



UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

CORSO DI LAUREA TRIENNALE IN INGEGNERIA EDILE

TECNICHE DI GENERATIVE DESIGN PER L'INGEGNERIA DELLE COSTRUZIONI

GENERATIVE DESIGN TECHNIQUES FOR CONSTRUCTION ENGINEERING

Relatore :

Prof. Alberto Giretti

Tesi di laurea di :

Marco Magini

A.A 2020/2021

I computer sono incredibilmente veloci, accurati e stupidi,
gli uomini sono incredibilmente lenti, inaccurati e intelligenti,
l'insieme dei due costituisce una forza incalcolabile

Albert Einstein

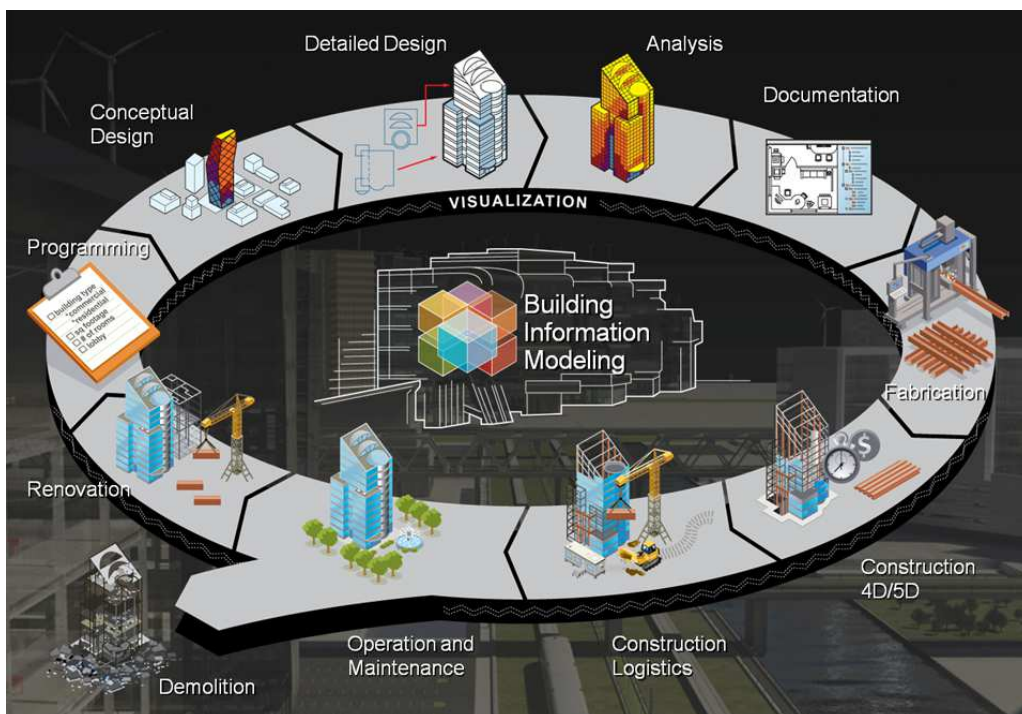
INDICE

1. INTRODUZIONE AL BIM	4
1.1 BIM come parola polisemica.....	5
1.2 TRE punti chiave del BIM.....	8
1.3 Costruire come un verbo, non come un sostantivo.....	10
1.4 Pensare all'ambiente costruito.....	11
1.5 Definizione di oggetti parametrici.....	12
1.6 Cosa non è la tecnologia BIM?.....	13
1.7 Quali sono i vantaggi del BIM ? Quali problemi affronta ?.....	14
1.8 Vantaggi del progetto.....	18
1.9 Vantaggi della costruzione e della prefabbricazione.....	19
1.10 Benefici post-costruzione.....	21
2. PIATTAFORME BIM	23
2.1 Revit.....	23
3. DYNAMO	25
3.1 Che cosa è ?.....	25
3.2 Su cosa lavora Dynamo ?	26
3.3 Code Block.....	31
3.4 Note.....	31
3.5 Raggruppamento.....	33
3.6 Che cosa sono i dati?.....	34
4. REVIT API	37
4.1 Introduzione alle API di Revit.....	37
4.2 Struttura delle API di Revit.....	39
4.3 Introduzione a .NET.....	39
4.4 Modello di Python Script.....	40

4.5	Lettura documentazione delle API di Revit.....	42
4.6	Filtraggio di elementi : Muri.....	45
4.7	Elenco di tutte le viste presenti nel disegno.....	46
4.8	Creazione di elementi: Muri.....	48
4.9	Creazione di elementi: Solaio.....	51
5.	Dynamo per la parte Architettonica.....	55
6.	Dynamo per la parte Strutturale.....	61
7.	Conclusioni.....	68
8.	Ringraziamenti.....	69
9.	Indice delle figure.....	70
10.	Bibliografia/Sitografia.....	72

1 INTRODUZIONE AL BIM

Il Building Information Modeling è la rappresentazione digitale delle caratteristiche fisiche e funzionali di una struttura che crea una risorsa di conoscenza condivisa per le informazioni su di esso e formando una base affidabile per le decisioni durante il suo ciclo di vita, dalla prima concezione alla demolizione.



1) Processo completo di vita di un fabbricato

Quindi non si tratta solo il mero modello geometrico 3D ma di tutte quelle informazioni sul ciclo di vita della risorsa.

Con il nuovo progredire di questa metodologia si iniziano a porre nuovi quesiti riguardanti la progettazione :

- 1) Cosa fare?
- 2) Quali informazioni usare e/o produrre?
- 3) Come usano gli altri le vostre informazioni, e quali informazioni chiedere agli altri?
- 4) Come collaborare e lavorare insieme?

Questo può quindi portare ad una maggiore efficienza e riduzione degli sprechi nel mondo dell'edilizia semplicemente perché stiamo rielaborando molto meglio le informazioni, e aumentando la qualità di quello che stiamo producendo e/o gestendo.

Il compito è quello di capire cosa viene prodotto, da chi, per cosa, cosa ne fanno, chi lo riceve, cosa ne fa e così via.

Secondo il National Building Information Model Standard (NIBS, 2015), Il Building Information Modelling (BIM) è una rappresentazione digitale delle caratteristiche fisiche e funzionali di una struttura.

1.1 BIM come parola polisemica

Il BIM può stare per la rappresentazione digitale delle caratteristiche fisiche e funzionali di una struttura (Building Information Model), o il processo di creazione del modello e l'ulteriore sfruttamento dei suoi dati (Building Information Modelling).



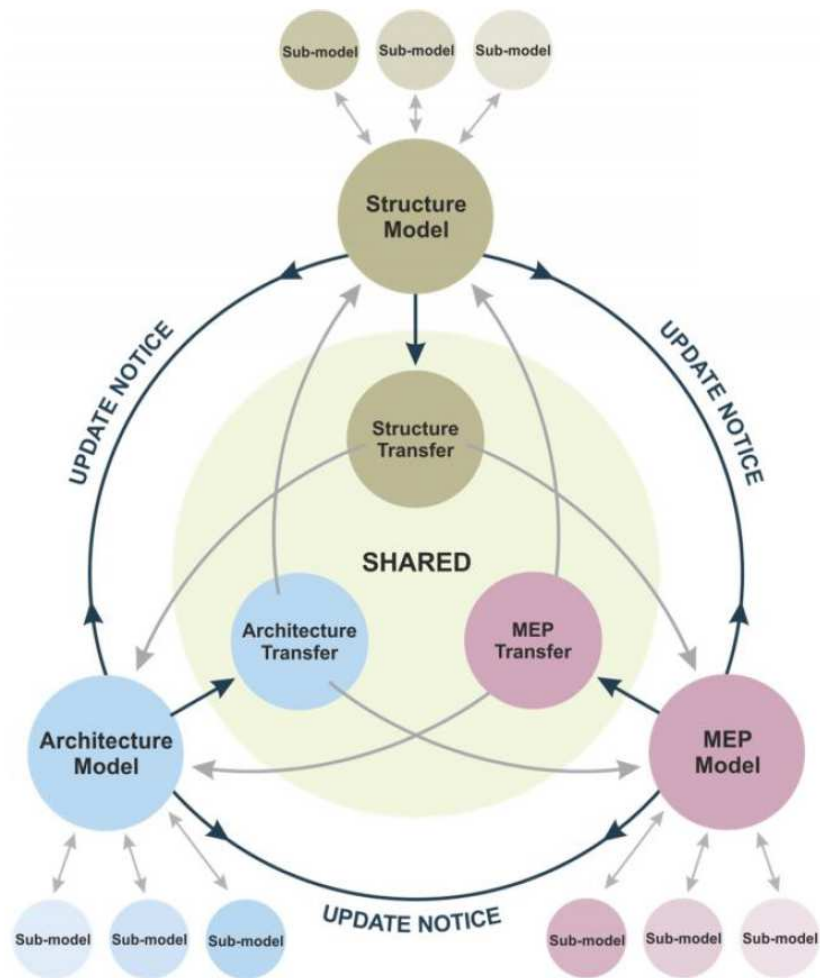
2) Molteplici usi del BIM

I tradizionali disegni CAD 2D sono insiemi di linee, poligoni o superfici mentre i disegni 2D possono essere assegnati con dimensioni (per esempio, lunghezza), sono per così dire grafici 'morti'. Al contrario, il BIM è 'vivo'. Un 'deposito più ricco' di informazioni.

Quando un progettista cambia un componente, i suoi componenti dipendenti cambiano di conseguenza. Il modello BIM dettaglia le proprietà di ogni suo componente, le sue caratteristiche termiche, il suo materiale e le informazioni sulla quantità e così via...

Il BIM ha quindi il potenziale per automatizzare la misurazione e facilitare la preparazione di stime accurate.

Prima dell'avvento del BIM, l'industria delle costruzioni lavorava generalmente in modo separato, dove ogni membro di un team di progetto guardava solo ai propri interessi e il progetto passava in secondo piano rispetto ad altre priorità. La promessa del BIM è di costruire una struttura virtualmente prima di costruirla fisicamente. Questo permette ai partecipanti al progetto di progettare, analizzare, sequenziare ed esplorare un progetto attraverso un ambiente digitale dove è molto meno costoso fare cambiamenti che sul campo durante la costruzione, dove i cambiamenti sono esponenzialmente più costosi. Inoltre, quando un team di gestione della costruzione è autorizzato a partecipare alla riunione di un architetto, sono in grado di vedere quali fattori sono importanti per il cliente e il team di progettazione e di utilizzare questa conoscenza in futuro, mentre si preparano a costruire permette ai team di funzionare in modo più efficiente e di usare il BIM per ottenere risposte migliori più velocemente. L'integrazione del team sposta l'attenzione oltre le esigenze individuali e la sposta su come i modelli ricchi di informazioni possono essere usati per esplorare opzioni e scenari che creare progetti migliori ed eliminare i rischi eliminando anche problemi del mondo reale, come la verifica dell'installazione, il sequenziamento e la stima



3) Organizzazione della struttura di ogni membro del team

1.2 TRE punti chiave del BIM

l'uso di successo del BIM richiede tre fattori chiave:

- Processi
- Tecnologie
- Comportamenti

Questi tre componenti possono fare o rompere un progetto usando il BIM e la tecnologia.

Pensate a questi come allo sgabello a tre gambe per l'integrazione e l'uso di successo del BIM

Processi

La gestione delle costruzioni e molte altre aziende focalizzate sull'ingegneria tendono a prendere nuove tecnologie e cercare di farle funzionare nei vecchi processi. Questo approccio crea sprechi non prendendo in considerazione le implicazioni del nuovo strumento e quali processi esistenti e i flussi di lavoro esistenti dovrebbero cambiare per rendere un risultato più efficiente. Un buon esempio è stata l'evoluzione del rilevamento e della risoluzione degli scontri. Quando il rilevamento degli scontri ha iniziato a prendere piede, molti team organizzavano una serie di riunioni ogni settimana che coinvolgevano l'intero team del progetto per coordinarsi tra loro usando questo nuovo ambiente 3D.



4) Riunione di coordinamento BIM, per rilevare le eventuali interferenze

Anche se la tecnologia era migliore, il processo utilizzato era simile a quello che era stato fatto prima in una revisione di coordinamento 2D. Di conseguenza, molti utenti trovavano che il nuovo processo non solo era inefficiente, ma addirittura dannoso per l'efficienza del progetto. Perché i membri del team erano impegnati in riunioni di revisione del rilevamento delle collisioni, i tempi di risposta per il progetto aumentavano. Al giorno d'oggi, queste riunioni si concentrano tipicamente su due o tre mestieri o ambiti particolari in un lasso di tempo definito di 2 o 3 ore per utilizzare al meglio le risorse di ogni membro del team.

Questi cambiamenti di processo sono fondamentali per il miglioramento, perché permettono agli utenti di pensare continuamente a modi per migliorare e consegnare il lavoro.

Tecnologie

Il successo dell'integrazione del BIM implica l'uso di strumenti BIM che funzionano

Dei tre componenti chiave per integrare con successo il BIM, i comportamenti sono i più difficile da cambiare.

- è un cambiamento culturale nella mentalità del modo in cui i team di gestione delle costruzioni collaborano.

team di gestione della costruzione collaborano.

Comportamenti

Quando consideriamo cosa fa funzionare il BIM all'interno di un progetto di costruzione, la componente centrale diventa il comportamento abilitante.

Le squadre devono rendersi pienamente conto che una mentalità che guarda al futuro è altrettanto importante quanto le tecnologie e i processi che vi stanno dietro. Coloro che fraintendono questo principio si troveranno rapidamente irrilevanti nel mercato della progettazione e della costruzione.

1.3 Costruire come un verbo, non come un sostantivo

Se si pensa a costruire come un verbo, che significa lo stesso di costruire o il processo di mettere cose insieme, allora si comincia ad espandere ciò a cui il BIM può applicarsi. il BIM non è solo adatto agli edifici, è l'atto di costruire cose :

- Ponti
- Ferrovie
- Autostrade
- Utilities

Si può anche immaginare come sia adatto ad altri settori come:

- Rilievo del territorio
- Architettura del paesaggio
- Tunneling
- Miniere

Ogni settore è in fasi diverse di esplorazione del BIM, ed esistono già grandi esempi di quelle industrie che usano il BIM su progetti dal vivo, quindi non bisogna fissarsi sulla parola costruzione. Per esempio, il progetto Virtual Construction for Roads (V-Con) è un'iniziativa europea per migliorare lo scambio di dati tra i team delle infrastrutture civili utilizzando processi BIM e sta cambiando gli appalti stradali in Olanda e Svezia

Il BIM quindi non riguarda solo l'architettura.

Anche se il design degli edifici e l'industria l'edilizia siano state le prime ad adottare i processi e i protocolli BIM stanno aumentando i settori a cui si applica la modellazione delle informazioni

Il BIM è stato dimostrato con successo in

- Architettura e progettazione di edifici
- Ingegneria civile e strutturale
- Energia e servizi pubblici
- Ingegneria autostradale e stradale
- Paesaggio e agrimensura
- Architettura offshore e marina
- Ingegneria del trasporto ferroviario e metropolitano

- Servizi e ingegneria
- Architettura di tunnel e metropolitane
- Master-planning urbano e progettazione di città intelligenti

Dimostra che costruire è un processo. Non si tratta di uno scambio di dati una tantum, ma di molti scambi nel corso della vita di un progetto. La maggior parte degli input del modello saranno nelle fasi di progettazione e costruzione, e la maggior parte delle informazioni in uscita sarà estratti durante la consegna, l'uso e la manutenzione del bene. La modellazione delle informazioni per il processo di costruzione potrebbe iniziare il primo giorno ed essere ancora forte anni dopo.

Si può usare il BIM per ogni tipo di progetto di costruzione, da ponti giganti a isole artificiali e persino montagne russe!

1.4 Pensare all'ambiente costruito

L'ambiente costruito è molto vario e ampio nella sua portata e comprende molte strutture che non sono edifici. Quando si parla di BIM, assicurarsi che non si stia parlando solo dell'architettura e dell'industria dell'architettura, dell'ingegneria e delle costruzioni (AEC). Le infrastrutture sono la rete di sistemi che mantengono le cose in movimento, che si tratti di acqua, gas, elettricità, traffico o dati Internet. La progettazione, la costruzione, e la manutenzione di queste strutture devono usare l'approccio del BIM per l'intero ciclo di vita.

Per esempio, Crossrail (www.crossrail.co.uk) è il più grande progetto di costruzione in Europa e, tra i tanti progetti, prevede lo scavo di 26 miglia di nuovissime linee metropolitane sotterranee. Ogni aspetto del progetto, dall'ingegneria dei tunnel alle nuove stazioni della metropolitana, ha utilizzato processi BIM innovativi per la gestione dei dati e il funzionamento del ciclo di vita.

1.5 Definizione di oggetti parametrici

Il concetto di oggetti parametrici è centrale per comprendere il BIM e la sua differenziazione

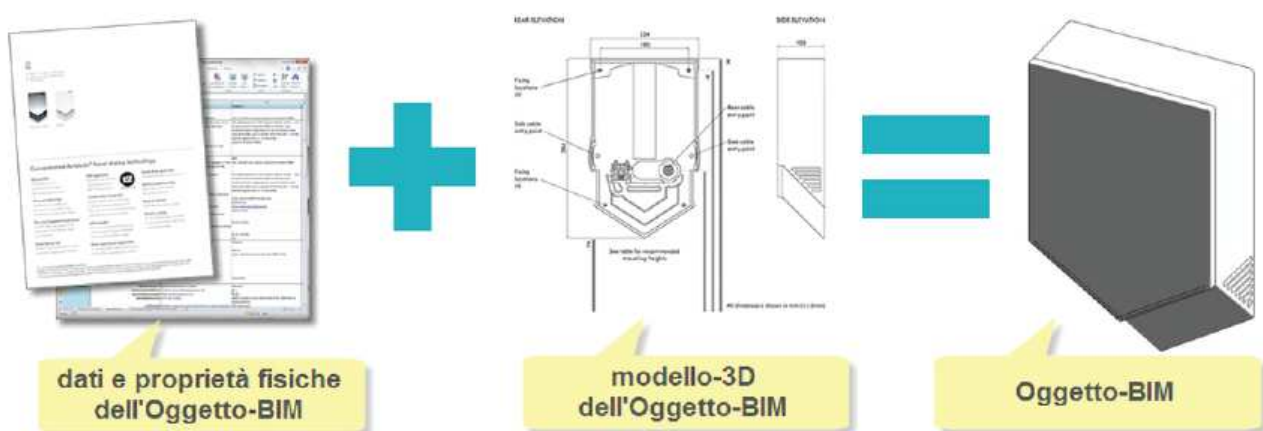
Rispetto agli oggetti 3D tradizionali gli oggetti parametrici BIM consistono in definizioni geometriche e dati e regole associate.

La geometria è integrata in modo non ridondante e non permette incongruenze. Quando un oggetto viene mostrato in 3D, la forma non può essere rappresentata internamente ridondante, per esempio, come molteplici viste 2D. Una pianta e un'elevazione di un dato oggetto devono essere sempre coerenti. Le dimensioni non possono essere "falsificate". Le regole parametriche per gli oggetti modificano automaticamente le geometrie associate quando vengono inseriti in un modello di edificio o quando vengono apportate modifiche a oggetti associati.

Per esempio, una porta si incastra automaticamente in un muro, un interruttore della luce si posizionerà automaticamente accanto al lato corretto della porta, un muro si ridimensionerà automaticamente per appoggiarsi a un soffitto o a un tetto, e così via.

Gli oggetti possono essere definiti a diversi livelli di aggregazione, così possiamo definire un muro così come i suoi componenti correlati. Gli oggetti possono essere definiti e gestiti a qualsiasi numero di livelli di gerarchia. Per esempio, se il peso di un sottocomponente del muro cambia, anche il peso del muro dovrebbe cambiare.

