

# Como implementar la automatización en sus proyectos de diseño

**David LICONA**

Consultor Implementación BIM | @David Licona





# Sobre el presentador

David LICONA

- Ingeniero Civil
- 5+ años de experiencia en la fase de diseño, construcción y operación de infraestructuras lineales
- Consultor para Autodesk basado en París
- 1ra participación en AU

# Objetivos de aprendizaje



## IDENTIFICAR LAS OPORTUNIDADES DE AUTOMATIZACIÓN

- Por qué la automatización ?
- Identificar casos de aplicación
- Establecer prioridades



## DESARROLLAR SOLUCIONES INDUSTRIALIZABLES

- Dynamo Player
- Instrucciones de utilización
- Soluciones modulares



## PENSAR ANALÍTICAMENTE

- Describir el problema
- Crear un « POC » rápidamente
- Conceptos claves de Dynamo



## IMPLEMENTAR LA AUTOMATIZACIÓN

- Mejores prácticas
- Comunicación de los resultados
- Recursos necesarios para triunfar





Foto por [Frame Harirak](#) - [Unsplash](#)

## Expectativas :

- Consejos, herramientas, metodologías para lograr implementar la automatización
- Caso práctico : modelización de movimientos de tierras (corte y relleno) como objetos (BIM 4D, BIM 5D)
- Dynamo para Civil 3D



Foto por [Lindsay Henwood](#) - [Unsplash](#)

## Y no ...

- Una demostración paso a paso de como construir gráficas en Dynamo
- Un curso de Dynamo para Civil 3D



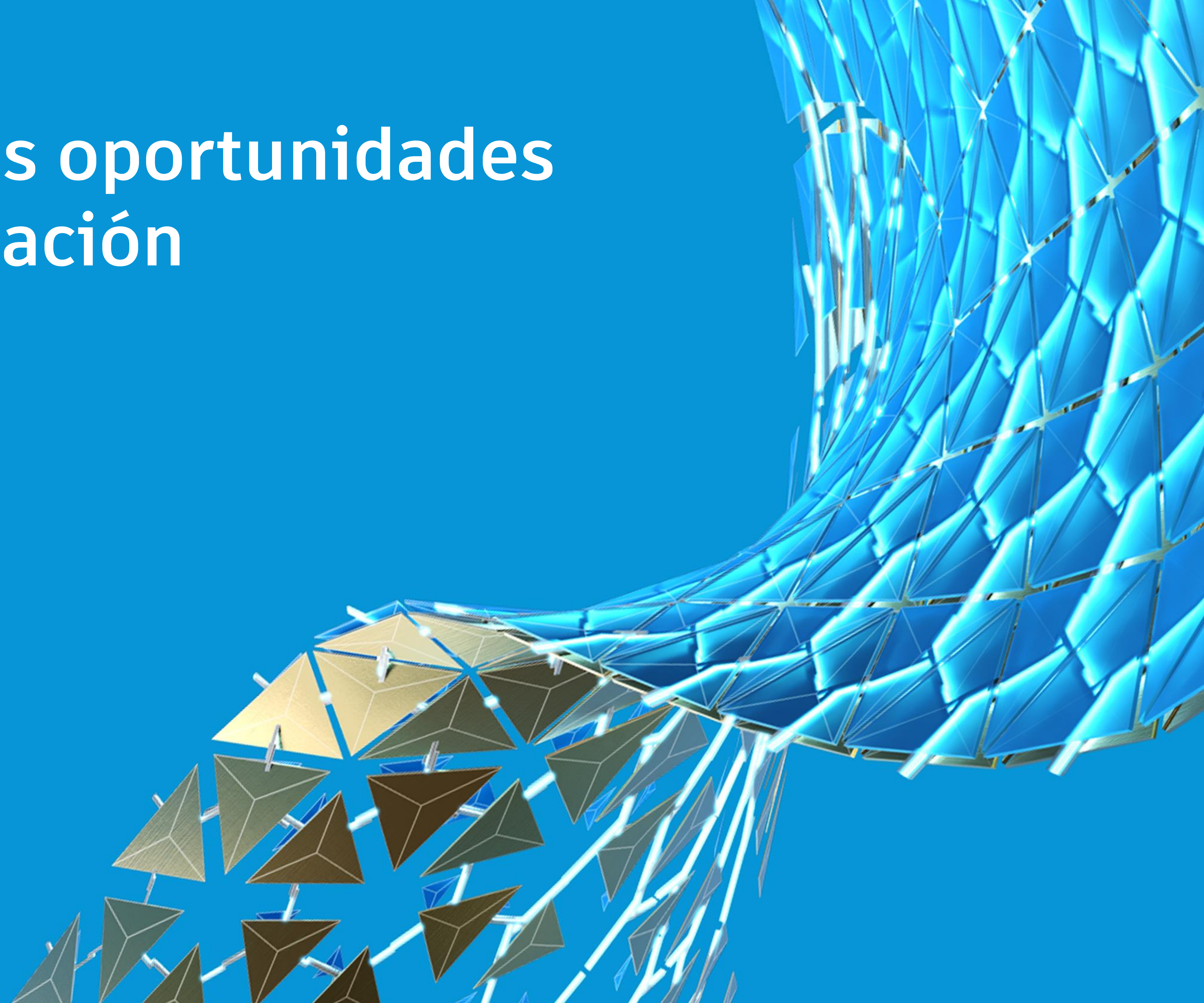
# Dynamo Foundation

- **Oferta de Autodesk Consulting**
  - Resolver los casos de utilización más complejos
  - Mejorar la colaboración dentro de la organización
  - Crear una comunidad alrededor de Dynamo y a través de disciplinas y localizaciones





# Identificar las oportunidades de automatización





# Definamos la automatización

- Reducir la intervención humana para realizar acciones
- No es el futuro, es el **presente**





# Programación Visual

- Un medio para los ingenieros
  - Accesible a todo
  - Liberar la creatividad



Foto por [Daniel Tuttle - Unsplash](#)

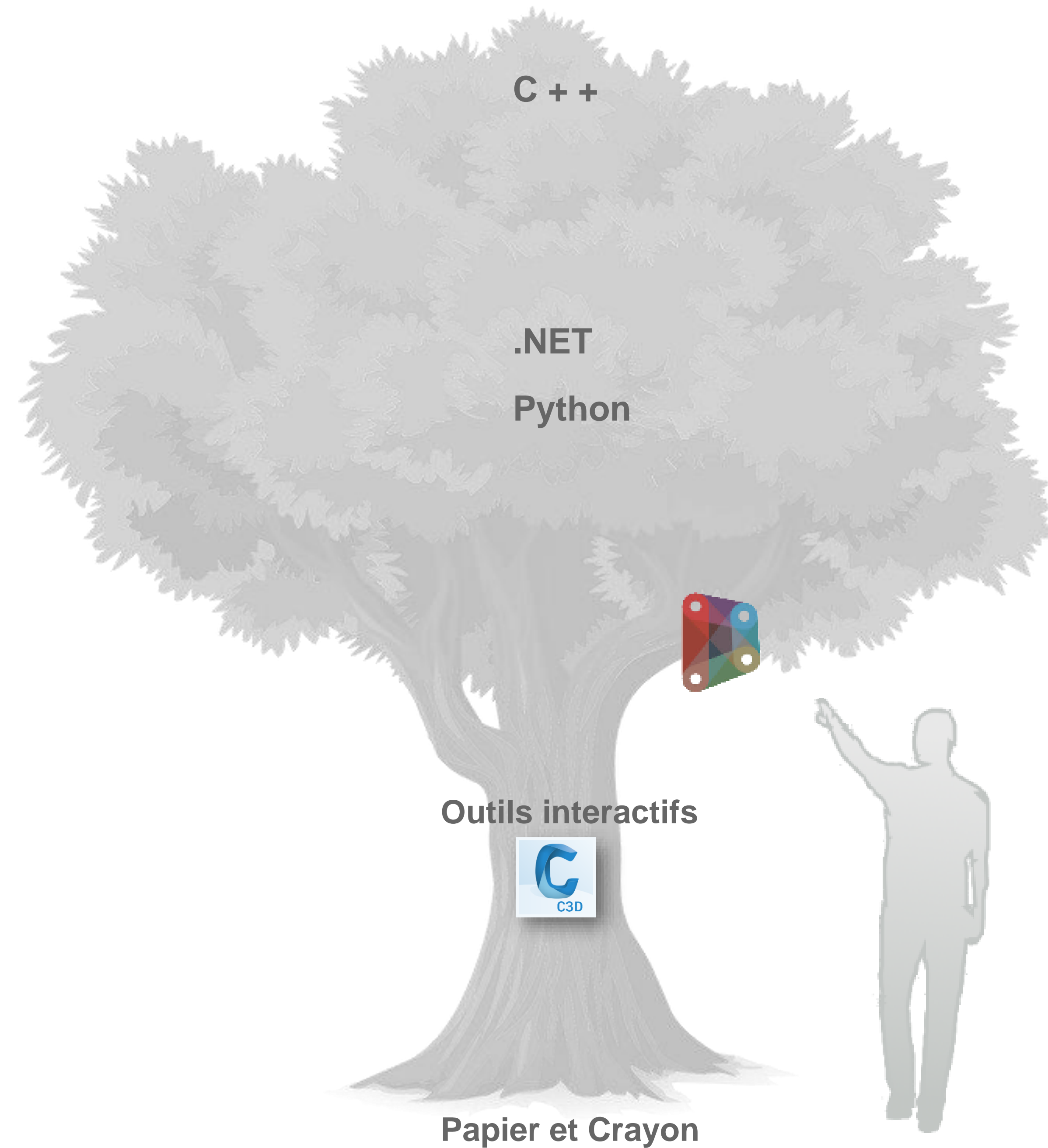


Foto por [Vitae London - Unsplash](#)



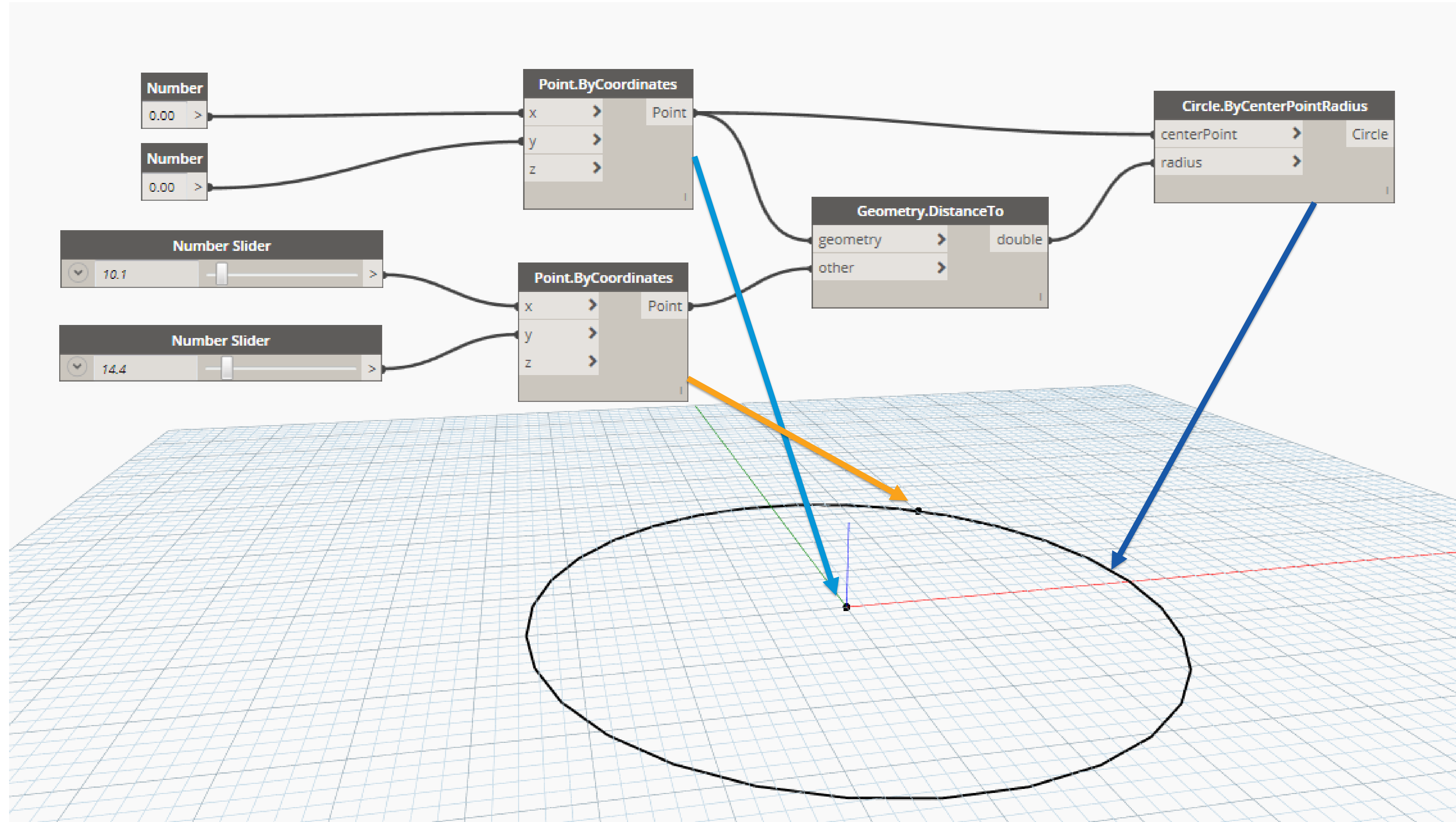
# Dynamo para Civil 3D

- Dynamo es una herramienta de programación visual
- Integrada en Civil 3D





# Programación Visual





# ¿Cuál propósito ?

- Casos **MÁS** complejos, modelos **MÁS** detallados
- **MEJORAR** la calidad
- Con **MENOS** :
  - ↓ personal
  - ↓ tiempo
  - ↓ retrabajar
  - ↓ trabajo repetitivo





# Concretamente...

- Automatizar tareas repetitivas
- Crear flujos de trabajo complejos, conectando múltiples fuentes de datos
- Mejorar la interoperabilidad de los programas
- Explorar opciones de diseño rápidamente
- Acceder a los datos del modelo que usualmente ignoramos
- Crear y analizar rápidamente formas geométricas complejas





# Concretamente...

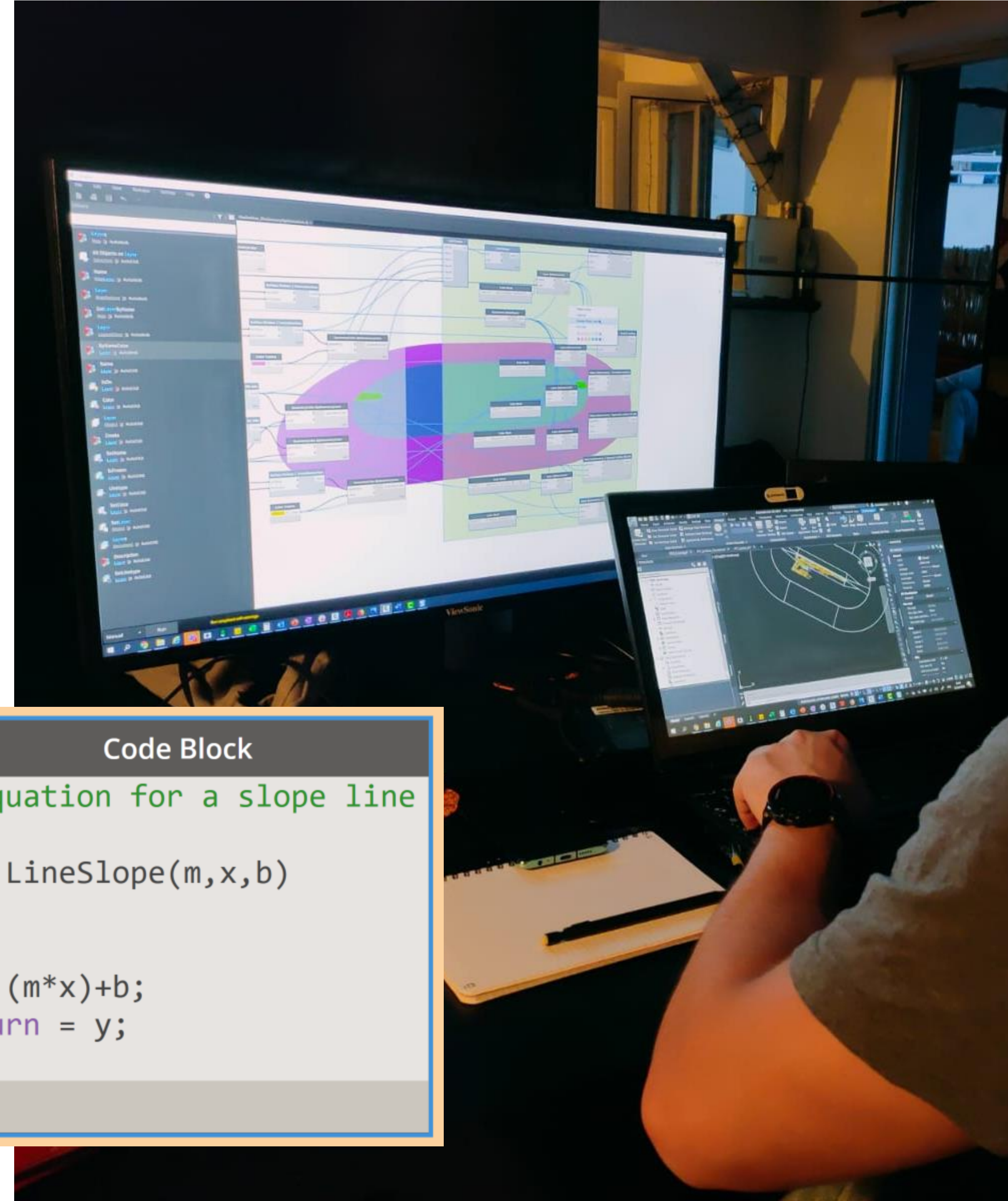
- Automatizar tareas repetitivas
- Crear flujos de trabajo complejos, conectando múltiples fuentes de datos
- Mejorar la interoperabilidad de los programas
- Explorar opciones de diseño rápidamente
- Acceder a los datos del modelo que usualmente ignoramos
- **Crear y analizar rápidamente formas geométricas complejas**





# Concretamente...

- Automatizar tareas repetitivas
- Crear flujos de trabajo complejos, conectando múltiples fuentes de datos
- Mejorar la interoperabilidad de los programas
- Explorar opciones de diseño rápidamente
- Acceder a los datos del modelo que usualmente ignoramos
- **Crear y analizar rápidamente formas geométricas complejas**



## Code Block

```
//Equation for a slope line  
  
def LineSlope(m,x,b)  
  
{  
y = (m*x)+b;  
return y;  
};
```



# Valor para todo tipo de organización

- **Empresas pequeñas**
  - Competir con las más grandes gracias a la automatización
- **Empresas medianas**
  - Reducción de costos internamente para reinvertir en innovación
- **Empresas grandes**
  - Escalar las mejores prácticas de manera más eficaz



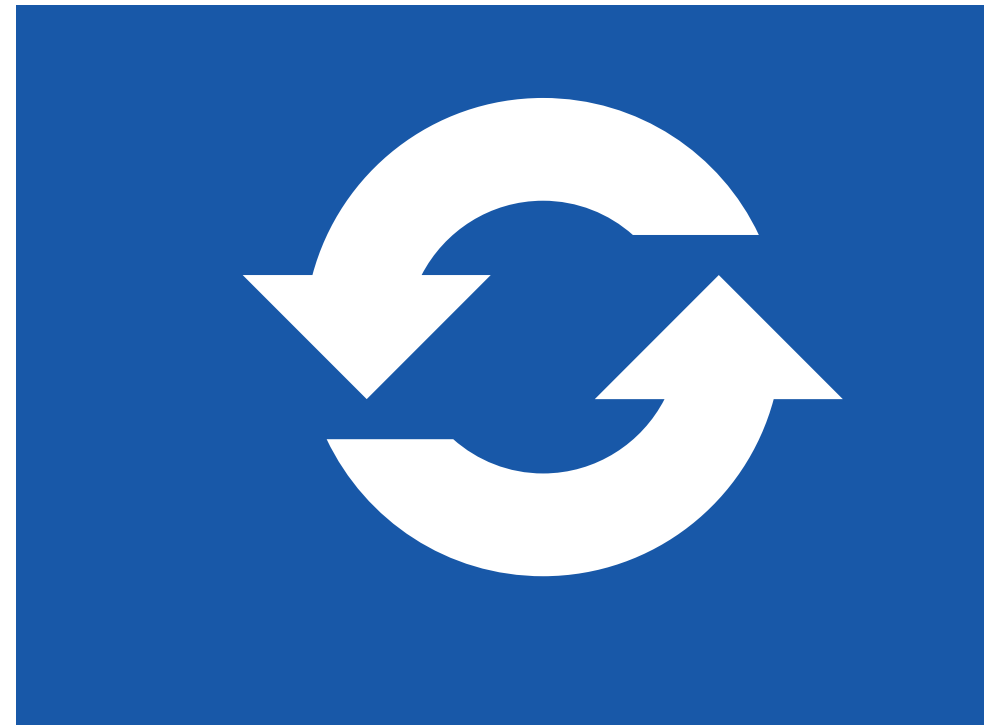


# Para identificar las oportunidades

¿El proceso es  
repetitivo ?



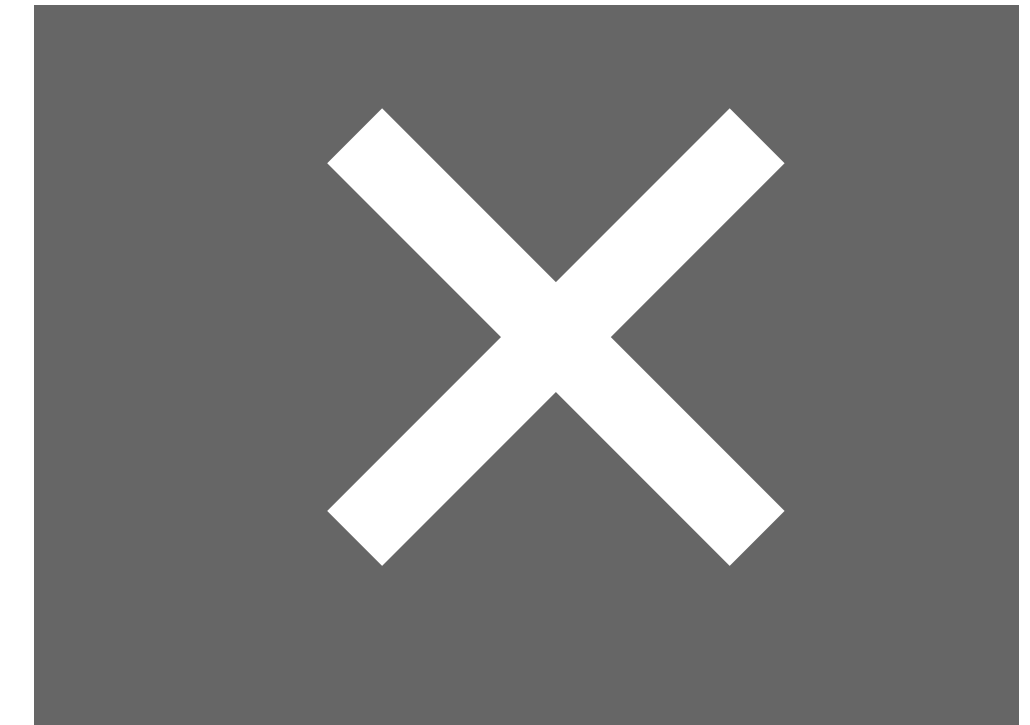
¿Es algo a realizar de  
manera sistemática?



¿Es algo que toma  
mucho tiempo?



