

**Instituto Tecnológico de Costa Rica**  
**Escuela de Ingeniería en Construcción**

Desarrollo de un sistema automatizado para la extracción de información, análisis  
y selección de estructuras temporales de encofrado a partir de modelos  
nDimensionales BIM

Proyecto final de graduación para optar por el grado de  
Licenciatura en Ingeniería en Construcción

Leonardo Sanabria Mejía

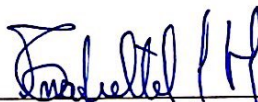
Cartago, Junio 2018.

## CONSTANCIA DE DEFENSA PÚBLICA DE PROYECTO DE GRADUACIÓN

Proyecto de Graduación defendido públicamente ante el Tribunal Evaluador, integrado por los profesores Ing. Gustavo Rojas Moya, Ing. Ana Grettel Leandro Hernández, Ing. Manuel Alán Zúñiga, Ing. Milton Sandoval Quirós, como requisito parcial para optar por el grado de Licenciatura en Ingeniería en Construcción, del Instituto Tecnológico de Costa Rica.



Ing. Gustavo Rojas Moya.  
Director



Ing. Ana Grettel Leandro Hernández.  
Profesora Guía



Ing. Manuel Alán Zúñiga.  
Profesor Lector



Ing. Milton Sandoval Quirós.  
Profesor Observador

# **Desarrollo de un sistema automatizado para la extracción de información, análisis y selección de elementos de encofrado modular a partir de modelos BIM nDimensionales**

# Abstract

Formwork Design is vital to the majority of big scale construction projects and represents a relevant percentage regarding concrete constructions costs, therefore, the automation of this process results in a direct positive impact on both the global project cost and execution time.

However, traditional practices in this area do not efficiently solve this matter due to their reliance on heuristic data extraction from 2D plans, computerized models or any other sort of guides that always end with manual work. Building Information Modelling philosophy purports a way to assemble high quantities of information in compact models and offer its availability for the stakeholders.

Through the use of a Custom Scripted Algorithm developed in the Visual Programming Platform of Dynamo and Custom Python Scripts, this Case Study shows an effective method of data extraction from a BIM Model and an incipient insight on how the outputs can be interpreted and used for the in site decision-making process regarding the selection of modular formwork panels for different reinforced concrete elements.

Finally, the author establishes the need for further research and development in the issue; the reason for which this work and the files herein described can be accessed and modified by any future researchers.

**Keywords:** Building Information Modelling, BIM, Formwork, Revit, Dynamo, Automated Data Extraction.

# Resumen

El diseño del encofrado es vital para la mayoría de los proyectos constructivos de gran envergadura y representa un porcentaje relevante en términos de costos, por lo tanto, la automatización de este proceso resulta en un impacto positivo directo tanto en el costo global del proyecto, como en su tiempo de ejecución.

Sin embargo, las prácticas tradicionales en esta área no responden eficientemente a este problema por su gran dependencia en la extracción heurística de información de planos bidimensionales y otros tipos de guías que implican labor manual. La filosofía BIM provee una presentación de altas cantidades de información en modelos accesibles a todos los involucrados.

A través del uso de algoritmos desarrollados en la plataforma de programación visual Dynamo y scripts en Python, este caso de estudio muestra un método efectivo de extracción de información desde modelos BIM y una demostración de cómo los resultados pueden ser interpretados para la toma de decisiones en la selección de paneles de encofrado modular para concreto reforzado.

Finalmente, el autor establece la necesidad de mayor investigación y desarrollo en este tema, por lo tanto, este trabajo y los archivos relacionados son de acceso libre y pueden ser modificados por cualquier futuro investigador.

**Palabras Clave:** Building Information Modelling, BIM, Encofrado, Revit, Dynamo, Extracción automatizado de la información.

# **Desarrollo de un sistema automatizado para la extracción de información, análisis y selección de elementos de encofrado modular a partir de modelos BIM nDimensionales**

# **Desarrollo de un sistema automatizado para la extracción de información, análisis y selección de elementos de encofrado modular a partir de modelos BIM nDimensionales**

LEONARDO SANABRIA MEJÍA

Proyecto final de graduación para optar por el grado de  
Licenciatura en Ingeniería en Construcción

Junio de 2018

INSTITUTO TECNOLÓGICO DE COSTA RICA  
ESCUELA DE INGENIERÍA EN CONSTRUCCIÓN

# Contenido

PREFACIO.....	1
RESUMEN EJECUTIVO.....	2
INTRODUCCIÓN.....	5
MARCO TEÓRICO.....	7
METODOLOGÍA.....	10
CASO DE ESTUDIO.....	13
RESULTADOS.....	23
ANÁLISIS DE RESULTADOS.....	26
TRABAJO FUTURO Y LIMITACIONES.....	29
CONCLUSIÓN.....	31
RECOMENDACIONES.....	32
APÉNDICES	
ANEXOS	
REFERENCIAS	

# Prefacio

La presente investigación responde a la necesidad de incorporar metodologías más eficientes y modernas en el proceso de preconstrucción de un proyecto constructivo, y más específicamente, en la selección de elementos de encofrado a ser utilizados en los proyectos con elementos estructurales de concreto reforzado, los cuales representan la mayoría en Costa Rica.

La automatización de procesos industriales no es una novedad en la industria manufacturera, de donde se extrae que, dentro de las prácticas establecidas por la filosofía Lean Construction, se busca la incorporación de esta tendencia a la automatización en la industria constructiva también, con el fin de obtener procesos más eficientes, eficaces y seguros.

En el escenario actual para la selección de elementos de encofrado modular en un proyecto constructivo típico, esta actividad se realiza mediante la decisión heurística a través del criterio de ingenieros experimentados, basados en: parámetros dimensionales, esfuerzos ejercidos por el concreto en los elementos de formaleta y el conocimiento obtenido de experiencias previas. Si bien es cierto esta práctica ha sido eficaz en entregar resultados, es una tarea matemáticamente sencilla, la cual puede ser emulada mediante una corroboración de reglas básicas previamente establecidas, obtenidas mediante criterios objetivos.

El objetivo principal de este proyecto es lograr el desarrollo de una herramienta que permita la automatización del proceso de selección de elementos de encofrado con el fin de evitar que el ingeniero competente invierta tiempo en una tarea que puede ser resuelta por un algoritmo, y el paradigma tradicional experimente un cambio en la inversión de varias horas en la investigación de métodos de encofrado, hacia la revisión puntual de los parámetros de salida del algoritmo por el mismo ingeniero. Es decir, reducir el tiempo que requiere

este proceso en la actualidad de varias horas de trabajo profesional a una cuestión de minutos, donde el sistema brinde una respuesta automática y procesada para la última revisión del ingeniero.

Agradecimientos:

Agradezco profundamente al Dr. Ehsan Rezazadeh Azar, al Departamento de Ingeniería Civil y a la Oficina Internacional de la Lakehead University, quienes hicieron este proyecto posible. También a la Ing. Ana Grettel Leandro y a todos mis profesores en el Tecnológico de Costa Rica por su atenta guía. A mi madre Dolly Mejía, mi hermana menor Dolly Sanabria y mi padre Keylor Sanabria, quienes siempre creyeron en mí, a mis hermanas Mónica, Kathy y Gigi y familia por apoyarme, también a Fernando, Priscilla, Javier, María Jesús y a Kevin por su cercanía y sus palabras de aliento durante la carrera. Así como a Luis Ángel Simón, por su ayuda técnica en el proceso.

Finalmente agradezco infinitamente a Mariana Montenegro, mi compañera de vida, sin quien absolutamente nada de esto hubiera sido posible y quien estuvo conmigo en todos y cada uno de los procesos necesarios para llegar hasta este cometido, brindándome aliento e inspiración. Esto es para Vos, es para Nosotros.



# Resumen ejecutivo

La tecnología Building Information Modeling (BIM) ha sido incorporada en la industria constructiva desde hace poco más de una década, lo cual ha abierto nuevos frentes de trabajo e investigación para los académicos relacionados con este sector de la economía. Una de las características principales que la incorporación de esta tecnología ha traído consigo desde sus inicios es que los límites o fronteras de su aplicación están en un proceso constante de expansión. Precisamente dentro de esta expansión se encuentra el diseño y desarrollo de estructuras temporales para los proyectos constructivos, y en particular, de los sistemas de encofrado modular para elementos de concreto reforzado.

Un Building Information Model o modelo BIM, como es popularmente referenciado, constituye una visualización tridimensional de una estructura en cualquiera de sus etapas constructivas, donde cada uno de sus elementos está cargado con información cuantitativa y cualitativa que puede ser accedida por los diferentes usuarios. La razón de ser de la tecnología BIM es que las distintas partes interesadas en un proyecto constructivo puedan acceder precisamente a la información que interesa a su quehacer.

El objetivo principal de este proyecto de investigación fue precisamente desarrollar un sistema automatizado (un algoritmo programado) que acceda a la información contenida en este tipo de modelos, trascendente para el proceso de diseño o selección de elementos de encofrado modulares, liberando así tiempo de trabajo por parte del ingeniero responsable que puede ser invertido en otras tareas. La razón por la cual se decidió abordar este tema desde un algoritmo es la facilidad de herramientas tecnológicas de software, tales como Autodesk Revit de facilitar la programación con los nodos de información contenidas en los mismos modelos, sin la necesidad de recurrir a software de terceras partes. En este caso en particular se decidió trabajar con el software Dynamo debido a su

interfaz de programación visual amigable con usuarios poco familiarizados con lenguajes de programación convencionales.

Para poder poner en marcha la lógica de extracción y procesamiento de información tras el algoritmo desarrollado para el proyecto, el autor desarrolló un modelo tridimensional arquitectónico y estructural en Autodesk Revit conteniendo los parámetros mínimos del diseño de una estructura tipo marco en concreto reforzado, donde cada una de sus vigas y columnas está indexada, de manera que puedan ser accedidas individualmente. El modelo desarrollado por el autor no pretende ser un modelo BIM, pero sí proveer los parámetros mínimos que tendría un modelo de este tipo para poder ser analizado por el algoritmo desarrollado.

La metodología seguida para el desarrollo de este algoritmo consistió en una evaluación constante del script sobre el modelo creado por el autor (Modelo Piloto), en la cual se determinaron seis fases distintas para garantizar su funcionamiento, las cuales fueron: (a) extracción automatizada de la información desde el modelo Revit, (b) Cálculo y verificación de unidades SI, (c) Reporte preliminar al archivo desde MS Excel, (d) Cálculo de la carga ejercida en el encofrado, (e) Proceso de selección de la combinación óptima de paneles para el elemento de concreto reforzado analizado y (f) La entrega de los parámetros de salida finales, tanto un documento resumen automáticamente generado en MS Excel, como la inscripción de la información concerniente a la selección del encofrado en los elementos del modelo.

A partir de la fase d, una vez que fue comprobada la capacidad del algoritmo de extraer y procesar la información contenida en el modelo, se comenzó a probar el algoritmo constantemente en el Modelo de Validación, que corresponde a un modelo BIM nDimensional brindado por el profesor tutor y desarrollado por una tercera parte, esto con el fin de valorar la efectividad de la extracción y el procesamiento de

la información por parte del algoritmo en un modelo más complejo que el piloto diseñado por el autor.

El algoritmo en su totalidad fue desarrollado sobre la plataforma de programación visual de Dynamo para Revit, la cual corresponde a diagramas de flujo que concatenan nodos funcionales, donde cada uno de estos es un pequeño script que recibe valores de entrada y entrega parámetros de salida para ser recibidos por el nodo subsecuente. Toda la información contenida y utilizada por el algoritmo para el procesamiento de la información es extraído directamente desde el modelo de Revit, o bien, desde la base de datos de los sistemas de encofrado seleccionados para el análisis, los cuales se encuentran debidamente identificados en el script. La información contenida en la base de datos es ingresada manualmente por el autor, donde se incluye el nombre del sistema, su orden de preferencia relativa, la carga máxima de presión lateral de concreto que pueden soportar sus elementos y las dimensiones de los elementos representadas como listas de *integers* independientes, ya que son intercombinables, es decir, todos los anchos pueden combinarse con todas las longitudes.

Con respecto de las funciones utilizadas en este algoritmo, este se sirve de nodos incluidos en las bibliotecas por defecto de Revit, de nodos importados desde bibliotecas encontradas en línea y de nodos programados mediante Python 2.7, tal y como es el caso de la fase e del algoritmo, ya que esta representa una de los procesos de verificación de reglas más complejos, entonces se hace necesario recurrir a esta sintaxis.

A grandes rasgos, la funcionalidad de este script consiste en identificar las dimensiones extraídas del elemento desde el modelo y fijarlas como una medida objetivo, donde, en caso de ser necesario, se añade una dimensión determinada por el autor para facilitar el proceso de colado de concreto en sitio. Una vez la dimensión objetivo se tiene definida, el algoritmo se encarga de iterar con las dimensiones de paneles existentes con el fin de encontrar la forma más eficiente de completar la distancia objetivo usando la menor cantidad de paneles posibles, y de ser viable, repetir los paneles de elementos previamente colados. Los valores de entrada que recibe este "script", por consiguiente, corresponden a las dimensiones de los elementos por ser colados y

las bases de datos respectivas para cada uno de los sistemas de encofrado modular.

Los resultados del algoritmo se componen de vectores cargados de variables *String* e *Integer* almacenadas en directorios temporales en Dynamo, por lo que, para poder interpretar los resultados, es necesaria la incorporación de la fase f, donde los vectores son ordenados para poder entregar una versión legible y entendible de los resultados. El autor decidió para este fin dar dos entregables como productos del algoritmo, donde el primero corresponde a un archivo de MS Excel automáticamente generado en el directorio del escritorio del usuario, el cual contiene dos pestañas denominadas "Columns" y "Beams", correspondientes a las Columnas y Vigas respectivamente y donde en cada una de dichas pestañas se presenta la información extraída y procesada de manera resumida y tabular.

Adicionalmente se incluye información importante, tal y como los volúmenes de concreto determinados para cada elemento por ser colado y la ubicación de cada uno de estos. El segundo entregable corresponde al mismo modelo de Revit con la información obtenida del algoritmo escrita en la celda "Comments" de la pestaña de propiedades en cada uno de los elementos analizados, de manera que al seleccionar cualquier elemento tipo marco en el modelo pueda verse la más óptima selección de paneles de encofrado modular para este.

