

Personalización de la exportación de Revit

Descripción detallada de la utilización de filtros y expresiones para crear criterios de medición personalizados en la exportación de Cost-It.

El objetivo prioritario de Cost-It es generar un presupuesto a partir del modelo Revit con el mayor grado de automatismo, aprovechando la gran cantidad de información estructurada que contiene el modelo.

Para ello, en cada categoría se puede:

- Elegir un criterio de medición predefinido distinto al de defecto
- Modificar el contenido de los campos de dimensiones y de la cantidad total

Cost-It 2020 aumenta esta personalización mediante dos nuevas posibilidades.

Asignar más de una unidad de obra a cada tipo

La duplicación de criterios de medición permite asignar dos o más unidades de obra a los tipos de cada categoría, cada uno con su propia forma de medición y código.

Si hay tipos de una misma categoría que requieren criterios de medición diferentes, se pueden medir todos de las distintas formas y eliminar en Presto los que sobran.

Expresiones y filtros

La utilización de expresiones en lugar de códigos fijos, junto con la duplicación de criterios de medición, permite personalizar aún más la exportación, aplicando cada criterio a los tipos o elementos cuyos parámetros cumplen determinadas condiciones y asignando libremente sus unidades de obra.

▲	ID	Categoría	Código	Unidad	Medida	Longitud	Anchura	Altura
11	<input checked="" type="checkbox"/>	2001330	Pilares estructurales	// Posibles pilares que no cumplan los filtros de los criterios de ...	u	Número		
12	<input checked="" type="checkbox"/>	2001330	Pilares estructurales de acero	// Pilares de acero // Los pilares que no pasen este filtro no se ...	m	Longitud.(L)	b/OD/Diameteri...	h/Nominal h
13	<input checked="" type="checkbox"/>	2001330	Pilares estructurales de hormig...	// Pilares de hormigón, seleccionados por el nombre del material...	m3	Volumen	b/OD/Diameteri...	h/Nominal h
14	<input checked="" type="checkbox"/>	2001330	Pilares estructurales_encofrado	// Encofrado de pilares circulares de hormigón medidos por m R...	m	Longitud.(L)	b/OD/Diameteri...	h/Nominal h
15	<input checked="" type="checkbox"/>	2001330	Pilares estructurales_armado	// Armado de pilares de hormigón // La cuantía se pone en un ...	kg	Usuario	50	Volumen

Expresiones y filtros personalizados para medir el modelo educativo de Revit

Se pueden utilizar filtros y expresiones, por ejemplo, para:

- Asignar códigos de unidad de obra a tipos o elementos durante la exportación.
- Asignar una unidad de obra distinta en función de uno o más parámetros, como la altura de cada muro.
- Medir de distinta manera tipos de una misma categoría, como los elementos de hormigón por volumen y los de acero por longitud o peso.
- Diferenciar los elementos de nueva creación y los demolidos del mismo tipo.
- Medir elementos no modelados, como los encofrados.

Para escribir distintos criterios, que pueden ser complementarios o excluyentes, se pueden duplicar los criterios de cada categoría las veces que sean necesarias.

Proceso general

Las expresiones se escriben en el campo "Código" de la pestaña "Categorías", accesible en la pestaña "Script del código".

- Pueden usar las funciones y demás recursos habituales del lenguaje JavaScript
- Pueden utilizar el valor de cualquier parámetro de proyecto, tipo o elemento del modelo.

El resultado de las expresiones debe ser uno o más códigos de unidad de obra. Los elementos que cumplen las condiciones de cada expresión se miden con el criterio de medición de la línea, como las demás categorías.

Para empezar a utilizar expresiones abra con Cost-It la hoja de configuración "Modelo Educativo de Revit.CostItLayout", válida para medir el modelo educativo de Revit, analice los distintos casos y copie el código para crear sus propias expresiones.

La ventana "Vista preliminar" de la pestaña "Categorías" puede filtrar para su comprobación los elementos que satisfacen los filtros aplicados a un criterio de medición con la opción "Filtrar por script" del menú contextual.