Gli oggetti hanno la capacità di collegarsi a o ricevere, trasmettere o esportare insiemi di attributi, per esempio, materiali strutturali, dati acustici, dati energetici, e simili, ad altre applicazioni e modelli.



5) Dati e definizione di un oggetto parametrico BIM

1.6 Cosa non è la tecnologia BIM?

Questi includono strumenti che creano i seguenti tipi di modelli: Modelli che contengono solo dati 3D e nessun (o pochi) attributi dell'oggetto. Questi sono modelli che possono essere usati solo per visualizzazioni grafiche e non hanno intelligenza a livello di oggetto. Sono adatti per la visualizzazione, ma forniscono poco o nessun supporto per l'integrazione dei dati e l'analisi del design. Un esempio è l'applicazione SketchUp di Google che è eccellente per lo sviluppo rapido di disegni schematici di edifici, ma di uso limitato per qualsiasi altro tipo di analisi perché non ha alcuna conoscenza degli oggetti nel progetto se non la loro geometria e il loro aspetto per la visualizzazione.

Modelli senza supporto di comportamento. Questi sono modelli che definiscono gli oggetti ma non possono regolare il loro posizionamento o le loro proporzioni perché non utilizzano l'intelligenza parametrica. Questo rende le modifiche estremamente laboriose e non fornisce alcuna protezione contro la creazione di viste incoerenti o viste incoerenti o imprecise del modello.

Modelli che sono composti da più file di riferimento CAD 2D che devono essere combinati per definire l'edificio. È impossibile assicurare che il modello 3D risultante sia fattibile, coerente, calcolabile e mostri intelligenza rispetto agli oggetti in esso contenuti.

Modelli che permettono di cambiare le dimensioni in una vista che non sono automaticamente riflessi nelle altre viste. Questo permette errori nel modello che sono molto difficili da rilevare

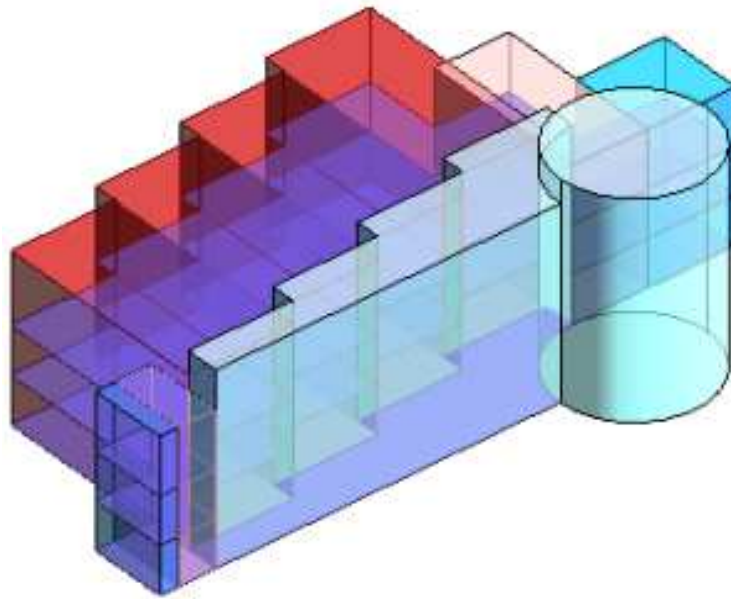
1.7 Quali sono i vantaggi del BIM ? Quali problemi affronta?

La tecnologia BIM può sostenere e migliorare molte pratiche commerciali. Anche se l'industria AEC/FM (facility management) sia agli inizi dell'uso del BIM, sono già stati realizzati miglioramenti significativi (rispetto alle tradizionali CAD 2D o pratiche cartacee).

La pratica tradizionale non è in grado di rispondere a queste pressioni. Prima che i proprietari ingaggino un architetto, è necessario determinare se un edificio di una data dimensione, livello di qualità e requisiti di programma desiderati può essere costruito entro un determinato budget di costi e tempi. In altre parole, un dato edificio può soddisfare i requisiti finanziari del proprietario? Se a queste domande si può

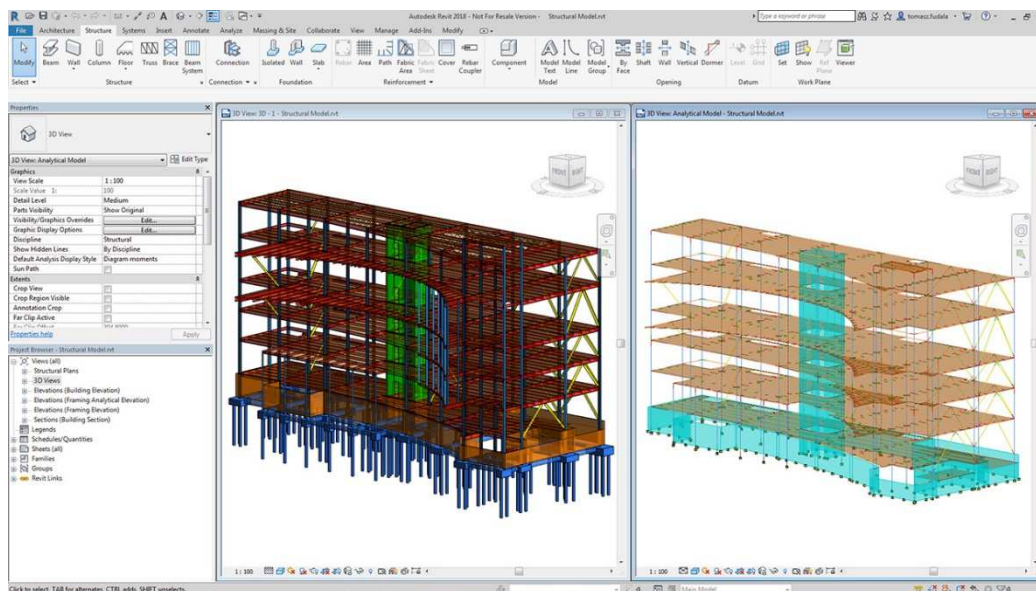
rispondere a queste domande con relativa certezza, i proprietari possono procedere con l'aspettativa che i loro obiettivi siano raggiungibili. Scoprire che un particolare progetto è significativamente al di sopra del budget dopo che è stata spesa una considerevole quantità di tempo e gli sforzi sono stati spesi è uno spreco. Un modello di costruzione approssimativo (o "macro") e collegato a un database dei costi può essere di enorme valore e assistenza per un proprietario

- Aumento delle prestazioni e della qualità dell'edificio
- Sviluppare un modello schematico prima di generare un modello dettagliato dell'edificio permette una valutazione più attenta dello schema proposto per determinare se soddisfa i requisiti funzionali e sostenibili dell'edificio

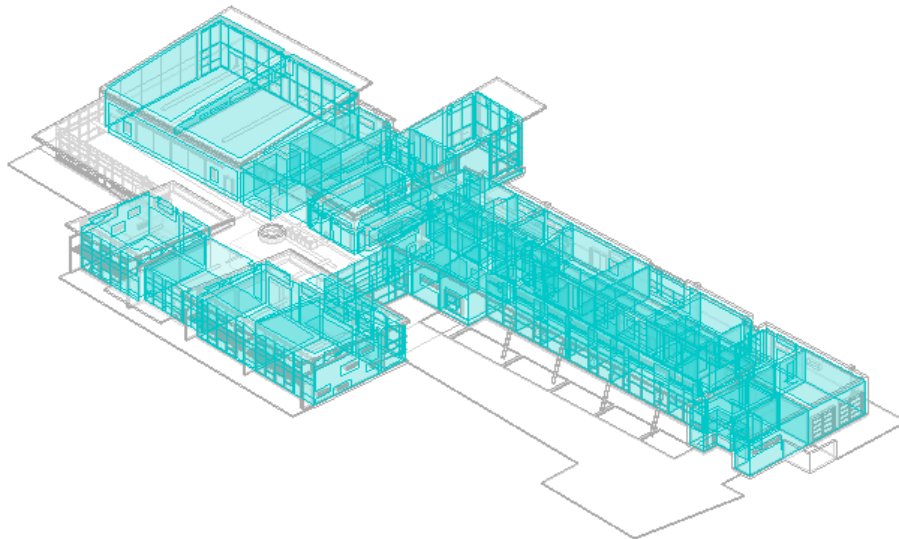


6) Modello Schematico di un edificio mediante l'uso di masse (Revit)

- valutazione delle alternative di progettazione usando strumenti di analisi/simulazione
aumenta la qualità complessiva dell'edificio



7) impostazione di un'analisi strutturale di un edificio (Revit)



8) impostazione di un'analisi energetica di un edificio (Revit)

- Collaborazione migliorata usando la consegna integrata del progetto

Il BIM può essere usato dalla squadra di progetto dall'inizio della progettazione per migliorare la loro comprensione dei requisiti del progetto e per estrarre le stime dei costi man mano che il progetto viene sviluppato. Questo permette al design e ai costi di essere meglio comprensione ed evita anche l'uso di scambi cartacei e i relativi ritardi.

The screenshot shows the 'Modifica abaco/quantità' dialog box in Autodesk Revit 2017. The main window displays a table with the following data:

Id	Destinazione d'uso	Superficie del locale	Volume Netto (m 2,70 m)
1	CAMERA 1	15,0 m²	58,5 m³
2	DISIMPEGNO	45,2 m²	178,4 m³
3	SOGGIORNO	38,6 m²	158,0 m³
4	CAMERA DA LETTO	32,9 m²	128,3 m³
5	BAGNO	13,4 m²	52,3 m³
6	CUCINA	6,8 m²	28,5 m³
7	PRANZO	8,0 m²	32,4 m³
		198,1 m²	812,3 m³

The 'Proprietà abaco' dialog box is also visible, showing a list of available fields for the schedule, including 'Numero', 'Nome', 'Area', and 'Volume'.

9) Stima dei costi mediante abachi creati dal programma Revit

1.8 Vantaggi del progetto

- Visualizzazioni più rapide e accurate di un progetto
- Il modello 3D generato dal software BIM è progettato direttamente piuttosto che essere generato da più viste 2D. Può essere usato per visualizzare il progetto in qualsiasi fase del processo con l'aspettativa che sia dimensionalmente coerente in ogni vista.
- Correzioni automatiche di basso livello quando vengono apportate al progetto
- Se gli oggetti usati nel progetto sono controllati da regole parametriche che assicurano allineamento corretto, allora il modello 3D sarà privo di errori di geometria, allineamento e errori di coordinazione spaziale. Questo riduce la necessità dell'utente di gestire le modifiche al progetto
- Generazione di disegni 2D accurati e coerenti in qualsiasi fase della progettazione. Questo riduce significativamente la quantità di tempo e di errori associati alla generazione di disegni di costruzione per tutte le discipline di progettazione. Quando sono richieste modifiche al progetto,
- disegni completamente coerenti possono essere generati non appena vengono inserite le modifiche al progetto.
- Collaborazione anticipata di più discipline di progettazione

La tecnologia BIM facilita il lavoro simultaneo di più discipline di progettazione. Mentre la collaborazione con i disegni è anche possibile, è intrinsecamente più difficile e dispendiosa in termini di tempo rispetto al lavoro con uno o più modelli 3D coordinati in cui il controllo delle modifiche può essere ben gestito. Questo accorcia il tempo di progettazione e riduce significativamente gli errori di progettazione e le omissioni. Dà anche una visione anticipata prima dei problemi di progettazione e presenta opportunità per un progetto da migliorare continuamente.

- Facile verifica della coerenza con l'intento del progetto
- Il BIM fornisce visualizzazioni 3D anticipate e quantifica l'area degli spazi e altre quantità di materiale, consentendo stime dei costi più precoci e accurate.

Per gli edifici tecnici (laboratori, ospedali e simili), l'intento progettuale è spesso definito quantitativamente, e questo permette di usare un modello di edificio per controllare questi requisiti.

Per i requisiti qualitativi (questo spazio dovrebbe essere un altro), il modello 3D può anche supportare valutazioni automatiche.

- Estrazione delle stime dei costi durante la fase di progettazione

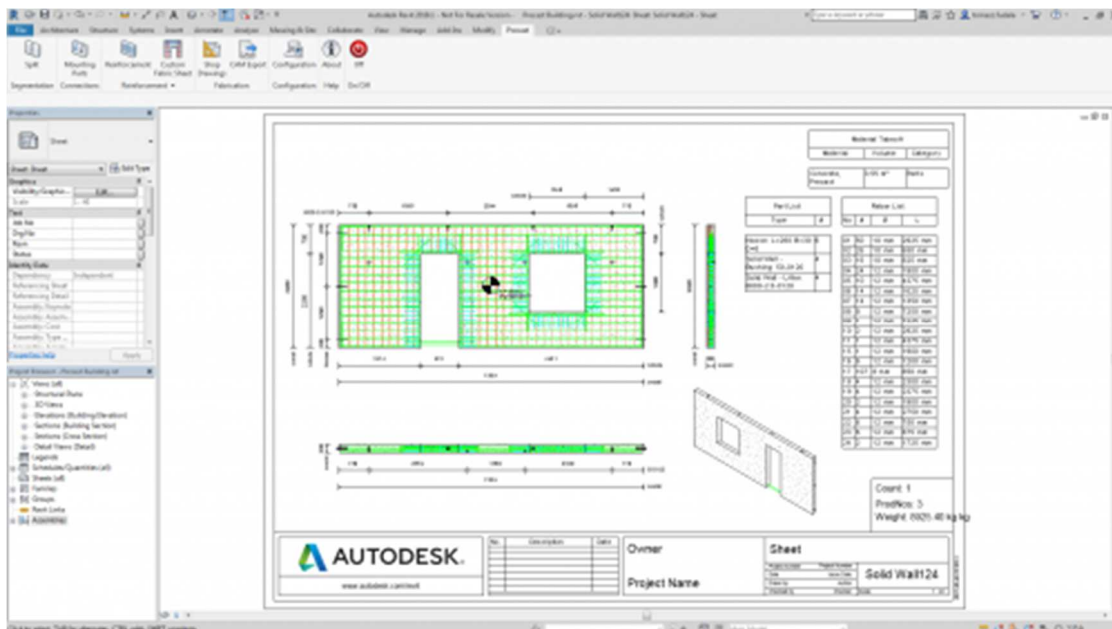
In qualsiasi fase della progettazione, la tecnologia BIM può estrarre un accurato elenco di quantità e spazi che possono essere usati per la stima dei costi. Nelle prime fasi di un progetto, le stime dei costi sono basate o su formule che sono legate a quantità significative del progetto, per esempio, il numero di posti auto, i metri quadrati di aree di lavoro di vario tipo, o costi unitari per metro quadrato. Man mano che il progetto progredisce sono disponibili quantità più dettagliate che possono essere usate per stime dei costi più accurate e dettagliate. Nella fase finale della progettazione, una stima basata sulle quantità di tutti gli oggetti contenuti nel modello permette la preparazione di una stima finale dei costi più accurata. Di conseguenza, è possibile prendere decisioni di progettazione più informate sui costi usando il BIM piuttosto che un sistema basato sulla carta. Quando si usa il BIM per le stime dei costi, è chiaramente desiderabile avere l'appaltatore generale e possibilmente gli appaltatori chiave che saranno responsabili della costruzione della struttura, come parte del progetto team. La loro conoscenza è necessaria per stime accurate dei costi e per la costruibilità durante il processo di progettazione.

- Miglioramento dell'efficienza energetica e della sostenibilità

Collegare il modello dell'edificio a strumenti di analisi energetica permette di valutare l'uso dell'energia durante le prime fasi di progettazione. Questo non è pratico usando i tradizionali strumenti 2D tradizionali a causa del tempo richiesto per preparare l'input pertinente. Se applicata l'analisi energetica viene eseguita alla fine del processo di progettazione 2D come controllo o un requisito normativo, riducendo così le opportunità per le modifiche che potrebbero migliorare le prestazioni energetiche dell'edificio. La capacità di collegare il modello dell'edificio a vari tipi di strumenti di analisi fornisce molte opportunità per migliorare la qualità dell'edificio

1.9 Vantaggi della costruzione e della fabbricazione

Uso del modello di progetto come base per i componenti fabbricati. Se il modello di progetto viene trasferito ad uno strumento di fabbricazione BIM e dettagliato al livello degli oggetti di fabbricazione (modello di negozio), esso conterrà una rappresentazione accurata degli oggetti di costruzione per la fabbricazione e la costruzione. Poiché i componenti sono già definiti in 3D, la loro fabbricazione automatizzata con macchine a controllo numerico è facilitata. Tale automazione è una pratica standard oggi nella fabbricazione dell'acciaio e in alcuni lavori di lamiera. È stata usata con successo nei componenti prefabbricati, nei serramenti e nella fabbricazione del vetro. Questo permette ai fornitori in tutto il mondo di elaborare il modello, di sviluppare i dettagli necessari per la fabbricazione, e di mantenere i collegamenti che riflettono l'intento del progetto. Questo facilita la fabbricazione fuori sede e riduce i costi e i tempi di costruzione. La precisione del BIM permette anche ai componenti più grandi del progetto di essere fabbricati fuori sede rispetto a quanto normalmente tentato usando i disegni 2D, a causa della probabile necessità di modifiche in loco (rilavorazione) e l'incapacità di prevedere le dimensioni esatte fino a quando altri elementi sono costruiti nel campo. Permette anche di avere squadre di installazione più piccole, tempi di installazione più rapidi e meno spazio per lo stoccaggio in loco.



10) Tavola di una parte in C.A prefabbricato usando il software Revit

Reazione rapida ai cambiamenti di design

L'impatto di un cambiamento di progetto suggerito può essere inserito nel modello dell'edificio e le modifiche agli altri oggetti del progetto si aggiorneranno automaticamente.

Alcuni aggiornamenti saranno fatti automaticamente in base alle regole parametriche stabilite. Ulteriori aggiornamenti trasversali possono essere controllati e aggiornati visivamente o attraverso il rilevamento delle collisioni. Le conseguenze di un cambiamento possono essere accuratamente riflesse accuratamente nel modello e in tutte le viste successive di esso. Inoltre, le modifiche al progetto possono essere risolte più rapidamente in un sistema BIM perché le modifiche possono essere condivise, visualizzate, stimate e risolte senza l'uso di lunghe transazioni cartacee.

Scoperta di errori di progettazione e omissioni prima della costruzione

Poiché il modello virtuale dell'edificio 3D è la fonte di tutti i disegni 2D e 3D gli errori di progettazione causati da disegni 2D incoerenti sono eliminati. Inoltre, perché i modelli di tutte le discipline possono essere riuniti e confrontati, le interfacce di più sistemi sono facilmente verificate sia sistematicamente (per scontri di spazio) che visivamente (per altri tipi di errori). Conflitti e problemi di costruzione sono identificati prima che vengano rilevati sul campo. Il coordinamento tra i progettisti e gli appaltatori partecipanti è migliorato e gli errori di omissione sono significativamente ridotti. Questo accelera il processo di costruzione, riduce i costi, minimizza la probabilità di controversie legali e fornisce un processo più fluido per l'intero team del progetto.

Sincronizzazione della pianificazione della progettazione e della costruzione

La pianificazione della costruzione con il CAD 4D richiede il collegamento di un piano di costruzione agli oggetti 3D in un progetto, in modo che sia possibile simulare il processo di costruzione

e mostrare l'aspetto dell'edificio e del sito in qualsiasi momento. Questa simulazione grafica fornisce una visione considerevole di come l'edificio sarà costruito giorno per giorno e rivela fonti di potenziali problemi e opportunità per possibili miglioramenti (sito, personale e attrezzature, conflitti di spazio, problemi di sicurezza e così via). Questo tipo di analisi non è disponibile nei documenti d'offerta cartacei. Tuttavia, fornisce un beneficio aggiunto se il modello include oggetti di costruzione temporanea come puntellamenti, impalcature, gru e altre

attrezzature principali in modo che questi oggetti possano essere collegati alle attività del programma e riflessi nel piano di costruzione desiderato.

