

Caio Borges do Amaral

**Automação do processo de concepção de estrutura
em concreto armado no Revit através do
desenvolvimento de rotina em Dynamo**

João Pessoa

2023

Caio Borges do Amaral

**Automação do processo de concepção de estrutura em
concreto armado no Revit através do desenvolvimento de
rotina em Dynamo**

Trabalho de Conclusão de Curso apresentado à
Coordenação de Engenharia Civil como requi-
sito para obtenção do grau de Bacharel em En-
genharia Civil.

Universidade Federal da Paraíba

Centro de Tecnologia

Departamento de Engenharia Civil e Ambiental

Orientador: Francisco Jácome Sarmento

João Pessoa

2023

Catálogo na publicação
Seção de Catalogação e Classificação

A485a Amaral, Caio Borges do.

Automação do processo de concepção de estrutura em concreto armado no Revit através do desenvolvimento de rotina em Dynamo / Caio Borges do Amaral. - João Pessoa, 2023.

160 f. : il.

Orientação: Francisco Jácome Sarmento.
TCC (Graduação) - UFPB/CT.

1. Automação. 2. Concepção estrutural. 3. BIM. 4. Revit. 5. Dynamo. I. Sarmento, Francisco Jácome. II. Título.

UFPB/CT/BSCT

CDU 624(043.2)

FOLHA DE APROVAÇÃO

CAIO BORGES DO AMARAL

AUTOMAÇÃO DO PROCESSO DE CONCEPÇÃO DE ESTRUTURA EM CONCRETO ARMADO NO REVIT ATRAVÉS DO DESENVOLVIMENTO DE ROTINA EM DYNAMO

Trabalho de Conclusão de Curso em 09/11/2023 perante a seguinte Comissão Julgadora:



Francisco Jácome Sarmiento
Departamento de Engenharia Civil e Ambiental do CT/UFPB



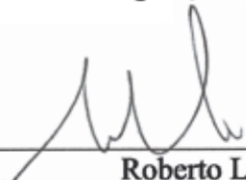
APROVADO



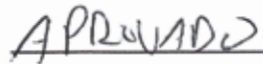
Claudino Lins Nóbrega Júnior
Departamento de Engenharia Civil e Ambiental do CT/UFPB



Aprovado



Roberto Leal Pimentel
Departamento de Engenharia Civil e Ambiental do CT/UFPB



APROVADO

AGRADECIMENTOS

A Deus, por ter me proporcionado todas as condições para a conclusão desse trabalho. Pelo dom da vida, pelos anos de paz e saúde que vivi e por todas as dificuldades que me permitiu enfrentar e contribuíram na formação do homem que sou.

À minha mãe Patrícia, minha primeira educadora, por toda dedicação, paciência, sacrifício e, acima de tudo, amor depositados na minha criação e determinantes na minha formação como pessoa. Sempre levarei comigo seu exemplo de mulher batalhadora.

Ao meu pai Gustavo, meu segundo educador, por sempre ter me orientado quanto ao certo e me ensinado que “tudo na vida é treino”. Sou grato por ter me ensinado a nunca me contentar com pouco e por acreditar em mim mais do que eu mesmo.

Ao meu irmão Sávio, meu colega nessa vida e melhor amigo, quem sempre tive ao lado e me proporcionou a oportunidade de ser irmão mais velho e todos os aprendizados inerentes a essa responsabilidade.

À minha namorada Amanda, com a qual compartilhei tantos momentos felizes nos últimos anos. Sou grato por todo o amor recebido e por ter sido sempre disponível ao escutar minhas angústias e expectativas sobre os desafios dessa vida.

Às amigas que fiz e cultivei nos anos de graduação por todo o conhecimento e experiências compartilhadas, em especial, aos meus amigos Diego, José Júlio, José Matheus, Lucas, Ricardo e Yussef.

Ao professor Francisco Jácome Sarmiento, pela orientação nesse trabalho e aos professores Claudino Lins e Roberto Leal, por aceitarem a participação na banca examinadora e por todo o conhecimento compartilhado.

Aos profissionais da Mediterranné, em especial ao mestre Welliton, da TECNCON, em especial ao engenheiro Guilherme Cavalcanti e da Porto Inc, em especial ao engenheiro Gabriel Guimarães, pelas oportunidades de estágio e aprendizados decorrentes.

*“Não é por ver fazer
E sim por fazer
Que o aprendiz se torna mestre
E o mestre se torna artista”
Estela Amaral*

RESUMO

A concepção de uma estrutura, como primeira etapa do projeto da mesma, é permeada de tomadas de decisões por parte do engenheiro projetista, haja vista a necessidade de buscar solução que atenda critérios impostos por questões orçamentárias, arquitetônicas e construtivas, para citar as principais. Nesse âmbito, os processos de projeto atuais são auxiliados pelos desenvolvimentos referentes à Modelagem da Informação da Construção (BIM), tais quais o surgimento e aprimoramento de softwares, a exemplo do Revit e do Dynamo. Nesse aspecto, a presente pesquisa se propõe a tratar do desenvolvimento de rotina em Dynamo capaz de automatizar o processo de concepção de estrutura em concreto armado no Revit. Para isso, é feita coleta dos principais conceitos relacionados ao processo de concepção estrutural, Modelagem da Informação da Construção e dos softwares usados no desenvolvimento da pesquisa. Em seguida, tem-se a definição do fluxograma que orienta o desenvolvimento da rotina, estabelecendo as entradas, saídas e etapas do processamento de dados. Cada etapa do processamento de dados do algoritmo é descrita, mediante a apresentação do código no ambiente do Dynamo que a define e de ilustrações dos resultados intermediários que surgem da execução da rotina. Uma vez que o desenvolvimento da rotina é finalizado, sua execução é feita tomando como entrada projeto arquitetônico cuja concepção estrutural foi previamente realizada por engenheiro, a fim de compará-la com o modelo estrutural produzido como saída da rotina.

Palavras-chave: Automação; Concepção estrutural; BIM; Revit; Dynamo.

ABSTRACT

The conception of a structure, as the first stage of its design, is permeated with decision-making on the part of the design engineer, considering the need to find a solution that meets criteria imposed by budgetary, architectural, and construction issues, to name the main ones. In this context, current design processes are aided by developments related to Building Information Modeling (BIM), such as the emergence and improvement of software, such as Revit and Dynamo. In this regard, the present research aims to address the development of a Dynamo routine capable of automating the process of designing a reinforced concrete structure in Revit. To do this, the main concepts related to the structural design process, Building Information Modeling, and the software used in the research development are collected. Next, the definition of the flowchart that guides the routine development is established, specifying the inputs, outputs, and data processing steps. Each data processing step of the algorithm is described, by presenting the code in the Dynamo environment that defines it and illustrations of the intermediate results that arise from the routine execution. Once the routine development is completed, its execution is carried out, taking as input an architectural project whose structural conception was previously performed by an engineer, in order to compare it with the structural model produced as the output of the routine.

Keywords: Automation; Structural conception; BIM; Revit; Dynamo.

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Versões dos <i>softwares</i> utilizados	18
Figura 2.2 – Fluxograma do algoritmo	19
Figura 3.1 – Caminho das forças em uma estrutura de edifício	21
Figura 3.2 – Equilíbrio de seção	24
Figura 3.3 – Seções recomendadas para elemento solicitado a tração	26
Figura 3.4 – Seção genérica	27
Figura 3.5 – Seções recomendadas para elemento solicitado a compressão	28
Figura 3.6 – Distribuição de tensões em elemento solicitado a flexão	29
Figura 3.7 – Seções recomendadas para elemento solicitado a flexão	29
Figura 3.8 – Diagramas tensão-deformação	31
Figura 3.9 – Laje maciça e nervurada	37
Figura 3.10–Classificação das vigas quanto ao número de apoios	39
Figura 3.11–Classificação da viga quanto a posição em relação a laje	39
Figura 3.12–Obtenção de área de influência de pilar	44
Figura 3.13–Superestrutura e infra-estrutura	45
Figura 3.14–Vista em planta de sapata	48
Figura 3.15–Vista em corte de sapata	49
Figura 4.1 – Estrutura conceitual de uma família de objeto parede	54
Figura 4.2 – Uso do BIM no ciclo de vida da construção	55
Figura 4.3 – Usos do BIM para projeto	56
Figura 4.4 – Usos do BIM para construção	57
Figura 4.5 – Usos do BIM para operação e manutenção	57
Figura 4.6 – Elementos para o Revit	59
Figura 4.7 – Hierarquia do Revit	60
Figura 4.8 – Programação	61
Figura 4.9 – Anatomia de um nó	62
Figura 4.10–Estados do nó	63
Figura 4.11–Categorização da hierarquia geométrica no Dynamo	63
Figura 4.12–Itens e lista	64
Figura 4.13–Listas de tamanhos diferentes	64
Figura 4.14–Opções de amarra de um nó	65
Figura 5.1 – Vista isométrica do Edifício Exemplo	66
Figura 5.2 – Vista em planta do Edifício Exemplo	67
Figura 5.3 – Visão geral da rotina	67
Figura 5.4 – Fluxograma da rotina	68
Figura 5.5 – Grupo para criação das linhas de localização das paredes	69

Figura 5.6 – Grupo para criação das linhas de localização das paredes (trecho 1)	69
Figura 5.7 – Grupo para criação das linhas de localização das paredes (trecho 2)	70
Figura 5.8 – Grupo para criação das linhas de localização das paredes (trecho 3)	70
Figura 5.9 – Vista isométrica das linhas de localização das paredes	71
Figura 5.10–Vista em planta das linhas de localização das paredes	72
Figura 5.11–Grupo para criação das geometrias de referência	73
Figura 5.12–Grupo para criação das geometrias de referência (trecho 1)	73
Figura 5.13–Vista isométrica dos pontos de interseção entre linhas de localização de paredes	74
Figura 5.14–Vista em planta dos pontos de interseção entre linhas de localização de paredes	75
Figura 5.15–Vista isométrica da lista final de pontos	75
Figura 5.16–Grupo para criação das geometrias de referência (trecho 2)	76
Figura 5.17–Grupo para criação das geometrias de referência (trecho 3)	76
Figura 5.18–Grupo para criação das geometrias de referência (trecho 4)	77
Figura 5.19–Vista isométrica das linhas de localização dos pilares	78
Figura 5.20–Vista isométrica das linhas de localização das vigas	78
Figura 5.21–Vista isométrica das linhas de localização de vigas e pilares	79
Figura 5.22–Grupo para pré-dimensionamento dos pilares	80
Figura 5.23–Ilustração do algoritmo de pré-dimensionamento de pilar (1)	80
Figura 5.24–Grupo para pré-dimensionamento dos pilares (trecho 1)	81
Figura 5.25–Grupo para pré-dimensionamento dos pilares (trecho 2)	81
Figura 5.26–Ilustração do algoritmo de pré-dimensionamento de pilar (2)	82
Figura 5.27–Grupo para pré-dimensionamento dos pilares (trecho 3)	83
Figura 5.28–Grupo para pré-dimensionamento dos pilares (trecho 4)	83
Figura 5.29–Ilustração do algoritmo de pré-dimensionamento de pilar (3)	84
Figura 5.30–Grupo para pré-dimensionamento dos pilares (trecho 5)	85
Figura 5.31–Grupo para pré-dimensionamento dos pilares (trecho 6)	85
Figura 5.32–Grupo para pré-dimensionamento dos pilares (trecho 7)	86
Figura 5.33–Ilustração do algoritmo de pré-dimensionamento de pilar (4)	87
Figura 5.34–Grupo para pré-dimensionamento dos pilares (trecho 8)	87
Figura 5.35–Grupo para pré-dimensionamento dos pilares (trecho 9)	88
Figura 5.36–Grupo para pré-dimensionamento dos pilares (trecho 10)	89
Figura 5.37–Grupo para pré-dimensionamento dos pilares (trecho 11)	90
Figura 5.38–Grupo para pré-dimensionamento das vigas	91
Figura 5.39–Grupo para pré-dimensionamento das vigas (trecho 1)	91
Figura 5.40–Grupo para pré-dimensionamento das vigas (trecho 2)	92
Figura 5.41–Grupo para pré-dimensionamento das vigas (trecho 3)	92
Figura 5.42–Grupo para pré-dimensionamento das vigas (trecho 4)	93
Figura 5.43–Grupo para pré-dimensionamento das vigas (trecho 5)	94
Figura 5.44–Grupo para pré-dimensionamento das sapatas	95

Figura 5.45–Grupo para pré-dimensionamento das sapatas (trecho 1)	95
Figura 5.46–Grupo para pré-dimensionamento das sapatas (trecho 2)	96
Figura 5.47–Grupo para pré-dimensionamento das sapatas (trecho 3)	96
Figura 5.48–Grupo para pré-dimensionamento das sapatas (trecho 4)	97
Figura 5.49–Grupo para criação dos elementos de pilar	98
Figura 5.50–Família Pilar Retangular Concreto	99
Figura 5.51–Grupo para criação dos elementos de pilar (trecho 1)	99
Figura 5.52–Grupo para criação dos elementos de pilar (trecho 2)	100
Figura 5.53–Vista isométrica das instâncias de pilar	101
Figura 5.54–Grupo para criação dos elementos de viga	102
Figura 5.55–Família Viga Retangular Concreto	102
Figura 5.56–Grupo para criação dos elementos de viga (trecho 1)	103
Figura 5.57–Grupo para criação dos elementos de viga (trecho 2)	103
Figura 5.58–Vista isométrica das instâncias de viga	104
Figura 5.59–Grupo para criação dos elementos de sapata	105
Figura 5.60–Família Sapata Cônica Concreto	105
Figura 5.61–Grupo para criação dos elementos de sapata (trecho 1)	106
Figura 5.62–Grupo para criação dos elementos de sapata (trecho 2)	106
Figura 5.63–Vista isométrica das instâncias de sapata	107
Figura 5.64–Dynamo Player	108
Figura 5.65–Vista isométrica das instâncias de pilar, viga e sapata	109
Figura 6.1 – Valores de entrada da rotina	111
Figura A.1 – Nó NivelAcima	118
Figura A.2 – Visão geral do nó NivelAcima	118
Figura A.3 – Nó NivelAcima (parte 1)	118
Figura A.4 – Nó NivelAcima (parte 2)	119
Figura A.5 – Nó NivelAcima (parte 3)	119
Figura B.1 – Nó PontosIntersecaoLinhas	121
Figura B.2 – Visão geral do nó PontosIntersecaoLinhas	121
Figura B.3 – Nó PontosIntersecaoLinhas (parte 1)	121
Figura B.4 – Lista de linhas passada como entrada	122
Figura B.5 – Nó PontosIntersecaoLinhas (parte 2)	122
Figura B.6 – Seleção de pontos iniciais e finais	123
Figura B.7 – Nó PontosIntersecaoLinhas (parte 3)	123
Figura B.8 – Nó PontosIntersecaoLinhas (parte 4)	124
Figura B.9 – Pontos obtidos por projeção	125
Figura B.10–Nó PontosIntersecaoLinhas (parte 5)	125
Figura B.11–Linhas traçadas	126
Figura B.12–Interseção das linhas traçadas	126

Figura B.13–Nó PontosIntersecaoLinhas (parte 6)	127
Figura B.14–Verificação da interseção entre pontos e linhas	127
Figura B.15–Nó PontosIntersecaoLinhas (parte 7)	128
Figura B.16–Nó PontosIntersecaoLinhas (parte 8)	128
Figura B.17–Pontos resultantes da interseção das linhas	129
Figura C.1 – Nó LinhasVigasPorPontos	130
Figura C.2 – Visão geral do nó LinhasVigasPorPontos	130
Figura C.3 – Nó LinhasVigasPorPontos (parte 1)	130
Figura C.4 – Lista de pontos passada como entrada	131
Figura C.5 – Nó LinhasVigasPorPontos (parte 2)	131
Figura C.6 – Pontos agrupados em listas	132
Figura C.7 – Nó LinhasVigasPorPontos (parte 3)	132
Figura C.8 – Ordenação dos pontos das listas	133
Figura C.9 – Nó LinhasVigasPorPontos (parte 4)	133
Figura C.10–Criação de linhas nas listas	134
Figura C.11–Nó LinhasVigasPorPontos (parte 5)	134
Figura C.12–Lista de pontos agrupados pelo valor da coordenada y	135
Figura C.13–Lista de linhas passada como saída	135

LISTA DE TABELAS

Tabela 3.1 – Propriedades mecânicas dos materiais	31
Tabela 3.2 – Valores de resistência a tração em função da resistência característica à compressão do concreto	34
Tabela 3.3 – Valores de β_{fl}	41
Tabela 3.4 – Valores do coeficiente adicional γ_n para pilares	42
Tabela 3.5 – Tensão ideal de cálculo do concreto (σ_{id}) para $\rho = 2\%$	42
Tabela 3.6 – Tensão ideal de cálculo do concreto (σ_{id}) para $\rho = 4\%$	43
Tabela 3.7 – Tensão ideal de cálculo do concreto (σ_{id}) para $\rho = 6\%$	43
Tabela 3.8 – Tensão ideal de cálculo do concreto (σ_{id}) para $\rho = 8\%$	43
Tabela 3.9 – Coeficiente γ_{corr}	43
Tabela 3.10 – Tensões admissíveis	47
Tabela 6.1 – Resumo dos quantitativos dos modelos estruturais	112

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Justificativa	15
1.2	Objetivos	16
1.2.1	Objetivo geral	16
1.2.2	Objetivos específicos	16
1.3	Delimitação da pesquisa	16
1.4	Prévia dos capítulos	16
2	METODOLOGIA	18
3	CONCEPÇÃO DE ESTRUTURA EM CONCRETO ARMADO	20
3.1	Conceito de estrutura	20
3.2	Elementos estruturais	22
3.3	Esforços solicitantes	23
3.4	Critérios iniciais para definição da seção do elemento	26
3.5	Materiais estruturais	29
3.6	Concreto armado	32
3.7	Pré-dimensionamento dos elementos estruturais	36
3.7.1	Lajes	36
3.7.2	Vigas	38
3.7.3	Pilares	41
3.7.4	Sapatas	44
3.8	Lançamento dos elementos estruturais	49
4	MODELAGEM DA INFORMAÇÃO DA CONSTRUÇÃO (BIM)	52
4.1	Conceito de BIM	52
4.2	Modelagem paramétrica	53
4.3	Usos do BIM	55
4.4	Revit	58
4.5	Dynamo	60
5	DESENVOLVIMENTO DA ROTINA	66
5.1	Visão geral da rotina	67
5.2	Leitura dos elementos do modelo arquitetônico	68
5.2.1	Criação das linhas de localização das paredes	68
5.3	Criação das geometrias de referência	72
5.3.1	Criação das geometrias de referência	72

5.4	Pré-dimensionamento dos elementos estruturais	79
5.4.1	Pré-dimensionamento de pilar	79
5.4.2	Pré-dimensionamento de viga	90
5.4.3	Pré-dimensionamento de sapata	94
5.5	Criação dos elementos estruturais	98
5.5.1	Criação dos elementos de pilar	98
5.5.2	Criação dos elementos de viga	101
5.5.3	Criação dos elementos de sapata	105
5.6	Execução da rotina	107
6	APLICAÇÃO DA ROTINA	110
6.1	Modelo arquitetônico	110
6.2	Modelo estrutural concebido manualmente	110
6.3	Valores passados como entrada da rotina	110
6.4	Modelo estrutural concebido pela rotina	111
6.5	Avaliação dos resultados da rotina	111
7	CONSIDERAÇÕES FINAIS	113
7.1	Recomendações de pesquisas futuras	113
	REFERÊNCIAS	115
	APÊNDICE A – DESENVOLVIMENTO DO NÓ NIVELACIMA . . .	118
	APÊNDICE B – DESENVOLVIMENTO DO NÓ PONTOSINTERSE- CAOLINHAS	121
	APÊNDICE C – DESENVOLVIMENTO DO NÓ LINHASVIGASPOR- PONTOS	130
	APÊNDICE D – ESTRUTURA CONCEBIDA PELA ROTINA	136
	ANEXO A – ARQUITETURA	143
	ANEXO B – ESTRUTURA CONCEBIDA MANUALMENTE	154

1 INTRODUÇÃO

1.1 Justificativa

Dados levantados pelo Sienge (2022) apontam que o custo referente à execução de uma estrutura corresponde a até 20% do total de uma obra. Helene (2019) vai além, ao considerar que a estrutura representa 30% do custo total de uma obra e 100% de sua segurança.

Não só em decorrência da participação significativa no orçamento de uma construção, mas a necessidade de atender condições impostas pela arquitetura, de segurança, de utilização, construtivas, estéticas e ambientais, como é apontado por Martha (2010), tornam necessária a avaliação das diversas alternativas de concepção estrutural.

Como primeira etapa do projeto da estrutura, a concepção estrutural é auxiliada por novos desenvolvimentos na indústria relacionada à arquitetura, engenharia e construção (AEC). Dentre esses, pode-se citar a Modelagem da Informação da Construção (BIM), que permite a construção digital de um modelo virtual preciso da edificação que oferece suporte a todas as fases de projeto, proporcionando análise e controle melhores do que aqueles através de processos manuais, como apontado por Sacks et al. (2021).

De acordo com Sena (2019), através do BIM o projeto assume o comportamento de um banco de dados, no qual informações podem ser acessadas e manipuladas por meio de linguagem de programação, possibilitando a criação de algoritmos capazes de otimizar tarefas, alterar parâmetros ou até inserir elementos.

Nesse contexto, empresas como a Autodesk trazem soluções como o Revit e o Dynamo, que se tratam de *softwares* que usam dos conceitos advindos do BIM ao permitir a modelagem de construções e criação de programas que automatizam processos de projeto, respectivamente.

Quanto ao ato de programar, Sarmento (2020) o define como “atividade de organizar, na forma de instruções escritas em linguagem computacional, os passos a serem executados pelo computador para alcançar a solução de um problema”.

Nessa perspectiva, diante do potencial do uso do BIM aliado às práticas de programação em processos de projeto somada à necessidade da avaliação de diversas soluções de concepção de uma estrutura, a presente pesquisa se propõe a tratar da automação do processo de concepção de estrutura em concreto armado no Revit através do desenvolvimento de rotina em Dynamo.

1.2 Objetivos

1.2.1 Objetivo geral

O objetivo geral da pesquisa elaborada é desenvolver rotina no Dynamo que automatize o processo de concepção de estrutura em concreto armado no Revit.

1.2.2 Objetivos específicos

A fim de alcançar o objetivo geral, são definidos os seguintes objetivos específicos:

- Descrever etapas do processo de concepção de estrutura em concreto armado;
- Explicar os conceitos que envolvem o BIM e como esses se fazem presente no *software* Revit;
- Identificar de que maneira o *software* Dynamo permite a automação de processos de projeto no Revit;
- Programar rotina no Dynamo, garantindo que o algoritmo criado seja capaz de gerar modelo de estrutura no Revit em concordância com as etapas do processo de concepção de estrutura em concreto armado;
- Avaliar rotina desenvolvida, tomando projeto arquitetônico existente como entrada e comparando o modelo estrutural gerado com modelo concebido manualmente.

1.3 Delimitação da pesquisa

A concepção de estrutura em concreto armado proposta se restringe àquela feita para edificações. Outras soluções de engenharia que envolvem conceber outros tipos de estruturas de concreto armado, a exemplo de estádios e pontes, não são incluídas nesse trabalho.

Ainda, a concepção se restringe à modelagem de vigas, pilares e sapatas. Lajes, outras soluções de fundação e estruturas de contenção não são abarcadas na fase atual de desenvolvimento do trabalho.

1.4 Prévia dos capítulos

No Capítulo 2, METODOLOGIA, constam os procedimentos metodológicos usados no desenvolvimento da pesquisa. São mencionadas as fontes de pesquisa bibliográfica, *softwares* utilizados e apresentado o fluxograma que orienta o desenvolvimento da rotina.

No Capítulo 3, CONCEPÇÃO DE ESTRUTURA EM CONCRETO ARMADO, são abordados os fundamentos necessários ao entendimento do processo de concepção de uma estrutura em concreto armado. Discute-se o conceito de estrutura, que introduz o estudo dos elemen-

tos estruturais, dos esforços solicitantes e de que maneira esses influem na determinação da geometria do elemento estrutural. Em seguida, é feita uma discussão sobre os materiais estruturais, com ênfase no concreto armado, a luz do estudo das propriedades que influenciam no comportamento da estrutura. Por fim, são apresentadas equações que norteiam o pré-dimensionamento dos elementos estruturais e recomendações a serem seguidas para locação desses.

O Capítulo 4, **MODELAGEM DA INFORMAÇÃO DA CONSTRUÇÃO (BIM)**, trata do conceito de BIM, trazendo sua definição e de que maneira a parametrização se faz presente no uso dessa tecnologia. São apresentadas as possibilidades de uso do BIM na AEC. Em seguida, o *software* Revit é introduzido, relacionando suas funções com os conceitos de BIM apresentados. Ao final, o *software* Dynamo é apresentado, mostrando como é feito seu uso e de que maneira ele se integra com o Revit permitindo a automação de tarefas.

No Capítulo 5, **DESENVOLVIMENTO DA ROTINA**, a rotina desenvolvida é apresentada. A explicação do algoritmo desenvolvido é feita através da apresentação do código utilizado e por meio de ilustrações dos resultados obtidos após cada etapa de execução do código. Para ilustração dos resultados intermediários e finais gerados, é tomada uma arquitetura exemplo passada como entrada na rotina.

No Capítulo 6, **APLICAÇÃO DA ROTINA**, a rotina desenvolvida é executada tomando como entrada uma arquitetura existente, de solução estrutural já concebida manualmente. A solução em questão é tomada para comparação com aquela gerada como saída da execução da rotina.

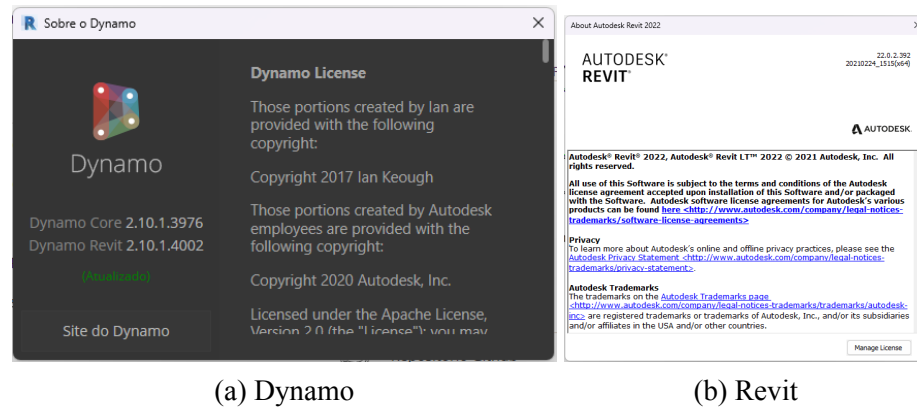
2 METODOLOGIA

Para a pesquisa bibliográfica acerca do processo de concepção de estrutura em concreto armado, buscou-se a leitura de livros e artigos nas áreas de análise estrutural, resistência dos materiais e concreto armado, selecionando aqueles já consagrados no meio acadêmico. Se fez valer também da leitura das normas técnicas brasileiras que tratassem do projeto de estruturas em concreto armado.

A pesquisa bibliográfica acerca de BIM foi feita a partir da leitura de livros de pesquisadores da área, guias de implementação da tecnologia elaborados por profissionais da AEC e artigos de orientação produzidos por empresas que trabalham com a comercialização de *softwares* que se baseiam na tecnologia, além também de artigos e dissertações que abordam sobre o tema.

O desenvolvimento da rotina se deu através do Dynamo para Revit, em sua versão 2.10 (Figura 2.1a). O acesso ao Dynamo é feito através da interface do Revit, esse utilizado na versão 22 (Figura 2.1b).

Figura 2.1 – Versões dos *softwares* utilizados



(a) Dynamo

(b) Revit

Fonte: Autodesk

Previamente à operação dos *softwares* foi feita leitura dos respectivos manuais, a fim de possibilitar conhecimento das capacidades e limitações de cada um, bem como desenvolver familiaridade com as ferramentas de modo a facilitar seus usos.

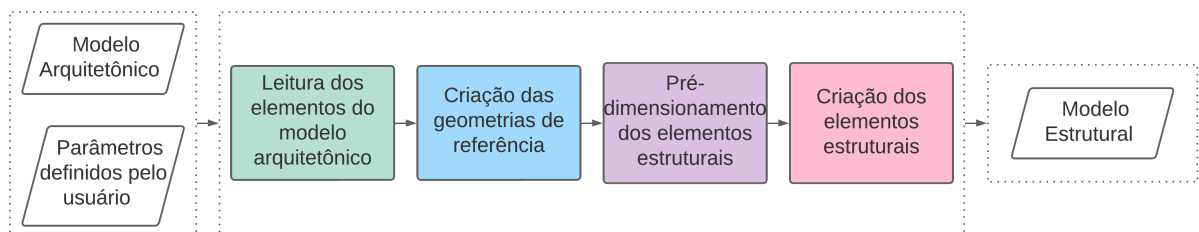
Para desenvolvimento da rotina, foram fixados quais valores deveriam ser passados como entrada e qual a saída esperada. Como entrada, definiu-se o conjunto de dados composto por: modelo da arquitetura feito em Revit; parâmetros que direcionam a concepção de uma estrutura, como resistência do concreto, taxa de armadura de elementos, carga por área da laje, tensão admissível do solo e número de diferentes valores de altura de viga. Como saída, definiu-se os elementos componentes da estrutura modelados no Revit: vigas; pilares; sapatas.

Pensou-se na segmentação do algoritmo de processamento dos dados em partes, de maneira que cada qual viria a ter suas entradas e saídas próprias e pudessem ser alteradas de maneira independente. Essa divisão em partes foi pensada como segue:

1. Leitura dos elementos do modelo arquitetônico
2. Criação das geometrias de referência
3. Pré-dimensionamento dos elementos estruturais
4. Criação dos elementos estruturais

A Figura 2.2 ilustra o fluxograma pensado para o algoritmo, na qual estão identificadas as entradas, saída e o fluxo de processamento dos dados.

Figura 2.2 – Fluxograma do algoritmo



Fonte: Autor

A fim de documentar o código criado, fez-se uso de capturas de tela e comentários explicando os trechos de código retratados. De modo a auxiliar o entendimento da rotina, também optou-se por ilustrar os resultados produzidos da execução de cada parte do algoritmo.

Pensando na avaliação da saída produzida pelo código, tomou-se um projeto arquitetônico cujo modelo em Revit se encontrava disponível e cuja solução estrutural já se encontrasse concebida, a fim de permitir a comparação desta com aquela gerada pelo código.

3 CONCEPÇÃO DE ESTRUTURA EM CONCRETO ARMADO

3.1 Conceito de estrutura

Segundo Rebello (2000, p. 21), por estrutura entende-se “um conjunto, um sistema, composto de elementos que se inter-relacionam para desempenhar uma função, permanente ou não”.

De maneira análoga, Sussekind (1981, p. 1) define:

As estruturas se compõem de uma ou mais peças, ligadas entre si e ao meio exterior de modo a formar um conjunto estável, isto é, um conjunto capaz de receber solicitações externas, absorvê-las internamente e transmiti-las até seus apoios, onde estas solicitações externas encontrarão seu sistema estático equilibrante.

Ainda, Martha (2010) considera que uma estrutura pode ser destinada como um empreendimento por si próprio, como pontes e estádios esportivos, ou pode ser o esqueleto de um outro empreendimento, a exemplo dos edifícios.

Mais especificamente no caso das edificações, Rebello (2000) explica que a estrutura é formada por um conjunto de elementos, nomeadamente lajes, vigas e pilares. Esses elementos se inter-relacionam: laje se apoia em viga e viga se apoia em pilar, a fim de criar um espaço no qual é possível o exercício de diversas atividades.

É através desse conjunto de elementos que se cria um “caminho” para que as forças possam transitar até seu destino final: o solo. O caminho das forças, como é chamado, é ilustrado na Figura 3.1.

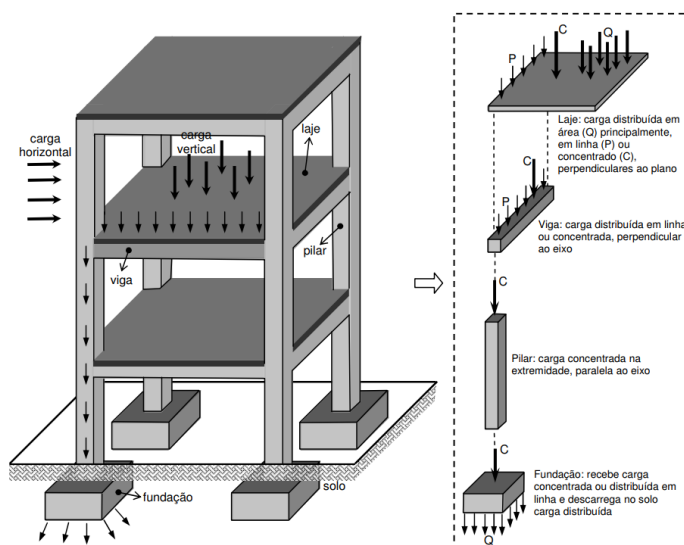
Nesse âmbito, surge a engenharia estrutural como área que trata do “planejamento, projeto, construção e manutenção de sistemas estruturais para transporte, moradia, trabalho e lazer”, como afirma Martha (2010, p. 1).

Por sua vez, o projeto estrutural tem como resultado a especificação da estrutura de forma completa, abrangendo aspectos de locação e detalhes necessários para sua construção. Para isso, o projetista parte de uma concepção geral e finaliza com a documentação que permite a execução.

A concepção estrutural se caracteriza como etapa inicial de um projeto estrutural. De acordo com Rebello (2000, p. 26):

Conceber uma estrutura é ter consciência da possibilidade da sua existência; é perceber a sua relação com o espaço gerado; é perceber o sistema ou sistemas capazes de transmitir cargas ao solo, da forma mais natural; é identificar os materiais que, de maneira mais adequada, se adaptam a esses sistemas.

