



## ÍNDICE

<b>1.LA BASE</b>	<b>3</b>
1.1. QUÉ ES DYNAMO.	3
1.2. INSTALAR DYNAMO.	4
<b>2. OPERACIONES BÁSICAS</b>	<b>5</b>
2.1. LA CONSOLA DE DYNAMO.	5
2.2. DESCARGAR PAQUETES DE INTERNET	6
2.2.1. PAQUETES RECOMENDADOS	8
2.2.2. DÓNDE SE GUARDAN LOS PAQUETES	10
2.3. QUÉ ES UN NODO	10
2.4. VINCULAR EL MODELO DE REVIT CON DYNAMO.	11
2.4.1. Select model elements.	11
2.4.2. All elements in active view.	11
2.4.3. Categories. All elements of category.	11
2.4.4. Element Types. All elements of type.	12
2.4.5. Levels. All elements at level.	12
2.4.6. Get All Line Styles	12
2.4.6. Get Scope Boxes	13
2.5. OPERACIONES BÁSICAS CON DYNAMO.	14
2.5.1. Codeblocks	16
2.5.2. Números	18
2.5.3. Conversión de elementos para trabajar	19
2.5.3.1. String from Object. Conversión de cualquier objeto en texto.	19
2.5.3.2. String.ToNumber: Conversión del texto en números	20
2.6. INFORMACIÓN EN DYNAMO DESDE REVIT & HACIA REVIT	20
2.7. INFORMACIÓN REVIT-DYNAMO-EXCEL, EXCEL-REVIT-DYNAMO	21
2.7.1. Excel.WriteToFile de Dynamo a Excel	21
2.7.2. Excel.ReadFromFile	22
3. EJEMPLO: TOMAR INFORMACIÓN DE LAS PUERTAS DE UN MODELO Y LLEVARLA A LAS HABITACIONES.	23
4. NOTA FINAL & CONTACTO	33



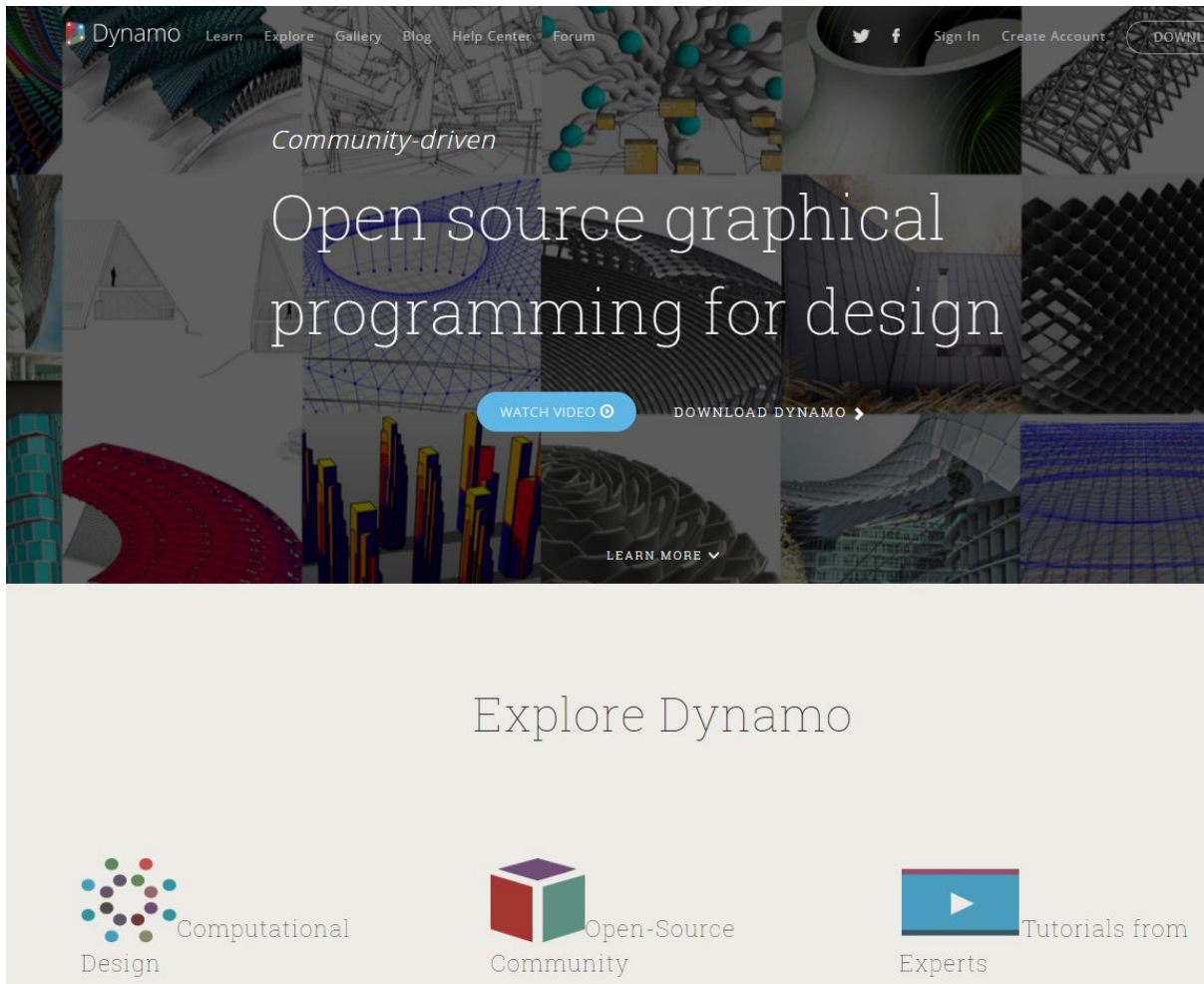
## **1.LA BASE**

### **1.1. QUÉ ES DYNAMO.**

Dynamo es un complemento de Revit. En la actualidad Dynamo se instala automáticamente al instalar Revit. En versiones anteriores de Revit Dynamo no venía incorporado con el programa sino que se instalaba como un complemento externo.

Dynamo es un algoritmo de programación visual que data del 2011 y cuyo diseñador fue Ian Keough. El programa está en constante desarrollo. Actualmente la última versión disponible estable es la 2.8.0 (en cuanto a core) y la 2.6.1 para Revit. El instalador de Dynamo puede descargarse en <http://dynamobuilds.com/>

Asimismo en la web del programa se puede acceder a cantidad de información para formarse en inglés en el manejo de Dynamo: <http://dynamobim.org/learn/>



## RECORTE DE DYNAMOBIM.ORG

### 1.2. INSTALAR DYNAMO.

Al acceder a la web <http://dynamobuilds.com/> podemos comprobar como aparecen dos tipos de versiones del programa. En la web se categorizan en dos grupos:

- Releases: en este grupo están las versiones de Dynamo recomendadas y más estables. La versión más actualizada sería la que aparece en la parte superior y según se baja en la lista la versión estable sería más antigua:
  - Revit 2020 y posteriores: Revit incluye Dynamo y las versiones estables se van actualizando solas.
  - Revit 2019: La versión más actualizada de Dynamo que se puede usar es la 1.3.4 de la versión 1 y de la versión 2 la 2.04



- 
- Revit 2018: La versión más actualizada de Dynamo que se puede usar es la 1.3.4 de la versión 1 y de la versión 2 la 2.04
- Revit 2017: La versión más actualizada de Dynamo que se puede usar es la 1.3.4 de la versión 1 y de la versión 2 la 2.04
- Revit 2016: La versión más actualizada de Dynamo que se puede usar es la 1.3.2
- Para Revit 2015: La versión más actualizada de Dynamo que se puede usar es la 1.2
- Para Revit 2014: La versión más actualizada de Dynamo que se puede usar es la 8.2
- Daily: en este grupo están las versiones más actualizadas del programa. Ahora bien, estas versiones son inestables. Esto significa que las versiones diarias están en periodo de prueba y pueden dar más errores de lo normal.

Al estar en constante evolución las versiones sucesivas de Dynamo van mejorando constantemente los errores de la anterior. Asimismo los desarrolladores de complementos de Dynamo suelen programar sus complementos en la última versión de Dynamo disponible. Más adelante explicamos cómo afectan estos complementos de los desarrolladores a la programación visual que se hace con Dynamo.



**Releases**

Most stable

Last Modified	Size	Key
2017-04-06T08:52:27.000Z	186946872	DynamoInstall11.3.0.exe
2016-11-24T06:08:13.000Z	148033064	DynamoInstall11.2.1.exe
2016-10-03T01:47:44.000Z	148617768	DynamoInstall11.2.0.exe
2016-08-09T13:40:59.000Z	82980900	DynamoInstall11.1.0.exe
2016-04-29T03:18:16.000Z	82793685	DynamoInstall11.0.0.exe
2016-03-23T10:33:20.000Z	68623024	DynamoInstall10.9.2.exe
2016-02-18T14:49:17.000Z	68623328	DynamoInstall10.9.1.exe
2015-11-09T23:10:14.000Z	58397313	DynamoInstall10.9.0.exe
2015-08-31T01:41:50.000Z	54739526	DynamoInstall10.8.2.exe
2015-07-14T12:07:22.000Z	54449570	DynamoInstall10.8.1.exe
2015-04-09T13:55:29.000Z	44548708	DynamoInstall10.8.0.exe
2015-01-07T03:53:42.000Z	28012992	DynamoInstall10.7.5.exe
2014-11-24T04:20:43.000Z	28252477	DynamoInstall10.7.4.exe
2014-11-05T17:59:53.000Z	28457963	DynamoInstall10.7.3.exe
2014-09-25T00:32:50.000Z	28311312	DynamoInstall10.7.2.exe
2014-07-23T17:52:21.000Z	67125387	DynamoInstall10.7.1.exe
2014-04-25T20:22:04.000Z	83455104	DynamoInstall10.6.3.exe
2014-04-03T11:24:47.000Z	66177256	DynamoInstall10.7.0.exe
2013-11-26T00:32:40.000Z	58876666	DynamoInstall10.6.2.exe
2013-10-14T18:27:14.000Z	53799606	DynamoInstall10.6.1.exe

VERSIONES ESTABLES:  
RECOMENDADAS

**Daily**

Latest software builds working towards the next stable release

Last Modified	Size	Key
2017-03-01T08:37:55.000Z	135547304	DynamoRevit11.2.3.20170301T0837.exe
2017-03-01T04:42:11.000Z	135539456	DynamoRevit11.2.3.20170228T0442.exe
2017-02-27T12:51:15.000Z	184666809	DynamoRevit11.2.3.20170227T0451.exe
2017-02-27T08:34:33.000Z	184665208	DynamoRevit11.2.3.20170227T0834.exe
2017-02-27T04:33:29.000Z	184666808	DynamoRevit11.2.3.20170226T0433.exe
2017-02-25T16:49:34.000Z	184611529	DynamoRevit11.2.3.20170225T0449.exe
2017-02-24T12:34:19.000Z	184666808	DynamoRevit11.2.3.20170224T0434.exe
2017-02-23T12:50:04.000Z	184665776	DynamoRevit11.2.3.20170223T0450.exe
2017-02-23T04:33:30.000Z	184665624	DynamoRevit11.2.3.20170222T0433.exe
2017-02-20T16:37:10.000Z	185546504	DynamoRevit11.2.3.20170220T0437.exe
2017-02-16T20:54:42.000Z	185596952	DynamoRevit11.2.3.20170216T0437.exe
2017-02-16T12:39:14.000Z	185596936	DynamoRevit11.2.3.20170216T0430.exe
2017-02-15T12:37:35.000Z	185596184	DynamoRevit11.2.3.20170215T0437.exe
2017-02-14T04:29:05.000Z	185615528	DynamoRevit11.2.3.20170213T0429.exe

VERSIONES INESTABLES:  
NO RECOMENDADAS



## 2. OPERACIONES BÁSICAS

### 2.1. LA CONSOLA DE DYNAMO.

Se accede a Dynamo desde la pestaña “Manage”, desde la versión de Revit 2016 y posteriores, y desde la pestaña de “Add-ins” para las versiones 2015 y anteriores.

Una vez se accede al programa y elegimos la opción “File”, “New”, “Home Workspace” (podemos acceder a la misma opción también desde CTRL+N) veremos la consola que aparece más abajo.



La consola de Dynamo se organiza de la siguiente manera:

- Zona de trabajo: Sobre la zona de trabajo el usuario generará la programación visual añadiendo nodos que se conectarán entre sí. Desde la zona de trabajo podemos tomar información de Revit e interactuar con ella para después volverla a volcar, una vez manipulada, en el modelo de Revit.
- Paso de visualización de nodos a geometría: Los botones de arriba a la derecha definen si el usuario desea visualizar nodos o geometría. Normalmente utilizamos este paso entre modos cuando queremos previsualizar los objetos que estamos modificando o visualizando mediante los nodos en forma de geometría. El modo geometría no deja de ser una previsualización de las operaciones ejecutadas con la geometría desde Dynamo.
- Paquetes de Nodos: Los paquetes de nodos equivaldrían a la barra de herramientas de Dynamo. Son las acciones que tenemos disponibles para ejecutar en el programa.
- Paso de modo automático a manual: Esta función define si el programa estará, o no, operando constantemente. Si Dynamo está en modo “Automático” todas las acciones que realicemos se ejecutarán de inmediato. Sin embargo, si el modo está en manual, entonces tendremos que pulsar la opción “ejecutar” para ver los resultados de la programación visual.

¿Cuándo debemos trabajar en manual y cuándo en automático? Debemos trabajar en manual siempre y cuando estemos trabajando con grandes cantidades de información paramétrica o geometría que, de otra forma, podrían sobrecargar el Dynamo o el modelo causando que ambos se cerraran.



Solo es aconsejable trabajar en modo automático cuando se trabaja con muy poca información paramétrica o de geometría, por ejemplo, cuando estamos haciendo pequeñas pruebas con la programación.