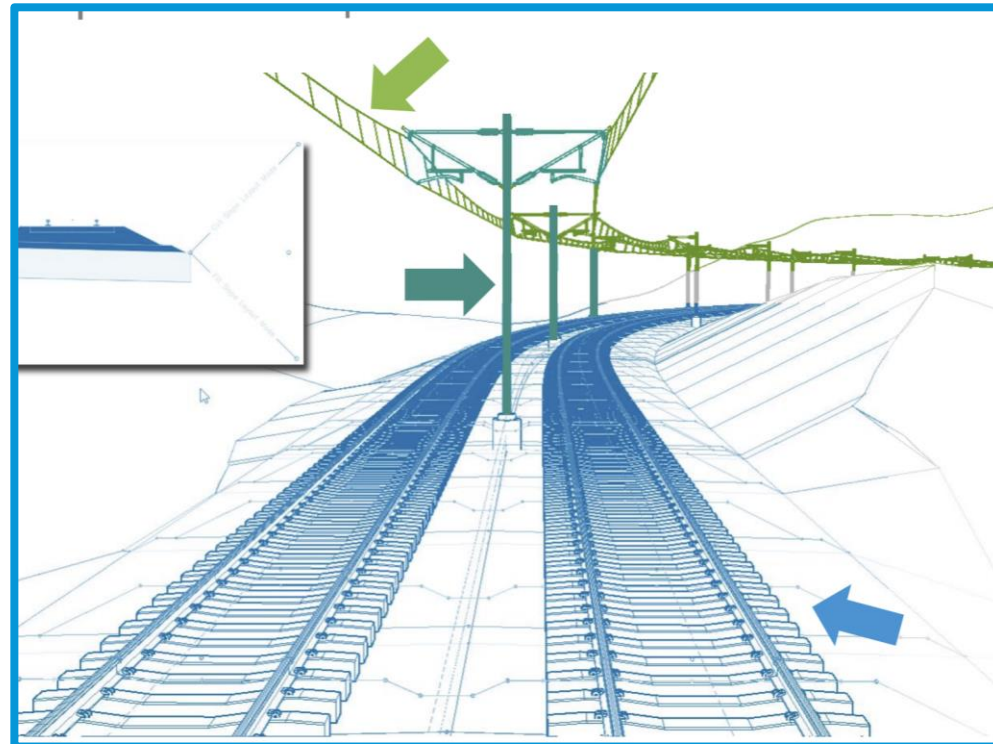
¿Es un proceso sensible al  
error humano?





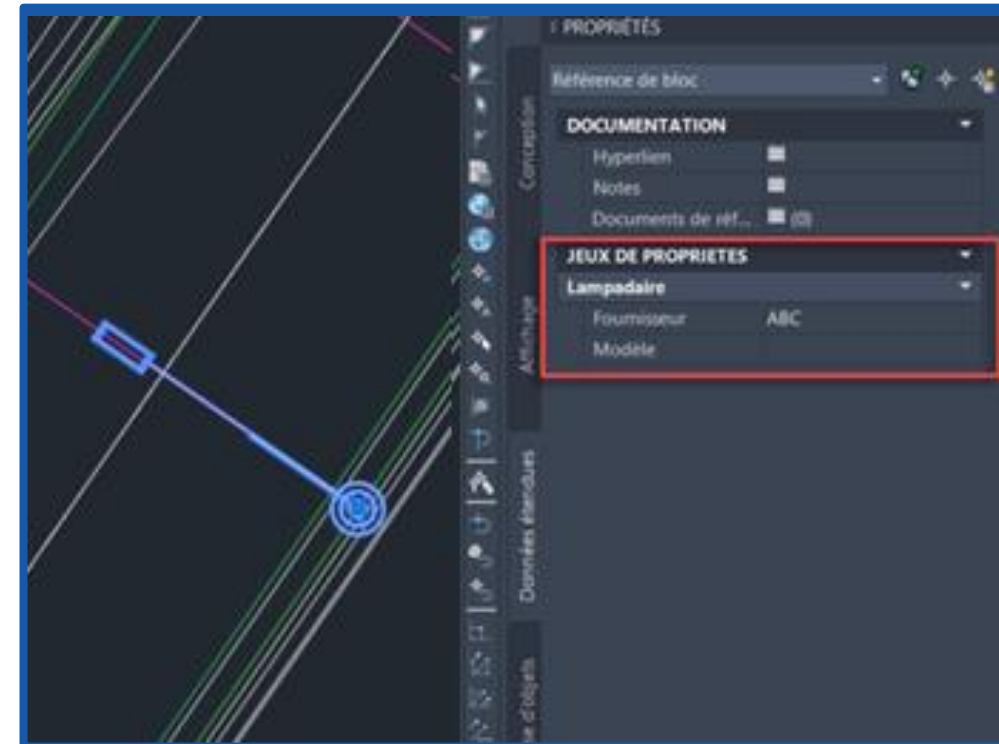
# Casos prácticos

¿El proceso es repetitivo ?



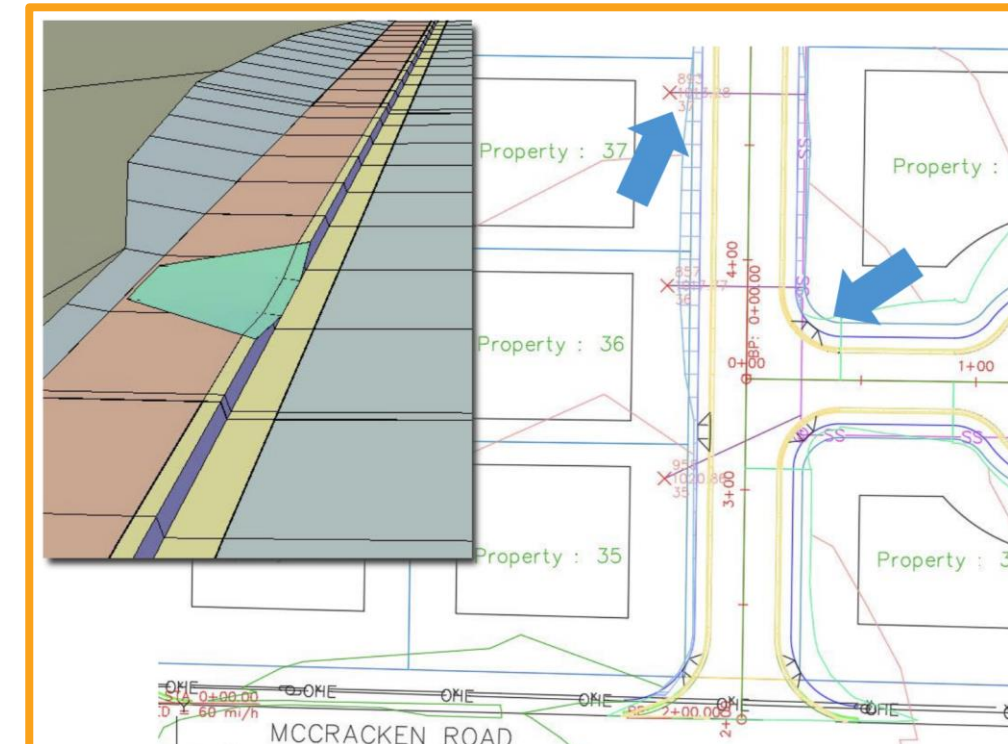
Insertar bloque a lo largo de un proyecto (durmientes, postes de luz, catenarias)

¿Es algo a realizar de manera sistemática?



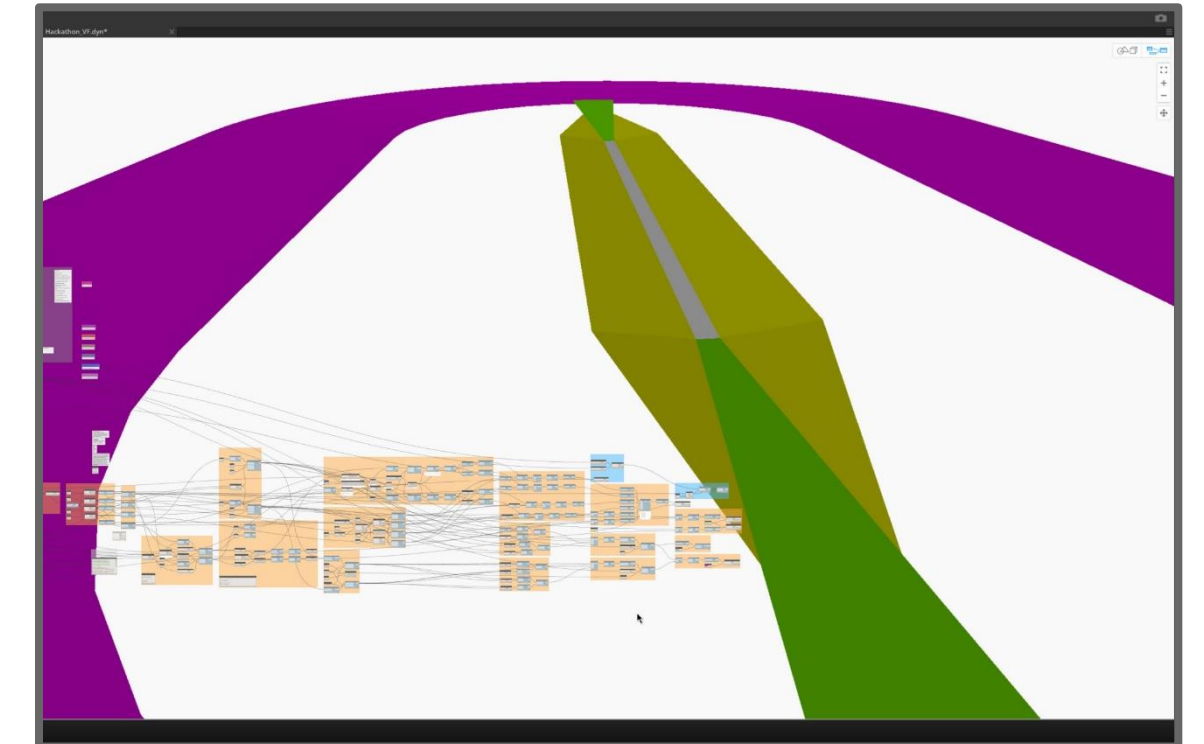
Aplicación de los property sets

¿Es algo que toma mucho tiempo?



Modelización de rampas para las aceras

¿Es un proceso sensible al error humano?



Geometría compleja

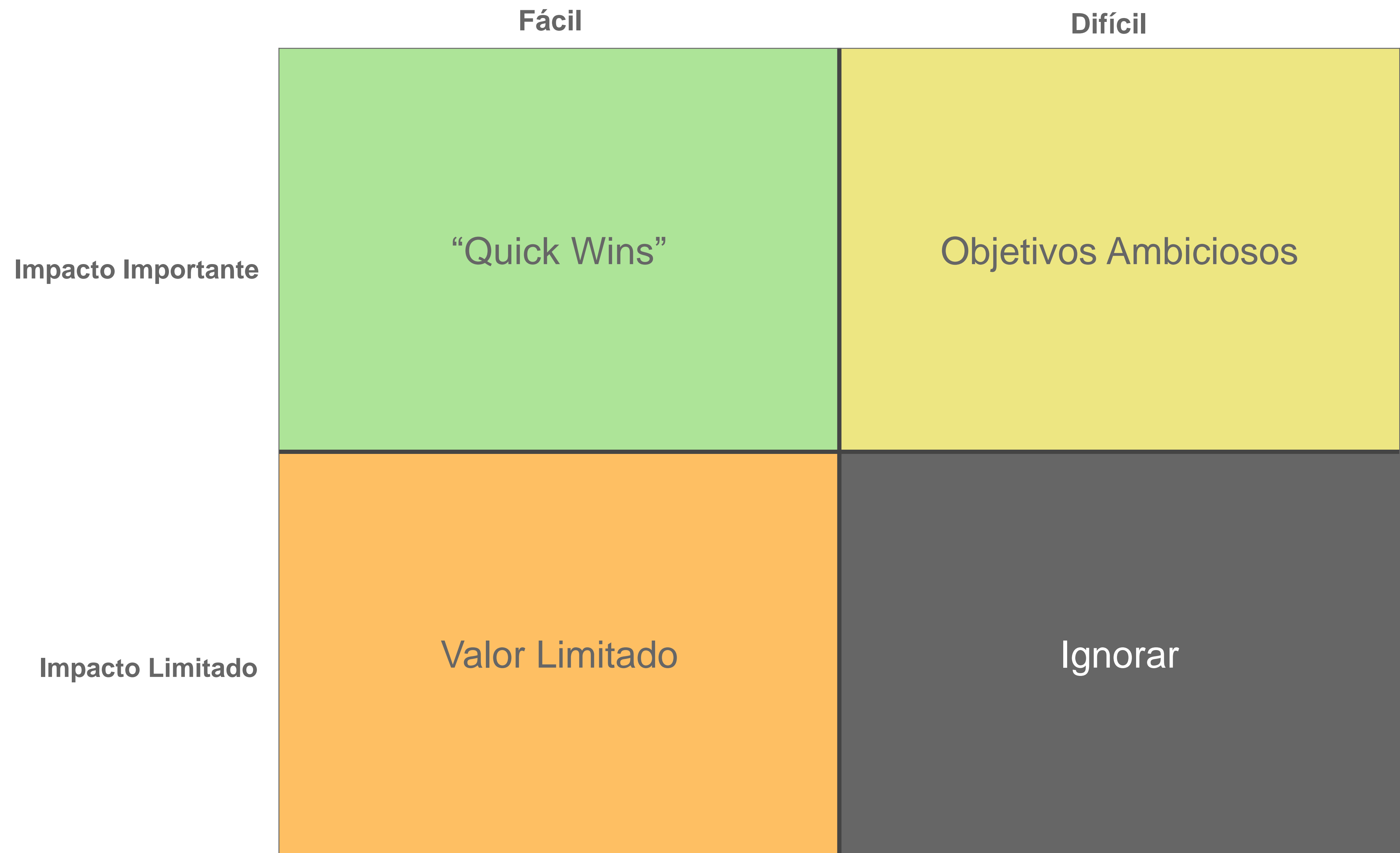


# Priorizar las ideas

- Concentrarse en los “quick wins”
- Valor visible con un “mínimo de esfuerzo”









# La base para abrir muchas posibilidades

## MACHINE LEARNING

## DISEÑO GENERATIVO

## PROGRAMACIÓN VISUAL

## DISEÑO PARAMÉTRICO

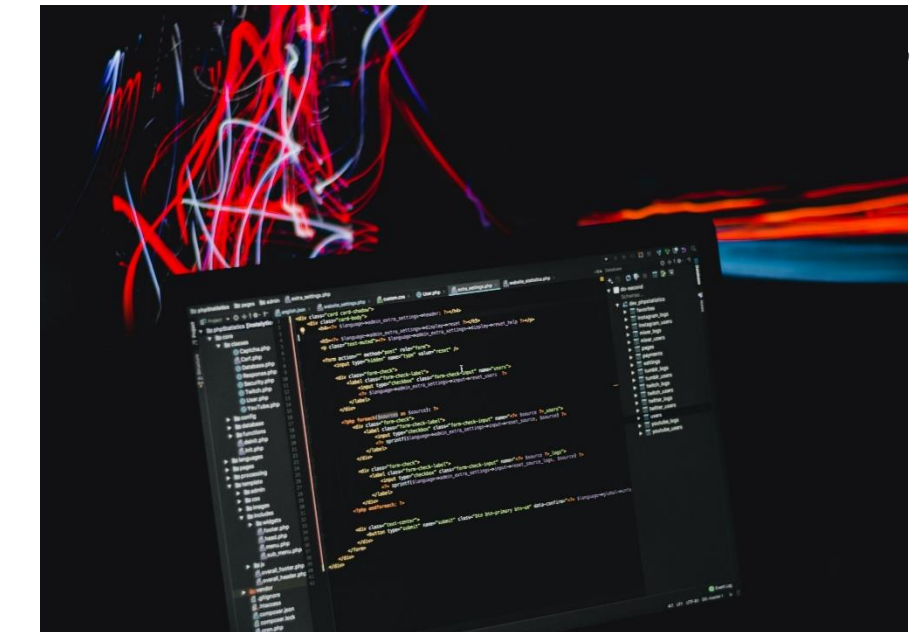
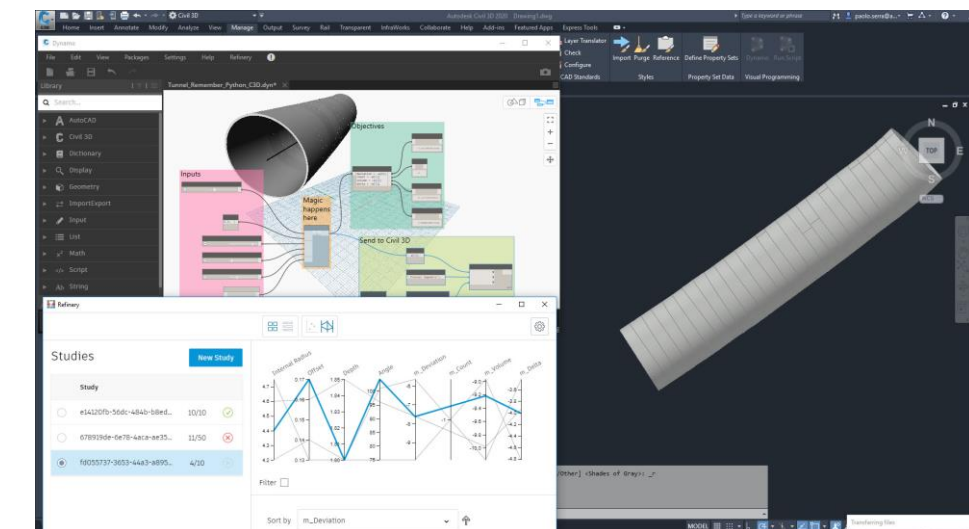
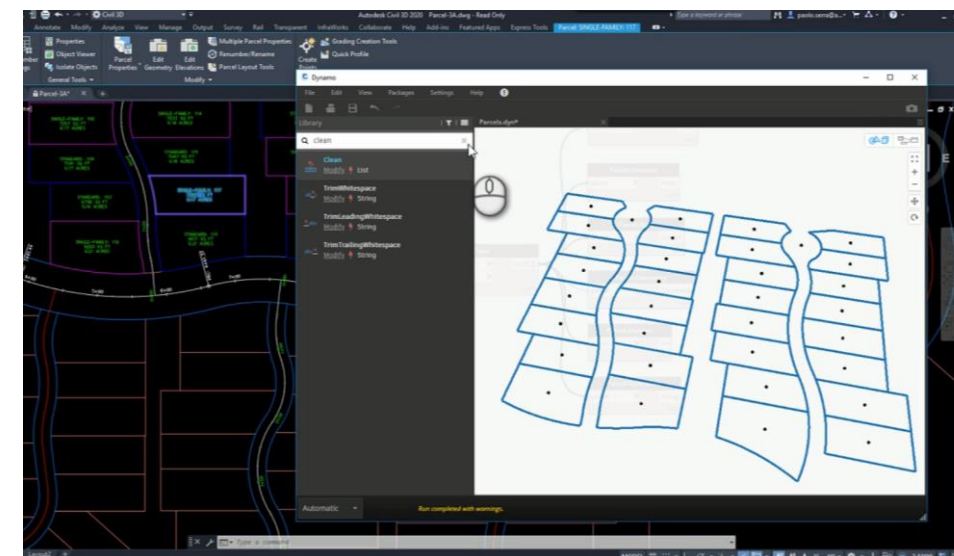
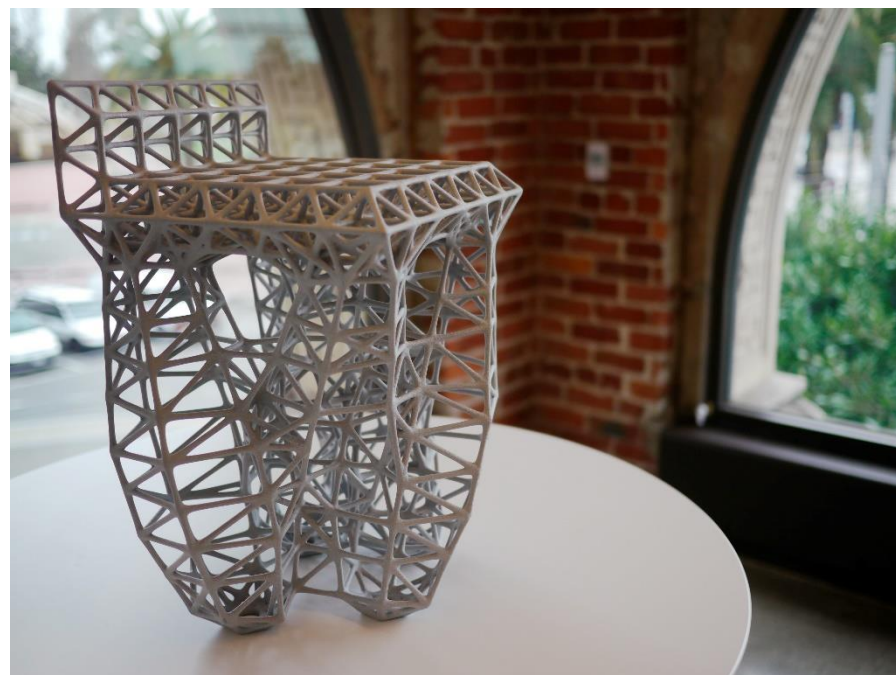


Foto por [AltumCode](#) - [Unsplash](#)



# Objetivos de aprendizaje



## IDENTIFICAR LAS OPORTUNIDADES DE AUTOMATIZACIÓN

- ✓ La automatización para realizar más, mejor, y con menos recursos
- ✓ Casos prácticos : tareas repetitivas, tareas con poco valor añadido, procesos iterativos, etc.
- ✓ Centrarse en los “quick wins”



## DESARROLLAR SOLUCIONES INDUSTRIALIZABLES



## PENSAR ANALÍTICAMENTE

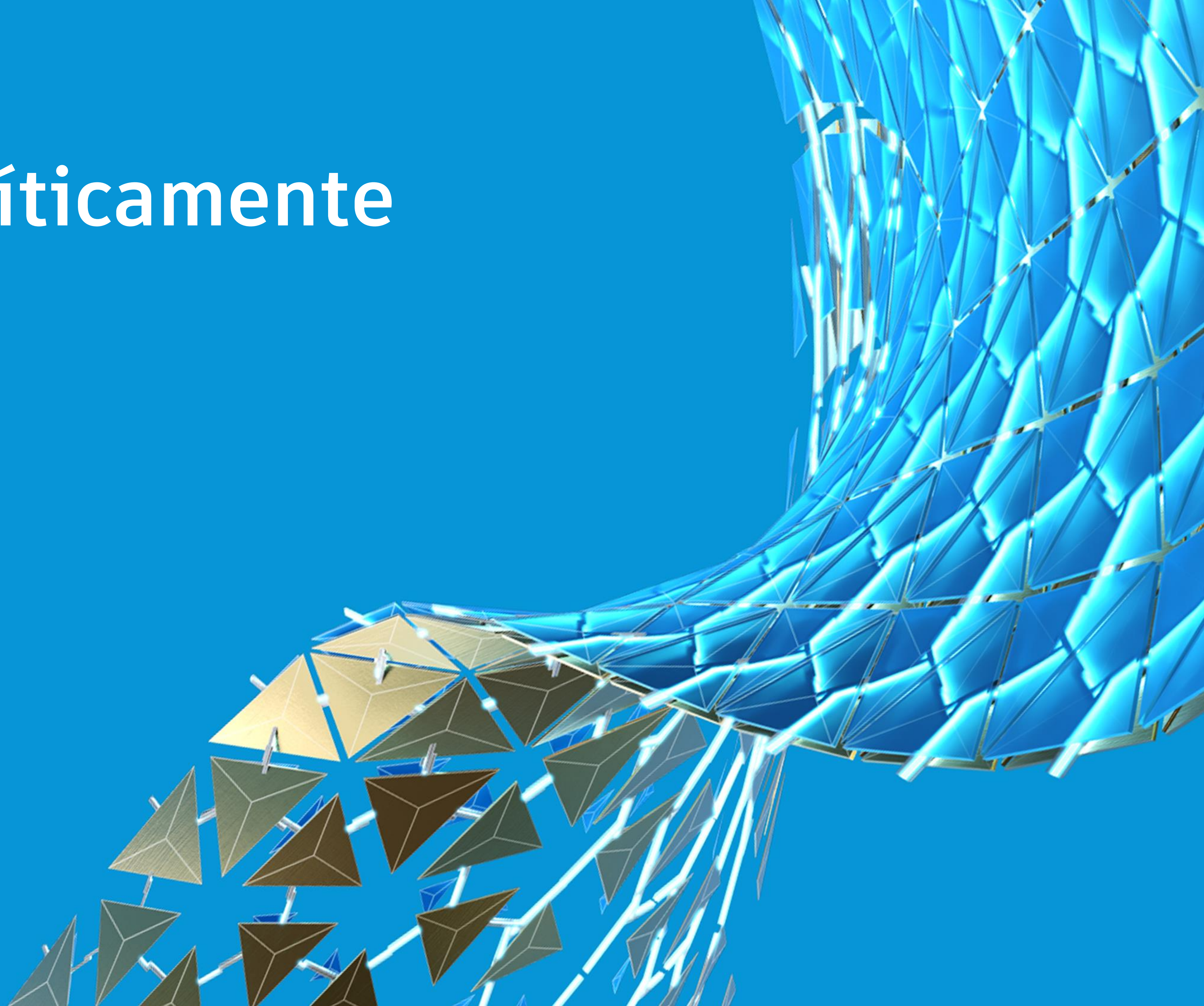
- Describir el problema
- Crear un « POC » rápidamente
- Conceptos claves de Dynamo