Figura 3.1 – Caminho das forças em uma estrutura de edifício



Fonte: Cunha (2014, p. 7)

As soluções para uma concepção estrutural são inúmeras. De antemão, pode-se antecipar que a melhor estrutura não existe. O que existe é uma boa solução, que atende bem a certos requisitos.

Assim sendo, anterior à concepção propriamente dita, é necessário que se defina a hierarquia de requisitos que devem ser atendidos, de maneira que a solução adotada atenda muito bem os mais importantes e bem os menos importantes. Martha (2010) aponta essas requisitos como sendo condições de segurança, de utilização, econômicas, estéticas, ambientais, construtivas e legais da estrutura.

É importante frisar que a concepção de uma estrutura não se confunde com o cálculo da mesma. “Não é o cálculo que concebe uma forma, mas sim o esforço idealizador da mente humana. O cálculo existe para comprovar e corrigir o que se intuiu”, como explica Rebello (2000, p. 27). Sendo assim, a atividade de concepção antecede ao cálculo que viabiliza a solução pensada.

Ao passo que o cálculo estrutural é tarefa reservada ao profissional engenheiro, conceber uma estrutura é uma atividade intimamente ligada à criação da arquitetura. De fato, conforme Rebello (2000, p. 26), “a concepção de uma forma implica na concepção de uma estrutura e, em consequência, dos materiais e processos para materializá-la. A estrutura e a forma são um só objeto e, assim sendo, conceber uma implica em conceber outra e vice-versa”.

Do exposto, conclui-se que a concepção estrutural é tarefa conjunta do responsável pela concepção da arquitetura e do engenheiro responsável pela viabilização da mesma através do cálculo.

Para entendimento dos pormenores de uma concepção estrutural, faz-se necessário o

conhecimento do comportamento das partes constituintes de uma estrutura: os elementos estruturais, assunto a ser tratado na seção 3.2.

3.2 Elementos estruturais

Carvalho e Filho (2014) definem elemento estrutural como peças com uma ou duas dimensões preponderantes em relação as demais, a exemplo de vigas, lajes e pilares, que compõem uma estrutura. Ao modo como os elementos estruturais são arranjados é o que se define por sistema estrutural.

Outro conceito importante, abordado pela análise estrutural, é o de modelo estrutural. Segundo Martha (2010), o modelo estrutural é uma representação matemática da estrutura, na qual são incorporadas as hipóteses para descrever o comportamento da mesma. Tais hipóteses incluem: geometria do modelo, condições de suporte, comportamento dos materiais e solicitações atuantes.

A identificação dos elementos componentes de uma estrutura se dá através da discretização que, segundo Carvalho e Filho (2014, p. 25), “consiste em desmembrá-las em elementos cujos comportamentos possam ser admitidos já conhecidos e de fácil estudo. Essa técnica possibilita que se consiga, da maneira mais simples possível, analisar uma estrutura com resultados satisfatórios”.

Rebello (2000) aponta que os elementos estruturais se diferenciam por suas relações geométricas. São essas relações que conferem a cada um dos diferentes elementos particularidades que os tornam capazes de figurar em diversas soluções de sistemas estruturais. Os elementos estruturais são classificados quanto às suas relações geométricas em três tipos básicos: o bloco, a barra e a lâmina.

O bloco é o elemento estrutural que apresenta a mesma ordem de grandeza para as três dimensões. Rebello (2000) considera que seu uso pode ser feito através da aplicação de uma força externa que aproxime blocos colocados lado a lado, evitando que eles escorreguem entre si e, conseqüentemente, possibilitando a criação de um sistema estrutural capaz de vencer um vão reto.

A barra é um elemento estrutural em que uma de suas dimensões, o comprimento, predomina em relação às outras duas, largura e altura da seção transversal. Diferentemente do bloco, a barra apresenta uso mais amplo e possível de ser feito de maneira isolada. A exemplo, pode ser usada para apoiar cargas, como um pilar, ou vencer vãos, como uma viga.

Sussekind (1981) define o eixo de uma barra como sendo o lugar geométrico dos centros de gravidade das seções transversais desse elemento. Esse conceito é importante, uma vez que uma barra pode ser considerada representada pelo seu eixo para fins de estudo estático da peça.

Rebello (2000) afirma que o uso de barras de maneira associada permite a criação de

sistemas estruturais mais complexos. É o exemplo da treliça: sistema composto da associação de pequenas barras, capaz de vencer grandes vãos.

A lâmina, também chamada por elemento de superfície, é um elemento estrutural em que duas de suas dimensões, comprimento e largura, prevalecem em relação a uma terceira, a espessura. A ABNT (2014, 14.4.2) os considera como “elementos em que uma dimensão, usualmente chamada de espessura, é relativamente pequena em face das demais”. São três os tipos de lâmina: membrana, placa e casca.

A membrana é uma lâmina muito fina que apresenta resistência apenas no seu plano. Em razão disso, tende a assumir a forma do carregamento que a solicita, como propõe Rebello (2000).

Por sua vez, a placa é uma lâmina de maior rigidez, o que a torna capaz de vencer vãos e suportar cargas transversais ao seu plano. As lajes são exemplos de placas. A ABNT (2014, 14.4.2.1) define placas como “elementos de superfície plana, sujeitos principalmente a ações normais a seu plano. As placas de concreto são usualmente denominadas lajes”.

Por último, a casca é um caso particular de lâmina que apresenta características intermediárias entre a membrana e a placa, sendo um elemento estrutural de pequena espessura e capaz de suportar cargas transversais ao seu plano. Isso só é possível em razão da presença de curvaturas em seu plano, como explica Rebello (2000).

O material constituinte de um elemento estrutural é parâmetro determinante na sua capacidade de suportar cargas, entretanto não é o único. Rebello (2000, p. 28) afirma que “não é só a resistência do material que garante a um elemento estrutural a capacidade de suportar cargas. Sua forma é muitas vezes mais determinante da sua resistência do que a própria resistência do material.”

Com isso, a adequação da forma do elemento estrutural para o vão e carregamento ao qual será submetido, permite que até mesmo materiais frágeis sejam aproveitados estruturalmente.

Dessa forma, a determinação dos elementos estruturais componentes do sistema a ser adotado passa pelo estudo dos esforços solicitantes, da geometria do elemento e do material constituinte, discutidos nas seções 3.3, 3.4 e 3.5 respectivamente.

3.3 Esforços solicitantes

Martha (2010) afirma que uma vez que um modelo estrutural representa o isolamento de uma estrutura em relação ao meio externo, as cargas externas devem estar em equilíbrio com as reações de apoio. Ainda, esse equilíbrio deve ser estático, uma vez que a estrutura considerada deve estar em repouso. A implicação física dessa condição é que as resultantes de forças e momentos, oriundas das cargas externas e reações de apoio, devem ser nulas. À satisfação dessa

condição Rebello (2000) dá o nome de equilíbrio externo.

No entanto, de acordo com Rebello (2000), o equilíbrio externo de uma estrutura é condição necessária, mas não suficiente para sua existência, uma vez que uma estrutura, mesmo aquelas com grande grau de estabilidade (caso das estruturas hiperestáticas), pode vir a perder o que se define como equilíbrio interno, caso o material constituinte da mesma não seja capaz de resistir às tensões internas solicitantes.

Dessa forma, quando as tensões no material são superiores aos valores limites suportados, ocorre perda do equilíbrio interno, acarretando deslocamento relativo entre as seções, o que caracteriza o fenômeno da ruptura.

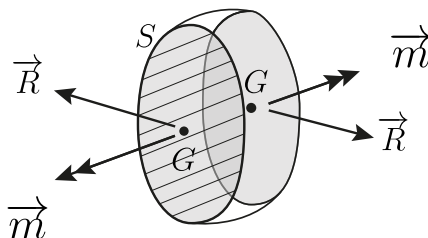
É através da comparação das tensões internas que surgem em decorrência dos carregamentos externos com as tensões limites do material estrutural que se determina a seção do elemento estrutural solicitado. Esse é o princípio do dimensionamento de uma estrutura.

Tais tensões internas solicitantes são resultados da atuação de esforços nas seções dos elementos da estrutura. Martha (2010, p. 24) afirma que “esforços internos em uma estrutura reticulada representam as forças e momentos de ligação entre partes separadas por um corte em uma seção transversal da estrutura. Esforços internos também são integrais de tensões ao longo de uma seção transversal de uma barra.”

A determinação dos esforços em uma seção é orientada por Sussekind (1981, p. 27) através da ideia de equilíbrio de seção (Figura 3.2):

Podemos, então, dizer que uma seção S de um corpo em equilíbrio está, em equilíbrio, submetida a um par de forças $\vec{R}$ e $(-\vec{R})$ e a um par de momentos $\vec{m}$ e $(-\vec{m})$, aplicados no seu centro de gravidade e resultantes da redução, a este centro de gravidade, das forças atuantes, respectivamente, à esquerda e à direita da seção S .

Figura 3.2 – Equilíbrio de seção



Fonte: Adaptado de Sussekind (1981, p. 27)

Rebello (2000) define cinco esforços que podem vir a atuar na seção de um elemento estrutural: tração simples, compressão simples, esforço cortante, momento fletor e momento torsor. Essas solicitações podem atuar concomitantemente em uma mesma seção. A seguir, os esforços serão discutidos, levando em conta as tensões, deformações e o movimento relativo que provocam em uma seção.

O esforço de tração simples é um caso particular do esforço normal. Sua definição é dada por Sussekind (1981, p. 28):

Podemos, então, definir esforço normal atuante numa seção como sendo a soma algébrica das componentes, na direção normal à seção, de cada uma das forças atuantes de um dos lados desta seção. O esforço normal será positivo quando de tração (isto é, quando tender a afastar duas seções infinitamente próximas ou, em linguagem mais simples, quando estiver “saído” da seção), sendo negativo em caso contrário (caso da compressão).

De acordo com Rebello (2000), o esforço de tração simples acarreta o aumento, de maneira uniforme, de todas as fibras do elemento na direção do eixo, em decorrência do surgimento de tensões uniformes ao longo de toda a seção.

O esforço de compressão simples, por sua vez, acarreta na diminuição, de maneira uniforme, de todas as fibras da seção na direção do eixo. Tais deformações são decorrentes de tensões uniformes de compressão que se distribuem pela seção.

O esforço cortante é definido por Sussekind (1981, p. 28) como sendo “igual à soma vetorial das componentes, sobre o plano da seção, das forças situadas de um dos lados desta seção”. Tal esforço tende a promover o deslizamento relativo entre duas seções adjacentes.

Segundo Rebello (2000), o esforço cortante promove o surgimento de tensões de cisalhamento nas seções transversais e longitudinais de um elemento. O efeito concomitante dessas tensões acarreta em tensões normais de tração e compressão inclinadas.

Sussekind (1981, p. 31) define o momento fletor como “a soma vetorial das componentes, sobre o plano da seção, dos momentos de todas as forças situadas de um dos lados da seção em relação ao seu centro de gravidade”.

Considerando duas seções infinitamente próximas, a tendência do momento fletor é de promover a rotação relativa entre as duas seções, em torno de um eixo situado no plano das seções.

Rebello (2000) aponta que o giro relativo entre as seções acarreta na aproximação das seções acima do eixo e no afastamento abaixo do mesmo. Esse efeito acarreta em tensões simultâneas de tração e compressão, que se distribuem pela seção.

Por fim, Sussekind (1981, p. 30) define o momento torsor como sendo “a soma algébrica dos momentos das forças situadas de um dos lados desta seção em relação ao eixo normal à seção que contém o seu centro de gravidade”.

Para duas seções infinitamente próximas, o momento torsor tende a provocar uma rotação relativa entre elas em torno de um eixo que lhes é perpendicular.

Segundo Rebello (2000), a torção provoca o surgimento de tensões de cisalhamento que se distribuem nas seções transversais e longitudinais do elemento. O efeito simultâneo dessas tensões é o aparecimento de tensões inclinadas de tração e compressão.

3.4 Critérios iniciais para definição da seção do elemento

A seção transversal de uma peça estrutural determina o espaço ocupado pela mesma. Por mais que possa ser desejável diminuir esse espaço, tanto por questões estéticas como por aumento de espaço útil em uma edificação, critérios como a maior facilidade de execução e a redução das tensões solicitantes devem ser levados em consideração, como aponta Rebello (2000).

Nesse aspecto, a presente seção se propõe a discutir quais critérios orientam a escolha da seção transversal de um elemento, em função dos esforços solicitantes e tensões decorrentes desses.

Quando solicitada a tração, de acordo com Rebello (2000), a quantidade de material da seção, logo, sua área, é fator determinante na sua capacidade de carga.

De fato, segundo Beer e Johnston (2008), a tensão σ em uma barra de seção transversal de área A e submetida a uma força axial P é dada pela Equação 3.1.

$$\sigma = \frac{P}{A} \quad (3.1)$$

Sendo assim, quanto maior a área da seção, menor a tensão atuante e maior a capacidade de carga do elemento estrutural.

Visando também a economia de espaço ocupado pelo elemento estrutural, Rebello (2000) recomenda a adoção de seção que concentre material bem próximo ao centro de gravidade: é o caso da seção circular cheia. Não só ela, mas as ilustradas na Figura 3.3 respondem bem aos esforços de tração.

Figura 3.3 – Seções recomendadas para elemento solicitado a tração



Fonte: Adaptado de Rebello (2000, p. 62)

Para um elemento estrutural solicitado a compressão axial, Rebello (2000) afirma que a perda de estabilidade da peça pode vir a ocorrer antes que a tensão solicitante atinja o valor que ocasione ruptura. A esse fenômeno dá-se o nome de flambagem.

O fenômeno da flambagem distingue o comportamento de barras tracionadas e barras comprimidas, levando a uma preocupação especial com essa.

Sobre esse fenômeno, Timoshenko (1984, p. 311) afirma que “no caso de pilar longo, comparado com a largura, pode haver falha por flambagem, isto é, por flexão e desvio lateral, ao invés de falhas por compressão direta”.

No caso da compressão simples, Rebello (2000) afirma que não apenas a quantidade de material é fator determinante na resistência da seção, mas também a maneira segundo a qual ele se distribui.

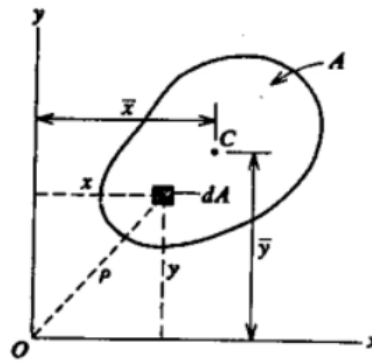
De fato, o equacionamento do fenômeno da flambagem permite que se determine as propriedades da seção comprimida que reduzem o risco de ocorrência de flambagem. Segundo Timoshenko (1984, p. 315), “pode-se considerar a carga crítica como a carga máxima, porquanto as deflexões tornam-se muito grandes, à medida que se aproxima desta carga”. Quanto à carga crítica, o autor propõe que seu valor é dado pela Equação 3.2.

$$P_{cr} = \frac{\pi EI}{L^2} \quad (3.2)$$

Sendo E o módulo de elasticidade do material, I o momento de inércia da seção e L o comprimento do elemento (quando a compressão se dá em elemento biarticulado nas extremidades).

O momento de inércia é justamente a grandeza que mede a distribuição de material em uma seção. Matematicamente, segundo Timoshenko (1983), seu valor é dado pelas equações 3.3 e 3.4, conforme a Figura 3.4.

Figura 3.4 – Seção genérica



Fonte: Timoshenko (1983, p. 203)

$$I_x = \int y^2 dA \quad (3.3)$$

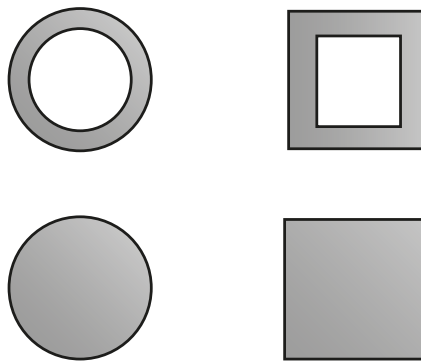
$$I_y = \int x^2 dA \quad (3.4)$$

Na Equação 3.2 o momento de inércia é aquele medido em relação a um eixo que passa pelo centroide da seção transversal. Quanto mais distanciados estiverem os elementos de área do eixo tomado para cálculo, maior será o momento de inércia e maior a carga crítica do elemento.

Dessa forma, como orienta Rebello (2000), para aumento da capacidade de carga de um elemento para efeito da flambagem é necessária a adoção de seção com material distribuído o mais afastado possível do centro de gravidade. Ainda, o fato da flambagem poder ocorrer em qualquer direção leva à necessidade de uma distribuição uniforme de material em todas as direções.

Seguindo os critérios apresentados, Rebello (2000) recomenda a adoção das seções apresentadas na Figura 3.5 para elementos solicitados a compressão. Vale salientar que, visando economia de material, é preferível o uso de seções vazadas.

Figura 3.5 – Seções recomendadas para elemento solicitado a compressão



Fonte: Adptado de Rebello (2000, p. 63)

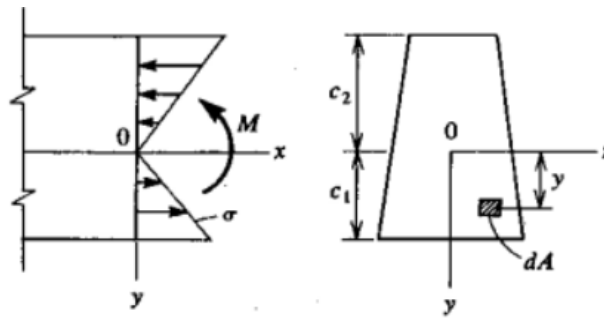
Para elementos solicitados a flexão, Rebello (2000) orienta pela escolha de seção com maior módulo de resistência, a fim de reduzir as tensões solicitantes e aumentar a capacidade de carga.

A explicação dessa proposição passa pelo estudo da flexão em elementos lineares. Timoshenko (1983) propõe a Equação 3.5 para cálculo das tensões normais atuantes na seção de elementos solicitado a flexão.

$$\sigma = \frac{My}{I} \quad (3.5)$$

Sendo M o momento fletor atuante na seção, I o momento de inércia da seção calculado a partir do eixo em torno do qual ocorre a flexão e y a distância entre o elemento no qual se deseja calcular a tensão e o eixo citado. Para validade da equação proposta, o eixo em questão deve ser principal. Tais conceitos são ilustrados na Figura 3.6.

Figura 3.6 – Distribuição de tensões em elemento solicitado a flexão



Fonte: Timoshenko (1983, p. 94)

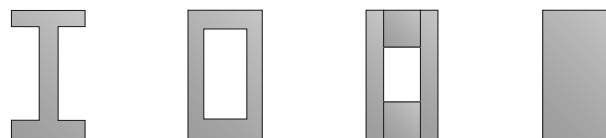
O maior e menor valor de tensão normal em seção solicitada a flexão ocorrem nos elementos de área da seção correspondentes ao maior e menor valores de y respectivamente. Quando positiva a tensão é de tração, quando negativa indica compressão. De acordo com a Figura 3.6, esses valores são respectivamente c_1 e c_2 . Nesse contexto, a Equação 3.5 pode ser reescrita como a Equação 3.6.

$$\sigma_{max} = \frac{M c_1}{I} = \frac{M}{Z_1} \quad \sigma_{min} = -\frac{M c_2}{I} = -\frac{M}{Z_2} \quad (3.6)$$

Os termos $Z_1 = I/c_1$ e $Z_2 = I/c_2$ são definidos como módulos de resistência da seção. Dessa forma, como já apresentado, quanto maior o módulo de resistência da seção, menor a tensão normal máxima decorrente da flexão e maior a capacidade de carga do elemento.

Para elementos solicitados a flexão, Rebello (2000) propõe a adoção das seções apresentadas na Figura 3.7.

Figura 3.7 – Seções recomendadas para elemento solicitado a flexão



Fonte: Adaptado de Rebello (2000, p. 64)

3.5 Materiais estruturais

De acordo com Martha (2010), são diversas as opções de materiais estruturais, desde os mais convencionais, como madeira, aço e concreto armado, àqueles que utilizam fibras vegetais ou mesmo materiais sintéticos, como plásticos. Para que o uso de cada um desses seja possível em uma estrutura, devem apresentar a capacidade de resistir às solicitações que são impostas durante sua vida útil, a exemplo do peso próprio e ações do vento.

Rebello (2000) classifica os materiais em isotrópos, quando apresentam propriedades iguais em todas as direções, ortótropos, quando apresentam propriedades iguais em duas direções e diferentes em uma terceira e anisótropos, quando essas propriedades são diferentes em todas as direções.

Outra classificação importante diz respeito à variação das propriedades em diferentes pontos do material. Quando não há variação, o material é dito homogêneo, caso contrário é heterogêneo.

A classificação do material segundo tais critério se faz importante tendo em vista que se tratando de um material homogêneo e isotrópo não há preocupação quanto à direção em que é solicitado por uma força. O mesmo não acontece com a madeira, material natural e formado por fibras, que apresenta uma direção mais resistente que outra, como exemplifica Rebello (2000).

De acordo com Rebello (2000), as propriedades que interessam na escolha de um material, do ponto de vista estrutural, são: tensão de ruptura, módulo de elasticidade e coeficiente de dilatação térmica.

Beer e Johnston (2008) definem o conceito de carga de ruptura (ou carga última) como sendo a máxima força aplicada em um corpo de prova durante ensaio que ocasiona sua ruptura ou perda de resistência.

Ao projetar um elemento estrutural, deve-se garantir que a maior carga que o solicite seja inferior à carga de ruptura. A esse valor dá-se o nome de carga admissível. A razão entre carga última e admissível recebe o nome de coeficiente de segurança. As tensões de ruptura e admissível são aquelas produzidas pelas cargas correspondentes.

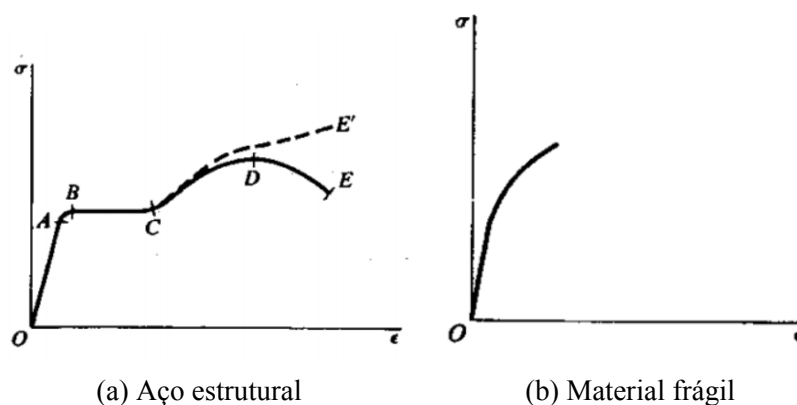
Segundo Timoshenko (1983), o módulo de elasticidade é a constante de proporcionalidade da Equação 3.7, que estabelece a relação entre a tensão (σ) e deformação (ε) em um material.

$$\sigma = E\varepsilon \quad (3.7)$$

O módulo de elasticidade mostra como o material se deforma quando solicitado, definindo se ele é muito ou pouco deformável. Sua quantificação permite, ainda, que se estabeleça como a associação de materiais se comporta em termos de deformação.

Parte desses conceitos pode ser entendida através do diagrama tensão-deformação de um material. De acordo com Timoshenko (1983) esse diagrama é obtido ao se medir as tensões e deformações de um corpo de prova quando submetido a determinado esforço em ensaio. Exemplos desse diagrama são exibidos na Figura 3.8.

Figura 3.8 – Diagramas tensão-deformação



Fonte: Timoshenko (1983, p. 3 e p. 4)

A diferença entre os diagramas exibidos nas Figuras 3.8a e 3.8b permite a introdução do conceito de ductilidade. Timoshenko (1983) classifica um material como dúctil quando é capaz de sofrer grande deformação antes da fratura (Figura 3.8a). Caso contrário, o material é dito frágil (Figura 3.8b)

A Tabela 3.1 apresenta as propriedades mecânicas de alguns materiais, incluindo os materiais estruturais aqui já citados.

Tabela 3.1 – Propriedades mecânicas dos materiais

Material	Massa específica (g/cm ³)	Módulo de elasticidade (kgf/mm ²)	Módulo de elasticidade transversal (kgf/mm ²)	Tensão de escoamento (kgf/mm ²)	Tensão máxima de ruptura (kgf/mm ²)
Alumínio	2,69	7.000	2.800	14	21
Ligas de alumínio	2,77 a 8,32	7.000	2.800	10,5 a 35	14 a 42
Latão	8,32	9.800	3.850	10,5 a 42	28 a 52,5
Bronze	8,32	9.800	3.850	7 a 38,5	21 a 42
Concreto (compressão)	2,36	1.400 a 2.800	-	-	1,4 a 7
Cobre	8,88	10.500	4.200	7 a 31,5	21 a 42
Ferro fundido	7,77	10.500	4.200	4,2 a 28	11,2 a 42
Magnésio	1,78	4.200	1.680	8,4 a 12,6	14 a 21
Aço	7,85	20.300 a 21.000	7.700 a 8.400	21 a 42	35 a 70
Aço (alta resistência)	7,85	20.300 a 21.000	7.700 a 8.400	35 a 112	70 a 196
Tungstênio	18,86	35.000	14.000	-	420
Madeira estrutural (compressão)	0,277 a 0,832	700 a 1.400	-	-	2,8 a 7

Fonte: Adaptado de Timoshenko (1983, p. 6)

Outra propriedade mecânica importante é o coeficiente de dilatação térmica, haja vista

que a associação de materiais com coeficientes de valores diferentes pode vir a acarretar esforços decorrentes das diferentes deformações que experimentam, o que pode ocasionar a ruptura do mais fraco, como afirma Rebello (2000).

Em relação ao uso de materiais associados, é uma solução adotada em dois cenários. O primeiro, quando um dos materiais não apresenta capacidade adequada para absorção de certos esforços. O segundo, quando a associação resulta num material mais interessante do que os dois isoladamente. Um exemplo de material resultante da associação de outros dois é o concreto armado, a ser discutido na seção 3.6.

Além das propriedades estruturais, deve-se levar em conta outros dois fatores na escolha do material estrutural: econômico e estético. De acordo com Rebello (2000), a análise do fator econômico leva em conta questões como quantidade do material, função de como o material responde ao esforço solicitante, e tempo de execução, função das condições de manuseio do material.

Deve-se ressaltar que a escolha do material estrutural impõe restrições na geometria do elemento estrutural, podendo inviabilizar o cumprimento dos critérios de escolha de seção apresentados na seção 3.4. Essas restrições se devem sobretudo a dificuldades de execução intrínsecas a cada material estrutural, como aponta Rebello (2000).

3.6 Concreto armado

Carvalho e Filho (2014) definem o concreto como o resultado da adição sucessiva de diferentes materiais: cimento, água, agregado miúdo e agregado graúdo. A adição dos agregados à mistura objetiva a redução de custo do material sem perda da qualidade, uma vez que o cimento é oneroso.

Em função da necessidade de melhoria de características do concreto em estado fresco ou endurecido, segundo Araújo (2003), podem ser acrescentados aditivos químicos (retardadores ou aceleradores de pega, plastificantes e superplastificantes) e adições minerais (escórias de alto-forno, pozolanas, fillers calcários e microssílica) à mistura.

Rebello (2000) considera que, mesmo se tratando de um material resultante da mistura de outros, o concreto pode ser considerado isótropo e homogêneo, em decorrência de apresentar as mesmas propriedades físicas em todas as direções.

A respeito das propriedades do concreto, devem ser avaliadas nas duas fases do material: o concreto fresco e o concreto endurecido. Para o concreto fresco, segundo Carvalho e Filho (2014), interessam a consistência, trabalhabilidade e homogeneidade do material, de maneira que sejam adequadas às condições de uso, a fim de se obter produto final com qualidade e durabilidade.

Para o concreto endurecido, as propriedades de interesse são as mecânicas, discutidas

na seção 3.5 para os materiais estruturais em geral. Para o concreto, a principal propriedade de interesse é a resistência a compressão, a partir da qual é possível a determinação das demais propriedades, fazendo uso de expressões empíricas fornecidas pela NBR 6118:2014.

A resistência do concreto a compressão é determinada por ensaios. A moldagem dos corpos de prova é normatizada pela NBR 5738:2015 e a realização do ensaio pela NBR 5739:2018. Dos valores obtidos de ensaio, obtém-se o valor característico da resistência do concreto a compressão (f_{ck}). A ABNT (2014, 12.2) apresenta a seguinte definição para valor característico:

Os valores característicos f_k das resistências são os que, em um lote de material, têm uma determinada probabilidade de serem ultrapassados, no sentido desfavorável para a segurança.

Usualmente é de interesse a resistência característica inferior $f_{k,inf}$, cujo valor é menor que a resistência média f_m , embora por vezes haja interesse na resistência característica superior $f_{k,sup}$, cujo valor é maior que f_m .

Para os efeitos desta Norma, a resistência característica inferior é admitida como sendo o valor que tem apenas 5% de probabilidade de não ser atingido pelos elementos de um dado lote de material.

A resistência a tração do concreto pode ser avaliada por meio das equações 3.8 a 3.11, fornecidas pela ABNT (2014, 8.2.5), onde $f_{ct,m}$ e f_{ck} são expressos em MPa.

$$f_{ctk,inf} = 0,7 f_{ct,m} \quad (3.8)$$

$$f_{ctk,sup} = 1,3 f_{ct,m} \quad (3.9)$$

Para concretos de classes até C50:

$$f_{ct,m} = 0,3 f_{ck}^{2/3} \quad (3.10)$$

Para concretos de classes C55 até C90:

$$f_{ct,m} = 2,21 \ln(1 + 0,11 f_{ck}) \quad (3.11)$$

Os valores de resistência a tração fornecidos pelas equações 3.8 a 3.11 para as diferentes classes de concreto são apresentados na Tabela 3.2.

Tabela 3.2 – Valores de resistência a tração em função da resistência característica à compressão do concreto

Classes de resistência	C20	C25	C30	C35	C40	C45	C50	C60	C70	C80	C90
f_{ck}	20	25	30	35	40	45	50	60	70	80	90
f_{ctm}	2,2	2,6	2,9	3,2	3,5	3,8	4,1	4,5	4,8	5,0	5,3
$f_{ctk,inf}$	1,5	1,8	2,0	2,2	2,5	2,7	2,9	3,1	3,3	3,5	3,7
$f_{ctk,sup}$	2,9	3,3	3,8	4,2	4,6	4,9	5,3	5,8	6,2	6,6	6,9
$f_{ctk,inf}/f_{ck}$	8%	7%	7%	6%	6%	6%	6%	5%	5%	4%	4%
$f_{ctk,sup}/f_{ck}$	14%	13%	13%	12%	11%	11%	11%	10%	9%	8%	8%

Fonte: Autor

O módulo de elasticidade inicial (E_{ci}) do concreto é dado pelas equações 3.12 e 3.13, fornecidas pela ABNT (2014, 8.2.8).

$$E_{ci} = \alpha_E \cdot 5600 \sqrt{f_{ck}} \quad \text{para } f_{ck} \text{ de 20 MPa a 50 MPa} \quad (3.12)$$

$$E_{ci} = 21,5 \cdot 10^3 \alpha_E \cdot \left(\frac{f_{ck}}{10} + 1,25 \right)^{1/3} \quad \text{para } f_{ck} \text{ de 55 MPa a 90 MPa} \quad (3.13)$$

Sendo $\alpha_E = 1,2$ para basalto e diabásio, $\alpha_E = 1,0$ para granito e gnaisse, $\alpha_E = 0,9$ para calcário e $\alpha_E = 0,7$ para arenito. E_{ci} e f_{ck} são dados em MPa.

Ainda, o módulo de elasticidade secante (E_{cs}) é fornecido pelas equações 3.14 e 3.15.

$$\alpha_i = 0,8 + 0,2 \cdot \frac{f_{ck}}{80} \leq 1,0 \quad (3.14)$$

$$E_{cs} = \alpha_i \cdot E_{ci} \quad (3.15)$$

Quanto ao coeficiente de dilatação térmica, a ABNT (2014, 8.2.3) considera que “para efeito de análise estrutural, [...] pode ser admitido como sendo igual a $10^{-5}/^{\circ}\text{C}$ ”.

Apesar de apresentar boa resistência a compressão, Carvalho e Filho (2014) consideram o uso do concreto como material estrutural inadequado. Isso deve ao fato do material apresentar baixa resistência à tração: cerca de 10% do valor da resistência à compressão (Tabela 3.2).

Essa deficiência do concreto conduz ao seu uso associado com o aço, originando o concreto armado. De acordo com Rebello (2000), a resistência do concreto armado à tração simples é dada apenas pela armação.

Ainda, o aço no concreto armado é capaz de absorver esforços de compressão, aumentando a capacidade de carga de peças comprimidas, como diz Araújo (2003).

Para que isso seja possível, aço e concreto devem trabalhar solidariamente, o que é possível em consequência das forças de aderência que surgem entre o aço e o concreto. Nesse contexto, Carvalho e Filho (2014, p. 20) afirmam:

As barras de aço tracionadas (armadura tracionada) só funcionam quando, pela deformação do concreto que as envolve, começam a ser alongadas, o que caracteriza as armaduras passivas. É a aderência que faz com que o concreto armado se comporte como material estrutural.

Nesse cenário, o concreto tracionado desempenha papel de proteger a armadura. Situações de grande solicitação da armadura podem conduzir a fissuras de grande abertura no concreto, levando à necessidade de reduzir as tensões de trabalho, como forma de controlar a abertura dessas fissuras, como afirma Rebello (2000).

Quanto ao uso em elementos estruturais, Rebello (2000) considera que o uso do concreto armado pode ser feito naqueles solicitados a compressão, uma vez que o concreto apresenta boa resistência a esse esforço e há redução no risco de flambagem, uma vez que as seções de concreto armado são robustas.