## **2.2. DESCARGAR PAQUETES DE INTERNET**

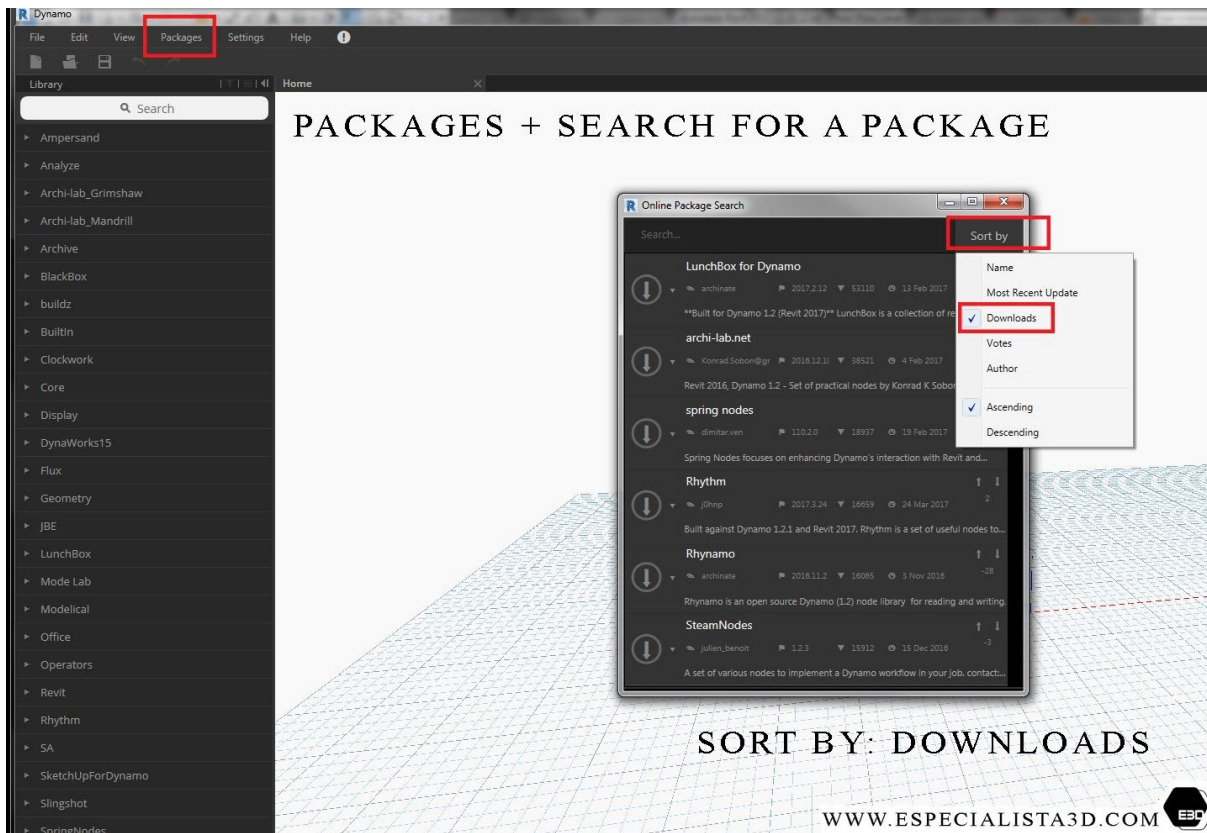
Dynamo trae incorporados unos nodos por defecto. Estos nodos aparecen en la zona de Packages/Nodos al iniciar el programa.

Sin embargo, Dynamo es un programa de programación visual de código abierto. Que sea de código abierto significa que cualquier usuario puede programar sus propias herramientas y ofrecerlas para que estén disponibles para todos los usuarios.

Por ejemplo, imaginemos que el programa original de Dynamo no me permitía seleccionar puertas. Un usuario puede detectar este problema y generar su propia programación visual para conseguir realizar esta acción que, sin la aportación del usuario, quedaría sin resolver. Una vez resuelve esta particularidad el usuario puede decidir compartir su solución, en forma de package, con el resto de usuarios de Dynamo. Para ello subiría la nueva resolución a los packages de internet, junto con una descripción de qué aporta el nuevo package.

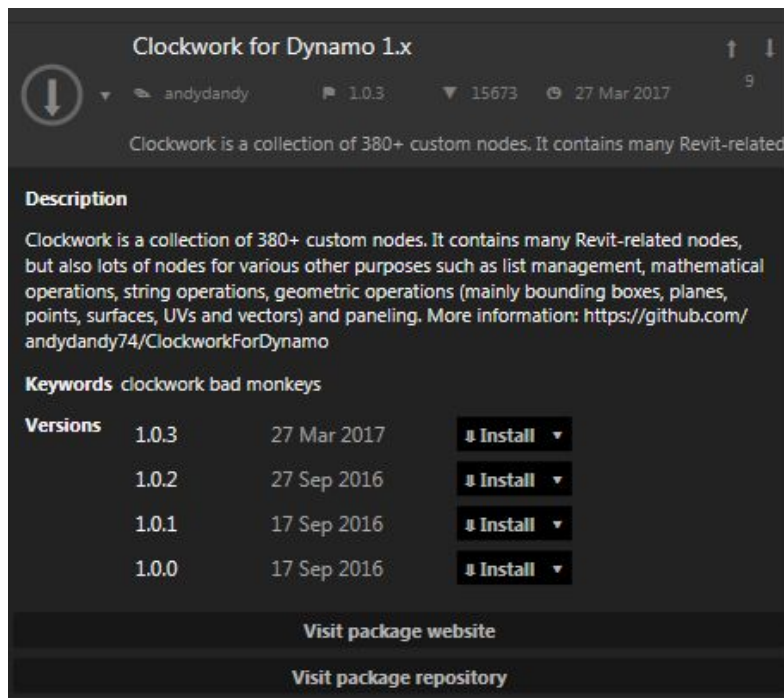
A día de hoy hay 1042 packages y 331 autores registrados que han subido sus resoluciones y las han incorporado a los packages de descarga de Dynamo.

Los packages se pueden descargar desde packages + search for a package. Cuando buscamos aparece un menú con todos los packages disponibles. Podemos organizar estos packages por número de descargas desde el panel de descargas de “packages”. De este modo podremos seleccionar aquellos packages más descargados por los usuarios. Los seleccionamos de esta manera desde el menú “Sort By” seleccionando la opción “Downloads”.



Los packages más descargados por los usuarios suelen ser los más eficaces. Ahora bien, debemos de tener cuidado con que la versión del package que descargamos se corresponda con la versión de Dynamo que tenemos instalada en ese momento. Cuando hacemos doble click sobre el nombre de cualquier Package nos aparece un desplegable con aquellas versiones para la que está disponible.





Por ejemplo, en el caso de la imagen superior. Si tuviéramos instalada la versión de Dynamo 1.0.0 (del 29 de Abril de 2016, ver imagen de abajo) tendríamos que irnos a una versión anterior a las disponibles en el acceso desde el panel de packages.

## Releases

Most stable

Last Modified	Size	Key
2017-04-06T08:52:27.000Z	186946872	<a href="#">DynamoInstall1.3.0.exe</a>
2016-11-24T06:08:13.000Z	140033064	<a href="#">DynamoInstall1.2.1.exe</a>
2016-10-03T01:47:44.000Z	140617768	<a href="#">DynamoInstall1.2.0.exe</a>
2016-08-01T13:40:58.000Z	82988900	<a href="#">DynamoInstall1.1.0.exe</a>
2016-04-29T03:18:16.000Z	82793685	<a href="#">DynamoInstall1.0.0.exe</a>
2016-03-23T10:33:20.000Z	68623024	<a href="#">DynamoInstall0.9.2.exe</a>
2016-02-18T14:49:17.000Z	68623328	<a href="#">DynamoInstall0.9.1.exe</a>
2015-11-09T23:10:14.000Z	58397313	<a href="#">DynamoInstall0.9.0.exe</a>
2015-08-31T01:41:50.000Z	54739526	<a href="#">DynamoInstall0.8.2.exe</a>
2015-07-14T12:07:22.000Z	54449570	<a href="#">DynamoInstall0.8.1.exe</a>
2015-04-09T13:55:29.000Z	44548708	<a href="#">DynamoInstall0.8.0.exe</a>
2015-01-07T03:53:42.000Z	28012992	<a href="#">DynamoInstall0.7.5.exe</a>
2014-11-24T04:20:43.000Z	28252477	<a href="#">DynamoInstall0.7.4.exe</a>
2014-11-05T17:59:53.000Z	28457963	<a href="#">DynamoInstall0.7.3.exe</a>
2014-09-25T00:32:50.000Z	28311312	<a href="#">DynamoInstall0.7.2.exe</a>
2014-07-23T17:52:21.000Z	67125387	<a href="#">DynamoInstall0.7.1.exe</a>
2014-04-25T20:22:04.000Z	83459184	<a href="#">DynamoInstall0.6.3.exe</a>
2014-04-03T11:24:47.000Z	66177256	<a href="#">DynamoInstall0.7.0.exe</a>
2013-11-26T00:32:40.000Z	58876666	<a href="#">DynamoInstall0.6.2.exe</a>
2013-10-14T18:27:14.000Z	53799606	<a href="#">DynamoInstall0.6.1.exe</a>

Para buscar la versión de esa fecha tendríamos que acceder a “visit package repository”. Dentro del repositorio hay una descripción de las distintas versiones de cada package y una



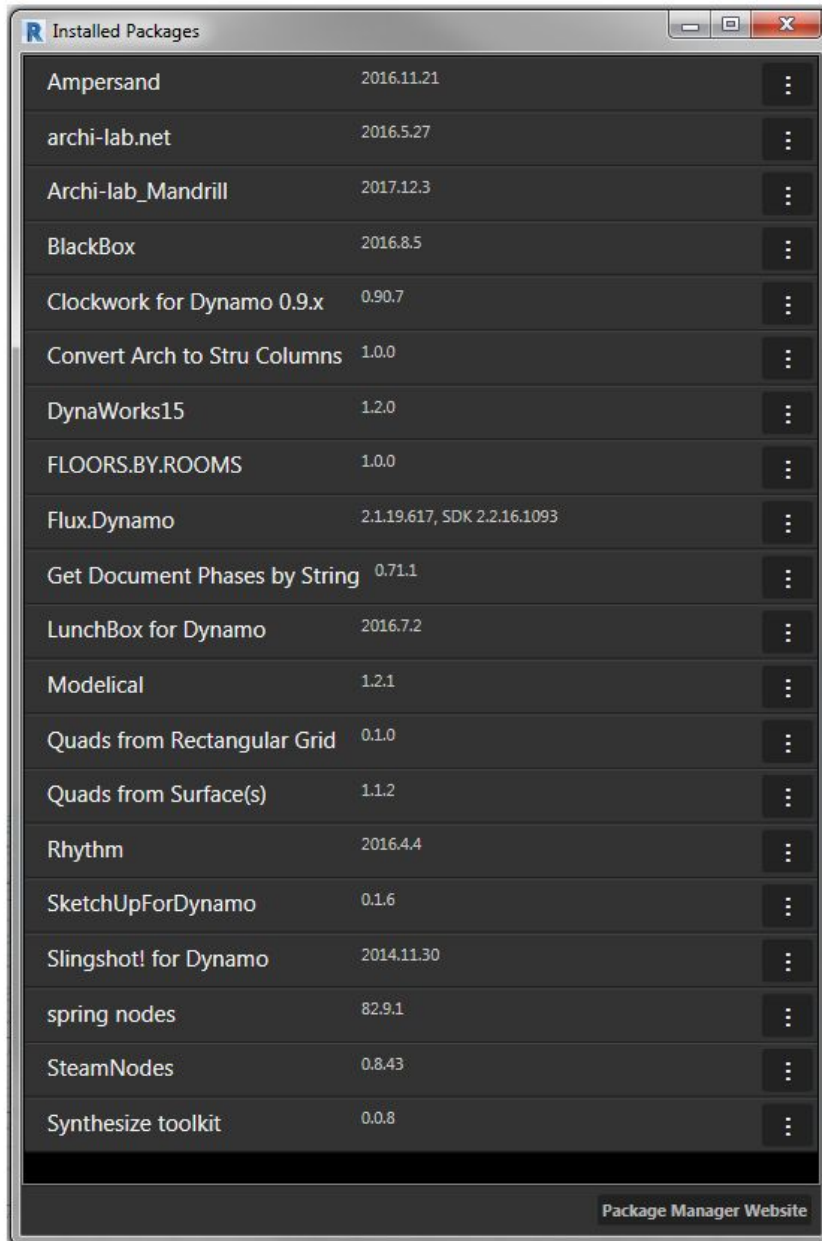
indicación sobre las versiones que ya han sido desactualizadas, con la posibilidad de descargarlas.

Como criterio general lo ideal es trabajar con la última versión de Revit y Dynamo disponibles así como las últimas versiones de los packages, que suelen estar actualizadas para la última versión por los desarrolladores.

### **2.2.1. PAQUETES RECOMENDADOS**

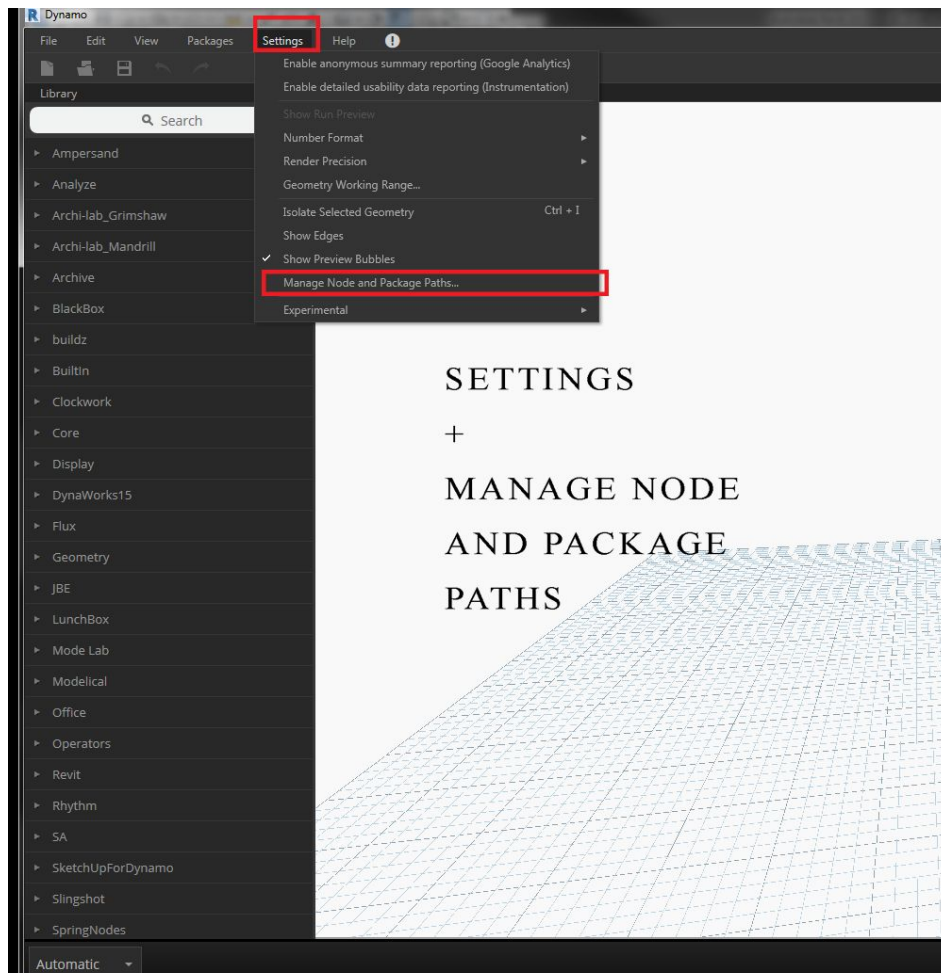
Existe una gran cantidad de Packages que se pueden descargar. Al estar Dynamo en constante desarrollo estos packages van creciendo y actualizándose. Algunos packages pueden dar problemas de compatibilidad con otros al instalarlos.