## IMPLEMENTAR LA AUTOMATIZACIÓN



# Pensar analíticamente



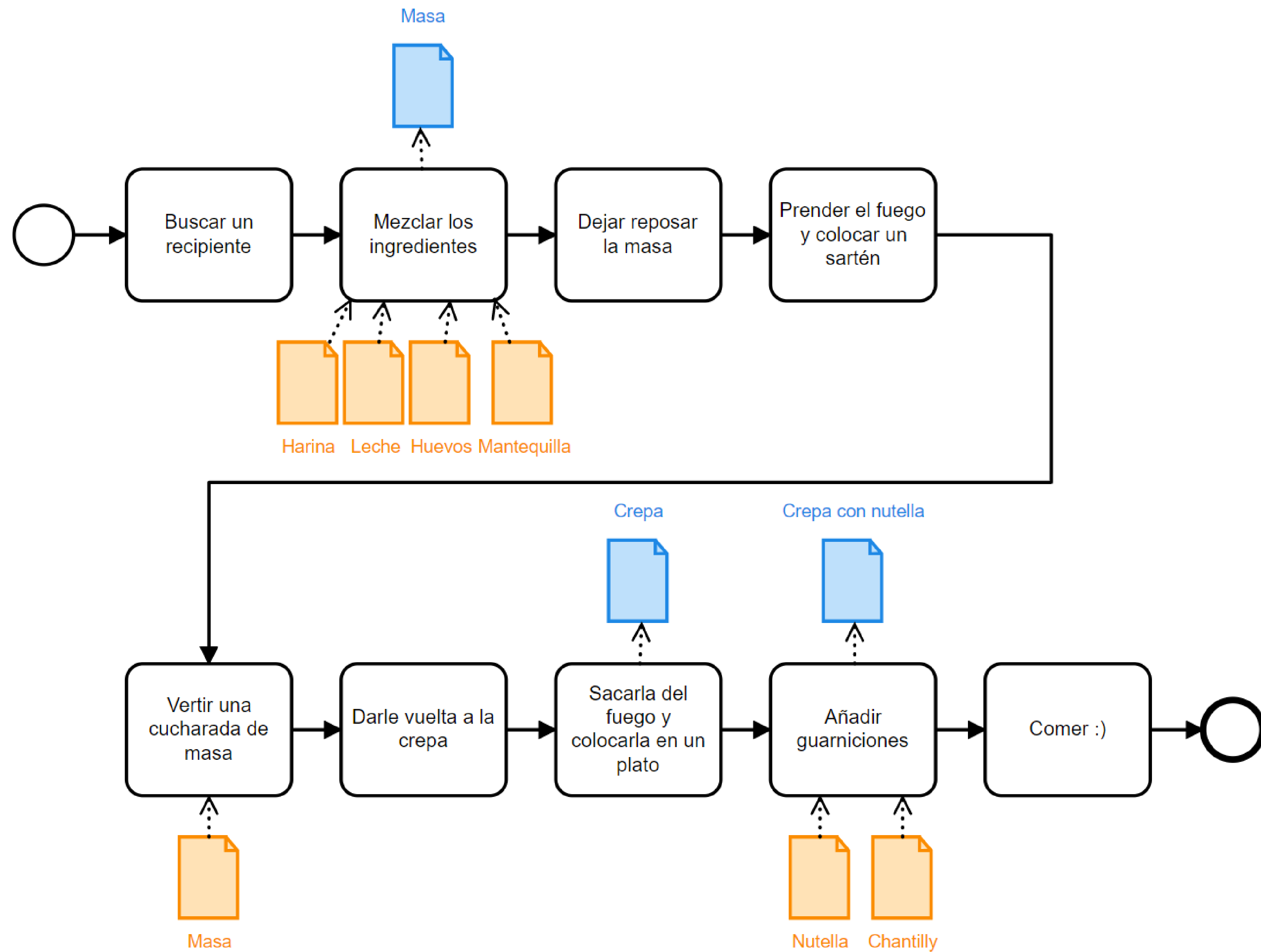


# Escribir la secuencia

- Ser capaz de describir el problema
  - Ordenar las tareas en una secuencia lógica
  - Visualizar el proceso
- Lenguaje BPMN
  - Programa : cawemo









# Caso práctico : Modelización de movimientos de tierras



Un script Dynamo para :

- Modelizar sólidos 3D para representar los cortes y rellenos para todo proyecto de carreteras
- Dividir los sólidos
  - Corte/Relleno
  - Al medio en la línea de base
  - Por cuerpo de tierra
- Añadir properties sets y sus propiedades :
  - Altura máxima
  - Altura promedio



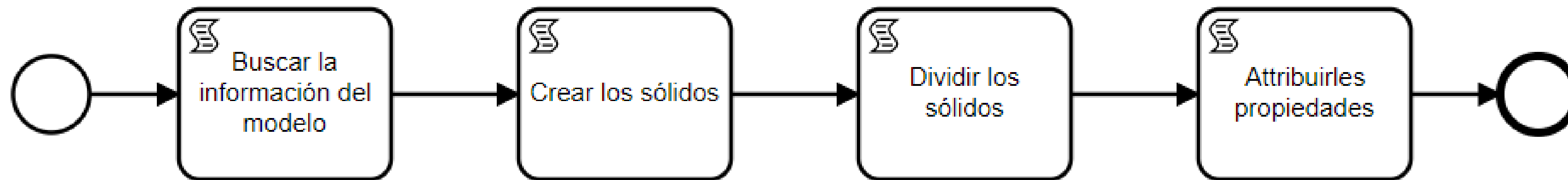
# Un proyecto exigente

- BIM4D & 5D
- 4 días para terminar el proyecto



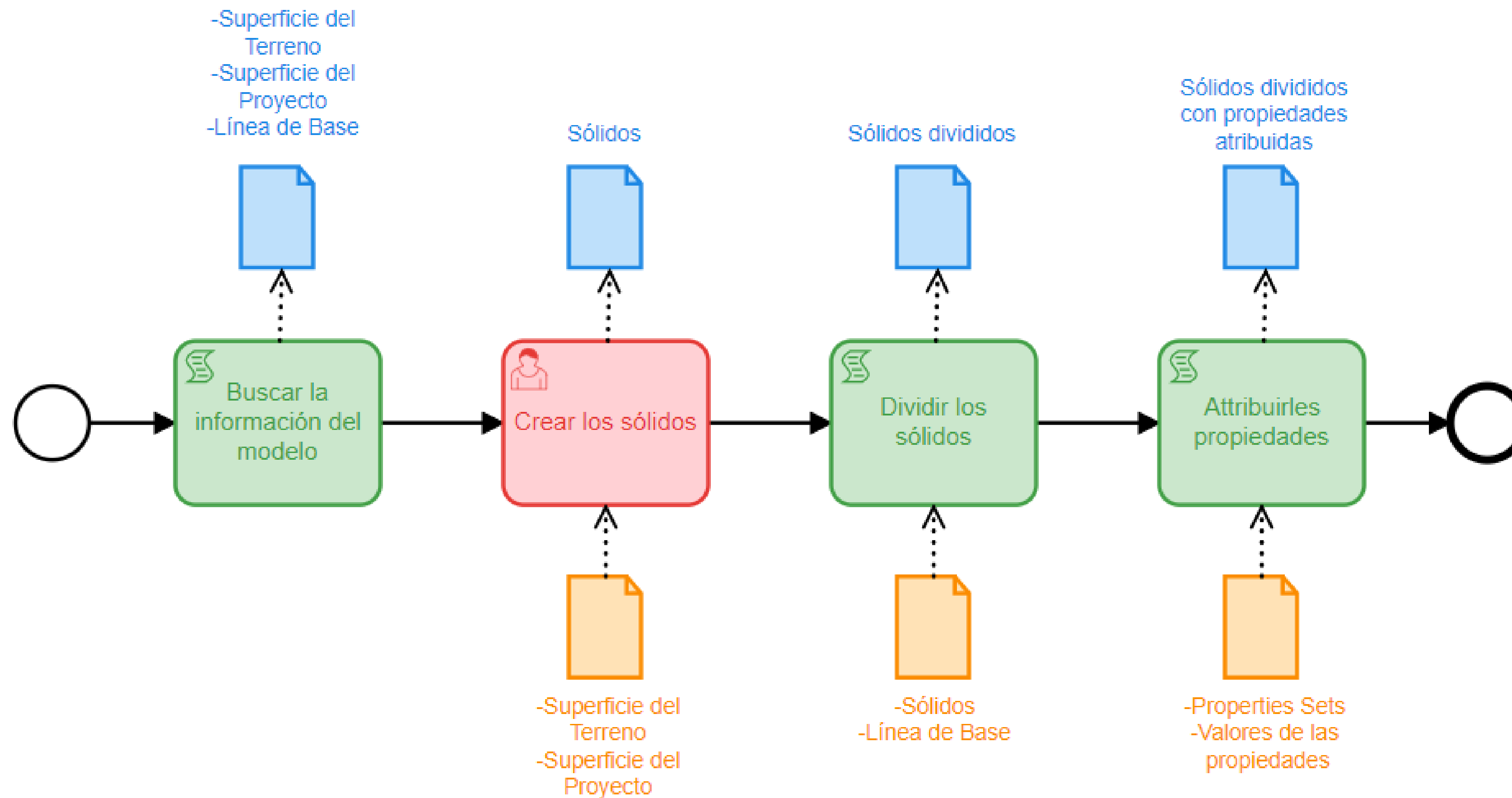


# 1 – ¿Cuál es el flujo de trabajo?



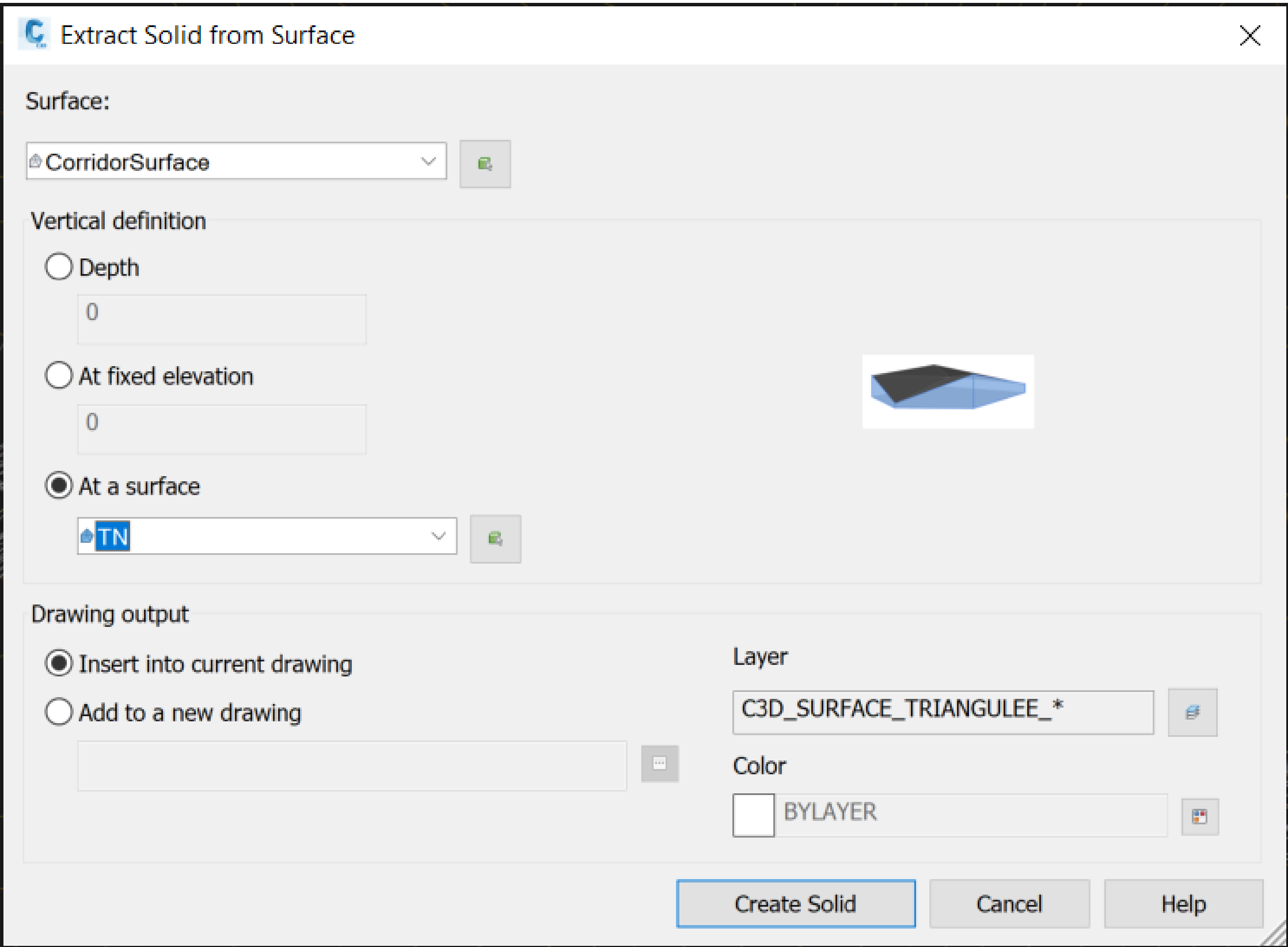


## 2 – Identificar las entradas/salidas y capacidades





# Necesidad de contar con recursos especializados



Extracts the solids between top and bottom Tin Surfaces.

```
TinSurfaceExtensions.CreateSolidsAtSurface (surface: TinSurface, bottomSurface: TinSurface, layer: string = "0", path: string = ""): Object[]
```

TinSurfaceExtensions.CreateSolidsAtSurface		
surface	>	Object[]
bottomSurface	>	
layer	>	
path	>	
AUTO		
Function		



# Construcción de gráficas

¿Qué tipo de gráficas?

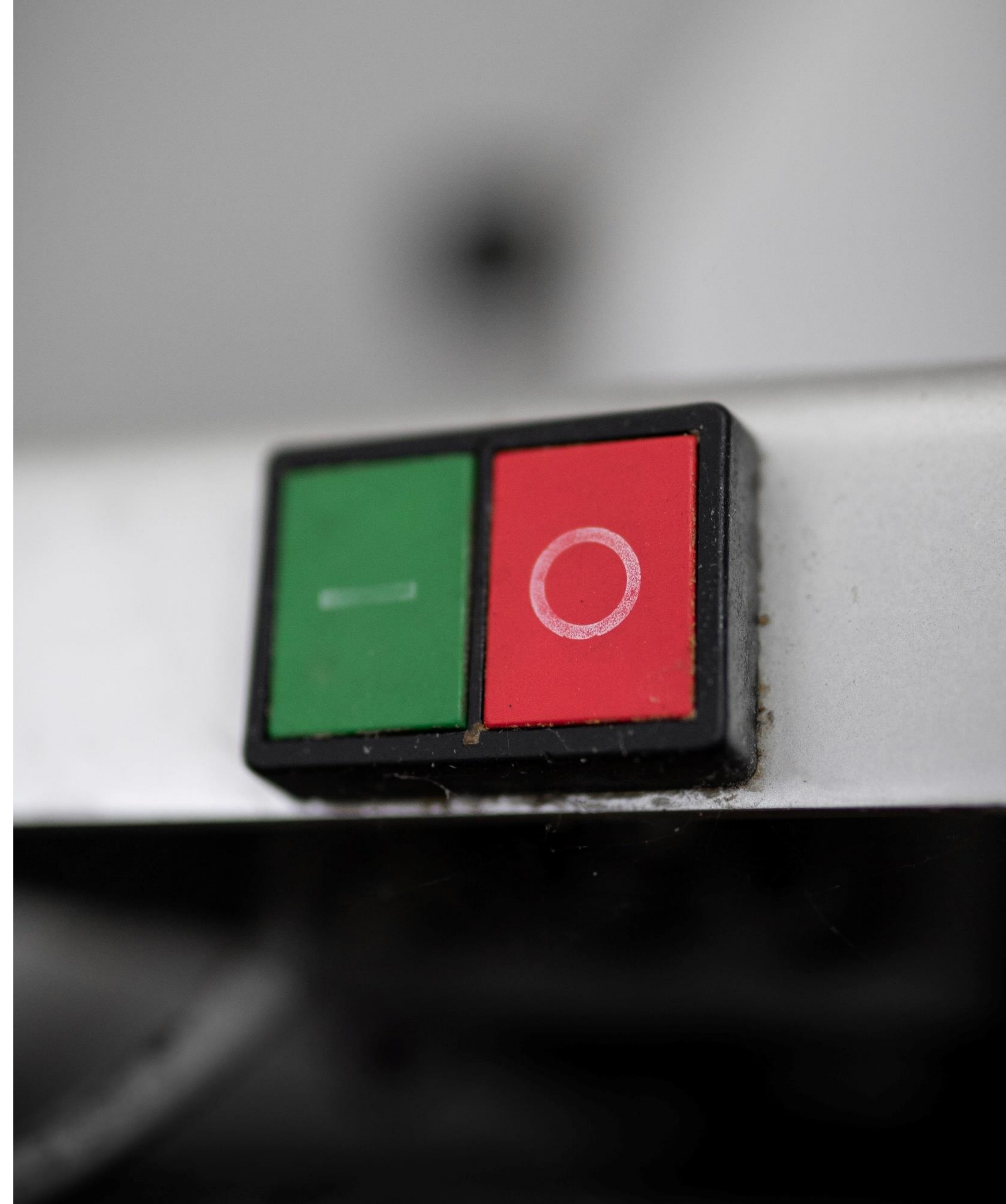
- Eficientes
- Genéricas
- Escalables





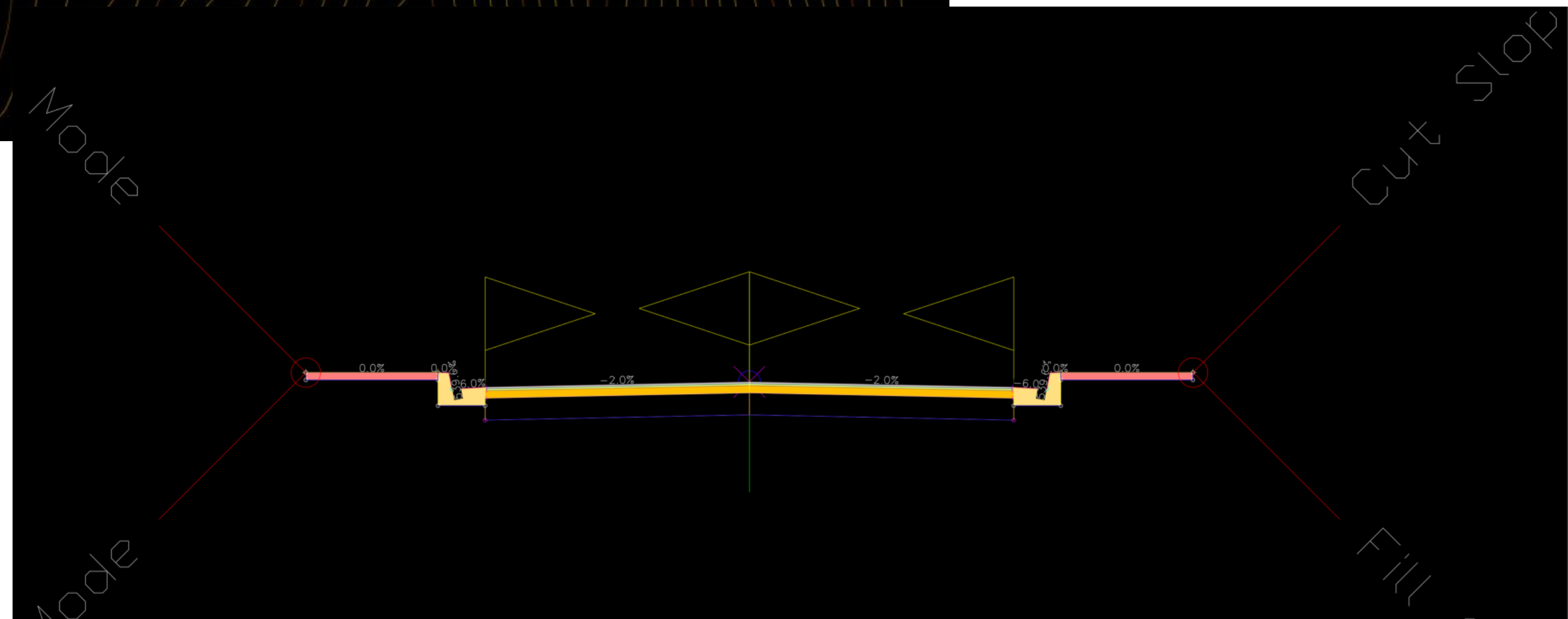
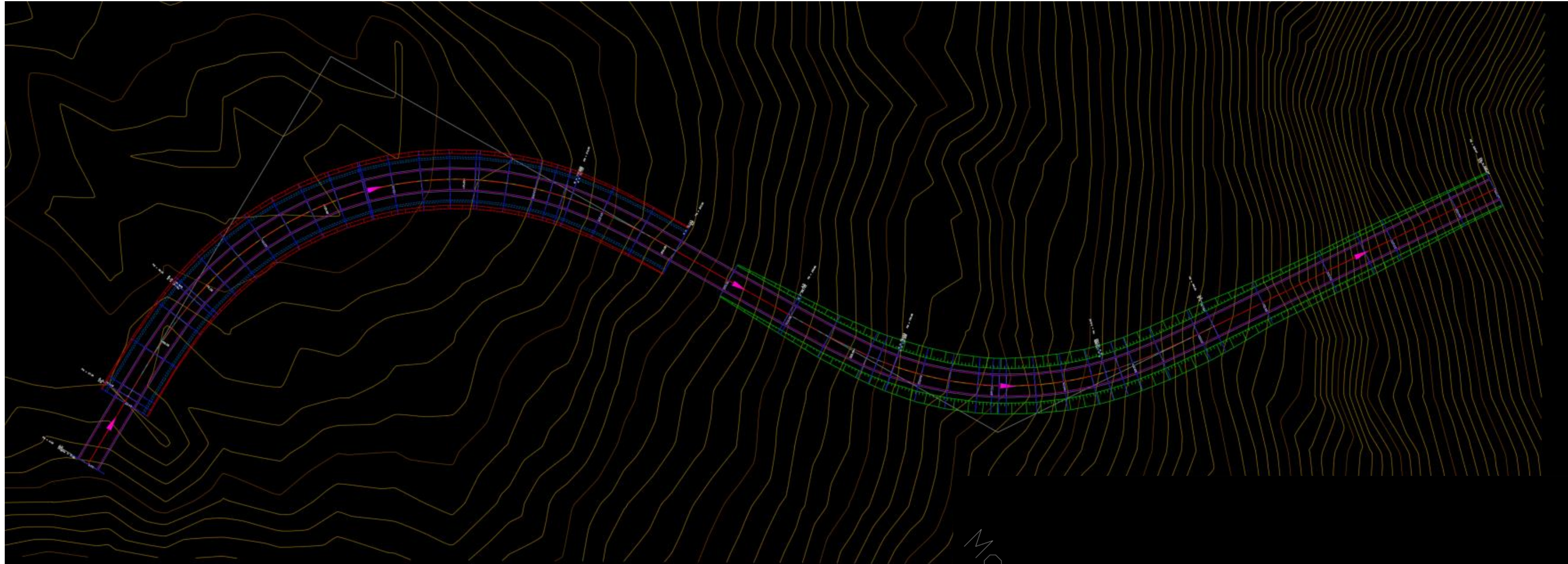
# Comenzar con un POC

- Realización de pruebas rápidamente :
  - ¿La lógica funciona?