Mientras se escribe en la ventana "Script del código", cada expresión se puede ir probando mediante dos opciones.

Evaluar 1

Aplica la expresión a un solo elemento para comprobar rápidamente que la sintaxis es correcta.

```
// Encofrado de pilares circulares de hormigón medidos por m
Result = null;
// Se miden en longitud del mismo diámetro porque la unidad de obra es así
// Lo cual exige separarlos por diámetros, en este caso, por tipos
switch (${Tipo})
{
  case "300mm":
    Result = "E05HSC030|m|ENCOFRADO PILARES CIRCULARES VISTO CARTÓN D=30 cm";
    break;
  case "450mm":
    Result = "E05HSC060|m|ENCOFRADO PILARES CIRCULARES VISTO CARTÓN D=45 cm";
    break;
}

/-- [141009] 300mm -----
[Tipo, ELEM_TYPE_PARAM] (string) = 300mm
[b, 140573] (double) = 300
[Result] = E05HSC030|m|ENCOFRADO PILARES CIRCULARES VISTO CARTÓN D=30 cm
```

Reglas para medir encofrados circulares y resultado de la evaluación

Evaluar todos

Aplica la expresión a todos los elementos de la categoría y presenta todos los valores de los parámetros utilizados.

[Anterior / Siguiente](#)

Permiten cambiar de categoría sin necesidad de pasar a la pestaña "Categorías".

Reglas de las expresiones

Sintaxis

La sintaxis se basa en JScript, el intérprete de Microsoft para el estándar ECMA 262, conocido como ECMAScript y que se encuentra en numerosas fuentes, como:

<https://www.w3schools.com/js/>

<https://www.ecma-international.org/ecma-262/>

Se soportan en general declaraciones, métodos, funciones y excepciones de JScript, excepto las que no tienen sentido en el entorno de trabajo, como clases o *import*.

RIB Spain no proporciona soporte de programación.

Referencia a parámetros

Para definir las condiciones de los elementos a los que se desea asignar cada unidad de obra se puede incluir en las expresiones cualquier parámetro de Revit, insertándolo entre los caracteres especiales "\$[" y "]"\$".

Se puede usar también el nombre interno, visible en Presto en el campo "Variables.Descripción", lo cual es útil cuando distintos parámetros comparten un mismo nombre de casa al usuario.

Por ejemplo, la altura de las ventanas puede ser:

`$(Altura)$`

`$(WINDOW_HEIGHT)$`

Cost-It busca en primer lugar el parámetro como parámetro de ejemplar. Si no existe o no tiene un valor en la base de datos, se busca como parámetro de tipo. Cuando el mismo parámetro puede ser de proyecto, tipo o elemento y se quiere evitar la ambigüedad o alterar la prioridad anterior se debe usar el prefijo "Project.", "Type." o "Item." que corresponda.

Por ejemplo, el parámetro "Código de montaje" del tipo existe también como parámetro de ejemplar, pero siempre está vacío. En este caso, es necesario utilizar:

`$(Type.Código de montaje)$`

Si se utiliza un prefijo en un parámetro, también deberá incluirse en los parámetros que se evalúen después y les corresponda un prefijo diferente. Es obligatorio añadir un prefijo a los parámetros cuyo nombre comienza por un número.

El parámetro de Revit usado debe existir en todos los elementos de la categoría, porque no se exportarán los elementos para los que no exista.

Al utilizar parámetros numéricos se deben usar las unidades de medida definidas en el modelo, tal y como aparecen en cualquiera de las ventanas de Revit.

Código de la unidad de obra

El código deseado para las unidades de obra de los elementos se devuelve mediante una variable "Result".

Hay dos casos especiales:

Result = "" (vacío)

- El elemento se mide utilizando el código de defecto definido en la pestaña "Opciones".

Result = null

- El elemento no se mide. Puede servir para saltarse elementos que se miden en otro criterio de medición.

El elemento tampoco se mide si el resultado de la expresión genera un error porque contiene un parámetro de Revit que no existe para ese elemento o por otra causa.