Se tratando de elementos solicitados a flexão, uma vez que o concreto resiste bem à compressão e a armadura passiva resiste às solicitações de tração, o uso de concreto armado é recomendado, como no caso de vigas.

No geral, Rebello (2000, p. 74) conclui que o concreto armado “é um material que não apresenta vantagens de uso quando predominam esforços de tração simples, sendo bastante vantajoso quando solicitado a esforços de compressão simples e a momento fletor”.

Na confecção de peças em concreto armado, a armadura é previamente posicionada no interior da fôrma e, em seguida, o concreto fresco é lançado envolvendo-as, a medida em que é realizado o processo de adensamento, como explica Bastos (2019). Após o processo de cura e endurecimento do concreto, retira-se a fôrma e assim a peça de concreto armado é originada.

Rebello (2000) aponta que a concepção de peças em concreto armado a partir de fôrmas permite a obtenção de elementos de qualquer forma de seção, residindo a dificuldade na obtenção das fôrmas.

Como visto na seção 3.4, a seção recomendada para elementos solicitados à compressão são aquelas que apresentam material distribuído de maneira mais afastada do centro de gravidade, exemplificados na Figura 3.5. No caso do uso de concreto armado, Rebello (2000) aponta que as seções vazadas apresentam grande dificuldade de execução, não sendo, portanto, usadas. Dessa forma, opta-se pelo uso de seção circular e seção quadrada.

No caso da flexão, no qual opta-se por seções com maior módulo de resistência, tais quais as exibidas na Figura 3.7, a seção I tem processo de execução facilitado através dos processos de pré-moldagem e a seção retangular é facilmente executada com concreto armado, sendo a mais usada em decorrência da facilidade de produção, como aponta Rebello (2000).

Por fim, Araújo (2003, p. 2) aponta as principais vantagens do concreto armado, sendo elas: “economia; facilidade de execução em diversos tipos de formas; resistência ao fogo, aos agentes atmosféricos e ao desgaste mecânico; praticamente não requer manutenção ou conservação; permite facilmente a construção de estruturas hiperestáticas”.

Como desvantagens, Araújo (2003, p. 2) cita: “o elevado peso das construções; dificuldades para a execução de reformas ou demolições; menor proteção térmica”.

3.7 Pré-dimensionamento dos elementos estruturais

O pré-dimensionamento dos elementos de uma estrutura é norteado pelos parâmetros que orientam sua concepção. Segundo Cunha (2014, p. 29), são eles: “material utilizado, vãos das peças, carregamento, condições de apoio, atendimento às condições de resistência, de estabilidade e de deformação/deslocamentos (estados limites últimos e de utilização)”.

A presente seção se propõe a apresentar critérios que orientem o pré-dimensionamento dos elementos de estruturas de concreto armado: lajes, vigas e pilares. Ainda, serão abordados os conceitos referentes ao pré-dimensionamento de um caso particular de solução de fundação direta: a sapata.

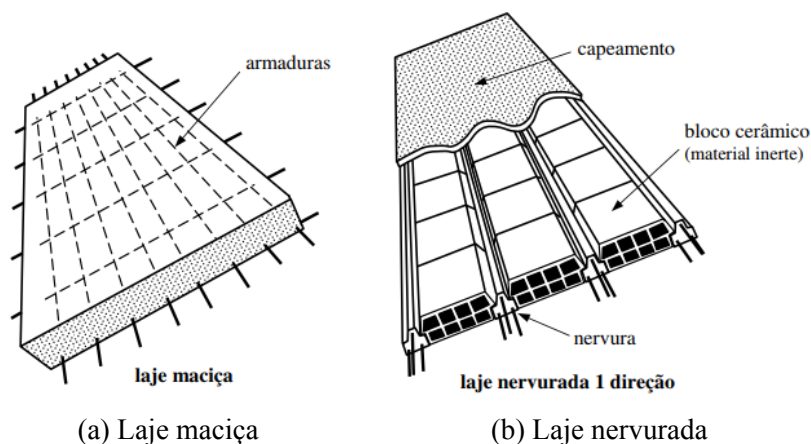
3.7.1 Lajes

De acordo com a ABNT (2014, 14.4.2.1), considera-se como placas “elementos de superfície plana, sujeitos principalmente a ações normais a seu plano. As placas de concreto são usualmente denominadas lajes. Placas com espessura maior que $1/3$ do vão devem ser estudadas como placas espessas”.

De acordo com Cunha (2014) as lajes atuam transferindo as ações verticais permanentes e variáveis nos pavimentos para os elementos de apoio (vigas e pilares). Ainda, desempenham papel de diafragma rígido, distribuindo as ações horizontais entre os sistemas de contraventamento.

Carvalho e Filho (2014) consideram que o pavimento de uma edificação pode ser concebido com elementos pré-moldados ou moldados no local. Quando moldado no local, pode-se fazer uso de uma única laje, sem vigas, ou um conjunto de lajes apoiadas em vigas. Ainda, pode-se fazer escolha entre o uso de lajes maciças (Figura 3.9a) e lajes nervuradas (Figura 3.9b).

Figura 3.9 – Laje maciça e nervurada



Fonte: Cunha (2014, p. 72)

Segundo Cunha (2014), as lajes nervuradas podem ser entendidas como constituídas por um conjunto de vigas (nervuras) solidarizado por uma mesa. É feito uso de material inerte entre as nervuras, que garante redução no peso e melhor aproveitamento do aço e do concreto. Estruturalmente, a resistência a tração é garantida pelas nervuras e a compressão pela mesa.

A ABNT (2014, 14.7.7) considera que “lajes nervuradas são as lajes moldadas no local ou com nervuras pré-moldadas, cuja zona de tração para momentos positivos esteja localizada nas nervuras entre as quais pode ser colocado material inerte”.

As lajes maciças, por sua vez, possuem arranjo estrutural mais simples, bastando que haja o posicionamento de armadura nas regiões tracionadas. Cunha (2014) aponta para a possibilidade de uso de telas soldadas como forma de agilizar o processo construtivo. Ainda, aponta para desvantagens dessa solução, como o peso próprio elevado, limitando seu uso para grandes vãos.

A ABNT (2014, 13.2.4) estabelece restrições para os parâmetros que definem a seção transversal de uma laje. Se tratando de uma laje maciça, devem ser atendidos os seguintes valores mínimos de espessura:

- a) 7 cm para cobertura não em balanço;
- b) 8 cm para lajes de piso não em balanço;
- c) 10 cm para lajes em balanço;
- d) 10 cm para lajes que suportem veículos de peso total menor ou igual a 30 kN;
- e) 12 cm para lajes que suportem veículos de peso total maior que 30 kN;
- f) 15 cm para lajes com protensão apoiadas em vigas, com o mínimo de $\frac{l}{42}$ para lajes de piso biapoiadas e $\frac{l}{50}$ para lajes de piso contínuas;
- g) 16 cm para lajes lisas e 14 cm para lajes-cogumelo, fora do capitel.

Para lajes nervuradas, as restrições impostas são:

A espessura da mesa, quando não existirem tubulações horizontais embutidas, deve ser maior ou igual a $1/15$ da distância entre as faces das nervuras (o) e não menor que 4 cm.

O valor mínimo absoluto da espessura da mesa deve ser 5 cm, quando existirem tubulações embutidas de diâmetro menor ou igual a 10 mm. Para tubulações com diâmetro Φ maior que 10 mm, a mesa deve ter a espessura mínima de 4 cm + Φ , ou 4 cm + 2Φ no caso de haver cruzamento destas tubulações.

A espessura das nervuras não pode ser inferior a 5 cm.

Nervuras com espessura menor que 8 cm não podem conter armadura de compressão.

O pré-dimensionamento de lajes é orientado por Cunha (2014) através do método do vão, segundo o qual se estabelece um intervalo para possíveis valores da espessura (h) da laje. Esses intervalos são definidos como segue:

$$L/40 \leq h \leq L/32 \quad (\text{Laje maciça armada em duas direções}) \quad (3.16)$$

$$L/38 \leq h \leq L/30 \quad (\text{Laje maciça armada em uma direção}) \quad (3.17)$$

$$L/30 \leq h \leq L/25 \quad (\text{Laje nervurada}) \quad (3.18)$$

$$L/36 \leq h \leq L/28 \quad (\text{Laje lisa}) \quad (3.19)$$

$$L/38 \leq h \leq L/30 \quad (\text{Laje cogumelo}) \quad (3.20)$$

$$L/40 \leq h \leq L/30 \quad (\text{Laje nervurada protendida}) \quad (3.21)$$

$$L/35 \leq h \leq L/25 \quad (\text{Laje cogumelo protendida}) \quad (3.22)$$

Para lajes armadas em duas direções toma-se L como sendo a média dos vãos. Para lajes armadas em uma direção L é o vão correspondente à direção das armaduras.

3.7.2 Vigas

De acordo com ABNT (2014, 14.4.1), a viga é considerada um elemento linear (seção 3.2) em que predomina a flexão.

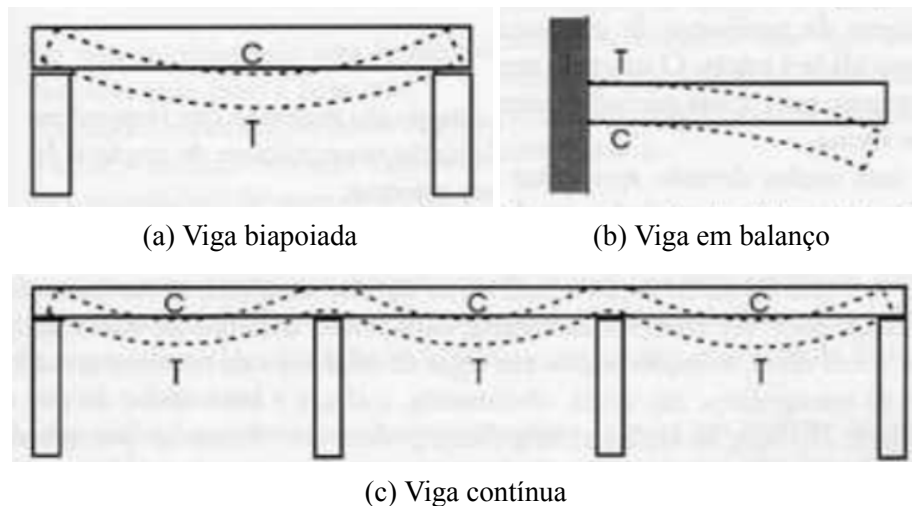
Segundo Cunha (2014), nas estruturas de edificações as vigas atuam conferindo apoio às lajes e, eventualmente, a outras vigas. Por vez, se apoiam nos pilares, atuando juntamente com esses na formação de pórticos rígidos que asseguram a estabilidade global do conjunto.

Como visto na seção 3.3, o momento fletor ocasiona a rotação relativa entre duas seções adjacentes. O esforço cortante, por sua vez, acarreta no deslizamento relativo entre seções. A soma desses efeitos acarreta na deformação do eixo da viga antes reto. Aos deslocamentos verticais do eixo, dá-se o nome de flecha, como propõe Rebello (2000).

Rebello (2000) propõe a classificação das vigas quanto ao número de apoios. Quando apoiadas em dois pontos, são vigas biapoiadas. Quando apoiadas em um único ponto (engaste),

são vigas em balanço. Quando apoiadas em mais de dois pontos, são vigas contínuas. A Figura 3.10 ilustra as diferentes classificações.

Figura 3.10 – Classificação das vigas quanto ao número de apoios

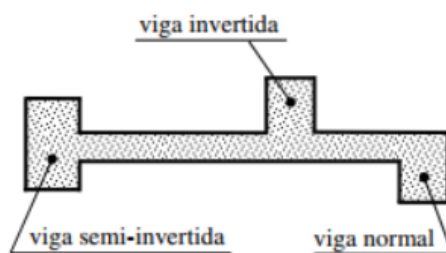


Fonte: Rebello (2000, p. 99)

Nas vigas biapoiadas surgem tensões de compressão nas fibras superiores e de tração nas fibras inferiores. Nas vigas em balanço há uma inversão, caracterizada por tensões de tração na porção superior e compressão na inferior. Por fim, nas vigas contínuas há comportamento de viga biapoiada nos vãos e inversão na região sobre os apoios.

Uma outra classificação das vigas diz respeito a sua posição em relação ao nível da laje, podendo ser viga normal, semi-invertida ou invertida, conforme ilustrado na Figura 3.11.

Figura 3.11 – Classificação da viga quanto a posição em relação a laje



Fonte: Cunha (2014, p. 103)

A viga é um elemento que se caracteriza pela transmissão de cargas verticais ao longo de um vão através de um eixo horizontal. A implicação disso, como aponta Rebello (2000), é o surgimento de um vão livre e aproveitável sob a viga. Tal fato torna a viga o elemento estrutural mais usado, principalmente na sustentação de pisos.

Como visto na seção 3.6 a principal seção utilizada em elementos feitos em concreto armado solicitados a flexão é a retangular. O pré-dimensionamento de vigas nesse material e seção é conduzido pelo critério, proposto por Cunha (2014), que determina a altura da viga (h) a partir do comprimento do seu vão (L) por meio da Equação 3.23.

$$h = L/10 \quad (3.23)$$

Para vigas contínuas, deve-se tomar o valor do maior vão.

Bastos (2017, p. 4) considera que “a altura das vigas deve ser preferencialmente modulada de 5 em 5 cm, ou de 10 em 10 cm. A altura mínima indicada é de 25 cm. Vigas contínuas devem ter a altura dos vãos obedecendo uma certa padronização, a fim de evitar várias alturas diferentes”.

Quanto ao pré-dimensionamento da largura da viga, deve-se atentar aos valores mínimos estabelecidos pela ABNT (2014) em seu item 13.2.2, no qual:

A seção transversal das vigas não pode apresentar largura menor que 12 cm e a das vigas-parade, menor que 15 cm. Estes limites podem ser reduzidos, respeitando-se um mínimo absoluto de 10 cm em casos excepcionais, sendo obrigatoriamente respeitadas as seguintes condições:

- a) alojamento das armaduras e suas interferências com as armaduras de outros elementos estruturais, respeitando os espaçamentos e cobrimentos estabelecidos nesta Norma;
- b) lançamento e vibração do concreto de acordo com a ABNT NBR 14931.

Para pré-dimensionamento da largura da viga (b_w), Cunha (2014) recomenda o uso da equação Equação 3.24, sendo L o comprimento do vão.

$$b_w = \begin{cases} 12 \text{ cm} & (L \leq 4 \text{ m}) \\ 20 \text{ cm} & (4 \text{ m} < L \leq 8 \text{ m}) \\ 25 \text{ a } 30 \text{ cm} & (L > 8 \text{ m}) \end{cases} \quad (3.24)$$

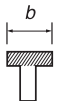
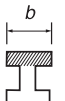
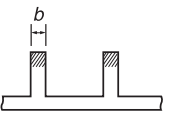
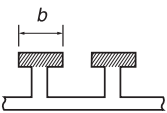
Por fim, a ABNT (2014, 15.10) recomenda a adoção de valor mínimo de largura para vigas de concreto, visando a garantia da segurança à instabilidade lateral. Esse valor é dado pelas equações 3.25 e 3.26.

$$b \geq l_0/50 \quad (3.25)$$

$$b \geq \beta_{fl} \cdot h \quad (3.26)$$

Sendo b a largura da viga na zona comprimida, h a altura total da viga, l_0 o comprimento do flange comprimido (medido entre suportes que garantem o contraventamento lateral) e β_{fl} o coeficiente que depende da forma da viga, fornecido na Tabela 3.3.

Tabela 3.3 – Valores de β_{fl}

Tipologia da viga	Valores de β_{fl}
  	0,40
 	0,20
onde  Zona comprimida	

Fonte: Adaptado de ABNT (2014, 15.10)

3.7.3 Pilares

A ABNT (2014, 14.4.1.2) considera pilares como “elementos lineares de eixo reto, usualmente dispostos na vertical, em que as forças normais de compressão são preponderantes”.

Segundo Rebello (2000), quando solicitado apenas por cargas verticais, o pilar sofre apenas compressão simples. No entanto, ações horizontais decorrentes do vento ou da frenagem de veículos levam ao surgimento de momento fletor.

Nas edificações, os pilares atuam dando apoio às vigas e transmitindo as cargas às fundações. Ainda, participam da estrutura de contraventamento, garantindo a estabilidade global do conjunto, como propõe Cunha (2014).

Cunha (2014) considera que, no projeto de uma edificação, além do dimensionamento do elementos estruturais, deve-se garantir a estabilidade global. Essa garantia é feita pela consideração da estrutura de contraventamento: conjunto de pórticos formados por pilares e vigas da edificação.

Não são todos os pilares que participam do sistema responsável pela estabilidade global. Diante disso, os pilares são divididos em pertencentes à estrutura de contraventamento e contraventados. A parcela de pilares contraventados pode ser considerada como pertencente a uma estrutura indeslocável, como afirma Cunha (2014).

Quanto às dimensões da seção do pilar, a ABNT (2014) considera que não pode apresentar menor dimensão inferior a 19 cm. Em casos especiais, permite-se a adoção de dimensões entre 19 cm e 14 cm, desde que os esforços de cálculo sejam multiplicados por um coeficiente adicional γ_n , dado na Tabela 3.4.

Tabela 3.4 – Valores do coeficiente adicional γ_n para pilares

b cm	≥ 19	18	17	16	15	14
γ_n	1,00	1,05	1,10	1,15	1,20	1,25

onde

$$\gamma_n = 1,95 - 0,05 b;$$

b é a menor dimensão da seção transversal, expressa em centímetros (cm).

NOTA O coeficiente γ_n deve majorar os esforços solicitantes finais de cálculo quando de seu dimensionamento.

Fonte: ABNT (2014, 13.2.3)

Como visto na seção 3.6, na determinação da seção de pilares de concreto armado, é preferível que se opte pela circular ou quadrada. A seção retangular também pode ser usada, no entanto com menos vantagens como considera Rebello (2000).

Para pré-dimensionamento da seção transversal de pilares em concreto armado, Cunha (2014) sugere o uso da Equação 3.27.

$$A_c = \frac{N_d}{\sigma_{id}} \gamma_{corr} \quad (3.27)$$

Sendo A_c a área da seção transversal, N_d o carregamento de cálculo do pilar, σ_{id} a tensão ideal de cálculo do concreto e γ_{corr} o fator de correção. O valor de σ_{id} é dado pela Equação 3.28.

$$\sigma_{id} = 0,85f_{cd} + \rho(f_{sd} - 0,85f_{cd}) \quad (3.28)$$

Sendo ρ a taxa de armadura da seção, f_{cd} o valor de cálculo da resistência do concreto e f_{sd} o valor de cálculo da resistência do aço. Para aço CA-50 adota-se $f_{sd} = 420$ MPa, valor de tensão referente a deformação de 0,2%. Os valores σ_{id} são indicados nas Tabelas 3.5 a 3.8, considerando aço CA-50.

Tabela 3.5 – Tensão ideal de cálculo do concreto (σ_{id}) para $\rho = 2\%$

$\rho = 2\%$							
f_{ck} (MPa)	20	25	30	35	40	45	50
σ_{id} (kgf/cm ²)	203	233	263	292	322	352	382

Fonte: Autor

Tabela 3.6 – Tensão ideal de cálculo do concreto (σ_{id}) para $\rho = 4\%$

$\rho = 4\%$							
f_{ck} (MPa)	20	25	30	35	40	45	50
σ_{id} (kgf/cm ²)	285	314	343	372	401	430	459

Fonte: Autor

Tabela 3.7 – Tensão ideal de cálculo do concreto (σ_{id}) para $\rho = 6\%$

$\rho = 6\%$							
f_{ck} (MPa)	20	25	30	35	40	45	50
σ_{id} (kgf/cm ²)	366	395	423	452	480	509	537

Fonte: Autor

Tabela 3.8 – Tensão ideal de cálculo do concreto (σ_{id}) para $\rho = 8\%$

$\rho = 8\%$							
f_{ck} (MPa)	20	25	30	35	40	45	50
σ_{id} (kgf/cm ²)	448	476	504	532	559	587	615

Fonte: Autor

O fator de correção γ_{corr} tem valor dado em função da posição do pilar, segundo a Tabela 3.9.

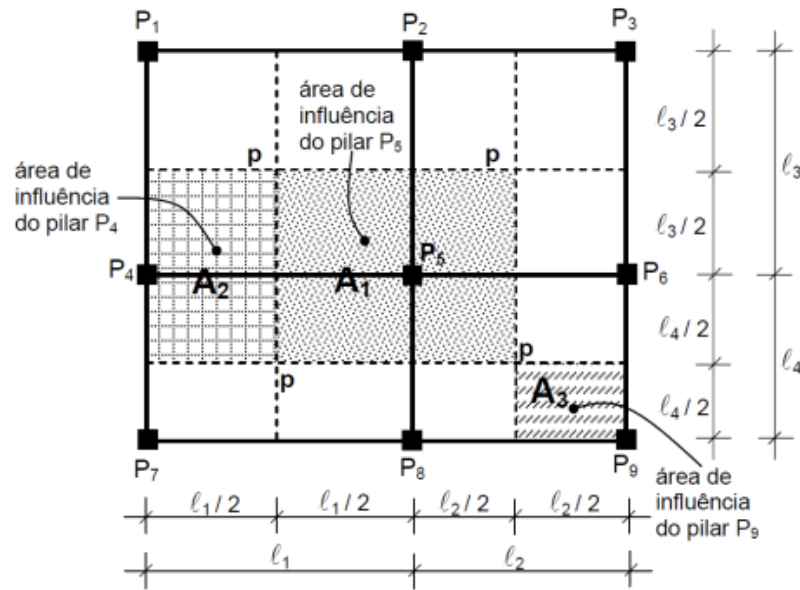
Tabela 3.9 – Coeficiente γ_{corr}

Posição do pilar	Coeficiente γ_{corr}
Intermediário	1,5
Extremidade	2
Canto	4

Fonte: Adaptado de Cunha (2014, p. 119)

O carregamento de cálculo do pilar N_d pode ser obtido pelo processo das áreas de influência, proposto por Cunha (2014), no qual se atribui uma área de influência (A_i) para cada pilar em um pavimento, como ilustra a Figura 3.12.

Figura 3.12 – Obtenção de área de influência de pilar



Fonte: Cunha (2014, p. 47)

Conhecida a área de influência do pilar, a carga N_d é determinada pela Equação 3.29.

$$N_d = 1,4 A_i (q_T (n_p + 0,7) + q_c) \quad (3.29)$$

Sendo q_T a carga do pavimento tipo, q_c a carga da cobertura e n_p o número de pavimentos. Para fins de pré-dimensionamento, Cunha (2014) sugere a adoção dos valores propostos nas equações 3.30 e 3.31.

$$q_T = \begin{cases} 1500 \text{ kgf/m}^2 & \text{(Laje maciça ou nervurada com blocos de concreto)} \\ 1200 \text{ kgf/m}^2 & \text{(Laje maciça ou nervurada com tijolos cerâmicos)} \\ 1000 \text{ kgf/m}^2 & \text{(Laje nervurada com blocos leves (EPS))} \\ 800 \text{ kgf/m}^2 & \text{(Laje não maciça com paredes em gesso acartonado)} \\ 600 \text{ kgf/m}^2 & \text{(Laje de forro)} \end{cases} \quad (3.30)$$

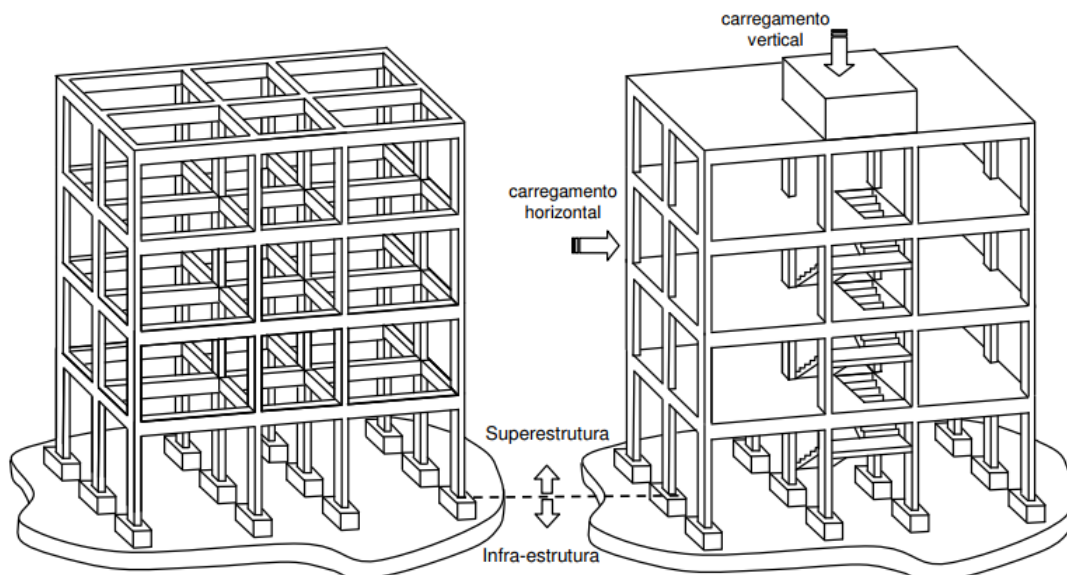
$$q_c = \begin{cases} 140 \text{ kgf/m}^2 & \text{(Cobertura com telhas cerâmicas e estrutura de madeira)} \\ 60 \text{ kgf/m}^2 & \text{(Cobertura com telhas de fibrocimento e estrutura de madeira)} \\ 50 \text{ kgf/m}^2 & \text{(Cobertura com telhas de alumínio e estrutura de aço)} \end{cases} \quad (3.31)$$

3.7.4 Sapatas

Em um projeto estrutural, a superestrutura é composta pelas vigas, lajes e pilares, podendo-se acrescentar reservatórios elevados e escadas. Para apoio desse conjunto no solo, é feito uso

de elementos estruturais que formam a infra-estrutura ou fundação, como propõe Cunha (2014) e é ilustrado na Figura 3.13.

Figura 3.13 – Superestrutura e infra-estrutura



Fonte: Cunha (2014, p. 11)

Os elementos de fundação podem ser separados em dois grupos: fundações rasas (ou diretas) e fundações profundas. A ABNT (1996, 3.1 e 3.8) considera elementos de fundação direta aqueles “em que a profundidade de assentamento em relação ao terreno adjacente é inferior a duas vezes a menor dimensão da fundação”. Por sua vez, considera-se elemento de fundação profunda aquele que “está assente em profundidade superior ao dobro de sua menor dimensão em planta, e no mínimo 3 m, salvo justificativa”.

Velloso e Lopes (1998) afirmam que a distinção entre a fundação direta e a profunda diz respeito ao mecanismo de ruptura de base. Enquanto na fundação direta esse mecanismo atinge a superfície do terreno, o mesmo não acontece na fundação profunda.

Uma outra distinção entre elementos de fundação direta e profunda é a maneira em que ocorre a transmissão de carga ao terreno. A ABNT (1996, 3.1 e 3.8) aponta que na fundação direta essa transmissão se dá por “pressões distribuídas sob a base da fundação”. Na fundação profunda, por vez, o elemento “transmite a carga ao terreno pela base (resistência de ponta), por sua superfície lateral (resistência de fuste) ou por uma combinação das duas”.

De acordo com Velloso e Lopes (1998), um elemento de fundação direta pode ser: bloco, sapata, viga de fundação, grelha, sapata associada e radier. Já um elemento de fundação profunda pode ser classificado como: estaca, tubulão e caixão.

Tratando do caso particular da sapata, Velloso e Lopes (1998, p. 212) a definem como “elemento de fundação de concreto armado [...] utilizando armadura para resistir aos esforços

de tração”. De maneira análoga, a ABNT (1996, 3.2) considera:

Elemento de fundação superficial de concreto armado, dimensionado de modo que as tensões de tração nele produzidas não sejam resistidas pelo concreto, mas sim pelo emprego da armadura. Pode possuir espessura constante ou variável, sendo sua base em planta normalmente quadrada, retangular ou trapezoidal.

Teixeira e Godoy (1998) apontam que a sapata, tal qual o bloco, é o elemento de fundação mais simples e, quando possível sua adoção, o mais econômico.

O projeto de uma fundação em sapata (e em outro tipo qualquer) necessita do conhecimento referente a: topografia da área; dados geológicos-geotécnicos; dados da estrutura a construir e dados sobre construções vizinhas, como propõem Velloso e Lopes (1998).

Uma vez concebido, de acordo com Velloso e Lopes (1998), o projeto deve atender aos seguintes requisitos: deformações aceitáveis sob as condições de trabalho; segurança adequada ao colapso do solo de fundação; segurança adequada ao colapso dos elementos estruturais; segurança adequada ao tombamento e deslizamento; níveis de vibração compatíveis com o uso da obra.

De acordo com Teixeira e Godoy (1998), o dimensionamento de sapatas e demais fundações diretas passa pela determinação de um valor de tensão admissível (σ_{adm}). Os autores propõem que esse valor será sempre fixado levando-se em conta dois critérios: o de segurança à ruptura e o de recalques admissíveis.

Teixeira e Godoy (1998) consideram que a segurança à ruptura é satisfeita mediante a aplicação de um coeficiente de segurança ao valor de tensão que provoca a ruptura do solo. Já o recalque admissível deve ser tal que seja suportado pela superestrutura.

A ABNT (1996) considera quatro possíveis métodos para determinação da tensão admissível: teóricos; provas de carga sobre placa; semi-empíricos e empíricos. Dentre os métodos empíricos, pode-se citar a adoção dos valores propostos na Tabela 3.10¹.

¹ Para rochas alteradas ou em decomposição, têm que ser levados em conta a natureza da rocha matriz e o grau de decomposição ou alteração

Tabela 3.10 – Tensões admissíveis

Classe	Descrição	Valor (MPa)
1	Rocha sã, maciça, sem laminação ou sinal de decomposição	3,0
2	Rocha laminadas, com pequenas fissuras, estratificadas	1,5
3	Rochas alteradas ou em decomposição	*
4	Solos granulares concrecionados - conglomerados	1,0
5	Solos pedregulhosos compactos a muito compactos	0,6
6	Solos pedregulhosos fofos	0,3
7	Areias muito compactas ($N_{spt} > 40$)	0,5
8	Areias compactas ($19 \leq N_{spt} \leq 40$)	0,4
9	Areias medianamente compactas ($9 \leq N_{spt} \leq 18$)	0,2
10	Argilas duras ($N_{spt} > 19$)	0,3
11	Argilas rijas ($11 \leq N_{spt} \leq 19$)	0,2
12	Argilas médias ($6 < N_{spt} \leq 10$)	0,1
13	Siltes duros (muito compactos)	0,3
14	Siltes rijos (compactos)	0,2
15	Siltes médios (medianamente compactos)	0,1

Fonte: Adaptado de ABNT (1996, 6.2)

Outro método empírico, apresentado por Soares (2021), consiste no uso da Equação 3.32, válida para solos argilosos.

$$\sigma_{adm} = \frac{N_{spt}}{5} \text{ kgf/cm}^2 \quad (3.32)$$

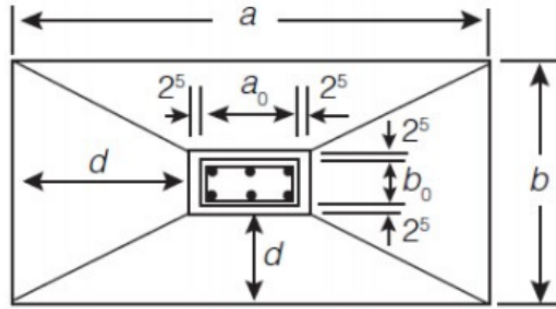
Sendo N_{spt} a média dos valores de N_{spt} referentes às camadas sob a fundação e consideradas sob efeito de aumento de tensão.

Uma vez determinado o valor de tensão admissível, Teixeira e Godoy (1998) propõem a Equação 3.33 para cálculo das dimensões da sapata.

$$a \cdot b = \frac{P}{\sigma_{adm}} \quad (3.33)$$

Sendo P a carga vertical no pilar e a e b as dimensões da sapata em planta, conforme a Figura 3.14.

Figura 3.14 – Vista em planta de sapata



Fonte: Soares (2021)

Teixeira e Godoy (1998) consideram que o dimensionamento econômico é aquele que conduz a momentos aproximadamente iguais nas duas abas, o que é garantido por valores iguais de d nas duas direções (Figura 3.14). Sendo a_0 e b_0 as dimensões do pilar que se apoia na sapata, a Equação 3.34 é proposta.

$$\begin{cases} a = 2d + 2 \cdot 2,5\text{cm} + a_0 \\ b = 2d + 2 \cdot 2,5\text{cm} + b_0 \end{cases} \Rightarrow a - a_0 = b - b_0 \quad (3.34)$$

O caso particular de se ter um pilar quadrado conduz à Equação 3.35 para dimensionamento da sapata.

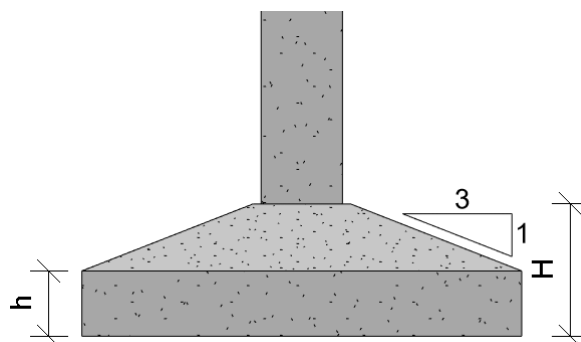
$$a = \sqrt{\frac{P}{\sigma_{adm}}} \quad (3.35)$$

Quanto as dimensões em corte, Rebello (2008) recomenda o uso das Equações 3.36 e 3.37, conforme a Figura 3.15. Ainda, é recomendada a adoção de uma inclinação de 1:3 (vertical:horizontal) para o trecho inclinado.

$$h \geq 10 \text{ cm} \quad (3.36)$$

$$H = 30\% \text{ do lado maior da sapata} \quad (3.37)$$

Figura 3.15 – Vista em corte de sapata



Fonte: Adaptado de Rebello (2008)

Soares (2021) recomenda que o centro de gravidade da sapata coincida com o centro de carga do pilar; a sapata não apresente dimensão inferior a 60 cm e a relação entre suas dimensões em planta seja menor ou igual a 2,5.