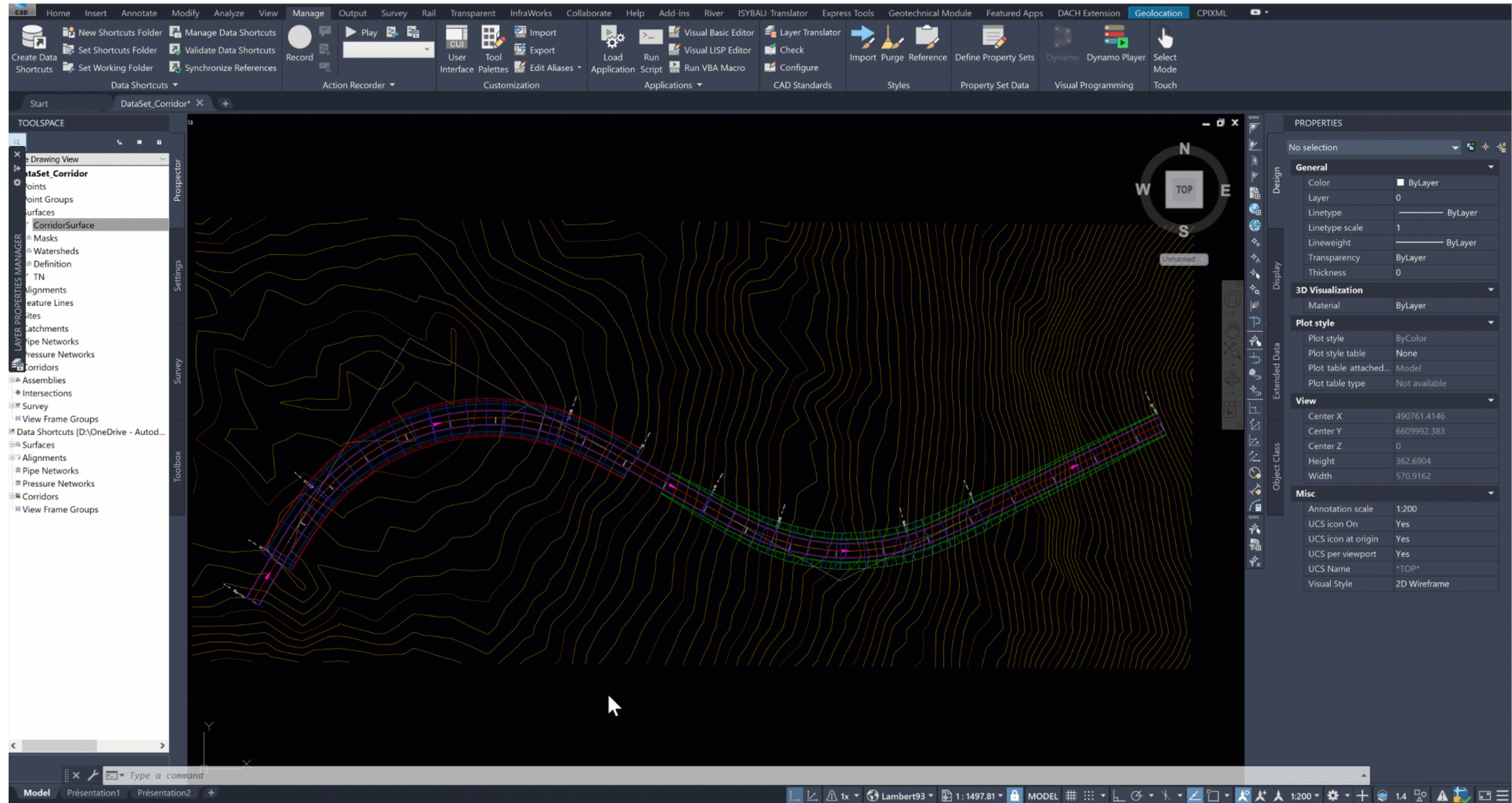


# Dataset neutro para realizar pruebas





# ¿Y en Dynamo?





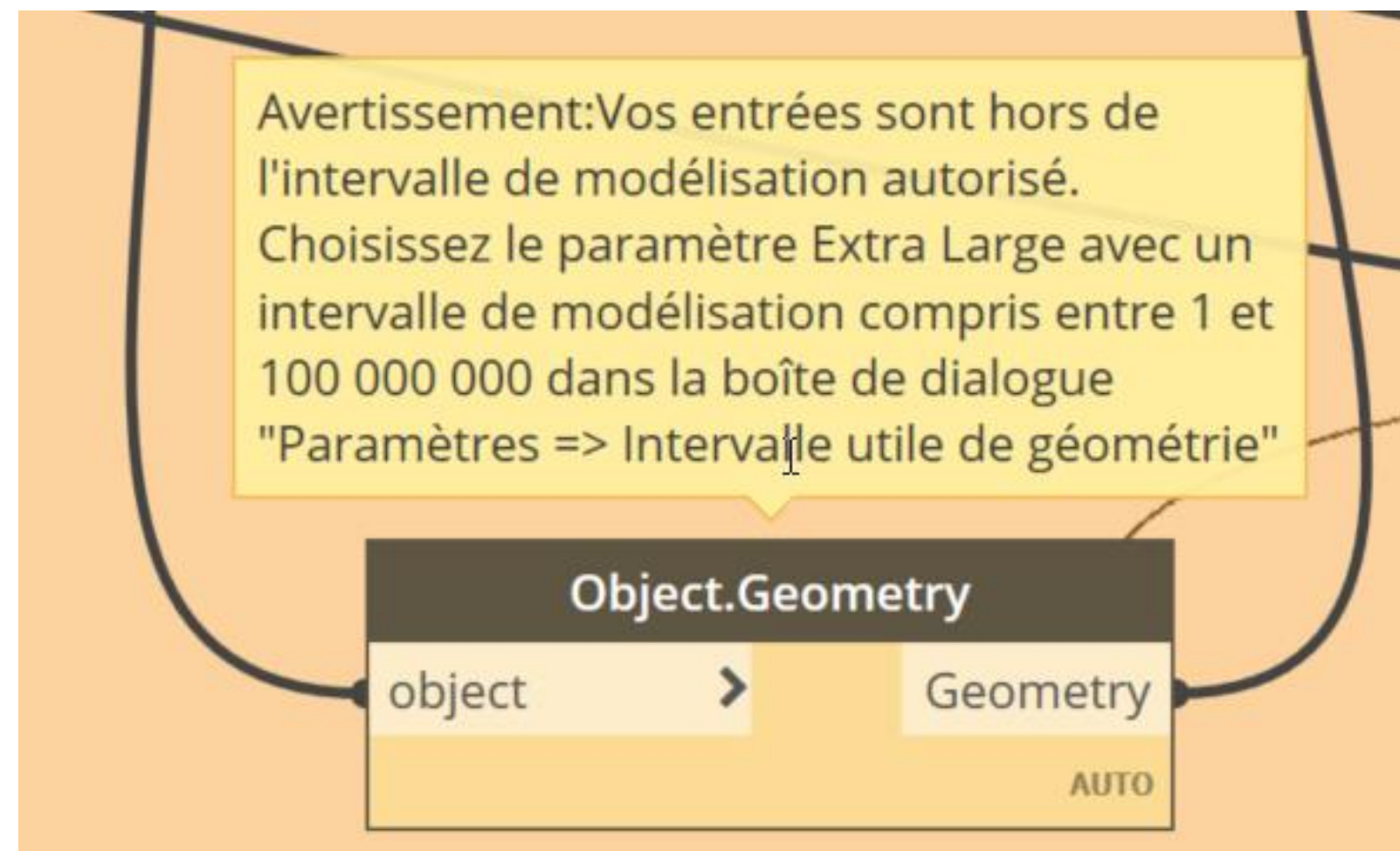
# Conceptos claves de Dynamo ...

A tener en cuenta para la construcción de gráficas



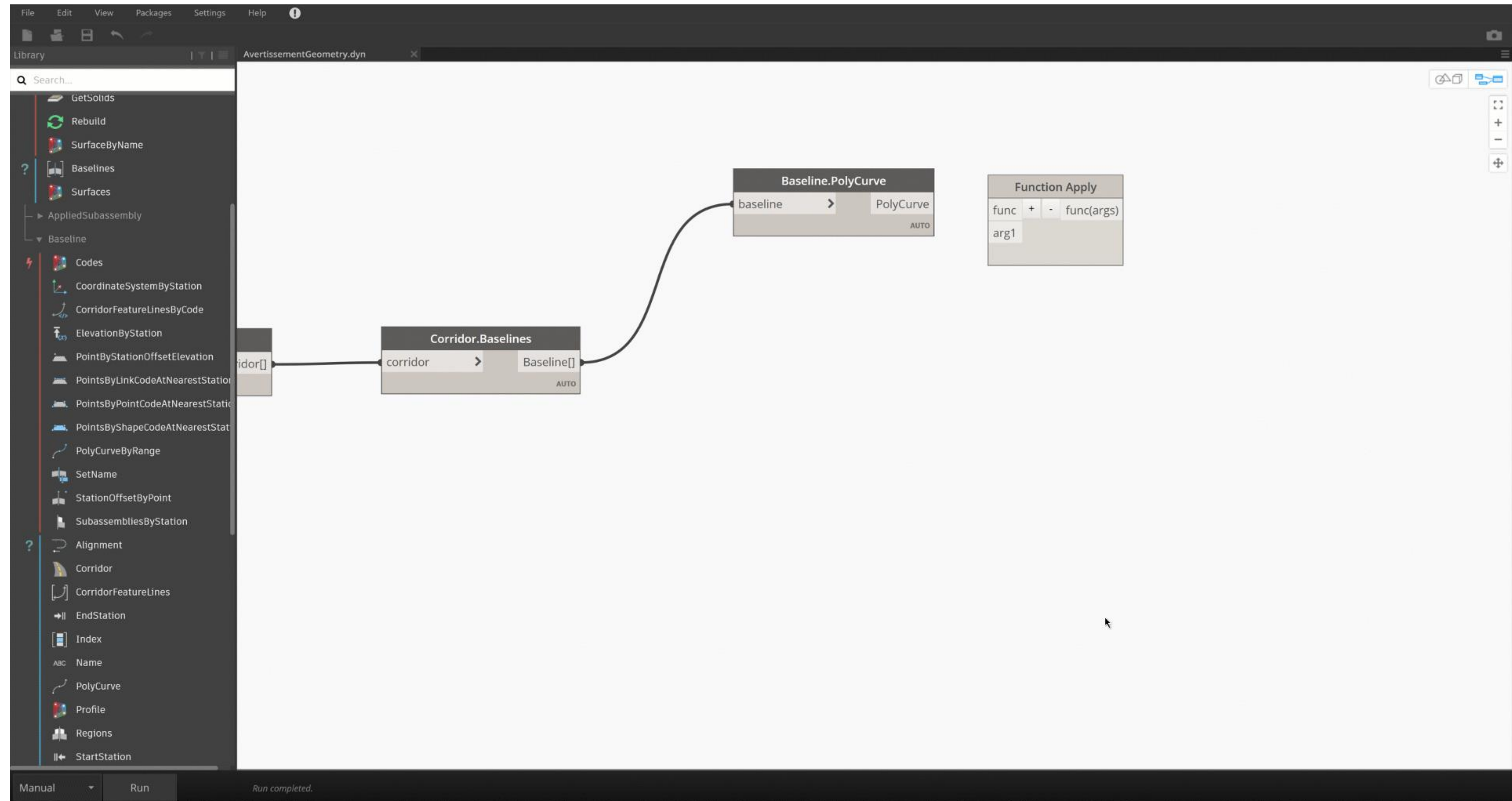
# Intervalo de geometría

- Advertencia muy común
- Aparece cuando estamos lejos del 0,0,0 en Dynamo (cada vez que tenemos un modelo geo referenciado)
- **NUNCA** cambiar el intervalo de geometría:
  - Aparecerán errores de modelización





# Intervalo de geometría





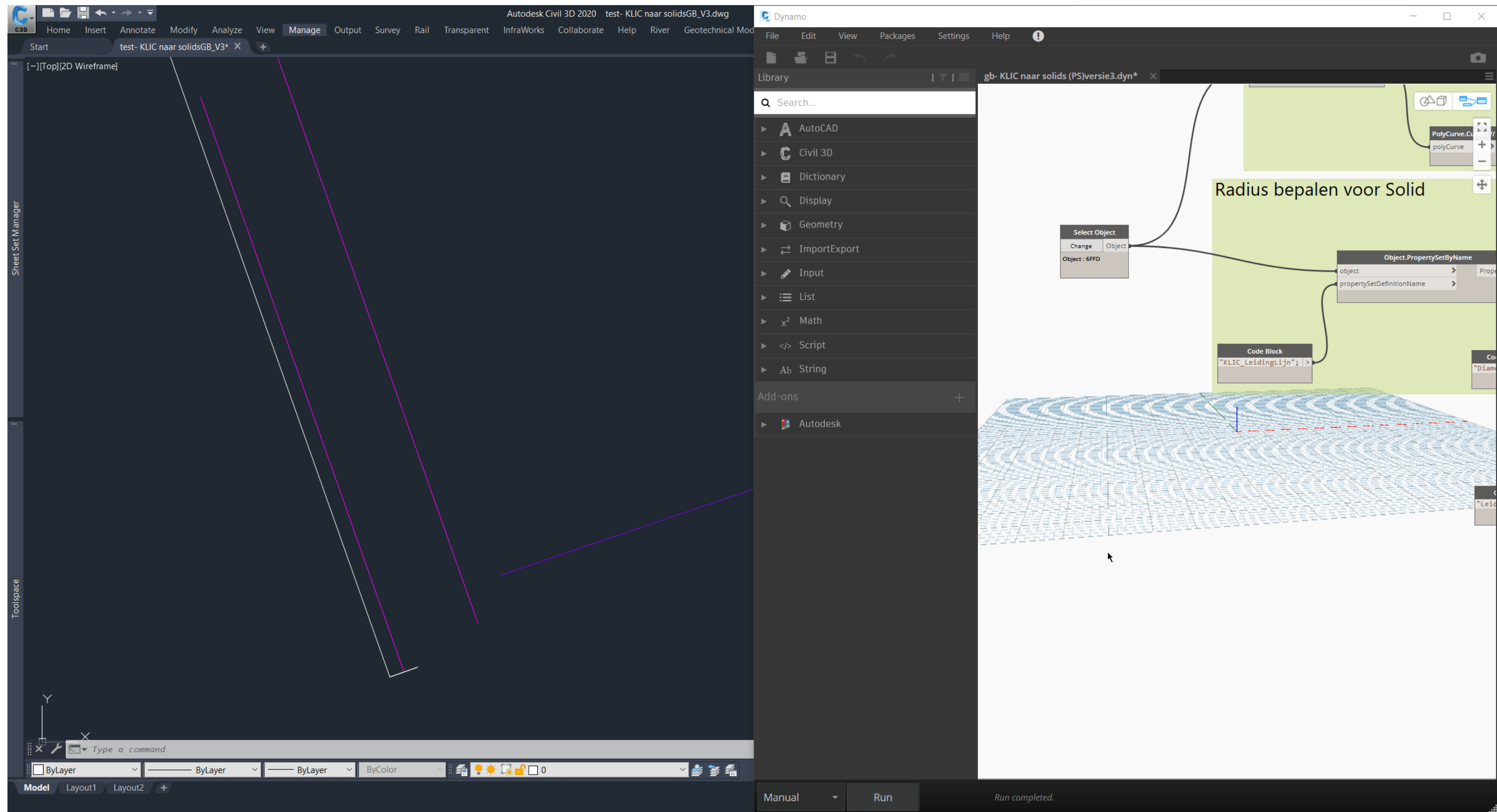
# « Binding »

- Muy importante a entender
- Gráfica atada con los elementos creados
- Problemático para la colocación de bloques



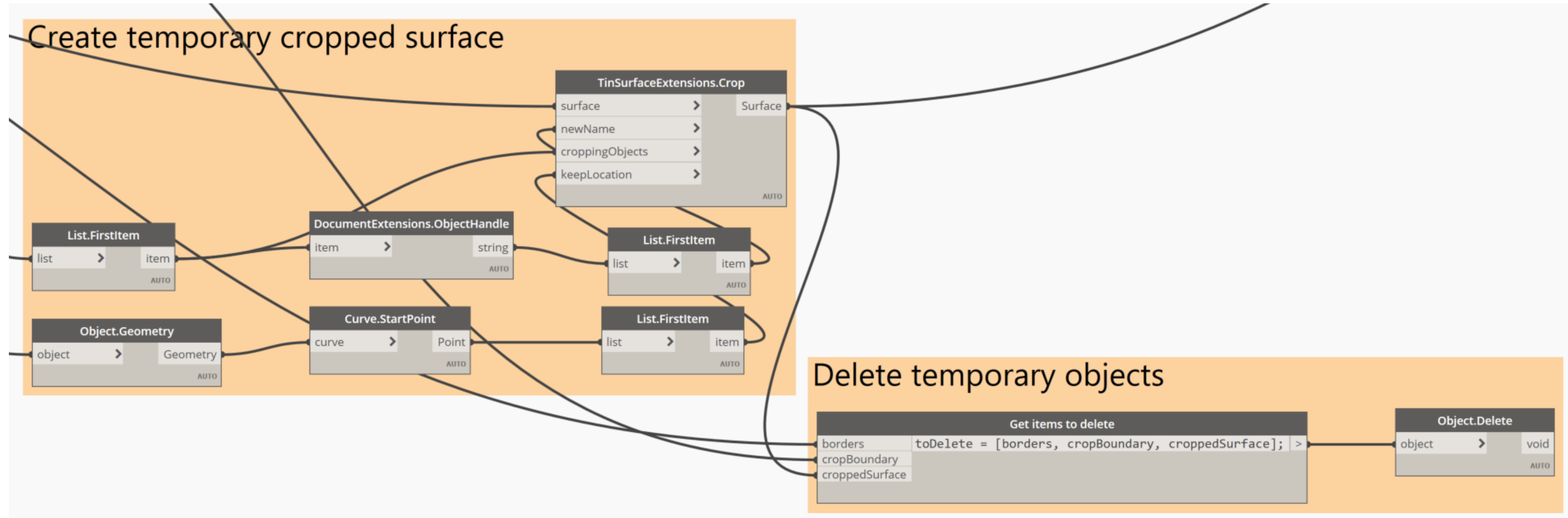


# Problemas con el binding





# Suprimir los objetos provisionales creados en Civil 3D



- Ciertos nodos o “acciones” solo utilizan objetos directos de AutoCAD (y no Dynamo)
- Sin repercusiones para el utilizador final



# Objetivos de aprendizaje



## IDENTIFICAR LAS OPORTUNIDADES DE AUTOMATIZACIÓN

- ✓ La automatización para realizar más, mejor, y con menos recursos
- ✓ Casos prácticos : tareas repetitivas, tareas con poco valor añadido, procesos iterativos, etc.
- ✓ Centrarse en los “quick wins”



## DESARROLLAR SOLUCIONES INDUSTRIALIZABLES

- Dynamo Player
- Instrucciones de utilización
- Soluciones modulares



## PENSAR ANALÍTICAMENTE

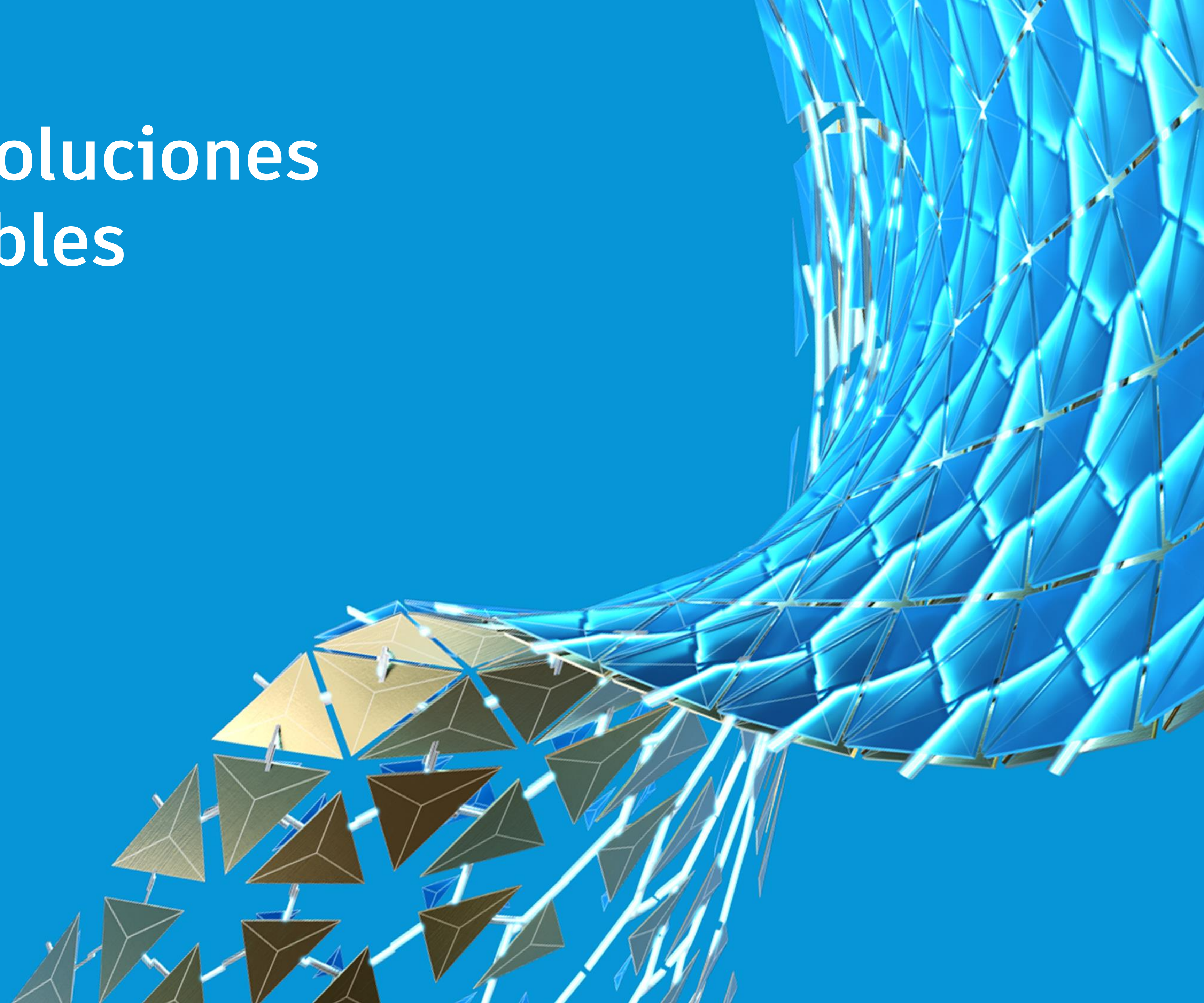
- ✓ Describir la secuencia lógica para resolver los problemas
- ✓ Crear un « POC » rápidamente, utilizar datasets neutros
- ✓ No cambiar el intervalo de geometría, tener en cuenta el binding y la utilización de objetos provisionales



## IMPLEMENTAR LA AUTOMATIZACIÓN



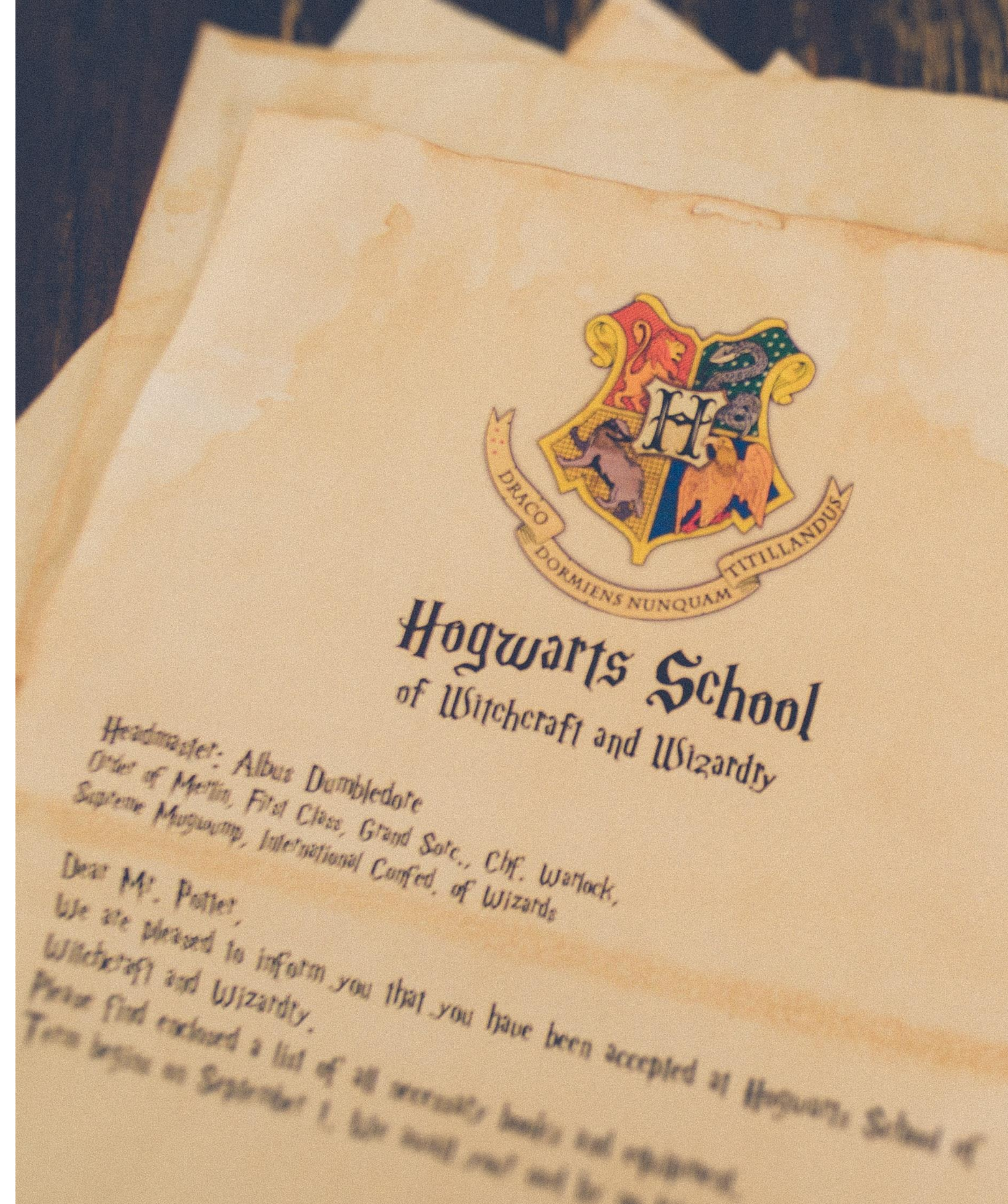
Desarrollar soluciones  
industrializables