Migliore implementazione delle tecniche di costruzione snella

Le tecniche di costruzione snella richiedono un attento coordinamento tra l'appaltatore generale e tutti i subappaltatori per garantire che il lavoro possa essere eseguito quando le risorse appropriate sono disponibili in loco. Questo minimizza gli sforzi sprecati e riduce la necessità di inventari di materiali in loco. Poiché il BIM fornisce un accurato modello del progetto e delle risorse di materiale necessarie per ogni segmento del lavoro, fornisce la base per una migliore pianificazione e programmazione dei subappaltatori e aiuta a garantire l'arrivo just-in-time di persone, attrezzature e materiali.

Questo riduce i costi e permette una migliore collaborazione in cantiere. Il modello può anche essere usato con computer palmari wireless per facilitare la tracciabilità dei materiali, il progresso dell'installazione e il posizionamento automatico nel campo.

Il modello completo dell'edificio fornisce quantità accurate per tutti (o la maggior parte, a seconda del livello di modellazione 3D) dei materiali e degli oggetti contenuti all'interno di un progetto. Queste quantità, specifiche e proprietà possono essere utilizzati per procurarsi i materiali dai fornitori di prodotti e dai subappaltatori (come subappaltatori di calcestruzzo prefabbricato). Al momento attuale (2010), le definizioni degli oggetti per molti prodotti fabbricati non sono ancora state sviluppate per rendere questa capacità una realtà completa. Tuttavia, quando i modelli sono stati disponibili (membri in acciaio membri in acciaio, membri prefabbricati in calcestruzzo, alcuni componenti meccanici, alcune finestre e porte), i risultati sono stati molto positivi.

1.10 Benefici post-costruzione

Miglioramento della messa in funzione e della consegna delle informazioni sulla struttura

Durante il processo di costruzione l'appaltatore generale e gli appaltatori MEP raccolgono informazioni sui materiali installati e sulla manutenzione per i sistemi nell'edificio. Queste informazioni possono essere collegate all'oggetto nel modello dell'edificio e quindi essere disponibili per la consegna al proprietario per l'uso nei loro sistemi di gestione delle strutture. Può anche essere usato per controllare che tutti i sistemi funzionino come progettato prima che l'edificio sia accettato dal proprietario.

Migliore gestione e funzionamento delle strutture

Il modello dell'edificio fornisce una fonte di informazioni (grafici e specifiche) per tutti i sistemi usati in un edificio. Le analisi precedenti usate per determinare attrezzature meccaniche, sistemi di controllo e altri acquisti possono essere forniti al proprietario, come mezzo per verificare le decisioni di progettazione una volta che l'edificio è in uso. Queste informazioni possono essere usate per controllare che tutti i sistemi funzionino correttamente dopo che l'edificio è stato completato.

Integrazione con i sistemi di funzionamento e gestione delle strutture

Un modello dell'edificio che è stato aggiornato con tutti i cambiamenti fatti durante la costruzione fornisce una fonte accurata di informazioni sugli spazi come costruiti e sistemi e fornisce un utile punto di partenza per la gestione e il funzionamento dell'edificio. Un modello informativo dell'edificio supporta il monitoraggio dei sistemi di controllo in tempo reale sistemi di controllo in tempo reale, fornisce un'interfaccia naturale per i sensori e la gestione remota delle strutture. Molte di queste capacità non sono ancora state sviluppate, ma il BIM fornisce una piattaforma ideale per il loro sviluppo.

2. PIATTAFORME BIM

Le piattaforme BIM possono essere utilizzate in diversi modi nella costruzione di edifici: dall'architetto per la modellazione del progetto e la produzione di disegni, da un ingegnere per la gestione dei dati strutturali o energetici, da un appaltatore per sviluppare un modello di coordinamento della costruzione, per i dettagli di fabbricazione o per la gestione della struttura, per esempio; includono vari tipi di funzionalità di strumenti. Alcuni sono commercializzati a più tipi di utenti. Le diverse strategie di marketing portano a pacchetti con diverse collezioni di funzionalità. In questa recensione, non ci affrontiamo questi diversi usi ma consideriamo le principali piattaforme BIM genericamente, dalla prospettiva del suo prodotto primario, con riferimenti ad altri prodotti che girano sulla stessa piattaforma. I loro usi e limitazioni saranno considerati più esplicitamente nei capitoli associati ai diversi tipi di utenti BIM.

2.1 Revit



11) Logo del programma Revit

Consideriamo la piattaforma Revit dal punto di vista di Revit Architecture.

Revit è il più noto e attuale leader di mercato per il BIM nella progettazione architettonica.

progettazione architettonica. È stato introdotto da Autodesk nel 2002 dopo che Autodesk ha acquisito il Revit da una società di startup. Revit è una piattaforma completamente separata da

AutoCAD, con una base di codice e una struttura di file diverse. Revit è una famiglia di prodotti integrati che attualmente comprende Revit Architecture, Revit Structure e Revit MEP.

Come strumento: Revit fornisce un'interfaccia facile da usare, con suggerimenti di trascinamento per ogni operazione e cursore intelligente. I suoi menu sono ben organizzati secondo di lavoro, e i suoi menu operatore rendono in grigio le azioni non disponibili all'interno del contesto di sistema corrente. Il suo supporto per la generazione di disegni è molto buono; la sua produzione di disegni è fortemente associativa, in modo che i rilasci dei disegni siano facilmente gestiti.

Offre un editing bidirezionale dai disegni verso e dal modello, e anche modifica bidirezionale da programmi per porte, ferramenta per porte e simili.

Revit supporta lo sviluppo di nuovi oggetti parametrici personalizzati e la personalizzazione di oggetti predefiniti. Il suo set di regole per la definizione degli oggetti è migliorato con ogni versione e include funzioni trigonometriche. Può vincolare le distanze e angoli e il numero di oggetti in una matrice. Supporta anche relazioni gerarchiche di parametri. Così, un oggetto può essere definito utilizzando un gruppo di sottooggetti con relazioni parametriche. È più difficile impostare parametri globali che possono vincolare gli assemblaggi di layout e dimensioni degli oggetti. La versione attuale API fornisce un buon supporto per lo sviluppo di applicazioni esterne.

I prodotti sono definiti in un misto di tipi di file: RVA, DWG, DWF, DGN, GSM, SKP, IES, e TXT. Sono accessibili dai formati Masterformat, Unifomat e Omniclass Table 23



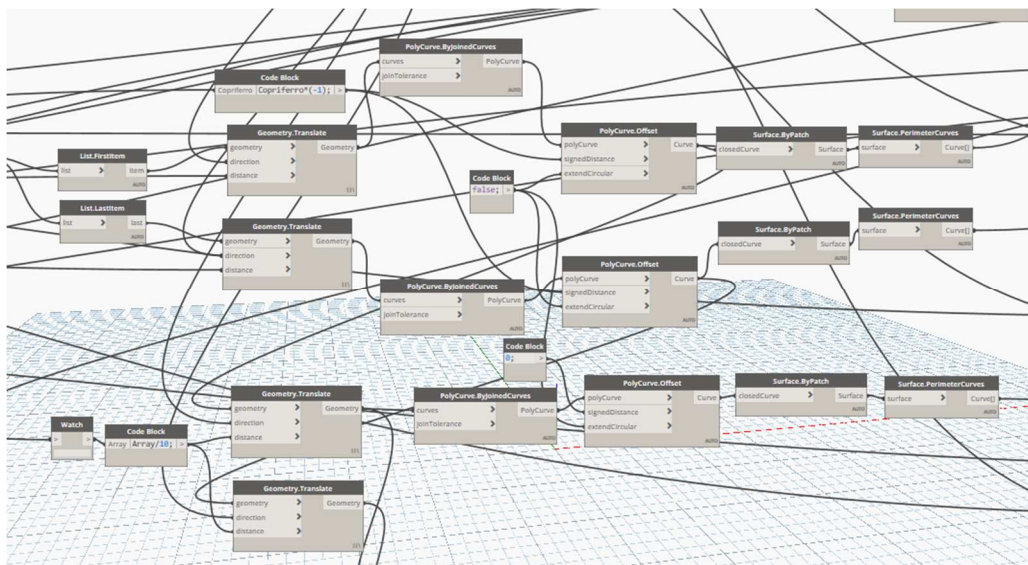
Dynamo

3.1 Che cosa è ?

Dynamo è una piattaforma che consente ai progettisti di immergersi nella programmazione visiva, risolvendo problemi di progetto, automatizzando i processi e ampliando il proprio livello di progettazione.

La progettazione nella quotidianità implica relazioni visive, sistemiche o geometriche tra elementi, consentendo di passare dal concetto al risultato

Questo processo ha alla base l'algoritmo (procedimento che risolve un determinato problema attraverso un numero finito di passi elementari in un tempo ragionevole) esso però può generare cose imprevedute, incontrollate o fantastiche. A favore del progettista un algoritmo è in grado di effettuare grandi quantità di operazioni con estrema precisione, e infinite volte.



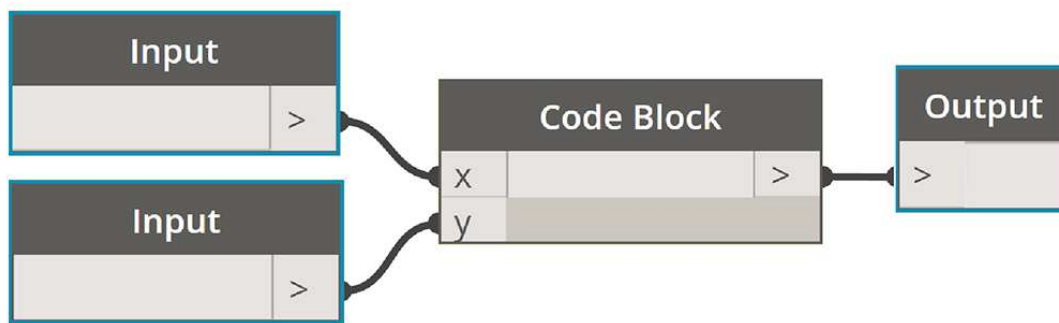
12) Nodi del programma Dynamo

3.2 Su cosa lavora Dynamo ?

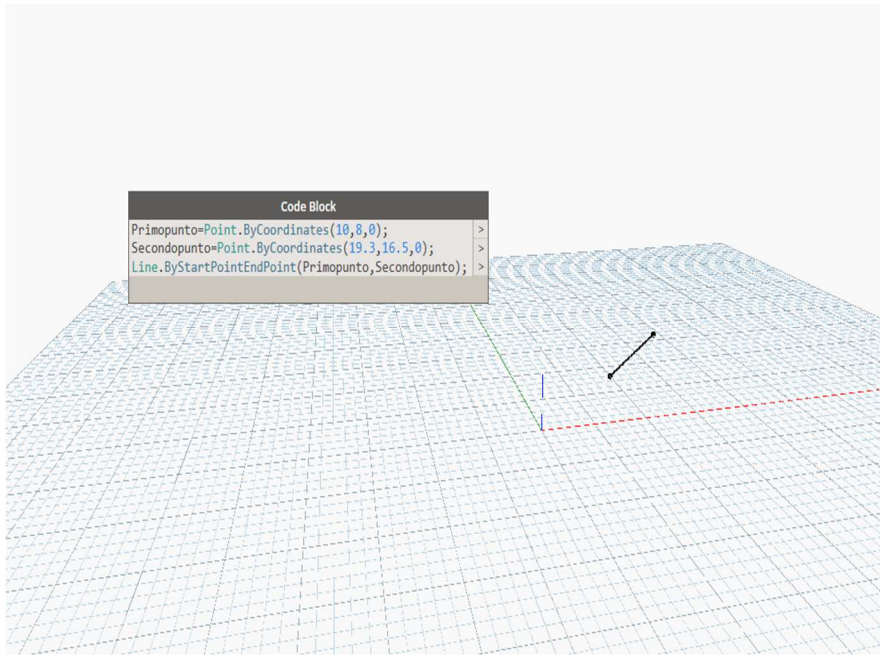
Dynamo lavora su due tipi di informazioni:

- Informazioni grafiche
- Informazioni testuali

Entrambe le informazioni vanno descritte sempre utilizzando una corretta procedura. La differenza sostanziale tra queste due è che le informazioni di tipo grafico è quella di utilizzare nodi che collegati tra loro restituiscono una sorta di gerarchia tra di essi.



Mentre le informazioni testuali, sono righe di codice scritte tramite il linguaggio della programmazione, esse possono essere scritte sia all'interno di Code Block oppure su appositi software di programmazione tra i quali JavaScript, Python o C.

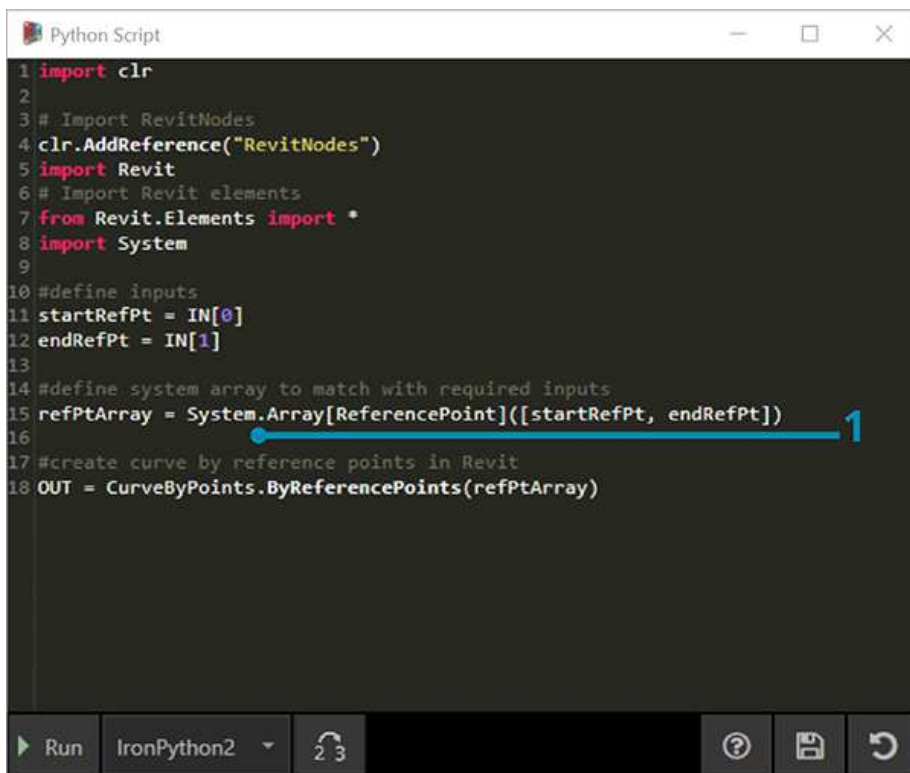


13) Script all'interno del Code Block

Primopunto = Point.ByCoordinates(10,0,0);

Secondopunto = Point.ByCoordinates(19.3,16.5,0);

Line.ByStartEndPoint(Primopunto,SecondoPunto)



14) Esempio di un codice scritto dentro al nodo Python script

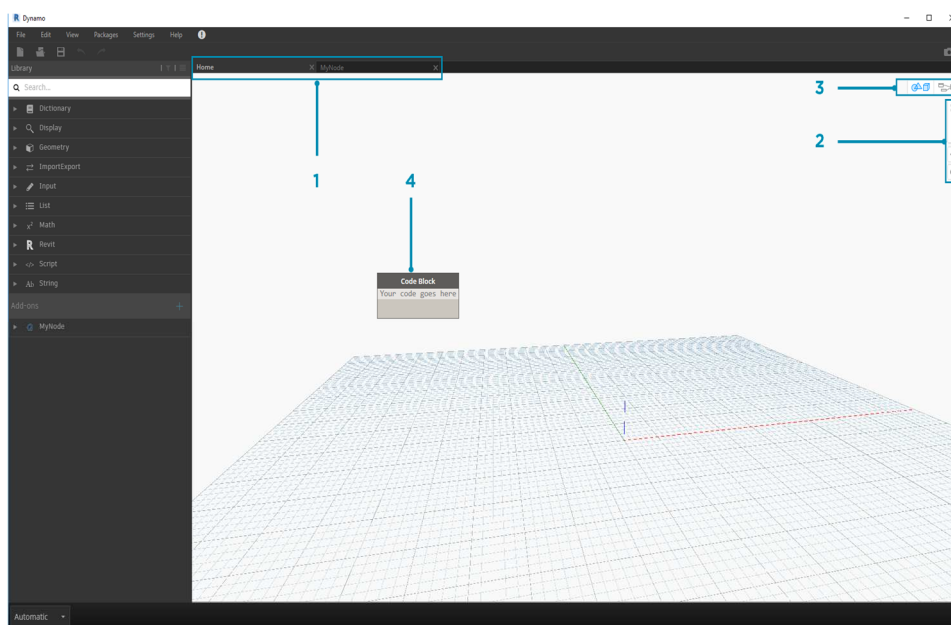
Questi due tipi di programmazione, hanno lo stesso risultato, la prima però porta ad un vantaggio visivo (di una notevole più facilitata creazione di script, anche se non si hanno le basi della programmazione), mentre la seconda porta ad un vantaggio sulla capacità di approfondire e di saper gestire meglio il software nella sua più completa potenzialità.

Questa è la chiave in Dynamo, ovvero poter realizzare degli script che facilitano notevolmente la propria esperienza lavorativa senza sapere in maniera approfondita il linguaggio della programmazione. Si ha un contatto diretto tra la programmazione e il progettista

“Gli utenti possono definire a livello visivo le sequenze di funzionamento, creare parti di codice personalizzate e creare script con vari linguaggi di programmazione testuale.” (Dynamo Primer)

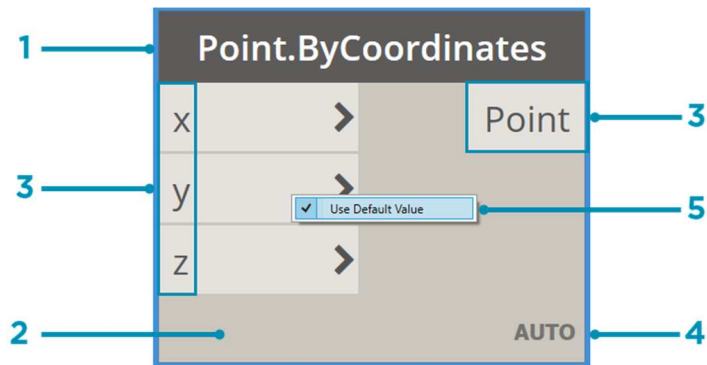
Essenzialmente, Dynamo è una piattaforma di programmazione visiva ed è uno strumento di progettazione flessibile ed espandibile. Poiché può funzionare come applicazione indipendente o come componente aggiuntivo di altro software di progettazione, è utilizzabile per sviluppare un'ampia gamma di workflow creativi.

L'**area di lavoro** di Dynamo è dove si sviluppano i programmi visivi, ma è anche dove viene visualizzata l'anteprima di qualsiasi geometria risultante.