Si bien es cierto, el autor encuentra la respuesta dada por el algoritmo como satisfactoria para efectos prácticos en proyectos constructivos, se determina que futuras investigaciones pueden extender el campo de aplicación del algoritmo o la lógica detrás del mismo, así como brindar un formato de valores de salida que sea más amigable y fácil de entender por parte de los usuarios. Una de las maneras más claras de brindar la información sería mediante respuesta gráfica, donde se importen modelos tridimensionales desde un software de modelado a Revit en formato ifc y se referencien los puntos extremos del elemento como coordenadas tridimensionales para presentarlo de manera conjunta con el elemento seleccionado, de forma aislada al resto del modelo.

Otra de las formas en las que el autor considera que se puede extender el campo de aplicación del algoritmo es incorporando las

ajustes correspondientes al "script" para permitirle al algoritmo trabajar con unidades del sistema imperial, así como otras geometrías para secciones transversales a parte de la rectangular. Dichas adecuaciones no son complejas en su naturaleza, pues requieren únicamente adaptaciones matemáticas simples a los nodos de procesamiento del "script".

En la actualidad y siguiendo la metodología de diseño convencional para los elementos de encofrado en estructuras de concreto, este proceso usualmente se extiende por horas, razón por la cual los costos de labor representan un sustancial 80% en los costos totales de encofrado para elementos de concreto, que a su vez representan un 49% de los costos totales de construcción de elementos de concreto. Este proyecto de investigación determinó y analizó una reducción en los tiempos de procesamiento de la información para la selección de elementos de encofrado modular de magnitud de orden de horas a minutos, ya que el procesamiento del algoritmo diseñado en la unidad donde fue procesado requirió menos de 30 segundos para brindar ambos entregables. Es decir, el proceso es simplificado de la lectura manual de los modelos, consulta de sistemas de encofrado y toma de decisiones por parte del ingeniero responsable (proceso que suele consumir horas) a la interpretación de los valores de salida del algoritmo por parte de un ingeniero competente, llevando tan solo algunos minutos.

Tanto el conocimiento extraído del proceso de investigación, como los productos finales del proyecto de investigación son puestos al servicio de la comunidad académica por parte del autor con el fin de continuar con el trabajo hasta transformarlo de una lógica incipiente, hasta un algoritmo completo e integral de análisis de elementos de concreto reforzado para la selección de su encofrado.

# Introducción

Las estructuras temporales están presentes en cada proyecto constructivo, indiferentemente de su ubicación y magnitud. Estas estructuras juegan un papel trascendental en proyectos de gran envergadura, tanto en edificaciones como en infraestructura.

Tal y como se muestra en la Figura 01, al referirse al encofrado utilizado para la elaboración de elementos de concreto reforzado, el costo de este rubro puede representar hasta un 49% del total, del cual un 80% describe equipo (herramientas, andamios, arneses, etc.) y labor (manual y diseño). (S. W. Nunally, 2004).

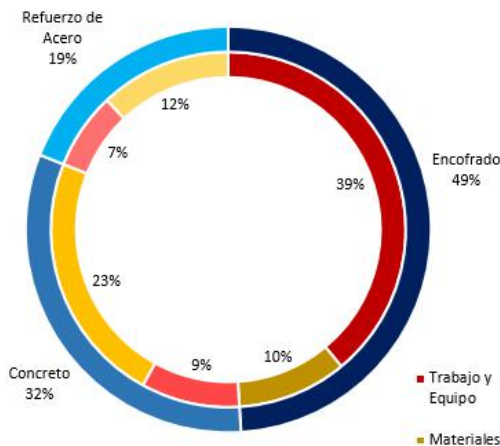


Figura 01. Distribución típica de los costos en la construcción de elementos de concreto reforzado.

Sin embargo, luego de reducir el campo de estudio a las prácticas actuales y tradicionales en el proceso de diseño y selección de elementos de encofrado, se puede notar que, en la mayoría de los proyectos, esta práctica se trata de un proceso reactivo, heurístico y subjetivo, frecuentemente basado en la experiencia personal de ingenieros competentes. (K. Kim, J. Teizer, 2014).

Si bien es cierto, la experiencia práctica y la experticia forman parte de los criterios más importantes y sustanciales en el equipo de un proyecto, usualmente su determinación es altamente subjetiva y no existe una forma de medirla cuantitativamente o garantizar su existencia. De este hecho surge la necesidad de desarrollar una herramienta para asistir a los ingenieros constructores en el proceso de determinación de los sistemas de encofrado que se ajusten mejor a las necesidades del proyecto analizado.

Para entender la importancia de este problema, es necesario comprender que el uso extendido de métodos manuales para el diseño de encofrado, incluso posterior a la inserción de sistemas de formaleta modulares y prefabricados al mercado, se conserva hasta la actualidad debido a la falta de estrategias sistematizadas o técnicas de planeamiento en estructuras temporales y, en particular, del proceso de encofrado. Esta realidad usualmente conlleva una lista de potenciales errores humanos, tales como: (1) la omisión de elementos necesarios en la matriz de encofrado, (2) la construcción de estructuras transitorias que no cumplen con los estándares mínimos, (3) la selección de elementos no adecuados en sistemas de formaleta y (4) la estimación incorrecta de cantidades de materiales para la construcción, basados en las dimensiones establecidas. (K. Kim, J. Teizer, 2014). Cualquiera de estos errores representa un aumento en los costos de construcción, lo cual puede ser fácilmente evitado mediante la estandarización del proceso de selección mediante la automatización en la extracción de información y la revisión de parámetros mediante algoritmos sencillos.

Aunado a esto, el uso de la filosofía Building Information Modelling en la construcción de proyectos de ingeniería representa una nueva gama de opciones para el procesamiento de la información contenida en los modelos y la

manera en la que esta se puede aplicar al diseño de estructuras temporales. Por consiguiente, cuando los conceptos de Constructability (Correctas prácticas de eficiencia en la industria de la construcción) son aplicados a la información extraída de un modelo, permiten la revisión de estos, de manera semejante a aquella en que un ingeniero experimentado lo realizaría, comparando el valor de entrada contra criterios establecidos anteriormente que han resultado exitosos. (Jing et al., 2013). En otras palabras, esta información permite la automatización de procesos simples de diseño mediante la revisión de ciertas reglas (previamente establecidas), comparando los parámetros intrínsecos de los elementos ingresados contra las propiedades características a cada sistema de encofrado y luego proponiendo la alternativa más adecuada para cada elemento.

Objetivo general: Desarrollar un sistema para asistir el proceso de toma de decisiones en la selección de los elementos de encofrado modular adecuados del sistema seleccionado.

Objetivos específicos:

1. Desarrollar el modelo BIM para el caso de estudio que permita la identificación de los elementos de concreto reforzado por analizar.
2. Implementar un algoritmo desarrollado en Dynamo que permita la extracción y el procesamiento de información desde el modelo BIM previamente preparado.
3. Plasmar la información procesada y analizada para la toma de decisiones en el proceso de selección de elementos de encofrado modular en formatos de uso comúnmente extendido.

Limitaciones:

1. Exclusividad de uso del algoritmo de análisis únicamente mediante la plataforma de Autodesk Revit, ya que Dynamo trabaja sobre este software.

2. El algoritmo desarrollado se limita a la aplicación en elementos de marcos estructurales de concreto reforzado con secciones transversales específicas.
3. El algoritmo desarrollado se limita a los procesos que puedan ser analizados mediante el uso de tecnologías BIM a través de la plataforma de Autodesk Revit.

# Marco Teórico

Building Information Modelling puede ser definida como la práctica o la filosofía de integrar información cuantificable y descriptiva en modelos multidimensionales, proveyendo un único modelo que cubre los intereses de múltiples interesados, el cual se beneficia de la tecnología disponible en la actualidad para el análisis de modelos n-dimensionales de construcción con información útil para todos los involucrados en el proyecto. Con el pasar del tiempo se ha demostrado y es generalmente aceptado el concepto de que la incorporación de prácticas propias de BIM en la industria de la construcción representa un incremento significativo en la eficacia y eficiencia en la entrega de resultados satisfactorios en los proyectos en cada una de sus fases (R. Volk et al. 2015). La razón tras de este incremento en la productividad experimentado por los proyectos en los que esta filosofía ha sido aplicada, reside en la mejoría directa lograda por la visualización, comunicación y, sobre todo, la integración de los diferentes procesos constructivos (L. Ding et al., 2014). En otras palabras, para múltiples procesos constructivos, tanto simples como complejos, se ha mostrado una mejora sustancial en el desarrollo de los mismos, debido al flujo constante de información entre todas (o casi todas) las partes del proyecto.

Dentro de la información contenida en los modelos BIM se encuentran los parámetros dimensionales de cada uno de los elementos estructurales y arquitectónicos de un proyecto, ya que estos no son definidos como conjuntos de líneas o vectores, como en las plataformas CAD, sino como objetos inteligentes e independientes entre sí, con características propias e índices para su referencia.

Un Building Information Model adecuadamente documentado puede ser utilizado para una amplia variedad de fines en proyectos de gran escala, tales como: (1) Estimación y cuantificación de materiales, (2) Planeamiento y

control de seguridad, (3) Control de Inventarios, (4) Identificación de errores de diseño, (5) Planeamiento de la construcción y análisis de la viabilidad de la construcción (constructability), (6) Seguimiento de tareas constructivas, (7) Proceso de prefabricación y modularización, (8) Planeamiento y distribución del sitio constructivo (D. Mansuri et al., 2017).

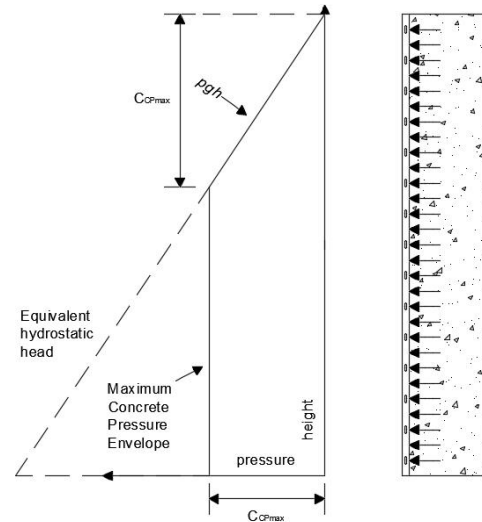


Figura 02. Diagrama de distribución de presión lateral del concreto en estado fresco.

Sin embargo, el límite de las aplicaciones de la filosofía BIM está en constante expansión, tal es así que, en años recientes, las estructuras temporales han ido ganando terreno en las prácticas de planeamiento y diseño de sitio. Este tipo de estructuras había sido dejado de lado en el pasado en los métodos de visualización de proyectos, tales como planos, diagramas y modelos tridimensionales; sin embargo, la inclusión de los sistemas de estabilización de taludes, andamiaje y encofrado es una práctica creciente entre los modeladores BIM.

Dentro de las estructuras transitorias previamente mencionadas, el caso de los elementos de encofrado destaca en importancia debido a su aplicación en una vasta mayoría de los proyectos constructivos, indiferentemente de su funcionalidad, naturaleza o emplazamiento. El diseño de este tipo de estructuras incluye también los procesos de selección de elementos modulares, ya que, si bien es cierto los mismo no exigen revisiones por flexión, momento, compresión, aplastamiento local y cortante, requieren análisis más orientados hacia su logística, aplicabilidad y funcionalidad en el proyecto donde han de ser utilizados. El tiempo invertido por los ingenieros en el diseño o la selección de estos elementos varía sustancialmente de proyecto a proyecto, sin embargo, usualmente se extiende de veinte a treinta minutos por cada detalle diferenciado en elementos de estructuras tipo marco, por lo que, como norma general, consume tiempos en el orden de horas para los ingenieros responsables. (A. Krawczybska-Piechna, 2017).

La investigación moderna por formas de acelerar el diseño en sistemas de encofrado fue primeramente investigada y desarrollada por Richard C. Ringwald (1985), quien comenzó por resumir y compilar las prácticas más comunes para el diseño de encofrados en su momento, predominantemente diseñadas según la particularidad de cada proyecto en madera, con la cual pudo desarrollar una guía paso a paso cubriendo los aspectos que él consideró como esenciales. Luego de describir las variables que afectan el encofrado como tal, propuso curvas gráficas específicas para acelerar el proceso de diseño para los demás ingenieros, con las cuales, cada revisión de las fuerzas ejercidas sobre la formaleta se limitaba a revisar la magnitud del esfuerzo de concreto en ella contra sus dimensiones en la gráfica que mejor describiera la situación en sitio.

Hasta la fecha, la practica más extendida para este proceso en los proyectos de construcción en Costa Rica consiste en que los ingenieros estructurales diseñan la estructura de concreto y posteriormente el contratista o ingeniero constructor adaptan la configuración del encofrado a dicho elemento, basado principalmente en las dimensiones y geometría de cada situación en particular y considerando las facilidades de anclaje propias de la ubicación del elemento. (D. Mansuri et al., 2017).

Desde la introducción de los sistemas de encofrado modular prefabricados y reutilizables en la industria moderna de la construcción, su uso ha pasado de ser un puñado de casos aislados a ser extendido a la mayoría de los proyectos e incluso a ser un entregable esperado al trabajar con grandes proyectos. Tales sistemas facilitan el proceso de colado de concreto, ya que, para elementos simples y repetitivos, tales como las vigas y columnas estructurales con sección transversal rectangular, el ingeniero encargado puede evitar los cálculos de flexión, cortante, momento y aplastamiento local, los cuales serían necesarios de trabajar con el sistema tradicional de encofrado diseñado en sitio para madera. Aunado al efecto ahorrador de tiempo resultado de la utilización de paneles prefabricados para los ingenieros responsables, la carga ejercida por el concreto en las paredes del encofrado representada en la Figura 02 y los parámetros dimensionales que definen los elementos por ser colados, deben ser siempre interpretados por ingenieros competentes para verificar su cumplimiento con las características técnicas del sistema utilizado. El autor, al revisar la bibliografía consultada, considera esta tarea como un proceso simple y repetitivo, sin embargo, como altamente demandante en términos de tiempo, por lo tanto, surge la necesidad de automatizarlo. En la construcción de edificios altos, el uso de elementos de encofrado modular de aluminio se extiende a más del 65% de los casos, y existen registros de esfuerzos realizados para mejorar la eficiencia del proceso de selección en investigaciones previas (T. Kim et al., 2012). Sin embargo, para los sistemas de encofrado modular, los esfuerzos de mejora nunca han sido intentados mediante la utilización de Dynamo para modelos de Revit con elementos de marco, tales como columnas y vigas.