Ainda, o autor aponta para a necessidade de realizar a ligação de fundações isoladas por meio de cintas em duas direções ortogonais, de maneira a impedir deslocamentos horizontais; limitar rotações; servir de fundação para as paredes e reduzir recalques diferenciais.

Teixeira e Godoy (1998) apontam para o caso de não se poder projetar sapatas isoladas para cada pilar, em decorrência de cargas elevadas e proximidade entre os pilares, levando à adoção de sapatas associadas, que atendem dois ou mais pilares.

Nesse caso, deve-se centrar a sapata no centro de carga do conjunto de pilares por ela atendidos. Quando todos os pilares de uma estrutura transmitirem as cargas ao solo através de uma única sapata, tem-se o que se denomina de fundação em radier.

3.8 Lançamento dos elementos estruturais

Por lançamento de elementos estruturais, Rebello (2000, p. 193) define como “o procedimento de locar, sobre a arquitetura, as vigas e pilares resultantes da concepção estrutural adotada”.

Como discutido na seção 3.1, a melhor concepção não existe. Faz-se necessário que se proponham diferentes soluções para o lançamento e, a partir da hierarquia de critérios definida, se escolha a que melhor os atende.

Rebello (2000) recomenda que o início do lançamento se dê pela locação das vigas, tendo em vista que iniciar pela locação dos pilares pode acarretar em grande indefinição, uma vez que podem ser locados em qualquer número e posição. Ainda, deve-se considerar o caminho natural das cargas, que passa pelas vigas e em seguida pelos pilares.

Cunha (2014) recomenda a mesma sequência de lançamento, no entanto aponta para a possibilidade de se iniciar pela locação dos pilares, desde que ao final do lançamento verifique-

se a compatibilização completa com o projeto arquitetônico.

Ainda, deve-se iniciar o lançamento pelo pavimento tipo. Caso não exista, o lançamento deve partir dos pavimentos superiores para os inferiores, de modo que se verifiquem eventuais interferências no posicionamento dos elementos.

Os itens apresentados a seguir são recomendações de Rebello (2000) e Cunha (2014) que norteiam o lançamento dos elementos da estrutura.

- a) Vigas devem ser locadas de maneira que conduzam a panos de laje da mesma ordem de grandeza;
- b) Vãos de laje considerados econômicos apresentam dimensões entre 3,5 m e 5 m. Para laje armada em uma direção adotar vão entre 2 m e 5 m. Para laje armada em duas direções adotar vão entre 3 m e 7 m. Para lajes nervuradas os vãos podem chegar a 12 m.
- c) Se possível, considerar vigas para apoiar todos os bordos da laje. O lançamento de uma viga pode ser feito para dividir uma laje com grandes dimensões.
- d) A locação de vigas deve ser feita preferencialmente sob alvenarias;
- e) A locação de vigas deve ser feita preferencialmente sobre alvenarias;
- f) Pode ser feito uso de viga invertida quando o lançamento dessa vier a interferir esteticamente no espaço onde se projeta;
- g) Pode ser feito uso de viga embutida, com altura igual à espessura da laje, denominada viga-faixa;
- h) Posicionar as vigas de forma que, em conjunto com os pilares, formem pórticos que garantam a estabilidade da estrutura;
- i) Vãos de vigas devem apresentar valor entre 3 m e 7 m;
- j) O número de pilares lançado deve ser tal que facilite a execução e a torne economicamente viável;
- k) Lançar inicialmente pilares de canto, seguido dos pilares comuns a todos os pavimentos (região da escada, elevadores e reservatório), demais pilares do contorno do pavimento e, por fim, os pilares internos;
- l) Evitar pilares muito próximos, o que dificulta a execução das fundações;
- m) Adotar espaçamento entre pilares entre 4 m e 6 m;
- n) Locação de pilares deve resultar em vãos de viga com mesma ordem de grandeza;
- o) Sempre que possível, os pilares devem ser lançados de maneira que se criem balanços nas vigas que aliviem o vão central;
- p) O posicionamento dos pilares deve ser tal que não gere descontinuidades da fundação à cobertura;

- q) O lançamento de pilares deve ser feito prioritariamente nos pontos de encontro de vigas;
- r) Os pilares devem ser locados sobre os mesmos eixos sempre que possível;
- s) Se possível, dispor pilares com maior dimensão paralela à menor dimensão em planta da edificação, a fim de propiciar maior rigidez às solicitações horizontais;
- t) Evitar grande variedade nas dimensões dos elementos estruturais. Três dimensões para vigas e pilares é um número razoável.

Ao final do lançamento, Cunha (2014) aponta para a necessidade de se verificar a interferência dos elementos estruturais com a arquitetura em todos os pavimentos. A exemplo, verificar o impacto dos pilares lançados com áreas de estacionamento e manobra de veículos.

Quanto ao Item a), panos de lajes de dimensões com diferenças significativas levam a espessuras diferentes, acarretando em dificuldade no processo construtivo. Nesse caso, adotar a mesma espessura para todas as lajes leva ao superdimensionamento das mesmas.

Os Itens d) e e) apontam para o posicionamento de vigas sob e sobre as alvenarias, respectivamente.

Sendo a viga mais rígida que a laje, sofre menores deformações decorrentes da carga da alvenaria, evitando trincas indesejáveis.

Ao locar a viga sobre a alvenaria, evita-se que a laje, ao se deformar, se apoie sobre o elemento de vedação, o que acarretaria no surgimento de esforços não previstos no dimensionamento da laje. Caso não seja possível atender a esse critério, é preferível que se execute a alvenaria após a laje ter sofrido as maiores deformações.

Quanto ao Item n), Rebello (2000) aponta que diferenças de até 20% entre os comprimentos dos vãos levam a um lançamento econômico. Diferenças significativas entre vãos sucessivos pode acarretar no surgimento de momento negativo no vão menor, o que pode conduzir ao tracionamento do pilar em sua extremidade.

Em relação ao Item l), descontinuidades nos pilares podem ser contornadas por vigas de transição. No entanto, o uso dessa solução acaba por encarecer a estrutura.

Por fim, Rebello (2000) aponta que o Item r) facilita a locação dos pilares em obra.

4 MODELAGEM DA INFORMAÇÃO DA CONSTRUÇÃO (BIM)

4.1 Conceito de BIM

Segundo Netto (2020), as ferramentas CAD (Computer Aided Design) revolucionaram a maneira de projetar, permitindo a obtenção de desenhos em 2D ou 3D que representam objetos da vida real de maneira mais rápida e precisa.

Ao passo em que houve a evolução dos sistemas CAD, Eastman et al. (2014) afirma que usuários passaram a ter a necessidade de compartilhar dados associados com o projeto, transferindo o foco dos desenhos e das imagens 3D para os próprios dados.

Nesse contexto surge o *Building Information Modeling* (BIM), com o qual construções passam a ser projetadas de nova forma, sem que projetistas se empenhem na criação de vistas 2D, mas no projeto virtual desse, como considera Netto (2020).

Quanto ao conceito de BIM, a Autodesk (2023b) considera:

A BIM (Modelagem de Informação da Construção) é o processo holístico de criação e gerenciamento de informações para um recurso construído. Com base em um modelo inteligente e habilitada por uma plataforma na nuvem, a BIM integra dados estruturados e multidisciplinares para produzir uma representação digital de um recurso em todo seu ciclo de vida, desde o planejamento e o projeto até a construção e as operações.

Em outras palavras, Netto (2020) considera que o BIM trata da construção virtual de um edifício antes da sua construção efetiva. No modelo digital criado, todas as informações necessárias para a construção são armazenadas. Tem-se assim um banco de dados, que permite a simulação virtual da construção verdadeira, beneficiando todos os profissionais envolvidos no processo, desde o planejamento inicial do projeto passando pelo anteprojeto e culminando no projeto executivo, construção e manutenção do edifício com o modelo 3D e suas informações.

Eastman et al. (2014, p. 13) considera o “BIM como uma tecnologia de modelagem e um conjunto associado de processos para produzir, comunicar e analisar modelos de construção”.

Um modelo de construção contém componentes de construção, que se caracterizam, segundo Eastman et al. (2014, p. 13), como “representações digitais inteligentes (objetos) que “sabem” o que eles são, e que podem ser associados com atributos (gráficos e de dados) computáveis e regras paramétricas”.

Os componentes do modelo incluem dados que descrevem seu comportamento. Tais dados são consistentes e não redundantes, de maneira que modificações nos dados de um componente são representadas em todas as suas visualizações, e coordenados, de forma que todas

as visualizações de um componente são representadas de maneira coordenada, como descreve Eastman et al. (2014).

Netto (2020, p. 23) aponta que através do uso de uma ferramenta BIM, é possível:

Examinar o edifício de qualquer ponto. Testar e analisar o edifício. Verificar interferências entre as várias disciplinas atuantes na construção. Quantificar os elementos necessários à construção. Simular a construção e analisar os custos em cada uma das fases. Gerar uma documentação vinculada ao modelo que seja fiel a ele.

4.2 Modelagem paramétrica

A principal tecnologia que distingue aplicativos para projetos em BIM de sistemas CAD é modelagem paramétrica baseada em objetos, na qual objetos não são representados com geometrias e propriedades fixas, mas através de parâmetros e regras que determinam a geometria e características não geométricas, como considera Eastman et al. (2014).

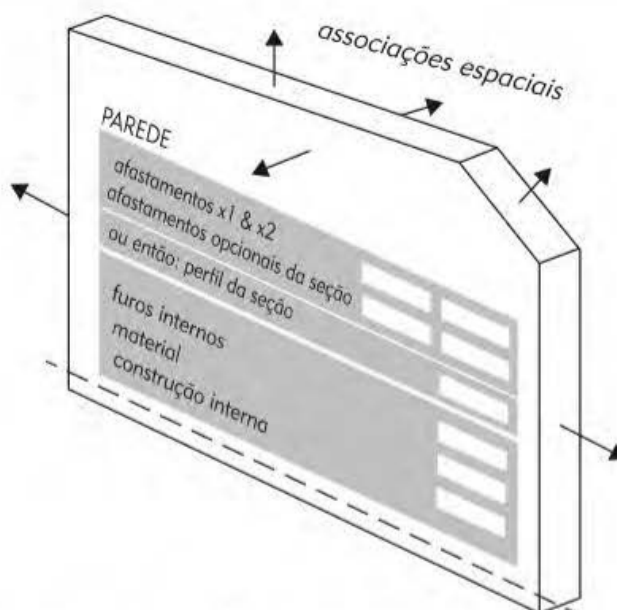
De acordo com Eastman et al. (2014, p. 29):

No CAD 3D tradicional, cada aspecto da geometria de um elemento deve ser editado manualmente pelos usuários; em um modelador paramétrico, as geometrias da forma e do conjunto ajustam-se automaticamente às modificações do contexto e aos controles de alto nível do usuário.

Netto (2020) exemplifica o conceito de modelagem paramétrica através do comportamento de uma parede, que pode ter sua altura definida como anexada a um pavimento ou mesmo a um telhado, de forma que alterações nesses elementos provocam alterações na parede.

A Figura 4.1 ilustra uma família de parede e alguns dos parâmetros a ela associados. De acordo com Eastman et al. (2014, p. 29), “usa-se a denominação família de parede porque ela é capaz de gerar muitas instâncias de seu tipo em diferentes localizações e com parâmetros variados”.

Figura 4.1 – Estrutura conceitual de uma família de objeto parede



Fonte: Eastman et al. (2014, p. 30)

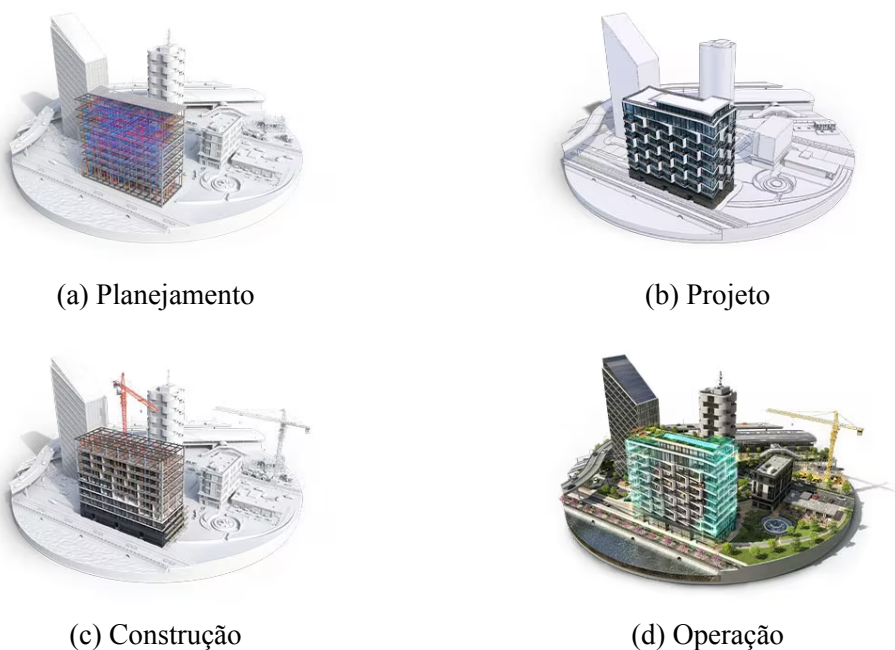
Assim sendo, Eastman et al. (2014, p. 14) apresenta uma definição alternativa para ferramentas BIM, considerando essas como “tecnologias que permitem aos usuários produzirem modelos de construção que consistem em objetos paramétricos”. Por objeto paramétrico, segundo o autor, entende-se como aquele que atende aos seguintes critérios:

- a) Consistem em definições geométricas e dados e regras associadas.
- b) A geometria é integrada de maneira não redundante e não permite inconsistências. Quando um objeto é mostrado em 3D, a forma não pode ser representada internamente de maneira redundante, por exemplo, como múltiplas vistas 2D. Uma planta e uma elevação de dado objeto devem sempre ser consistentes. As dimensões não podem ser “falsas”.
- c) As regras paramétricas para os objetos modificam automaticamente as geometrias associadas quando inseridas em um modelo de construção ou quando modificações são feitas em objetos associados. Por exemplo, uma porta se ajusta imediatamente a uma parede, um interruptor se localizará automaticamente próximo ao lado certo da porta, uma parede automaticamente se redimensionará para se juntar a um teto ou telhado, etc.
- d) Os objetos podem ser definidos em diferentes níveis de agregação, então podemos definir uma parede, assim como seus respectivos componentes. Os objetos podem ser definidos e gerenciados em qualquer número de níveis hierárquicos. Por exemplo, se o peso de um subcomponente de uma parede muda, o peso de toda a parede também deve mudar.
- e) As regras dos objetos podem identificar quando determinada modificação viola a viabilidade do objeto no que diz respeito a tamanho, construtibilidade, etc.
- f) Os objetos têm a habilidade de vincular-se a ou receber, divulgar ou exportar conjuntos de atributos, por exemplo, materiais estruturais, dados acústicos, dados de energia, etc. para outras aplicações e modelos.

4.3 Usos do BIM

A Autodesk (2023b) considera que “o processo de BIM oferece suporte à criação de dados inteligentes que podem ser usados em todo o ciclo de vida de um projeto de edificações ou infraestrutura”, como ilustrado na Figura 4.2.

Figura 4.2 – Uso do BIM no ciclo de vida da construção



Fonte: Autodesk (2023b)

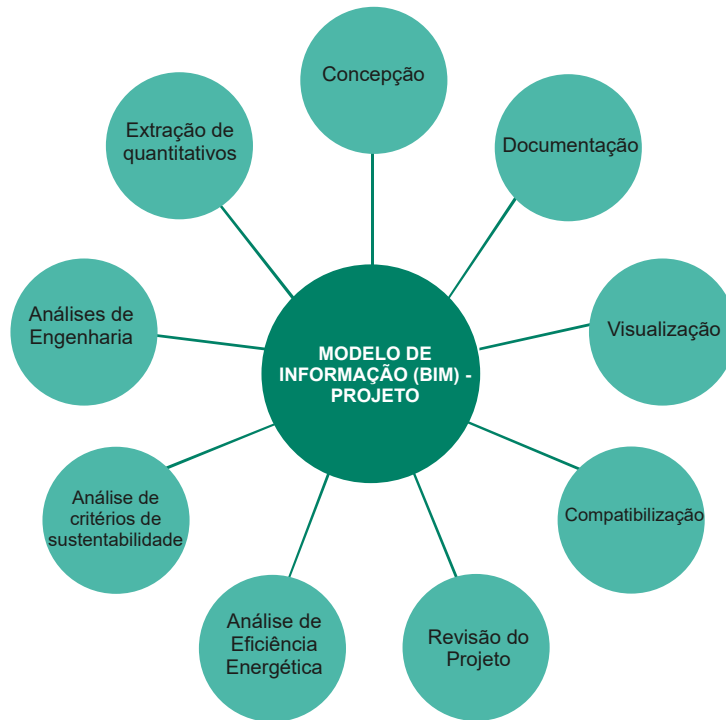
O uso do BIM em todo o ciclo de vida de um projeto é reforçado pela AsBEA (2013), que considera o BIM como um processo que parte de modelos tridimensionais no qual todas as informações relativas à construção são alocadas em um só modelo integrado, paramétrico, intercambiável e passível de simulação, cujo uso pode ser feito nas fases de concepção dos projetos, obra e vida útil do espaço construído.

Na etapa de pré-construção, Eastman et al. (2014) aponta que o uso do BIM permite a análise de viabilidade do projeto, determinando se a construção pode ser feita dentro de um dado orçamento e cronograma, através de um modelo de construção vinculado a uma base de dado de custos. Ainda, há incremento na qualidade do projeto através da avaliação de alternativas de projeto.

Na etapa de projeto, Eastman et al. (2014) considera que o BIM permite a visualização antecipada e mais precisa de um projeto; correções automáticas quando mudanças são feitas; geração de desenhos 2D precisos e consistentes; colaboração antecipada entre múltiplas disciplinas; verificação facilitada das intenções de projeto; extração de estimativas de custo e incrementação da eficiência energética e de sustentabilidade.

A AsBEA (2013) considera o uso do BIM na etapa de projeto para concepção; documentação; visualização; compatibilização; análise de eficiência energética; análise de critérios de sustentabilidade; análises de engenharia e extração de quantitativos, como ilustrado na Figura 4.3.

Figura 4.3 – Usos do BIM para projeto

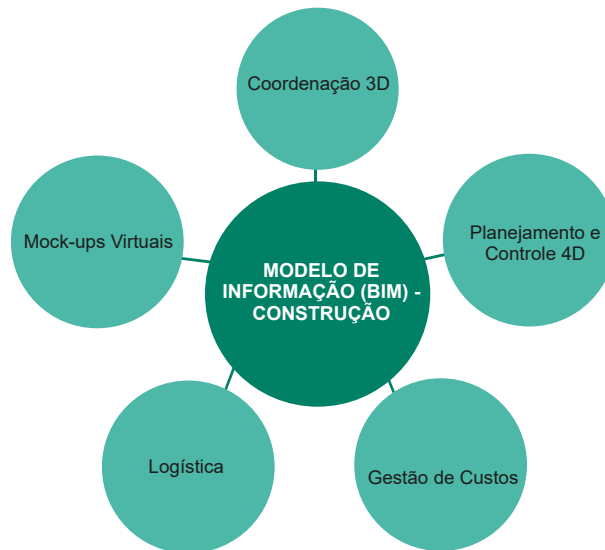


Fonte: AsBEA (2013, p. 9)

Na etapa de construção, segundo Eastman et al. (2014), o uso do BIM permite a sincronização de projeto e planejamento; detecção de interferências; reação rápida a problemas de projeto ou do canteiro; uso do modelo de projeto como base para componentes fabricados; melhor implementação e técnicas de construção enxuta e sincronização da aquisição de materiais com o projeto e a construção.

De acordo a AsBEA (2013), o uso do BIM na etapa de construção permite a coordenação 3D; planejamento e controle 4D; gestão de custos; logística e mock-ups virtuais, conforme a Figura 4.4.

Figura 4.4 – Usos do BIM para construção

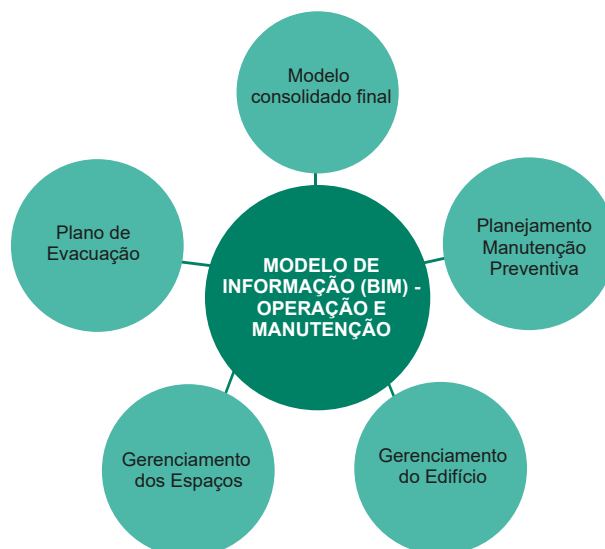


Fonte: AsBEA (2013, p. 9)

Na etapa pós-construção, Eastman et al. (2014) considera o uso do BIM no melhor gerenciamento e operação das edificações, permitindo, por exemplo, a verificação do funcionamento de todos os sistemas da edificação, e na integração com sistemas de operação e gerenciamento de facilidades.

Na operação e manutenção, a AsBEA (2013) aponta para o uso do BIM na obtenção de modelo consolidado final; no planejamento e manutenção preventiva; no gerenciamento do edifício; no gerenciamento dos espaços e na concepção de plano de evacuação, conforme a Figura 4.5.

Figura 4.5 – Usos do BIM para operação e manutenção



Fonte: AsBEA (2013, p. 9)

4.4 Revit

O Revit é um software BIM comercializado pela Autodesk. Sobre seu uso, a Autodesk (2023a) traz a seguinte descrição:

O Revit é uma plataforma de projeto e documentação que suporta projetos, desenhos e tabelas necessários para a Modelagem de Informação da Construção (BIM).[...] No modelo do Revit, todas as folhas de desenho, as vistas 2D e 3D e as tabelas consistem em apresentações de informação do mesmo modelo de construção virtual. Enquanto você trabalha no modelo de construção, o Revit coleta informações sobre o projeto de construção e coordena essas informações com todas as outras representações do projeto. O mecanismo de alteração paramétrica do Revit coordena automaticamente as alterações realizadas em qualquer parte — em vistas de modelo, folhas de desenho, tabelas, cortes e plantas.

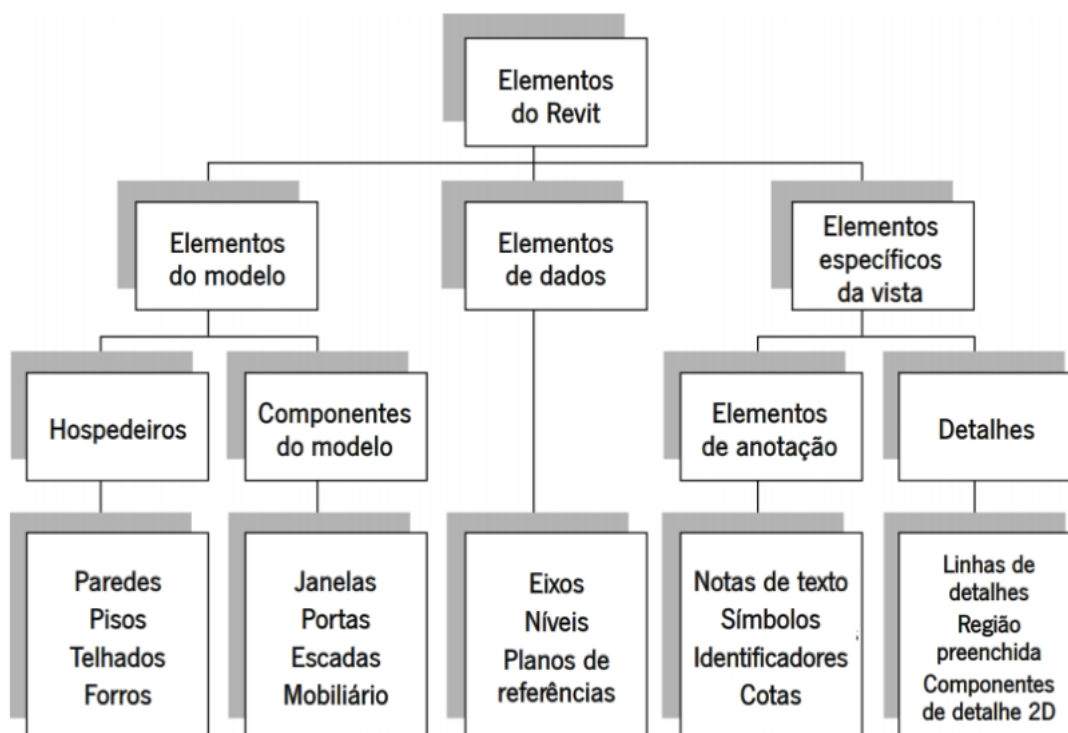
Ao conceber um projeto com o Revit, a modelagem envolve a criação de elementos construtivos, tais como paredes, pisos, vigas, pilares entre outros, os quais têm todas as características definidas em suas propriedades, como explica Netto (2020).

Ainda, de acordo com Netto (2020), a alteração de qualquer das propriedades de um elemento em uma vista, repercute em todas as demais em que é exibido, já que se trata do mesmo objeto. Esse comportamento garante a consistência do modelo e da documentação gerada a partir dele.

Isso é explicado pela Autodesk (2023a) através do conceito de modelagem paramétrica: “altere qualquer coisa a qualquer momento e em qualquer lugar no projeto, e o Revit coordena aquela alteração através de todo o projeto”. Essa é uma das características fundamentais do Revit, o tornando capaz de coordenar alterações e manter a consistência em todos os momentos.

De acordo com a Autodesk (2023a), elementos no Revit podem ser de três tipos: elementos do modelo, elementos de dados e elementos específicos da vista. Essa relação é ilustrada na Figura 4.6.

Figura 4.6 – Elementos para o Revit



Fonte: Netto (2020, p. 42)

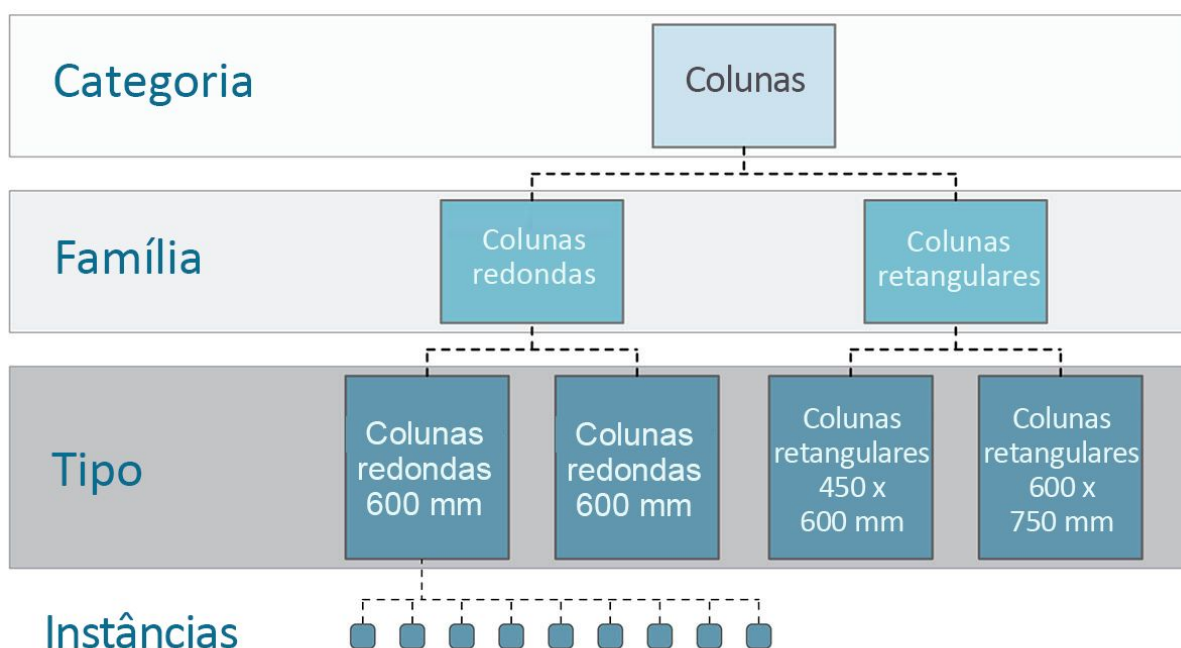
Os elementos de modelo representam a geometria real 3D de uma construção, sendo exibidos em vistas relevantes do modelo. Os elementos de dados ajudam a definir o contexto do projeto. Já os elementos específicos da vista são aqueles exibidos apenas nas vistas em que são colocados, usados para documentação do projeto.

Os elementos de modelo são classificados em hospedeiros, que são geralmente construídos no local do terreno da construção, e componentes do modelo. Por sua vez, elementos específicos da vista podem ser elementos de anotação, usados na documentação do modelo, e detalhes, que correspondem aos itens 2D que fornecem detalhes da construção em uma determinada vista, como explica a Autodesk (2023a).

A diferença entre elementos de modelo e elementos específicos de vista é exemplificada por Netto (2020): enquanto elementos de modelo, a exemplo de uma parede, são exibidos em todas as vistas (plantas, elevações, cortes, vistas 3D), os elementos específicos da vista, como cotas, símbolos e texto, aparecem apenas na vista em que foram criados.

Outra maneira de categorizar os elementos no Revit é através da hierarquia definida por categoria, família, tipo e instância. Essa hierarquia é ilustrada na Figura 4.7, tomando como exemplo a categoria Colunas.

Figura 4.7 – Hierarquia do Revit



Fonte: Autodesk (2019)

De acordo com Netto (2020), todos os objetos no Revit são pertencentes a uma família e cada família pertence a uma categoria. De fato, a Autodesk (2023a) define por instância cada elemento de um tipo de família inserido no modelo. Uma instância tem seu comportamento controlado por propriedades que podem ser de tipo e de instância. Quanto a esses dois grupos de propriedades, a Autodesk (2023a) define:

- Propriedades de tipo: O mesmo conjunto de propriedades de tipo é comum a todos os elementos em uma família, e cada propriedade tem o mesmo valor para todas as instâncias de um tipo de família particular.
- Propriedades de instância: Um conjunto comum de propriedades de instância também se aplica a todos os elementos que pertencem a um tipo de família em particular, mas os valores dessas propriedades podem variar de acordo com o local de um elemento em um edifício ou projeto.

4.5 Dynamo

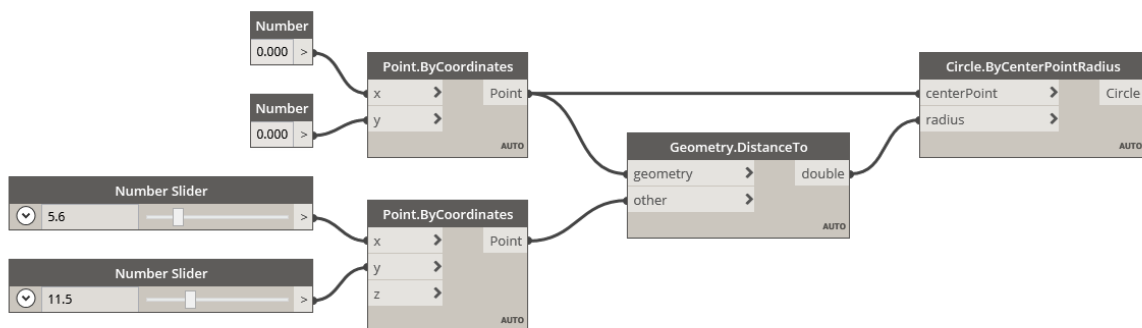
Esta seção se propõe a introduzir os principais conceitos acerca do Dynamo, a fim de possibilitar o entendimento do desenvolvimento da rotina, feito na Capítulo 5.

O Dynamo é uma plataforma que permite aos projetistas a exploração da programação visual e criação de ferramentas próprias. No contexto desse trabalho, essa plataforma é operada como um complemento do software Revit.

O Dynamo permite que se trabalhe em um processo de programação visual, no qual se conectam elementos que definem relações e sequências de ações que compõem algoritmos personalizados. Os algoritmos desenvolvidos podem ser usados para uma gama de aplicações, envolvendo desde o processamento de dados até a geração de geometria.

De maneira geral, entende-se o conceito de programação como o ato de formalizar o processamento de uma série de ações em um programa executável. A diferença entre a programação visual e a programação textual é que na primeira as instruções são definidas por meio de uma interface gráfica enquanto na segunda é feita a digitação do texto correspondente ao código. Essa diferença é ilustrada na Figura 4.8.