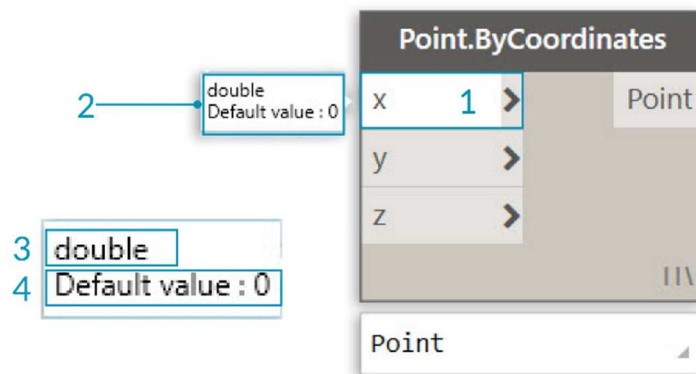
15) Area di lavoro in Dynamo

la struttura di ogni nodo può essere descritta come segue :



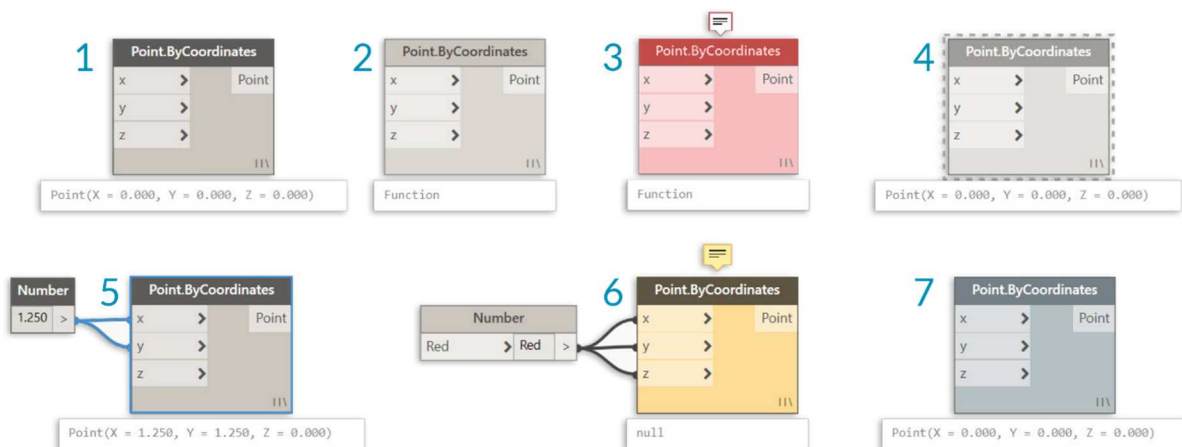
16) nodo in Dynamo

1. Nome: il nome del nodo con una convenzione di denominazione Category.Name.
2. Principale: il corpo principale del nodo. Facendo clic con il pulsante destro del mouse qui vengono visualizzate le opzioni a livello dell'intero nodo.
3. Porte (entrata e uscita): i ricettori dei fili che forniscono i dati di input al nodo e i risultati dell'azione del nodo.
4. Icona di collegamento: indica l'opzione Collegamento specificata per gli input dell'elenco corrispondenti (verranno fornite ulteriori informazioni in seguito).
5. Valore di default: fare clic con il pulsante destro del mouse su una porta di input. Alcuni nodi presentano valori di default che è possibile utilizzare o meno.
6. Si prevede che le porte ricevano dati di un certo tipo. Ad esempio, se si collega un numero come 2.75 alle porte su un nodo Point.ByCoordinates, la creazione di un punto verrà eseguita correttamente. Se tuttavia si specifica *Red* nella stessa porta, verrà generato un errore.
7. Suggerimento: posizionare il cursore su una porta per visualizzare una descrizione comando contenente il tipo di dati previsto.



8.

In Dynamo viene fornita un'indicazione dello stato di esecuzione del programma visivo tramite il rendering dei nodi con diversi schemi di colori basati sullo stato di ogni nodo. Inoltre, se si posiziona il cursore o si fa clic con il pulsante destro del mouse sul nome o sulle porte, vengono visualizzate informazioni e opzioni aggiuntive.



1. Attivo: i nodi con uno sfondo con nome di colore grigio scuro sono collegati correttamente, così come tutti i relativi input.
2. Inattivo: i nodi grigi sono inattivi e devono essere collegati con i fili per far parte del flusso di programma nell'area di lavoro attiva.
3. Stato di errore: il colore rosso indica che il nodo è in uno stato di errore.

4. Congelamento: un nodo trasparente ha l'opzione Congelamento attivata, con conseguente sospensione dell'esecuzione del nodo.
5. Selezionato: i nodi attualmente selezionati hanno un'evidenziazione di color azzurro sul bordo.
6. Avvertimento: i nodi gialli si trovano in uno stato di avviso, pertanto potrebbero contenere tipi di dati errati.
7. Anteprima di sfondo: il colore grigio scuro indica che l'anteprima della geometria è disattivata.

Se il programma visivo contiene avvertimenti o errori, in Dynamo verranno fornite ulteriori informazioni sul problema. Qualsiasi nodo giallo avrà anche una descrizione comando sopra il nome. Posizionare il cursore del mouse sulla descrizione comando per espanderla.

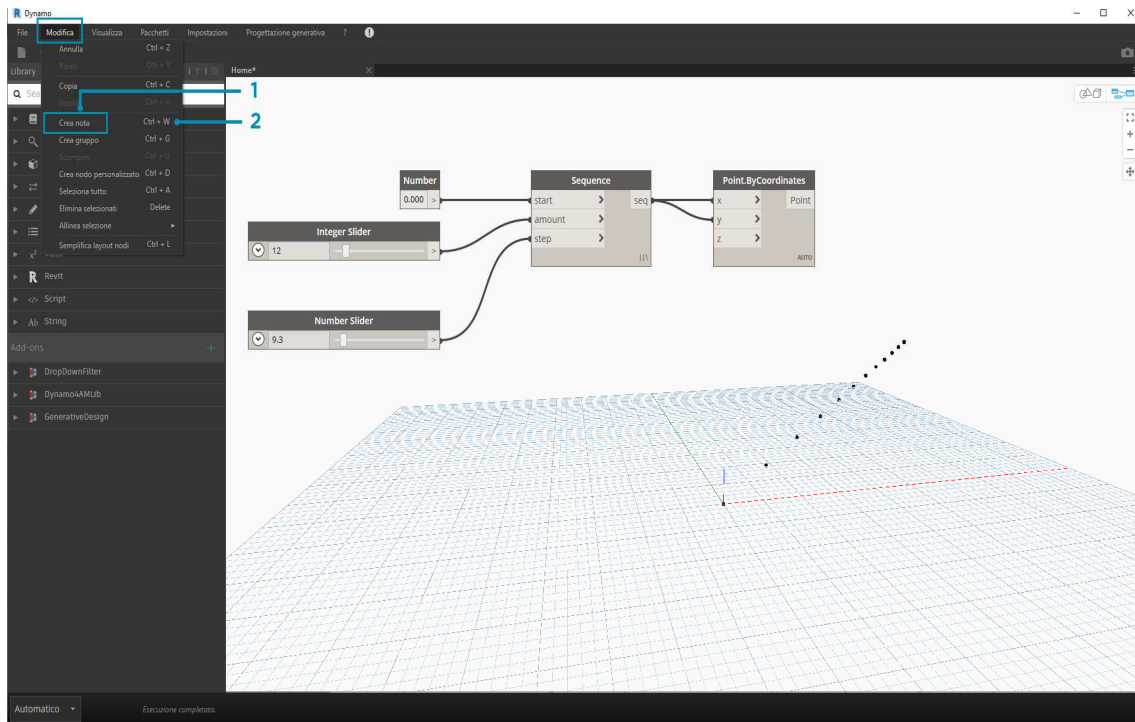
Suggerimento: con queste informazioni sulla descrizione comando, esaminare i nodi a monte per verificare se la struttura di dati o il tipo di dati richiesto è errato.

3.3 Code Block

I nodi Code Block possono essere utilizzati per definire un blocco di codice con righe separate da punti e virgola. Questo può essere semplice come `X/Y`. È inoltre possibile utilizzare i blocchi di codice come scorciatoia per definire un input numerico o chiamare la funzionalità di un altro nodo

3.4 Note

Avendo maturato un po' di esperienza, si potrebbe "leggere" il programma visivo esaminando i nomi dei nodi e seguendo il flusso di programma. Per gli utenti di tutti i livelli di esperienza, è inoltre consigliabile includere etichette e descrizioni con linguaggio semplice. A tale scopo, in Dynamo è disponibile un nodo delle note con un campo di testo modificabile. È possibile aggiungere note all'area di lavoro in due modi:

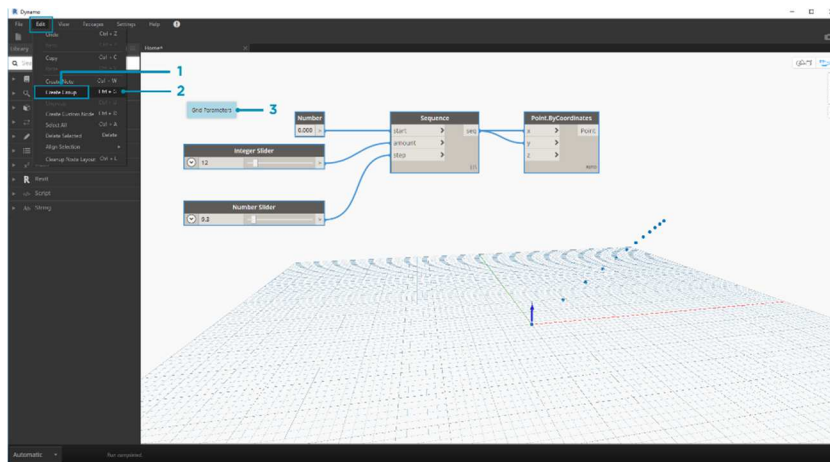


1. Individuare il menu Modifica > Crea nota.
2. Utilizzare il tasto di scelta rapida da tastiera CTRL+W.

Una volta aggiunta la nota all'area di lavoro, verrà visualizzato un campo di testo che consente di modificare il testo nella nota. Una volta creata, è possibile modificare la nota facendo doppio clic o facendo clic con il pulsante destro del mouse sul nodo della nota.

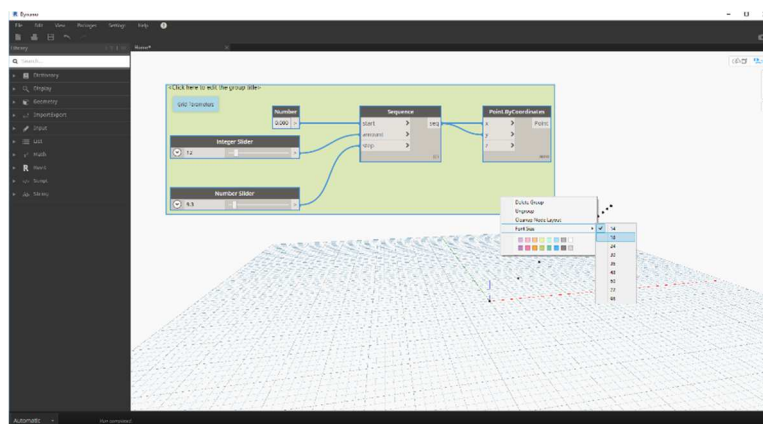
3.5 Raggruppamento

Quando le dimensioni del programma visivo aumentano, è utile identificare i passaggi più lunghi che verranno eseguiti. È possibile evidenziare raccolte di nodi di dimensioni maggiori con un gruppo per etichettarle con un rettangolo colorato nello sfondo e un titolo. Esistono tre modi per creare un gruppo con più di un nodo selezionato:



1. Individuare il menu Modifica > Crea gruppo.
2. Utilizzare il tasto di scelta rapida da tastiera CTRL+G.
3. Fare clic con il pulsante destro del mouse sull'area di lavoro e selezionare Crea gruppo.

Una volta creato un gruppo, è possibile modificarne le impostazioni, come titolo e colore.



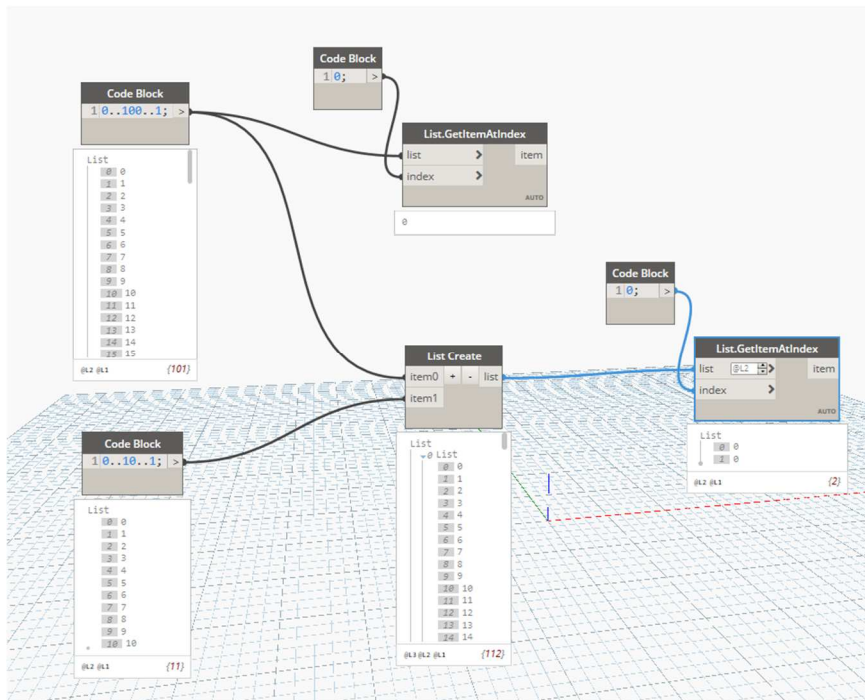
3.6 Cosa sono i dati?

I dati sono un insieme di valori di variabili qualitative o quantitative. Il formato più semplice di dati è costituito da numeri quali 0, 3.14 o 17. Tuttavia, i dati possono anche essere di diversi tipi: una variabile che rappresenta numeri che cambiano (`height`), caratteri (`myName`), geometria (`Circle`) o un elenco di elementi di dati (1, 2, 3, 5, 8, 13,...). Sono necessari dati da aggiungere alle porte di input dei nodi di Dynamo: è possibile disporre di dati senza azioni, ma sono necessari dati per elaborare le azioni rappresentate dai nodi. Quando è stato aggiunto un nodo all'area di lavoro, se non sono stati forniti eventuali input, il risultato sarà una funzione, non il risultato dell'azione stessa.

Il tipo "`null`" rappresenta l'assenza di dati. Sebbene si tratti di un concetto astratto, è probabile incontrarlo utilizzando la programmazione visiva. Se un'azione non crea un risultato valido, il nodo restituirà un valore `null`. I test per i valori `null` e la rimozione dei valori `null` dalla struttura dei dati sono una parte fondamentale per la creazione di programmi efficienti.

il ruolo delle Strutture di dati, è quello di organizzare gli schemi. Le specifiche delle Strutture di Dati e come usarle variano da linguaggio di programmazione a linguaggio di programmazione. In Dynamo, aggiungiamo la gerarchia ai nostri dati attraverso le Liste.

Una lista rappresenta una collezione di elementi inseriti in una struttura di dati:



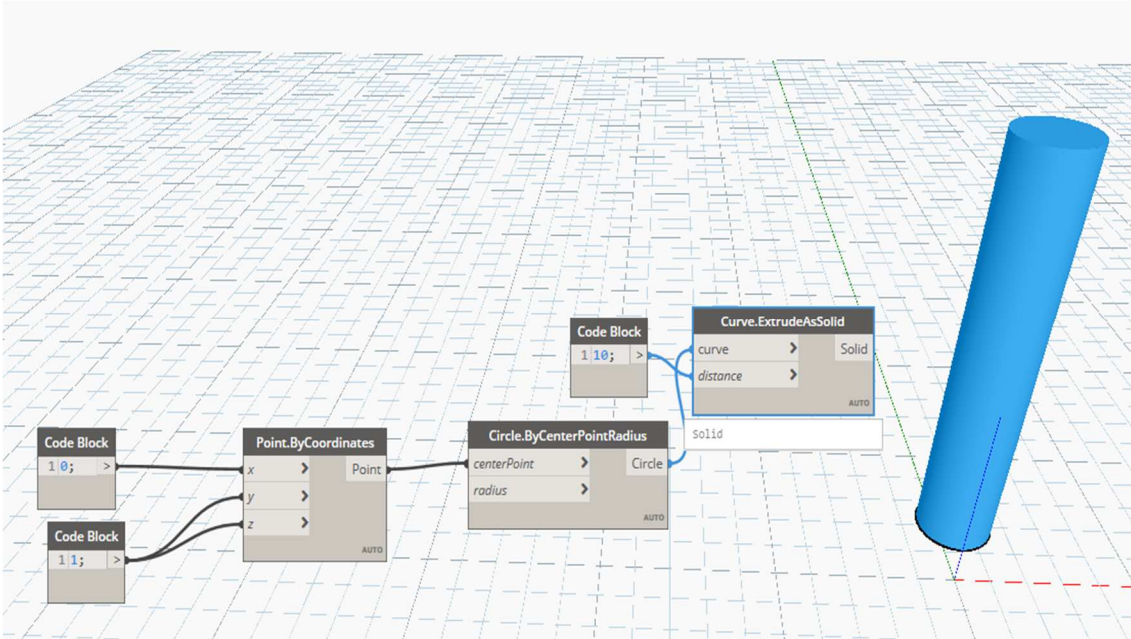
17) Script in Dynamo per la creazione di una lista di dati

In questo esempio ho creato due liste all'interno di due code block nel primo abbiamo una lista di 101 elementi che corrispondono a valori tra 0 e 100 mentre nel secondo 11 elementi con valori compresi tra 0 e 10. Con il nodo `GetItemAtIndex` possiamo specificando la lista prendere il dato corrispondente all'indice specificato, mentre con il nodo `list.Create` possiamo unire più liste in una sola.

Le liste sono considerate elementi. Ed è possibile tramite la finestra sottostante al nodo, scorrere tutti gli elementi che sono all'interno di quel nodo.

Nel secondo esempio ho creato mediante il nodo `Point.ByCoordinates` un punto nello spazio specificando le sue coordinate.

Con il nodo successivo `Circle.ByPointsRadius` tramite il punto appena creato si crea una circonferenza e tramite `Curve.ExtrudeSolid` si crea un solido come estrusione della circonferenza stessa.



18) Script in Dynamo per la creazione di un cilindro

2020 ▾ 

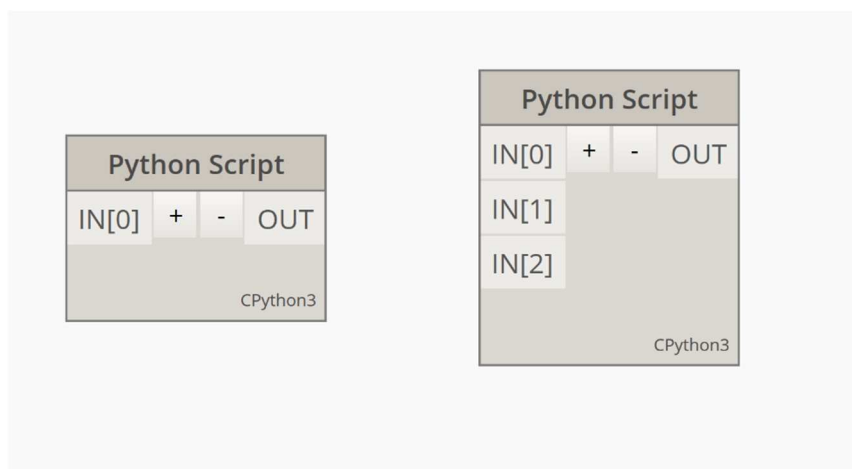
4.1 Introduzione alle API di Revit

API è un acronimo in inglese che significa Interfaccia di programmazione dell' applicazione (application programming interface).

Questo è un accesso specifico per la programmazione che permette di utilizzare la struttura interna dell'applicazione.

La maggior parte degli utenti di Revit lo utilizza tramite la sua interfaccia, cliccando sui comandi del programma , selezionando gli elementi del menù.