Partiendo del entendimiento que la metodología tradicional para el proceso de selección de paneles de encofrado modular es altamente dependiente en la subjetividad y la intuición del ingeniero responsable, es importante señalar que la interpretación más amplia y precisa podría ser obtenida mediante técnicas de programación de lógica difusa (fuzzy logic). Estudios previos han demostrado la aplicabilidad de los conceptos de series de datos de lógica difusa en la industria de la construcción, con sistemas capaces de recibir variables tipo *Integer*,

*String* y *Floating* como parámetros de entrada e interpretarlos de acuerdo con las reglas obtenidas del proceso de razonamiento de ingenieros experimentados (E. E. Elbertagi et al., 2012). Sin embargo, para el objetivo de la presente investigación, se prefiere el razonamiento matemático determinístico por encima de la lógica difusa, debido a la objetividad y simplicidad de la tarea analizada.

En años recientes, varios autores han cubierto la investigación referente a la automatización de procesos generadores de horarios y cronogramas para colado de elementos de concreto y uso de formaleta o encofrado (T. Kim et al., 2012), planeamiento efectivo de encofrado (A. Krawczybska-Piechna, 2017), Herramientas para la administración del encofrado (D. Mansuri et al., 2017), Revisiones basadas en reglas previamente establecidas para la viabilidad de uso de encofrado en coladas de elementos de concreto horizontales (L. Jiang y R. M. Leight, 2015), así como la incorporación de métodos de medida de productividad utilizando BIM para encofrados estructurales. (J. Lee et al., 2017).



# Metodología

Como se explicó anteriormente, el objetivo principal de este proyecto fue la creación de un sistema, en este caso un algoritmo, para la extracción de información y procesamiento automatizado, orientado hacia la creación de un parámetro de salida que describa la combinación de paneles de encofrado modular más adecuada para cada elemento de concreto reforzado en el sistema de marcos de la estructura. Al inicio de la investigación, el algoritmo estaba originalmente concebido para trabajar con columnas estructurales, vigas estructurales coladas en sitio y zapatas de cimentación. Sin embargo, en vista de la gran variedad de opciones para el ingreso y creación de zapatas que existen en Autodesk Revit, incluyendo incluso elementos personalizados, el caso de estudio fue limitado a

los elementos de concreto reforzado en marcos estructurales (columnas y vigas estructurales).

El algoritmo para la extracción de información fue creado en Dynamo para Revit, en conjunto con subscrippts desarrollados en Python 2.7 en distintos nodos del algoritmo. El campo de trabajo del mencionado algoritmo se limitó entonces a los elementos tipo marco en concreto reforzado, analizando cada uno como un Smart object (objeto cargado con distintas clases de información en un modelo).

Considerando las diferencias en el proceso de colocación y montaje de formaleta para vigas y columnas, existen ligeras diferencias en la lógica para cada una, tal y como se muestra en las Figuras 03 y 04.

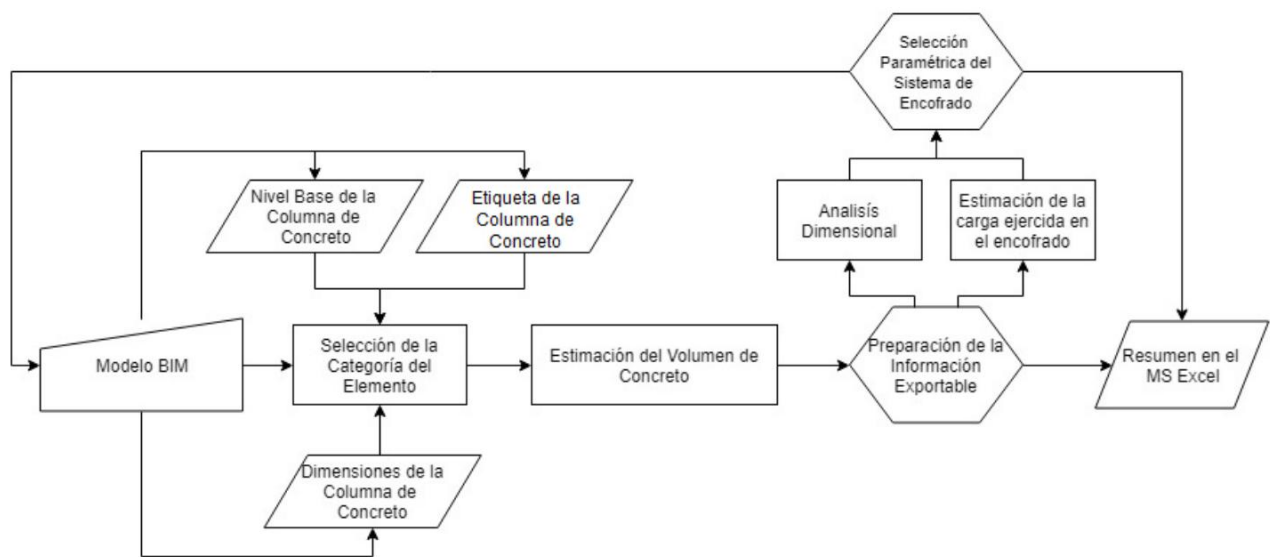


Figura 03. Esquema general para la selección de encofrado de Columnas de Concreto Reforzado

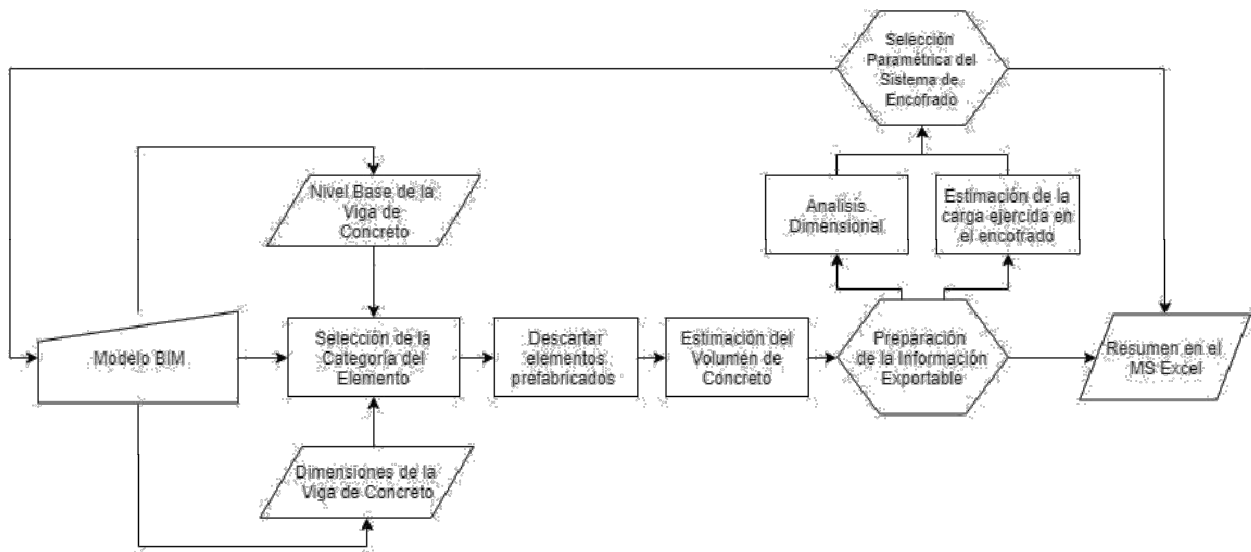


Figura 04. Esquema general para la selección de encofrado de Vigas de Concreto Reforzado

El algoritmo, al haber sido desarrollado en Dynamo, sobre un modelo de Autodesk Revit, requiere ser ejecutado de la misma manera cada vez que se requiera su aplicación. Para ambos tipos de elementos el algoritmo comienza filtrando la categoría para determinar si se trata de vigas o columnas. Se prosigue extrayendo la información dimensional desde el modelo, identificando cada Smart object según se encuentre indexado en él; información la cual luego es utilizada para estimar la cantidad de concreto requerido, así como la dimensión de cada una de las áreas que van a estar en la interfaz de contacto concreto – encofrado. En el

caso de las vigas estructurales es necesario discriminar los elementos prefabricados del análisis del algoritmo, ya que estos no requieren ningún tipo de formaleta según su naturaleza.

Una vez los volúmenes de concreto son estimados por el algoritmo, la preparación de la información para su exportación se hace siguiendo el diagrama descrito en la figura 5, donde ambos parámetros de selección para la formaleta son revisados, tanto los dimensionales como la carga del concreto fresco ejercida sobre las caras del encofrado, el cual se estima usando la Ecuación 01 y considerando el escenario más crítico para el colado de concreto, con una tasa

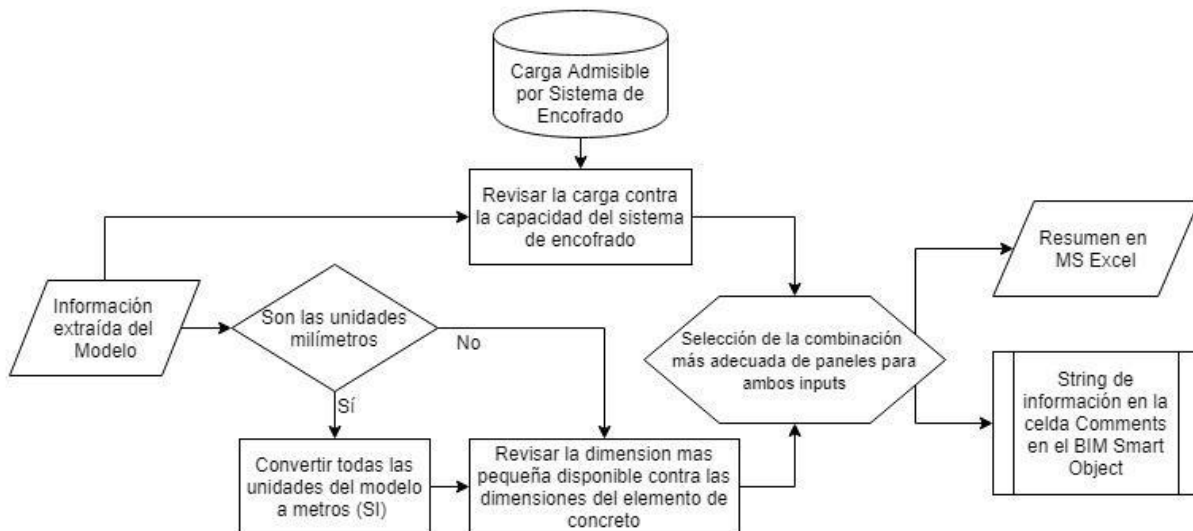


Figura 05. Diagrama de Flujo Específico para la preparación de la Información

de colocación (R) de 1,5 m/h para las columnas y 1,0 m/h para las vigas. Adicionalmente se considera una temperatura (T) de 10° centígrados (la mínima recomendada por el ACI), un coeficiente para cemento tipo I sin adiciones de ceniza volante o escoria de alto horno ( $C_c$ ) igual a 1 y un coeficiente de peso unitario ( $C_w$ ) igual a 1 de acuerdo a la tabla 4.2.2.1a(c) del ACI 347R-14.

*Ecuación 01. Máxima presión lateral del Concreto*

$$C_{CP\max} = C_c C_w \left[ 7.2 + \frac{1156}{T+17.8} + \frac{244R}{T+17.8} \right]$$

Una vez que la información ha seguido el proceso descrito en la Figura 05, ambos parámetros de salida son almacenados como se explicó en las Figuras 03 y 04, tanto en el archivo de Excel automáticamente generado, como en el modelo de Autodesk Revit, en la celda de comentarios para cada Smart Object. En resumen, el único parámetro de entrada que requiere el algoritmo para poder funcionar es un modelo existente en Autodesk Revit que incluya una estructura marco de elementos de concreto reforzado. La forma como el algoritmo reporta sus parámetros de salida es mediante el informe generado en el documento de Excel, previamente mencionado, y en la pestaña de propiedades de cada uno de los elementos en el modelo en Autodesk Revit.

# Caso de Estudio

El modelo piloto sobre el cual el algoritmo fue originalmente implementado se trata de un edificio tipo marco de seis niveles con elementos estructurales de concreto reforzado y divisiones internas livianas, desarrollado y diseñado por el autor en Autodesk Revit 2018 con licencia de estudiante. Dicho modelo no representa la complejidad de un modelo BIM, pues para los efectos prácticos del presente proyecto de investigación, se busca el desarrollo de un algoritmo que pueda extraer la información desde este tipo de modelos y no la extensión de un modelo BIM como tal. Con el fin de validar la aplicación, este algoritmo fue utilizado en un modelo BIM de validación facilitado por el profesor tutor, representa una edificación real y

con varios de los vectores de información propios de este tipo de modelos.

La información fue extraída y procesada a través de un algoritmo desarrollado en la plataforma de programación visual Dynamo para Revit, auxiliada por subscripsts de Python 2.7. Tanto el modelo piloto desarrollado para la implementación del algoritmo, como este último, fueron desarrollados y ejecutados para su evaluación en una unidad Lenovo G50, con un procesador Intel i7-4510U @ 2.60 GHz, 8.00 GB de memoria de acceso aleatorio, Intel HD Graphics 4000 y Windows 10 como Sistema Operativo, sin embargo, indistintamente de la unidad en la cual se ejecute el algoritmo, los resultados deben ser congruentes.