Por ejemplo, esta expresión mide los elementos de un determinado tipo con un código fijo y el resto con el código de defecto.

```
Result = "";  
if ($[Tipo]$ == "Partición interior")  
{  
    Result = "E07P010";  
}
```

Expresiones de una línea

Si la expresión figura en una sola línea se trata como una sentencia y no es necesario utilizar la variable "Result".

La misma expresión del ejemplo, en una sola línea:

```
[$[Tipo]$ == "Partición interior" ? "E07P010" : ""
```

Código ampliado

El código devuelto puede opcionalmente contener la unidad de medida, el resumen y el precio unitario siguiendo el formato del registro "~C" del formato FIE, por ejemplo:

```
"E04AB040 | kg | ACERO CORRUGADO | 2,35"
```

Cuando la unidad de medida del código ampliado coincide con las unidades de medida de uno los criterios de medición predefinidos para la categoría, se utilizará ese criterio de medición en vez del que estuviera seleccionado para la categoría.

Para facilitar la tarea de añadir en un script los códigos ampliados, en la lista de expresiones de Presto se incluye "Registro ~C del formato FIEBDC". Si emplea esta expresión en una columna de usuario, podrá copiar los códigos ampliados desde una base de precios y pegarlos en el script.

Ejemplos de expresiones

Además de los ejemplos descritos a continuación, al pulsar con el botón derecho del ratón en la ventana "Script del código", en el menú contextual encontrará los ejemplos

"Asignación de códigos según un parámetro", "Casos anidados para discriminar por dos parámetros", "Condición if else" y "Búsqueda en el nombre", para que pueda utilizarlos como punto de partida para crear su propio script.

Asignar un código de unidad de obra a cada tipo

No requiere incluir los códigos en el modelo ni añadirlos posteriormente en el presupuesto.

Utilizaremos en cada categoría un script similar al siguiente.

```
Result = "";
switch ($[Tipo])
{
  case "Concrete Deck - Tapered Insulation":
    Result = "E09AFB010";
    break;
  case "Sloped Glazing":
    Result = "E16UTA010";
    break;
  case "Generic - 400mm":
    Result = "E09CPB010";
    break;
  case "Generic - 75mm":
    Result = "E09OTH120";
    break;
}
```

Medir los muros de hormigón en volumen y el resto por superficie

En este ejemplo identificamos los muros de hormigón buscando la palabra "HORMIGÓN" en el material estructural y suponemos que cada tipo tiene el código adecuado definido en el modelo.

Para definir dos criterios de medición distintos hay que duplicar la categoría "Muros" y crear dos criterios de medición, el de defecto y por volumen.

En el criterio de medición por superficie excluimos los elementos de hormigón. Lo demás se mide con el criterio de defecto:

```
Result = "";
var material = "";
if (typeof($[Material estructural]) == "string")
{
  material = $[Material estructural];
  if (material.search(/HORMIGÓN/i) != -1)
  {
    Result == null;
  }
}
```

Utilizamos una variable intermedia "material" para comprobar el material de los muros, porque el parámetro "Material estructural" no está definido para los muros cortina y en ese caso Revit lo trata como un parámetro numérico con valor -1.

Si el material se deduce del nombre del tipo, se puede utilizar directamente el tipo en las expresiones, en vez del material estructural.

En el criterio de medición por volumen se miden sólo los elementos de hormigón.

```
Result = null;
if ($[Material estructural]$.search(/HORMIGÓN/i) != -1)
{
    Result = "";
}
}
```

Para filtrar los muros de hormigón ya no es necesario utilizar la variable "material", porque los muros cortina se van a descartar igualmente.

Medir los pilotes en longitud y el resto de la cimentación en volumen

Para medir de esta manera el modelo "rac_basic_sample_project.RVT", el ejemplo de vivienda unifamiliar de Revit, duplique la categoría "Cimentación estructural" definiendo los dos criterios de medición con las siguientes expresiones, que en este ejemplo hemos escrito como funciones de una línea.