Figura 4.8 – Programação



(a) Visual

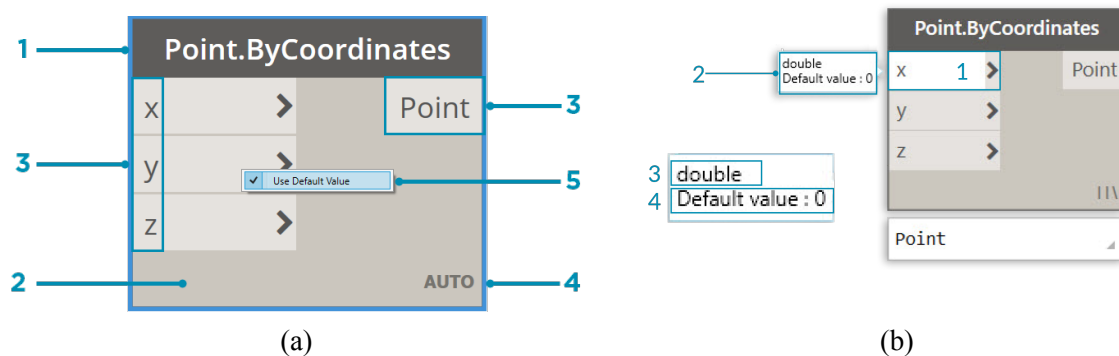
```
myPoint = Point.ByCoordinates(0.0,0.0,0.0);
x = 5.6;
y = 11.5;
attractorPoint = Point.ByCoordinates(x,y,0.0);
dist = myPoint.DistanceTo(attractorPoint);
myCircle = Circle.ByCenterPointRadius(myPoint,dist);
```

(b) Textual

Fonte: Autodesk (2019)

A programação no ambiente do Dynamo é feita através de objetos que se conectam para formar um programa visual: os nós. A anatomia de um nó é como mostrada na Figura 4.9.

Figura 4.9 – Anatomia de um nó



Fonte: Autodesk (2019)

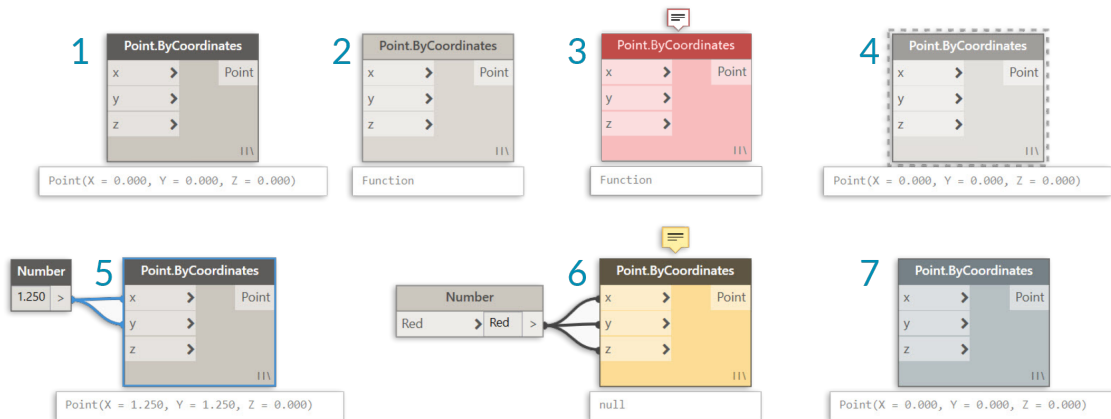
Na Figura 4.9a os números de 1 a 5 indicam respectivamente o nome do nó, seu corpo, suas portas (entradas e saída), a amarra especificada para as entradas e a opção de usar o valor padrão das entradas. Na Figura 4.9b são mostradas informações acerca de uma das entradas do nó: o tipo de valor esperado e o valor padrão.

O ato de programar no Dynamo ocorre através da ligação, por meio de fios, entre a saída de um nó e a entrada do que o sucede (Figura 4.8a). Os fios se encarregam de “transportar” os dados, que ao chegar no nó são processados e encaminhados em uma nova forma para a saída, onde encontram o próximo fio.

De acordo com a Autodesk (2019), essa característica de fios e nós estabelece o fluxo de dados em um programa visual no Dynamo: “uma vez que as portas de saída estão localizadas no lado direito dos nós e as portas de entrada estão no lado esquerdo, geralmente podemos dizer que o fluxo do programa se move da esquerda para a direita”.

A depender do resultado do processamento do nó, o Dynamo fornece uma indicação visual do seu status. Na Figura 4.10 os números de 1 a 7 indicam respectivamente os status ativo, inativo, estado de erro, congelado, selecionado, aviso e visualização desativada.

Figura 4.10 – Estados do nó



Fonte: Autodesk (2019)

Uma vez que se tem a execução de um programa, há processamento de dados nos nós que o define. O Dynamo lida com os tipos comuns de dados de linguagens de programação, como aqueles do tipo *integer*, *double*, *boolean* e *string*. Além desses, há operação com dados de geometria, categorizados conforme a Figura 4.11.

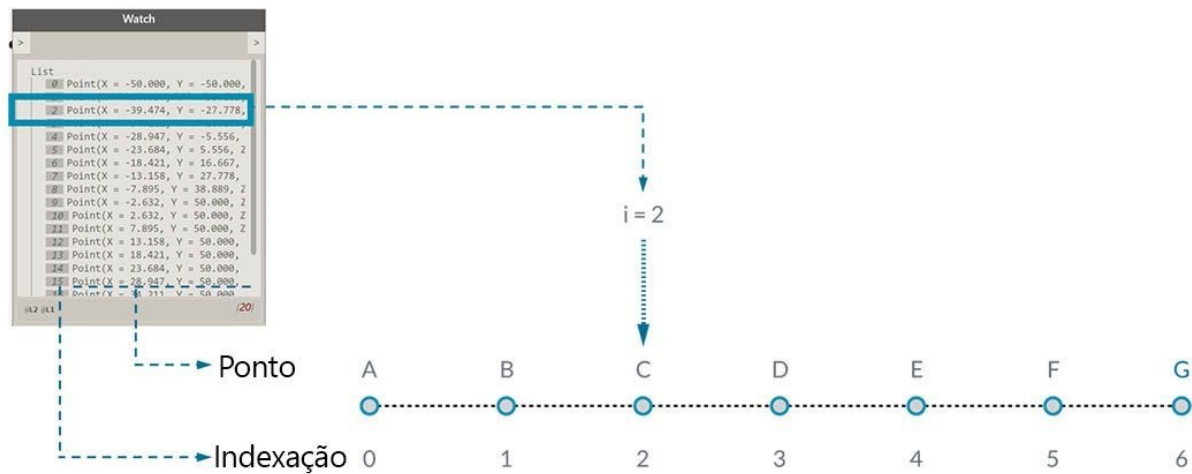
Figura 4.11 – Categorização da hierarquia geométrica no Dynamo

Hierarquia de tipos de dados							
Tipos abstratos			Tipos de geometria				
Define a localização + a orientação	Define a posição + o volume	Define as relações	Elementos do modelo				
Vetor	Caixa delimitadora	Topologia	Ponto	Curva	Superfície	Sólido	Malha
- Vetor - Plano - Sistema de coordenadas	Caixa delimitadora	- Vértice - Edge - Face	- Coordenada XYZ - Coordenada UV	- Linha - Polígono - Arco - Circle - Elipse - Curva NURBS - PolyCurve	- Superfície NURBS - PolySurface	- Cuboide - Esfera - Cone - Cilindro	Malha

Fonte: Autodesk (2019)

Conjuntos de dados no Dynamo são organizados segundo listas. Entende-se uma lista como um conjunto de itens ordenados segundo índices em ordem crescente, iniciando do índice zero, como ilustrado na Figura 4.12.

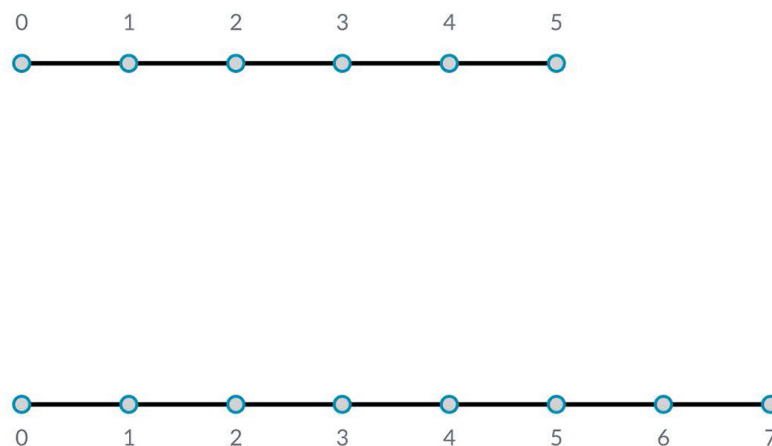
Figura 4.12 – Itens e lista



Fonte: Autodesk (2019)

Operações entre listas de tamanhos diferentes (Figura 4.13) são determinadas pela definição da amarra do nó em que há o processamento de dados.

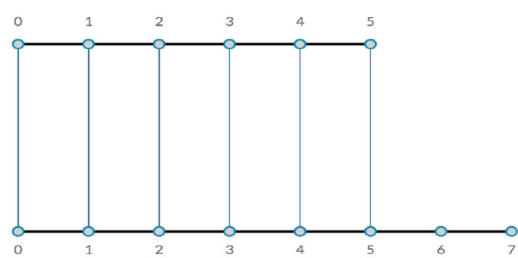
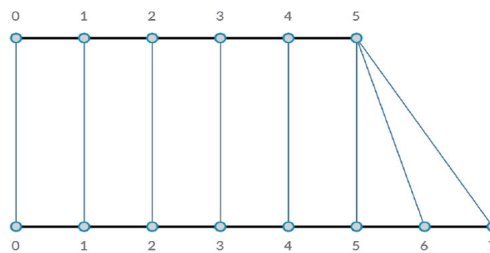
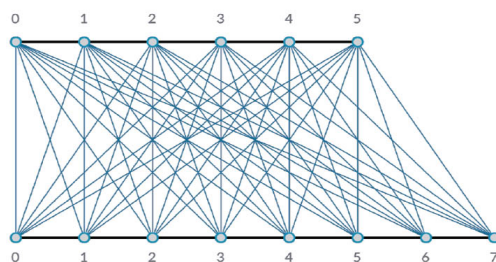
Figura 4.13 – Listas de tamanhos diferentes



Fonte: Autodesk (2019)

As possíveis opções de amarra são definidas conforme a Figura 4.14: *Shortest List*, *Longest List* e *Cross Product*.

Figura 4.14 – Opções de amarra de um nó

(a) *Shortest List*(b) *Longest List*(c) *Cross Product*

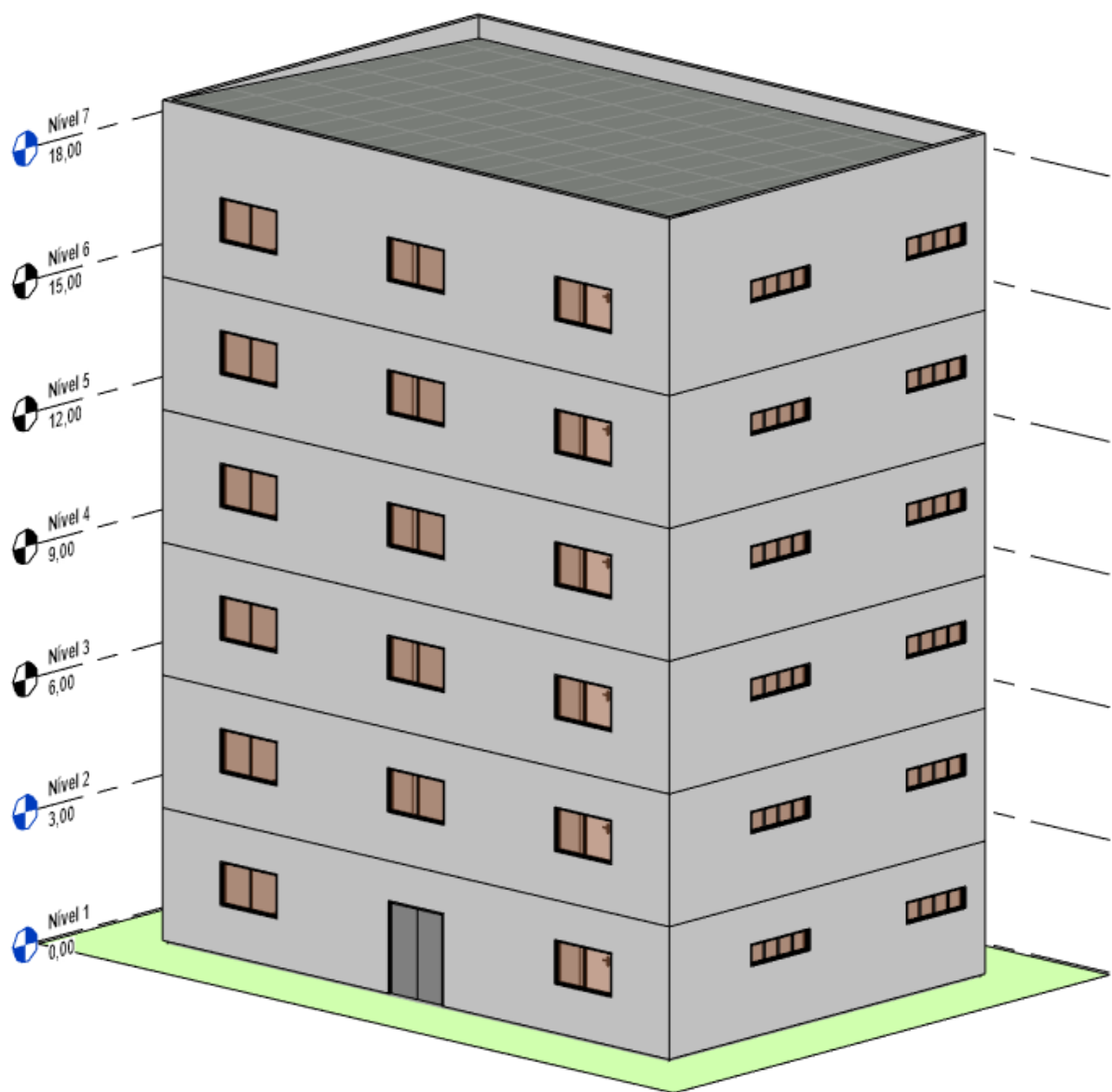
Fonte: Autodesk (2019)

5 DESENVOLVIMENTO DA ROTINA

O presente capítulo se propõe a explicar as partes da rotina desenvolvida. Serão fornecidas explicações acerca dos comandos utilizados, bem como dos resultados intermediários obtidos.

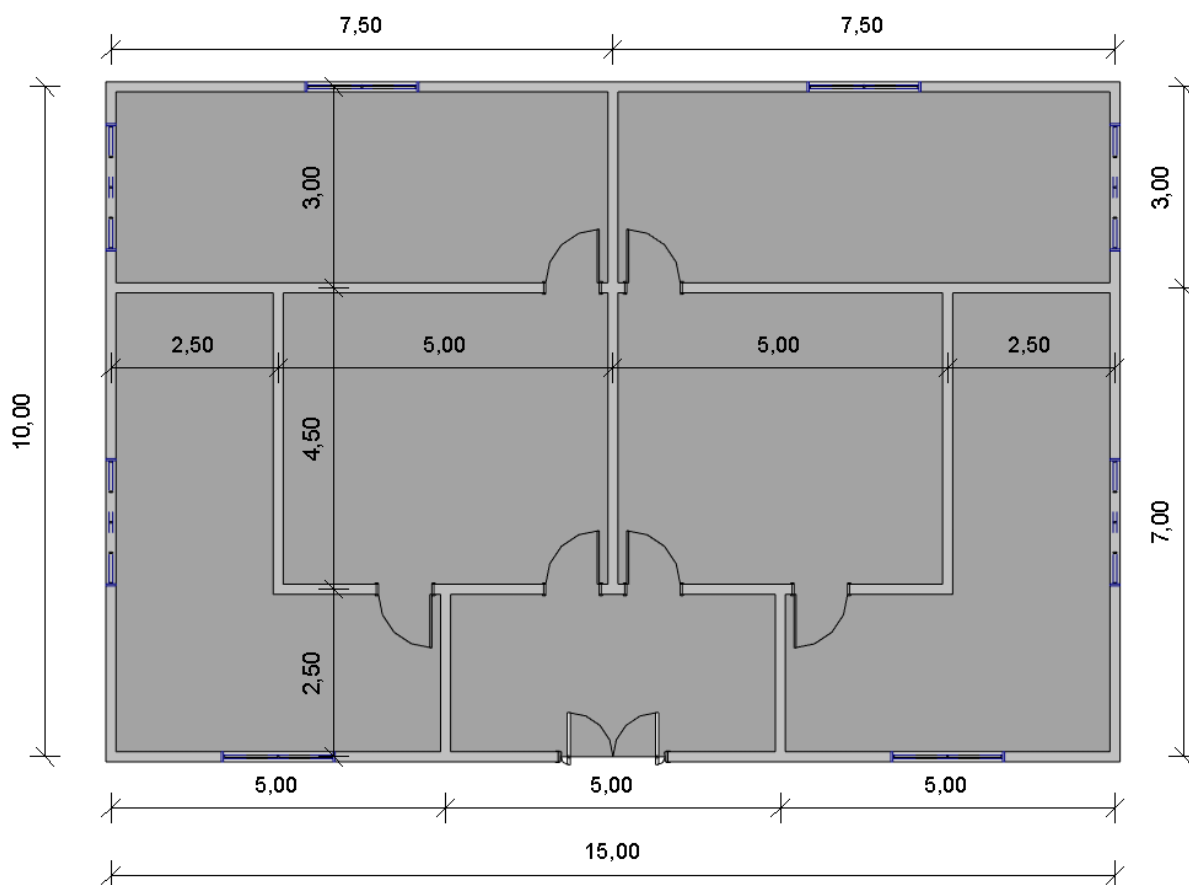
A fim de ilustrar tais resultados, será tomado o Edifício Exemplo, apresentado nas Figuras 5.1 e 5.2, que será passado como entrada da rotina.

Figura 5.1 – Vista isométrica do Edifício Exemplo



Fonte: Autor

Figura 5.2 – Vista em planta do Edifício Exemplo

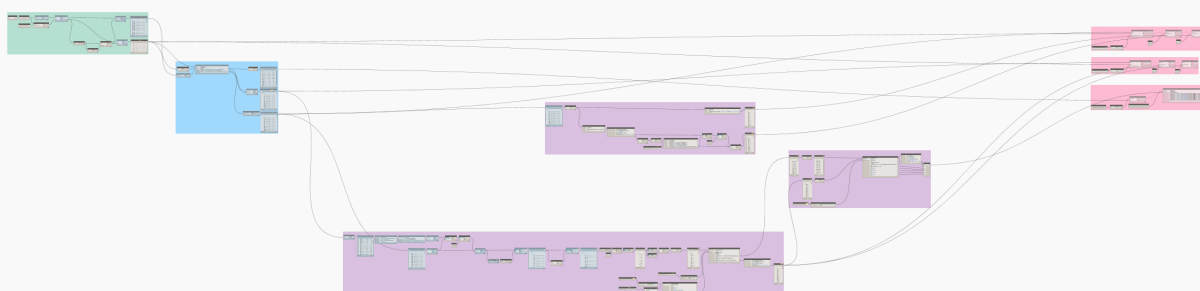


Fonte: Autor

5.1 Visão geral da rotina

A Figura 5.3 traz o ambiente do Dynamo, mostrando uma visão geral da rotina desenvolvida. É possível visualizar as diferentes partes da mesma. Cada parte é formada por grupos, sendo grupos da mesma parte identificados com retângulos da mesma cor.

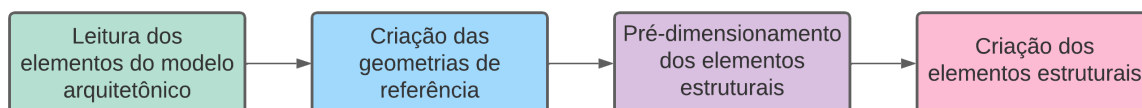
Figura 5.3 – Visão geral da rotina



Fonte: Autor

A Figura 5.4 apresenta um fluxograma que indica as partes da rotina, as cores que as identificam e a sequência de execução da mesma. Esse fluxograma foi retirado da Figura 2.2 com o intuito de retratar apenas as etapas do processamento de dados. As entradas e a saída da rotina serão apresentadas quando for conveniente.

Figura 5.4 – Fluxograma da rotina



Fonte: Autor

Da esquerda para direita, tem-se as partes Leitura dos elementos do modelo arquitetônico, Criação das geometrias de referência, Pré-dimensionamento dos elementos estruturais e Criação dos elementos estruturais, descritas nas seções 5.2, 5.3, 5.4 e 5.5 respectivamente.

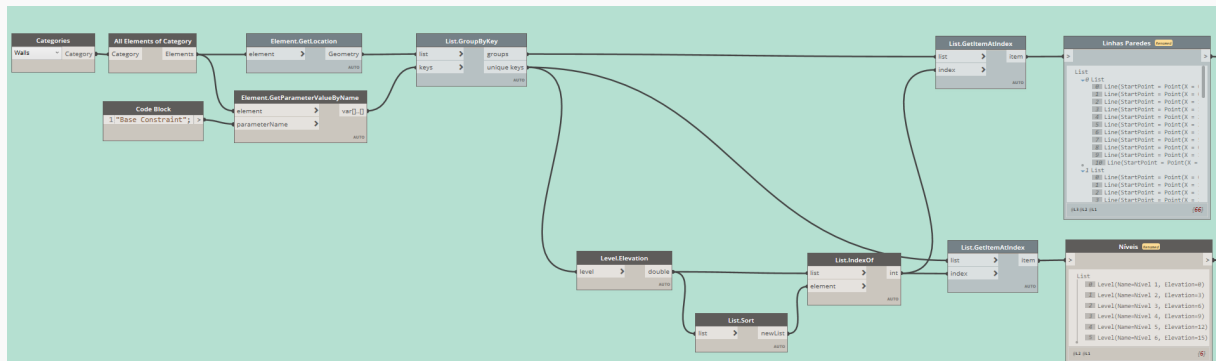
Em cada uma das seções mencionadas constarão subseções referentes a cada um dos grupos contidos na respectiva parte. Nas subseções serão apresentados os trechos de código utilizados acompanhados de comentários que explicam o funcionamento dos comandos e de que maneira os dados são manipulados na execução da rotina. Ainda, serão feitas explicações do algoritmo ao nível de cada grupo, cujo entendimento dispensa a leitura dos trechos de código.

5.2 Leitura dos elementos do modelo arquitetônico

5.2.1 Criação das linhas de localização das paredes

O grupo apresentado na Figura 5.5 tem por finalidade criar e agrupar as linhas de localização das paredes do modelo em listas, de acordo com o valor do parâmetro *Base Constraint* de cada parede. Linhas correspondentes a paredes com o mesmo valor do parâmetro são agrupadas na mesma lista, que são ordenadas de maneira crescente em função do valor da elevação do nível a que correspondem.

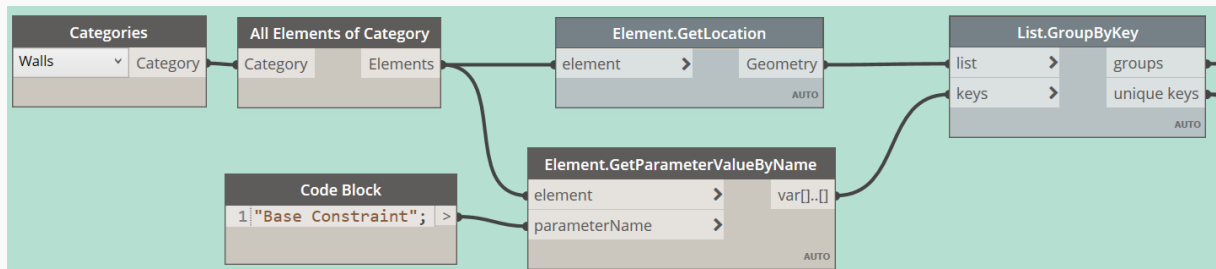
Figura 5.5 – Grupo para criação das linhas de localização das paredes



Fonte: Autor

Os trechos do grupo são apresentadas nas Figuras 5.6, 5.7 e 5.8, abaixo das quais são fornecidos comentários com as explicações do código exibido.

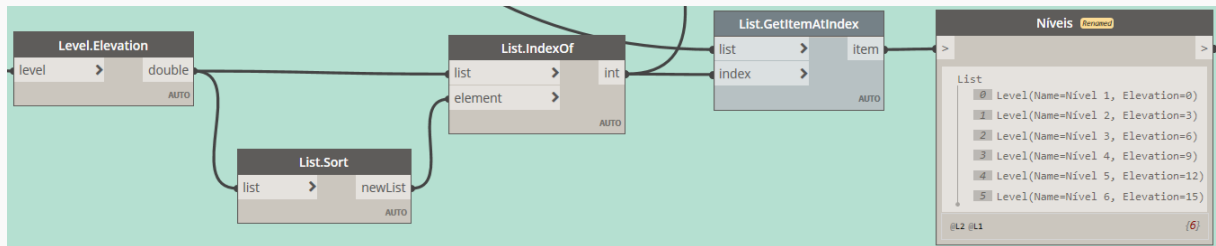
Figura 5.6 – Grupo para criação das linhas de localização das paredes (trecho 1)



Fonte: Autor

- As paredes do modelo são selecionadas através dos nós *Categories* e *All Elements of Category*.
- Da seleção de paredes são extraídos os valores do parâmetro *Base Constraint* através do nó *Element.GetParameterValueByName*. O parâmetro *Base Constraint* armazena elementos da categoria *Nível*.
- Da seleção de paredes também é extraída a linha de localização de cada parede através do nó *Element.GetLocation*. A linha de localização da parede é um elemento do tipo *Line* que coincide com o eixo da base da parede.
- O nó *List.GroupByKey* agrupa as linhas de localização em listas de acordo com o valor do nível da base da parede a que corresponde. Linhas correspondentes a valores de nível iguais são agrupadas nas mesmas listas. As listas de linhas de localização são retornadas na saída *groups*. Os valores dos níveis são retornados na saída *unique keys*.

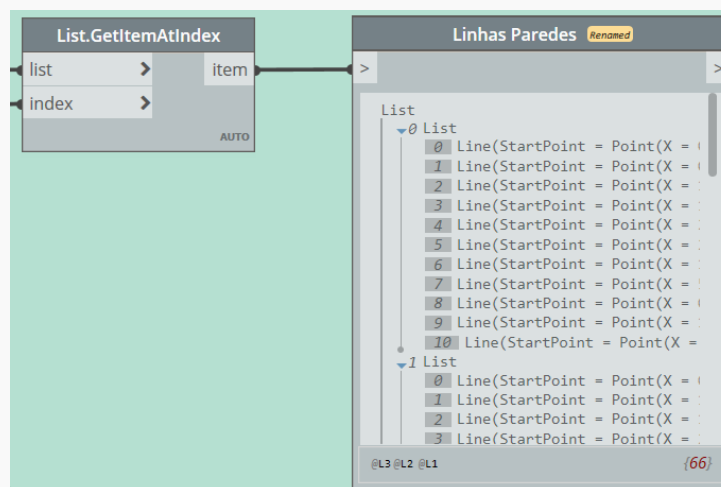
Figura 5.7 – Grupo para criação das linhas de localização das paredes (trecho 2)



Fonte: Autor

- O nó *Level.Elevation* recebe como entrada os níveis das bases das paredes, proveniente da saída *unique keys* do nó *List.GroupByKey* da Figura 5.6.
- O nó *Level.Elevation* retorna a elevação de cada nível. O nó *List.Sort* ordena esses valores em ordem crescente.
- O nó *List.IndexOf* retorna o índice de cada elemento da lista proveniente de *List.Sort* na lista gerada em *Level.Elevation*.
- O nó *List.GetItemAtIndex* recebe os índices provenientes de *List.IndexOf* e seleciona, a partir desses índices, os níveis das bases das paredes, gerados na saída *unique keys* do nó *List.GroupByKey* (Figura 5.6), aqui passado na entrada *list*.
- O nó *Watch*, renomeado para *Níveis*, exibe a saída do nó *List.GetItemAtIndex*: os níveis das bases das paredes do projeto em ordem crescente de elevação.

Figura 5.8 – Grupo para criação das linhas de localização das paredes (trecho 3)

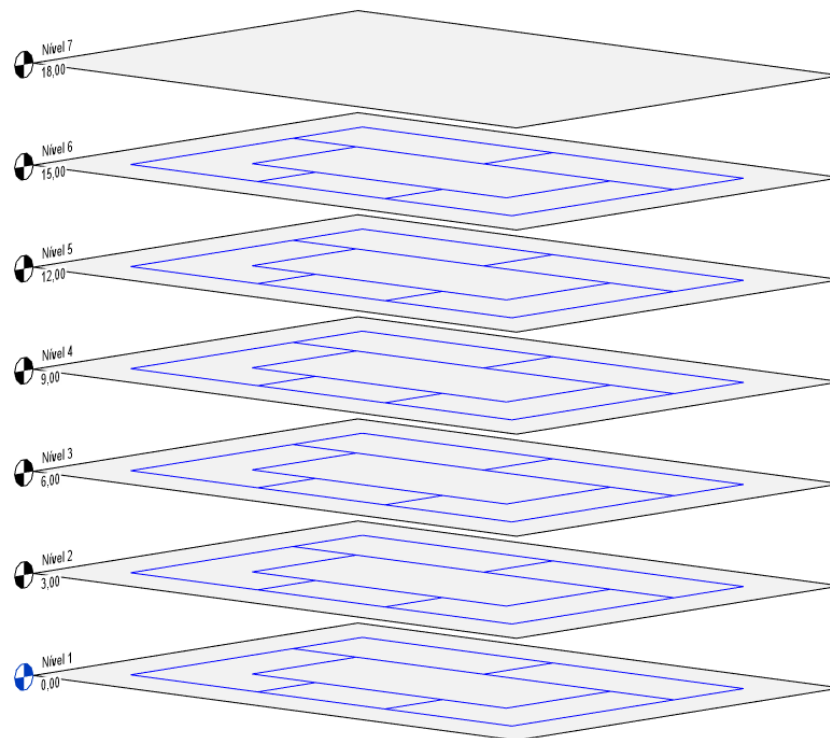


Fonte: Autor

- O nó *List.GetItemAtIndex* recebe na entrada *list* as listas de linhas de localização de parede do nó *List.GroupByKey* (Figura 5.6), geradas na saída *groups*. Na entrada *index* recebe os índices produzidos como saída do nó *List.IndexOf* exibido na Figura 5.7. Como saída, ele ordena as listas de linha de localização de parede em ordem crescente de nível.
- O nó *Watch*, renomeado para *Linhas Paredes*, exibe as listas de linha de localização de parede ordenadas.

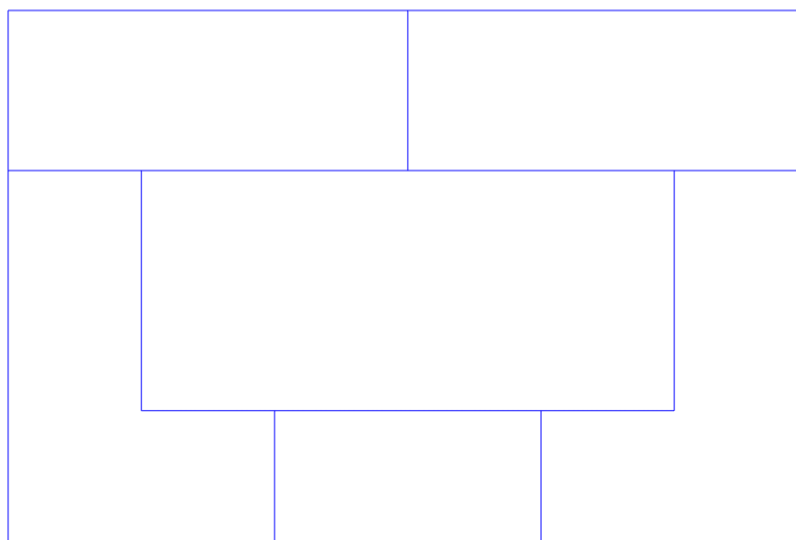
São geradas as listas de linhas de localização das paredes. Cada lista armazena linhas que se encontram no mesmo nível. As listas são ordenadas de maneira crescente em função do valor da elevação do nível a que correspondem. As Figuras 5.9 e 5.10 ilustram as linhas de localização obtidas até então, ao lado das quais são indicados os níveis a que correspondem.

Figura 5.9 – Vista isométrica das linhas de localização das paredes



Fonte: Autor

Figura 5.10 – Vista em planta das linhas de localização das paredes



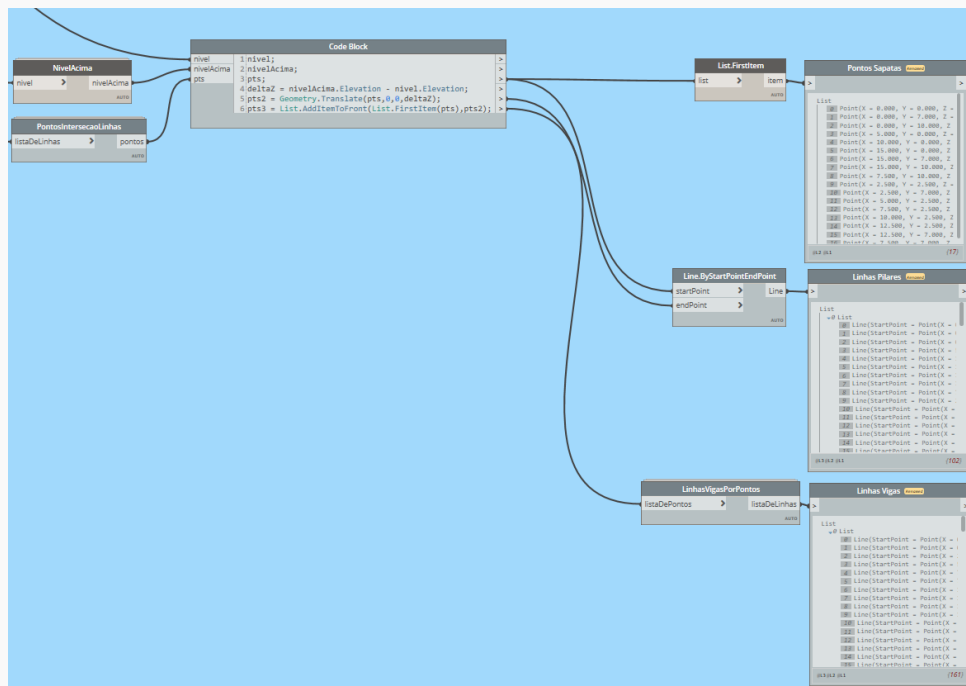
Fonte: Autor

5.3 Criação das geometrias de referência

5.3.1 Criação das geometrias de referência

O grupo apresentado na Figura 5.11 tem por finalidade gerar as geometrias de referência para criação dos elementos estruturais. Essas geometrias incluem linhas que definem a localização das vigas e dos trechos de pilar do modelo a ser concebido e pontos que definem o posicionamento das sapatas. A posição desses elementos é norteadada pelo proposto na seção 3.8, Lançamento dos elementos estruturais.