# Dynamo Player

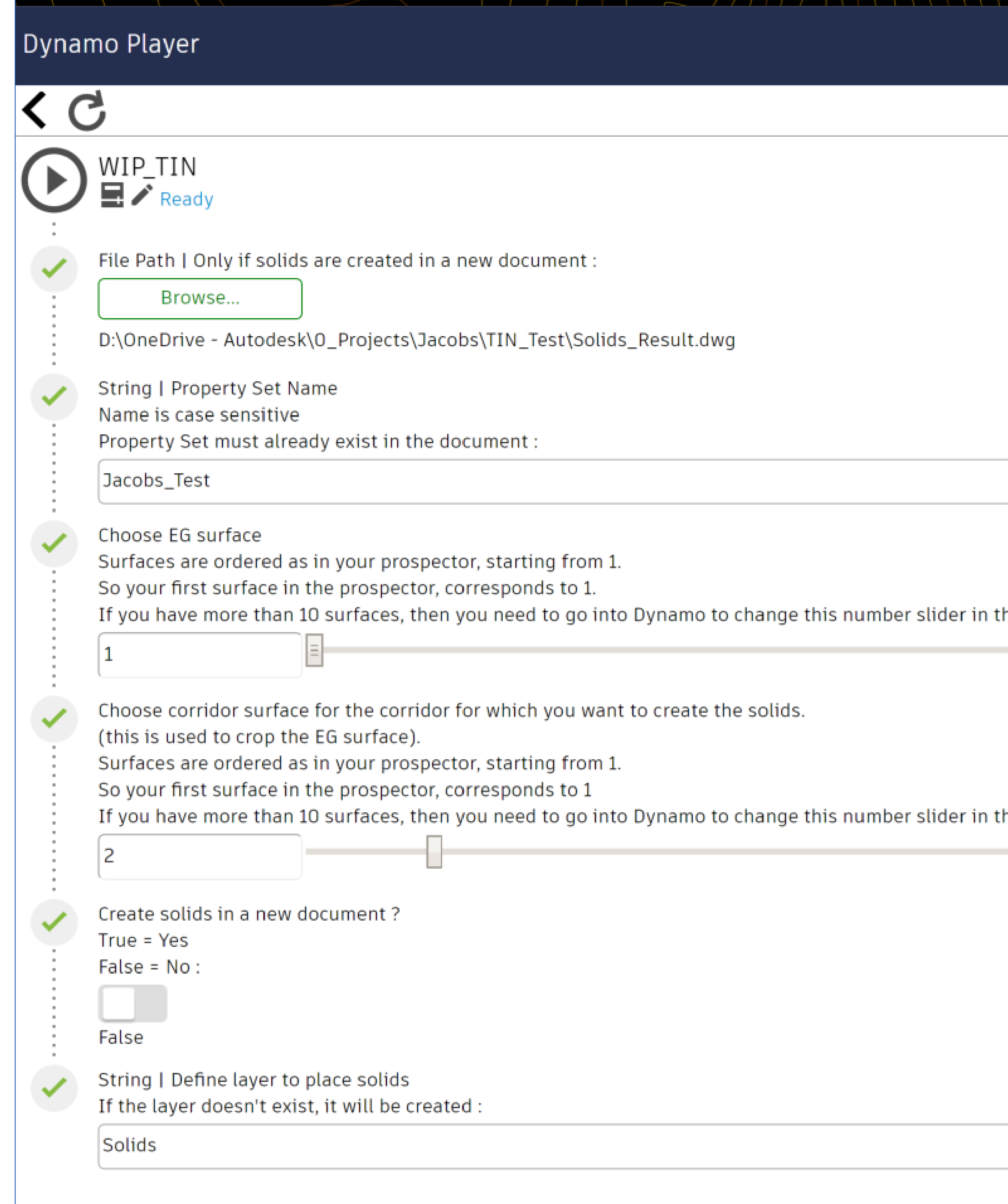
- Directamente desde Civil 3D
- Ninguna necesidad de entrar en Dynamo
- Con solamente unos clics / botón **mágico**





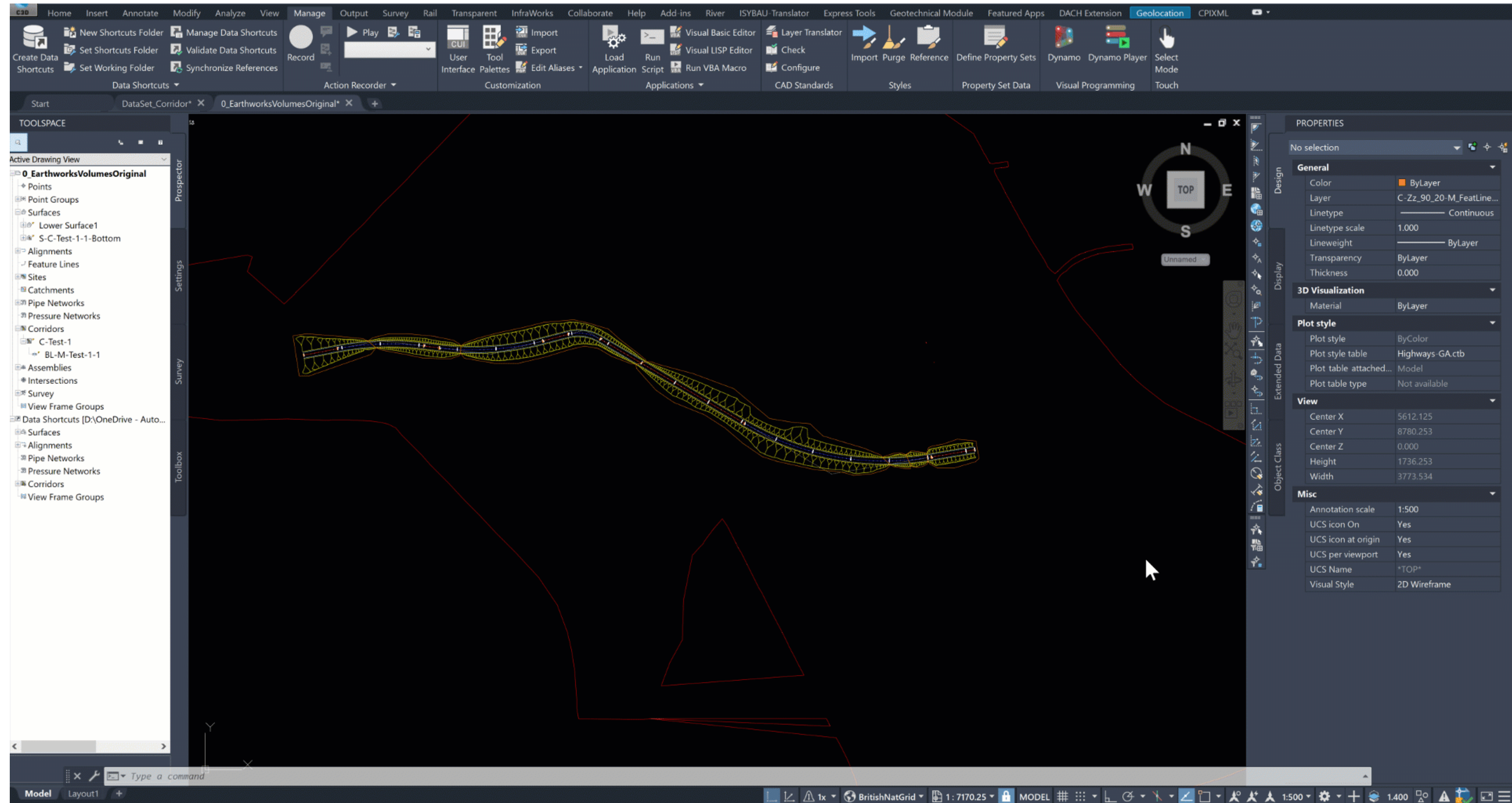
# Gráficas Paramétricas

- Dynamo Player permite de cambiar las entradas
- El utilizador puede probar diferentes configuraciones rápidamente:
  - Ejemplo : poste de luz todos los 20/25/30 m





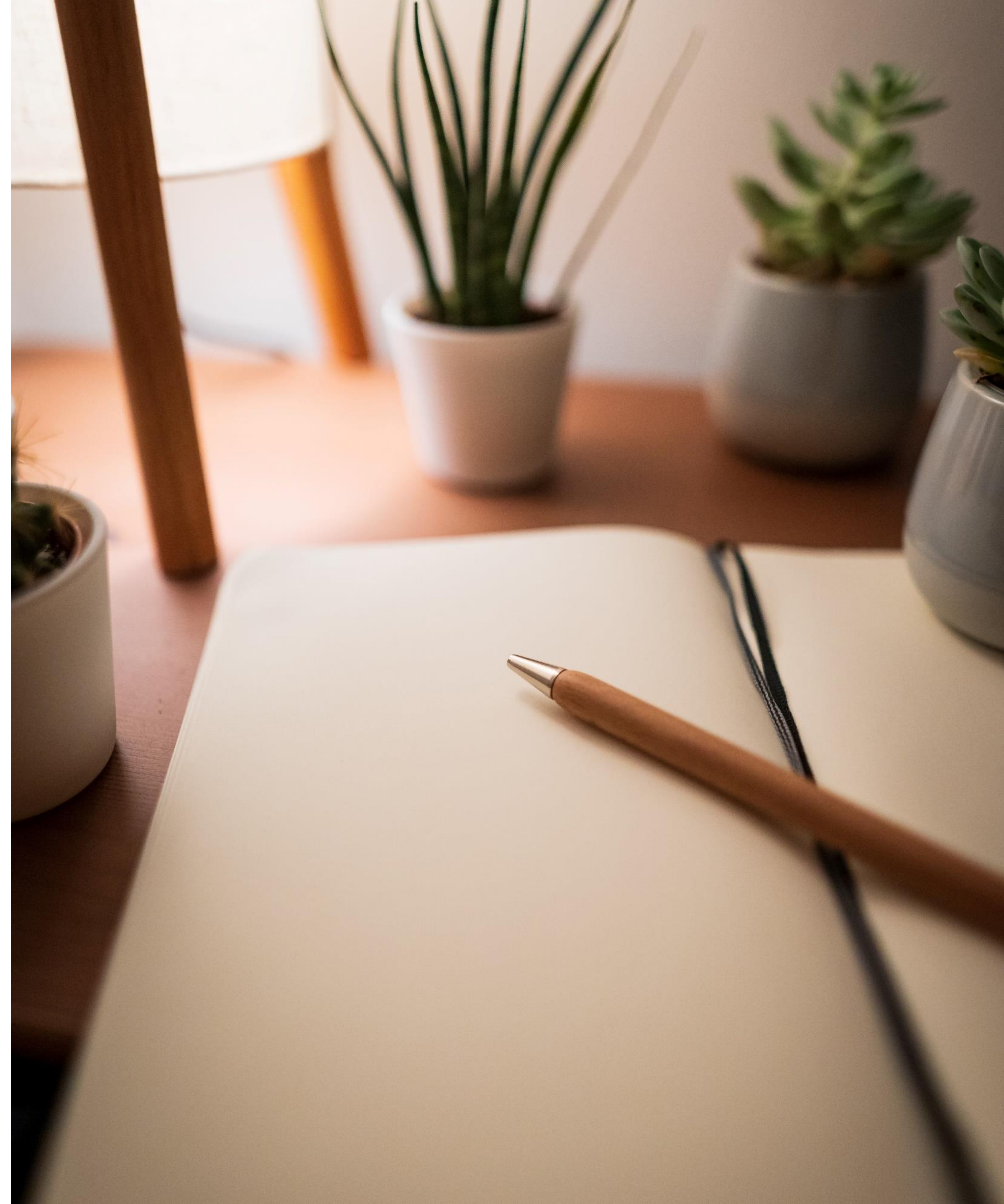
# ¿Y en Dynamo ?





# Instrucciones

- Instrucciones integradas en la gráfica :
  - El propósito de la gráfica
  - Suposiciones
  - Entradas
  - Posibles optimizaciones
  - Problemas identificados
  - Pruebas realizadas
  - ....





# Ejemplos de instrucciones

## NOTES

### GRAPH INFO

Copyright 2020 Autodesk, Inc. All rights reserved.  
Company: Jacobs  
Office: <office>

Version: 1.0.0  
Author: David Licona  
david.licona@autodesk.com

Keywords: Earthworks Solids,

Tested on:  
Dynamo: 2.4.1  
Civil 3D: 2020.4  
Civil 3D Toolkit Package : 1.1.9

### Useful Links

Your code goes here

### INSTRUCTIONS

The purpose of this graph is to create earthwork solids.

To achieve this goal, the Dynamo graph is executing the following operations :

- Crop existing ground surface using the corridor surface as the cropping limit. This is due to a limitation of Civil 3D: being able to create solids between tin surfaces made up of less than 100K triangles.

- Create earthwork solids. As with the Civil 3D UI, the solids are created between two tin surfaces:
  - o The corridor surface,
  - o The existing ground surface (bottom surface).

Two solids will be initially created, one for the fill and one for the cut. Tests show that the first solid in the resulting list is always the fill solid (more tests may be required as this was tested in 3 different documents only).

- Split the solids to have separate solids for the two sides of the carriageway. Solids are split along the corridor baseline.

- The resulting fill and cut solids are sent into Civil 3D on user-defined layers

#### User inputs

- Choose parametrically the existing ground surface (surfaces are sorted alphabetically as in the prospector – index 1 corresponds to the first surface in alphabetical order) [slider],

- Choose parametrically the corridor surface (surfaces are sorted alphabetically as in the prospector – index 1 corresponds to the first surface in alphabetical order) [slider] ,

- Choose parametrically the corridor (corridors are

### KNOWN ISSUES AND LIMITATIONS

The following issues have been identified:

- Solids will not be created if any surface exceeds 100K triangles
  - o The graph will crop the surface using the corridor surface as the object to crop

- Cropping will only work if the corridor surface style is set to "\_No Display". It may occur that depending on the country kit used, the graph will not work also with the "no display" style. Tests were done with Client provided datasets and no issues were found.

- If the corridor is long enough, it may happen that the surface created from the crop exceeds the 100K triangles, today the graph will not be able to handle this

- The graph will not select corridors who have a feature line as a baseline

- In a file with multiple corridors, only one corridor can be created at a time

- Created solids will only be split by type (cut or fill) and by the baseline. This means that solids may consist of different earthworks.

- Solids are created by lofting the EG surface and the corridor surface, this means that solids will be created taking into account every point (on the X and Y plane) where the corridor surface intersects the EG, even when variations are really small.

- Errors may occur because of the offset distance used to create the cropping object and based on the corridor surface. If this happens, try lowering the value of the Offset (for cropping purposes) node in the Input section

### GUIDELINES

Read the instructions.

Add Notes and Comments to the graph.

Use Node Groups and the Standard Color Coding.

Rename Nodes: <OriginalName> | <Description>.

Write Input and Output Notes for Python Scripts.

Prefer repeatable simple node structures.

Simple is better than complex.

Complex is better than complicated.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, do not guess.

There should be one obvious way to do it.

Now is better than never.

Although never is often better than \*right\* now.



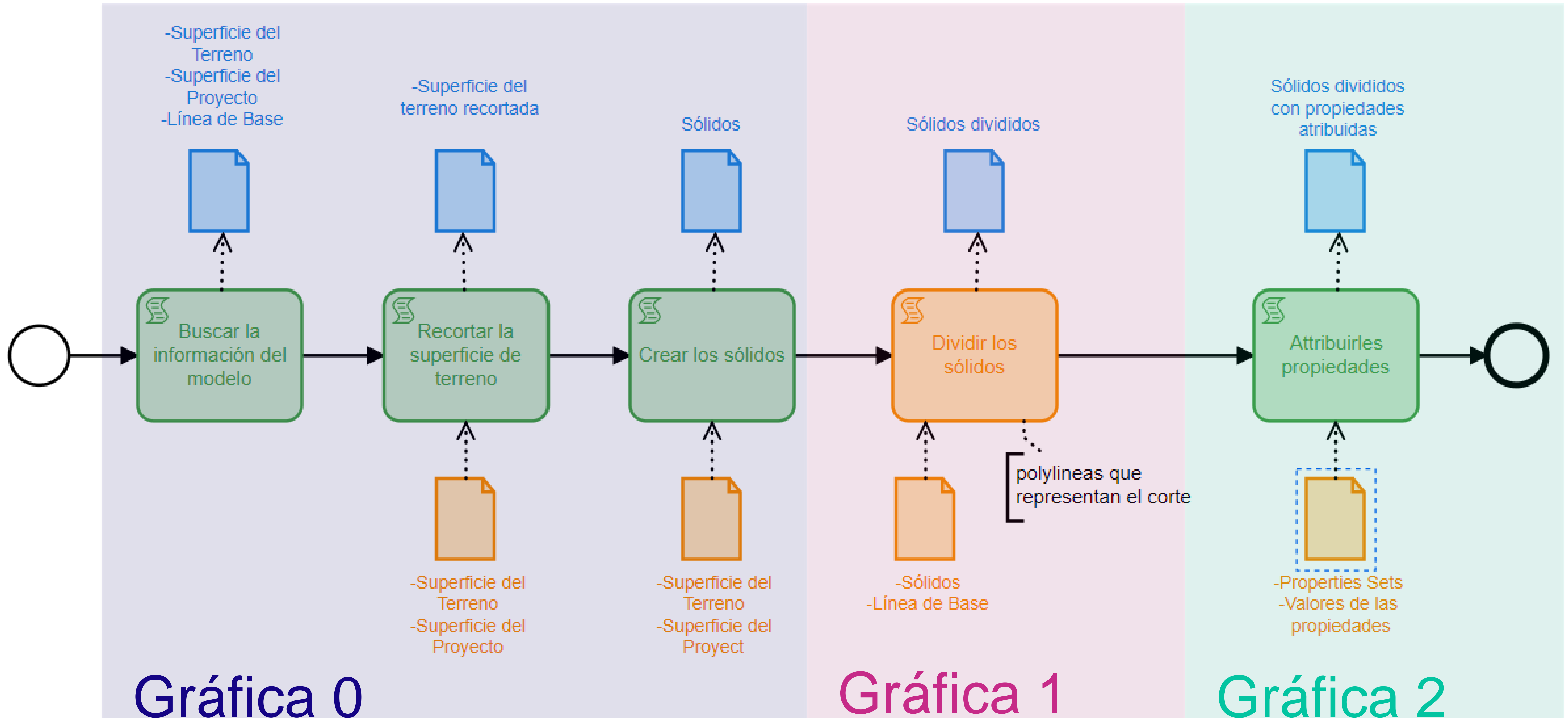
# Modularizar

- Realización de pruebas más fáciles
- Más fácil de entender
- Más fácil de encontrar fallos



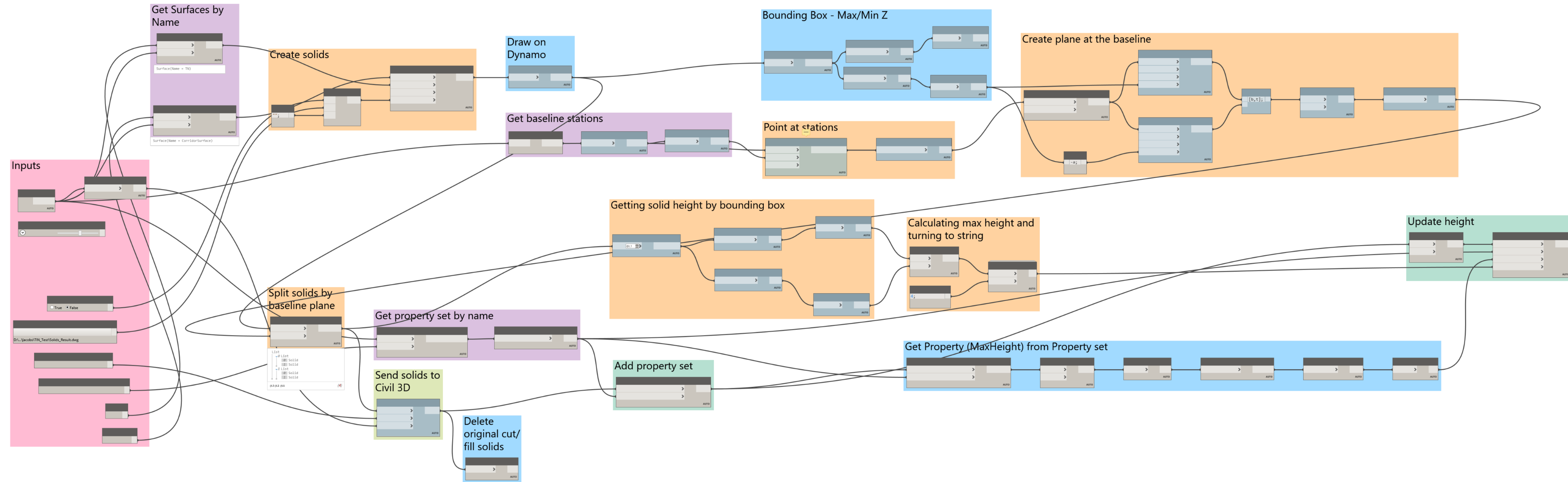


# Resultado final



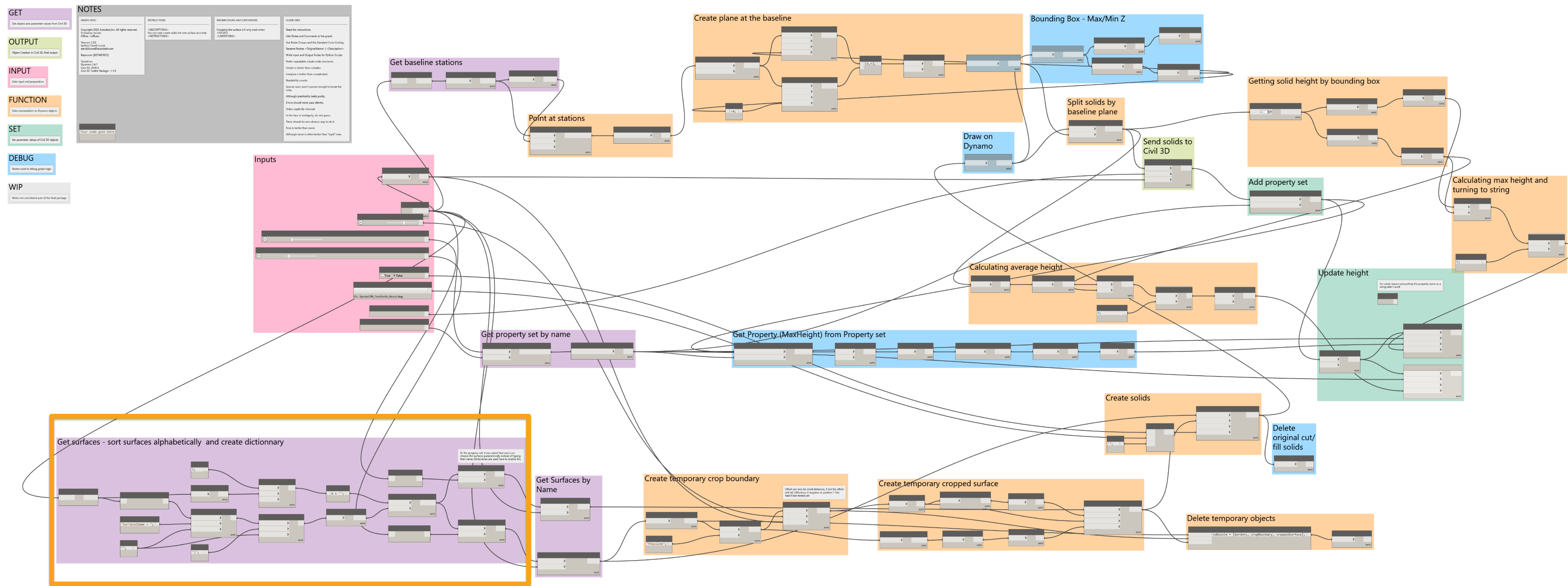


# POC





# Paramétrica





# Gráfica 0: Crear sólidos

- GET
- Get objects and parameter values from Civil 3D
- OUTPUT
- Object Creation in Civil 3D, final output
- INPUT
- Data input and preparation
- FUNCTION
- Data manipulation on Dynamo objects
- SET
- Set parameter values of Civil 3D objects
- DEBUG
- Nodes used to debug graph logic
- WIP
- Notes not considered part of the final package