En la imagen de abajo hay una lista de los packages que utilizo a diario como programadora visual. Estos son los packages que comparto hoy día con los alumnos a los que doy clase de Dynamo. Están testados y funcionan bien estando todos ellos instalados a la vez, sin problemas de compatibilidades entre sí.





## 2.2.2. DÓNDE SE GUARDAN LOS PAQUETES



Todos los packages descargados siguiendo las indicaciones anteriores se descargarán en la carpeta que indique la ruta que tenga guardada Dynamo. Esta ruta puede verse desde settings, manage node and package paths.

## 2.3. QUÉ ES UN NODO

Un nodo es un elemento de Dynamo que ejecuta una acción sobre los elementos de Revit que se designen.

Como hemos dicho Dynamo es un complemento de programación visual que ejecuta acciones sobre elementos, ya sean físicos o en forma de información, los transforma y los puede devolver a Revit una vez manipulados.



Pongamos un ejemplo más sencillo. Si Revit fuera una cocina el sistema que trae incorporado solo nos aportaría platos concretos. En el otro extremo estaría la programación por scripting. Este tipo de programación nos permitiría poder interactuar con cada ingrediente hasta los niveles más concretos. Dynamo sería un punto intermedio, con los nodos de Dynamo podríamos seleccionar cada grupo de ingredientes, pero no ingredientes por separado.

Digamos que un nodo de Dynamo sería como una batidora en la similitud del restaurante. Una batidora coge un grupo de ingredientes y los modifica en masa.

Cada nodo de Dynamo sirve para ejecutar una acción concreta. A continuación veremos nodos concretos para vincular el modelo de Revit con Dynamo.

## 2.4. VINCULAR EL MODELO DE REVIT CON DYNAMO.

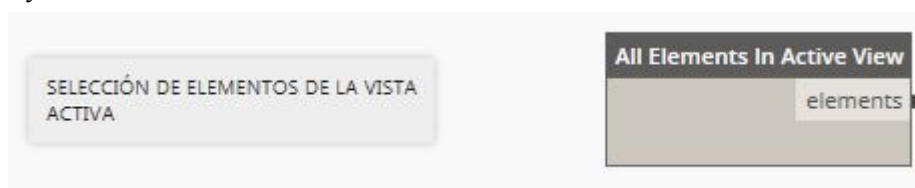
### 2.4.1. Select model elements.

Selecciona elementos desde Revit en la vista activa para su posterior manipulación en Dynamo.



### 2.4.2. All elements in active view.

Selecciona todos los elementos desde Revit en la vista activa para su posterior manipulación en Dynamo.





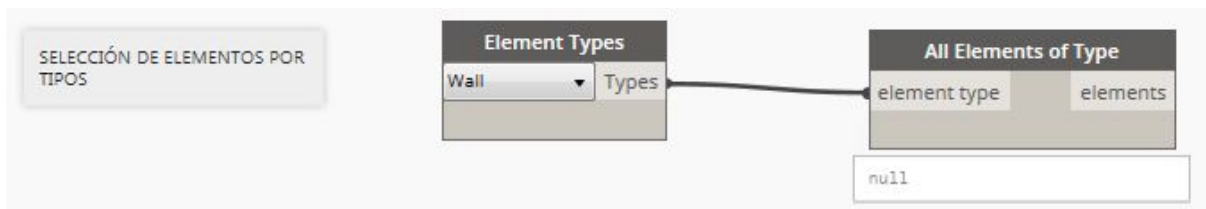
### 2.4.3. Categories. All elements of category.

El desplegable “Categories” permite seleccionar una categoría de Revit, “All elements of category” permite seleccionar todos los elementos de una categoría, independientemente de en qué vista o lugar del modelo se encuentren.



### 2.4.4. Element Types. All elements of type.

El desplegable “Element Types” permite seleccionar una tipología de elementos de Revit, “All elements of type” permite seleccionar todos los elementos de ese tipo, independientemente de en qué vista o lugar del modelo se encuentren.



### 2.4.5. Levels. All elements at level.

El desplegable “Levels” permite seleccionar un nivel de todos los que estén incorporados en el modelo de Revit, “All elements at” permite seleccionar todos los elementos que estén referenciados en este nivel.



### 2.4.6. Get All Line Styles

Existen nodos concretos de Packages, como este de Archilab Grimshaw, que permite seleccionar todos los estilos de línea de un modelo.



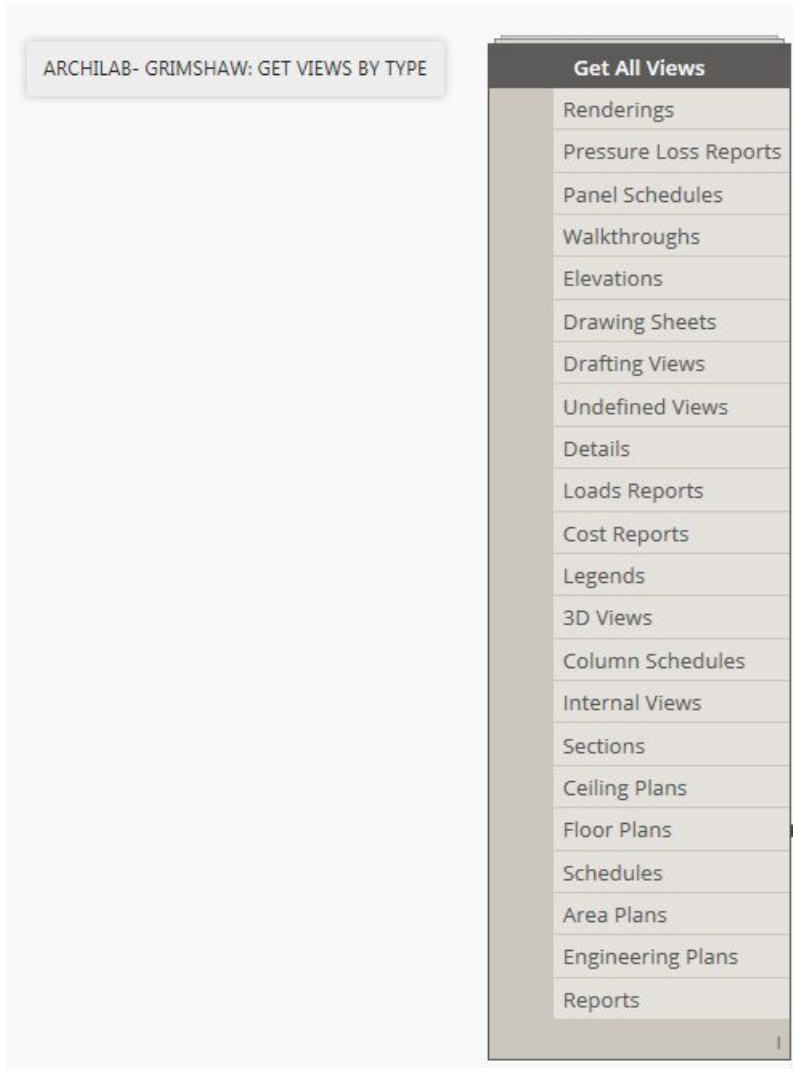
#### 2.4.6. Get Scope Boxes

Existen nodos concretos de Packages, como este de Modelical, que permite seleccionar todas las scope boxes de un modelo.

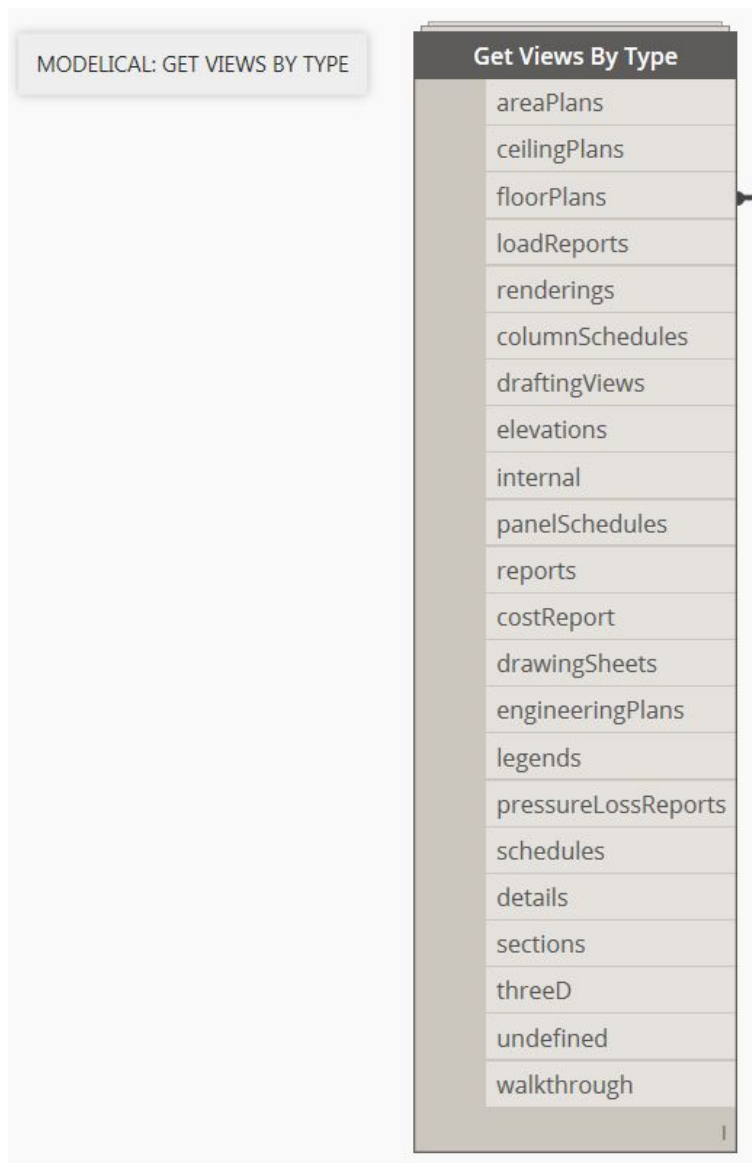


#### 2.4.7. Get All Views

Los packages de Modelical y Archilab-Grimshaw resuelven la misma necesidad, seleccionar vistas de un modelo en función del tipo de vista.





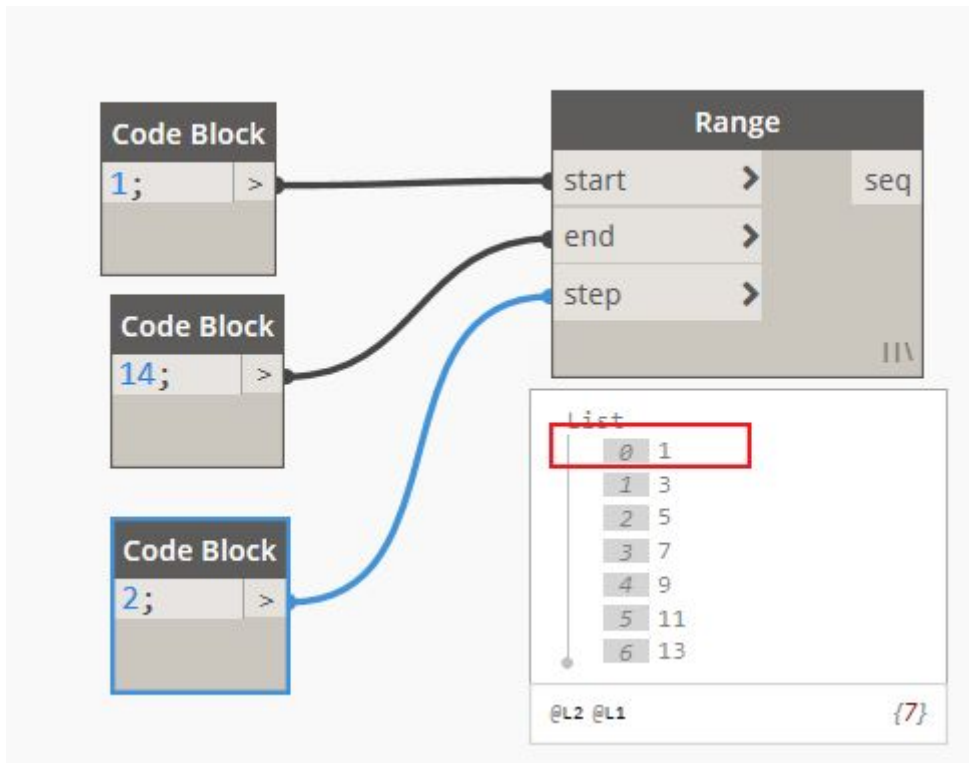


Ahora que hemos visto cómo seleccionar elementos de un modelo vamos a ver cómo realizar operaciones básicas con estos elementos.

## 2.5. OPERACIONES BÁSICAS CON DYNAMO.

Para realizar operaciones con elementos lo primero que debemos saber es cómo actúan los nodos de Dynamo. Los nodos de Dynamo gestionan operaciones con listas. Una lista es la forma en la que se organiza la información. Igual que en nuestros sistemas operativos tenemos archivos y carpetas, en Dynamo tenemos elementos y listas. Una lista es un conjunto de elementos. Las listas se ordenan en índices, estos índices empiezan a nombrarse desde el índice 0.

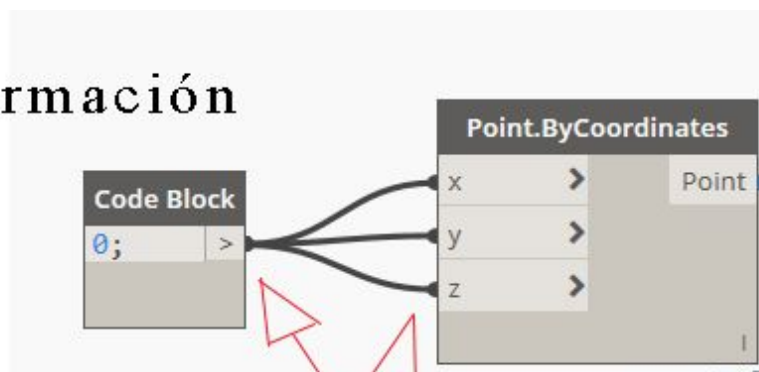
Por ejemplo, más abajo vemos una lista de números que van del uno hasta el catorce saltando de dos en dos. Podemos ver cómo en esta lista el primer elemento, 1, tiene el índice 0.



Los nodos de Dynamo gestionan estos elementos y listas y consisten fundamentalmente en tres partes:

- Inputs: entradas de información, por cada entrada de información se admite un único cable de información, es decir una única lista de información que entra en el nodo.
- Nodo: El nodo realiza la acción que indica su nombre, por ejemplo, el nodo de abajo genera un punto de coordenadas (0,0,0)
- Ouputs: salidas de información, podemos sacar tantos cables de salida de información de un nodo como necesitemos.

