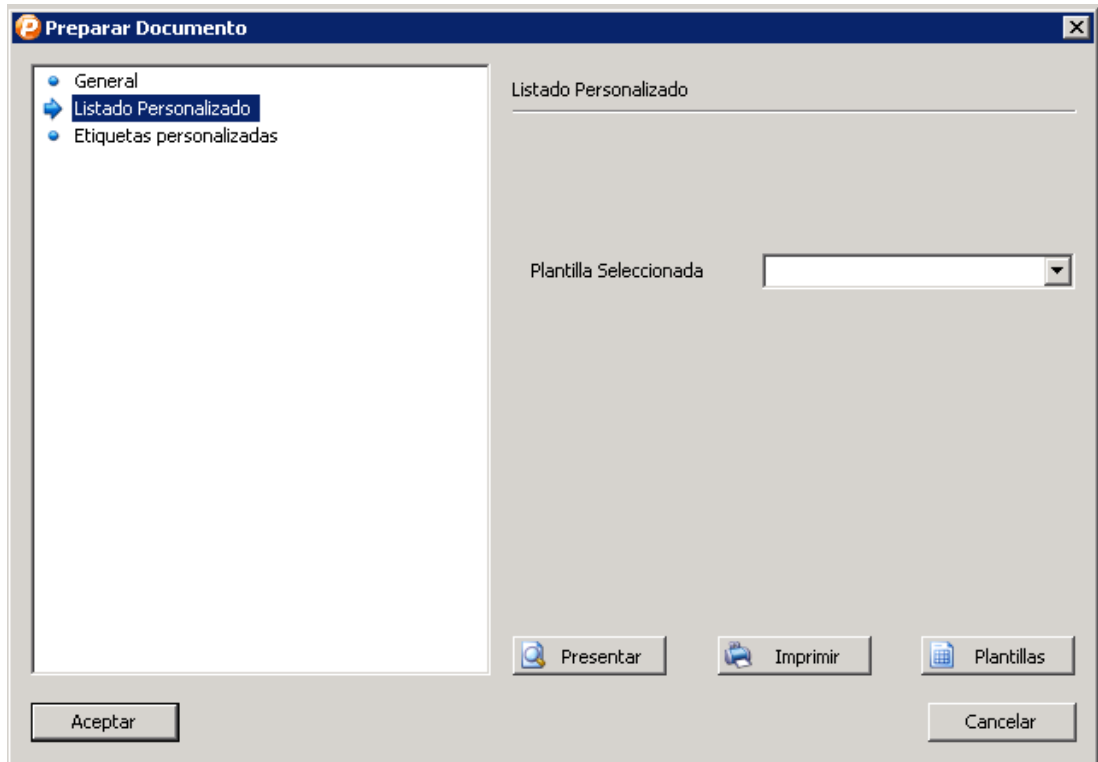


Figura 23. Pantalla para preparar documentos.

```
Public Sub Purchases_RecalculatePerVolume (ByVal Purchases As
PrefGest.IPurchases, ByVal Number As Long, ByVal Numeration As Long, ByVal
SupplierTariff As String, ByVal TariffExceptions As
PrefPrices.ITariffExceptions, ByVal RawMaterial As
PrefCAD.IRawMaterialVolumeItems)
```

Se lanza cuando se selecciona la opción “calculo de precios según volumen” del menú “Operaciones”, en un PAF de compras. Se utiliza básicamente para modificar el precio del material según su peso en este tipo de documentos.

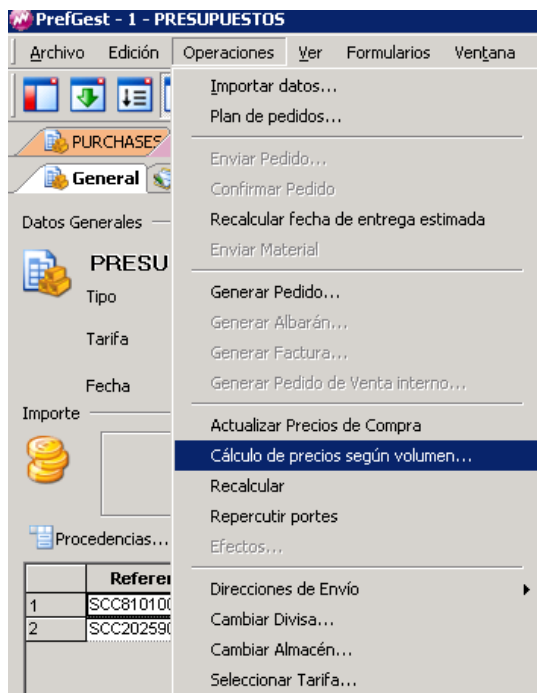


Figura 24. Opción "Cálculo de precios según volumen".

```
Public Sub Purchases_SetFields(ByVal Purchases As PrefGest.IPurchases, ByVal
hJob As Long, ByVal Number As Long, ByVal Numeration As Long, ByVal Position As
Long, ByVal DocType As PurchasesPrintDocumentType, ByVal Layout As String)
```

Se ejecuta cuando le damos valor a aquellos campos que hemos añadido a la plantilla del pedido de compras.

```
Public Sub Purchases_SetVariables(ByVal Purchases As PrefGest.IPurchases, ByVal
hJob As Long, ByVal Number As Long, ByVal Numeration As Long, ByVal DocType As
PurchasesPrintDocumentType, ByVal Layout As String)
```

Se ejecuta cuando le damos valor a aquellas variables que hemos añadido a la plantilla del pedido de compras.

8. DataManagerModule

Este módulo engloba los eventos específicos de la aplicación PrefDBManager. Contiene estos eventos:

8.1. Eventos de DataManagerModule

```
Public Sub PrefDatabase_AfterLoadStructure(ByVal PrefDatabase As
PrefDataManager.IPrefDatabase)
```

Este evento se lanza después de cargar la estructura de la base de datos (desde PrefDBManager o PrefBackup).

Útil si el usuario necesita añadir una estructura personalizada a la estructura de la base de datos de PrefSuite.

```
Public Sub PrefDBManager_AfterExecutePriority(ByVal PrefDBManager As  
PrefDataManager.IPrefDBManager, ByVal Conn As  
PrefDataManager.PrefOLEDBConnection, Priority As Long)
```

Las tareas que ejecuta PrefDBManager van siguiendo un orden o una “prioridad”.

Este evento se ejecuta después de cada prioridad.

Las prioridades de 1 a 10 son las internas de la aplicación, por lo tanto, si se necesita añadir código antes o después de que se ejecute PrefDBManager, tendrá que hacerse para Priority = 1 o Priority = 10 respectivamente.

Desde este módulo se puede por ejemplo, lanzar la ejecución de un script SQL.

```
Public Sub PrefDBManager_BeforeExecutePriority(ByVal PrefDBManager As  
PrefDataManager.IPrefDBManager, ByVal Conn As  
PrefDataManager.PrefOLEDBConnection, ByVal Priority As Long)
```

Tiene el mismo sentido que el evento anterior, pero se ejecuta antes de cada prioridad.