NOTES

GRAPH INFO

Copyright 2020 Autodesk, Inc. All rights reserved.  
Company: Jacobs  
Office: Valencia  
  
Version: 1.0.0  
Author: David Llorca  
david.llibre@autodesk.com  
  
Keywords: Earthworks Solids,  
  
Tested on:  
Dynamo 3.4.1  
Civil 3D 2024A  
Civil 3D Toolset Package 1.1.8  
  
Your code goes here

INSTRUCTIONS

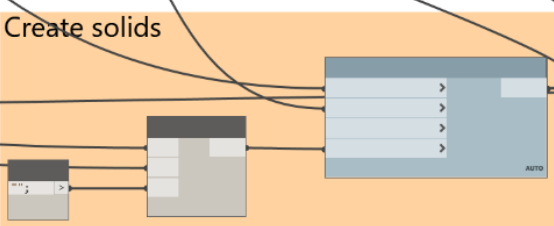
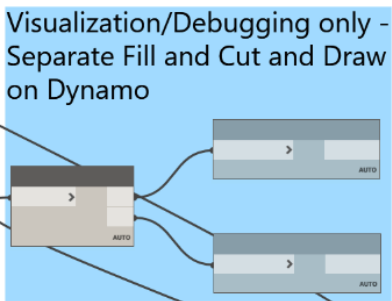
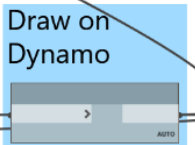
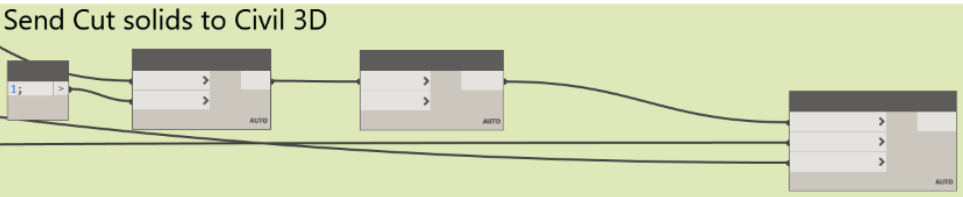
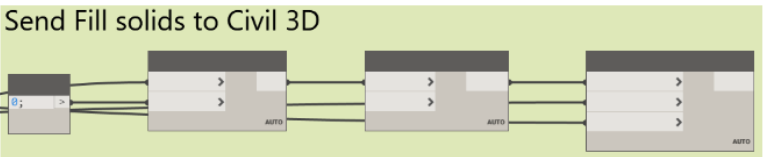
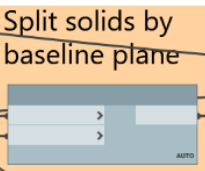
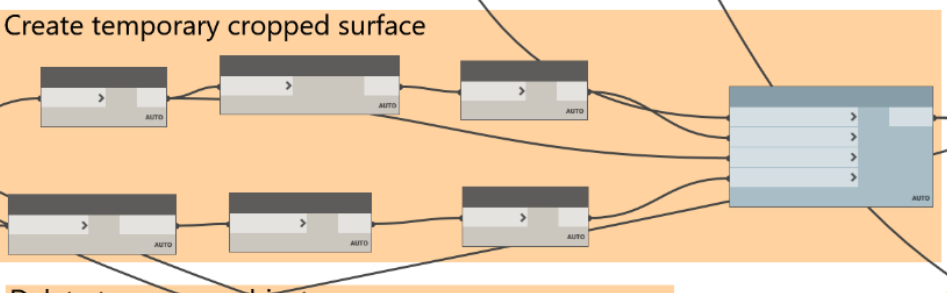
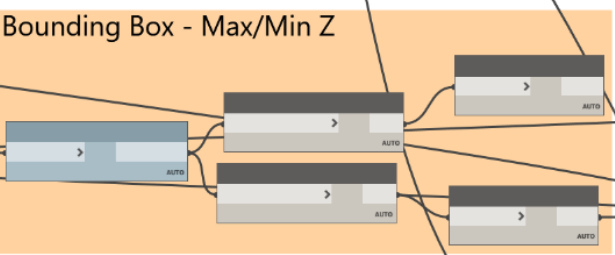
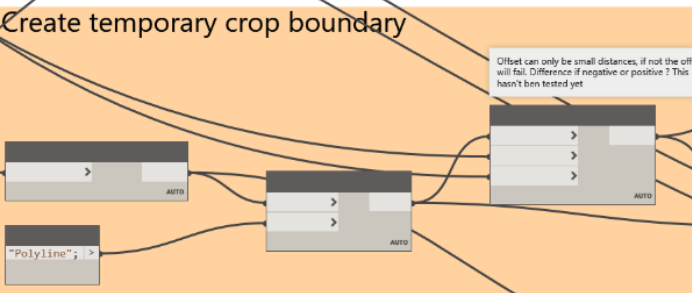
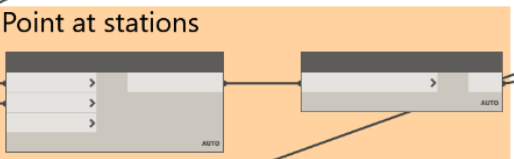
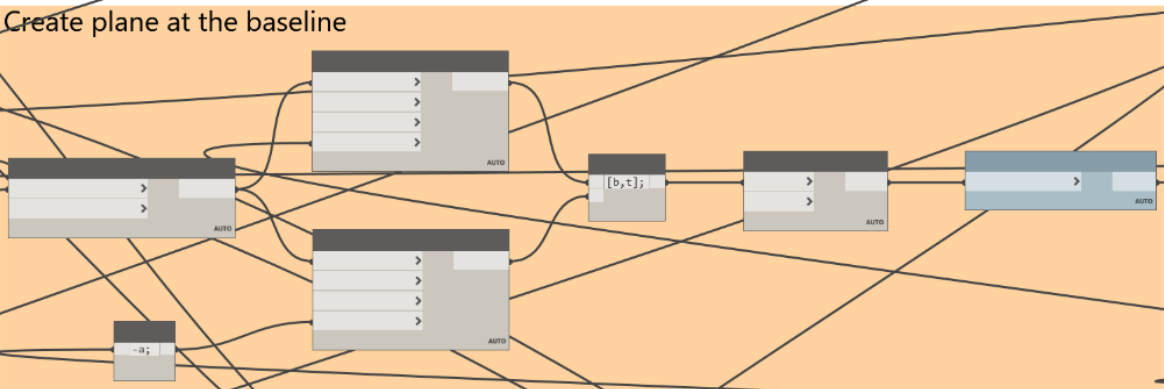
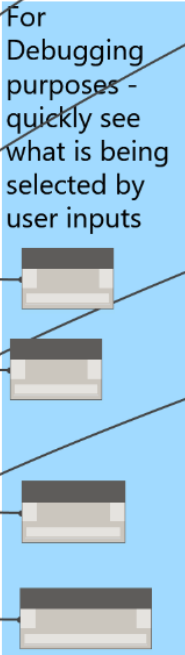
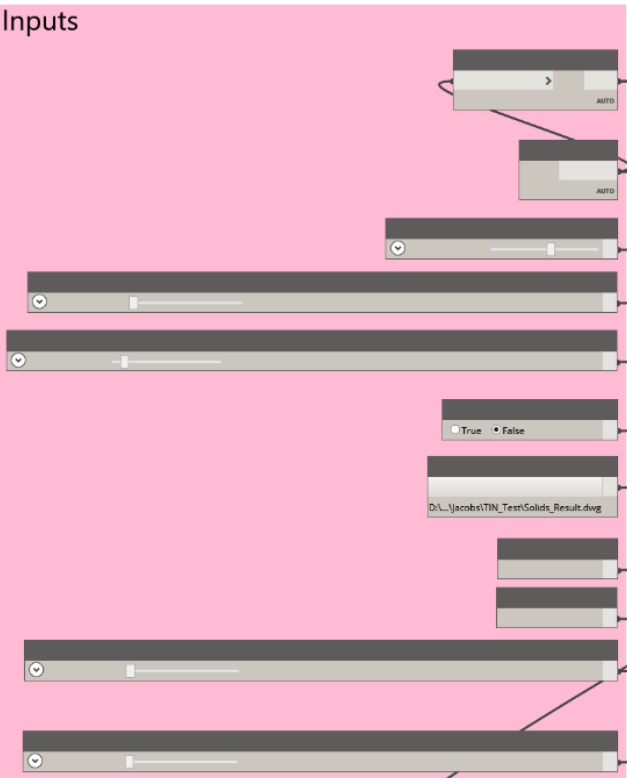
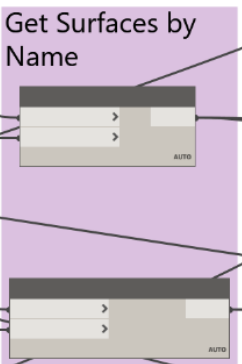
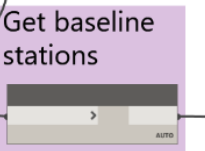
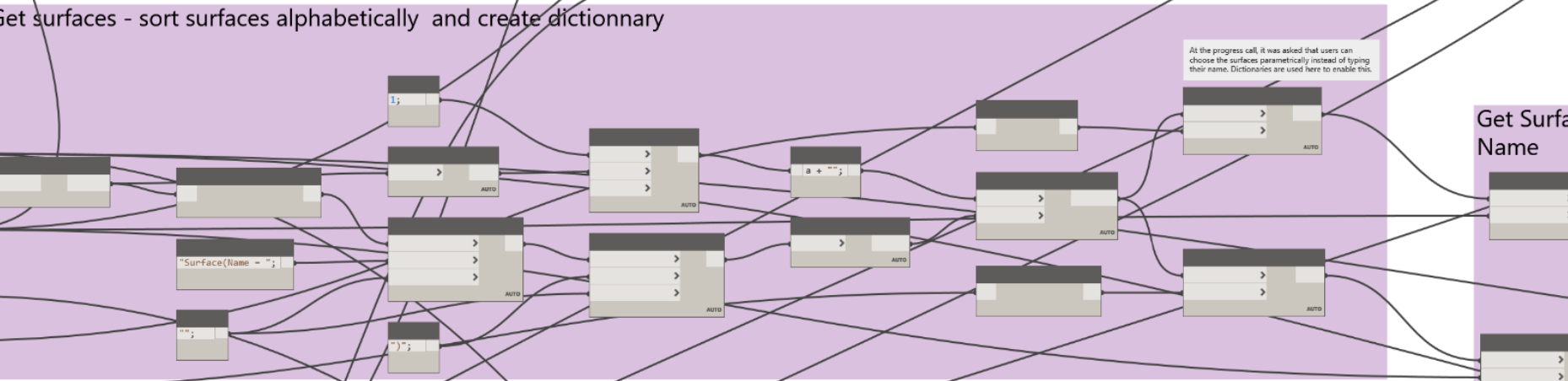
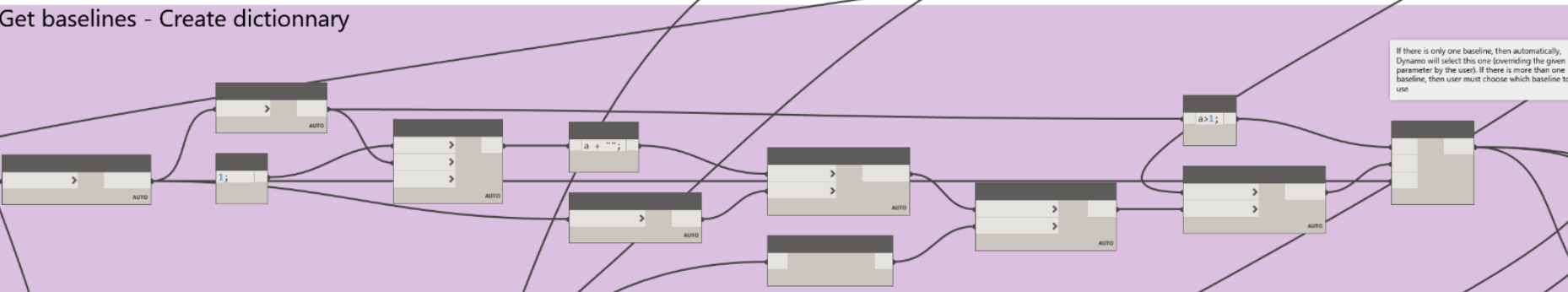
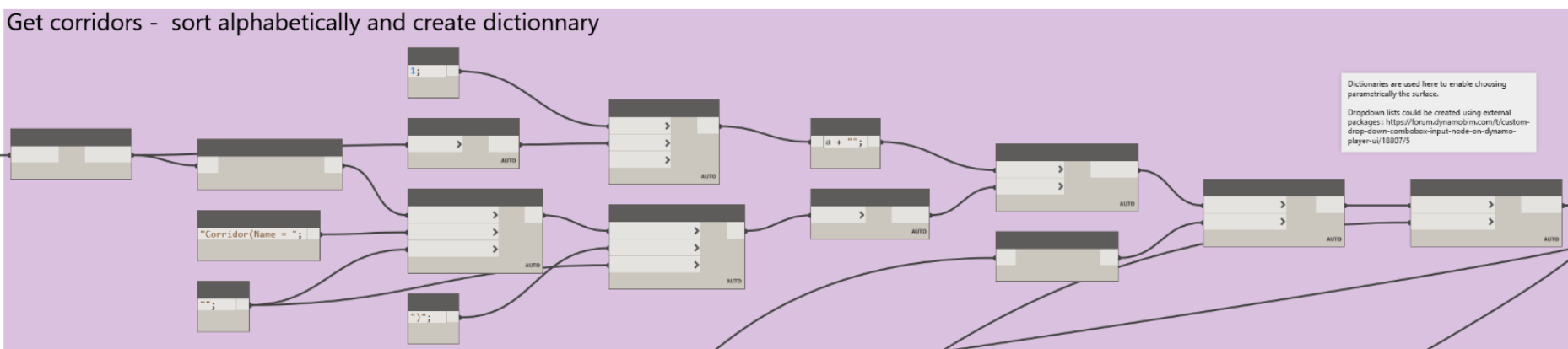
The purpose of this graph is to create earthwork solids.  
  
To achieve this goal, the Dynamo graph is executing the following operation:  
  
-Crop existing ground surface using the corridor surface as the cropping link. This is due to a limitation of Civil 3D, being able to create solids between two surfaces made up of less than 100K triangles.  
  
-Create earthwork solids. As with the Civil 3D UI, the solids are created between two surfaces:  
o The corridor surface  
o The existing ground surface (bottom surface).  
  
Two solids will be initially created, one for the fill and one for the cut. Tests show that the first solid in the resulting list always the fill solid (even tools may be required as this was locked in 3 different documents only).  
  
Split the solids to have separate solids for the two sides of the carriageway. Solids are split along the corridor baseline.  
  
The resulting fill and cut solids are sent into Civil 3D as user-defined types.  
  
User inputs:  
  
-Choose parametrically the existing ground surface (surface or corridor) alphabetically as in the prospector - index 1 corresponds to the first surface in alphabetical order (solid).  
  
-Choose parametrically the corridor surface (surface or corridor) alphabetically as in the prospector - index 1 corresponds to the first surface in alphabetical order (solid).  
  
-Choose parametrically the baseline (baseline) will be shown as created - normally, the fill baseline will correspond to the middle of the carriageway (solid).  
  
-Defining the layer on which the fill solid objects will be created (text).  
  
-Defining the layer on which the cut solid objects will be created (text).  
  
-Choose if the solids are to be created in a new document or not (Boolean).  
o Give the file path to the new document (file path).  
  
Output nodes on the player  
  
Watch nodes are provided on the player to indicate to the user the chosen surfaces, corridor and baseline

KNOWN ISSUES AND LIMITATIONS

The following issues have been identified:  
  
-Solids will not be created if any surface exceeds 100K triangles.  
o The graph will crop the surface using the corridor surface as the object to crop.  
  
-Cropping will only work if the corridor surface style is set to "No Display". It may occur that depending on the styling link used, the graph will not work with the "No display" style. Tests were done with Client provided datasets and no issues were found.  
  
-If the corridor is long enough, it may happen that the surface created from the crop exceeds the 100K triangles, today the graph will not be able to handle this.  
  
-The graph will not select corridors who have a feature line as a baseline.  
  
-Is a file with multiple corridors, only one corridor can be created at a time.  
  
-Created solids will only be split by type (cut or fill) and by the baseline. This means that solids may consist of different earthworks.  
  
-Solids are created by lifting the EG surface and the corridor surface, this means that solids will be created taking into account every point (on the X and Y plane) where the corridor surface intersects the EG, even when variations are really small.  
  
-Errors may occur because of the offset distance used to create the cropping object and based on the value of the Offset for cropping (reported node in the input section).  
  
-Solids may occur because of the offset distance used to create the cropping object and based on the value of the Offset for cropping (reported node in the input section).

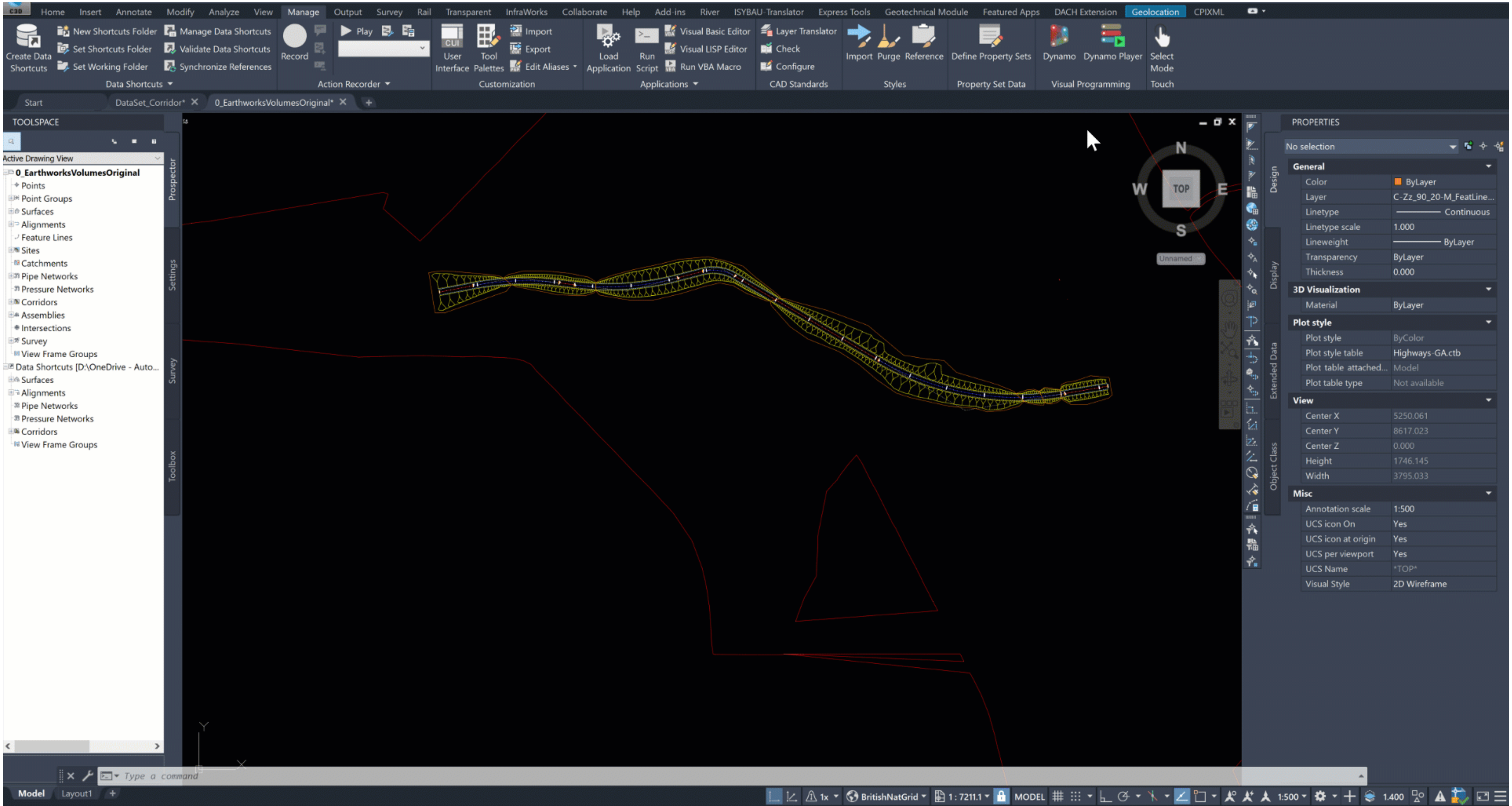
GUIDELINES

Read the instructions.  
  
Add Notes and Comments to the graph.  
  
Use Node Groups and the Standard Color Coding.  
  
Rename Nodes: <OriginalName> - <Description>.  
  
Write Input and Output Notes for Python Scripts.  
  
Prefer repeatable simple node structures.  
  
Simple is better than complex.  
  
Complex is better than complicated.  
  
Readability counts.  
  
Special cases aren't special enough to break the rules.  
  
Although practically beats purity.  
  
Errors should never pass identity.  
  
Unless explicitly allowed.  
  
In the face of ambiguity, do not guess.  
  
There should be one obvious way to do it.  
  
Now is better than never.  
  
Although never is often better than "right" now.