Le API non sono applicazioni, non sono un archivio e non hanno un'interfaccia grafica , tuttavia ci sono ambienti di scrittura che permettono di accedervi. Si tratta ad esempio il nodo Python Script all'interno di Dynamo



19) Nodo Python Script

Accedere a un programma tramite le sue API significa entrare nel nucleo centrale del programma, si possono utilizzare tutti i comandi possibili dell'applicazione e creare script personalizzati

Imparare a maneggiare un'applicazione di programmazione ha una serie di benefici :

- Automatizzare elementi di routine
- Costruire o automatizzare le informazioni
- Realizzare operazioni matematiche complesse o analisi dei dati
- Creare operazioni completamente personalizzate, che permettono di creare nuove funzionalità del programma

Le API di Revit sono accessibili dalla versione 2000 e si possono utilizzare per controllare tutti gli aspetti del software.

In realtà quando si scrivono gli script digitali si sta solo richiamando una serie di comandi che risiedono nelle API del programma

```
Python Script : Creazione di un muro
1 import clr
2
3 clr.AddReference('ProtoGeometry')
4 from Autodesk.DesignScript.Geometry import *
5
6 clr.AddReference('RevitNodes')
7
8
9 clr.AddReference('RevitServices')
10 from RevitServices.Persistence import DocumentManager
11 from RevitServices.Transactions import TransactionManager
12
13 clr.AddReference('RevitAPI')
14 clr.AddReference('RevitAPIUI')
15 import Autodesk
16 from Autodesk.Revit.DB import *
17 from Autodesk.Revit.UI import *
18
19 doc = DocumentManager.Instance.CurrentDBDocument
20
21 livellobase = UnwrapElement(IN[0])
22 livellosuperiore = UnwrapElement(IN[1])
23 tipodimuro = UnwrapElement(IN[2])
24
25 punto1 = XYZ(0,0,0)
26 punto2 = XYZ(10,0,0)
27 TransactionManager.Instance.EnsureInTransaction(doc)
28 linea = Line.CreateBound(punto1, punto2)
29 muro = Wall.Create(doc, linea, livellobase.Id, False)
30 muro.WallType = tipodimuro
31 vincolosuperiore = muro.get_Parameter(BuiltInParameter.WALL_HEIGHT_TYPE)
32 vincolosuperiore.Set(livellosuperiore.Id)
33
34 TransactionManager.Instance.TransactionTaskDone()
35 OUT = muro
36
```

20) Ambiente di lavoro del nodo Python Script

4.2 Struttura delle API di Revit

Le API sono strutturate in questo modo :

- 1700 classi con i loro rispettivi metodi e proprietà
- 50 interfacce
- 500 enumerazioni

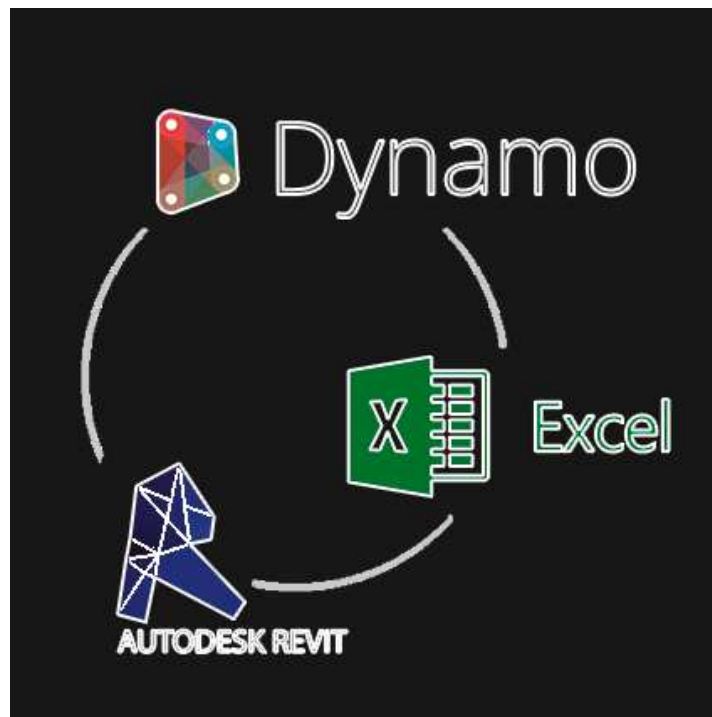
Le classi delle API si trovano nell'archivio RevitAPI.dll e RevitAPIUI.dll

4.3 introduzione a .NET

.NET è un insieme di risorse di sviluppo del software per il sistema operativo di Windows

Essenzialmente tutte le applicazioni di windows hanno un'estensione della biblioteca .NET. essa contiene molte biblioteche per la gestione dei dati.

Questo significa che Revit, Dynamo e molte altre applicazioni (per esempio Excel e Outlook si possono utilizzare insieme perché riescono a leggersi a vicenda.



21) Interazione tra Revit, Dynamo e excel

4.4 Modello di Python Script

```
import clr
import sys
sys.path.append('C:/Program Files...')
import System
from System import Array
from System.Collections.Generic import *
clr.AddReference('ProtoGeometry')
from Autodesk.DesignScript.Geometry import *

clr.AddReference('RevitNodes')
import Revit
clr.ImportExtension(Revit.Elements)
clr.ImportExtension(Revit.Geometry)

clr.AddReference('RevitServices')
from RevitServices.Persistence import DocumentManager
from RevitServices.Transactions import TransactionManager

clr.AddReference('RevitAPI')
clr.AddReference('RevitAPIUI')
import Autodesk
from Autodesk.Revit.DB import *
from Autodesk.Revit.UI import *

doc = DocumentManager.Instance.CurrentDBDocument
uiapp = DocumentManager.Instance.CurrentUIApplication
app = uiapp.Application
uidoc = uiapp.ActiveUIDocument
```

22) Modello Python Script

- Clr : è il Common Language Runtime di .Net , è l'ambiente di esecuzione che permette di eseguire i codici dei vari linguaggi di programmazione
- Sys : è una biblioteca fondamentale di Python che si utilizza per caricare le biblioteche la cui estensione è IronPython

- System : lo spazio dei nomi di sistema nella radice .NET, all'inizio si importa la biblioteca Array e Collection, che permettono di lavorare con matrici e collezioni generiche

Riferimenti :

- ProtoGeometry : è una biblioteca di Dynamo pensata per interagire con la geometria di dynamo
- RevitNodes Carica i nodi di Dynamo, le biblioteche di elementi di Revit in Dynamo e la conversione della geometria
- RevitServices : Permette di gestire i documenti e modificarli tramite DocumentManager e TransactionManager
- RevitAPI : inserisce le classi dall'archivio dll
- RevitAPIUI : permette di accedere all' interfaccia del programma

Identificatori :

- CurrentDBDocument : identificatore del documento attuale di Revit
- CurrentUIApplication : identificatore dell'interfaccia dell'utente del documento attivo di Revit
- Application : identificatore dell'applicazione di Revit aperta
- ActiveUIDocument : identificatore dell'interfaccia dell'utente del documento aperto

4.5 Lettura documentazione delle API di Revit

- Revit API docs

The screenshot displays the Revit API documentation interface. At the top, a navigation bar (1) shows software versions from 2017.1 to 2022, with 'Code' selected. Below this is a search bar (4) with the text 'Unchanged' and a magnifying glass icon. The left sidebar (2) lists various API classes, with 'FilteredElementCollector Class' highlighted. The main content area (3) shows the details for 'FilteredElementCollector Class', including a description, syntax for C#, Visual Basic, and Visual C++, and a 'Remarks' section.

Autodesk Revit DB

FilteredElementCollector Class

Members | See Also

This class is used to search, filter and iterate through a set of elements.

Syntax

C#

```
public class FilteredElementCollector : IEnumerable<Element>,
    IDisposable
```

Visual Basic

```
Public Class FilteredElementCollector _
    Implements IEnumerable(Of Element), IDisposable
```

Visual C++

```
public ref class FilteredElementCollector : IEnumerable<Element^>,
    IDisposable
```

Remarks

Developers can assign a variety of conditions to filter the elements that are returned. This class re
Revit will attempt to organize the filters in order to minimize expansion of elements regardless of
There are three groups of methods that you can use on a given collector once you have applied fi

23) Pagina internet di Revit Api Docs

- 1- Versione del Software
- 2- Navigazione delle classi di Revit
- 3- Contenuto delle classi
- 4- Barra di ricerca

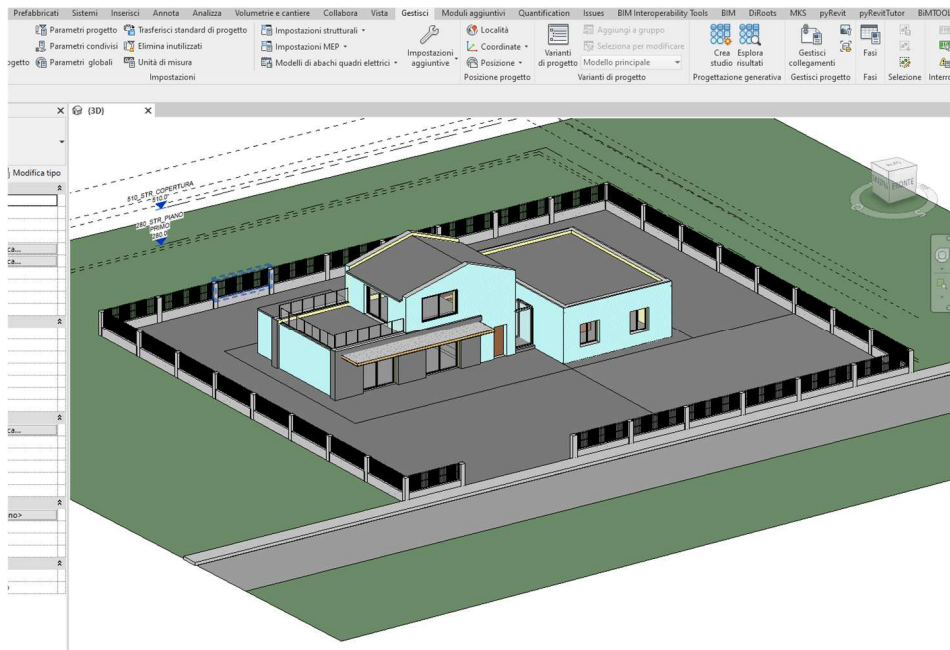
- Membri : sono tutti i metodi, proprietà, costruttori e elenchi di una classe riuniti in una sola pagina
- Metodi : sono le funzioni che si possono incontrare dentro le istanze di ciascuno oggetto
- Proprietà : sono come gli aggettivi di una classe, una proprietà fornisce un meccanismo flessibile non solo per leggere, ma per scrivere e calcolare i valori di un campo
- Costruttori : dicono come creare esempi di ciascuna classe
- Elenchi : sono liste codificate che generalmente sono destinate a non cambiare e a restringere la scelta a un insieme di opzioni

Alcuni elenchi chiave delle API sono :

- BuiltInCategory : una lista di tutte le BuiltInCategory in Revit
- BuiltInParameter : una lista di tutti i BuiltInParameter in Revit
- ViewType : una lista di tutti i tipi di vista disponibile
- ParameterType : una lista di tutti i parametri disponibili
- DisplayUnitType : tutte le unità disponibili in Revit

4.6 Filtraggio di elementi : Muri

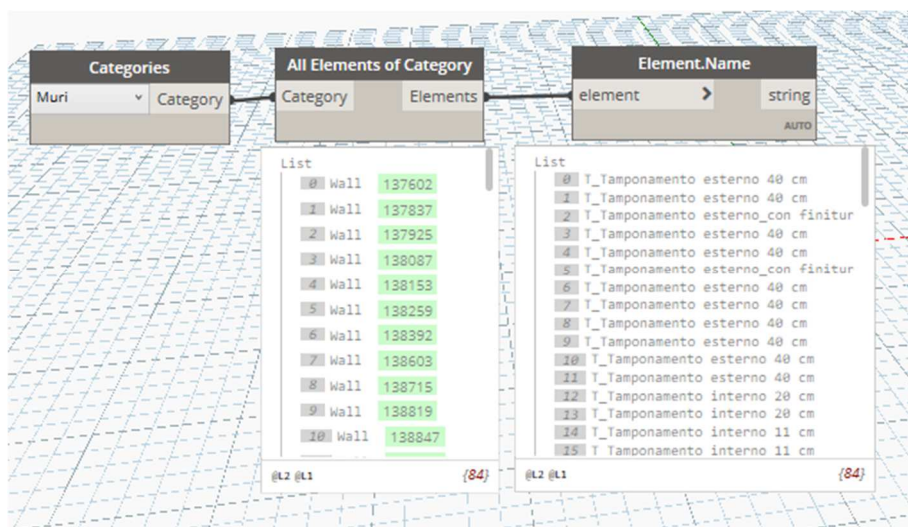
Prendendo un edificio (in foto) come esempio andremo ad addentrarci nei nodi di dynamo e a vedere come effettivamente possiamo ricreare tali nodi con degli script di python grazie all'utilizzo delle API



24) Modello di un edificio, creato con Autodesk Revit

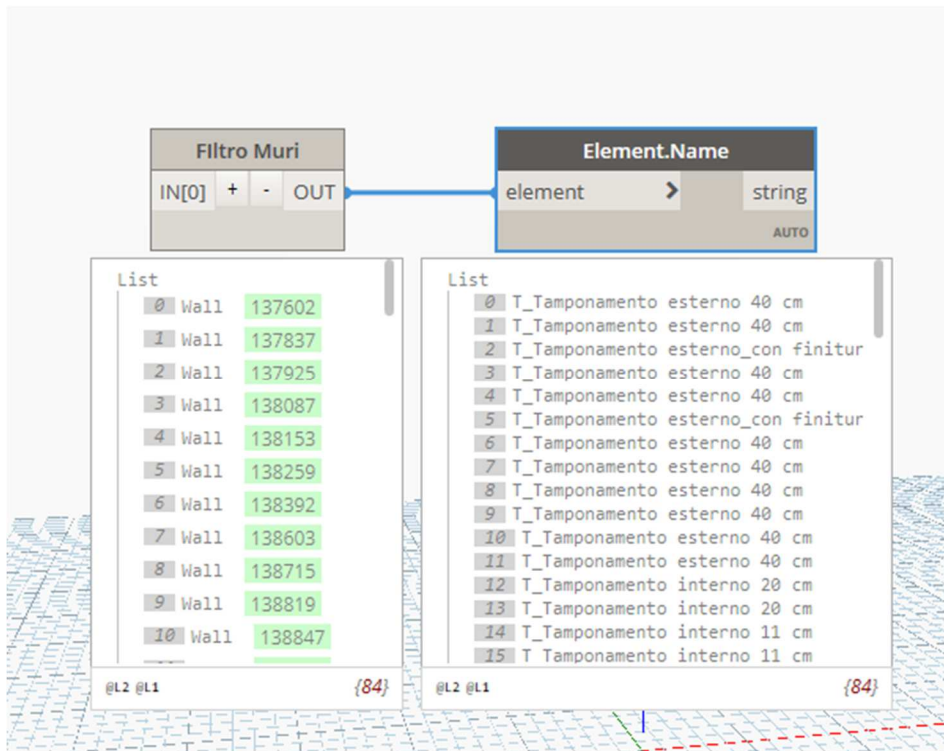
Con Dynamo andremo a utilizzare i seguenti nodi per poter filtrare tutti i muri architettonici presenti nel nostro progetto :

- Categories
- All Element of category
- Element Name



25) Nodi in Dynamo

Otteniamo una lista di 84 elementi Wall con il relativo codice Id e grazie al nodo Element.Name leggiamo il nome relativo all'istanza di muro



26) Nodi Python script rinominato : Filtro Muri

Ora con il nodo di Python Script rinominato Filtro Muri, otteniamo lo stesso risultato

```

1
2 import clr
3 import sys
4 sys.path.append('C:/Program Files...')
5 import System
6 from System import Array
7 from System.Collections.Generic import *
8 clr.AddReference('ProtoGeometry')
9 from Autodesk.DesignScript.Geometry import *
10
11 clr.AddReference('RevitNodes')
12 import Revit
13
14
15 clr.AddReference('RevitServices')
16 from RevitServices.Persistence import DocumentManager
17 from RevitServices.Transactions import TransactionManager
18
19 clr.AddReference('RevitAPI')
20 clr.AddReference('RevitAPIUI')
21 import Autodesk
22 from Autodesk.Revit.DB import *
23 from Autodesk.Revit.UI import *
24
25 doc = DocumentManager.Instance.CurrentDBDocument
26
27
28 muri = FilteredElementCollector(doc).OfCategory(BuiltInCategory.OST_Walls).WhereElementIsNotElementType().ToElements()
29 OUT = muri
  
```

27) Script Python

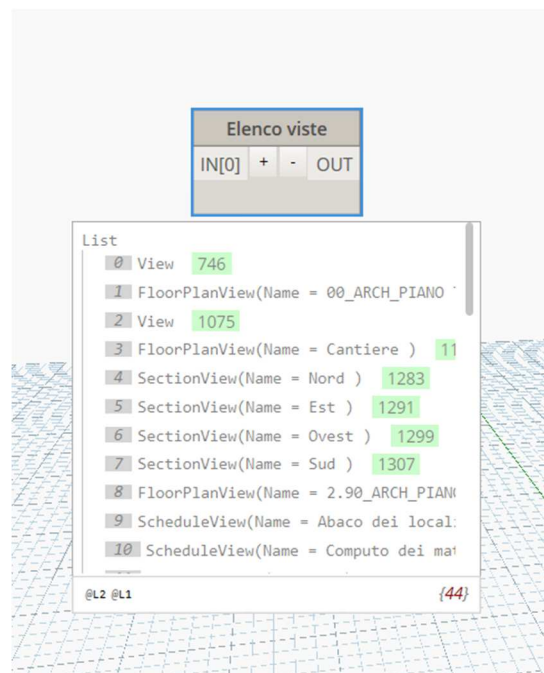
Tralasciando la prima parte dello script che serve per richiamare tutte le librerie e le classi di Revit, la parte che viene usata per filtrare i muri sostanzialmente è :

```
- muri =  
    FilteredElementCollector(doc).OfCategory(BuiltInCategory.OST_Walls).WhereElementIs  
    NotElementType().ToElements()
```

Questa riga di codice ci consente di filtrare prima tutta la categoria che riguarda ai muri quindi sia tipi che istanze poi tramite la proprietà WhereElementIsNotElementType riusciamo a eliminare dal filtro i tipi di muro così da ottenere solo le istanze degli elementi

4.7 Elenco di tutte le viste presenti nel disegno

Con lo stesso ragionamento che abbiamo usato per filtrare i muri possiamo usare la stessa classe FilteredElementCollector per creare un elenco di tutte le viste create nel nostro progetto