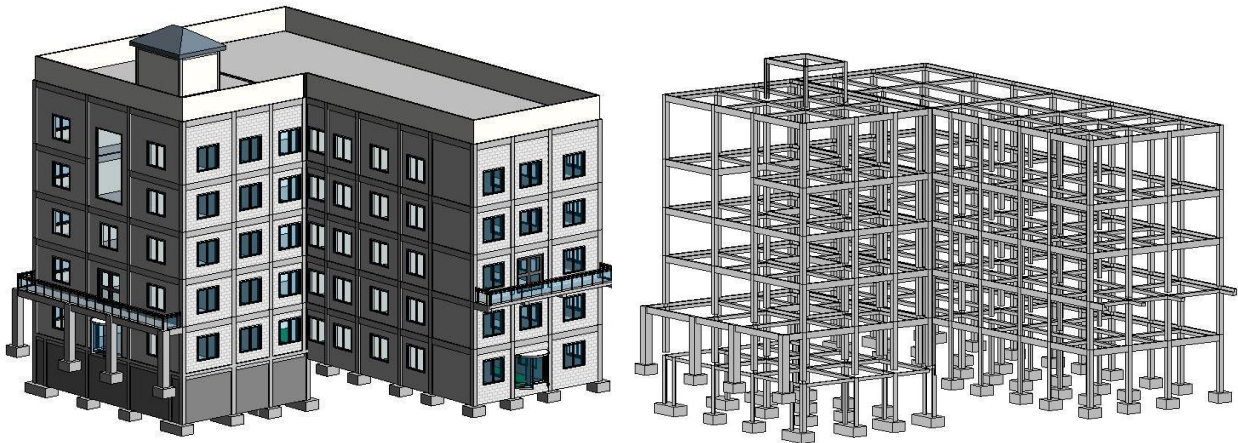


Figura 06. Modelo Piloto (izquierda) y Marco Estructural de Concreto Reforzado del Modelo Piloto (derecha)

Tal y como se mencionó, el algoritmo fue concebido originalmente para determinar la combinación más óptima de paneles modulares para columnas y vigas estructurales de concreto reforzado, así como para zapatas del mismo material, sin embargo, durante el proceso continuo de control cruzado realizado en el proyecto para ambos, el modelo piloto y el de

validación, se notó que las zapatas de cimentación o fundaciones en concreto reforzado pueden ser ingresadas a los modelos de Revit en muchas maneras distintas, contemplando el diseño personalizado de estos elementos es una de esas formas de ingreso; razón por la cual se decidió no cubrir dentro del algoritmo la selección para los elementos de encofrado para

fundaciones dentro del área de enfoque de este algoritmo en particular y dejar el espacio abierto para esto en trabajos futuros. En otras palabras,

este caso de estudio se limita a vigas coladas en sitio y columnas estructurales de concreto reforzado.

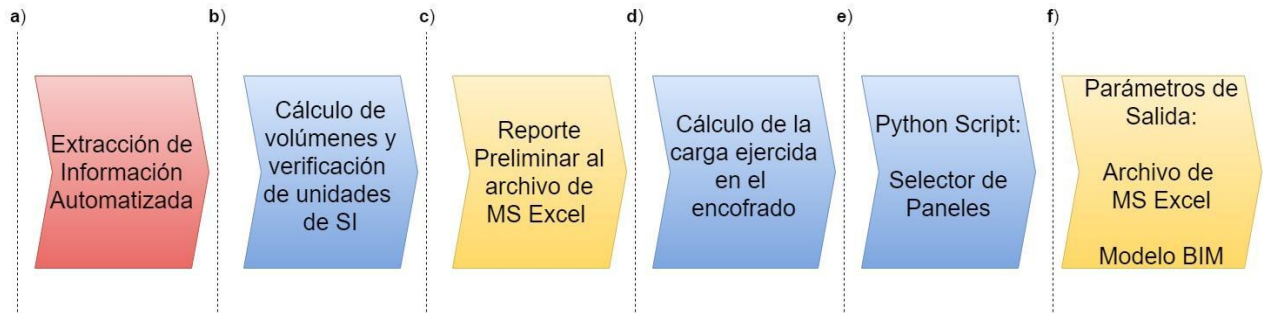


Figura 07. Fases Generales de funcionamiento del Algoritmo

La aplicación del algoritmo a un modelo BIM o un modelo estructural de Autodesk Revit desarrollado por una tercera parte es independiente a la magnitud de este en términos del número de elementos estructurales, niveles de la edificación, área y cualquier otro parámetro cuantitativo que pueda variar en cada modelo. Como se mencionó anteriormente, el algoritmo se desarrolló sobre el modelo piloto y fue ejecutado constantemente para su control sobre el mismo. Ahora bien, con respecto del modelo de validación, este fue aportado por el profesor supervisor, Dr. Ehsan Rezazadeh Azar, representa una estructura existente dentro de la

misma Universidad (Lakehead University) y desarrollado por una tercera parte. Este comenzó a ser utilizado para la ejecución en el algoritmo a partir de la fase d) de la Figura 07, una vez fue demostrada la capacidad del algoritmo para extraer la información dimensional a partir del modelo piloto. El mencionado modelo de validación consiste en una estructura de cinco niveles, compuesta de marcos estructurales de concreto reforzado y con divisiones internas en liviano, tal y como se muestra en la Figura 08. El área de ninguno de ambos modelos es relevante para el presente caso de estudio.

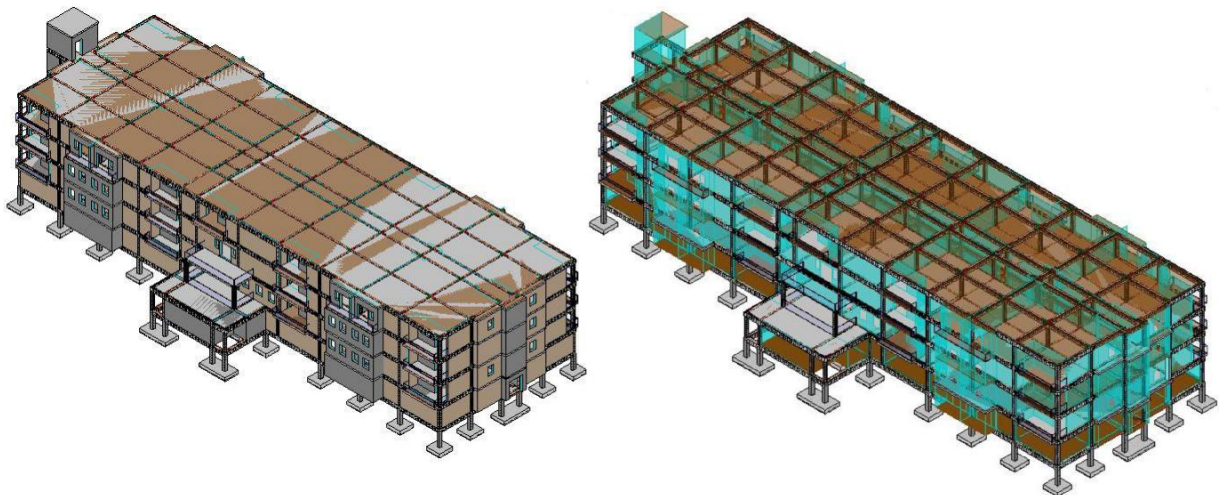


Figura 08. Modelo de Validación (izquierda) y Marco Estructural de Concreto Reforzado del modelo de Validación (derecha)

# Fases de operación del algoritmo

El algoritmo empleado para la aplicación descrita en el presente proyecto fue desarrollado en Dynamo, como plataforma de programación visual. Dynamo es un software de tipo Open Source que permite al usuario desarrollar “scripts” para automatizar o programar procesos variados en complejidad a manera de diagramas de flujo, donde cada uno de los nodos representa una función, una variable o un ciclo lógico de programación. Esta plataforma funciona sobre el lenguaje de programación Python 2.7, por lo que existe la posibilidad de insertar nodos con “scripts” personalizados en caso de no encontrar ninguno que cumpla con la función deseada. Para este proyecto en particular, dos de los procesos más importantes fueron programados en Python por el autor y adaptados para el algoritmo en nodos de Dynamo.

## Fase a) Extracción automatizada de la información

Tal y como se explicó en el párrafo anterior, Dynamo es una plataforma de programación visual, por lo que sus scripts pueden diferir en muchas maneras de aquellos desarrollados en otros lenguajes de programación. La diferencia fundamental es que los scripts de Dynamo trabajan a través de los mencionados nodos y se concatenan cual diagramas de flujo en un proceso secuencial; sin embargo, es importante señalar que, al estar basados en Python 2.7, un nodo, en su explicación más simplificada, no es más que un pequeño script con valores de entrada y al menos un valor de salida. Ahora bien, al extender la comprensión del funcionamiento de este tipo de scripts, es de esperarse que para poder extraer automáticamente la información de los modelos de Autodesk Revit, el algoritmo trabaje a través de distintos nodos, siendo *Categories* y *All\_Elements\_of\_Category*, ambos de las bibliotecas incluidas en Dynamo, los primeros utilizados con el fin de permitir al algoritmo extraer el vector que contiene la categoría del elemento y en función de esta determinar si debe ser procesada como una viga o una columna estructural de concreto reforzado.

Una vez esta información es analizada, la secuencia del algoritmo se divide en dos caminos, donde uno de sus ramales extrae las variables tipo *Integer* de sus elementos dimensionales (ancho, alto y largo), y el otro se encarga de procesar las variables tipo *String* de los elementos, donde se incluye información tal como el índice y la ubicación de los elementos de una familia determinada. Esta tarea se logra a través de los nodos *Element.GetParameterByName* y *Parameter.ParameterByName* respectivamente. En el caso de las vigas estructurales, esta fase del diagrama también se encarga de descartar la información extraída de los elementos prefabricados de concreto, los cuales normalmente son utilizados a manera de viguetas o estructuras en la construcción de entornos. Esto es posible gracias a la utilización de *Element.FilterByName* para identificar los elementos según su familia.

Además, es importante entender que los nodos de Dynamo leen la información directamente desde el modelo de Revit, por lo que cualquier modificación, adición o eliminación de parámetros desde el modelo o desde el algoritmo va a afectar ambas plataformas. La forma de referirse al almacenaje de cualquier variable en el algoritmo es sensible a mayúsculas, por lo que, para la extracción de la información contenida bajo cualquier nombre desde el modelo, esta consideración fue incluida también.

## Fase b) Estimación de volúmenes y verificación de unidades SI

Una vez almacenadas las dimensiones físicas del modelo en variables temporales del algoritmo, la naturaleza de las unidades puede leerse desde el modelo de Autodesk Revit. Para leer dicha información, el algoritmo requiere de “scripts” adicionales desarrollados en Python, pues no existe ningún nodo en las bibliotecas incluidas por defecto en Dynamo que permita extraer esta información. En este caso, el “script” desarrollado por el autor está detallado en la Figura 09, donde el primer paso corresponde a importar las funciones para Autodesk dentro del lenguaje de programación desde la referencia *RevitAPI*. Una vez importadas las funciones necesarias, se

extrae en la línea 16 el tipo de unidades utilizadas como un objeto, por lo que es necesario convertir esta información a *String* con el nodo *String\_from\_Object* para poder ejecutar el nodo booleano *String.Contains* con el objetivo de determinar si las unidades del modelo corresponden a milímetros y mover esta información a un ciclo condicional para determinar si es necesario convertir las unidades a metros, o si estas ya están en dicha unidad.

Una vez que las dimensiones están en metros, según el sistema de unidades SI, la estimación del concreto por colocar se realiza mediante la multiplicación de las dimensiones del elemento y la inclusión de un porcentaje de desperdicio. Este parámetro es considerado de importancia por el autor y por lo tanto, es incluido en el resumen presentado en el archivo generado automáticamente en MS Excel.

```
1 import clr
2 clr.AddReference('RevitAPI')
3 import Autodesk
4 from Autodesk.Revit.DB import *
5
6 clr.AddReference('RevitServices')
7 import RevitServices
8 from RevitServices.Persistence import DocumentManager
9 from RevitServices.Transactions import TransactionManager
10 from System.Collections.Generic import *
11
12 import System
13
14 doc = DocumentManager.Instance.CurrentDBDocument
15
16 result = Document.GetUnits(doc).GetFormatOptions
17 (UnitType.UT_Length).DisplayUnits
18 OUT = result
```

Figura 08. Script para la determinación de unidades utilizadas en el modelo analizado

## Fase c) Reporte preliminar al documento de Excel

Una vez que las unidades del proyecto han sido corregidas para su procesamiento en el algoritmo y que el volumen de concreto ha sido determinado, el algoritmo automáticamente reporta la información extraída del modelo y estimada matemáticamente. La información mostrada en el archivo de MS Excel generado automáticamente se organizará acorde con la Tabla 01, con los parámetros de identificación (Índice de columnas y nivel de base o referencia para ambas categorías), parámetros de referencia (Familia del elemento), parámetros dimensionales (ancho, alto y largo en metros), parámetros volumétricos y cuantitativos (volumen de concreto en metros cúbicos). Estos parámetros son entonces procesados

automáticamente por el algoritmo acorde con su naturaleza y son también impresos en el documento generado de MS Excel, previo a la selección de la combinación óptima de los paneles de encofrado.

Con respecto del documento de MS Excel en sí, el nodo *File\_Path*, incluido en las bibliotecas por defecto de Dynamo, es utilizado con el directorio del escritorio de la unidad Windows PC donde se esté utilizando, con el fin de generar un archivo automático denominado "Project Building Summary" en MS Excel para poder contener las tablas de archivos generados. La decisión de generar este archivo en el escritorio surge de la conveniencia y simplicidad de ubicarlo precisamente en este directorio, así como para evitar cualquier eventual error vinculado a la existencia de una carpeta de archivos determinada. Una vez generado el archivo, la información es importada a este

utilizando el nodo *Excel.WriteToFile*, con el que se asignó almacenar la información a partir de la celda B2 en una de las nuevas pestañas generadas llamada “Columns” o “Beams” según corresponda el caso a Columnas o Vigas respectivamente.

Es importante indicar en este punto que la información detallada en esta fase es almacenada en el archivo MS Excel en este punto del algoritmo con el fin de simplificar los ciclos de procesamiento y liberar memoria de

acceso aleatorio en la etapa final del algoritmo, ya que al seleccionar la combinación óptima de paneles se requiere cierto tiempo de procesamiento en función del tipo de modelo analizado. Una vez la información ha sido importada en el archivo generado, el algoritmo continúa trabajando en paralelo sin leer información desde dicho directorio; sin embargo, las variables temporales que generaron las tablas siguen funcionando desde Dynamo.

Tabla 01. Parámetros mostrados en el resumen plasmado en el archivo MS Excel

Parámetro	Tipo de Variable	Fuente
Column Tag ID	String	Grilla generada por el Usuario
Base/Reference Level	String	Modelo de Autodesk Revit
Family Name	String	Modelo de Autodesk Revit
Width [m]	Integer	Modelo de Autodesk Revit
Depth [m]	Integer	Modelo de Autodesk Revit
Height [m]	Integer	Modelo de Autodesk Revit
Concrete Volume [m <sup>3</sup> ]	Integer	Calculo del Algoritmo

## Fase d) Estimación de la carga ejercida en el encofrado

La carga del concreto fresco en las paredes del encofrado es estimada de acuerdo con la Ecuación 01, para el fondo (cara inferior) del elemento de concreto colado en sitio, de acuerdo al comportamiento descrito en la Figura 02 y considerando un coeficiente químico (Cc) de 1.00 para el tipo de cemento 1 sin adiciones de escoria de alto horno o ceniza volante, un coeficiente de peso unitario (Cw) de 1.00 de acuerdo con la tabla 4.2.2.1a(c) del ACI 347R-14, una temperatura (T) de 10° Celsius, siendo esta la mínima indicada para la colocación de concreto fresco; adicionalmente para este caso de estudio, por criterio del Dr. Ehsan R. Azar se utiliza una tasa de colocación para vigas de 1,0 m/h y de 1,5 m/h para columnas.