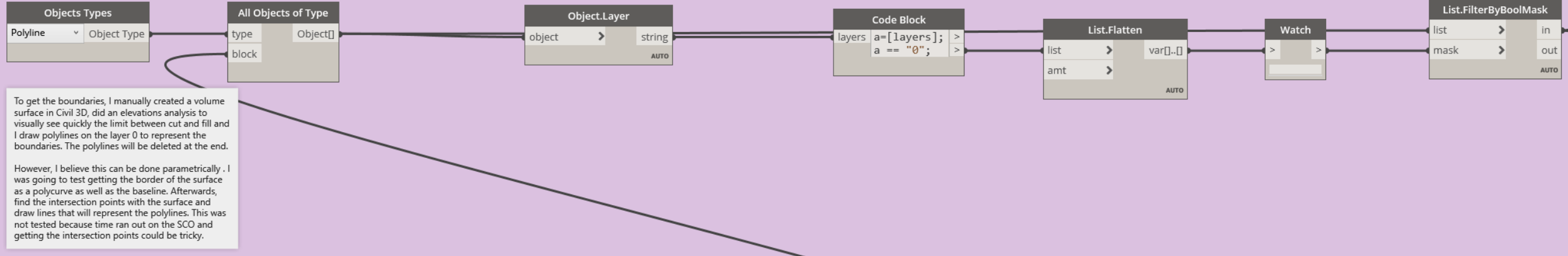
# Gráfica 0 : Crear sólidos



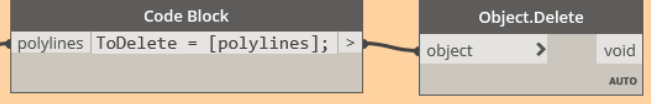


# Gráfica 1: Dividir los sólidos

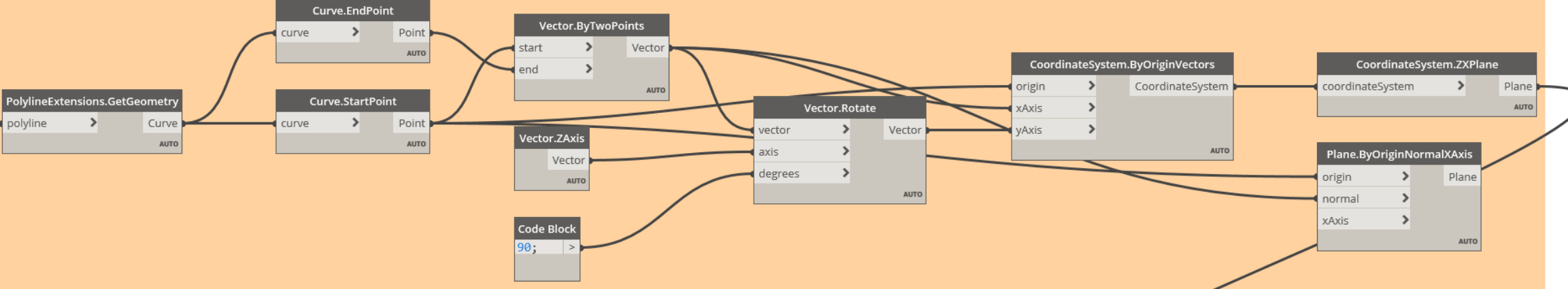
## Get boundaries between earthworks



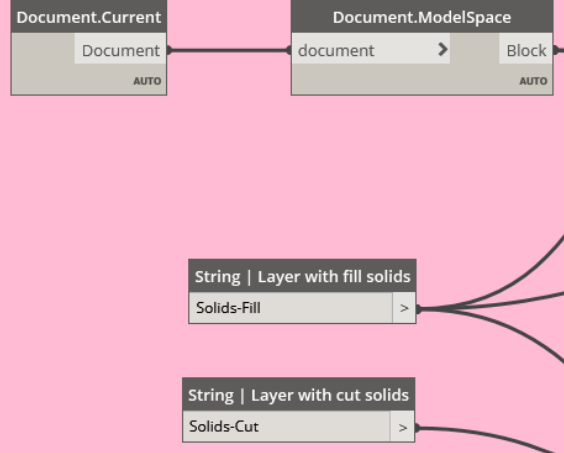
## Delete polylines



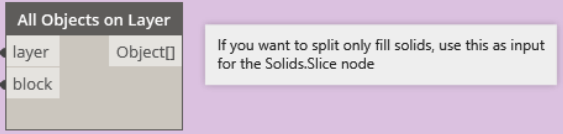
## Getting ZX plane of polylines



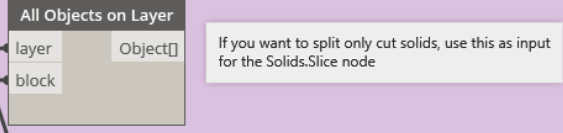
## Inputs



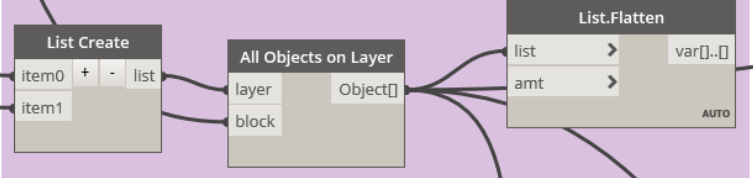
## Get Fill solids



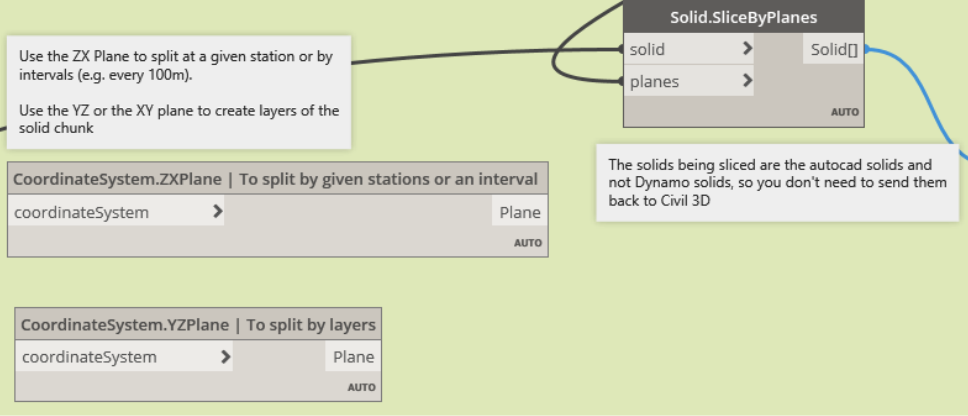
## Get Cut solids



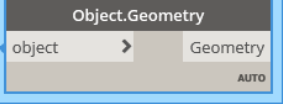
## Get all solids



## Slicing solids by planes



## Draw Geometry on Dynamo



## GET

Get objects and parameter values from Civil 3D

## OUTPUT

Object Creation in Civil 3D, final output

## INPUT

Data input and preparation

## FUNCTION

Data manipulation on Dynamo objects

## SET

Set parameter values of Civil 3D objects

## DEBUG

Nodes used to debug graph logic

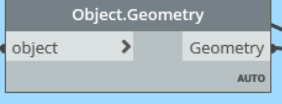
## WIP

Notes not considered part of the final package

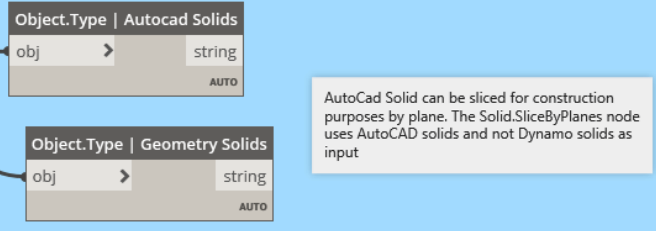
## NOTES

GRAPH INFO	INSTRUCTIONS	KNOWN ISSUES AND LIMITATIONS	GUIDELINES
<p>Copyright 2020 Autodesk, Inc. All rights reserved. Company: Jacobs Office: &lt;office&gt; Version: 1.0.0 Author: David Licona david.licona@autodesk.com Keywords: [KEYWORDS] Tested on: Dynamo: 2.4.1 Civil 3D: 2020.4 Civil 3D Toolkit Package: 1.1.9</p>	<p>The purpose of this graph is to manipulate earthwork solids created by the Get Solids Graph.</p> <p>The following manipulation is being done:</p> <p>Slicing solids by earth work.</p> <p>To achieve this goal, the Dynamo graph is executing the following operations:</p> <ul style="list-style-type: none"><li>-Looking for all the polylines in the graph</li><li>-Filtering the polylines in order to get only polylines on layer 0</li><li>-Determining the ZX plane of this polylines</li><li>-Selecting solids from fill and cut layers</li><li>-Slicing the selected solids by using the ZX plane of the polylines</li><li>-Deleting the polylines used for slicing (polylines on layer 0). If you don't want to delete the polylines used for the slicing, then freeze the delete node</li></ul> <p>User inputs</p> <ul style="list-style-type: none"><li>-User must create manually the polylines that indicate where the slicing is going to occur. These polylines must be on layer 0</li><li>-Defining the layer on which the fill solid objects are [text]</li><li>-Defining the layer on which the cut solid objects are [text]</li></ul> <p>Notes are provided along the groups to clarify or explain nodes used as well as possibilities.</p>	<ul style="list-style-type: none"><li>-For the moment, polylines will be needed to be created manually. The graph needs to be modified in such a way to identify the earthwork change.</li><li>-Graph will only work with polylines on layer 0</li><li>-Only run the graph once in a file. If the graph is run more than one time on the same file, it will delete previously created solids.</li></ul>	<p>Read the instructions.</p> <p>Add Notes and Comments to the graph.</p> <p>Use Node Groups and the Standard Color Coding.</p> <p>Rename Nodes: &lt;OriginalName&gt;   &lt;Description&gt;.</p> <p>Write Input and Output Notes for Python Scripts.</p> <p>Prefer repeatable simple node structures.</p> <p>Simple is better than complex.</p> <p>Complex is better than complicated.</p> <p>Readability counts.</p> <p>Special cases aren't special enough to break the rules.</p> <p>Although practicality beats purity.</p> <p>Errors should never pass silently.</p> <p>Unless explicitly silenced.</p> <p>In the face of ambiguity, do not guess.</p> <p>There should be one obvious way to do it.</p> <p>Now is better than never.</p> <p>Although never is often better than "right" now.</p>

## Draw on Dynamo

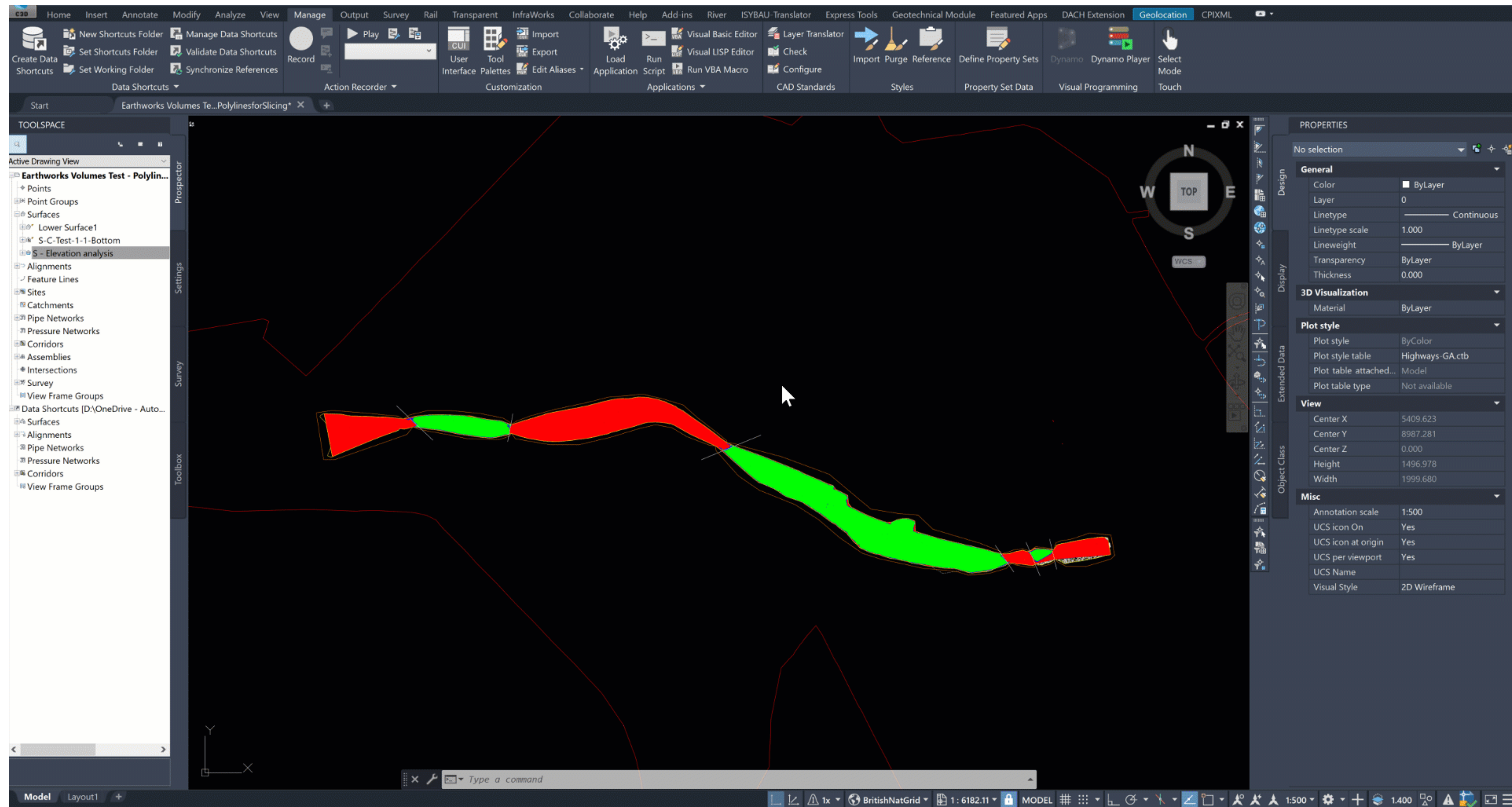


## For context > difference between ACAD solids and Geometry solids



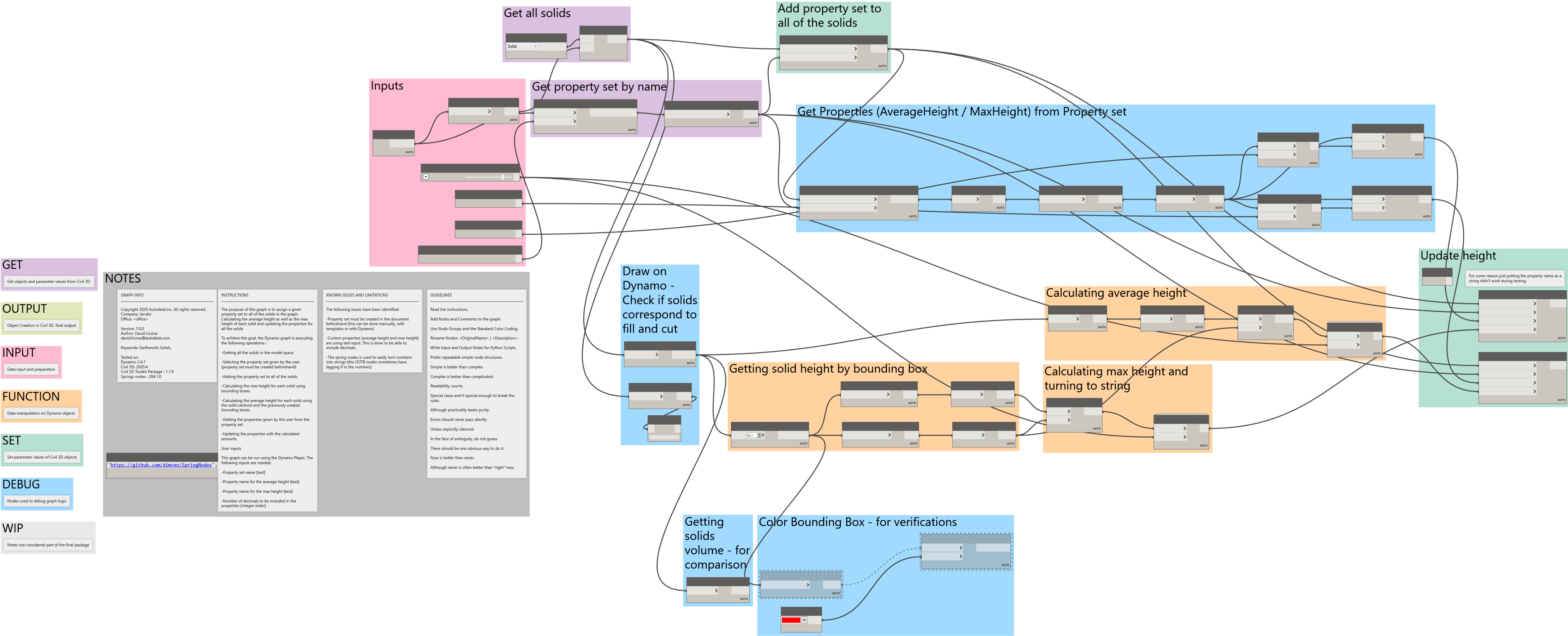


# Gráfica 1 : Dividir los sólidos



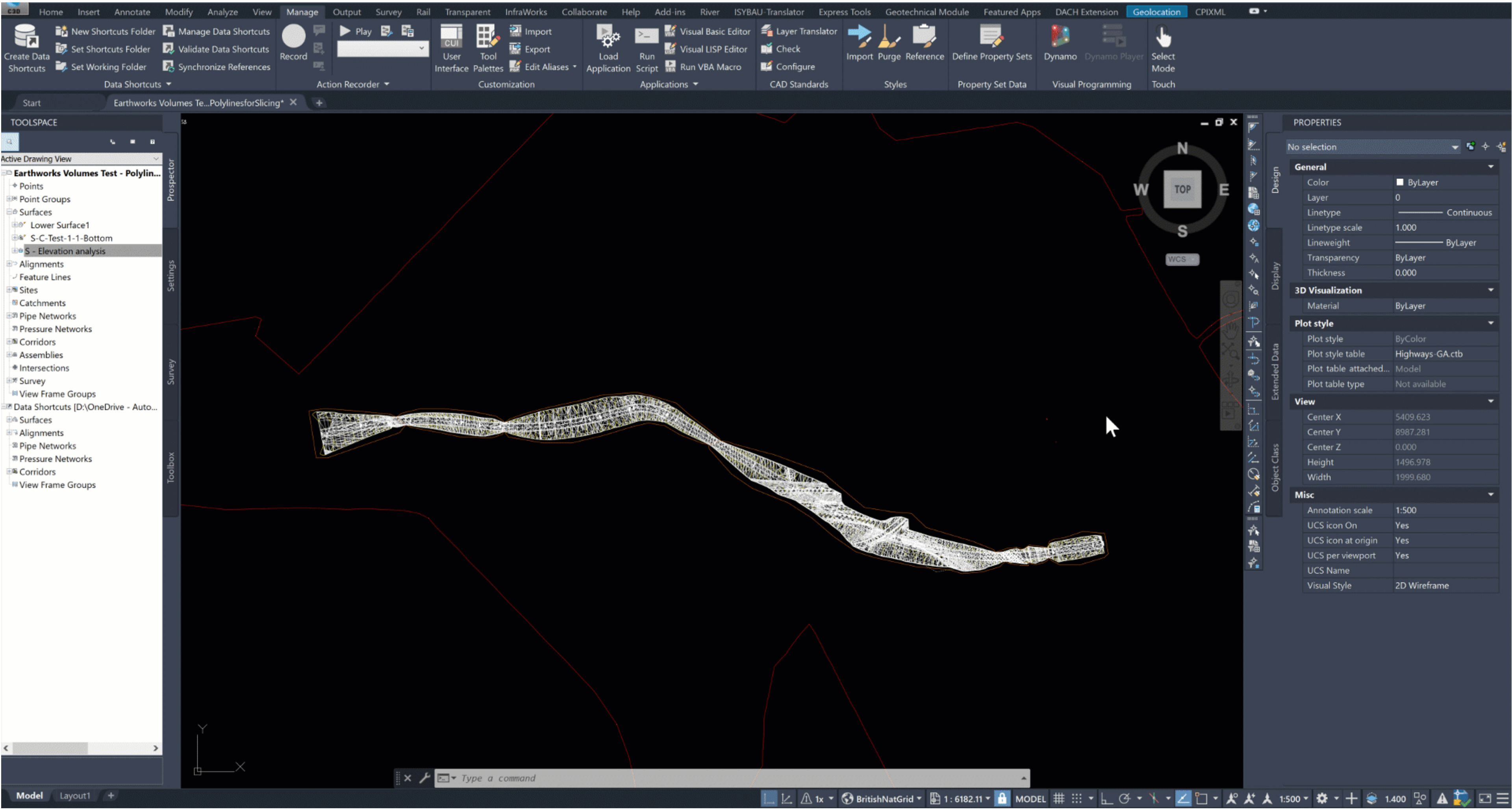


# Gráfica 2 : Property Sets





# Gráfica 2 : Property Sets





# Objetivos de aprendizaje



## IDENTIFICAR LAS OPORTUNIDADES DE AUTOMATIZACIÓN

- ✓ La automatización para realizar más, mejor, y con menos recursos
- ✓ Casos prácticos : tareas repetitivas, tareas con poco valor añadido, procesos iterativos, etc.
- ✓ Centrarse en los “quick wins”



## DESARROLLAR SOLUCIONES INDUSTRIALIZABLES

- Construir soluciones que puedan ser utilizadas con el Dynamo Player
- Integrar las gráficas con notas
- Modularizar las soluciones para realizar pruebas y detectar fallos más fácilmente



## PENSAR ANALÍTICAMENTE

- ✓ Describir la secuencia lógica para resolver los problemas
- ✓ Crear un « POC » rápidamente, utilizar datasets neutros
- ✓ No cambiar el intervalo de geometría, tener en cuenta el binding y la utilización de objetos provisionales

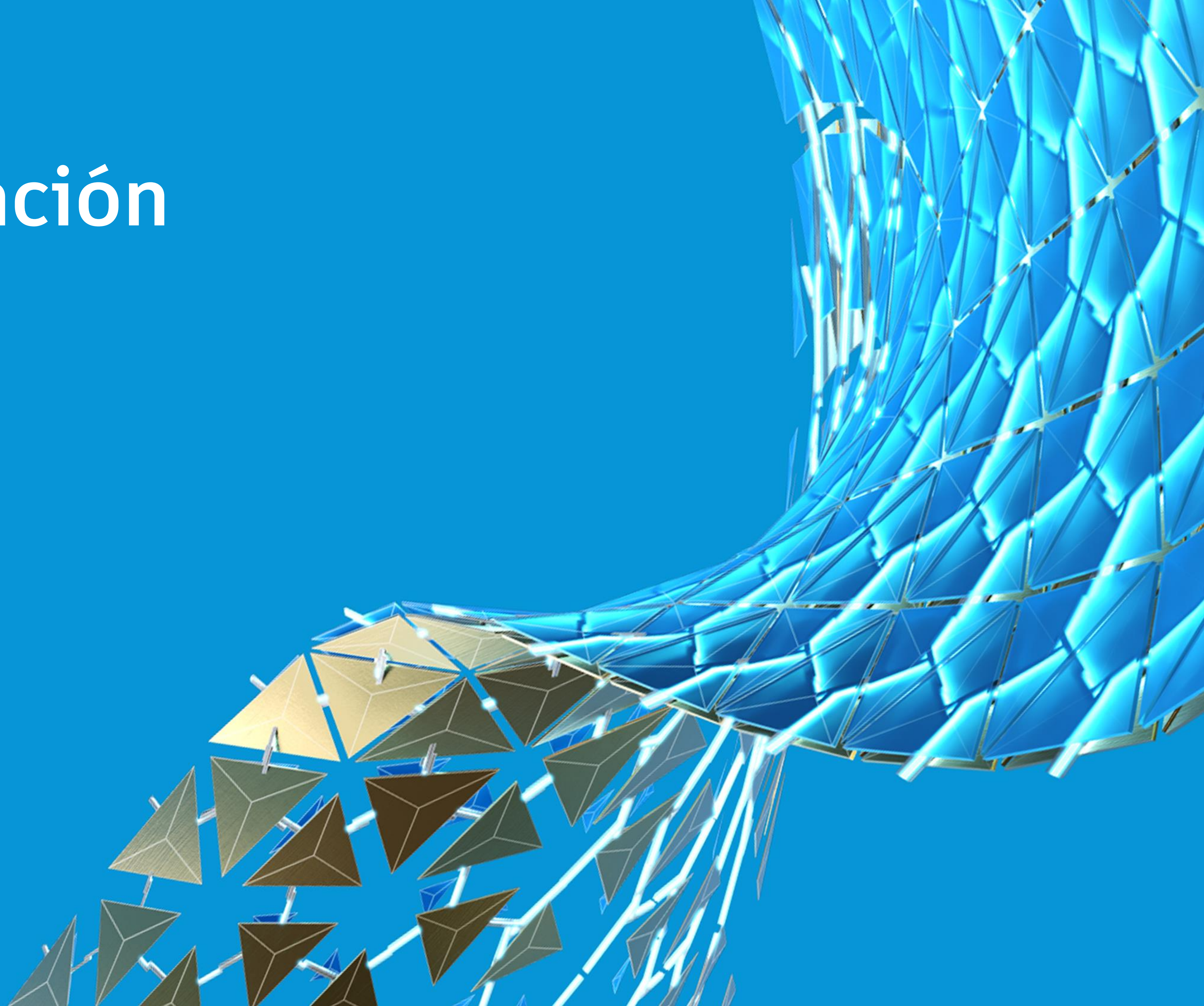


## IMPLEMENTAR LA AUTOMATIZACIÓN

- Mejores prácticas
- Comunicación de los resultados
- Recursos necesarios para triunfar