## información

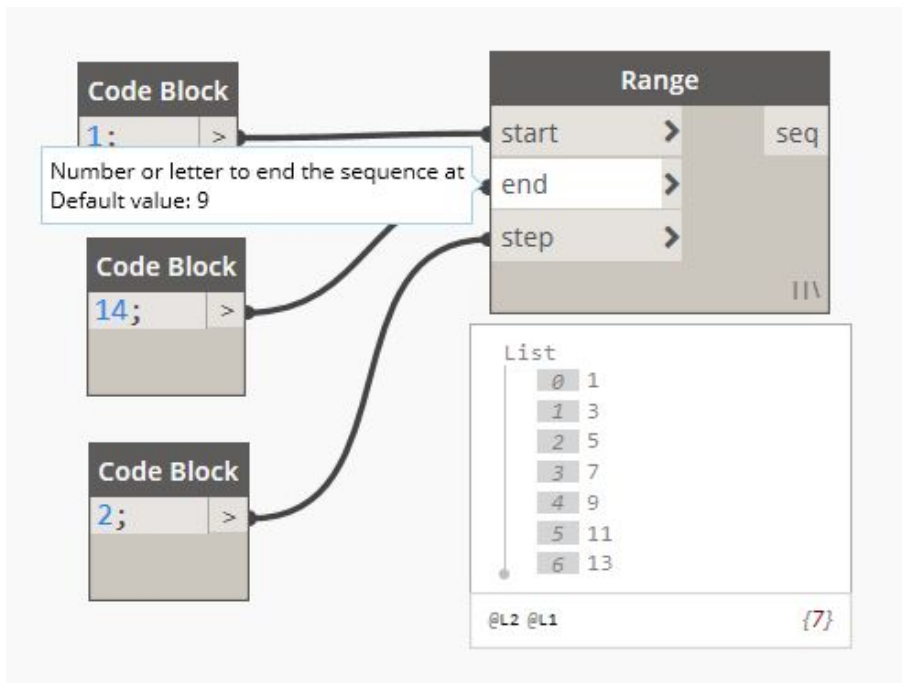


salida de información

entrada de información



Si ponemos el puntero del ratón sobre el input entonces nos aparece un mensaje con información. Es conveniente prestar atención a esta información, ya que nos indica qué tipo de variable espera el nodo que introduzcamos como información en el nodo. Ej. En un nodo de tipo “Number by range” si pasamos el ratón sobre el input vemos que espera “Number o letter” es decir, un número o una letra.

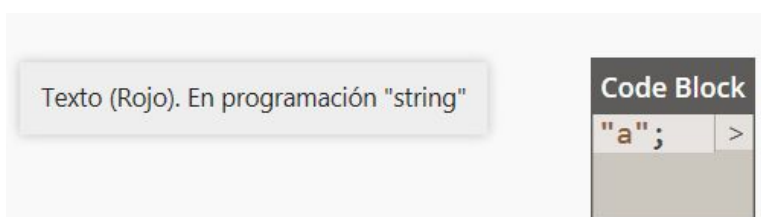


### 2.5.1. Codeblocks

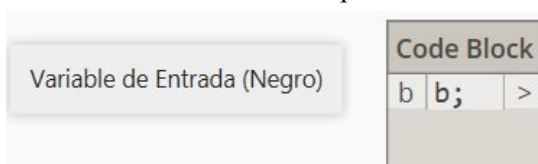
Uno de los tipos de nodo más útiles de Dynamo es el llamado Code Block. El Code Block es la forma de vincular la programación visual con una programación basada en texto. Podemos obtener el nodo de Code Block buscando directamente “Code Block” en el menú de nodos o haciendo doble click en el área de trabajo.

Para definir variables desde los Code Blocks debemos escribirlas del siguiente modo:

- Textos, en programación denominados “string”, entre comillas.



- Variables: como una palabra o letra empezando por minúscula.

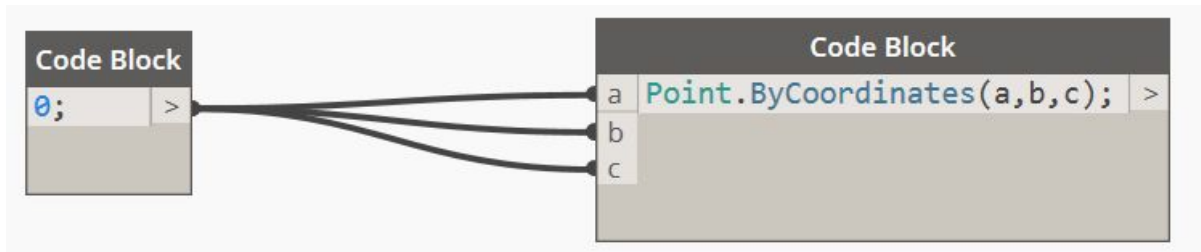




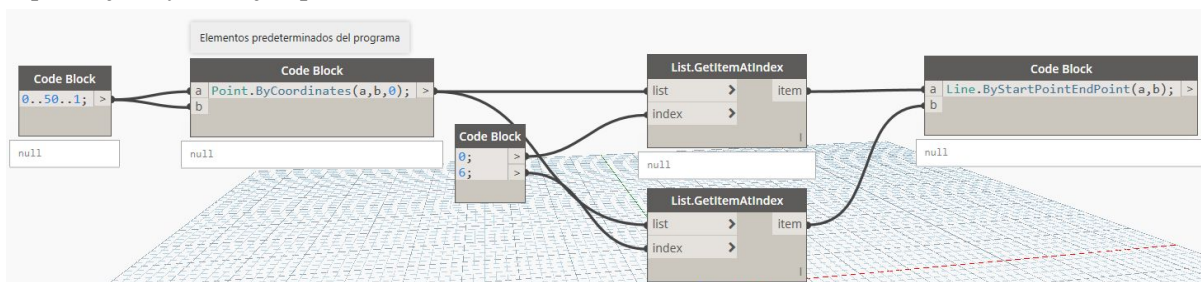
- Números: Bastará con escribir la cifra, aparecen en azul.



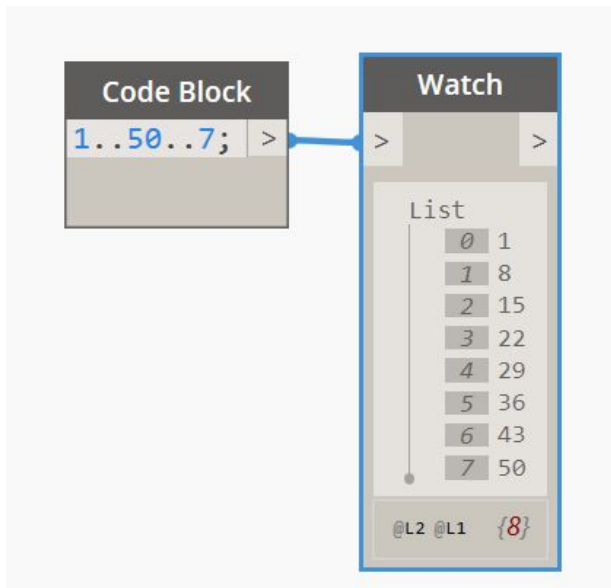
- Entidades: para escribir entidades concretas podemos empezar a escribir con mayúscula y aparecerá un desplegable con opciones. Por ejemplo, el caso de un punto.



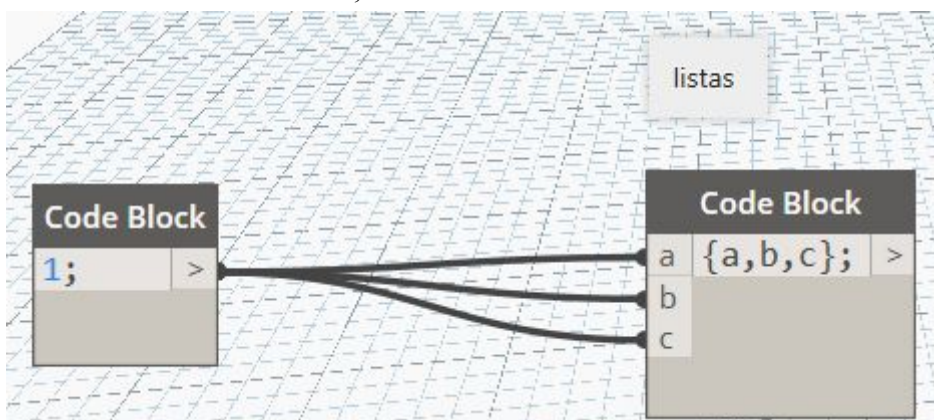
Aquí abajo hay otro ejemplo con el caso de una línea.



- Otras funciones: por ejemplo más abajo se ve cómo se escribiría en Code Block una lista de 1 al 50 con 7 divisiones.



- Listas: para definir listas separamos los elementos con comas y abrimos las listas con un “{“ y las cerramos con un “}”



### 2.5.2. Números

Podemos definir números en Dynamo de varias formas. La primera, como hemos visto antes, es añadirlos en un Code Block.

En Dynamo los números se dividen en dos tipologías: “Number” en general, para todo tipo de números: negativos, con decimales... e “Integer” para los números enteros.

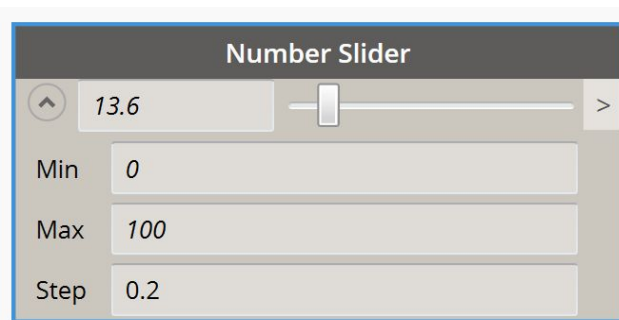
Podemos seleccionar estas tipologías de números con un nodo llamado “Slider”





El slider es especialmente útil para introducir un valor que luego vayamos a querer modificar posteriormente.

Si desplegamos la flecha del slider podremos ver que la forma en que funciona es estableciendo un valor mínimo, un máximo y un salto entre valores. Por ejemplo, en la imagen de abajo el salto es de 0,2.



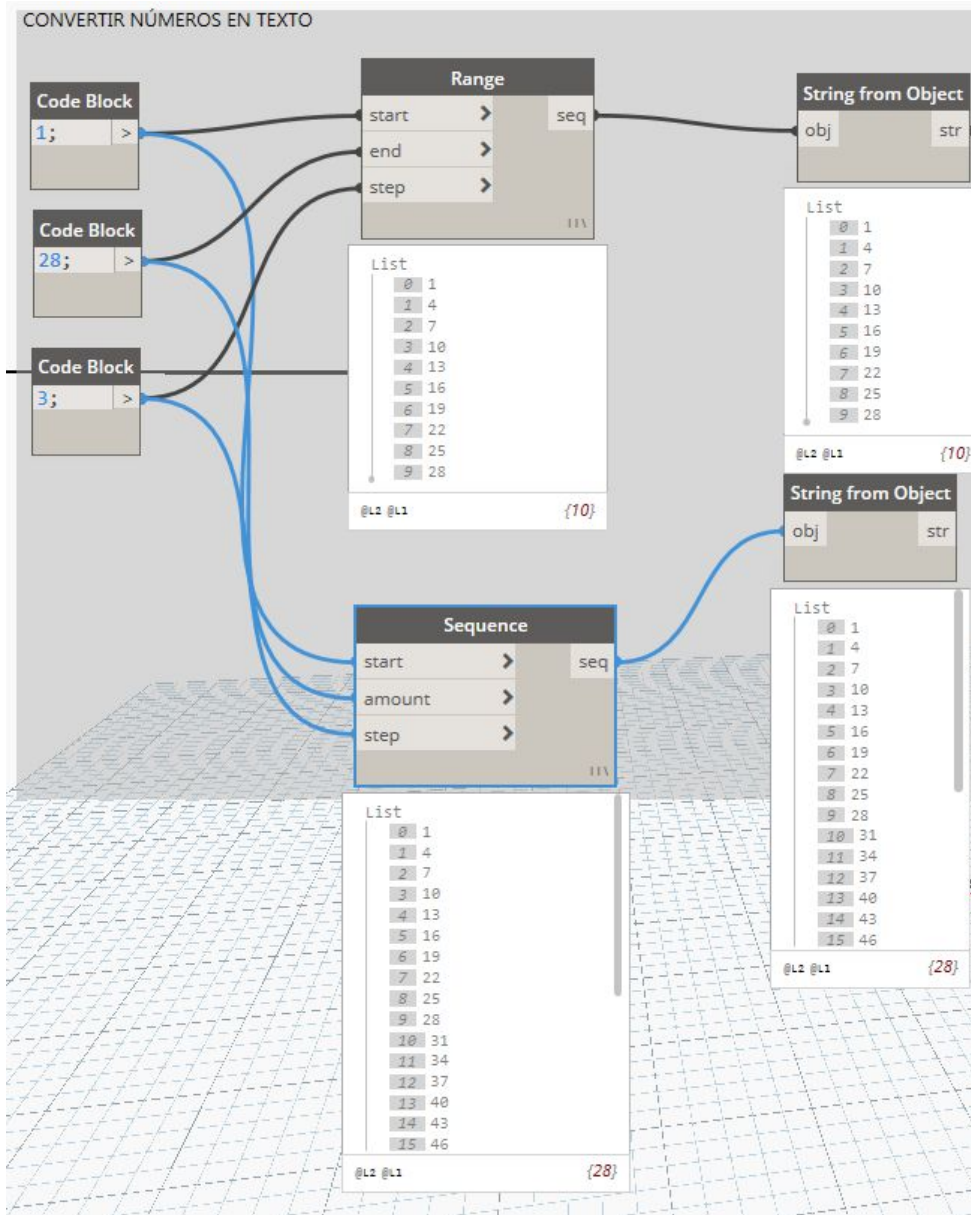
Sin embargo si el slider es de número entero no podemos establecer el step con decimales, lógicamente.

### 2.5.3. Conversión de elementos para trabajar

Cuando estamos operando en Dynamo a veces encontraremos que necesitamos transformar las variables para poder trabajar con ellas. Imaginemos que queremos introducir información en un parámetro que es de tipo “number” pero la información original está en forma de texto. Tendremos pues que transformar el texto en número.