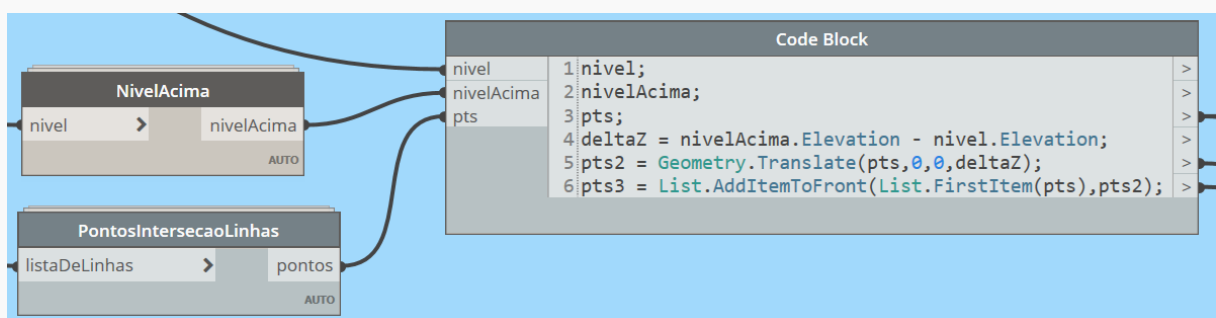
Figura 5.11 – Grupo para criação das geometrias de referência



Fonte: Autor

Os trechos do grupo são apresentadas nas Figuras 5.12, 5.16, 5.17 e 5.18, abaixo das quais são fornecidos comentários com as explicações do código exibido.

Figura 5.12 – Grupo para criação das geometrias de referência (trecho 1)



Fonte: Autor

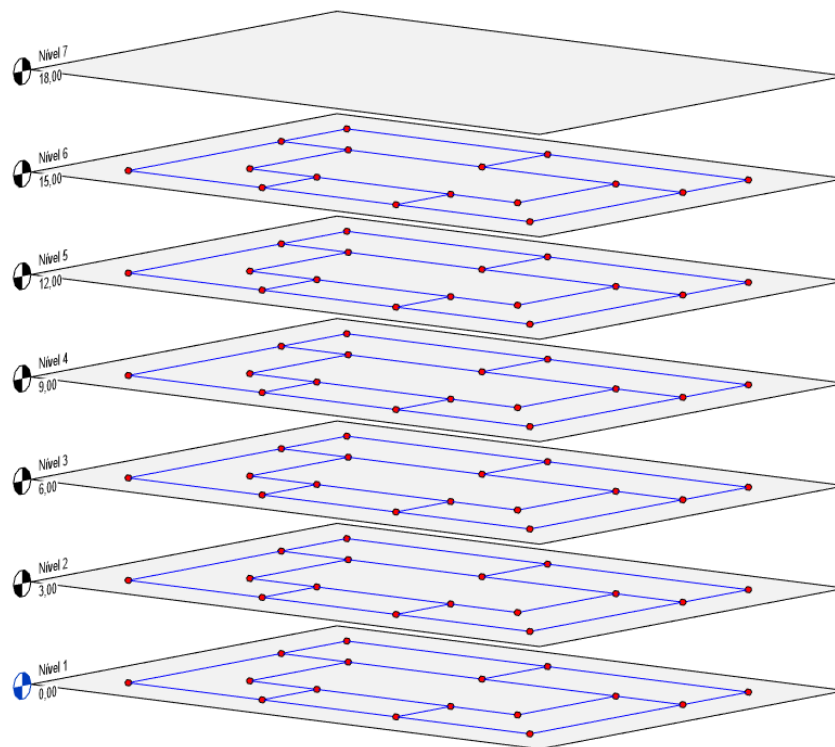
- O nó *NivelAcima* retorna o nível do projeto localizado imediatamente acima daquele passado como entrada. No caso, é passada como entrada a lista dos níveis das bases das paredes, saída do nó *Níveis* exibido na Figura 5.7. O nó *NivelAcima* é um nó personalizado, cujo desenvolvimento é abordado no Apêndice A.
- O nó *PontosIntersecaoLinhas* retorna o conjunto de pontos resultante da interseção das li-

nhas passadas na entrada. São passadas as listas de linhas de localização das paredes, saída do nó *Linhas Paredes* da Figura 5.8. O nó *PontosIntersecaoLinhas* é um nó personalizado, cujo desenvolvimento é abordado no Apêndice B.

- Definem-se três variáveis no *Code Block*: `nivel`, `nivelAcima` e `pts`. A variável `nivel` recebe os níveis gerados no nó *Níveis* da Figura 5.7. A variável `nivelAcima` recebe a saída do nó *NivelAcima*. A variável `pts` recebe a saída do nó *PontosIntersecaoLinhas*.
- Define-se a variável `deltaZ`, que armazena a diferença de elevação entre dois níveis consecutivos.
- A variável `pts2` armazena as listas de pontos resultante da translação das listas de pontos armazenada em `pts`, sendo cada lista transladada na direção z pelo valor `deltaZ` a que corresponde.
- Por fim, é criada uma lista com a primeira lista da variável `pts` e as listas da variável `pts2`. Essa lista de listas é armazenada na variável `pts3`.

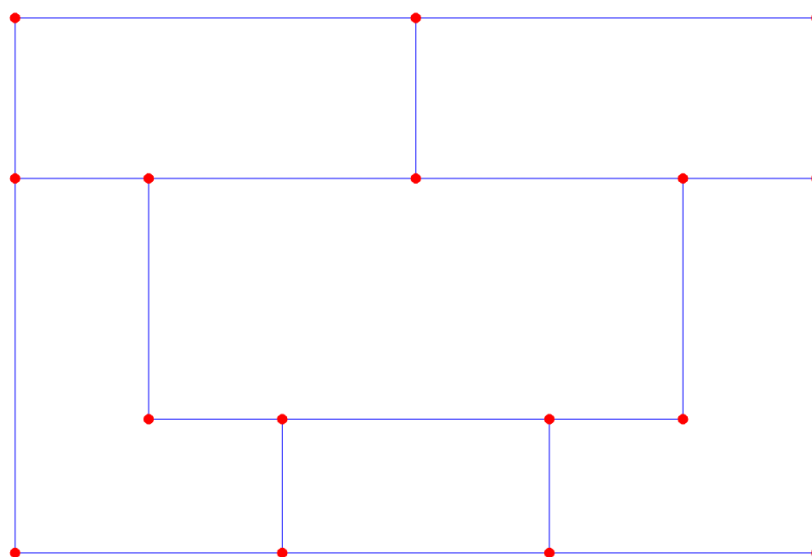
Das linhas de localização exibidas nas Figuras 5.9 e 5.10 são determinados os pontos de interseção entre elas, como exibido nas Figuras 5.13 e 5.14.

Figura 5.13 – Vista isométrica dos pontos de interseção entre linhas de localização de paredes



Fonte: Autor

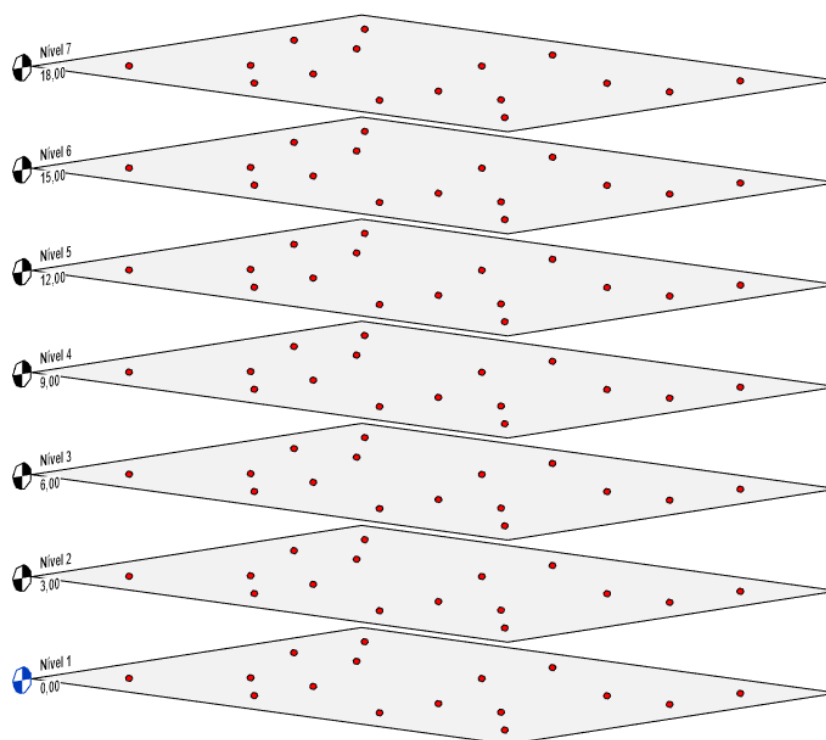
Figura 5.14 – Vista em planta dos pontos de interseção entre linhas de localização de paredes



Fonte: Autor

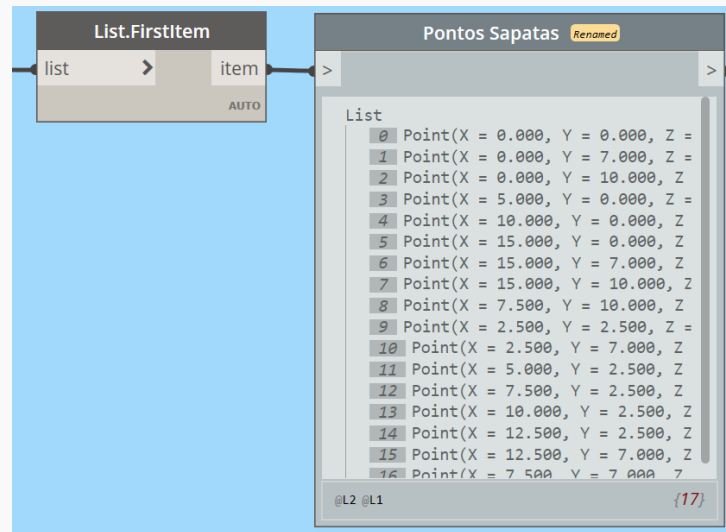
Os pontos obtidos são translados para o pavimento imediatamente superior. Ao conjunto de pontos translados, adicionam-se aqueles referentes ao pavimento mais inferior, gerando a lista final de pontos exibida na Figura 5.15.

Figura 5.15 – Vista isométrica da lista final de pontos



Fonte: Autor

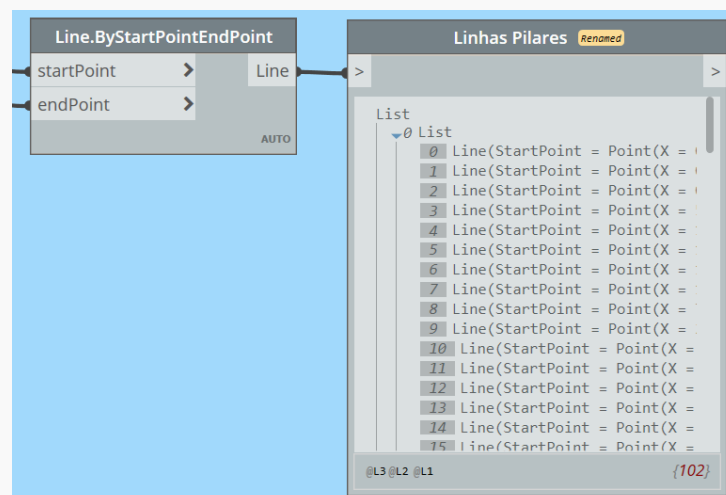
Figura 5.16 – Grupo para criação das geometrias de referência (trecho 2)



Fonte: Autor

- O nó *List.FirstItem* recebe como entrada a variável `pts`, definida na Figura 5.12, retornando a lista de pontos correspondentes ao primeiro nível.
- O nó *Watch*, renomeado para *Pontos Sapatas*, exibe a lista de pontos retornada. Esses pontos serão tomados posteriormente para definir a posição das sapatas.

Figura 5.17 – Grupo para criação das geometrias de referência (trecho 3)



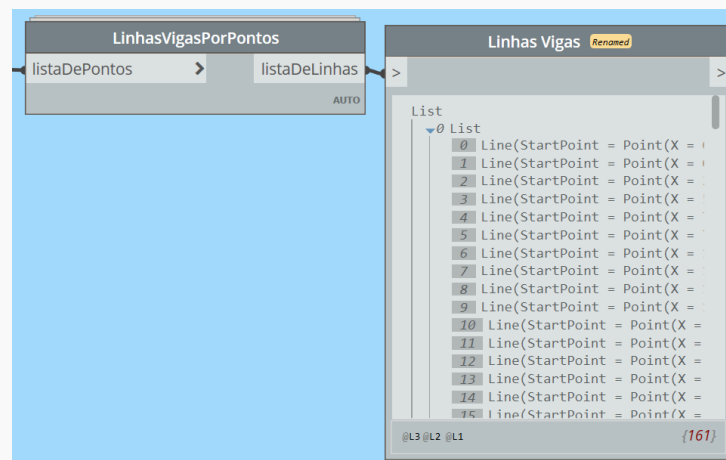
Fonte: Autor

- Através do nó *Line.ByStartPointEndPoint* são criadas linhas que unem os pontos armazenados em `pts` aos armazenados em `pts2`, definidos na Figura 5.12. Essas linhas

definem a localização dos trechos de pilar.

- O nó *Watch* renomeado para *Linhas Pilares* exibe o resultado do nó *Line.ByStartPointEndPoint*. São criadas listas correspondentes a cada um dos pavimentos, exceto o último. Cada lista armazena as linhas de localização dos trechos de pilar, que tem início no nível a que correspondem e fim no nível localizado imediatamente abaixo.

Figura 5.18 – Grupo para criação das geometrias de referência (trecho 4)



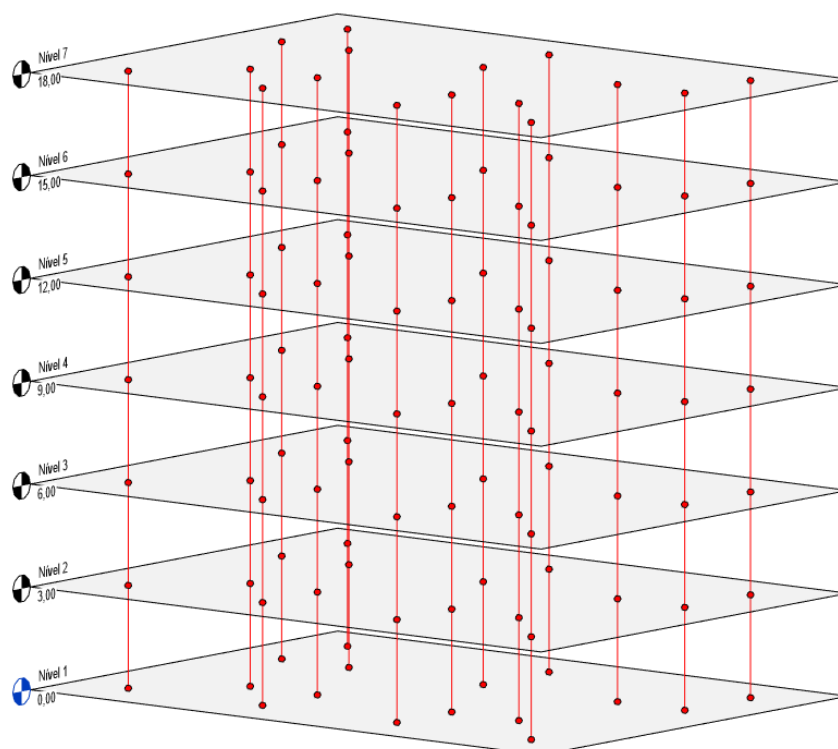
Fonte: Autor

- As listas de pontos armazenadas em `pts3`, definida na Figura 5.12, são passadas como entrada do nó *LinhasVigasPorPontos*. A partir de uma lista de pontos coplanares, esse nó define uma lista de linhas nas direções x e y unindo dois pontos adjacentes. O nó *LinhasVigasPorPontos* é um nó personalizado, cujo desenvolvimento é abordado no Apêndice C.
- O nó *Watch* renomeado para *Linhas Vigas* exibe o resultado do nó *LinhasVigasPorPontos*. Vale observar que a saída desse nó é um conjunto de listas, uma para cada nível do projeto, que armazena as linhas de localização das vigas.

A partir da lista de pontos gerada (Figura 5.15) são criadas as linhas dos pilares, traçando para cada um dos pontos, exceto aqueles no pavimento mais inferior, um segmento vertical até o pavimento situado imediatamente abaixo, conforme a Figura 5.19.

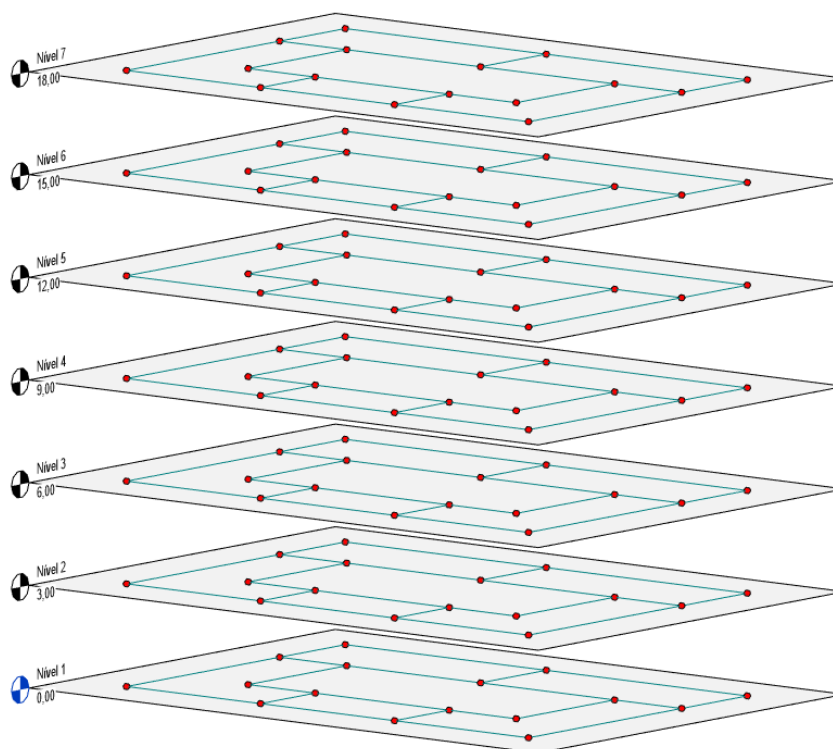
Ainda, são criadas as linhas de localização das vigas. Para cada pavimento, são traçados segmentos nas direções x e y que unem dois pontos consecutivos. O resultado pode ser observado na Figura 5.20.

Figura 5.19 – Vista isométrica das linhas de localização dos pilares



Fonte: Autor

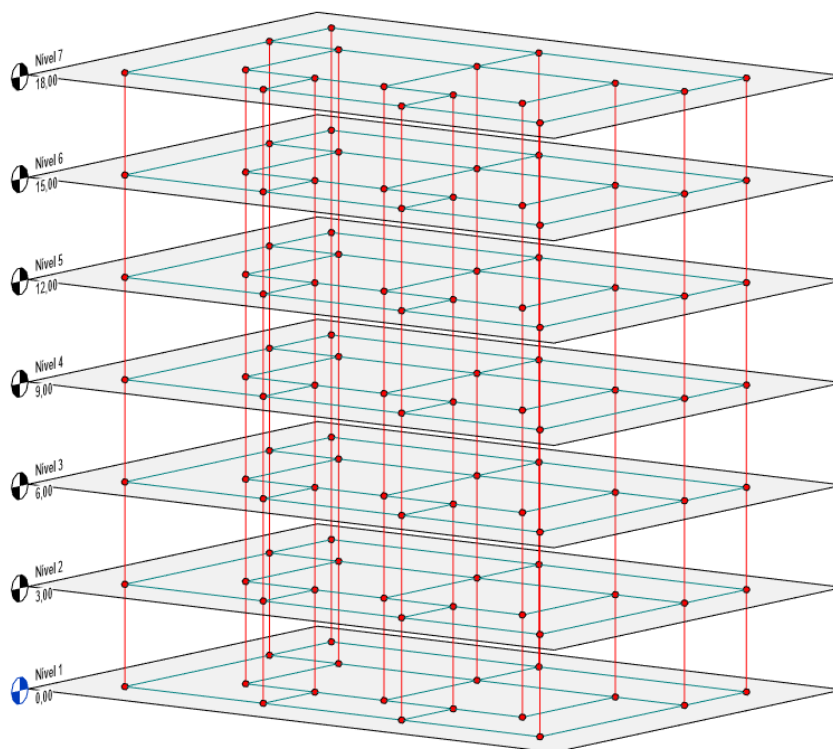
Figura 5.20 – Vista isométrica das linhas de localização das vigas



Fonte: Autor

A Figura 5.21 ilustra a saída da parte Criação das geometrias de referência. É possível observar as linhas de localização das vigas e dos trechos de pilares, bem como os pontos que as determinaram. Os pontos situados ao nível do primeiro pavimento serão usados na definição da posição dos elementos de fundação.

Figura 5.21 – Vista isométrica das linhas de localização de vigas e pilares



Fonte: Autor

5.4 Pré-dimensionamento dos elementos estruturais

5.4.1 Pré-dimensionamento de pilar

O grupo apresentado na Figura 5.22 tem por finalidade definir as dimensões das seções dos trechos dos pilares. No algoritmo que será descrito, serão tomadas as linhas de localização dos trechos dos pilares. Para cada linha, será determinada uma área de influência correspondente, a partir da qual será possível estimar o carregamento que solicita o trecho e definir sua seção transversal. O pré-dimensionamento realizado é norteado pela teoria exposta na subseção 3.7.3, Pilares.

Figura 5.22 – Grupo para pré-dimensionamento dos pilares

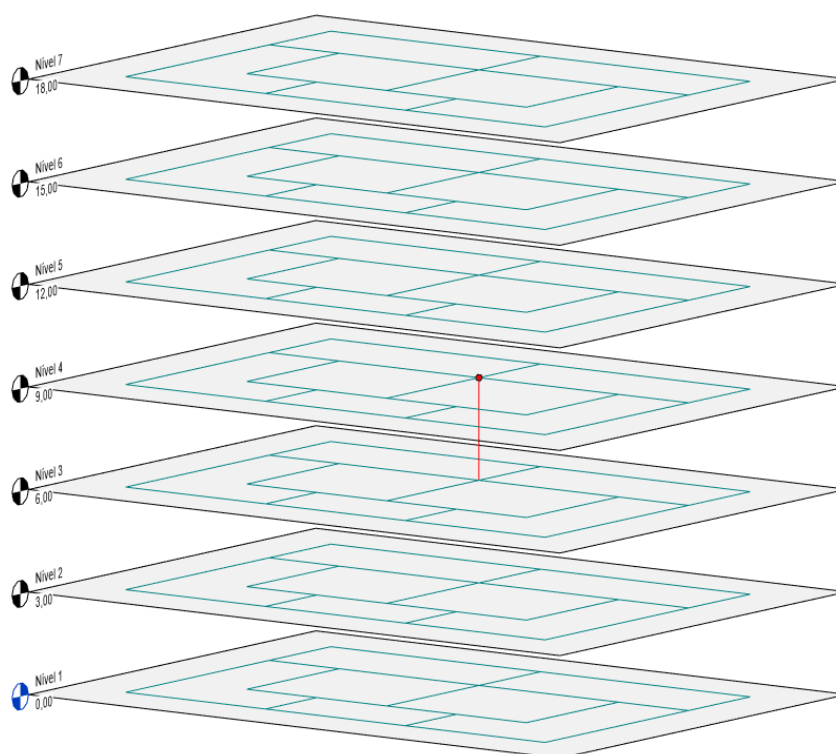


Fonte: Autor

Os trechos do grupo são apresentadas nas Figuras 5.24, 5.25, 5.27, 5.28, 5.30, 5.31, 5.32, 5.34, 5.35, 5.36 e 5.37, abaixo das quais são fornecidos comentários com as explicações do código exibido.

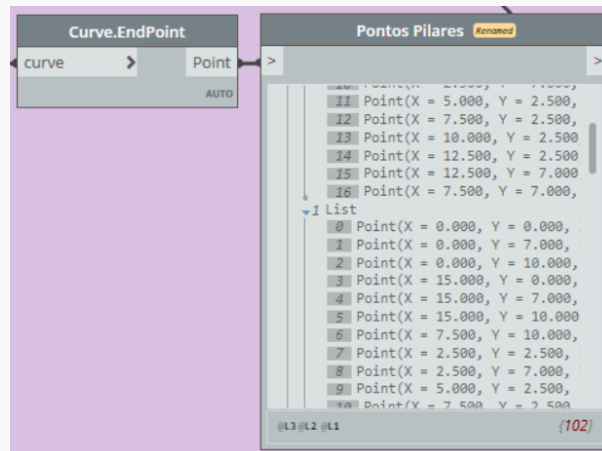
Na Figura 5.23 isola-se uma linha de trecho de pilar a fim de ilustrar o algoritmo que origina o código em questão, exemplificando como é o processo de pré-dimensionamento realizado.

Figura 5.23 – Ilustração do algoritmo de pré-dimensionamento de pilar (1)



Fonte: Autor

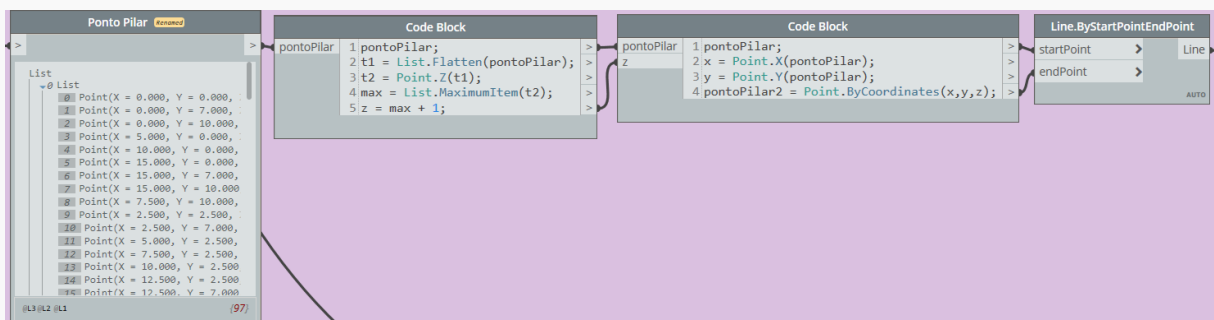
Figura 5.24 – Grupo para pré-dimensionamento dos pilares (trecho 1)



Fonte: Autor

- O nó *Curve.EndPoint* recebe a saída do nó *Linhas Pilares* exibido na Figura 5.17. Esse nó retorna a extremidade superior das linhas de localização dos trechos de pilar.
- O nó *Watch*, renomeado para *Pontos Pilares*, apresenta as listas de pontos correspondentes à extremidade superior de cada trecho de pilar.

Figura 5.25 – Grupo para pré-dimensionamento dos pilares (trecho 2)



Fonte: Autor

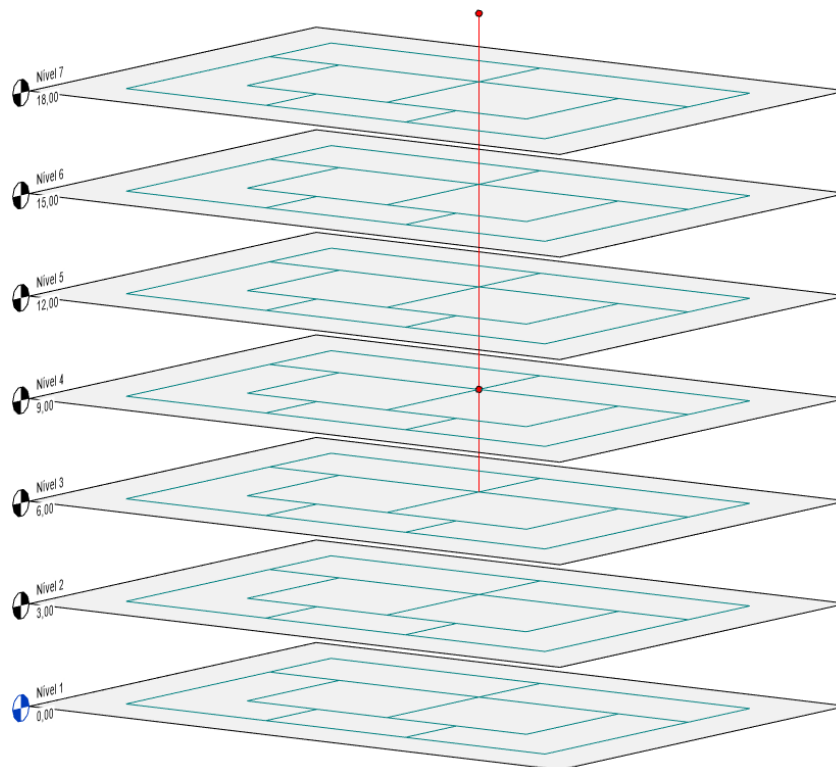
- O primeiro *Code Block* armazena o valor de saída do nó *Ponto Pilar* na variável `pontoPilar`. O conteúdo da variável `pontoPilar` é aplainado e armazenado na variável `t1`. Tomam-se os valores da coordenada `z` dos pontos dessa lista aplainada, armazenando-os em `t2`. Em seguida, o maior valor das coordenadas `z` é armazenado na variável `max`. Por fim, define-se a variável `z`, que armazena o valor de `max` adicionado de uma unidade.
- No segundo *Code Block* define-se a variável `pontoPilar`, que recebe o conteúdo ar-

mazenado na variável de mesmo nome definida no *Code Block* anterior. Tomam-se os valores das coordenadas x e y dos pontos armazenados em `pontoPilar`, armazenando-as nas variáveis `x` e `y` respectivamente. Ao final, cria-se um ponto com coordenadas x e y dadas pelas variáveis `x` e `y` e coordenada z obtida da variável `z` do *Code Block* anterior.

- O nó *Line.ByStartPointEndPoint* recebe na entrada *startPoint* os pontos armazenados na variável `pontoPilar` do segundo *Code Block* e na entrada *endPoint* a variável `pontoPilar2`.

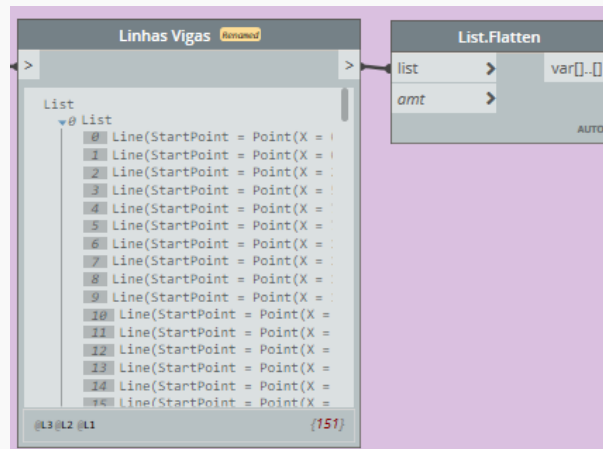
É criado, para cada ponto superior dos diferentes trechos de pilar, um ponto projetado acima de todos eles. Em seguida, é determinada uma linha que parte do ponto superior do trecho de pilar ao ponto projetado. Na Figura 5.26 é traçada essa linha.

Figura 5.26 – Ilustração do algoritmo de pré-dimensionamento de pilar (2)



Fonte: Autor

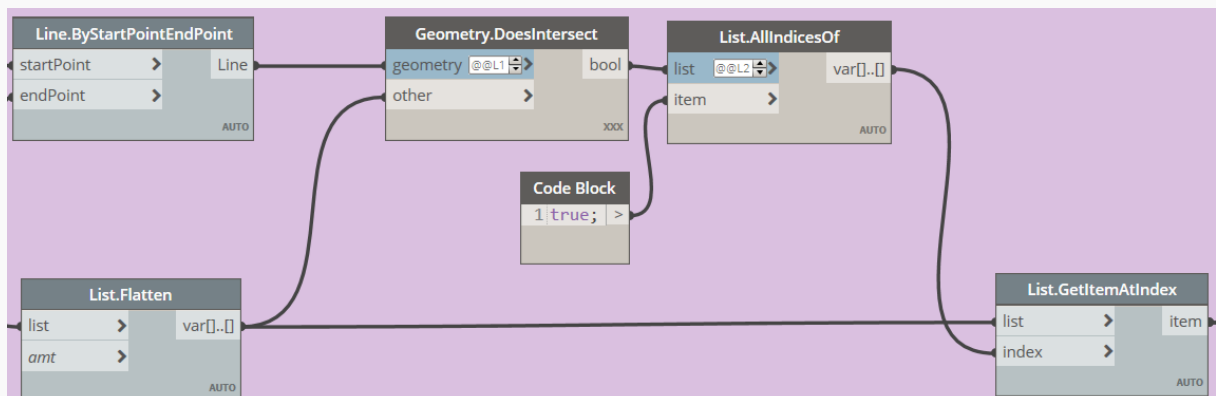
Figura 5.27 – Grupo para pré-dimensionamento dos pilares (trecho 3)



Fonte: Autor

- O nó *Watch* renomeado para *Linhas Vigas* recebe as linhas de localização das vigas, saída do nó *Linhas Vigas* exibido na Figura 5.18.
- Essa lista de listas de linhas é aplanada através do nó *List.Flatten*.