# Implementación





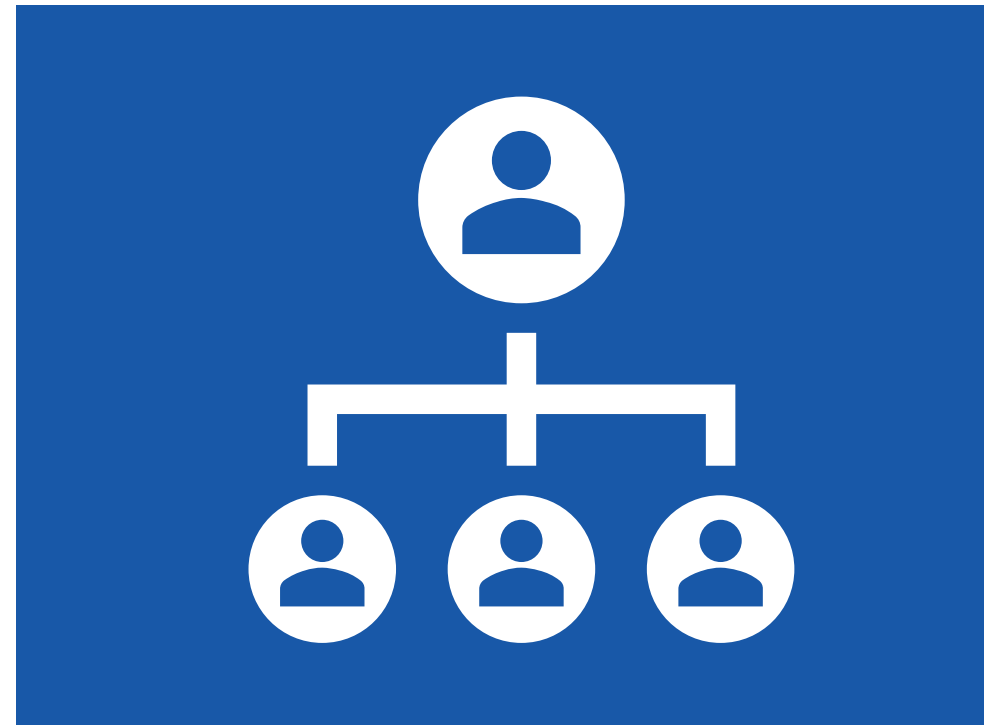
# Un cambio consecuente

## PERSONAL



Capacitación o  
contratar recursos  
especializados

## ORGANIZACIÓN



Nuevos puestos y  
responsabilidades

## PROCESOS



Estrategia de  
implementación y plan  
de acción

## TECNOLOGÍAS

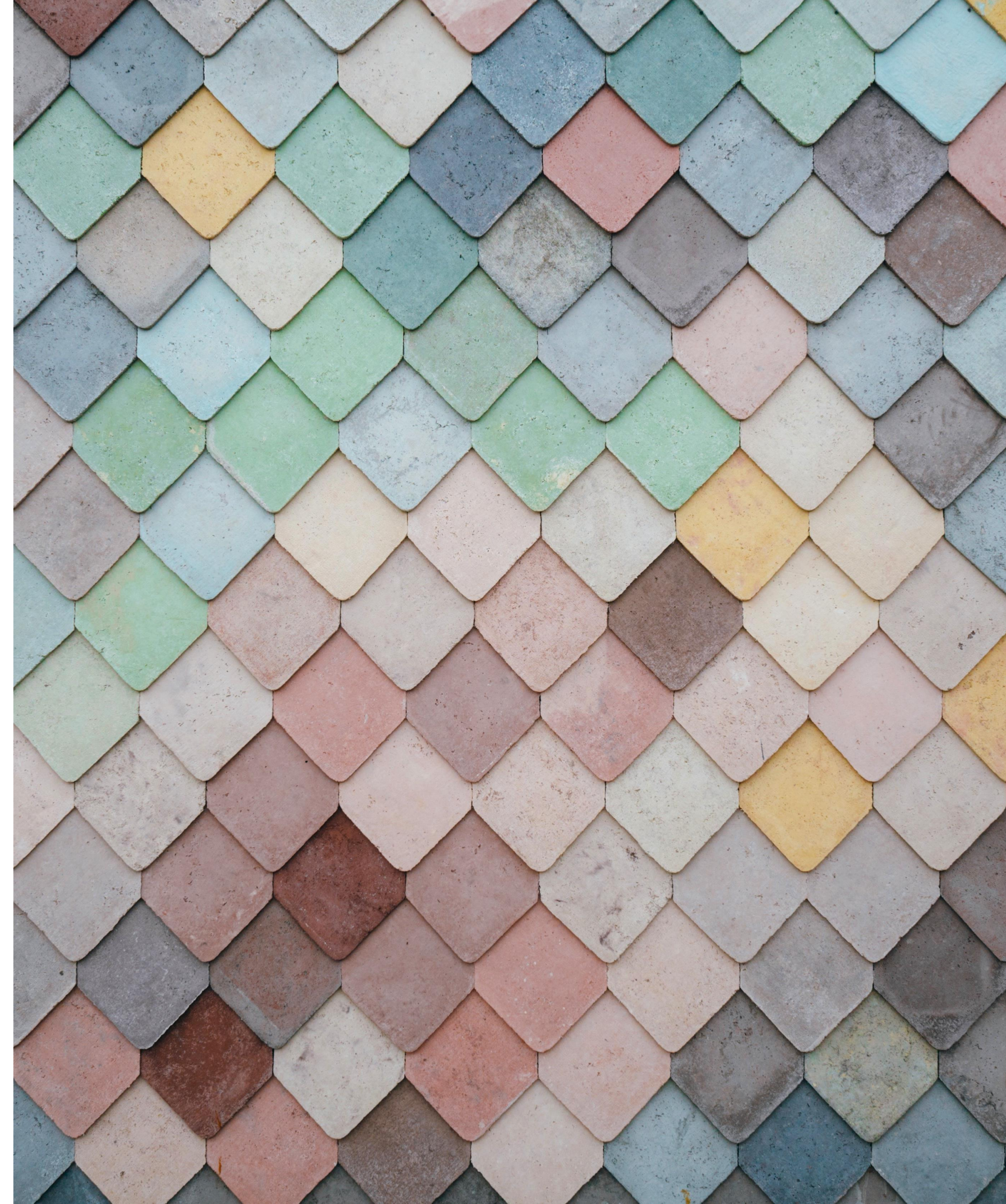


Herramientas  
tecnológicas



# Nomenclatura de archivos

- Clarifica el contenido
- Provee una estructura
- Facilita la búsqueda
- Ejemplo :
  - Programa\_Disciplina\_UtilizaciónBIM\_Propósito\_Versión
  - « C3D\_Carreteras\_Documentación\_PerfilLongitudinal\_V1 »





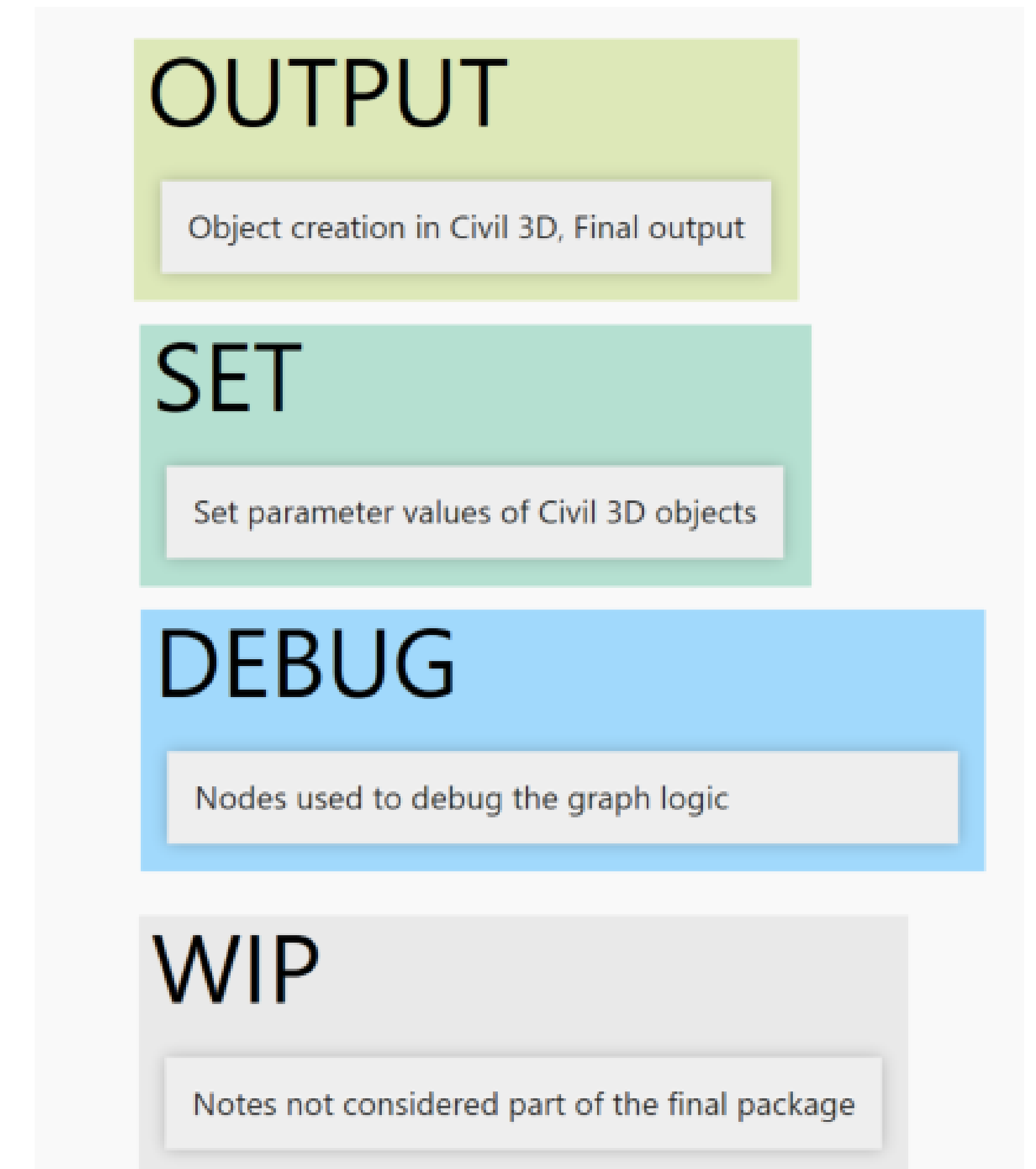
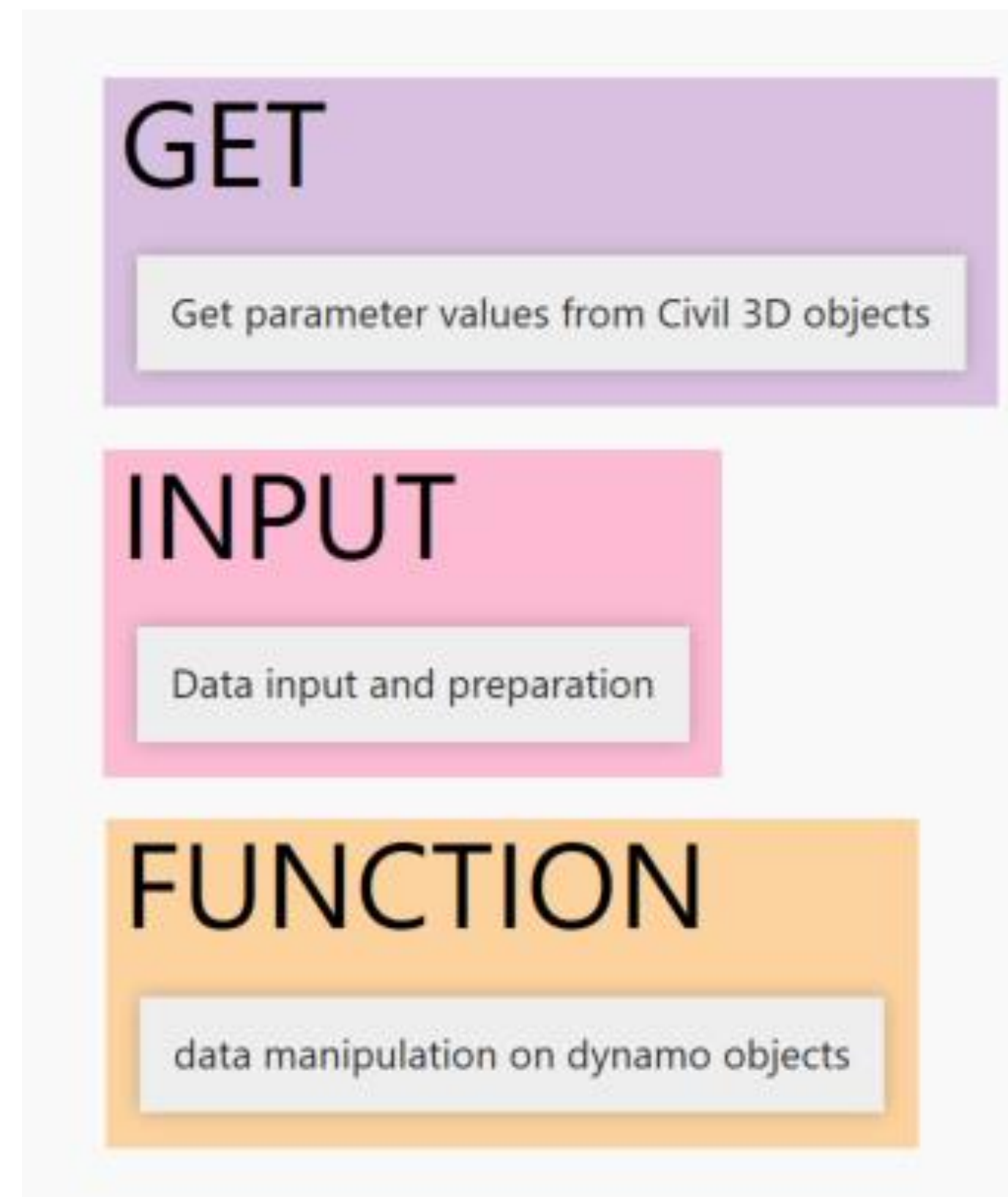
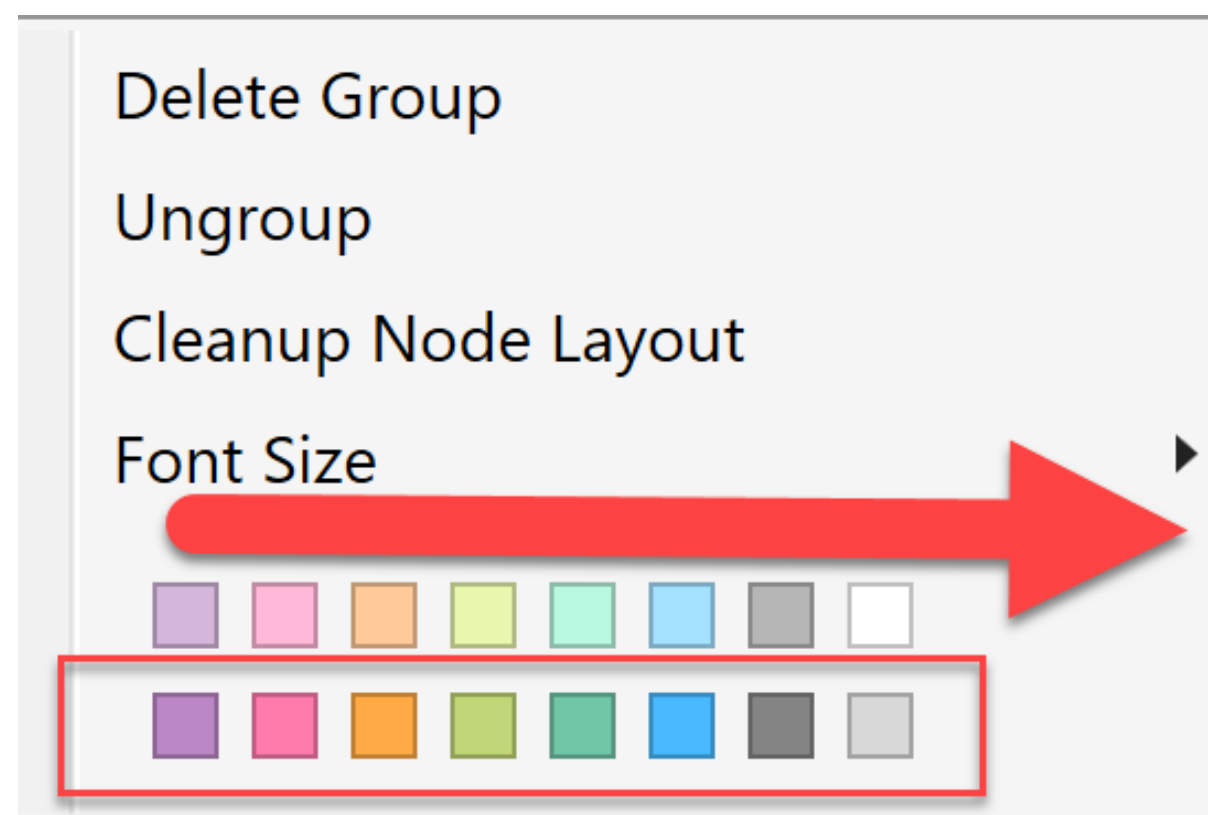
# Plantilla estándar

- Información general
- Instrucciones
- Limitaciones





# Utilización de colores





# NOTES

GRAPH INFO

Copyright 2017 Autodesk, Inc. All rights reserved.  
Company: <company>  
Office: <office>  
  
Version: 1.0.0  
Author: <author>  
paolo.serra@autodesk.com  
  
Keywords: [KEYWORDS]  
  
Tested on:  
Dynamo : 2.X.X  
Civil 3D : 20XX

INSTRUCTIONS

<DESCRIPTIONS>  
  
<INSTRUCTIONS>

KNOWN ISSUES AND LIMITATIONS

<ISSUES>  
  
<LIMITATIONS>

GUIDELINES

Read the instructions.  
  
Add Notes and Comments to the graph.  
  
Use Node Groups and the Standard Color Coding.  
  
Rename Nodes: <OriginalName> | <Description>.  
  
Write Input and Output Notes for Python Scripts.  
  
Prefer repeatable simple node structures.  
  
Simple is better than complex.  
  
Complex is better than complicated.  
  
Readability counts.  
  
Special cases aren't special enough to break the rules.  
  
Although practicality beats purity.  
  
Errors should never pass silently.  
  
Unless explicitly silenced.  
  
In the face of ambiguity, do not guess.  
  
There should be one obvious way to do it.  
  
Now is better than never.  
  
Although never is often better than \*right\* now.

RATING: 5

This Python Script creates a CSV file to track the usage of the Dynamo file.

Python Script | Usage Tracker

+ - OUT

Useful Links | CTRL + Click the link below

"[http://dynamoprimer.com/en/10\\_Packages/10-Packages.html](http://dynamoprimer.com/en/10_Packages/10-Packages.html)"; >  
"[http://dynamoprimer.com/en/12\\_Best-Practice/12-1\\_Introduction.html](http://dynamoprimer.com/en/12_Best-Practice/12-1_Introduction.html)"; >

## GET

Get parameter values from Civil 3D objects

## INPUT

Data input and preparation

## FUNCTION

data manipulation on dynamo objects

## OUTPUT

Object creation in Civil 3D, Final output

## SET

Set parameter values of Civil 3D objects

## DEBUG

Nodes used to debug the graph logic

## WIP

Notes not considered part of the final package



# Almacenamiento de archivos

- Almacenamiento centralizado
- Diferencia entre las fuentes y los archivos Dynamo utilizados para el proyecto (**BINDING**)





# Pruebas y validación

- Validación dada por el utilizador
- Proceso iterativo
- Integrar los comentarios relevantes





# Seguimiento

- **Cuantificar** la utilización :
  - Saber cuales son las gráficas más utilizadas
  - Calcular el ROI sobre el **VALOR**
- En Autodesk Consulting, desarrollamos una herramienta para cuantificar la utilización de las gráficas Dynamo





# Comunicación

- Comunicar los casos exitosos
- Utilizar indicadores de desempeño para comunicar
- Consejo : Realizar videos
- « Story telling »
  - Definir el problema
  - Como fue solucionado
  - Dar pruebas del valor aportado





# Grupo de utilizadores

- Super utilizadores
- Experiencia de “campo”
- Facilitar la comunicación y adopción





# Recursos humanos

- Necesidad de tener personal especializado:
  - **Capacitación** del personal
  - Adquisición de talento externo
  - Acompañamiento para los primeros casos





# Otras Referencias

- Primer
- Foro Dynamo – Civil 3D
- Clases anteriores AU de Dynamo para Civil 3D :
  - [Computational Design for Civil Engineers](#) | Paolo Serra et Safi Hage
  - [Dynamo in Civil 3D Introduction–Unlocking the Mystery of Scripting](#) | Jowenn Lua et Andrew Milford
  - [Accesorize your Design – Dynamo for Civil 3D](#) | Stacey Morkin et Dylan Kahle
  - [Generating, Transforming, and Analyzing Railway Design Data in Civil 3D and Dynamo](#) | Wouter Bulens et Steve Crokaert
- Clases AU 2020 (link no disponible) :
  - Diseño Generativo | Raquel Bascones y Paolo Serra
  - Civil 3D Toolkit | Jowenn Lua





# Objetivos de aprendizaje



## IDENTIFICAR LAS OPORTUNIDADES DE AUTOMATIZACIÓN

- ✓ La automatización para realizar más, mejor, y con menos recursos
- ✓ Casos prácticos : tareas repetitivas, tareas con poco valor añadido, procesos iterativos, etc.
- ✓ Centrarse en los “quick wins”



## DESARROLLAR SOLUCIONES INDUSTRIALIZABLES

- Construir soluciones que puedan ser utilizadas con el Dynamo Player
- Integrar las gráficas con notas
- Modularizar las soluciones para realizar pruebas y detectar fallos más fácilmente



## PENSAR ANALÍTICAMENTE

- ✓ Describir la secuencia lógica para resolver los problemas
- ✓ Crear un « POC » rápidamente, utilizar datasets neutros
- ✓ No cambiar el intervalo de geometría, tener en cuenta el binding y la utilización de objetos provisionales



## IMPLEMENTAR LA AUTOMATIZACIÓN

- Plantillas con códigos de colores, cuantificar la utilización, nomenclatura de archivos
- Comunicación para demostrar el valor añadido, apoyarse en el “campo” para una comunicación efectiva
- Plan de gestión de recursos humanos



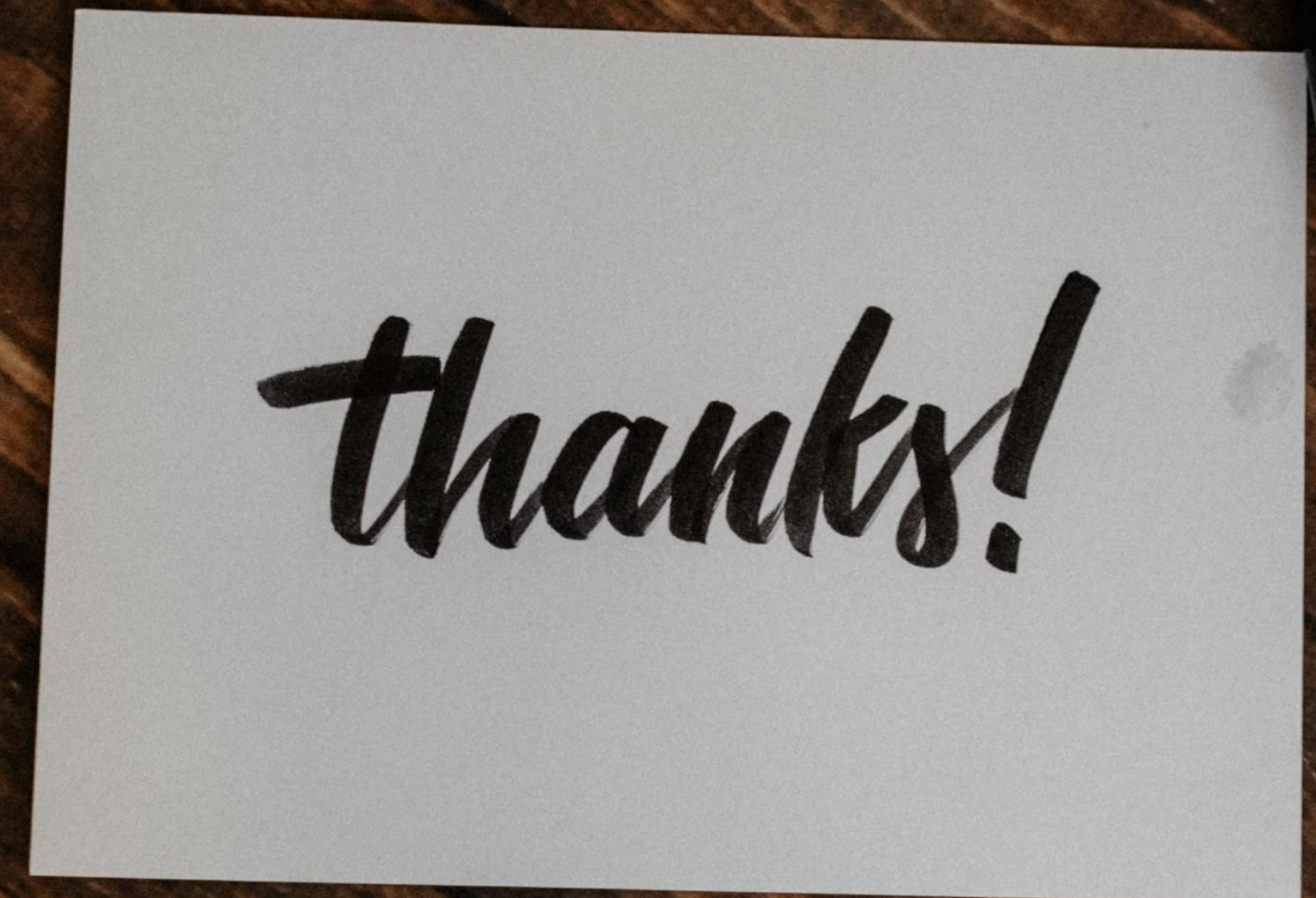
# Agradecimientos

POR LA AUTORIZACIÓN DE COMPARTIR ESTE CASO  
PRÁCTICO CON USTEDES:

El equipo de Jacobs y especialmente a Duncan Taylor

POR SU AYUDA Y CONSEJOS TÉCNICOS:

Paolo SERRA, Jowenn LUA et Philippe BONNEAU





**La automatización no es tan aterradora como parece, puede ser realmente simple.**

**Solo hay que pensar en las tareas frustrantes o en las que no se necesite reflexionar, podría hacerlos ganar mucho tiempo.**

Sol AMOUR, Product Manager Dynamo





Autodesk and the Autodesk logo are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product and services offerings, and specifications and pricing at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.

© 2020 Autodesk. All rights reserved.