28) Nodo Python Script rinominato : Elenco viste

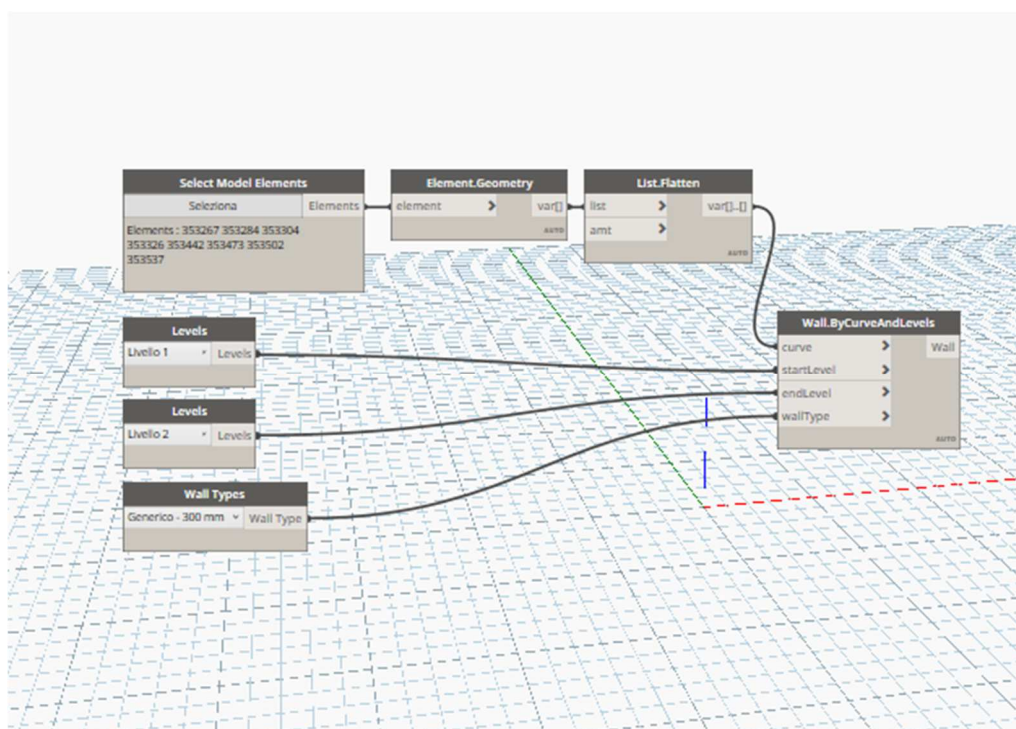
```
Elenco viste
1 import clr
2 import sys
3
4 import System
5 from System import Array
6 from System.Collections.Generic import *
7 clr.AddReference('ProtoGeometry')
8 from Autodesk.DesignScript.Geometry import *
9
10 clr.AddReference('RevitNodes')
11 import Revit
12
13
14 clr.AddReference('RevitServices')
15 from RevitServices.Persistence import DocumentManager
16 from RevitServices.Transactions import TransactionManager
17
18 clr.AddReference('RevitAPI')
19 clr.AddReference('RevitAPIUI')
20 import Autodesk
21 from Autodesk.Revit.DB import *
22 from Autodesk.Revit.UI import *
23
24 doc = DocumentManager.Instance.CurrentDBDocument
25 viste = FilteredElementCollector(doc).OfClass(View).ToElements()
26 OUT = viste
```

29) Script Python

Viste = FilteredElementCollector(doc).OfClass(View).ToElements()

4.8 Creazione di elementi : Muri

In questa parte invece andremo a creare una serie di muri partendo da delle linee di modello disegnate nel programma e a inserire degli input nel il nodo Wall.ByCurveAndLevels

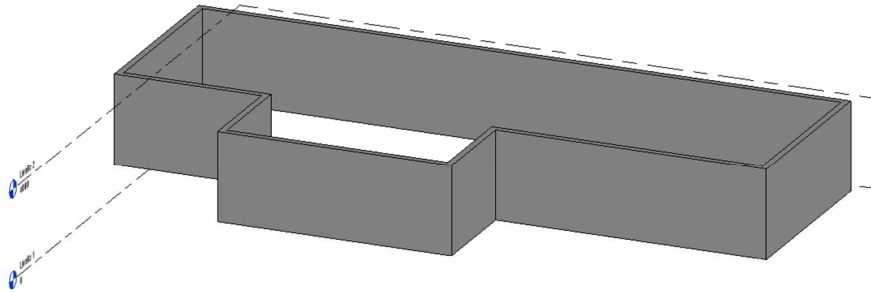


30) Nodi in Dynamo

Grazie al nodo Select Model Elements possiamo selezionare le curve di modello create nel programma , successivamente lo colleghiamo al nodo ELeMent.Geometry per ottenerne la geometria e tramite gli altri input :

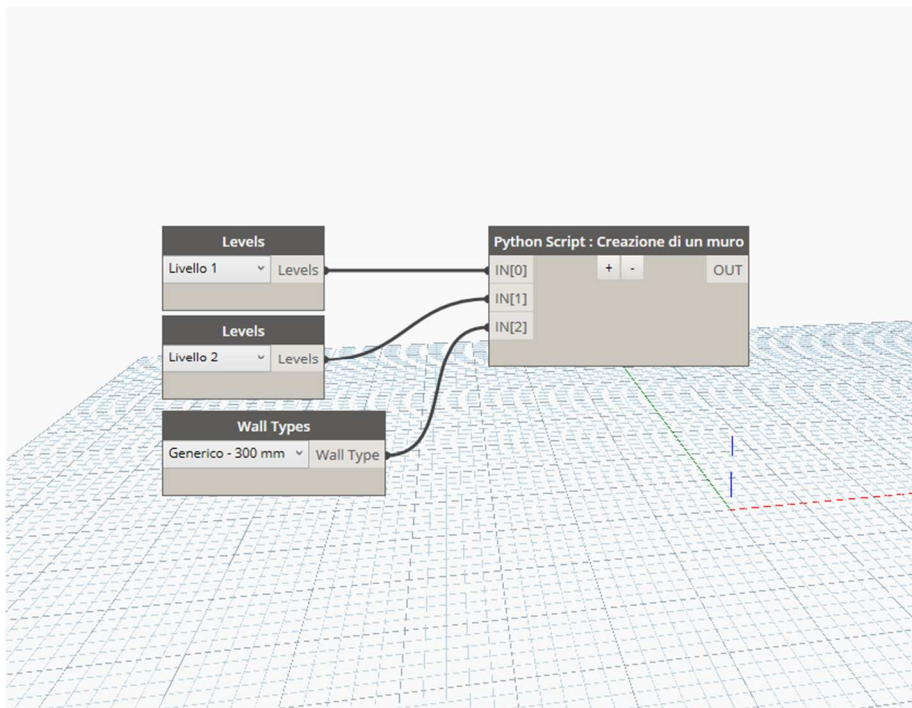
- Levels : che stanno ad indicare i due vincoli di associazione del muro (da dove parte fino a che livello arriva)
- Wall Types : indica il tipo di muro che si vuole creare

Li colleghiamo tutti al nodo di creazione del muro e otteniamo il seguente risultato



31) Muri nell'ambiente Revit

Otteniamo il medesimo risultato sempre tramite un nodo di python script che ho rinominato Python script : Creazione di un muro



32) Nodo Python Script rinominato : creazione di un muro

Anche in questo caso gli input sono :

- Levels : che indicano sempre i due vincoli di associazione
- Wall Types : che indica sempre il tipo di muro che si vuole creare

```
Python Script : Creazione di un muro
1 import clr
2
3 clr.AddReference('ProtoGeometry')
4 from Autodesk.DesignScript.Geometry import *
5
6 clr.AddReference('RevitNodes')
7
8
9 clr.AddReference('RevitServices')
10 from RevitServices.Persistence import DocumentManager
11 from RevitServices.Transactions import TransactionManager
12
13 clr.AddReference('RevitAPI')
14 clr.AddReference('RevitAPIUI')
15 import Autodesk
16 from Autodesk.Revit.DB import *
17 from Autodesk.Revit.UI import *
18
19 doc = DocumentManager.Instance.CurrentDBDocument
20
21 livellobase = UnwrapElement(IN[0])
22 livellosuperiore = UnwrapElement(IN[1])
23 tipodimuro = UnwrapElement(IN[2])
24
25 punto1 = XYZ(0,0,0)
26 punto2 = XYZ(10,0,0)
27 TransactionManager.Instance.EnsureInTransaction(doc)
28 linea = Line.CreateBound(punto1, punto2)
29 muro = Wall.Create(doc, linea, livellobase.Id, False)
30 muro.WallType = tipodimuro
31 vincolosuperiore = muro.get_Parameter(BuiltInParameter.WALL_HEIGHT_TYPE)
32 vincolosuperiore.Set(livellosuperiore.Id)
33
34 TransactionManager.Instance.TransactionTaskDone()
35 OUT = muro
36
```

33) Python Script

La parte centrale del nodo è quella di utilizzare la classe Wall.Create che ci permette di creare delle istanze di muro partendo da linee e dal livello di base e poi tramite la proprietà Set possiamo associare il vincolo ad un livello a nostra scelta

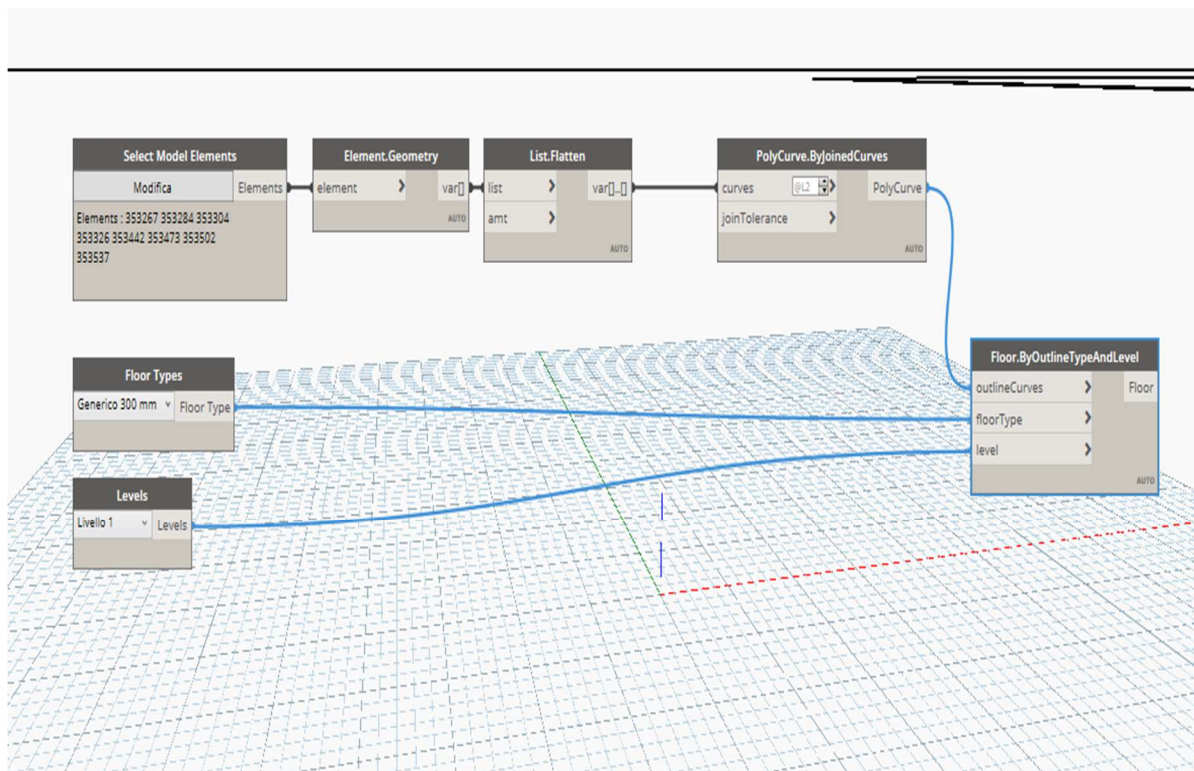
```
Muro = Wall.Create(doc, linea, livellobase.Id, false)
```

```
Vincolosuperiore.Set(livellosuperiore.Id
```

4.9 Creazione di elementi : Solaio

Per la creazione di un solaio in Dynamo utilizzeremo i seguenti nodi :

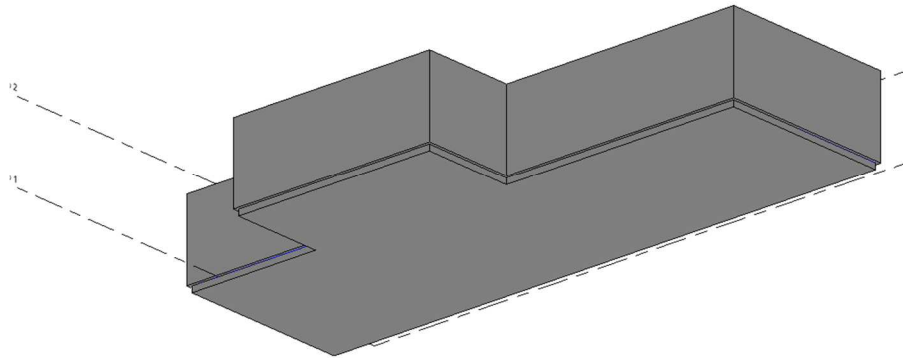
- Select Model Elements
- Element.Geometry
- PolyCurve.ByJoinedCurves
- FloorTypes
- Levels
- Floor.ByOutlineTypeAndLevel



34) Nodi in Dynamo

Grazie al nodo Select Model Elements selezioniamo le linee di modello create nel nostro programma e con Element.Geometry ne otteniamo la geometria per poi tramite il nodo PolyCurve.ByJoinedCurve rendere tutte le linee di modello un'unica curva per poter creare il contorno del nostro solaio

Tramite gli altri due input Floor Types (tipo di solaio desiderato) e Levels (livello selezionato) , possiamo creare il nostro solaio con il nodo Floor.ByOutlineTypeAndLevel

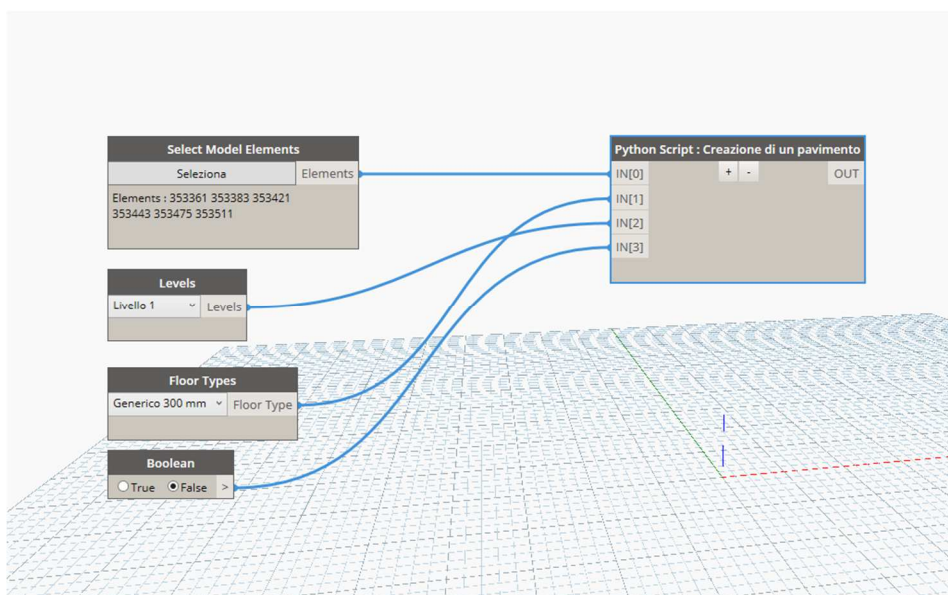


35) Solaio nell'ambiente Revit

Allo stesso modo utilizzando il nodo Python script rinominato Python Script : Creazione di un pavimento otteniamo lo stesso risultato

Gli input sono sempre :

- Select Model Elements
- Levels
- Floor Types
- Boolean : indica se il solaio è strutturale (True) oppure no (False)



36) Nodo Python Script rinominato : creazione di un pavimento

```
Python Script : Creazione di un pavimento
1 import clr
2
3 clr.AddReference('ProtoGeometry')
4 from Autodesk.DesignScript.Geometry import *
5
6 clr.AddReference('RevitNodes')
7
8
9 clr.AddReference('RevitServices')
10 from RevitServices.Persistence import DocumentManager
11 from RevitServices.Transactions import TransactionManager
12
13 clr.AddReference('RevitAPI')
14 clr.AddReference('RevitAPIUI')
15 import Autodesk
16 from Autodesk.Revit.DB import *
17 from Autodesk.Revit.UI import *
18
19 doc = DocumentManager.Instance.CurrentDBDocument
20 Curve = UnwrapElement(IN[0])
21 Tipodipavimento = UnwrapElement(IN[1])
22 livello = UnwrapElement(IN[2])
23 strutturale = IN[3]
24
25 Carray = CurveArray()
26 for c in Curve :
27     Carray.Append(c.GeometryCurve)
28
29 Out = []
30 TransactionManager.Instance.EnsureInTransaction(doc)
31 nuovopavimento = doc.Create.NewFloor(Carray, Tipodipavimento, livello, strutturale)
32
33
34 TransactionManager.Instance.TransactionTaskDone()
35 OUT = nuovopavimento
36
```

37) Python Script

Il nodo consiste nel utilizzare la classe Create. NewFloor che tramite la curva , il tipo di pavimento , livello e il valore booleano ci consente di creare il nostro pavimento

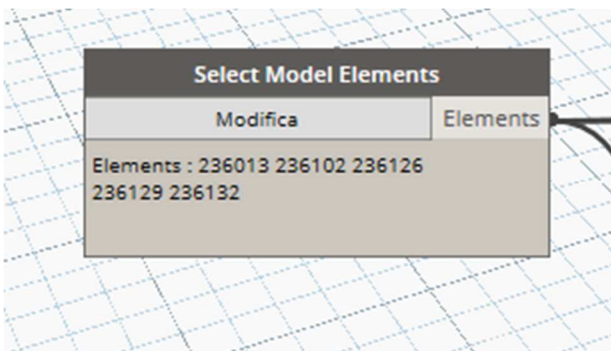
```
Nuovopavimento = doc.Create.NewFloor(Carray, Tipodipavimento, livello , strutturale)
```

5. Dynamo per la parte Architettonica

In questo script da me progettato, ho voluto creare in maniera automatica piante e prospetti di porte e finestre (nell'esempio qui presente riporto solo il posizionamento delle porte) all'interno di un progetto architettonico. Per poter eseguire questo script mi sono servito di alcuni pacchetti :

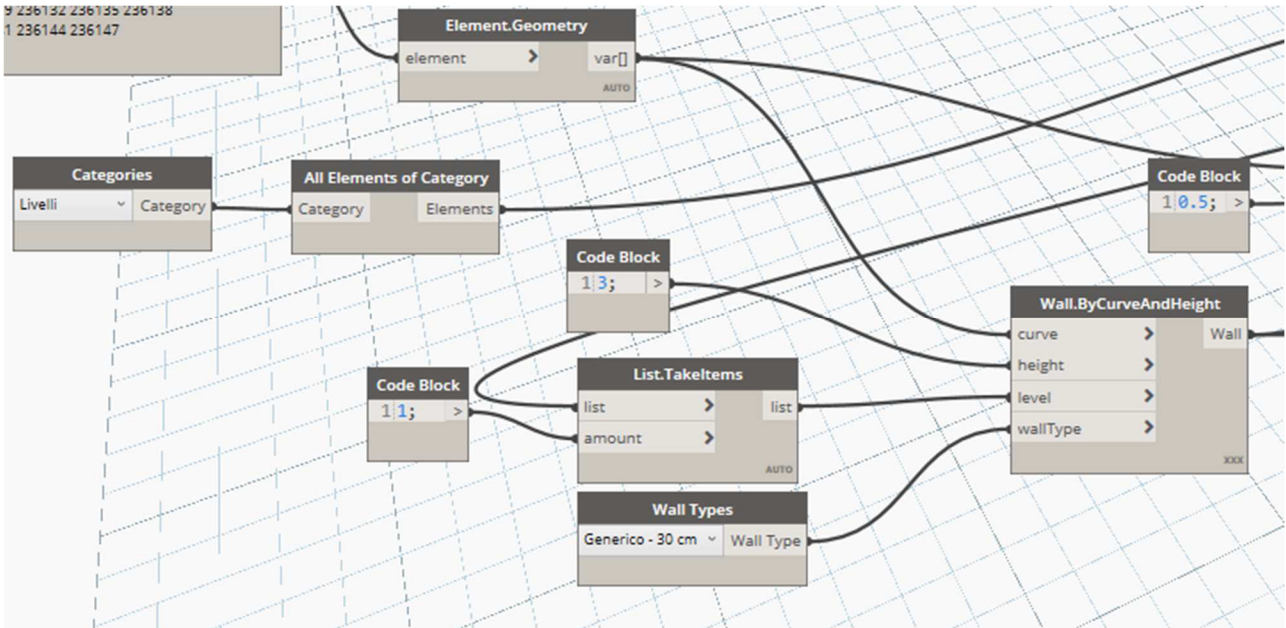
- Clockwork for Dynamo 2.x
- Springd nodes
- Rhythm
- Data – Shapes
- Clockwork for Dynamo 1.x
- Genius Loci

Lo script si basa sostanzialmente nella creazione di muri i quali dovranno ospitare le eventuali porte o finestre presenti nel progetto, a sua volta verranno create delle sezioni di pianta e prospetto.