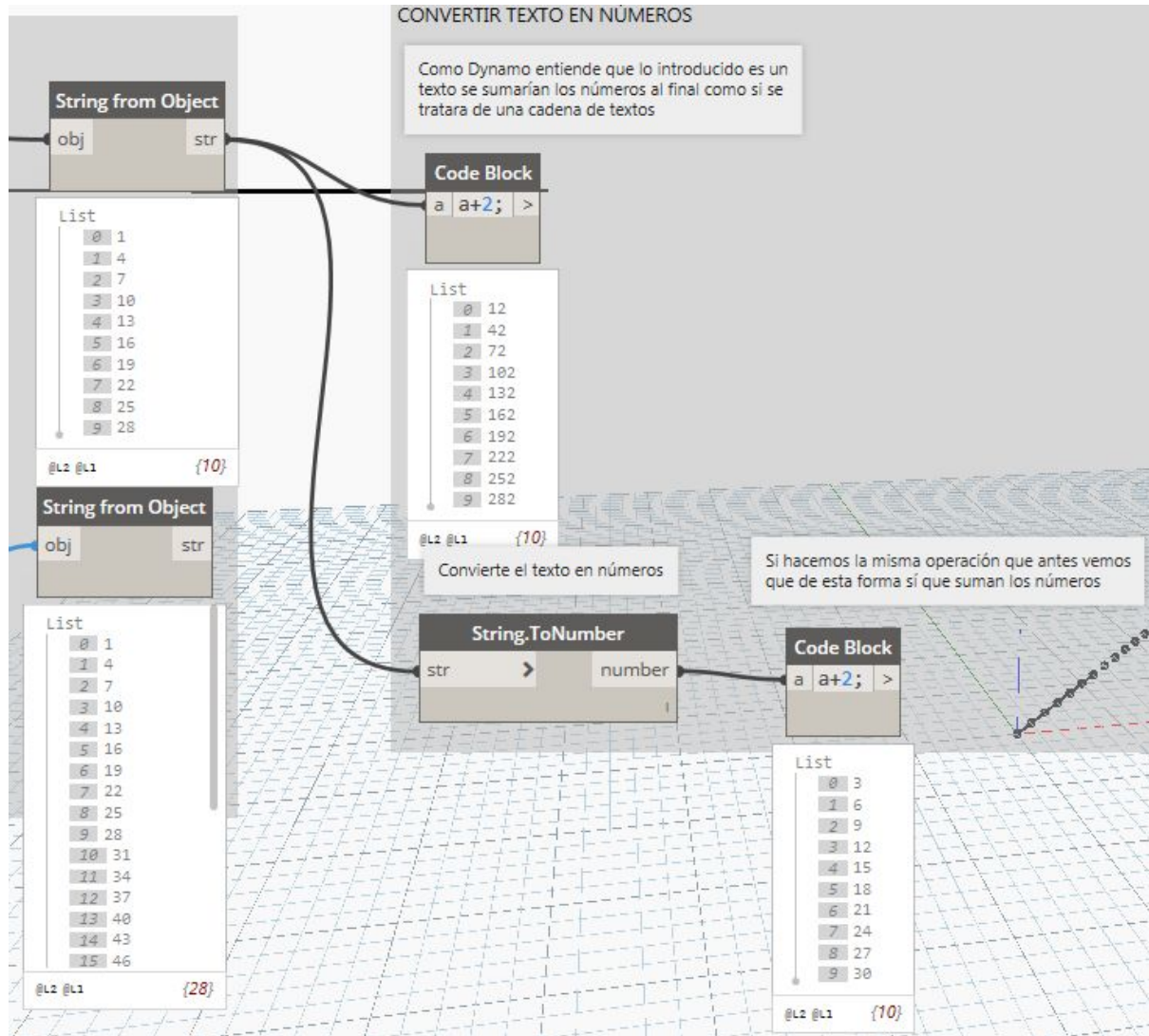
Vamos a ver algunos ejemplos de los nodos de conversión más habituales.

#### 2.5.3.1. String from Object. Conversión de cualquier objeto en texto.





### 2.5.3.2. String.ToNumber: Conversión del texto en números



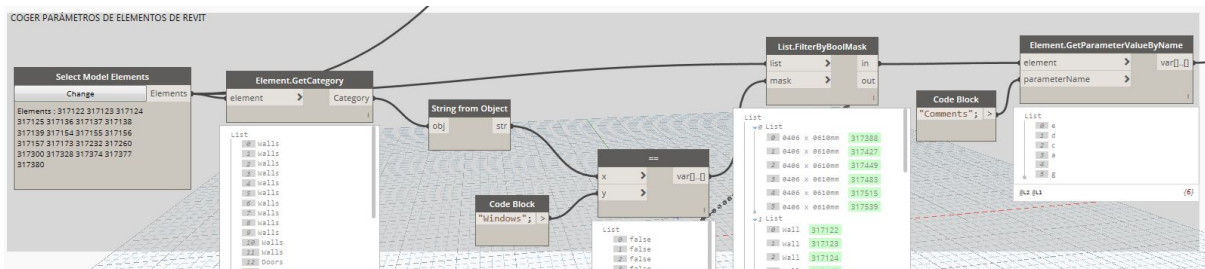
## 2.6. INFORMACIÓN EN DYNAMO DESDE REVIT & HACIA REVIT

Una de las operaciones principales que podemos realizar con Dynamo es gestionar grandes cantidades de información. Revit es un software BIM. La propia definición de BIM incluye la palabra “información”. Esta información suele venir en forma de parámetros que se incorporan a los distintos elementos de Revit.

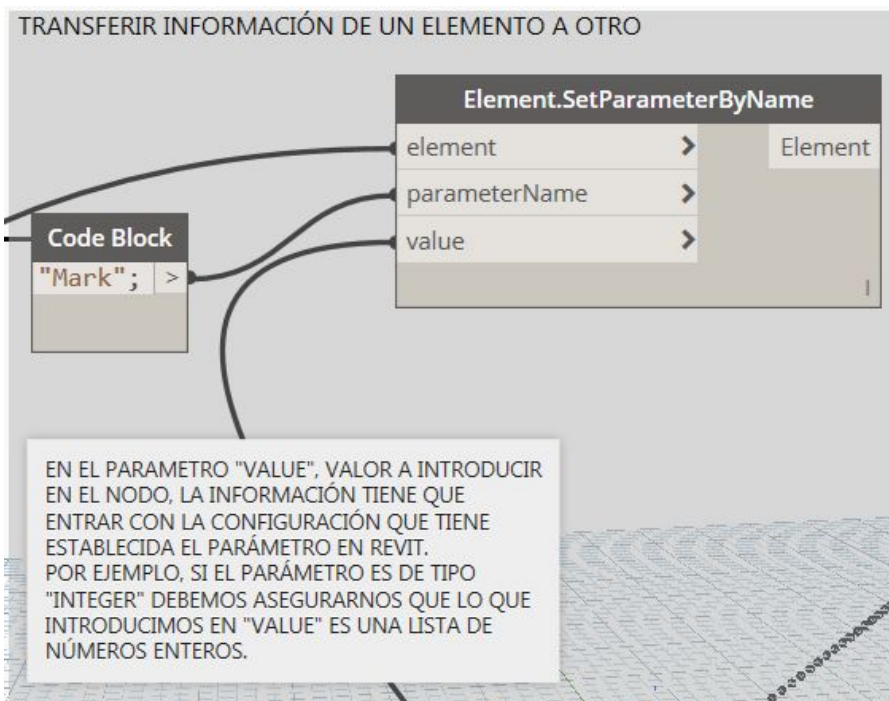
Por ello uno de los nodos más usados de Dynamo es “Get Parameter Value By Name”. ¿Qué es lo que hace este nodo? Permite coger el valor de un elemento o una lista de elementos. Normalmente partimos de una base de valores que modificamos en Dynamo.

Abajo podéis ver un ejemplo concreto donde este nodo está integrado dentro de una definición más amplia.





Otro de los nodos principales es `Element.SetParameterByName`, éste permite transferir los valores de la información a los elementos teniendo en cuenta el nombre del parámetro en el que introducir la información.



## 2.7. INFORMACIÓN REVIT-DYNAMO-EXCEL, EXCEL-REVIT-DYNAMO

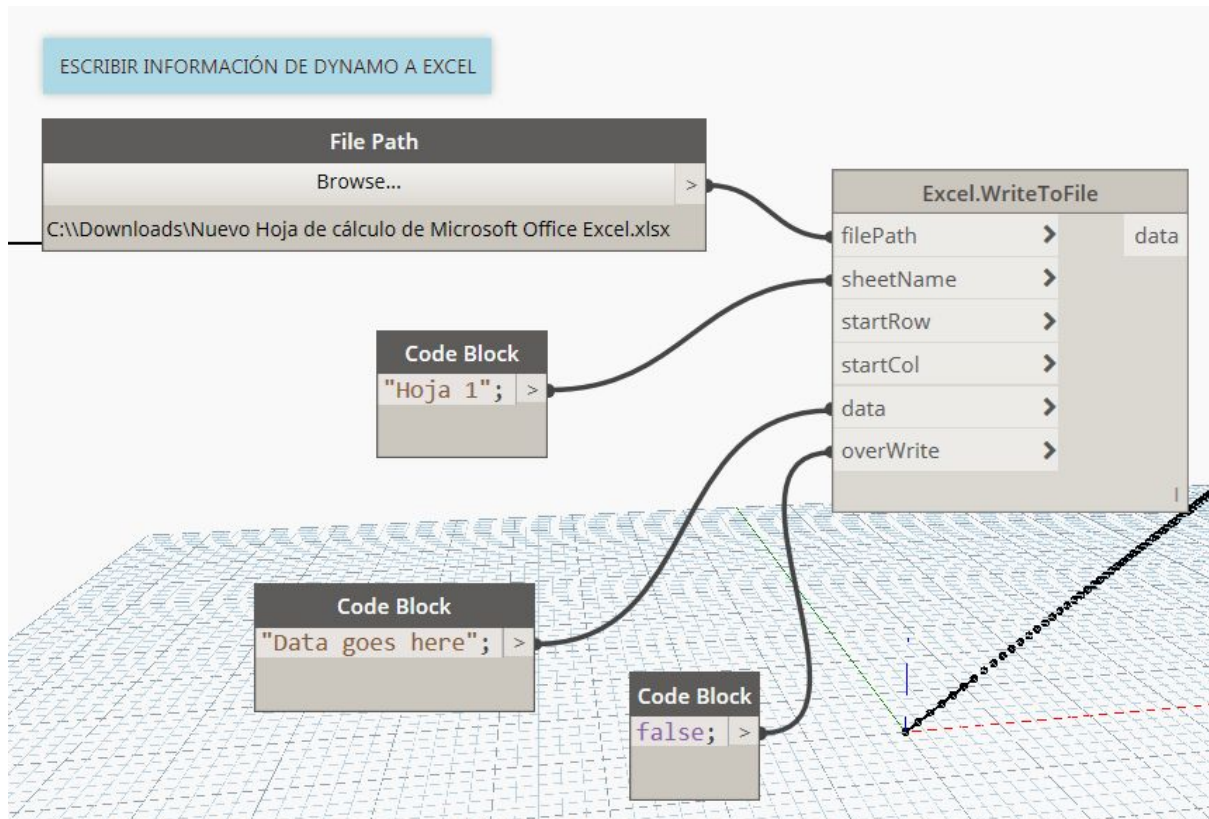
Una de las opciones más interesantes que incorpora Dynamo a Revit es la de traer información desde distintos programas con los que Revit no está vinculado por defecto. Uno de los programas que se suele utilizar más en las empresas para la gestión de información es Excel, de Microsoft.

Dynamo incorpora varios nodos para poder leer desde Excel y mandar la información desde Revit a Excel. Este procedimiento puede ahorrar muchísimas horas de trabajo puesto que Excel incorpora operaciones de las que no dispone Dynamo y viceversa. Se utiliza mucho puesto que, por ahora, es poco común contar con profesionales que manejen Dynamo de manera óptima y, sin embargo, el uso del Excel está muy extendido.



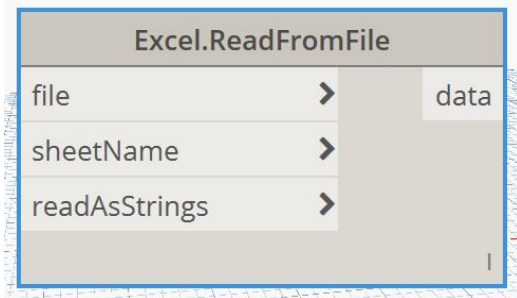
### 2.7.1. Excel.WriteToFile de Dynamo a Excel

Este nodo permite transferir información de Dynamo a Excel introduciendo la ruta del archivo Excel “File Path” y el nombre de la hoja de Excel a la que queremos transferir la información “Sheet Name”.



### 2.7.2. Excel.ReadFromFile

Este nodo permite transferir información desde Excel a Dynamo introduciendo el archivo Excel del que parte la información, el nombre de la hoja de la que queremos coger la información y si queremos o no leer toda la información como texto.





### 3. EJEMPLO: TOMAR INFORMACIÓN DE LAS PUERTAS DE UN MODELO Y LLEVARLA A LAS HABITACIONES.

Para concluir esta guía vamos a poner un ejemplo muy sencillo y verídico. Numerosos ejemplos de ejercicios de Dynamo pueden encontrarse internet, incluso en la propia [web oficial de Dynamo](http://www.especialista3d.com).

Partamos de un ejemplo en el que tenemos habitaciones cerradas por muros y puertas vinculadas a estos muros. Imaginemos que queremos trasladar información de las habitaciones a las puertas o al revés.

Lo primero que tendríamos que hacer es examinar los elementos para saber de qué parámetros disponemos, tanto en las habitaciones como en las puertas.

The image displays two screenshots of the Revit Properties panel and a 3D model of a building. The 3D model shows four rooms: EDIFICIO 1 (233 m²), EDIFICIO 2 (62 m²), EDIFICIO 3 (156 m²), and EDIFICIO 4 (25 m²). Doors are labeled as PUERTA PEQUEÑA and PUERTA GRANDE.