Con respecto de la aplicación de la ecuación en Dynamo para el algoritmo, esta es resuelta mediante un nodo *code\_block*,

definiendo cada uno de los parámetros mencionados en el párrafo anterior como variables temporales para este cálculo.

## Fase e) Script de Python: Selector de Paneles

A pesar de que el algoritmo automatiza la selección de los elementos modulares de encofrado, es importante señalar que el ingreso de los parámetros de distintos sistemas de encofrado debe ser realizado de manera manual, y puede ser fácilmente modificado por cualquier usuario únicamente abriendo el script desde Dynamo; esto con el fin último de permitir la inserción de sistemas personalizados o variados de formaleta sin limitar el sistema únicamente a los pocos incluidos en este caso de estudio. La Figura 10 muestra una captura de pantalla recortada del “script” de Dynamo, específicamente, la sección que se refiere a los



grupos de nodos que incluyen los parámetros básicos requeridos para su selección y extraídos desde las fichas técnicas de cada uno de los sistemas de encofrado modular utilizados.

Este caso de estudio cubre únicamente la aplicación de cuatro sistemas distintos, debido a su naturaleza académica y a la fácil obtención de las fichas y especificaciones técnicas de dichos sistemas en línea, sin embargo, en un caso práctico, se pueden ingresar cuántos sistemas de

encofrado sean necesarios desde la misma lógica de base de datos, tal y como se ejemplifica en la Figura 10 y como puede ser corroborado en el “script”. La adición de nueva información y bases de datos puede ser realizada por cualquier usuario con cualificaciones básicas en el uso de Dynamo, de manera que pueda entender la lógica de la programación tras las conexiones tras los nodos y reconectarlos según sea necesario.

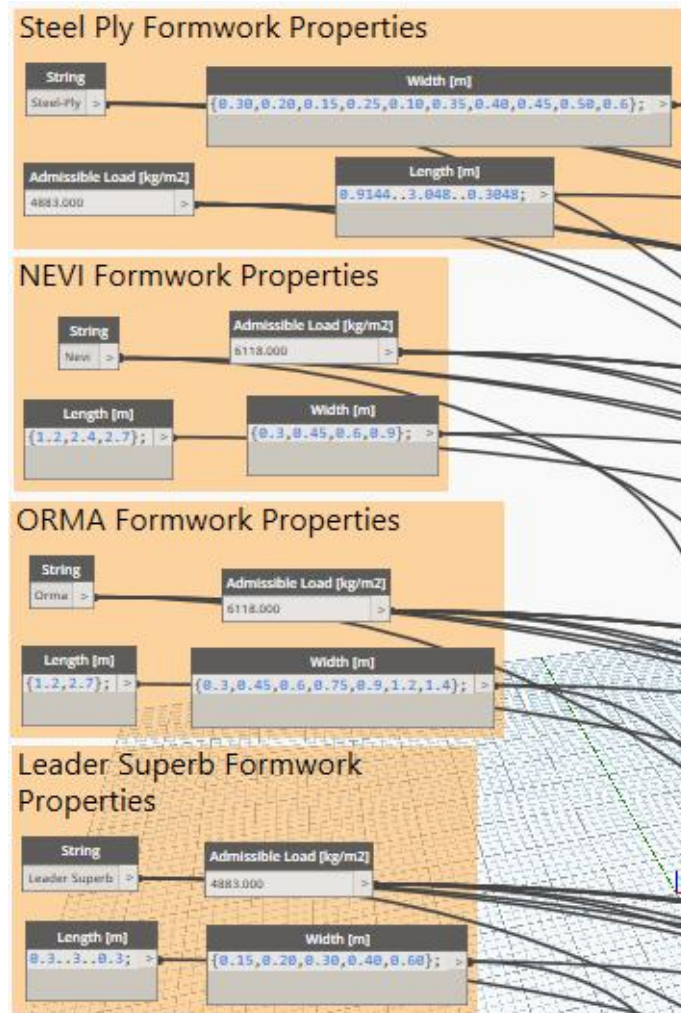


Figura 10. Recorte de captura de pantalla mostrando los sistemas de encofrado modular insertados como base de datos tal y como aparecen en el “script” de Dynamo.

La información requerida para evaluar un nuevo sistema de encofrado modular consiste en (1) un nodo de tipo *String* para el nombre comercial del sistema, (2) un nodo de tipo *Integer* para la carga admisible de los paneles de formaleta en unidades de  $\text{kg/m}^2$ , (3) un nodo *code\_block* que incluya una lista de variables

numéricas (*Integer*) que representen las dimensiones de ancho disponible para ese sistema de encofrado en metros y (4) un nodo *code\_block* que incluya una lista de variables numéricas (*Integer*) que representen las dimensiones de largo disponible para ese sistema de encofrado en metros.

```

1:import clr
2:clr.AddReference('Protostructure')
3:from Autodesk.DesignScript.Geometry import *
4:#The inputs to this node will be stored as a list in the IN variables.
5:datapointingnode = IN
6:
7:import itertools
8:class nnums(ObjEct):
9:
10:#get all combinations from list
11:def combinations(self, elements):
12:    allCombs = []
13:    for i in range (0, len(elements) + 1):
14:        for comb in itertools.combinations(elements, i):
15:            allCombs.append(comb)
16:    return allCombs[1:]
17:
18:#Remove greater than target or return if the same
19:def purge (self, elements, target):
20:    purged = []
21:    for i in range (0, len(elements)):
22:        if elements[i] >= (target-0.001):
23:            return [target-0.001]
24:        elif elements[i] < target:
25:            purged.append(elements[i])
26:    return purged
27:
28:#Add list subsets
29:def sums (self, elements):
30:    added = []
31:    for i in range (0, len(elements)):
32:        add = 0
33:        for j in range (0, len(elements[i])):
34:            add += elements[i][j]
35:        added.append(add)
36:    return added
37:
38:#get distances from elements to target
39:def distances (self, elements, target):
40:
41:    diffs = []
42:    for i in range (0, len(elements)):
43:        diffs.append(target - elements[i])
44:    return diffs
45:
46:#get the closest element index to zero
47:def closest (self, elements):
48:    positives = []
49:    for i in range (0, len (elements)):
50:        if elements[i] >= 0:
51:            positives.append(elements[i])
52:
53:    closetozero = min (positives)
54:    return elements.index ( closetozero )
55:
56:#return combination at index
57:def getCombElement ( self, index, combs):
58:    return combs[index]
59:
60:#return the closest dimension to a target
61:def getClosest (self, dims, target):
62:    purged = self.purge(dims, target)
63:    if isinstance(purged, float) or isinstance(purged, int):
64:        return [purged]
65:    combs = self.combinations(purged)
66:    added = self.sums(combs)
67:    diffs = self.distances(added, target)
68:    close = self.closest(diffs)
69:    comb = self.getCombElement(close, combs)
70:    return comb
71:
72:#return the closest dimension to a list of targets
73:def getClosestForAll ( self, dims, targets):
74:    closests = []
75:    for i in range (0, len(targets)):
76:        closests.append([targets[i], self.getClosest(dims, targets[i])])
77:    return closests
78:
79:
80:mynums = nnums()
81:allCombs = mynums.getClosestForAll(IN[0], IN[1] )
82:print allCombs
83:
84:#Assign your output to the OUT variable.
85:OUT = allCombs

```

Figura 11. Script para la selección óptima de paneles de encofrado modular

Las dimensiones disponibles para los paneles deben ser listas de variables numéricas completamente independientes, cuyas dimensiones son combinables entre sí, es decir, que cada panel del sistema puede ser encontrado en cualquiera de los anchos con cualquiera de los largos especificados.

El proceso de selección de paneles para los elementos de concreto por ser colados es realizado mediante un “script” personalizado de Python descrito en la Figura 11, desarrollado por el autor e insertado a través de un nodo *Python Script*, cuyos valores de entrada son las dimensiones objetivo del elemento de concreto reforzado a ser colado y las listas de dimensiones existentes en el sistema de encofrado en particular, de acuerdo con la fuente de la información procesada. El nodo con el script mencionado es utilizado en dos ocasiones por cada elemento estructural y sistema de encofrado seleccionado, en la primera ocasión para ancho y en la segunda para largo.

Cuando el elemento analizado se trata de una columna estructural, la dimensión de la longitud es considerada como la correspondiente con el eje vertical, por lo tanto, para este caso de estudio y para efectos de facilidad en el proceso constructivo, el autor considera 0.30 m adicionales para esta dimensión en la selección de su encofrado, es decir, habrá 0.30 m de encofrado sobresaliendo de la altura final del elemento de concreto colado para evitar derramamientos de concreto por rebase y pérdida del material. El algoritmo desarrollado para este caso de estudio cubre únicamente a las columnas de sección cuadrada. Para adecuar el algoritmo y su ejecución a columnas de sección rectangular o cualquier otro tipo de sección transversal se

requieren algunas modificaciones que serán discutidas en el apartado de trabajo futuro y limitaciones en el presente informe. Ahora bien, considerando el caso de las vigas estructurales, el presente caso de estudio considera secciones rectangulares y cuadradas para las vigas, por lo que provee al usuario con una combinación de paneles para la cara inferior del elemento y para las caras laterales considera 0.30 m adicionales para el proceso constructivo, como se explicó para el caso de las columnas. Esta medida es definida por el autor y puede ser modificada de ser necesario por cualquier usuario.

## Fase f) Parámetros de salida finales.

El paso final del algoritmo es reportar los resultados obtenidos en dos parámetros finales de salida, siendo uno de ellos el documento de MS Excel completo y el otro el modelo de Autodesk Revit con los vectores de información. Antes de reportar los resultados hacia cualquiera de los parámetros de salida, deben poseer un formato homologado, transpuesto y normalizado para ser comprendido por el usuario. En este punto, la información que viene del algoritmo es una serie de datos almacenados en distintos órdenes de listas de variables en Dynamo, compuestas por los resultados del script de Python del punto anterior, ya que este únicamente procesa variables numéricas y no asocia los resultados con el elemento de concreto en particular. Dichas listas están anidadas en tres niveles, cuya jerarquía está explicada en la Figura 12.

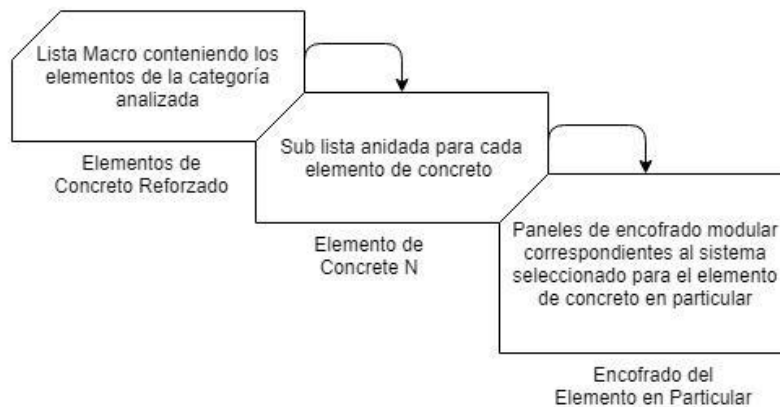


Figura 12. Jerarquía de las listas anidadas en el formato de parámetros de salida del “script” en la Figura 11.

Una vez el resultado del “script” de Python está listo, el nodo *List.Flatten* es utilizado para reducir los grados de anidamiento de la lista a únicamente dos niveles: (1) los elementos de concreto y (2) los paneles de encofrado modular seleccionados; el vínculo entre los elementos y sus respectivos paneles a partir de este punto depende completamente de la organización de la lista. Por la razón mencionada el comando *List.Sort* no es utilizado en ningún “script” a partir de este punto. La nueva lista es entonces reducida a una lista ordenada de la selección de paneles para cada cara agrupados bajo el índice del elemento de concreto en particular y estos parámetros son linealizados una vez más mediante el nodo *List.TakeEveryNthItem*, con la dimensión objetivo para el elemento de concreto como titular en esta lista.

Una vez el arreglo deja de ser una lista anidada y se convierte en una lista unidimensional (una matriz simple) de *Integers*, el contenido de cada uno de los grupos en los índices es convertido a un *String* separado por comas con los nodos *List.Combine* y *String.Join* respectivamente donde las letras “w” y “l” se añaden para cada uno de los casos, dependiendo si la lista de dimensiones del elemento de encofrado corresponde a ancho o largo respectivamente. Este proceso da como resultado una lista de dimensiones que deben ser combinadas para responder a la necesidad de paneles para cada una de las caras de los elementos. La forma adecuada de leer las combinaciones de dimensiones para estos elementos se detalla en la Figura 13.

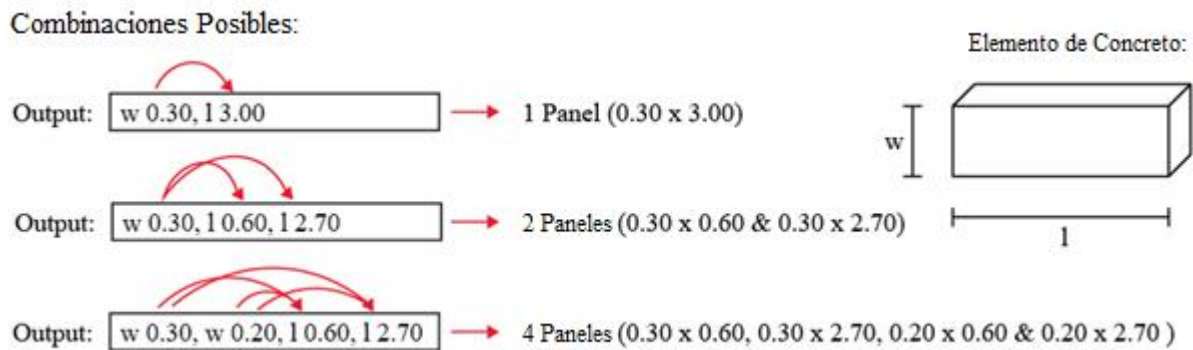


Figura 13. Diagrama de lectura para los parámetros de salida del Algoritmo.