Para la categoría medida en volumen:

```
($[Descripción de montaje]$.search(/PIPE/i) >= 0) ? null : $[Código de montaje]$
```

Expresión para la categoría medida en longitud:

```
($[Descripción de montaje]$.search(/PIPE/i) >= 0) ? $[Código de montaje]$ : null
```

Para discriminar comprobamos si aparece el término "PIPE" en la descripción de montaje.

Muros existentes y para demolición

Si en un modelo hay elementos existentes, como muros, de los cuales hay que demoler algunos, es conveniente asignarles un tipo distinto. Si, por el contrario, se ha utilizado el mismo tipo para todos los muros y se han asignado a diferente fase, se pueden desglosar duplicando la categoría de muros y utilizando los siguientes scripts.

En una categoría se excluyen los muros existentes sin fase de derribo:

```
Result = "";
if ($[Fase de creación]$ == "Existente" && $[Fase de derribo]$ != "")
{
    Result = null;
}
}
```

En la otra categoría tenemos en cuenta solo los muros con fase de derribo y les asignamos la unidad de obra de demolición:

```
Result = null;
if ($[Fase de derribo]$ != "")
{
    Result = "E01DFC320 | m2 | DEMOLICIÓN LADRILLO HUECO DOBLE";
}
}
```

```
}
```

Medir solo los muros de un tipo que tengan una altura menor o igual a 3 m.

```
Result = null;  
if ($[Tipo]$ == "Tipo de muro" && $[Altura desconectada]$ <= 3000)  
{  
    Result = $[Código de montaje]$  
}
```

Medir el armado y el encofrado no modelado de los muros de hormigón

Para medir elementos que no están modelados directamente hay que duplicar la categoría de base y definir el criterio de medición.

Para medir el encofrado duplicamos la categoría "Muros", la renombramos como "Encofrado muros de hormigón" y en el criterio de medición seleccionamos "Área (LxH)". Filtramos los muros de hormigón por el nombre del material y asignamos el código de la unidad de obra con el siguiente script.

```
Result = null;  
if ($[Material estructural]$.search(/HORMIGÓN/i) != -1)  
{  
    Result = "E04FMM010 | m2 | ENCOFRADO EN MUROS 2 CARAS 3,00 m";  
}
```

Para el armado creamos una nueva categoría, con la cuantía deseada.

Categoría	Unidad	Medida	Longitud	Anchura
Armado de muros de hormigón	kg	Usuario	50	Volumen

Y aplicamos el siguiente script.

```
Result = null;  
if ($[Material estructural]$.search(/HORMIGÓN/i) != -1)  
{  
    Result = "E04AB040 | kg | ACERO CORRUGADO / ARMADO B 500 S/SD";  
}
```

Medir las tuberías clasificándolas en función del sistema y del diámetro

En instalaciones es habitual que el precio unitario de los elementos de un mismo tipo sea diferente en función del valor de uno o más parámetros, como sucede en las tuberías, donde el precio depende del sistema al que pertenece y del diámetro.

Para asignar el código en función de estos dos parámetros, anidamos dos sentencias switch(), una dentro de otra.

```
Result = "";  
switch ($[Nombre de sistema]$)  
{  
    case "Fire Protection Wet 1":  
        switch ($[Diámetro]$.toString())
```

```
{
  case "15":
    Result = "E26ATN010|m|TUBERÍA INCENDIOS ACERO NEGRO |19.5";
    break;
}
break;
case "Sanitary 1":
  switch ([Diámetro].ToString())
  {
    case "32":
      Result = "E20WTV010|m|TUBERÍA PVC SERIE B JUNTA PEGADA D=32
mm|3.9";
      break;
    case "50":
      Result = "E20WTV030|m|TUBERÍA PVC SERIE B JUNTA PEGADA D=50
mm|5.11";
      break;
    case "100":
      Result = "E20WTV060|m|TUBERIA PVC SERIE B JUNTA PEGADA D=110
mm|12.06";
      break;
  }
  break;
}
```