**Properties Panel - Room (R)**

Category	Property	Value
Rooms (1)	Level	Level 1
Constraints	Upper Limit	Level 1
	Limit Offset	4000.0
	Base Offset	0.0
Identify Data	Number	1
	Name	ASEO EDIFICIO 1
Dimensions	Area	32.339 m²
	Perimeter	22800.0
	Unbounded Height	4000.0
	Volume	Not Computed
	Computation Height	0.0
Identity Data	Number	1
	Name	ASEO EDIFICIO 1
	Image	
	Comments	
	Occupancy	
	Department	
	Base Finish	
	Ceiling Finish	
	Wall Finish	
	Floor Finish	
	Occupant	
Phasing	Phase	New Construction

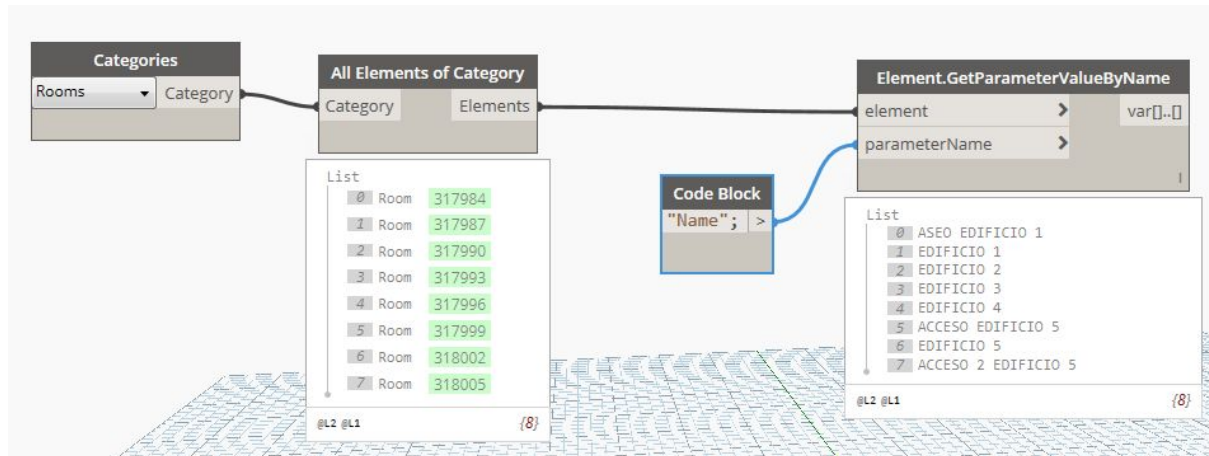
**Properties Panel - Door (M\_Single-Flush)**

Category	Property	Value
Doors (1)	Level	Level 1
Constraints	Sill Height	0.0
Construction	Frame Type	
Identify Data	Number	1
	Name	PUERTA PEQUEÑA
	Comments	m² edificio
Phasing	Phase Created	New Construction
	Phase Demolished	None
Other	Head Height	2134.0

Para obtener, por ejemplo, el nombre de las habitaciones bastaría con utilizar varios nodos de los que hemos hablado en esta guía. Por ejemplo, para seleccionar todas las habitaciones de un modelo podemos usar los nodos “Categories”, seleccionando las habitaciones y después “All elements of category” para seleccionar todos los elementos que tengan esta categoría.



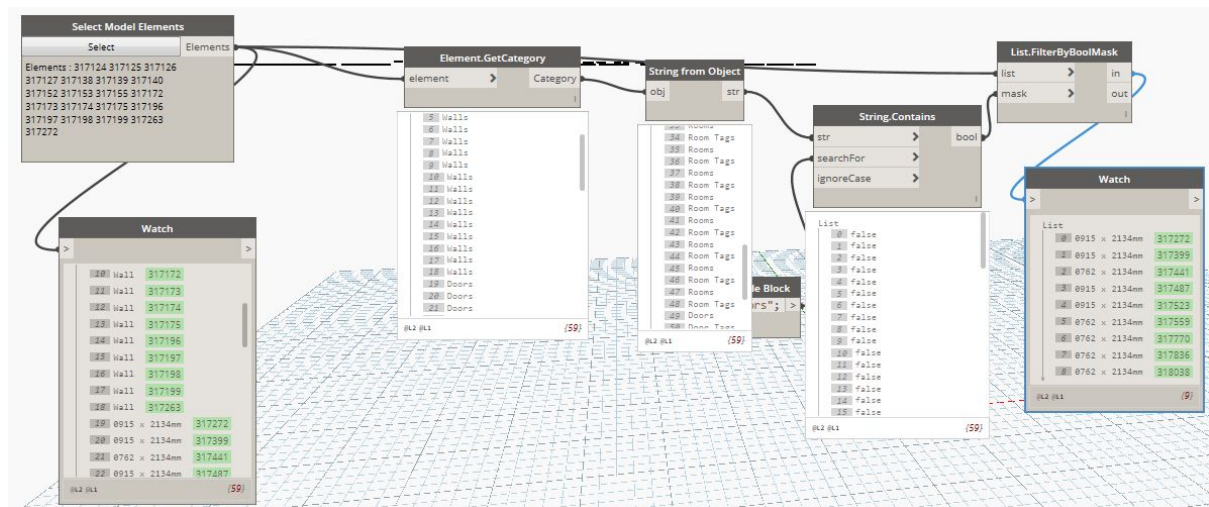
Después utilizaríamos Element.GetParameterValueByName e introduciríamos el valor del nombre del parámetro “Name” para obtener estos valores.



Vamos ahora a poner un ejemplo de cómo se seleccionarían las puertas del modelo. Lo podríamos hacer exactamente como en el ejemplo anterior pero en este caso vamos a variar el método.

Seleccionamos muchos objetos del modelo con el nodo “Select Model Elements”. Después tendremos que filtrar estos elementos para quedarnos solo con las puertas.

Para saber cuáles de estos elementos son puertas utilizaríamos el nodo Element.GetCategory, este nodo da la categoría dando un elemento. Como la categoría viene en forma de objeto utilizamos el nodo String from Object para transformar las categorías en textos.



Una vez convertidas las categorías en textos podemos filtrar estos textos con el nodo “String.Contains”. En este nodo introducimos por un lado las categorías en forma de texto por otro, en el input “searchFor” (buscar) introducimos el texto “Doors” como un Code Block. El nodo “String.Contains” da como resultado una lista de “true” y “false”, es decir, de verdaderos y falsos. ¿Qué quiere decir esto? Al utilizar el nodo “String.Contains” es como si le hiciéramos a cada elemento de la lista de textos la pregunta “¿contiene este texto la palabra “Doors”?” y el nodo nos da como resultado sí (en forma de verdaderos, es decir, true) o no (en forma de falsos, es decir, false).



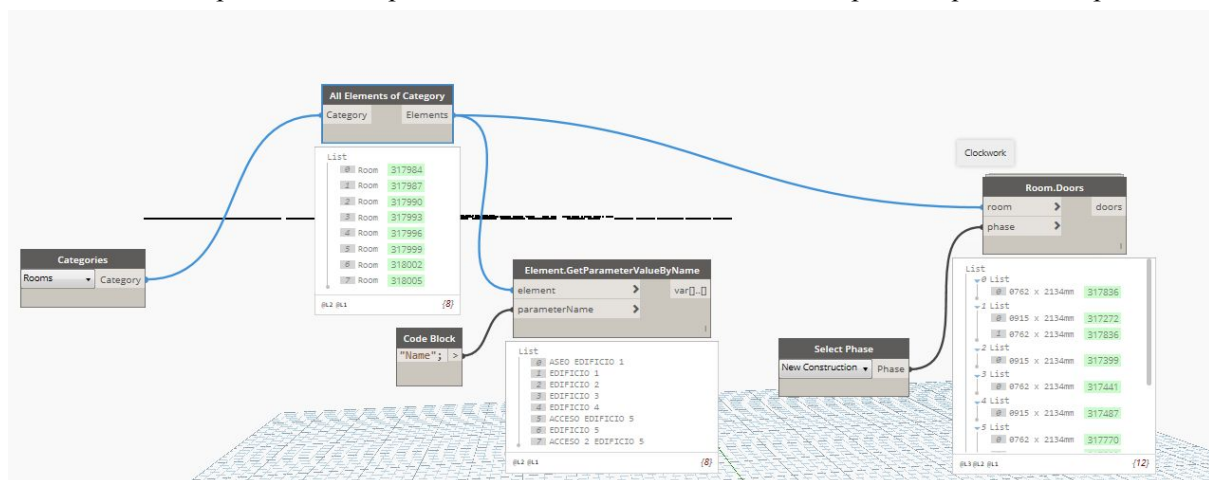
Con esta lista de verdaderos y falsos necesitaremos el nodo List.FilterByBoolMask. Este nodo es uno de los más utilizados para filtrar elementos. Para que funcione debemos introducir por un lado los elementos en el input “List”, en este caso los elementos son los seleccionados inicialmente, y por otro lado la máscara “mask” que tiene que ser una lista de verdaderos y falsos, como por ejemplo la que hemos obtenido desde el nodo “String Contains”.

Este nodo discrimina los elementos en el orden en el que entran en el el input “list” y los divide en dos en función de la lista de verdaderos y falsos que introduzcamos. Como en este caso los verdaderos son aquellos elementos cuya categoría es “Doors” la lista que saldría del “IN” del nodo FilterByBoolMark serían únicamente las puertas.

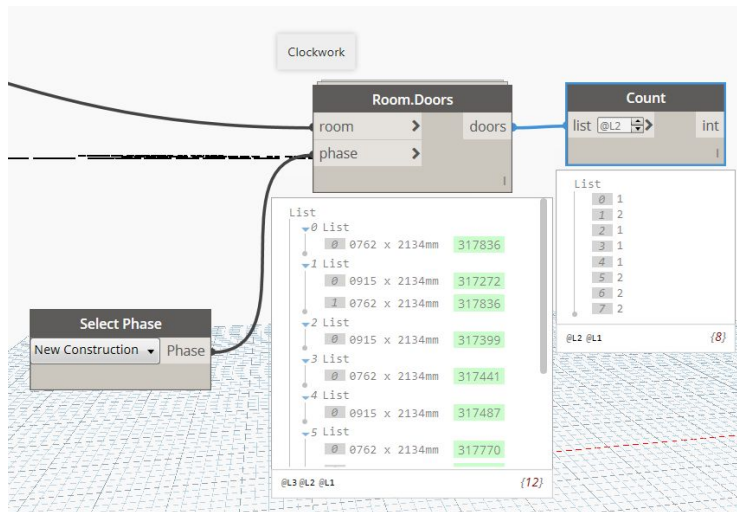
Estos ejemplos los hemos puesto para contar con distintos métodos para obtención de elementos.

Volvamos a la base del ejercicio, queríamos trasladar información desde las habitaciones a las puertas o viceversa.

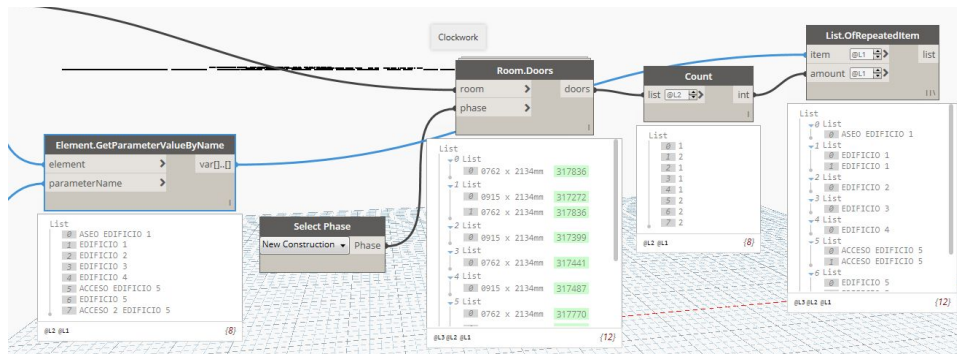
Una manera muy sencilla de vincular las habitaciones con sus puertas correspondientes es usando el package de Clockwork “Rooms.Doors”, puedes revisar cómo se instalan los packages en el punto 2.2. Este nodo sirve específicamente para relacionar cada habitación con las puertas que le corresponden.



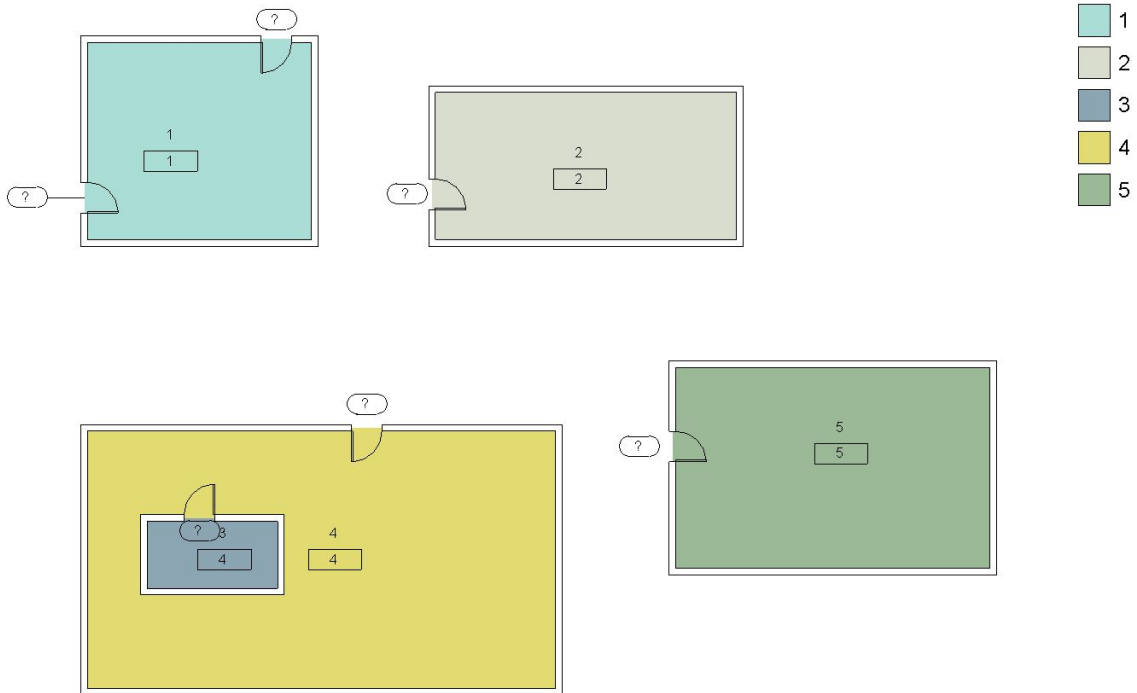
Como se puede ver en la imagen a cada habitación le puede corresponder desde una a infinitas puertas. Por ejemplo, a la habitación de índice [1] e id 317987 (los id de cada elemento aparecen en verde) le corresponden dos puertas de ids 317272 y 317886. Podemos usar el nodo “Count”, que cuenta el número de elementos de una lista, para saber cuántas puertas hay por habitación.



Si sumamos todos los elementos del count vemos que da en total 12 (puertas que corresponden a habitaciones) y, sin embargo, si vemos el principio del ejemplo veremos que había un total de 9 puertas. Esto se debe a que hay puertas que pertenecen a varias habitaciones simultáneamente. Queremos que cada puerta se asocie con el nombre de la habitación a la que corresponde. Pero tenemos más puertas que habitaciones. Para repetir los nombres de habitaciones tantas veces como puertas hay por habitación podemos utilizar el nodo “List.OfRepeatedItem”. Este nodo repite una lista de elementos tantas veces como “amount” introduzcamos.