Hasta este punto el único filtro entre los distintos sistemas de encofrado han sido las dimensiones físicas del elemento en particular y la carga admisible. Parámetros tales como costos y disponibilidad no han sido especificados hasta ahora. Para este caso de estudio el autor define como necesario que los parámetros de salida estén en términos de un único sistema de encofrado modular para todo el proyecto constructivo. Para proyectos de gran envergadura donde el presupuesto permite el uso de más de un sistema de encofrado las correcciones y adecuaciones son discutidas en el capítulo de trabajo futuro y limitaciones del presente informe.

Ahora bien, con respecto de la selección entre distintos sistemas de encofrado, y considerando que en el futuro el algoritmo puede ser modificado respondiendo a las necesidades locales y posibilidades de cada proyecto según

su lugar de emplazamiento en la construcción, todo el proceso de selección final se hace mediante un ciclo *If* con varios valores de entrada, de los cuales la ordenanza genérica es: (1) La lista de combinaciones óptimas de paneles para cada elemento en cada sistema de encofrado modular, (2) la carga admisible por las paredes de los elementos para cada sistema, (3) la carga objetivo ejercida en las paredes del encofrado, previamente determinada.

En términos de preferencia y considerando la naturaleza relativa de esta selección, basada principalmente en la conveniencia y el criterio del Ingeniero residente o el contratista, el ciclo es tratado como un condicional *If\_then* en un nodo *Code\_Block* donde la primera iteración corresponde a la realizada con el sistema más conveniente, la

segunda con el segundo sistema más conveniente y así sucesivamente.

El reporte de los vectores de información obtenidos de ambos parámetros de salida hacia sus respectivos archivos destino es realizado mediante nodos separados; en el caso del reporte al archivo MS Excel, se utiliza el nodo *Excel.WriteToFile* siguiendo la misma lógica de la fase c. En el caso de la información que va a ser integrada en el modelo de Autodesk Revit, el autor seleccionó la celda “Comments” de cada uno de los elementos de concreto para insertar en ella la optimización sugerida de los paneles de encofrado, ya que esta celda no afecta el modelo en particular y permite la rápida lectura de información en él. Esta acción se realiza mediante el nodo *Element.SetParameterByName*, el cual recibe los valores de entrada y los organiza en el parámetro especificado dentro del modelo, que, como se mencionó, corresponde a “Comments”.

# Resultados

Los resultados para el presente proyecto de investigación son extraídos del modelo piloto y el modelo de validación con el fin de analizar cualquier potencial diferencia entre los parámetros de salida obtenidos al recibir la información del modelo diseñado por el autor y el modelo aportado por la tercera parte.

Como se mencionó anteriormente, los parámetros de salida van a ser reportados tanto en el archivo generado de MS Excel como en el modelo de Revit de donde originalmente se

extrajo la información del proyecto. En el caso del archivo generado, los resultados serán mostrados en distintas pestañas para cada categoría, una para columnas y una para vigas, en cada caso indicando el índice particular del elemento analizado. Esta información será organizada como se muestra en la Figura 14 dentro del archivo MS Excel, y los vectores de información referentes a la óptima combinación de paneles de encofrado modular deberá ser leída tal y como se indica en la Figura 13.

Column ID	Column Base Level	Family Name	Column Width [m]	Column Depth [m]	Column Height [m]	Concrete Volume [m3]	Formwork System: Leader Superb
T-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U(-648)-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U(-648)-3	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U-6	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U-9	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U(-648)-11	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
U(-648)-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
T-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
S-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
R-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
Q-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
M-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
J-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
H-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
G-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
F-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
E-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
C-12	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
C-11	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
A-10	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
A-5	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
C-3	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
C-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
E-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
F-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
G-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
H-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
K-0	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
O-0	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
Q-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
R-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70
S-1	Base Level : Ground	300 x 300mm	0.3	0.3	3.000103381	0.270009304	w 0.30, l 0.60, l 2.70

Figura 14. Recorte de captura de pantalla para el reporte de columnas de concreto reforzado en el documento de MS Excel generado automáticamente "Project Building Summary"

Los resultados mostrados en la Figura 14 corresponden a un recorte de una captura de pantalla desde el MS Excel que contiene los resultados obtenidos al correr el algoritmo en el modelo de validación. Esta imagen se incluye únicamente con el fin de ilustrar la manera en la que el algoritmo presenta los resultados finales

en el archivo generado. Cada una de las etiquetas mostradas en la columna "Column ID" corresponde a la información extraída como un String de la grilla generada por el usuario en el modelo en particular. Esta consideración parte del hecho de que es una práctica comúnmente extendida el utilizar grillas en la creación de

modelos en Revit; de no existir, el resultado para cada una de las celdas en esta columna será *null*. Ahora bien, con respecto de las columnas de “*Base Level*” y “*Family Name*”, ambos parámetros son extraídos desde la pestaña de propiedades en Autodesk Revit, así como los parámetros dimensionales.

El archivo generado en MS Excel no es más que una forma práctica de resumir y tabular los resultados extraídos del modelo analizado, por lo tanto, luego de que el algoritmo ha sido ejecutado, el archivo mencionado puede ser utilizado y modificado a libertad por el usuario, ya que este no tiene vínculo bidireccional con el modelo en Revit, por lo tanto, la modificación de la información allí contenida no afectará de ninguna manera el modelo. Dentro de los parámetros incluidos en este archivo se encuentran volúmenes de concreto colado en sitio, cantidades de paneles y otros tipos de información que pueden ser utilizados por el Ingeniero en sitio, según considere oportuno.

En el caso de los resultados reportados para las vigas estructurales, existe una pequeña

diferencia, ya que deben dedicarse dos columnas de la tabla por reportar los paneles seleccionados, una para la combinación óptima de paneles para el fondo del elemento por colar y una para la combinación para las caras laterales de dicho elemento. Es importante indicar en este punto que los elementos de encofrado recomendados para las caras laterales del elemento de concreto colado se especifican para cada cara y en un eventual conteo de paneles, esta especificación debe ser considerada dos veces. Los resultados obtenidos para las vigas de concreto se tabulan tal y como se indica en la Figura 15.

Finalmente, con respecto de la información almacenada como un *String* en la celda “*Comments*” para cada uno de los elementos, la visualización en la interfaz es como se muestra en las Figuras 16 y 17, donde la pestaña de propiedades de Revit muestra la información deseada al seleccionar el elemento del cual se desee consultar información.

Beam Reference Level	Family Name	Beam Width [m]	Beam Height [m]	Beam Length [m]	Concrete Volume [m3]	Bottom Leader Superb	Sides Leader Superb
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	4.591	0.9182	w 0.40, l 1.50, l 3.00	w 0.40, w 0.40, l 1.50, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	3.639	0.7278	w 0.40, l 0.60, l 3.00	w 0.40, w 0.40, l 0.60, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	2.031474108	0.406294822	w 0.40, l 1.80	w 0.40, w 0.40, l 1.80
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	2.031474108	0.406294822	w 0.40, l 1.80	w 0.40, w 0.40, l 1.80
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	0.453	0.0906	w 0.40, l 0.30	w 0.40, w 0.40, l 0.30
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	0.953	0.1906	w 0.40, l 0.90	w 0.40, w 0.40, l 0.90
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	0.961	0.1922	w 0.40, l 0.90	w 0.40, w 0.40, l 0.90
Reference Level : Level 2	400x500mm (reg)	0.4	0.5	0.648	0.1296	w 0.40, l 0.60	w 0.40, w 0.40, l 0.60
Reference Level : Level 2	400x500mm (reg)	0.4	0.5	0.961	0.1922	w 0.40, l 0.90	w 0.40, w 0.40, l 0.90
Reference Level : Level 2	400x500mm (reg)	0.4	0.5	0.452	0.0904	w 0.40, l 0.30	w 0.40, w 0.40, l 0.30
Reference Level : Level 3	400x500mm (reg)	0.4	0.5	0.961	0.1922	w 0.40, l 0.90	w 0.40, w 0.40, l 0.90
Reference Level : Level 3	400x500mm (reg)	0.4	0.5	0.458	0.0916	w 0.40, l 0.30	w 0.40, w 0.40, l 0.30
Reference Level : Level 3	400x500mm (reg)	0.4	0.5	0.453	0.0906	w 0.40, l 0.30	w 0.40, w 0.40, l 0.30
Reference Level : Roof	400x500mm (reg)	0.4	0.5	0.470950481	0.094190096	w 0.40, l 0.30	w 0.40, w 0.40, l 0.30
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	6.239373212	1.247874642	w 0.40, l 3.00, l 3.00	w 0.40, w 0.40, l 3.00, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	8.229	1.6458	w 0.40, l 3.00, l 3.00, l 2.1	w 0.40, w 0.40, l 3.00, l 3.00, l 2.10
Reference Level : Roof	400x500mm (reg)	0.4	0.5	0.453	0.0906	w 0.40, l 0.30	w 0.40, w 0.40, l 0.30
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	6.239373212	1.247874642	w 0.40, l 3.00, l 3.00	w 0.40, w 0.40, l 3.00, l 3.00
Reference Level : Level 2	400x500mm (reg)	0.4	0.5	1.787	0.3574	w 0.40, l 1.80	w 0.40, w 0.40, l 1.80
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	6.239373212	1.247874642	w 0.40, l 3.00, l 3.00	w 0.40, w 0.40, l 3.00, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	4.4415	0.8883	w 0.40, l 1.50, l 3.00	w 0.40, w 0.40, l 1.50, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	4.635	0.927	w 0.40, l 1.50, l 3.00	w 0.40, w 0.40, l 1.50, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	0.648	0.1296	w 0.40, l 0.60	w 0.40, w 0.40, l 0.60
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	2.551	0.5102	w 0.40, l 2.40	w 0.40, w 0.40, l 2.40
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	0.648	0.1296	w 0.40, l 0.60	w 0.40, w 0.40, l 0.60
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	4.595	0.919	w 0.40, l 1.50, l 3.00	w 0.40, w 0.40, l 1.50, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	4.595	0.919	w 0.40, l 1.50, l 3.00	w 0.40, w 0.40, l 1.50, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	5.965	1.193	w 0.40, l 3.00, l 3.00	w 0.40, w 0.40, l 3.00, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	4.595	0.919	w 0.40, l 1.50, l 3.00	w 0.40, w 0.40, l 1.50, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	4.595	0.919	w 0.40, l 1.50, l 3.00	w 0.40, w 0.40, l 1.50, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	4.595	0.919	w 0.40, l 1.50, l 3.00	w 0.40, w 0.40, l 1.50, l 3.00
Reference Level : Level 1	400x500mm (reg)	0.4	0.5	5.965	1.193	w 0.40, l 3.00, l 3.00	w 0.40, w 0.40, l 3.00, l 3.00

Figura 15. Recorte de captura de pantalla para el reporte de vigas de concreto reforzado en el documento de MS Excel generado automáticamente “Project Building Summary”

Los resultados mostrados en ambos parámetros de salida representan la recomendación del algoritmo desde el sistema de revisión de reglas para la selección del elemento.

Sin embargo, debe siempre ser revisado por un ingeniero competente para determinar si la combinación sugerida representa los mejores intereses del proyecto. También es necesario

señalar que el proceso de selección puede realizarse a partir de cuántos sistemas de encofrado modular desee incluir el ingeniero en el sistema, siguiendo el mismo diagrama de los nodos programados para cada uno de los ya utilizados.

Finalmente, la razón por la cual se decidió mostrar los resultados en cada uno de los elementos analizados es porque se consideró que esta función puede facilitar la determinación de los elementos de formaleta en cada una de las instancias, dependiendo del índice particular del elemento y las necesidades del proyecto en sitio.

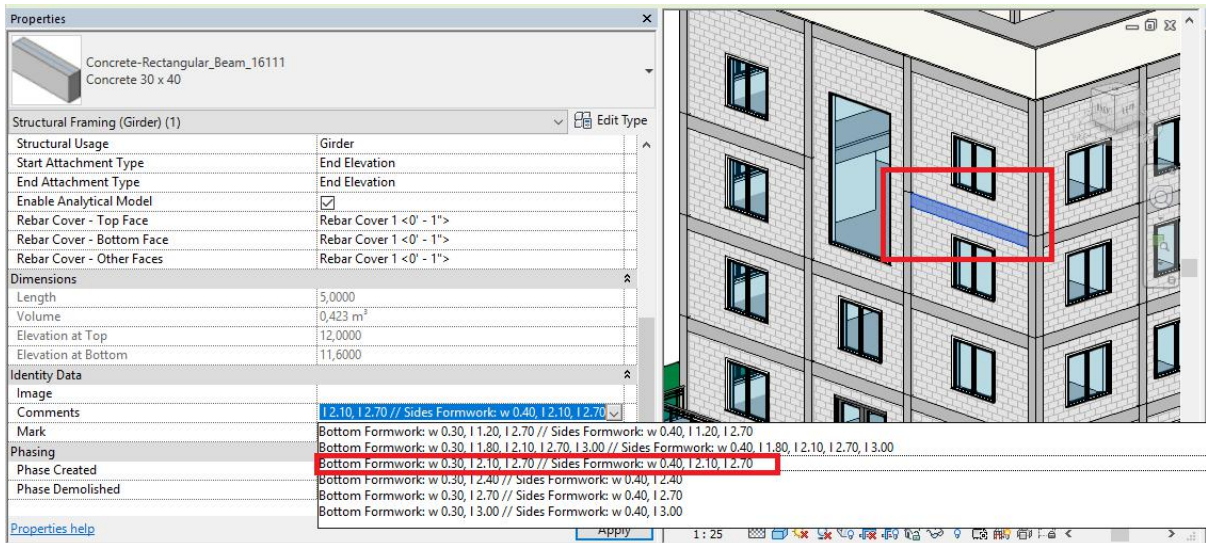


Figura 16. Formato visual de entrega de resultados para una viga de concreto reforzado en el modelo piloto