38) Nodo Dynamo e linee di modello nell'ambiente Revit

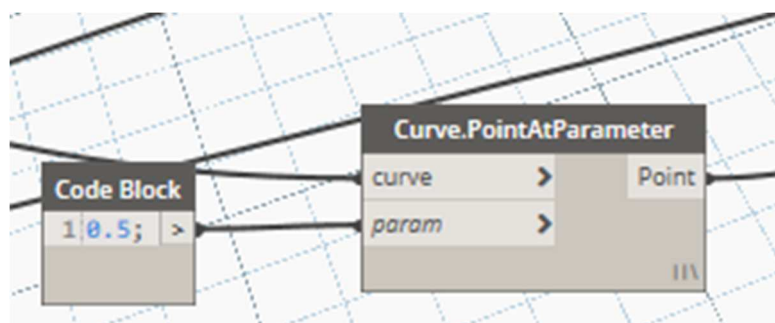
Lo script incomincia nel selezionare dal modello di Revit attraverso il nodo Select Model Elements le linee di modello nelle quali andranno creati i nostri muri di base.



39) Nodi in Dynamo

Attraverso il nodo Element.Geometry che ha come inputt Il nodo Select Model Elements otteniamo la geometria delle linee di modello, che collegate al nodo Wall.ByCurveAndHeight ci permette di creare in tutte queste linee di modell dei muri che hanno un altezza indicata nel nodo Code Block (metri), il Wall Type ovvero il tipo di muro che andrà creato (possibile selzionarlo nel menù a tendina nel nodo Wall Types, il tipo di muro deve essere già stato creato nel modello di Revit), e il livello al quale deve essere associato la base del muro.

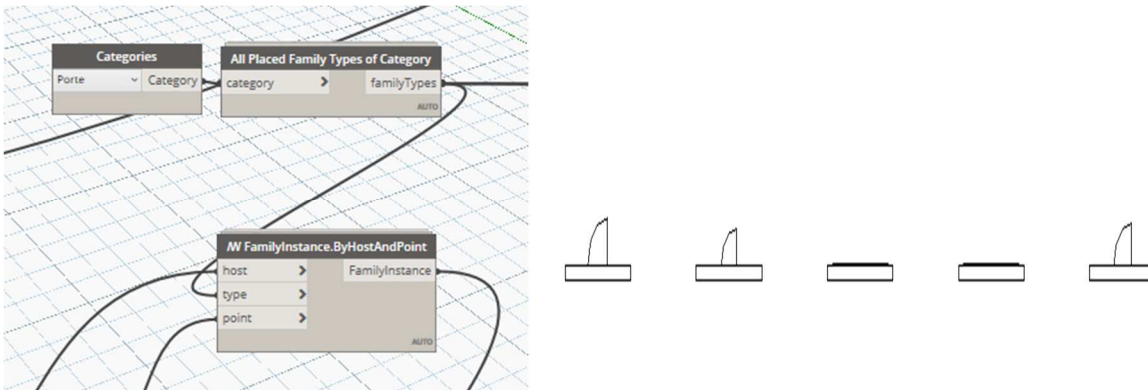
Attraverso il nodo Curve.PointAtParameter che ha come input dell curva le nostre linee di modello di Revit, è possibile tramite il valore 0.5 ricavare il punto medio di esse (0 ci restituisce il punto iniziale e 1 il punto finale)



40) Nodi in Dynamo

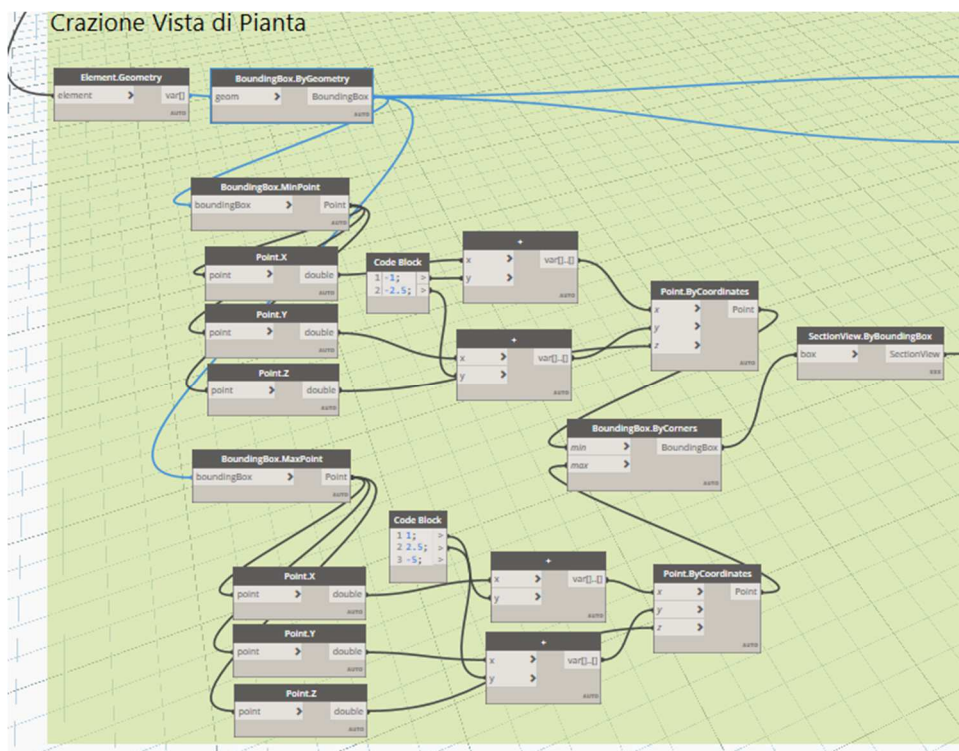
Con il nodo All Placed Family Types of Category , abbiamo la possibilità di restituire tutti i tipi della categoria inserita nell'Input attraverso il nodo Categories (in questo caso la categoria Porte).

E tramite il nodo FamilyInstance.ByHostAndPoint una volta collegati agli input i rispettivi Host (muri di base creati in precedenza) e i punti medi delle linee di modello, otterremo le nostre instance di tutte le porte incluse nel progetto



41) Nodi in Dynamo e creazione di muri e finestre nell'ambiente Revit

Il passo successivo è quello di creare le viste di pianta dei relativi elementi appena creati :



42) Insieme di Nodi in Dynamo

Tramite il nodo `Element.Geometry` otteniamo la geometria delle porte, e con il nodo `BoundingBox.ByGeometry` creiamo un riquadro di definizione sugli elementi.

Come mostrato in figura isoliamo il punto minimo e il punto massimo dei nostri riquadri di definizione, tramite i nodi `BoundingBox.MinPoint` e `BoundingBox.MaxPoint` e tramite i nodi `Point X, Y, Z` isoliamo le coordinate dei nostri punti, le coordinate X, Y dovranno essere aumentate o diminuite delle quantità desiderate per aumentare l'ampiezza della regione di vista della sezione, mentre la coordinata Z non deve essere alterata in quanto nella vista di pianta che a noi interessa la coordinata Z deve rimanere tale e quale.

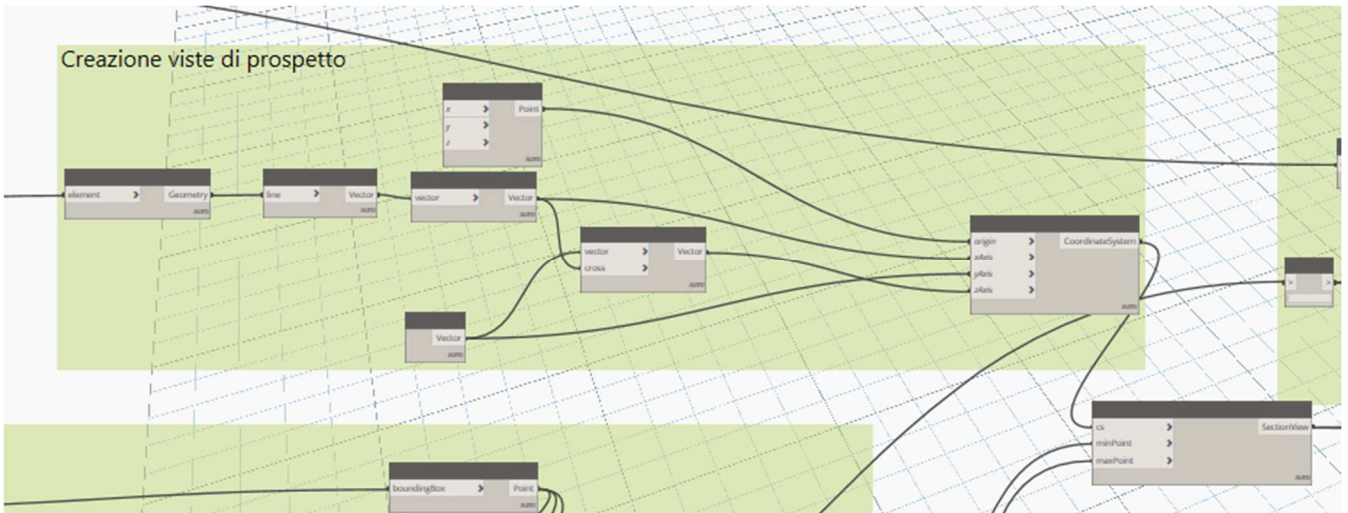
Una volta effettuate le modifiche si ricrea il nostro `BoundingBox` tramite il nodo `BoundingBox.ByCorners`, e tramite il nodo `SectionView.ByBoundingBox` creiamo le nostre viste di pianta.

Per quanto riguarda la creazione delle viste di prospetto il ragionamento è più o meno lo stesso, utilizzeremo sempre il nostro `BoundingBox` dei nostri elementi appuramente modificato per allargare la vista delle nostre sezioni.

La differenza sta nel fatto che non possiamo utilizzare lo stesso nodo per la creazione delle viste in quanto il nodo `SectionView.ByBoundingBox` crea viste solo di pianta degli elementi. Di conseguenza dovremo utilizzare il nodo `SectionView.ByCoordinateSystemMinPointMaxPoint`.

Per utilizzare questo nodo dobbiamo prima creare le nostre coordinate system tramite il nodo `CoordinateSystem.ByOriginVectors`.

Il punto Origin è impostato come il punto con coordinate 0, 0, 0 mentre Xaxis è il vettore direzione della posizione dei nostri muri, trovato grazie al nodo `Element.GetLocation` a cui segue `Line.Direction` e `Vector.Normalized`, per il nostro Yaxis dato che utilizzeremo una vista di prospetto non potrà coincidere con il reale Yaxis ma dovrà essere cambiato con Zaxis come mostrato nella figura sottostante



43) Insieme di nodi in Dynamo

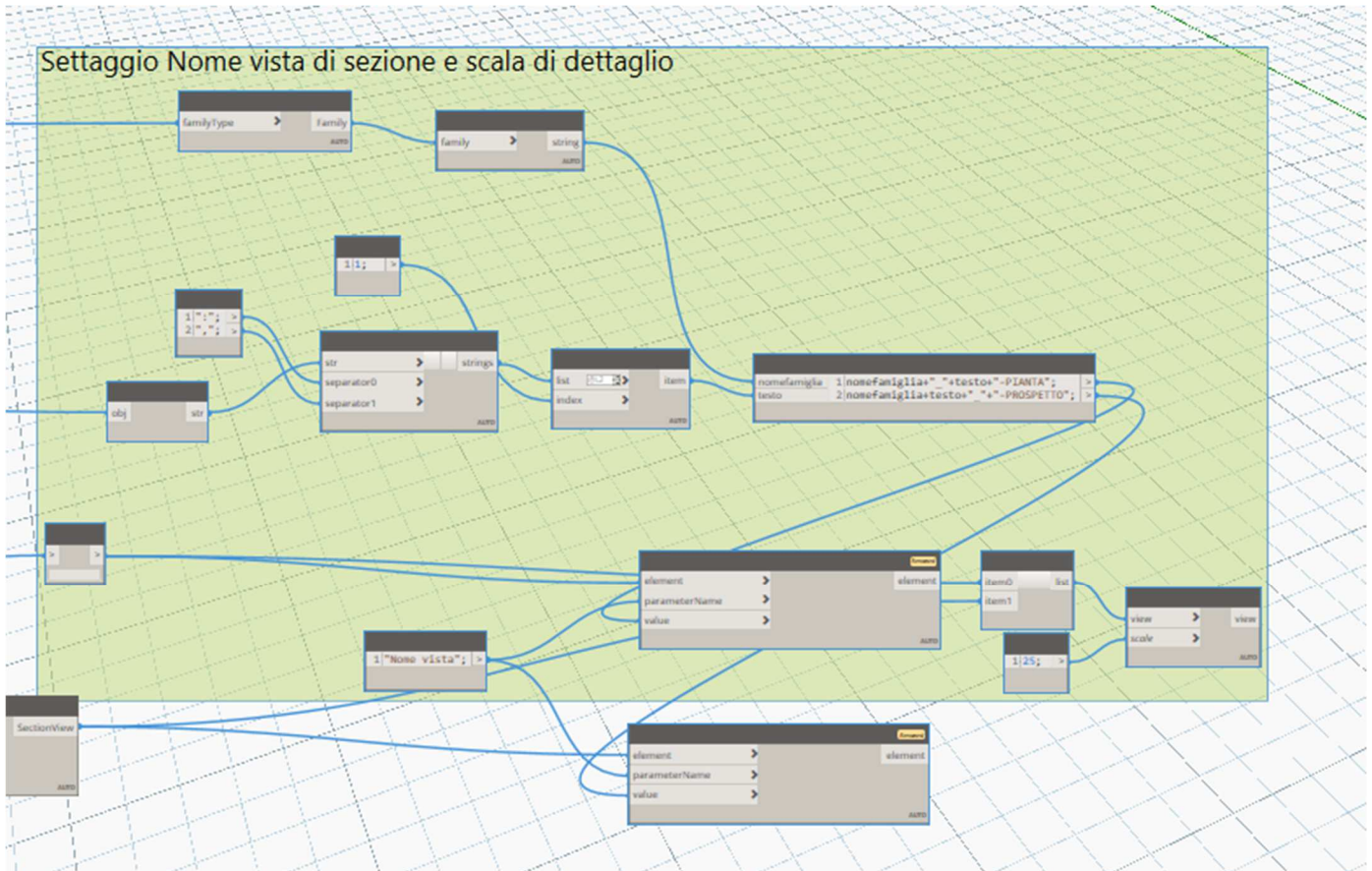
Nel nodo `SectionView.ByCoordinateSystemMinPointMaxPoint` vengono inseriti come `MinPoint` e `MaxPoint` i punti minimi e massimi dei `BoundingBox` creati e appuratamente modificati

Una volta eseguito lo script otteniamo il seguente risultato :



44) Creazione di sezioni nell'ambiente Revit

Una volta create le viste di pianta e di prospetto occorre settare il nome delle viste e la scala di dettaglio



45) Nodi in Dynamo

Con il nodo `Element.SetParameterByName` è possibile tramite l'inserimento delle viste nell'input, il nome del parametro e il valore testuale da inserire, modificare il nome delle viste

E con il nodo `View.SetScale` modificare tramite un semplice `CodeBlock` su cui viene digitata la scala come valore Intero, la scala delle viste.



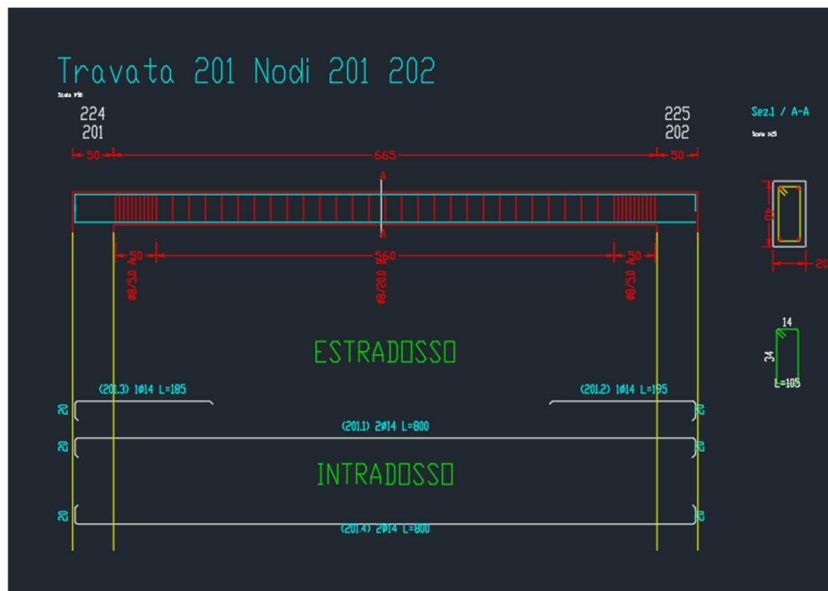
46) Sezioni nell'ambiente Revit

6. Dynamo per la parte Strutturale

In questo script ho voluto automatizzare la creazione di armatura, più in precisione parleremo di staffe nelle travi. La creazione di armature in Revit è un'operazione lunga e dispendiosa se si vuole creare armature con i soli comandi dell'interfaccia del programma, ecco perché ho voluto creare una serie di codici che agevolassero questo procedimento.

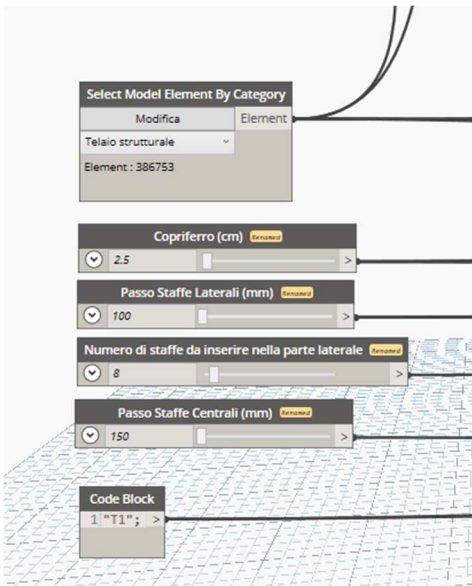
Pacchetti utilizzati :

- Rhythm
- Structural Design



47) Carpenteria di una trave in C.A.

Immaginiamo che un ingegnere che ha effettuato i calcoli di un edificio non abbia un software certificato IFC e quindi non è in grado di trasferirci le informazioni riguardanti l'armatura, dobbiamo quindi ricreare la nostra armatura all'interno del software Revit, seguendo le indicazioni delle tavole e dei disegni consegnateci.