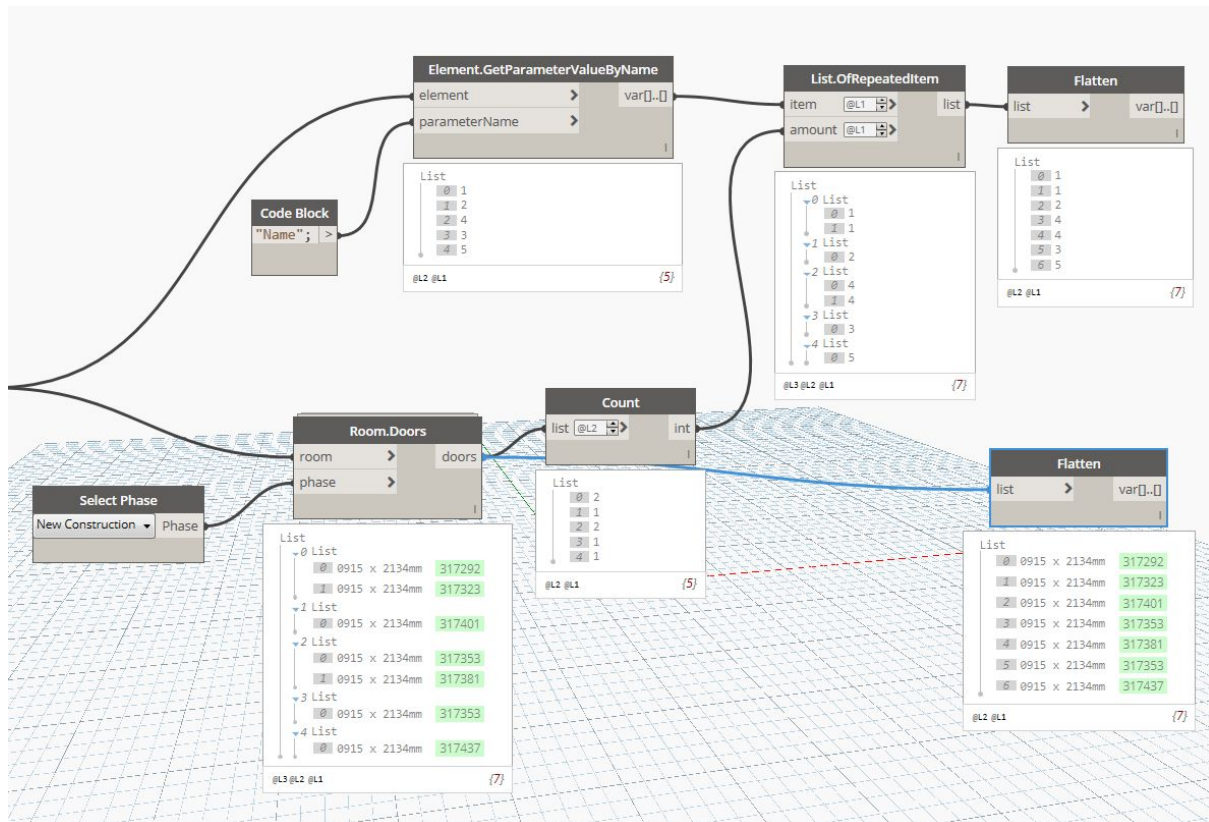
Vamos a simplificar un poco más el ejercicio para que haya menos variables, nos quedamos para ello solo con cinco habitaciones de nombres: 1,2,3,4 y 5 y vamos a trasladar la información de los nombres a las puertas. Dejamos una única puerta compartida entre dos habitaciones.



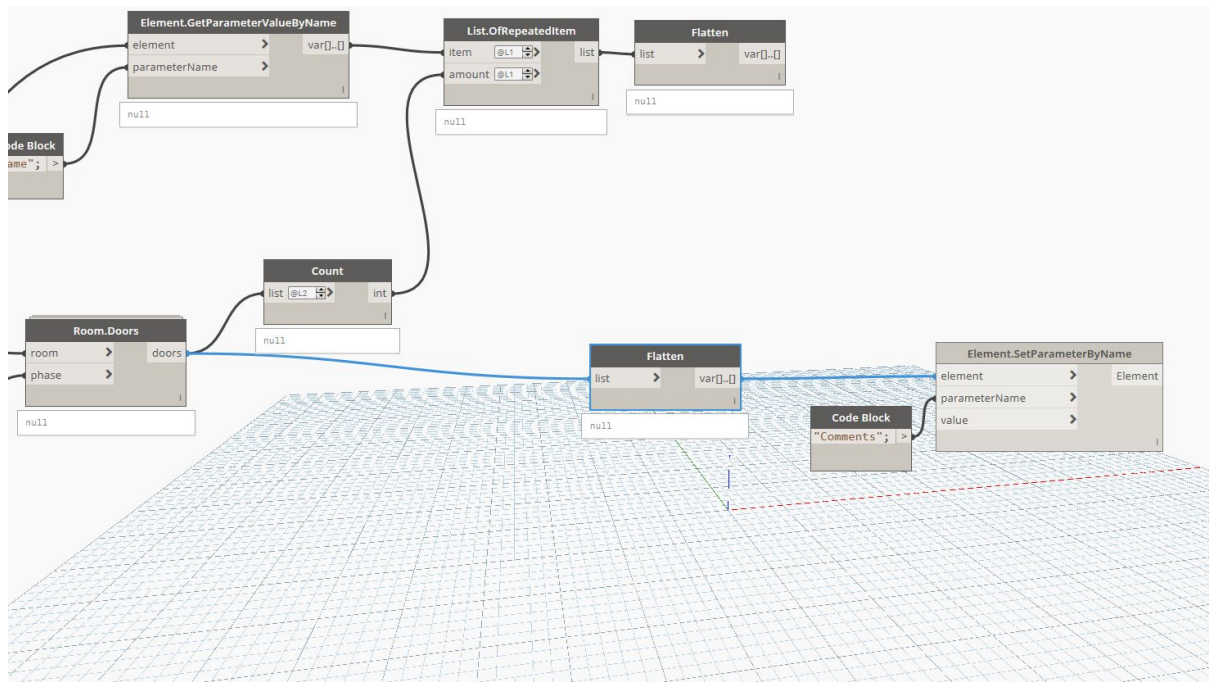
Desde este punto volvemos a donde estábamos en el ejemplo anterior, con los nombres repetidos tantas veces como puertas hay por habitación y las puertas por otro lado.

Vamos a “aplanar” las listas que hemos obtenido, ahora que ya sabemos que cada número corresponde con una o varias puertas. Este aplanamiento, dejar las listas como una única lista en lugar de una lista con varios niveles, se consigue con el nodo “Flatten”.





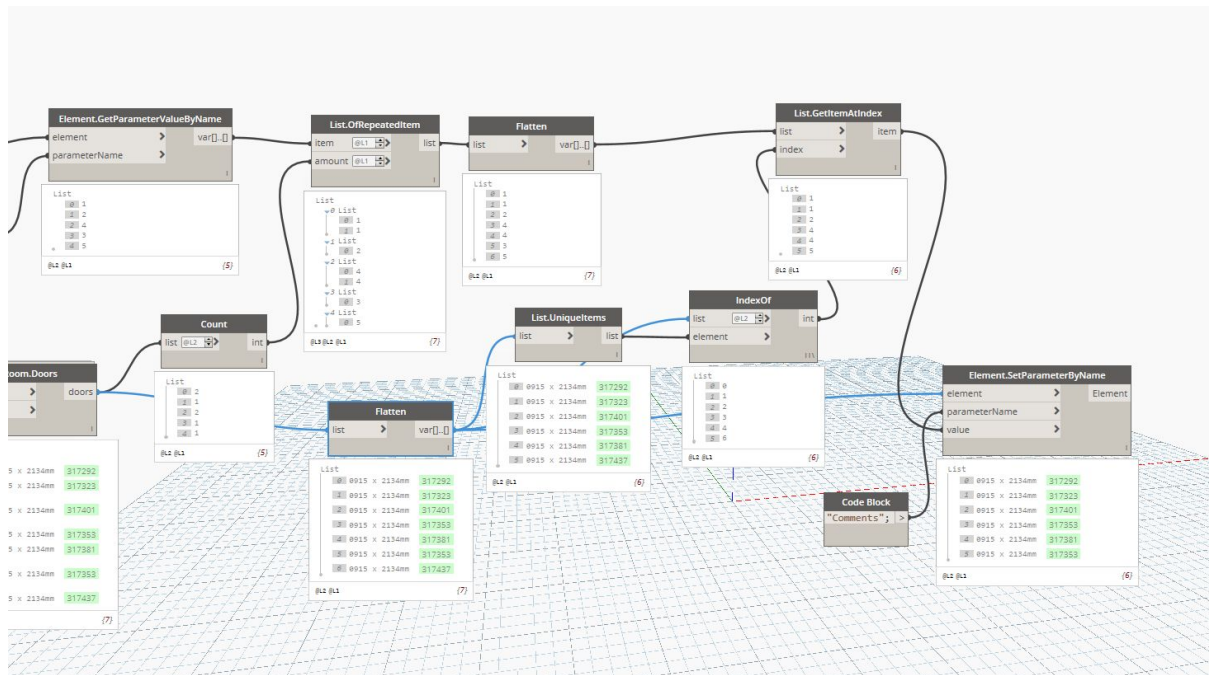
OJO EN ESTE PUNTO: si intentásemos tal cual introducir la información del nombre en las puertas nos daría un error que podría hacer que el programa se cerrara. Esto se debe a que hay dos elementos con el mismo id de puerta, ¿Recordáis? es la puerta que compartían dos habitaciones. El programa intentaría introducir dos informaciones contradictorias en el elemento. Solo queremos un valor por cada elemento. Entonces ¿qué podríamos hacer?



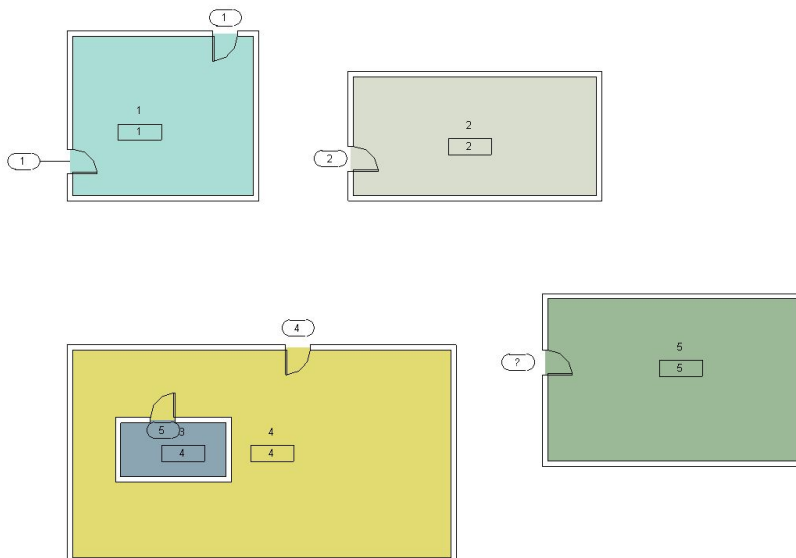
Una opción para quedarnos solamente con un valor, en caso de que nos diera igual cuál de los dos valores escoger es usar el nodo “List.UniqueItems”. Este nodo coge todos los elementos de una lista y se queda solo con aquellos que no están repetidos.

Después, para recuperar el orden en el que estaba la lista de puertas, usamos el nodo “IndexOf”. Este nodo busca los elementos en una lista y devuelve en orden el índice en que estaba el elemento en una lista.

Queremos coger los nombres de las habitaciones en el orden en el que hemos cogido las puertas. Para conseguirlo usamos el nodo “List.GetItemAtIndex”. Este nodo selecciona elementos de una lista en función de su índice. La lista que introducimos es la del nombre de las habitaciones en el orden de los tipos de puertas.



Para finalizar utilizamos el nodo “Element.SetParameterByName” para asignar al parámetro “Comments” de la lista de elementos “Puerta” la lista de nombres que hemos adquirido.

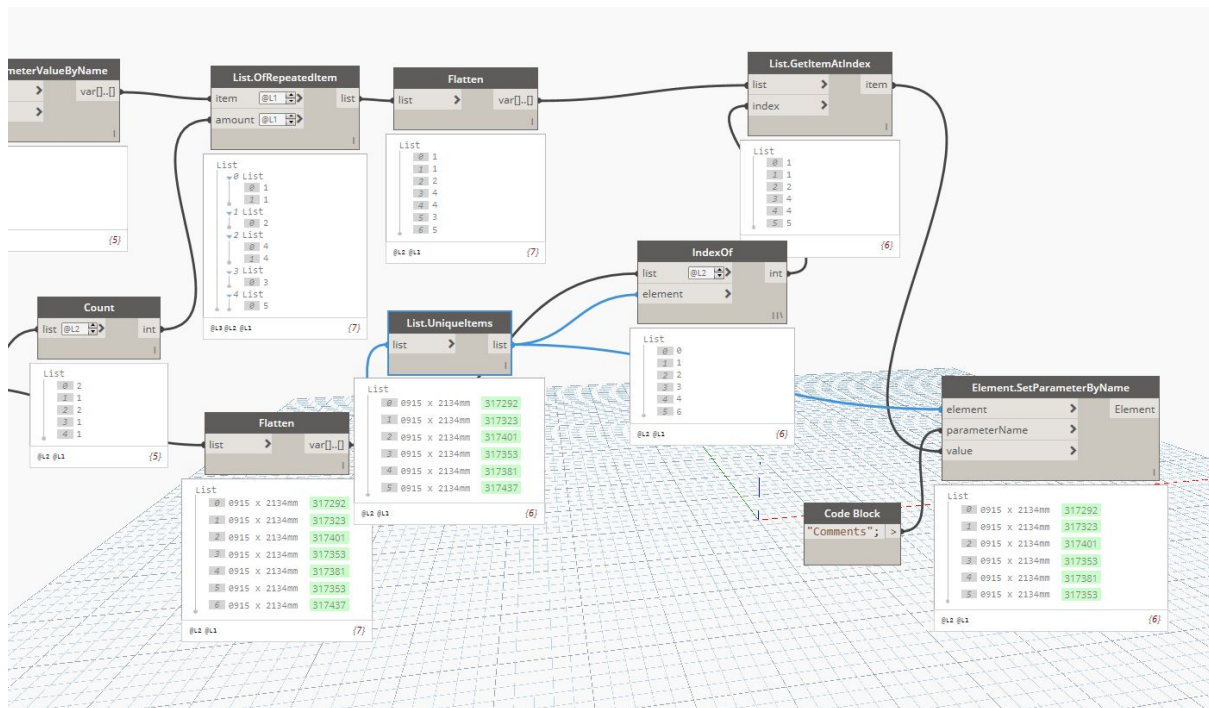


¿Y ya está? Pues no, fijaos bien en la imagen de arriba. Nos hemos confundido en algún paso. ¿Qué ha pasado?