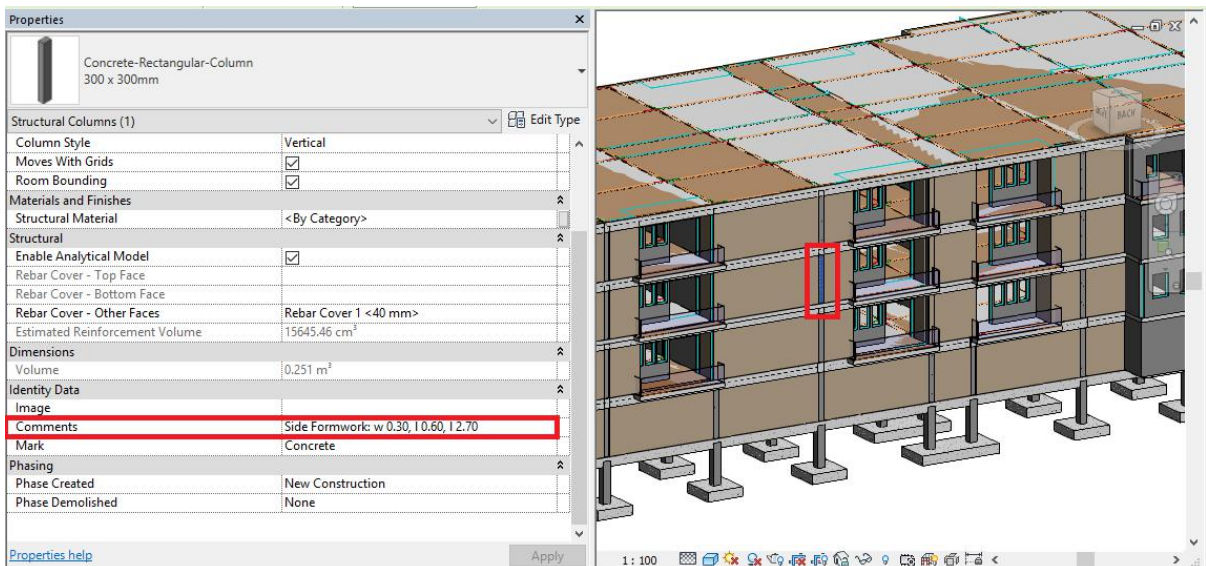


Figura 17. Formato visual de entrega de resultados para una columna de concreto reforzado en el modelo de validación



# Análisis de los resultados

La metodología convencional de diseño y selección de elementos de encofrado modular en proyectos necesariamente involucra la inversión de tiempo por parte del Ingeniero residente o el contratista en distintas etapas del proceso, tales como la extracción de información desde los planos del proyecto o modelos, el análisis de la información y la discriminación de información relevante, así como el proceso de toma de decisión ejecutiva con respecto de la selección de paneles. Como se indicó anteriormente, el objetivo principal del proyecto fue el desarrollo de una herramienta para auxiliar al responsable del proceso, representando un ahorro en tiempo y por ende, costos finales del proyecto.

De los resultados mostrados en el capítulo anterior puede deducirse que el proceso de extracción de la información contenida en los modelos es exitoso tanto para variables de tipo *Integer* como de tipo *String* y que ambas pueden ser almacenadas, según su naturaleza, por el algoritmo. En los casos en que las variables sean de naturaleza numérica, pueden ser utilizadas para estimaciones, comparaciones y otros procedimientos matemáticos. La combinación de ambos tipos de variables y el procesamiento de los valores contenidos en ellas a través de los scripts de Python descritos es también evidencia del éxito en dicho proceso de extracción. Todos los resultados son reportados en unidades del SI y los modelos desarrollados utilizando el sistema imperial de unidades no podrán ser interpretados por el algoritmo sin las adecuaciones correspondientes; este tema será discutido en el capítulo de trabajo futuro y limitaciones del presente informe.

La información extraída del modelo permite un marco de opciones claras para el proceso de toma de decisiones por parte de los responsables directos. Para este caso de estudio, el algoritmo involucra razonamiento cuantitativo y matemático simple tras el procesamiento de los

datos, sin embargo, eso no significa que la información aquí analizada no pueda ser procesada por lógicas matemáticas más complejas, tales como aquellas involucrando algoritmos discretos (L. Hou et al., 2017). El objetivo principal de exportar la información a un formato generalmente utilizado y de uso extendido como un archivo de MS Excel (.xls) es precisamente abordar uno de los problemas principales encontrados en el uso de Building Information Modelling desde sus etapas tempranas, la interoperabilidad. En otras palabras, hacer que la información esté disponible para todos los usuarios y no únicamente limitado a aquellos que cuenten con softwares específicos (R. Volk et al., 2014).

Durante el proceso de ejecución del algoritmo, varias evaluaciones fueron realizadas con el fin de identificar posibles errores y establecer el patrón en el que estos se desarrollan en paralelo al proyecto; lo que en inglés se denomina "*troubleshooting framing*". Los resultados mostrados en el capítulo anterior corresponden a la última iteración de una serie, donde algunas de estas terminaron en errores de ejecución o información incompleta. Ahora bien, a pesar de que el algoritmo es capaz de extraer exitosamente la información deseada desde el modelo, procesarla y reportar los resultados en los parámetros de salida, debido a la naturaleza de la plataforma en la que se desarrolló, el algoritmo mostrará advertencias cada vez que sea ejecutado, para cada uno de los sistemas de encofrado no seleccionados. Este tipo de problema en la codificación del "script" puede ser fácilmente resuelto en el futuro por nuevos desarrolladores más experimentados.

El autor en este caso en particular, define la interface y la operatividad del algoritmo desarrollado como incipiente, sin embargo, esta no es una razón para rechazar el sistema, ya que la lógica tras el "script" sienta las bases para

futuras investigaciones, partiendo del hecho de que Dynamo tiene, efectivamente, la capacidad de extraer información de cualquier modelo deseado; en otras palabras, el algoritmo muestra que información relevante puede ser extraída y procesada de manera automática por Dynamo para auxiliar a los involucrados en el proceso de

toma de decisiones con respecto de los elementos de encofrado modular, sin embargo, se indica que en futuras investigaciones y desarrollo por parte de desarrolladores más capacitados, resultará necesario desarrollar una solución de interfaz más amigable e integral.

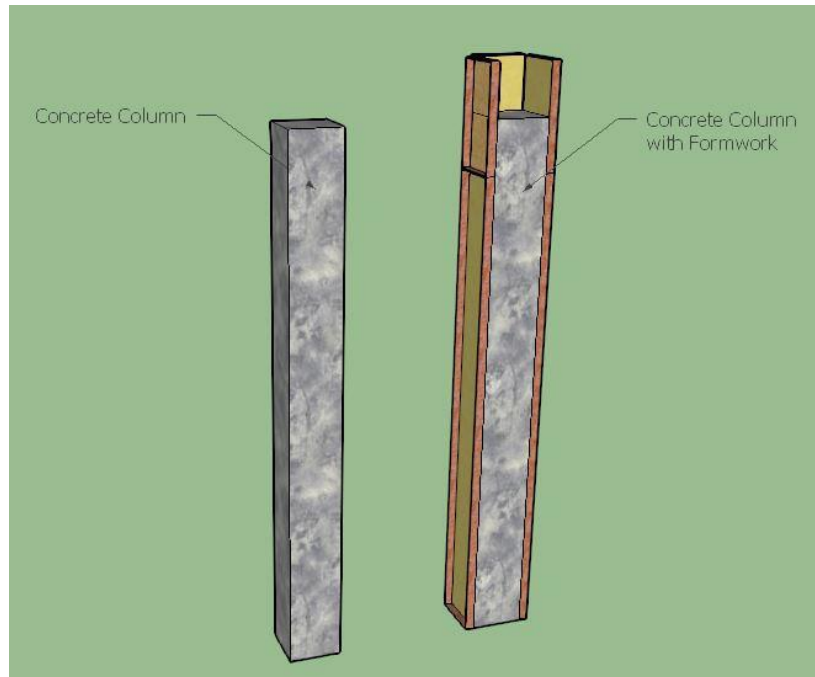


Figura 18. Columna de concreto reforzado aislada representada con y sin la selección final de elementos de encofrado modular.

Los resultados obtenidos mostrados en cada uno de los elementos del modelo analizado son *Strings* de información que representan la selección final de paneles de encofrado modular para cada una de las caras correspondientes del elemento seleccionado y deben ser leídos como se indicó en la Figura 13 para obtener el arreglo adecuado de formaleta para cada elemento. Por ejemplo, cuando los elementos de las Figuras 16 y 17 son leídos y se obtiene la combinación correspondiente para cada una de las caras, de logran configuraciones como las mostradas en las Figuras 18 y 19 respectivamente. Es importante entender que para ambas figuras la cara frontal no se muestra cubierta con paneles para permitir ver la diferencia del nivel entre el encofrado y el elemento de concreto reforzado a

su altura final, una vez terminado el proceso de colado en sitio.

Ambas visualizaciones 3D son la interpretación deseada de los parámetros de salida dadas como *String* desde el algoritmo. Para este caso de estudio en particular, el algoritmo no es capaz de desarrollar una visualización tridimensional de los paneles seleccionados y ambas imágenes fueron desarrollados por el autor en un software de modelaje de una tercera parte. La aplicabilidad de la información extraída no es delimitada por el autor, ya que otros desarrolladores con mayor conocimiento de la sintaxis de Dynamo pueden ser capaces de importar modelos en su formato IFC hacia Dynamo con el fin de representar gráficamente los resultados.

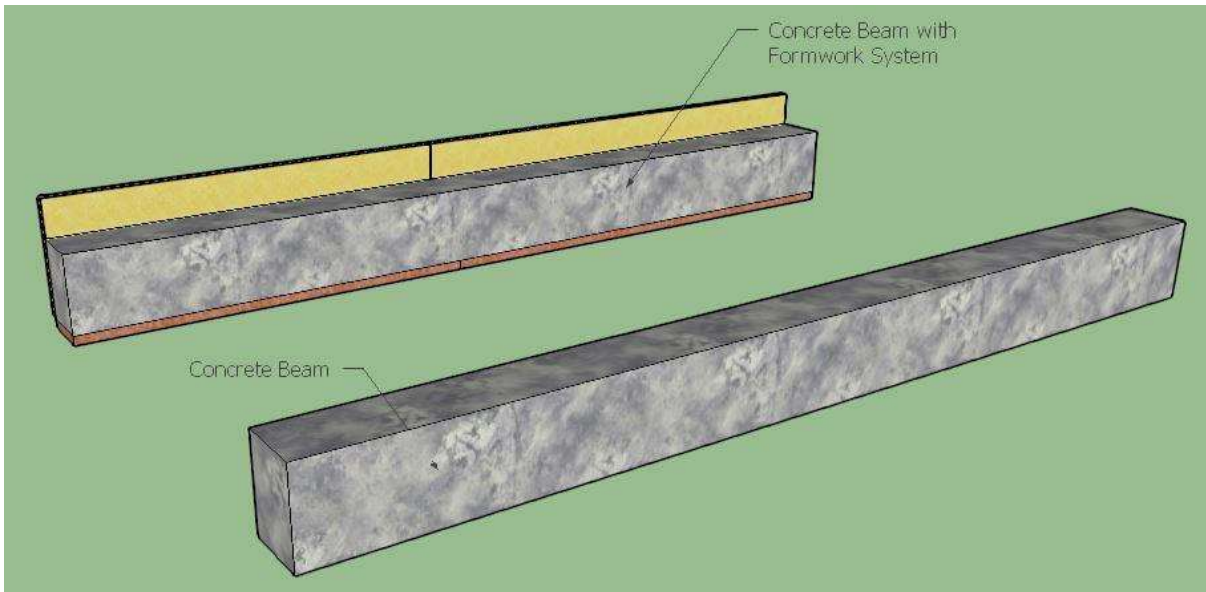


Figura 19. Viga de concreto reforzado aislada representada con y sin la selección final de elementos de encofrado modular.

# Trabajo futuro y limitaciones

El autor manifiesta por este medio que el algoritmo desarrollado en su totalidad por su persona es un enfoque y acercamiento incipiente hacia la extracción y análisis automatizado de información que puede ser logrado mediante la plataforma de programación visual Dynamo. Cualquier usuario o desarrollador que tenga la necesidad de modificar el "script" existente para adaptarse a sus necesidades es libre e invitado a hacerlo.

Es importante entender que el presente informe no pretende más que demostrar la aplicabilidad del software Dynamo en la extracción y procesamiento de datos desde modelos de Autodesk Revit y la manera en la que los valores de entrada de dichos procesos pueden ser almacenados y usados para diferentes propósitos. Es necesario también recordar que este informe representa únicamente un caso de estudio y describe un camino para la extracción de información, de todos los posibles y distintos caminos, en el cual, el único sistema de unidades considerado fue el sistema internacional SI. Por lo tanto, si algún usuario quisiera procesar un modelo desarrollado con unidades del sistema imperial, los ajustes matemáticos de equivalencia de unidades deben ser insertados al script de Dynamo, con el objetivo de convertir todas las unidades antes de ser procesadas, de manera que no afecten la lógica tras el algoritmo. Esta acción debe ser realizada para todos los cálculos y fórmulas matemáticas procesadas, así como en los nodos que contengan las listas de dimensiones de los elementos modulares de encofrado.

Si bien es cierto, los parámetros de salida del algoritmo son considerados como suficientes, el autor indica necesario profundizar la investigación hacia una forma de importar modelos tridimensionales creados en softwares de modelación, mediante la funcionalidad intrínseca de Revit para trabajar con modelos IFC referenciados a sus coordenadas, los cuales

serían insertados al modelo actual y ubicados junto al elemento analizado de manera aislada para poder dar un acercamiento gráfico al proceso de construcción, montaje y configuración del encofrado. (Kyungki K. et al., 2014).

La funcionalidad descrita puede ser lograda mediante la utilización de nodos que especifiquen las coordenadas de las cuatro esquinas de la cara de los modelos y las cuatro esquinas de la cara del elemento de concreto con el que estarán en contacto, de manera que pueda ubicarse en ese espacio tridimensional. Parte de los problemas esperados con este proceder corresponden a la especificación de las coordenadas de los elementos que requieren más de un panel consecutivo, los elementos que necesitan espacio de formaleta sobresaliente para el proceso de colado de concreto y la forma de isolar los elementos antes de la presentación gráfica de la configuración propuesta de encofrado. Si el proceso requiere un software de modelaje de una tercera parte, la existencia del archivo que contenga los modelos de los paneles requerirá índices que permitan a los parámetros de salida indexados llamar a dichos modelos; sin embargo, en este caso sería necesario acarrear los archivos en un folder en conjunto con los demás archivos necesarios para la ejecución de este. Por ende, de convertirse en una serie de archivos autoextraíbles, transformando el proyecto de un algoritmo a un programa.