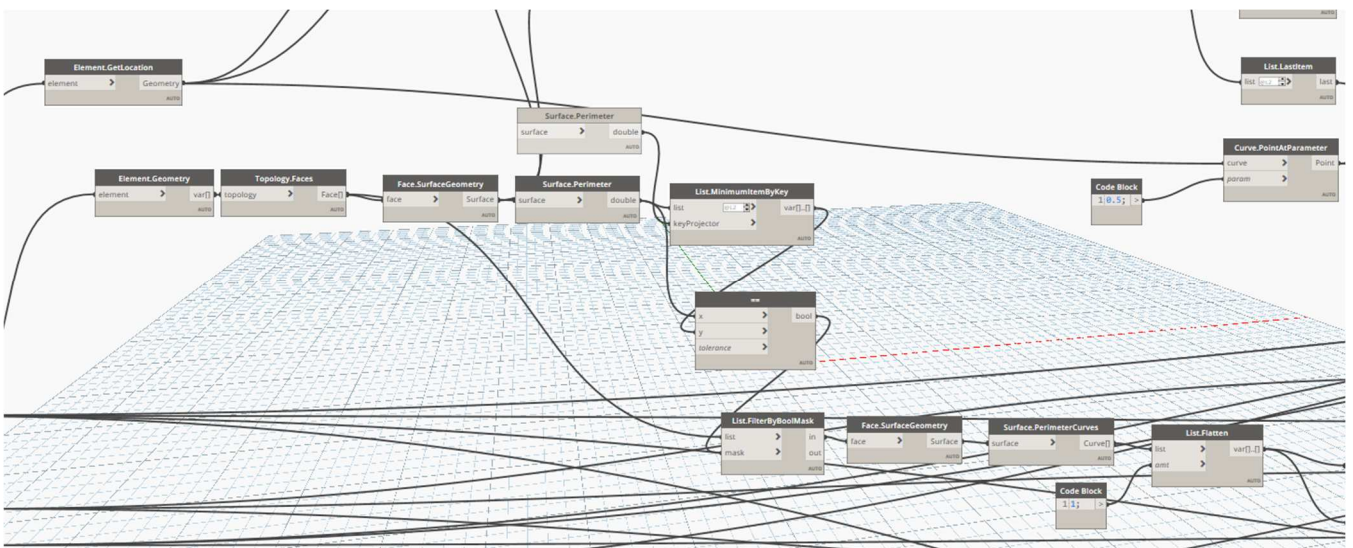


48) Nodi in Dynamo

L'inizio del nostro script ci mostra gli input che ci permetteranno la creazione delle staffe :

- Selezione del nostro elemento (Trave)
- Copriferro
- Passo delle staffe laterali
- Passo delle staffe centrali
- Numero delle staffe da inserire nelle parti laterali

In questi campi verranno inseriti i dati che ci verranno forniti nelle tavole.

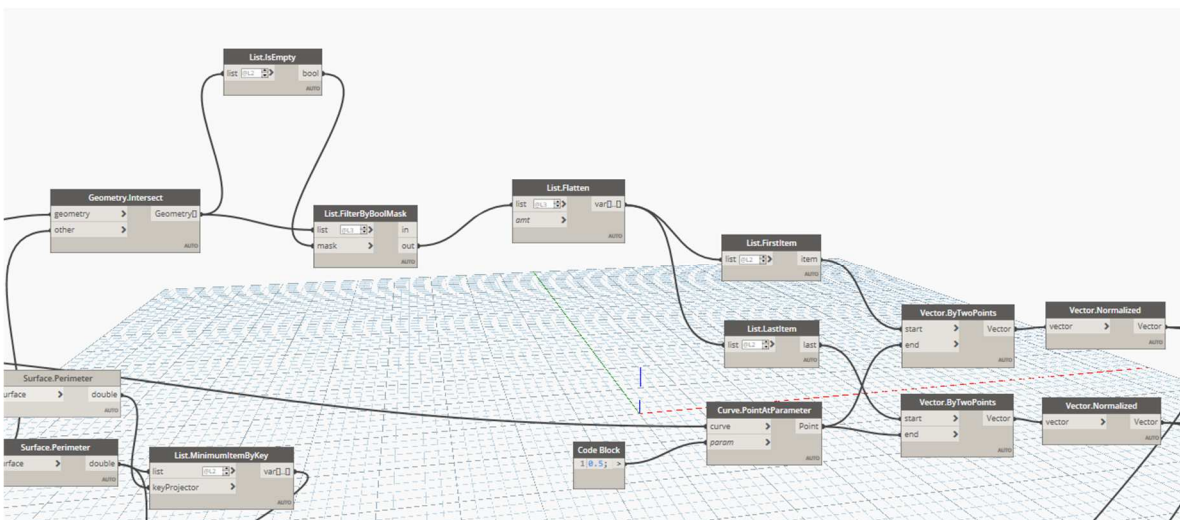


49) Nodi in Dynamo

Tramite il nodo Element.GetLocation collegato al select Model Element ci permtterre di trovare la linea di locazione della trave, e grazie al nodo Point.AtParameter e al valore 0.5 ci ricaviamo il punto a metà della linea di locazione.

Tramite il nodo Element.Geometry otteniamo la geometria dell'elemento, successivamente unito a Topology.Face e Face.surfaceGeoemetry otteniamo tutte le facce della trave.

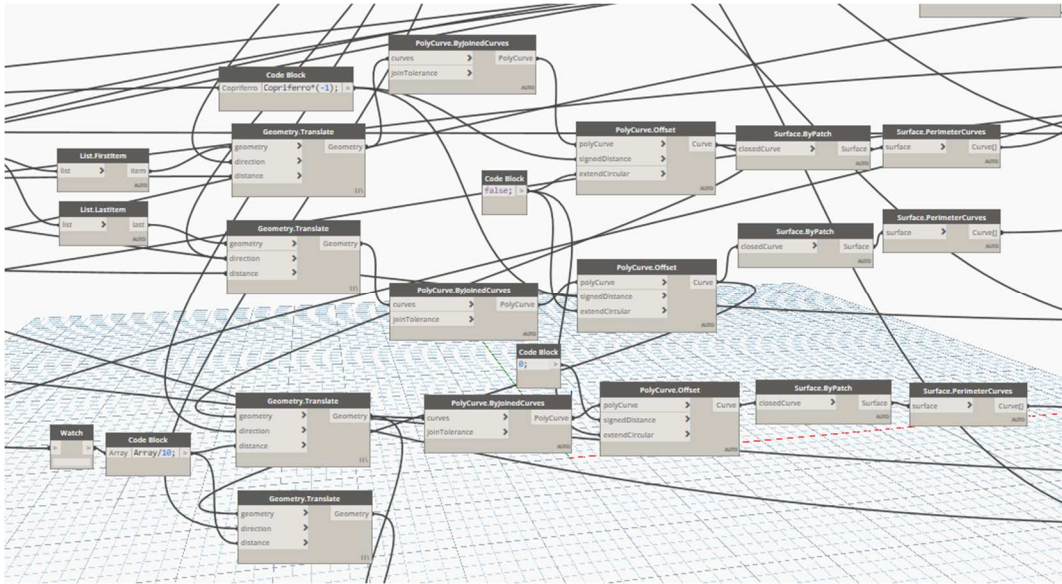
Con il filtro list.MinimumItemByKey uguagliando il valore minimo con la lista dei perimetri delle superfici otteniamo le facce laterali della trave, e tramite Surface.PerimeterCurves otteniamo le linee che delimitano queste superfici.



50) Nodi in Dynamo

Tramite il nodo Geometry.Intersect, a cui vengono collegati la linea di locazione della trave con le due superfici laterali, troviamo i due punti che vanno a intersecarsi.

Con questi due punti e con il punto medio, creiamo due vettori la cui direzione è dall'esterno (superfici laterali) verso l'interno (punto medio), tramite i nodi Vector.ByTwoPoints e Vector.Normalized



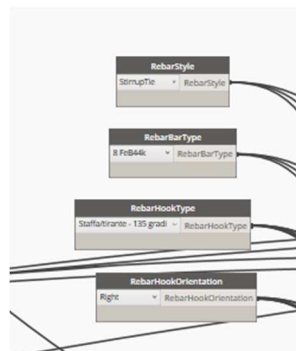
51) Nodi in Dynamo

Ora il nostro obiettivo è quello di creare all'interno della nostra trave, parallelo alle facce laterali il contorno della nostra armatura che poi andrà ripetuto per tutta la lunghezza della trave stessa.

Useremo principalmente il nodo Geometry.Translate che ci permette di effettuare una copia di una curva in una direzione e ad una distanza che desideriamo. Applicheremo prima una copia della curva verso la lunghezza della trave utilizzando i due vettori che ci siamo trovati sopra , e poi con il nodo

E poi con il nodo PolyCurve.Offset creeremo un offset interno della curva in modo da simulare un copriferro della trave, però prima bisognerà convertire la geometria del nodo Geoemtry.Translate in una curva utilizzando il nodo PolyCurve.ByJoinedCurve.

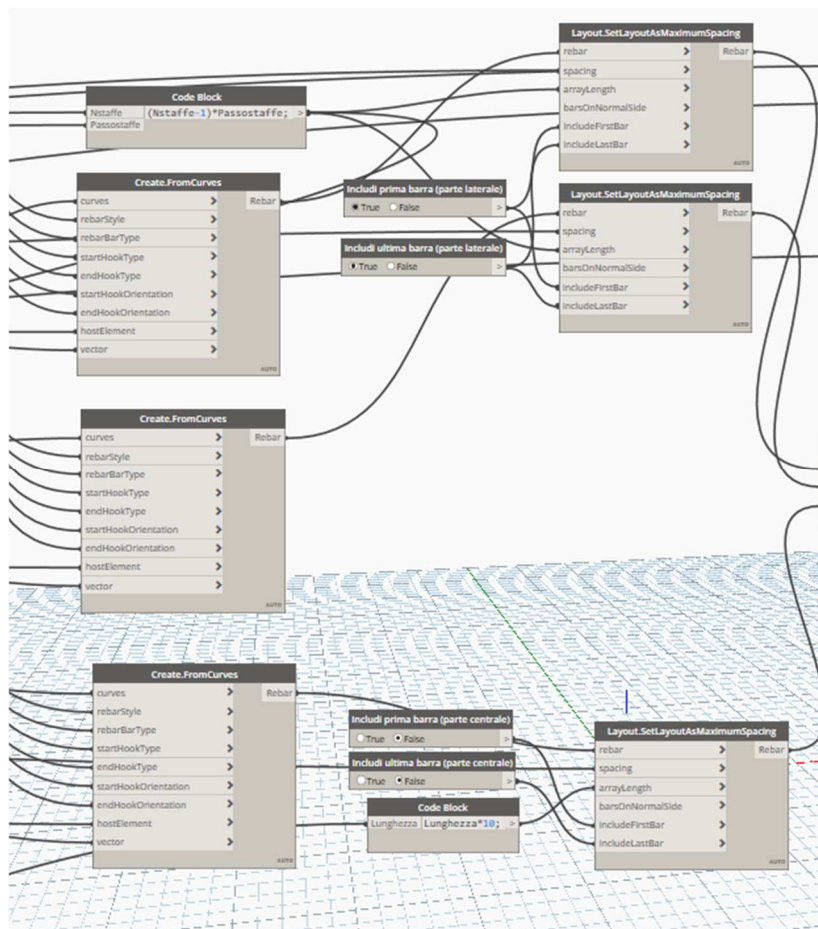
Dopo di che con i nodi Surface.ByPatch e Surface.PerimeterCurves otteniamo il contorno di tale geometria.



52) Nodi in Dynamo

È giunto il momento di impostare i parametri dell'armatura con i nodi :

- RebarStyle che indica lo stile dell' armatura
- RebarBarType che indica il tipo di armatura
- RebarHookType che indica con che angolo dovranno essere piegati i ferri alle estremità
- RebarHkOrientation che indica il verso della ripiegatura a fine estremità



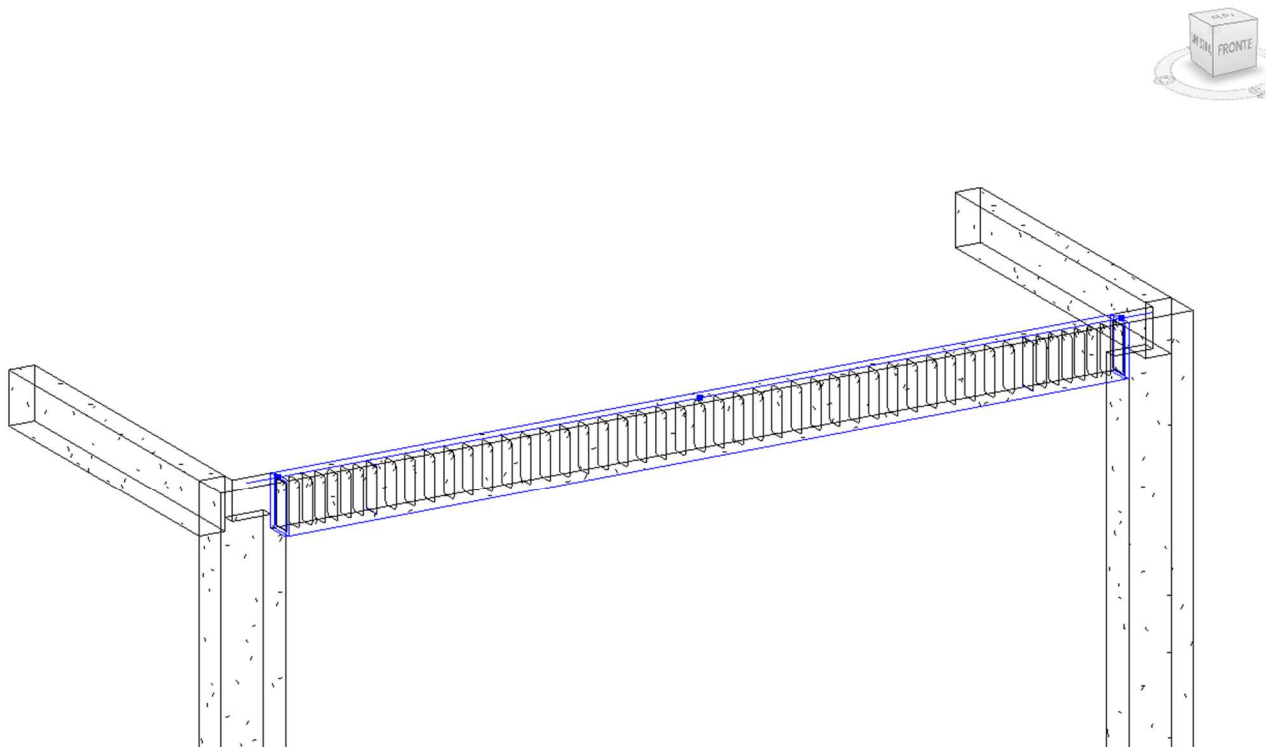
53) Nodi in Dynamo

In conclusione useremo il nodo Create.FromCurves per creare la singola armatura nella geometria che abbiamo creato in precedenza e poi il nodo Layout.SetLayoutAtMaximumSpace che ci permetterà di creare una serie di armature lungo una direzione ad un passo stabilito.

Useremo Dei Blocchi Booleani (True/False) per indicare se vogliamo o no includere la prima e l'ultima barra nel layout

E infine tramite il nodo `Layout.SetLayoutAsMaximumSpacing` possiamo specificare il passo delle staffe lungo uno specifico tratto e tramite dei Blocchi Booleani decidiamo se includere oppure no la prima e l'ultima staffa

Una volta selezionata la trave, impostati i vari input e avviato lo script otterremo un risultato di questo tipo :



54) Creazione di barre di armatura nell'ambiente Revit

7. Conclusioni

Attraverso l'analisi, la gestione, la creazione di dati tramite il generative Design, è possibile ampliare e facilitare, tutte quelle decisioni che riguardano l'ambito lavorativo, dalla progettazione di edifici (generare molteplici soluzioni, con differenti vantaggi e svantaggi caratterizzanti ogni caso), ad estrapolare qualsiasi tipo di informazione durante tutta la vita dell'opera, dalla pianificazione alla sua realizzazione e infine alla sua demolizione, in modo da controllare in maniera dettagliata ogni soluzione, tipo di spesa e imprevisto che sopraggiungono in ogni opera. Inoltre il tutto viene reso più veloce dalla rapidità che offre Dynamo nel cambiare in maniera istantanea tutti gli input inseriti e nel ricevere immediatamente i risultati desiderati in modo da confrontarli con efficacia e rapidità all'interno del team di lavoro, riducendo di gran lunga i tempi di attesa che nel mondo del lavoro sono la quotidianità.

8. Ringrazimenti

Vorrei ringraziare il Prof. Alberto Giretti, mio relatore,
oltre che alla disponibilità e gentilezza dimostrate, per la fiducia che ha riposto in me
durante la stesura della tesi.

Un ringraziamento speciale va ai miei genitori, per lasciarmi libero di intraprendere la strada
che desidero, spero siate orgogliosi di me.

L'ultima frase va a me: non mollare mai !

9. Indice delle figure

1) Processo completo di vita di un fabbricato.....	5
2) Molteplici usi del BIM.....	6
3) Organizzazione della struttura di ogni membro del team.....	8
4) Riunione di coordinamento BIM, per rilevare le eventuali interferenze.....	9
5) Dati e definizione di un oggetto parametrico BIM.....	13
6) Modello Schematico di un edificio mediante l'uso di masse (Revit).....	16
7) impostazione di un'analisi strutturale di un edificio (Revit).....	16
8) impostazione di un'analisi energetica di un edificio (Revit).....	17
9) Stima dei costi mediante abachi creati dal programma Revit.....	17
10) Tavola di una parte in C.A prefabbricato usando il software Revit.....	20
11) Logo del programma Revit	24
12) Nodi del programma Dynamo.....	26
13) Script all'interno del Code Block.....	28
15) Area di lavoro in Dynamo.....	29
16) nodo in Dynamo.....	30
17) Script in Dynamo per la creazione di una lista di dati.....	36
18) Script in Dynamo per la creazione di un cilindro.....	37
19) Nodo Python Script.....	38
20) Ambiente di lavoro del nodo Python Script.....	39
21) Interazione tra Revit, Dynamo e excel.....	40
22) Modello Python Script.....	41
23) Pagina internet di Revit Api Docs.....	43
24) Modello di un edificio, creato con Autodesk Revit.....	45
25) Nodi in Dynamo.....	45
26) Nodi Python script rinominato : Filtro Muri.....	46
27) Script Python.....	46
28) Nodo Python Script rinominato : Elenco viste.....	47
29) Script Python.....	48
30) Nodi in Dynamo.....	49
31) Muri nell'ambiente Revit.....	50
32) Nodo Python Script rinominato : creazione di un muro.....	50
33) Python Script.....	51

34) Nodi in Dynamo.....	53
35) Solaio nell'ambiente Revit.....	54
36) Nodo Python Script rinominato : creazione di un pavimento.....	54
37) Python Script.....	55
38) Nodo Dynamo e linee di modello nell'ambiente Revit.....	56
39) Nodi in Dynamo.....	57
40) Nodi in Dynamo.....	57
41) Nodi in Dynamo e creazione di muri e finestre nell'ambiente Revit.....	58
42) Insieme di Nodi in Dynamo.....	58
43) Insieme di nodi in Dynamo.....	60
44) Creazione di sezioni nell'ambiente Revit.....	60
45) Nodi in Dynamo.....	61
46) Sezioni nell'ambiente Revit.....	61
47) Carpenteria di una trave in C.A.....	62
48) Nodi in Dynamo.....	63
49) Nodi in Dynamo.....	63
50) Nodi in Dynamo.....	64
51) Nodi in Dynamo.....	65
52) Nodi in Dynamo.....	65
53) Nodi in Dynamo.....	66
54) Creazione di barre di armatura nell'ambiente Revit.....	67

10. Bibliografia/ Sitografia

Bibliografia :

- BIM Handbook, A guide to Building Information Modeling for Owners, Managers, Designers, Engineers, And Contractors, Chuck Eastman-Paul Teicholz-Rafael Sacks-Kathleen Liston
- Mas alla de Dynamo: Manual de Python para Revit, Kevin Himmelreich

Sitografia :

- Manuale di Dynamo : https://primer.dynamobim.org/it/01_Introduction/1-3_dynamo_in_action.html
- Dizionario delle API di Revit : <https://www.revitapidocs.com/>
- Manuale di Python : <https://docs.python.org/3/>

Programmi utilizzati :

- Autodesk Revit 2022
- Dynamo