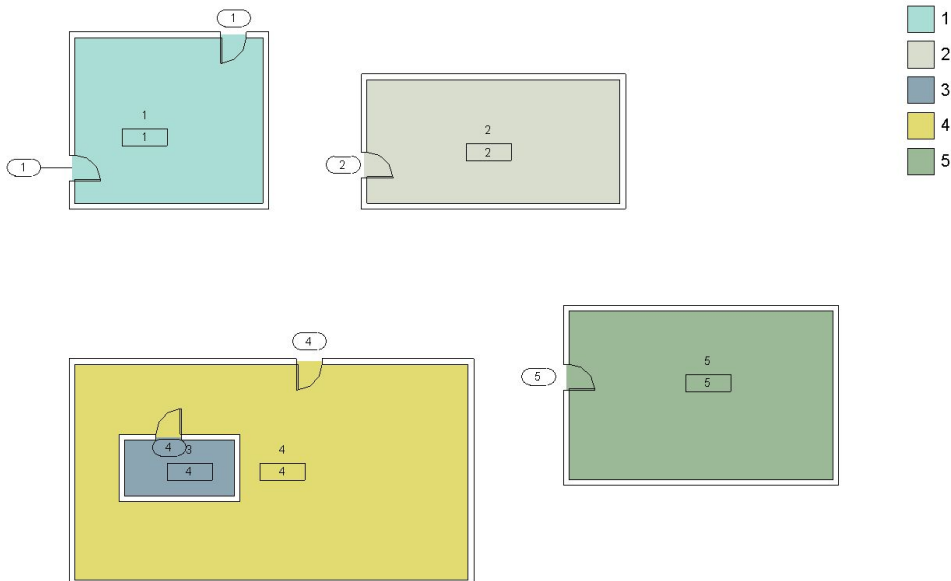
Si os fijáis arriba, en lugar de coger la lista de elementos únicos de puerta, que para eso la habíamos filtrado y ordenado por índices, y luego utilizado como base para ordenar los nombres, hemos cogido directamente las puertas que salían del nodo de “Flatten” como resultado el orden no era el correcto. Esto suele pasar en la práctica diaria y tenemos que ser capaces de detectar y corregir los errores.



En este caso bastaría con coger el input de element del nodo Element.SetParameterByName directamente desde el output “list” del nodo de List.UniqueItems que contenía los elementos únicos de las puertas.

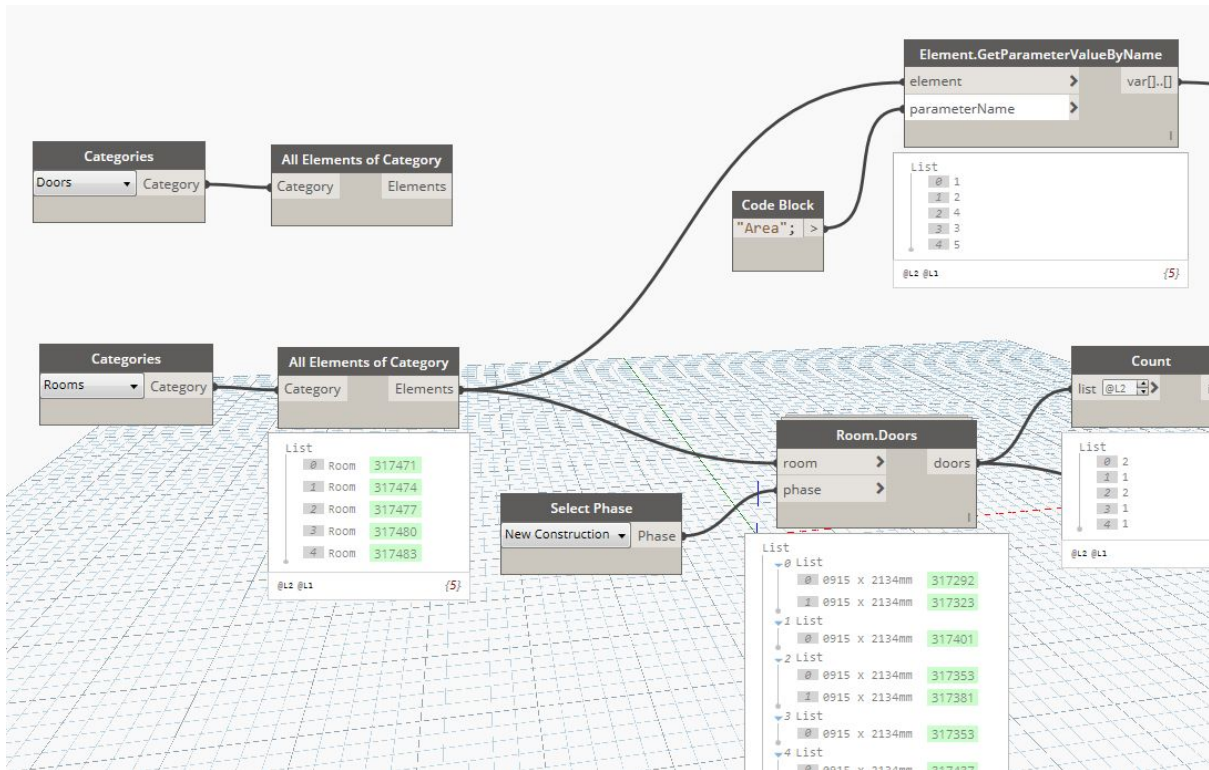


Una vez reordenado el Dynamo podemos volverlo a pasar obteniendo, esta vez, el resultado deseado.

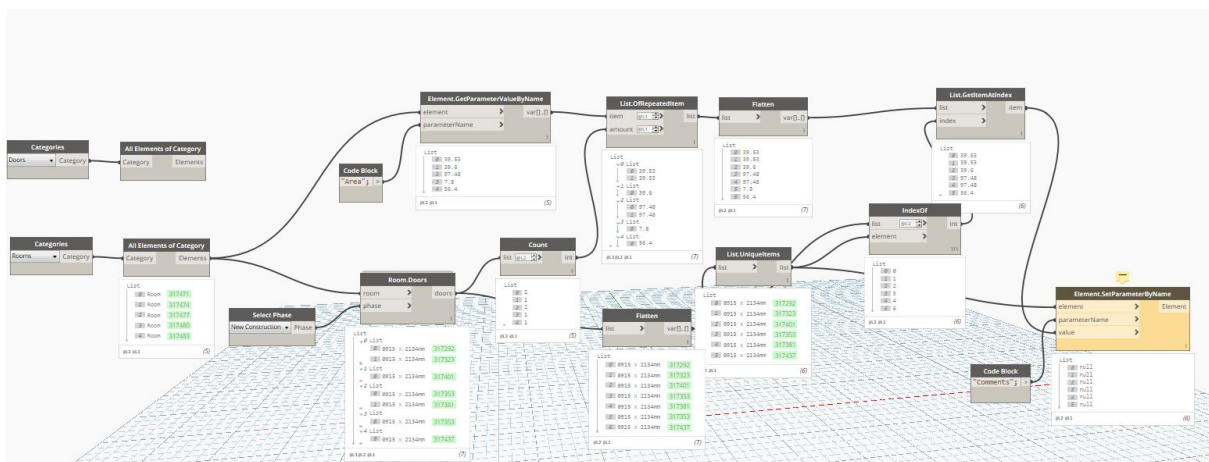


¿Y si en vez de querer transferir el nombre de las habitaciones a las puertas quisiéramos utilizar otro parámetro, como por ejemplo el área?

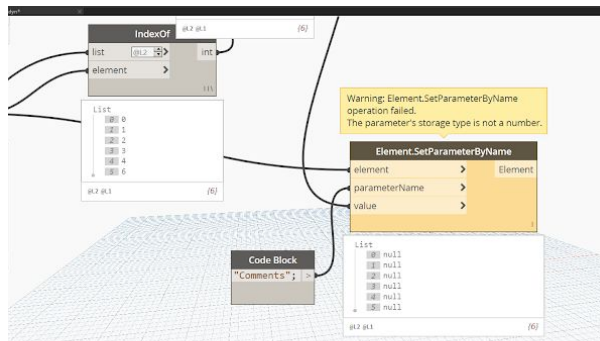
Lo primero que intentaríamos sería cambiar directamente el nombre del parámetro al principio de la definición, donde tomábamos el valor del parámetro nombre (Name) pondríamos ahora “Area”.



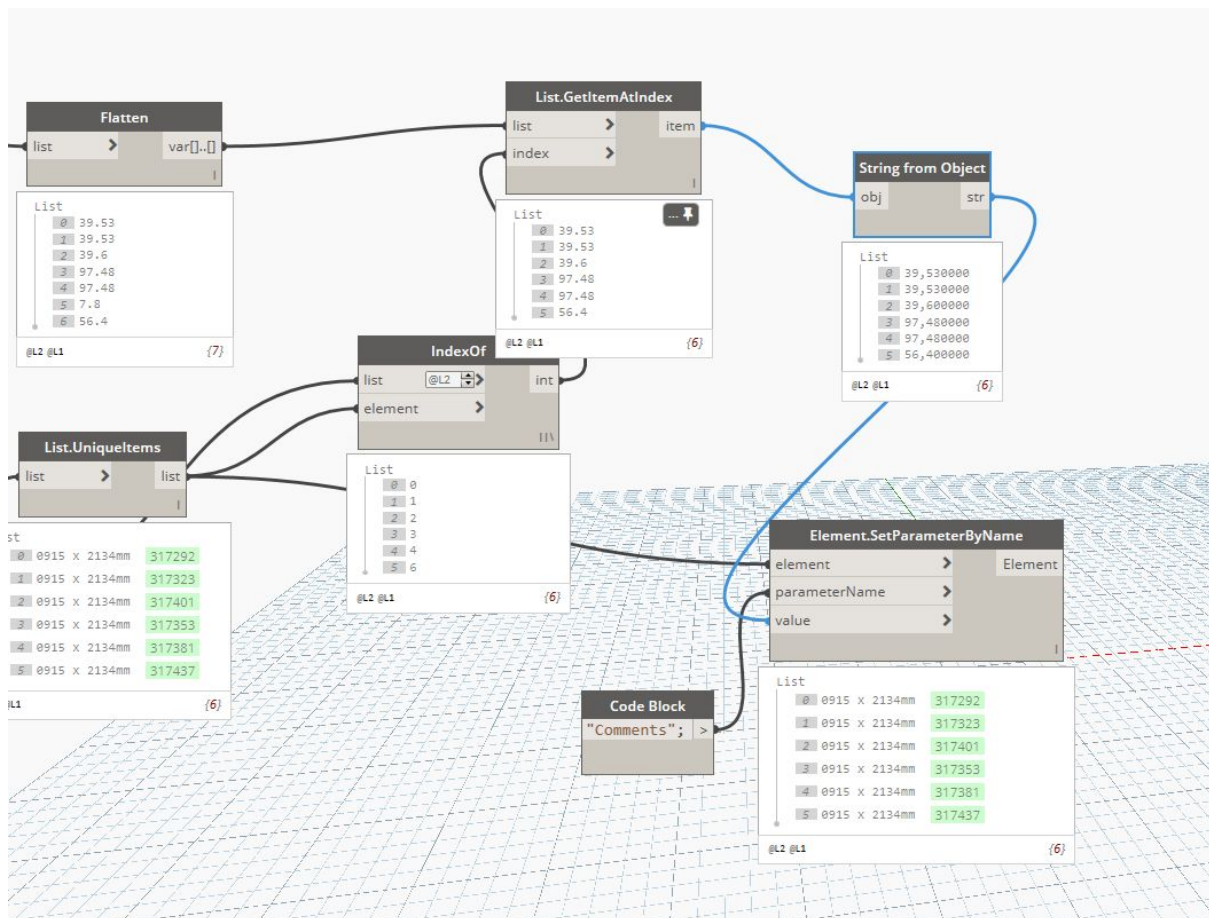
Al intentar pasar directamente la definición ésta nos daría un error en el último paso. ¿A qué se debe? Fijaos en que estamos intentando introducir la información del área de las habitaciones, un número, en el parámetro "Comments", que es de tipo texto.



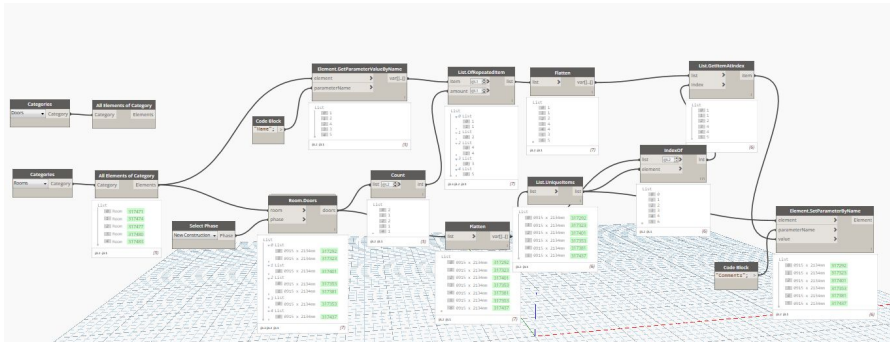
Si lees el error que da el último nodo, podéis leerlo pinchando sobre el diálogo amarillo, os dice "parameter storage type is not a number", es decir, que los valores que esperaba el nodo de "Element.SetParameterByName" en función del tipo de parámetro "Comments" no es un número.



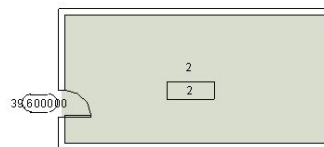
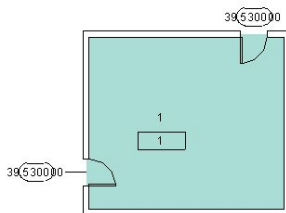
¿Qué podemos hacer en este caso? Es posible convertir el número en un texto utilizando el nodo “String from Object” del que hablábamos anteriormente.



Una vez hecho este cambio el Dynamo funcionará como lo hacía cuando transferíamos nombres de habitaciones.

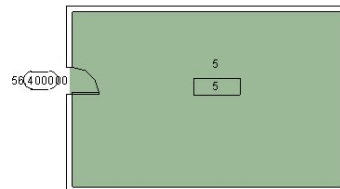
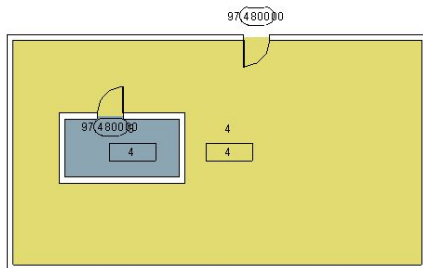


Este sería el resultado final de la definición.



Room Legend

- 1
- 2
- 3
- 4
- 5



Podéis descargar el dynamo en el link: <http://bit.ly/2oNrQBC>

Y la base de Revit (versión 2017) en el link: <http://bit.ly/2pfYj6T>

**WWW.ESPECIALISTA3D.COM**

**FUNDAMENTOS DE DYNAMO**

**Aprende mucho más Dynamo aquí:**

**<https://especialista3d.com/dynamo-like-a-pro/>**



#### **4. NOTA FINAL & CONTACTO**

ESTA GUÍA PERTENECE A [WWW.ESPECIALISTA3D.COM](http://WWW.ESPECIALISTA3D.COM)  
SI HAS RECIBIDO ESTA GUÍA ES PORQUE ERES SUScriptor DE ESPECIALISTA3D.  
POR FAVOR, ELIMINA ESTA GUÍA SI TE HA LLEGADO POR ERROR

Podéis enviar vuestros comentarios de la guía a [Luisa@especialista3d.com](mailto:Luisa@especialista3d.com)  
Leo personalmente todos los emails.