El caso de estudio analizado cubre únicamente las columnas de concreto reforzado de sección transversal cuadrada y vigas de concreto reforzado coladas in situ de sección transversal rectangular o cuadrada. Para el análisis potencial de secciones transversales distintas de las estipuladas, sería necesario incluir ajustes y nodos nuevos que permitan interpretar la sección según el número de caras expuestas al contacto con el encofrado. Además, la estimación del volumen de concreto, así como la cantidad de columnas donde se concentrarían

los vectores de información debe ser ajustada también. El presente trabajo incluye únicamente los paneles necesarios para los elementos mencionados, sin entregar un conteo final de los mismos ni las llaves de conexión para ajuste del sistema de encofrado. Ahora bien, para la obtención de estas cantidades es precisamente que se entrega el archivo en MS Excel para que el usuario pueda editar y disponer libremente de la información aquí contenida.

El algoritmo no posee una interface y su ejecución se limita a la apertura desde Dynamo en el archivo de Autodesk Revit, lo que conlleva pasos innecesarios en la ejecución de este, por lo que se sugiere como trabajo futuro el desarrollo de una interfaz que permita al algoritmo ser ejecutado abriendo desde este directorio del modelo y no abriendo el modelo primero para ejecutar el algoritmo desde ahí.

# Conclusión

Este artículo sienta las bases para ingenieros con escaso a nulo conocimiento en lenguajes de programación, sirviéndose de la plataforma de programación visual Dynamo para lograr una tarea tan esencial como la extracción de datos de modelos de tecnología Building Information Modelling. El algoritmo desarrollado asiste al usuario proveyendo una lógica fácil de seguir enfocada directamente a la información que pueda ser requerida para procesos tan específicos como la selección de paneles modulares de encofrado para el colado de elementos de concreto reforzado, permitiendo un incremento en la productividad del proceso de toma de decisiones, dándole al usuario una visión de la combinación óptima de paneles para cada elemento. La información extraída en el proceso puede ser compartida mediante distintos parámetros de salida en distintas plataformas de software tal y como se mostró, tanto en forma de *Strings* de información contenidos en las propiedades de los elementos analizados del modelo BIM, como en valores tabulados en un resumen en MS Excel, mostrando la interoperabilidad que puede obtenerse con scripts sencillos en Dynamo.

La extracción de la información, así como su procesamiento, son ejecutados siguiendo una lógica de revisión de reglas previamente establecidas, en la cual los parámetros obtenidos del modelo son evaluados contra valores esperados a partir de criterios y fórmulas comúnmente utilizados. El algoritmo como tal, es, por tanto, un proceso iterativo en el que los valores son almacenados en constantes temporales que son frecuentemente evaluadas y cuyo parámetro de salida final puede ser un *String* conteniendo la información solicitada o una advertencia anunciando un error de compilación de la información o una disconformidad con los parámetros evaluados. La respuesta final es dada individualmente para cada uno de los elementos de concreto determinados al inicio del trabajo como elementos de marco estructural (Vigas y

Columnas). Tal información puede ser vinculada a los elementos a través de la organización de la lista o con los valores de indexación en el documento de MS Excel.

El algoritmo desarrollado satisface el objetivo principal para el cual fue concebido en este trabajo, el cual fue la creación de un sistema que auxiliara el proceso de toma de decisiones en términos de seleccionar los elementos de encofrado modular adecuados y su efectividad puede ser demostrada en la información contenida en cada uno de los elementos del modelo.

La creación del modelo piloto permite generar un algoritmo que extraiga información desde los elementos de concreto de un modelo BIM, y el algoritmo desarrollado puede ser implementado sobre la plataforma de Autodesk Revit para su aplicación. Por su parte los resultados quedan plasmados en el formato .xls, lo que permite su procesamiento desde cualquier programa, o bien su análisis visual.

# Recomendaciones

Para optimizar la funcionalidad del algoritmo y extender su campo de aplicación, es necesario generar un parámetro de salida más sencillo y amigable con el usuario que permita interpretar los resultados de manera más eficiente. La manera más recomendada para facilitar la visualización de la información contenida en los resultados es mediante una interfaz gráfica que permita al usuario una cuantificación rápida y representativa de los elementos de encofrado necesarios para cada elemento de concreto.

Adicionalmente, se recomienda de parte del autor a los futuros investigadores, con un mayor conocimiento de lenguajes de programación, la interpretación de la información desde el formato ifcXML del modelo de Revit, para seguir el mismo procedimiento establecido por el algoritmo desarrollado sin la necesidad de la utilización de plataformas de programación visual.

Finalmente, es importante también hacer hincapié en el aporte que representaría el desarrollo de una interfaz propia para el algoritmo, para que el mismo comience a tomar la forma de un software, desde donde pueda ejecutarse sin la necesidad de programas de terceras partes.

# Apéndices

A continuación, se adjuntan a manera de apéndices los resultados impresos en MS Excel como parámetro de salida final del algoritmo desarrollado en el proyecto.

La caracterización de los mismos corresponde a:

1. Tabla resumen de los resultados obtenidos para vigas en el modelo piloto.
2. Tabla resumen de los resultados obtenidos para columnas en el modelo piloto.
3. Tabla resumen de los resultados obtenidos para columnas en el modelo de validación.
4. Tabla resumen de los resultados obtenidos para vigas en el modelo de validación.



**Apéndice 1. Tabla resumen de los resultados obtenidos para vigas en el modelo piloto.**





**Apéndice 2. Tabla resumen de los resultados obtenidos para columnas en el modelo piloto.**











## **Apéndice 3. Tabla resumen de los resultados obtenidos para columnas en el modelo de validación.**









J-3	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
J-1	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
F-7	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
F-8	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
E-7	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
E-8	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
G-7	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
G-8	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
H-7	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
H-8	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
J-7	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
J-8	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
M-7	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
M-8	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
Q-7	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
Q-8	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
R-7	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
R-8	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
S-7	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
S-8	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
T-7	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
T-8	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
U-6	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
U-9	Base Level : Foundation Bottom	400 x 400mm	0,4	0,4	2,2	0,352 w 0.40, l 2.40
O-000(171)	Base Level : Level 1	300 x 300mm	0,3	0,3	3	0,27 w 0.30, l 3.00
K-000(171)	Base Level : Level 1	300 x 300mm	0,3	0,3	3	0,27 w 0.30, l 3.00

**Apéndice 4. Tabla resumen de los resultados obtenidos para vigas en el modelo de validación.**



















# Anexos

Dentro de los documentos utilizados por terceras partes no vinculados directamente al desarrollo del proyecto, destacan las tablas brindadas en el reglamento ACI 347R-14: "Guía para el Encofrado del Concreto", las cuales son adjuntadas a continuación.



Anexo 01. Tablas del ACI 347R-14 Guías para el Encofrado en Concreto.

**Table 4.2.2.1a(a)—Applicable lateral pressure equations for concrete other than SCC - Inch-pound version**

Slump <sup>a</sup>	Internal vibration depth	Element	Rate of placement	Pressure equation
Greater than 7 in.	Any	Any	Any	4.2.2.1a(a)
Less than or equal to 7 in.	Greater than 4 ft	Any	Any	4.2.2.1a(a)
Less than or equal to 7 in.	Less than or equal to 4 ft	Column <sup>b</sup>	Any	4.2.2.1a(b)
		Wall <sup>c</sup> less than or equal to 14 ft tall	Less than 7 ft/h	4.2.2.1a(b)
		Wall <sup>c</sup> greater than 14 ft tall	Less than 7 ft/h	4.2.2.1a(c)
		Wall <sup>c</sup>	7 to 15 ft/h	4.2.2.1a(c)
			Greater than 15 ft/h	4.2.2.1a(a)

<sup>a</sup>Slump for determination of lateral pressure shall be measured after the addition of all admixtures.

<sup>b</sup>For the purpose of this document, columns are defined as vertical elements with no plan dimension exceeding 6.5 ft.

<sup>c</sup>For the purpose of this document, walls are defined as vertical elements with at least one plan dimension exceeding 6.5 ft.

**Table 4.2.2.1a(b)—Chemistry coefficient C<sub>c</sub>**

Cement type	Slag	Fly ash	Retarders <sup>a</sup>	C <sub>c</sub>
I, II, or III	None	None	None	1.0
			Included	1.2
Any	Less than 70 percent	Less than 40 percent	None	1.2
			Included	1.4
	Greater than or equal to 70 percent	Greater than or equal to 40 percent	None	1.4
			Included	1.5

<sup>a</sup>Retarders include any admixture, such as a retarder, retarding water reducer, retarding mid-range water-reducing admixture, or high-range water-reducing admixture, that delays setting of concrete.

**Table 4.2.2.1a(c)—Unit weight coefficient C<sub>w</sub>**

Inch-pound version		SI version	
Unit weight of concrete, lb/ft <sup>3</sup>	C <sub>w</sub>	Density of concrete, kg/m <sup>3</sup>	C <sub>w</sub>
w < 140	0.5[1 + (w/145 lb/ft <sup>3</sup> )] but not less than 0.80	ρ < 2240	0.5[1 + (w/2320 kg/m <sup>3</sup> )] but not less than 0.80
140 ≤ w ≤ 150	1.0	2240 ≤ ρ ≤ 2400	1.0
w > 150	w/145 lb/ft <sup>3</sup>	ρ > 2400	w/2320 kg/m <sup>3</sup>

**Table 4.2.2.1b—Applicable lateral pressure equations for concrete other than SCC - SI version**

Slump <sup>a</sup>	Internal vibration depth	Element	Rate of placement	Pressure equation
Greater than 175 mm	Any	Any	Any	4.2.2.1b(a)
Less than or equal to 175 mm	Greater than 1.2 m	Any	Any	4.2.2.1b(a)
Less than or equal to 175 mm	Less than or equal to 1.2 m	Column <sup>b</sup>	Any	4.2.2.1b(b)
		Wall <sup>c</sup> less than or equal to 4.2 m tall	Less than 2.1 m/h	4.2.2.1b(b)
		Wall <sup>c</sup> greater than 4.2 m tall	Less than 2.1 m/h	4.2.2.1b(c)
		Wall <sup>c</sup>	2.1 to 4.5 m/h	4.2.2.1b(c)
			Greater than 4.5 m/h	4.2.2.1b(a)

<sup>a</sup>Slump for determination of lateral pressure shall be measured after the addition of all admixtures.

<sup>b</sup>For the purpose of this document, columns are defined as vertical elements with no plan dimension exceeding 2 m.

<sup>c</sup>For the purpose of this document, walls are defined as vertical elements with at least one plan dimension exceeding 2 m.

(Fuente. ACI 347R-14, 2014)

# Referencias

- ACI Committee 306 (2016). "Guide to Cold Weather Concreting (ACI 306R-16)". Miami, USA: American Concrete Institute.
- ACI Committee 347 (2014). "Guide to Formwork for Concrete (ACI 347R-14)". Miami, USA: American Concrete Institute.
- Ding, L., Zhoyu, Y., and Akinci, B. (2014). " Building Information Modeling ( BIM ) application framework: the process of expanding from 3D to computable nD." *Automation in Construction* (46), 82-93.
- Dolly, M., Debaditya, C., Hazem, E., and Abhijeet, D. (2017)." Building Information Modeling enabled Cascading Formwork Management Tool." *Automation in Construction* (83), 259-272.
- Emad, E. E., Ossama, A. H., Ahmed, E., Mohamed, E. A., and Mohammed, S. E.-A. (2012). " Fuzzy Logic Model for Selection of Vertical Formwork Systems." *ASCE Journal of Construction Engineering Management*, 832-840.
- Hou, L., Zhao, C., Wu, C., Moon, S., and Wang, X. (2017). " Discrete Firefly Algorithm for Scaffolding Construction Scheduling." *ASCE Journal of Computing in Civil Engineering*(31), 04016064.
- Junbok, L., Young-Jin, P., Chang-Hoon, C., and Choong-Hee, H. (2017). " BIM-assisted labor productivity measurement method for structural formwork." *Automation in Construction* (84), 121-132.
- Jungjun Park, B. K. (2011). "3D/4D CAD Applicability for Life-Cycle." *ASCE Journal of Computing in Civil Engineering* 25(2), 129-138.
- Kartam, N. A., and Levitt, R. E. (1990). " Intelligent Planning of Construction Projects." *ASCE Journal of Computing in Civil Engineering* (4), 155-176.
- Kim, H., Anderson, K., Lee, S., and Hildreth, J. (2013). "Generating construction schedules through automatic data extraction using BIM (building information modeling) technology." *Automation in Construction* (35), 285-295.
- Kim, J., Fischer, M., Kunz, J., and Levitt, R. (2015). "Semiautomated Scaffolding Planning: Development of the Feature Lexicon for Computer Application." *ASCE Journal of Computing in Civil Engineering*(29), 04014079.
- Kim, K., and Teizer, J. (2014). "Automatic design and planning of scaffolding systems using building information modeling." *Advanced Engineering Informatics*(28), 66-80.
- Krawczyńska-Piechna, A. (2017) "Comprehensive Approach to Efficient Planning of Formwork Utilization on the Construction Site." 7th International Conference on Engineering, Project, and Production Management (pp. 366-372). Warsaw, Poland: Procedia Engineering.

- Kyungki, K., Yong, C., and Sijie, Z. (2016). "Integrating work sequences and temporary structures into safety planning: Automated scaffolding-related safety hazard identification and prevention in BIM." *Automation in Construction* (70), 128-142.
- Leicht, L. J. (2015). "Automated Rule-Based Constructability Checking: Case Study of Formwork." *ASCE Journal of Management in Engineering* (31), A4014004.
- Nunally, S. W. (2004). "Construction Methods and Management, 6th Edition." New Jersey, Pearson – Prentice Hall
- Qi, J., Issa, R. R., Olbina, S., and Hinze, J. (2014). "Use of Building Information Modeling in Design to Prevent Construction Worker Falls." *ASCE Journal of Computing in Civil Engineering*(28), A4014008.
- Rebekka Volk, J. S. (2014). "Building Information Modeling (BIM) for existing buildings: Literature review and future needs." *Automation in Construction* (38), 109-127.
- Ringwald, R. C. (1985). "Formwork Design." *ASCE Journal of Construction Engineering Management*, 391-403.
- Taehoon Kim, H. L.-K.-i. (2012). "Advanced formwork method integrated with a layout planning model for tall building construction." *Canadian Journal of Civil Engineering* (39), 1173 - 1183.