Figura 5.28 – Grupo para pré-dimensionamento dos pilares (trecho 4)



Fonte: Autor

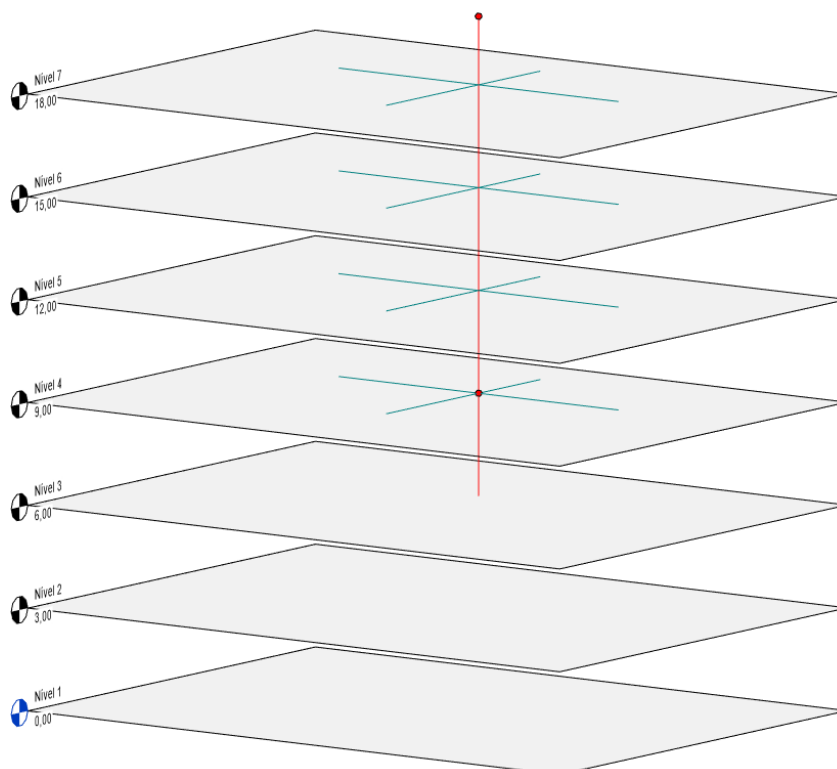
- O nó *Line.ByStartPointEndPoint* produz, para cada linha de localização de trecho de pilar, uma linha que parte do seu topo, como ilustrado na Figura 5.26.
- O nó *List.Flatten* reúne todas as linhas de localização de vigas.
- Através do nó *Geometry.DoesIntesect* verifica-se a interseção de cada linha correspondente a um trecho de pilar com todas as linhas de localização das vigas. Para cada trecho

de pilar é gerada uma lista com valores *true* e *false*, em função de haver ou não interseção da linha correspondente ao trecho com cada uma das linhas das vigas.

- O nó *List.AllIndicesOf* retorna para cada lista correspondente aos trechos de pilar, uma lista que armazena os índices dos valores *true*. Esses índices são os das linhas das vigas no nó *List.Flatten* que efetuam interseção com a linha correspondente ao trecho.
- Por fim, por meio do nó *List.GetItemAtIndex*, são tomados, para cada trecho de pilar, as linhas de localização das vigas que efetuam interseção com a linha que corresponde ao trecho.

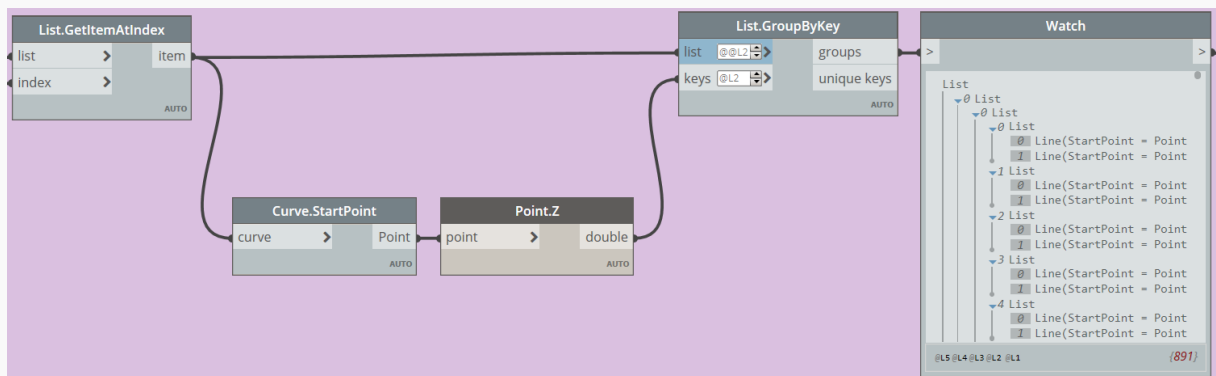
Verifica-se, para cada trecho de pilar, quais são as linhas de localização de vigas que efetuam interseção com a linha que parte do topo desse pilar. Esse procedimento determina uma lista de linhas de localização de vigas que corresponde a cada trecho de pilar, como ilustrado na Figura 5.33.

Figura 5.29 – Ilustração do algoritmo de pré-dimensionamento de pilar (3)



Fonte: Autor

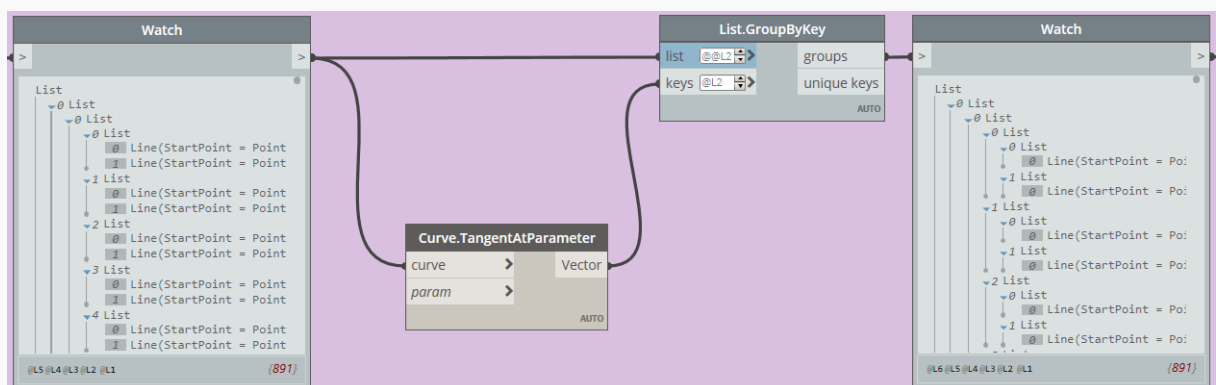
Figura 5.30 – Grupo para pré-dimensionamento dos pilares (trecho 5)



Fonte: Autor

- O nó *List.GetItemAtIndex* reúne as listas de linhas de localização de vigas que correspondem a cada trecho de pilar.
- De cada lista de linhas de localização são tomadas as coordenadas z dos pontos iniciais das linhas, por meio dos nós *Curve.StartPoint* e *Point.Z*.
- O nó *List.GroupByKey* vai criar sublistas em cada lista de linhas de localização de vigas geradas do nó *List.GetItemAtIndex*. A criação dessas sublistas vai agrupar em listas diferentes linhas que possuam o mesmo valor de coordenada z.
- O nó *Watch* permite a visualização das listas aninhadas geradas. O último nível de listas agrupa linhas de mesma coordenada z.

Figura 5.31 – Grupo para pré-dimensionamento dos pilares (trecho 6)



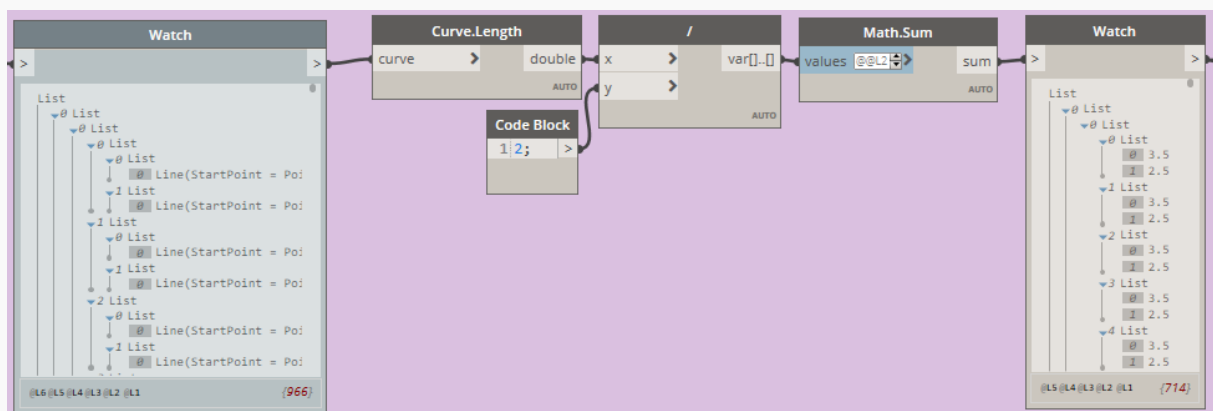
Fonte: Autor

- Das listas aninhadas exibidas no nó *Watch*, nas quais o último nível agrupa linhas de mesma coordenada z, tomam-se os vetores tangentes às linhas armazenadas, por meio do

nó *Curve.TangentAtParameter*. Esse nó possui valor padrão 0 para a entrada *param*, o que leva ao retorno do vetor tangente à curva na sua extremidade inicial.

- Esses vetores são usados para a criação de novas sublistas, que agrupam as linhas de mesma direção em listas separadas, por meio do nó *List.GroupByKey*.
- O nó *Watch* à direita mostra o conjunto de listas aninhadas. O último nível de listas agrupa linhas de mesma direção (x ou y).

Figura 5.32 – Grupo para pré-dimensionamento dos pilares (trecho 7)

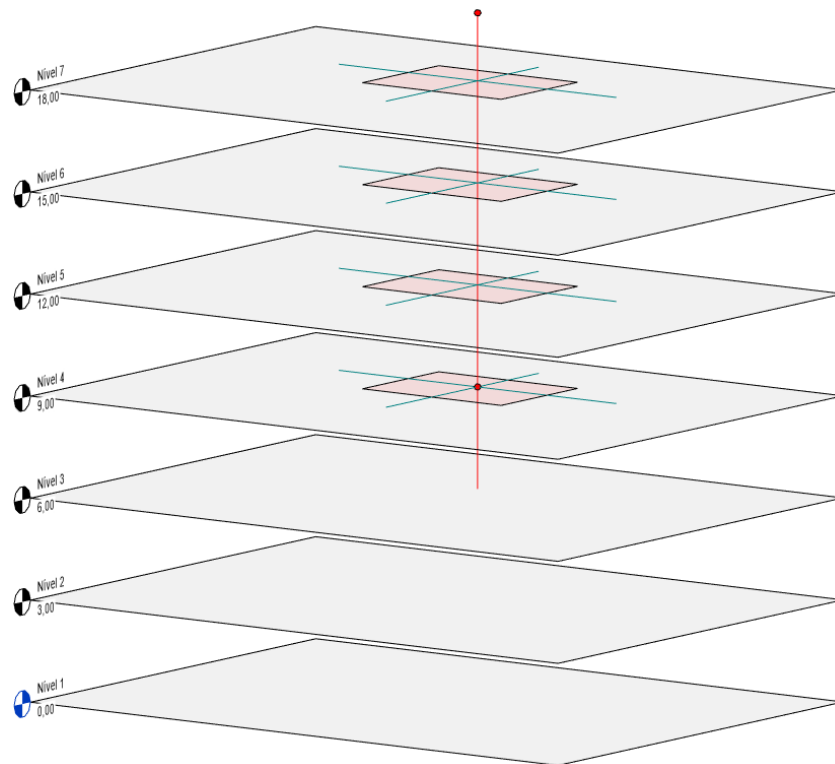


Fonte: Autor

- Das listas aninhadas exibidas no nó *Watch* à esquerda, nas quais o último nível agrupa linhas de mesma direção, toma-se o comprimento das linhas, através do nó *Curve.Length*.
- O valor do comprimento de cada linha é dividido por dois e em seguida são somados os valores no último nível de listas, no qual são agrupadas as linhas que possuem mesma direção. A soma é feita pelo nó *Math.Sum* e o resultado é exibido no nó *Watch* à direita.

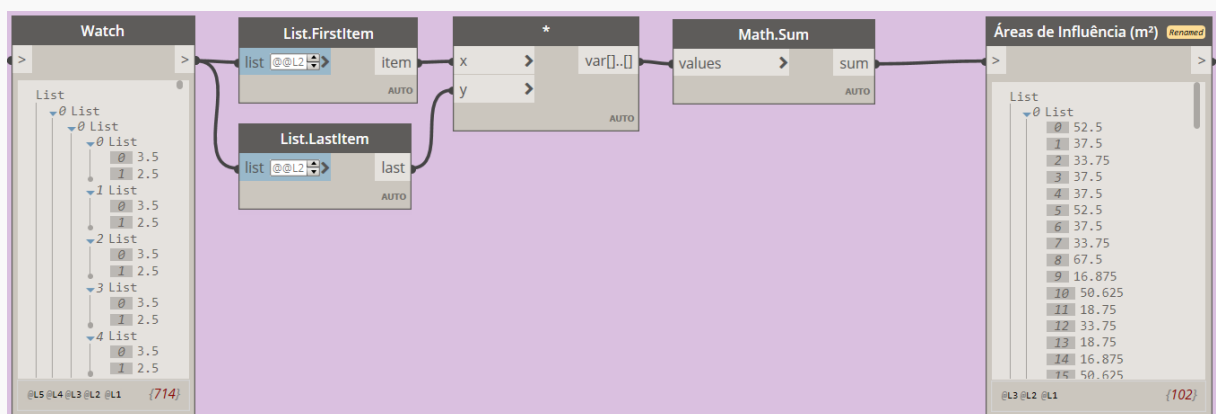
Das linhas de localização das vigas correspondentes a cada trecho de pilar, determinam-se as áreas de influência correspondentes. É determinada uma área de influência para cada pavimento em que linhas de vigas são identificadas. Essa determinação envolve localizar o retângulo cujos lados são paralelos às linhas de vigas e cujos lados passam pelo ponto médio das mesmas. O procedimento é ilustrado na Figura 5.33.

Figura 5.33 – Ilustração do algoritmo de pré-dimensionamento de pilar (4)



Fonte: Autor

Figura 5.34 – Grupo para pré-dimensionamento dos pilares (trecho 8)

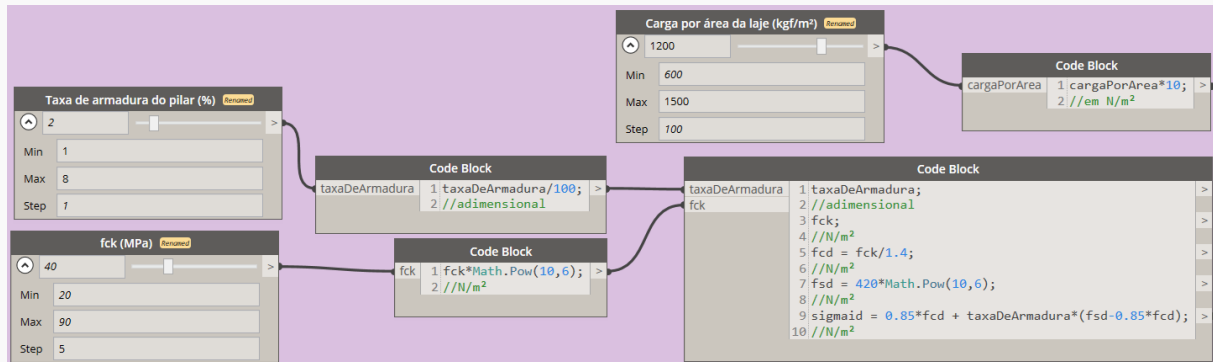


Fonte: Autor

- O nó *Watch* à esquerda armazena em seu último nível as dimensões das áreas de influência correspondentes a um trecho de pilar.
- Dos nós *List.FirstItem*, *List.LastItem* e *** é feito o produto entre as duas dimensões que definem cada área de influência, resultando no valor da área.
- O nó *Math.Sum* faz a soma das diferentes áreas de influência que correspondem a um

mesmo trecho de pilar. O resultado dessa soma é exibido no nó *Watch* renomeado para *Áreas de Influência (m²)*.

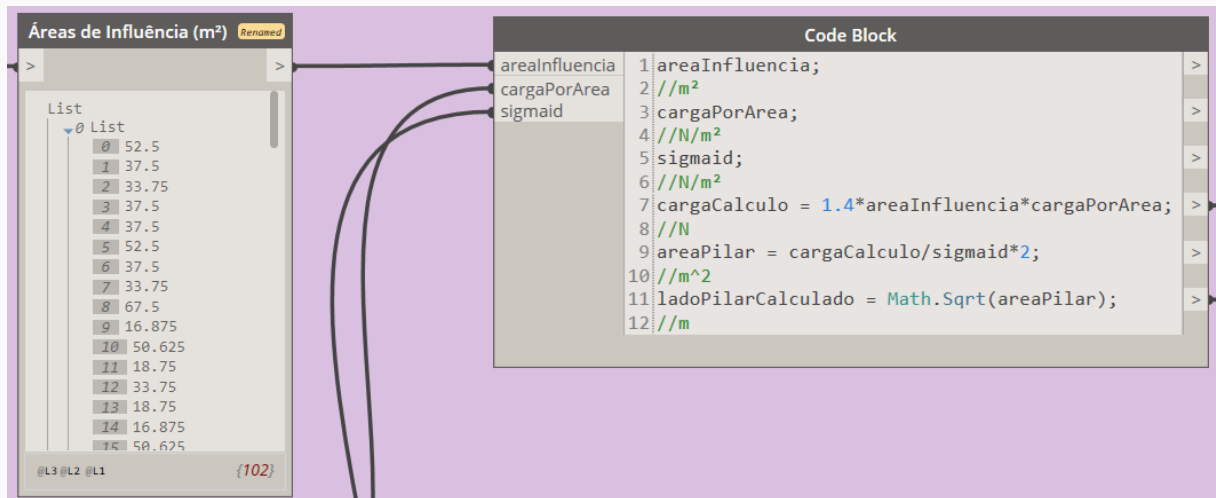
Figura 5.35 – Grupo para pré-dimensionamento dos pilares (trecho 9)



Fonte: Autor

- Os nós *Taxa de armadura do pilar (%)*, *fck (MPa)* e *Carga por área da laje (kgf/m²)* são nós *Number Slider* renomeados. Esses nós permitem a definição de um valor a partir de um controle deslizante. Os valores mínimo e máximo e o intervalo entre valores consecutivos são definidos para cada nó.
- O valor do nó *Taxa de armadura do pilar (%)* é armazenado na variável `taxaDeArmadura` e dividido por 100 para se ter um valor adimensional.
- O valor do nó *fck (MPa)* é armazenado na variável `fck` e multiplicado por 10^6 para se ter um valor em N/m^2 .
- No *Code Block* exibido no canto inferior direito as variáveis `taxaDeArmadura` e `fck` são novamente definidas e recebem os valores já corrigidos para adequação das unidades.
- Nesse mesmo *Code Block* são definidas as variáveis `fcd` e `sigmaid`. O cálculo desses valores se dá por meio da Equação 3.28.
- O valor do nó *Carga por área da laje (kgf/m²)* é armazenado na variável `cargaPorArea` e multiplicado por 10 para se ter um valor em N/m^2 .

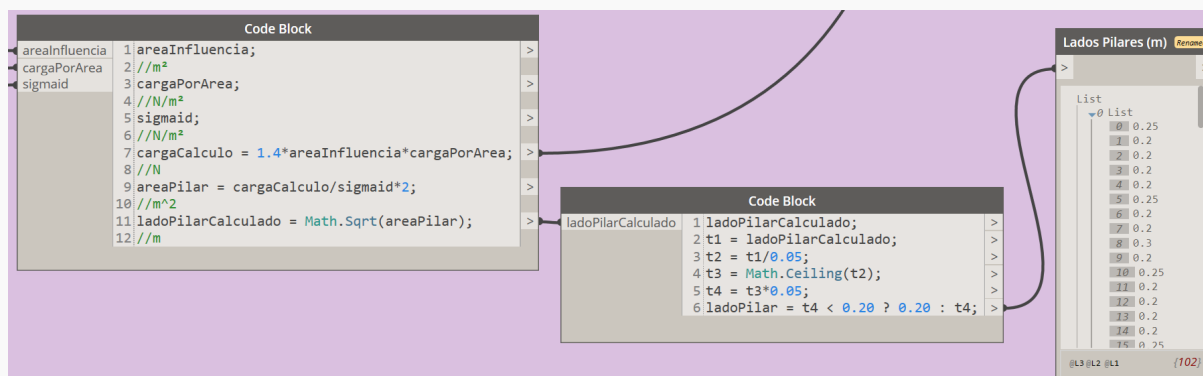
Figura 5.36 – Grupo para pré-dimensionamento dos pilares (trecho 10)



Fonte: Autor

- No *Code Block* define-se a variável `areaInfluencia`, que armazena a saída do nó *Áreas de Influência (m²)*, já mostrado na Figura 5.34. Como a unidade de comprimento do projeto é metro, essa área tem valor em m², o que indicado é na linha de comentário abaixo da sua definição. As unidades das demais variáveis são também indicadas nas linhas abaixo das que as definem.
- As variáveis `cargaPorArea` e `sigmaid` recebem os valores armazenados nas variáveis de mesmo nome definidas no *Code Block* da Figura 5.35.
- A variável `cargaCalculo` tem seu valor definido de maneira análoga ao que é apresentado na Equação 3.29.
- A variável `areaPilar` é definida segundo a Equação 3.27. Vale ressaltar que adotou-se o valor de γ_{corr} igual a 2 para todos os trechos de pilar.
- Ao final, calcula-se a variável `ladoPilarCalculado`, obtida como sendo a raiz quadrada da área da seção do trecho do pilar em razão de se considerar os pilares como sendo de seção quadrada.

Figura 5.37 – Grupo para pré-dimensionamento dos pilares (trecho 11)



Fonte: Autor

- O valor armazenado na variável `ladoPilarCalculado` no *Code Block* à esquerda é armazenado em variável de mesmo nome definida no *Code Block* à direita.
- A variável `t1` armazena o conteúdo da variável `ladoPilarCalculado`. Os cálculos realizados nas definições de `t2`, `t3` e `t4` objetivam arredondar o valor de `t1` para o primeiro múltiplo de 0,05 superior a esse valor.
- Por fim, define-se o valor de `ladoPilar` a partir de uma expressão condicional, que garante que o valor armazenado seja maior ou igual a 0,20 m. Esse valor é o que vai ser passado como lado do trecho de pilar a ser concebido no modelo.
- O nó *Watch*, renomeado para *Lados Pilares (m)*, armazena os valores do lado de cada trecho de pilar, a serem usados na criação dos elementos de modelo.

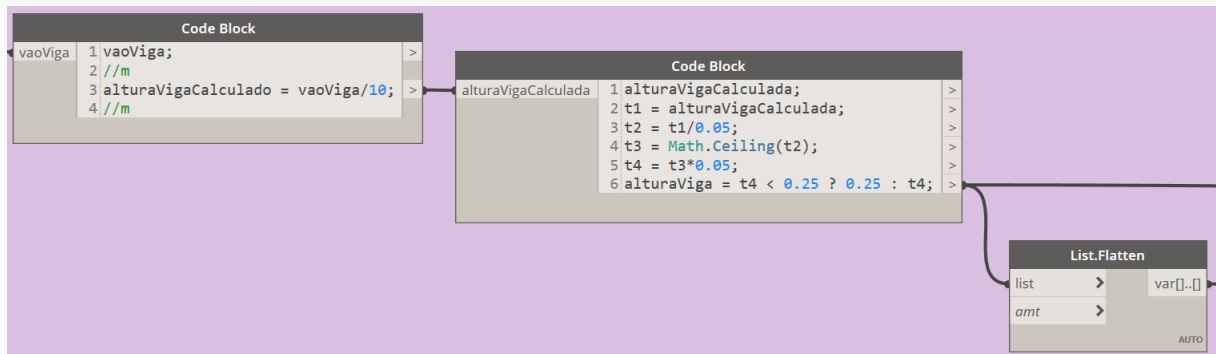
Calcula-se o somatório das áreas de influência correspondentes a cada trecho de pilar. A partir dos valores de f_{ck} , taxa de armadura do pilar e carga por área da laje, passados como entrada da rotina, aplicam-se as Equações 3.27, 3.28, 3.29, das quais se obtém o lado de cada pilar de seção quadrada.

Dos valores de lado calculados, é feito arredondamento para múltiplo de 0,05 m imediatamente superior. Ainda, em razão do exposto na subseção 3.7.3, as dimensões finais tomadas são superiores ou iguais a 20 cm.

5.4.2 Pré-dimensionamento de viga

O grupo apresentado na Figura 5.38 tem por finalidade definir as dimensões das seções das vigas. No algoritmo que será descrito, serão tomadas as linhas de localização das vigas, a partir das quais, com auxílio do proposto na subseção 3.7.2, Vigas, será feito o pré-dimensionamento da seção.

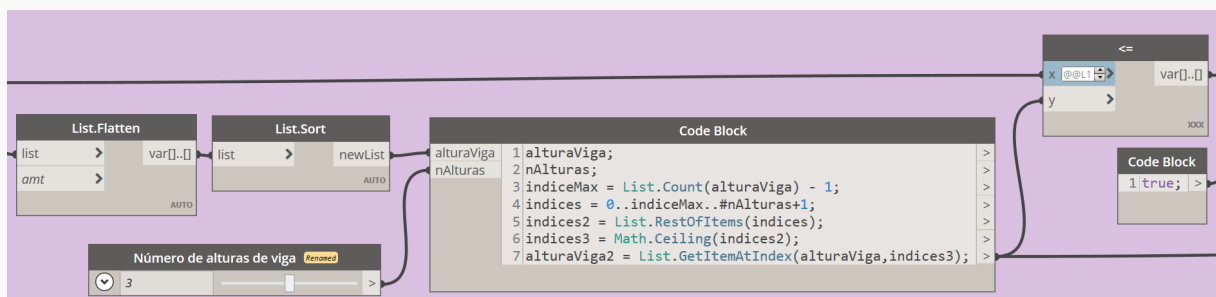
Figura 5.40 – Grupo para pré-dimensionamento das vigas (trecho 2)



Fonte: Autor

- A variável `alturaVigaCalculada` é definida no segundo *Code Block* e recebe o valor armazenado na variável de mesmo nome do *Code Block* que o antecede.
- As variáveis `t1`, `t2`, `t3` e `t4` conduzem ao arredondamento do valor armazenado em `alturaVigaCalculada` para o primeiro múltiplo de 0,05 superior a esse valor.
- Ao final, define-se uma expressão condicional que garante que o valor armazenado em `alturaViga` seja superior ou igual a 0,25 m.
- O nó *List.Flatten* aplaina a lista de valores da variável `alturaViga`.

Figura 5.41 – Grupo para pré-dimensionamento das vigas (trecho 3)

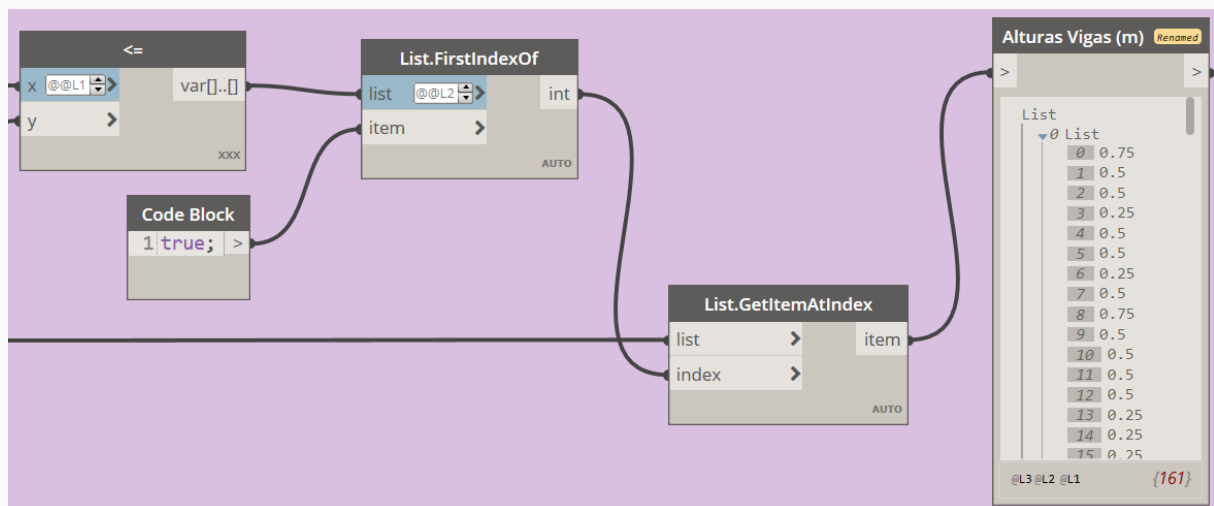


Fonte: Autor

- A lista aplainada saída do nó *List.Flatten* é ordenada de maneira crescente no nó *List.Sort*. Em seguida, é armazenada na variável `alturaViga` do *Code Block*.
- O nó *Number Slider* renomeado para *Número de alturas de viga* retorna um número inteiro entre 1 e 5, definido pela posição do controle deslizante. Esse valor irá definir quantos valores diferentes de altura de viga serão retornados ao final. O valor do nó *Número de alturas de viga* é armazenado em `nAlturas` no *Code Block*.

- As variáveis `indiceMax`, `indices`, `indices2` e `indices3` participam do processo de definição dos diferentes valores de altura de viga. Esses valores são os armazenados na variável `alturaViga2`.
- A variável `alturaViga2` armazena os possíveis valores de altura de viga. A quantidade de valores é aquela passada no nó *Número de alturas de viga*.
- O nó `<=` compara cada valor da variável `alturaViga`, definido na Figura 5.40, com os valores armazenados em `alturaViga2`. Quando o primeiro for menor ou igual ao segundo, retorna-se o booleano *true*, em caso contrário, retorna-se *false*. Para cada viga é gerada uma lista contendo tantos valores booleanos quantos são os valores em `alturaViga2`.

Figura 5.42 – Grupo para pré-dimensionamento das vigas (trecho 4)



Fonte: Autor

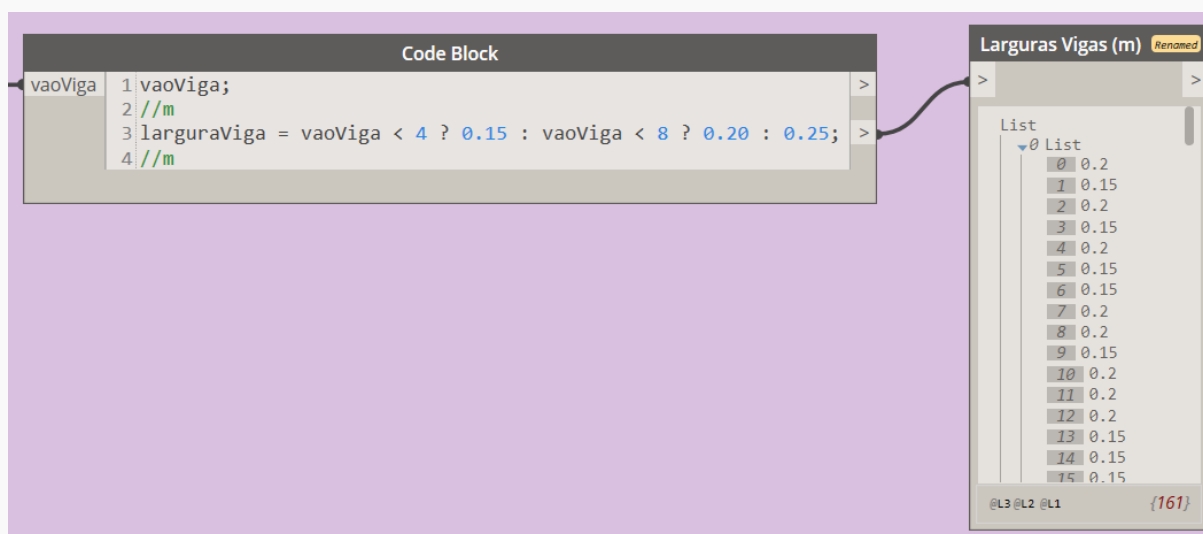
- Das listas de valores booleanos geradas como saída do nó `<=` obtém-se os índices do primeiro valor *true*, através do nó *List.FirstIndexOf*.
- O nó *List.GetItemAtIndex* retorna para cada índice gerado como saída no nó *List.FirstIndexOf* um valor de altura de viga, obtida da variável `alturaViga2` passada na entrada *list*.
- O nó *Watch* renomeado para *Alturas Vigas (m)* exibe o valor de altura correspondente a cada linha de localização de viga passada como entrada no grupo em questão.

Cada linha de localização de viga (Figura 5.20) corresponde a um vão livre de viga. Dos comprimentos das linhas de viga, aplica-se a Equação 3.23 da qual se obtém um valor de altura.

Esse valor é arredondado para múltiplo de 0,05 m e tomado superior ou igual a 25 cm.

Da lista de alturas obtidas, é feito um procedimento de uniformização de valores, a partir do valor de número de alturas de viga passado como entrada na rotina. Esse procedimento retorna uma lista final com tantos valores diferentes quanto o passado na entrada, de maneira que nenhum valor seja inferior ao inicialmente calculado.

Figura 5.43 – Grupo para pré-dimensionamento das vigas (trecho 5)



Fonte: Autor

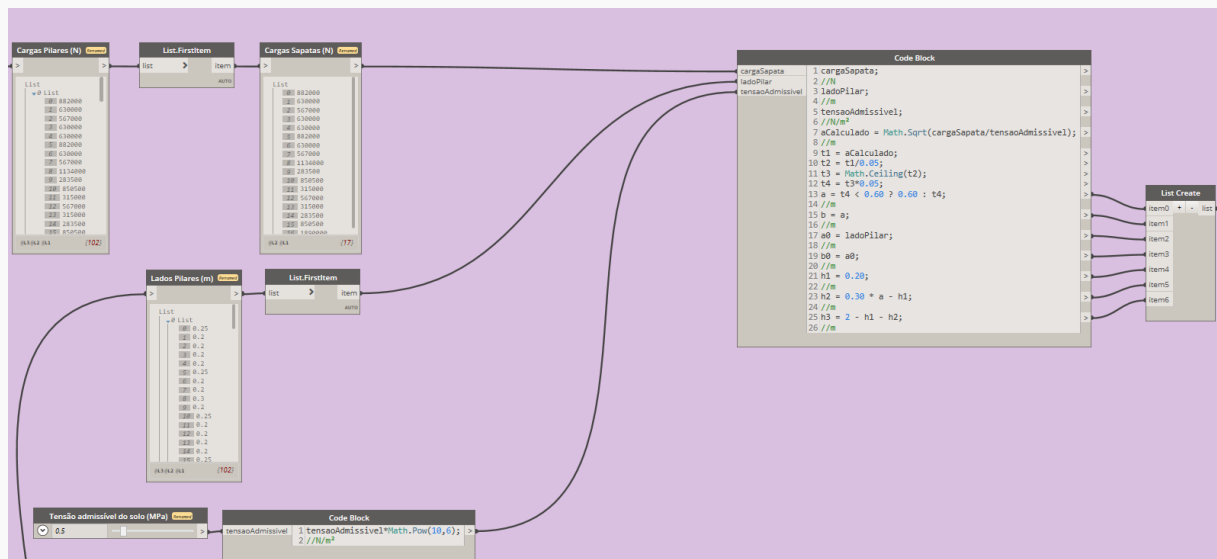
- A variável `vaoViga` recebe a saída do nó *Curve.Length* exibido na Figura 5.39. O valor é dado em metro, como exibido na linha de comentário abaixo da definição da variável.
- A variável `larguraViga` tem seu valor dado por uma expressão condicional, definida tomando como base a Equação 3.24.
- O nó *Watch*, renomeado para *Larguras Vigas (m)* exibe os valores de largura em metro calculados para cada viga.

O cálculo das larguras das vigas é feito tomando como base a Equação 3.24, adicionando-se a restrição de se ter valores múltiplos de 0,05 m.

5.4.3 Pré-dimensionamento de sapata

O grupo apresentado na Figura 5.44 tem por finalidade definir as dimensões das seções das sapatas. No algoritmo que será descrito, serão tomados os valores de carga obtido do pré-dimensionamento dos pilares, bem como as dimensões desses. O processo de pré-dimensionamento das sapatas é norteado pelo proposto na subseção 3.7.4, Sapatas.

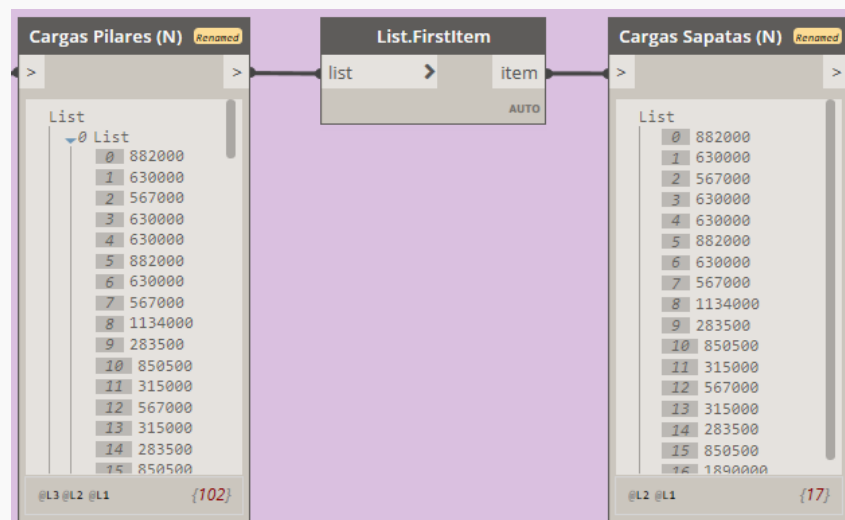
Figura 5.44 – Grupo para pré-dimensionamento das sapatas



Fonte: Autor

Os trechos do grupo são apresentadas nas Figuras 5.45, 5.46, 5.47 e 5.48 abaixo das quais são fornecidos comentários com as explicações do código exibido.

Figura 5.45 – Grupo para pré-dimensionamento das sapatas (trecho 1)

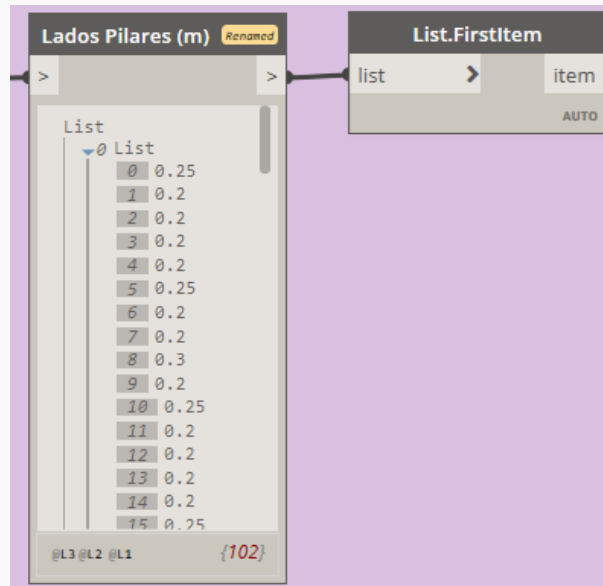


Fonte: Autor

- O nó *Watch* renomeado para *Cargas Pilares (N)* recebe o valor das cargas dos pilares, proveniente da variável `cargaCalculo`, definida no *Code Block* apresentado na Figura 5.37.
- O nó *List.FirstItem* toma a primeira lista do valor passado como entrada. No caso, será retornada a lista correspondente a carga dos trechos de pilar do primeiro pavimento.

- Essa carga é assumida como a carga recebida na fundação, exibida no nó *Watch* renomeado para *Cargas Sapatas (N)*.

Figura 5.46 – Grupo para pré-dimensionamento das sapatas (trecho 2)



Fonte: Autor

- O nó *Watch*, renomeado para *Lados Pilares (m)*, recebe a saída do nó de mesmo nome exibido na Figura 5.37.
- O nó *List.FirstItem* vai selecionar apenas a primeira lista do nó *Lados Pilares (m)*, lista essa que armazena as dimensões dos trechos de pilar do primeiro pavimento.

Figura 5.47 – Grupo para pré-dimensionamento das sapatas (trecho 3)

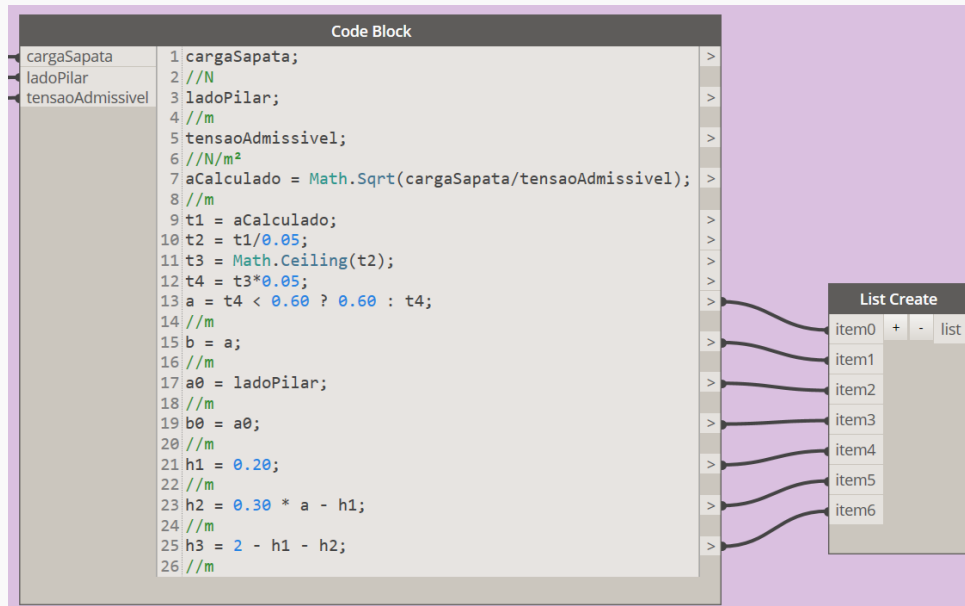


Fonte: Autor

- O nó *Tensão admissível do solo (MPa)* é um nó *NumberSlider* renomeado, cujos valores mínimo, máximo e intervalo entre valores consecutivos são definidos.
- O valor é passado para a variável `tensaoAdmissivel` do *Code Block* e multiplicado por

10^6 , a fim de se ter o valor na unidade N/m^2 .

Figura 5.48 – Grupo para pré-dimensionamento das sapatas (trecho 4)



Fonte: Autor

- No primeiro *Code Block* define-se a variável `cargaSapata`, que recebe a saída do nó *Cargas Sapatas (N)* exibido na Figura 5.45. A variável `ladoPilar`, que recebe a saída do nó *List.FirstItem* da Figura 5.46. A variável `tensaoAdmissivel`, que recebe a saída do *Code Block* exibido na Figura 5.47.
- A variável `aCalculado` tem seu valor dado pela Equação 3.35.
- As variáveis `t1`, `t2`, `t3` e `t4` conduzem ao arredondamento do valor armazenado em `aCalculado` para o primeiro múltiplo de 0,05 superior a esse valor.
- A variável `a` recebe o valor arredondado. É feita uma declaração condicional que garante que esse valor seja superior ou igual a 0,60. A variável `b` recebe o valor de `a`.
- A variável `a0` recebe `ladoPilar`. O mesmo valor é armazenado em `b0`.
- As variáveis `h1`, `h2` e `h3` definem a geometria da sapata. A variável `h1` recebe 0,20 m. A variável `h2` tem seu valor dado pela Equação 3.37. A variável `h3` é definida de maneira que todas as sapatas tenham base situada em profundidade de 2 m.
- Ao final, é criada uma lista por meio do nó *List Create*. Essa lista armazena o conteúdo das variáveis `a`, `b`, `a0`, `b0`, `h1`, `h2` e `h3`, cada qual em uma sublista diferente.

Toma-se como valor de carga em cada sapata aquele calculado para o trecho de pilar situado ao nível do pavimento mais inferior correspondente. Ainda, são tomadas as dimensões desses mesmos trechos de pilar, usadas na determinação da geometria das sapatas.

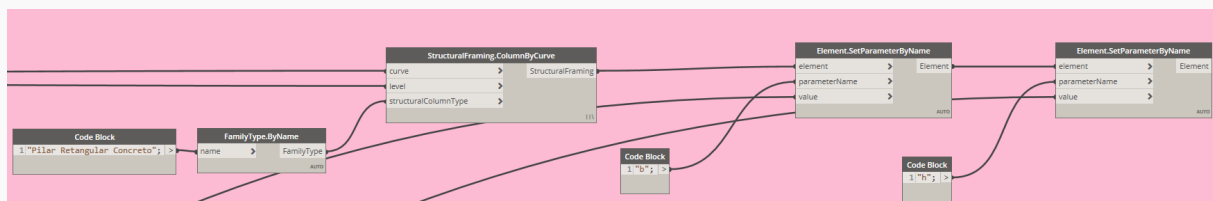
De posse do valor da carga e da tensão admissível do solo passada como entrada da rotina, faz-se uso da Equação 3.35 pra cálculo do lado da base da sapata. As demais dimensões que determinam as dimensões da sapata são calculadas conforme proposto na subseção 3.7.4.

5.5 Criação dos elementos estruturais

5.5.1 Criação dos elementos de pilar

O grupo apresentado na Figura 5.49 cria os elementos de pilar no modelo. Para isso, é feito uso das geometrias que definem a posição dos trechos de pilar e dos valores de dimensões de seção obtidos do pré-dimensionamento.

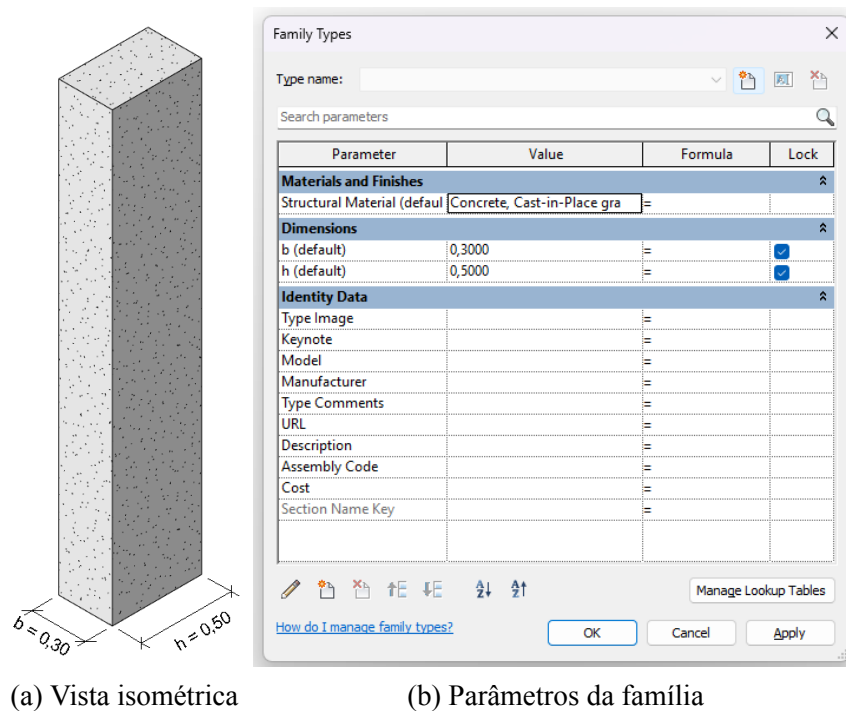
Figura 5.49 – Grupo para criação dos elementos de pilar



Fonte: Autor

Os elementos a serem inseridos no modelo são instâncias da família Pilar Retangular Concreto, exibida na Figura 5.50. As dimensões da seção transversal desses elementos são governadas pelos parâmetros de instância b e h . Essa família tem um único tipo também de nome Pilar Retangular Concreto.

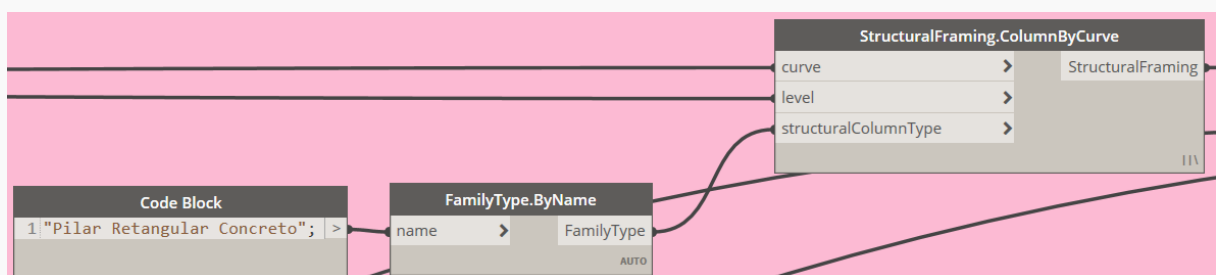
Figura 5.50 – Família Pilar Retangular Concreto



Fonte: Autor

Os trechos do grupo são apresentadas nas Figuras 5.51 e 5.52, abaixo das quais são fornecidos comentários com as explicações do código exibido.

Figura 5.51 – Grupo para criação dos elementos de pilar (trecho 1)

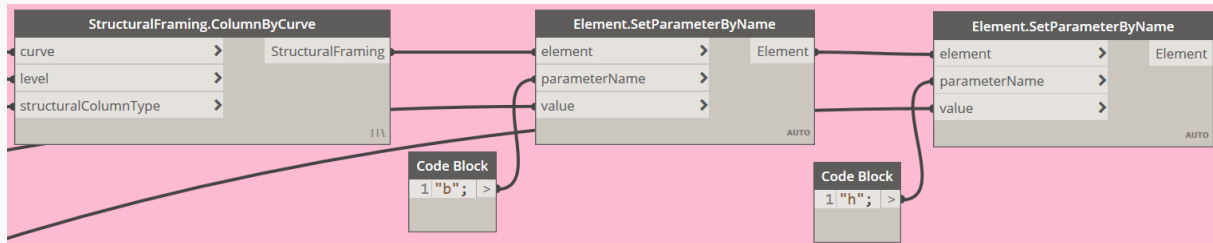


Fonte: Autor

- O *Code Block* e o nó *FamilyType.ByName* selecionam o tipo de família Pilar Retangular Concreto.
- O nó *StructuralFraming.ColumnByCurve* cria uma instância do tipo de família passado na entrada *structuralColumnType*, determinado pela curva passada em *curve* e de nível da base passado em *level*.
- Na entrada *curve* é passada a saída do nó *Linhas Pilares*, exibido na Figura 5.17.

- Na entrada *level* é passada a saída do nó *Níveis*, exibido na Figura 5.7.

Figura 5.52 – Grupo para criação dos elementos de pilar (trecho 2)

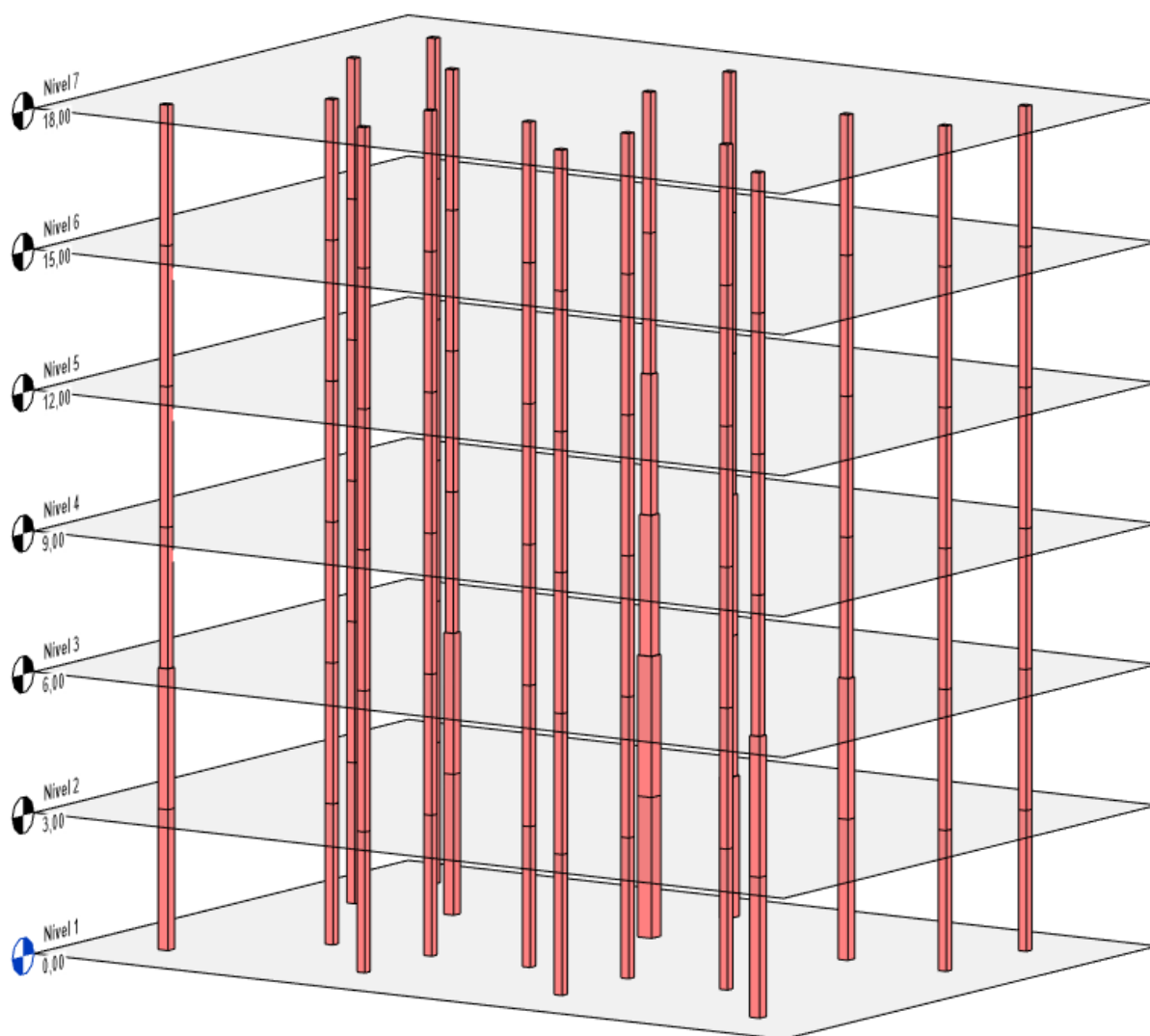


Fonte: Autor

- O nó *StructuralFraming.ColumnByCurve* tem como saída as instâncias da família Pilar Retangular Concreto criadas.
- O nó *Element.SetParameterByName* altera o parâmetro passado na entrada *parameterName* para o valor passado em *value* da instância passada em *element*.
- No primeiro nó *Element.SetParameterByName* o parâmetro *b* das instâncias de Pilar Retangular Concreto criadas tem valor alterado para aqueles armazenados no nó *Lados Pilares (m)* exibido na Figura 5.37.
- De maneira análoga, o segundo nó *Element.SetParameterByName* altera o parâmetro *h* das instâncias para aqueles também passados como saída do nó *Lados Pilares (m)* exibido na Figura 5.37.

Ao final da execução do grupo, tem-se criadas em projeto as instâncias da família Pilar Retangular Concreto. A posição dessas instâncias foi definida ao final da execução da parte Criação das geometrias de referência (seção 5.3) conforme a Figura 5.19. As dimensões das seções dos trechos de pilar, determinadas pelos parâmetros *b* e *h* (Figura 5.50), foram definidas ao final da execução da parte Pré-dimensionamento dos elementos estruturais (seção 5.4). O resultado é exibido na Figura 5.53.

Figura 5.53 – Vista isométrica das instâncias de pilar

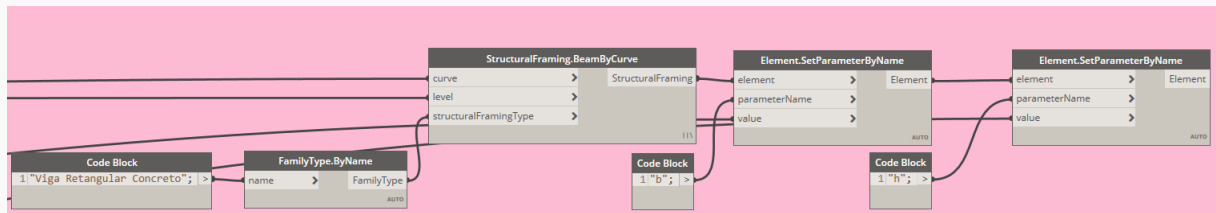


Fonte: Autor

5.5.2 Criação dos elementos de viga

O grupo apresentado na Figura 5.54 cria os elementos de viga no modelo. Para isso, é feito uso das geometrias que definem a posição das vigas e dos valores de dimensões de seção obtidos do pré-dimensionamento.

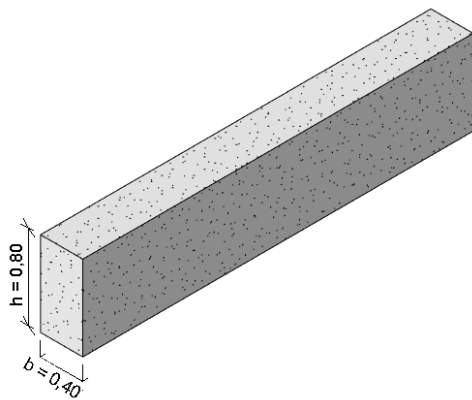
Figura 5.54 – Grupo para criação dos elementos de viga



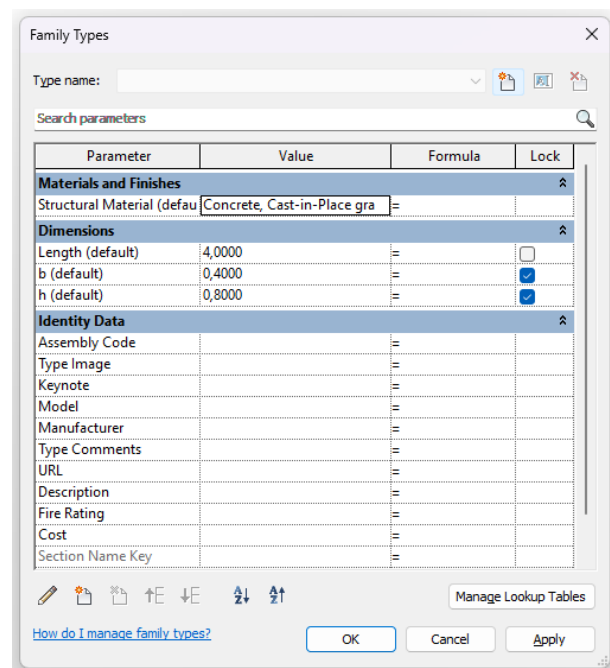
Fonte: Autor

Os elementos a serem inseridos no modelo são instâncias da família Viga Retangular Concreto, exibida na Figura 5.55. As dimensões da seção transversal desses elementos são governadas pelos parâmetros de instância b e h . Essa família tem um único tipo também de nome Viga Retangular Concreto.

Figura 5.55 – Família Viga Retangular Concreto



(a) Vista isométrica

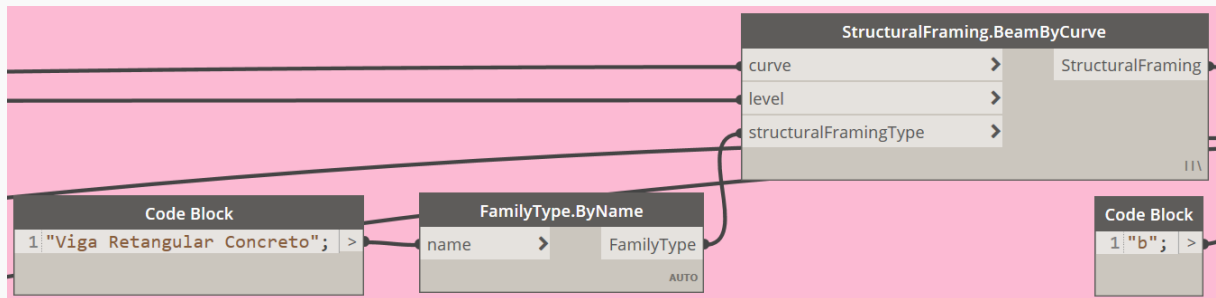


(b) Parâmetros da família

Fonte: Autor

Os trechos do grupo são apresentadas nas Figuras 5.56 e 5.57, abaixo das quais são fornecidos comentários com as explicações do código exibido.

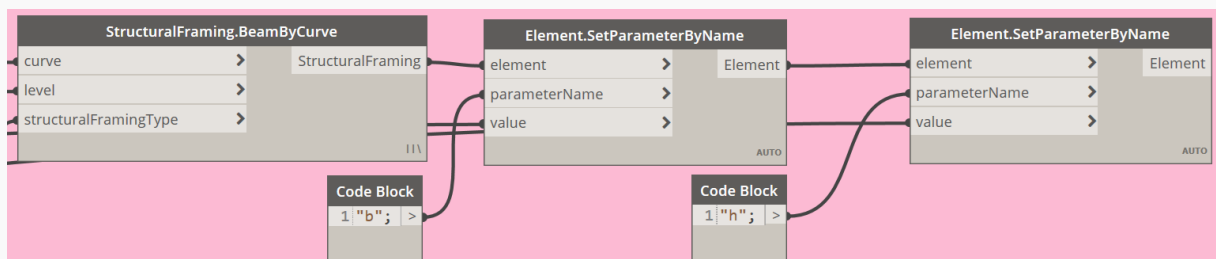
Figura 5.56 – Grupo para criação dos elementos de viga (trecho 1)



Fonte: Autor

- O *Code Block* e o nó *FamilyType.ByName* selecionam o tipo de família Viga Retangular Concreto.
- O nó *StructuralFraming.BeamByCurve* cria uma instância do tipo de família passado na entrada *structuralFramingType*, determinado pela curva passada em *curve* e de nível da base passado em *level*.
- Na entrada *curve* é passada a saída do nó *Linhas Vigas*, exibido na Figura 5.18.
- Na entrada *level* é passada a saída do nó *Níveis*, exibido Figura 5.7.

Figura 5.57 – Grupo para criação dos elementos de viga (trecho 2)



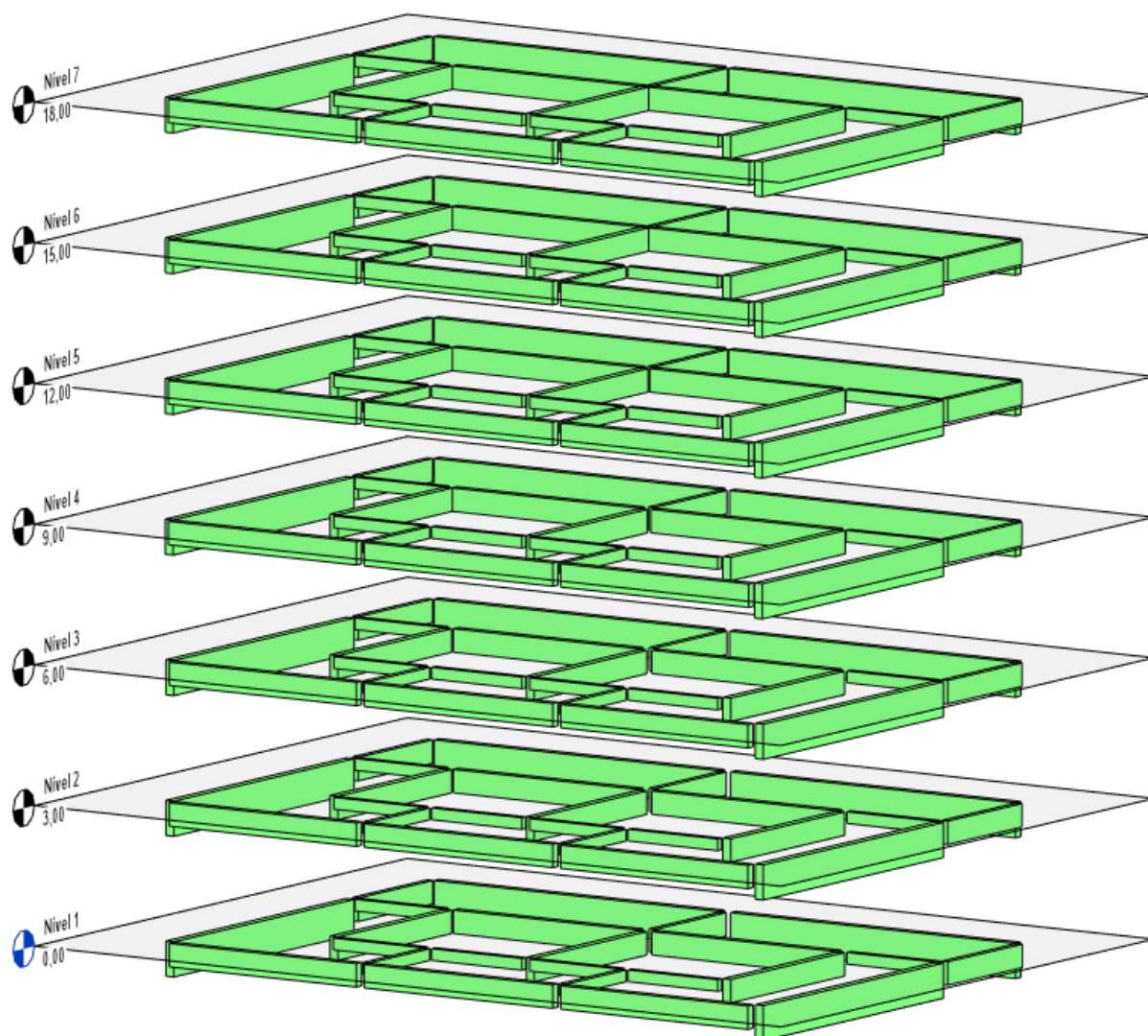
Fonte: Autor

- O nó *StructuralFraming.BeamByCurve* tem como saída as instâncias da família Viga Retangular Concreto criadas.
- O nó *Element.SetParameterByName* altera o parâmetro passado na entrada *parameterName* para o valor passado em *value* da instância passada em *element*.
- No primeiro nó *Element.SetParameterByName* o parâmetro *b* das instâncias de Viga Retangular Concreto criadas tem valor alterado para aqueles armazenados no nó *Larguras Vigas (m)* exibido na Figura 5.43.

- De maneira análoga, o segundo nó *Element.SetParameterByName* altera o parâmetro *h* das instâncias para aqueles passados como saída do nó *Alturas Vigas (m)* exibido na Figura 5.42.

Ao final da execução do grupo, tem-se criadas em projeto as instâncias da família Viga Retangular Concreto. A posição dessas instâncias foi definida ao final da execução da parte Criação das geometrias de referência (seção 5.3) conforme a Figura 5.20. As dimensões das seções das vigas, determinadas pelos parâmetros *b* e *h* (Figura 5.55), foram definidas ao final da execução da parte Pré-dimensionamento dos elementos estruturais (seção 5.4). O resultado é exibido na Figura 5.58.

Figura 5.58 – Vista isométrica das instâncias de viga

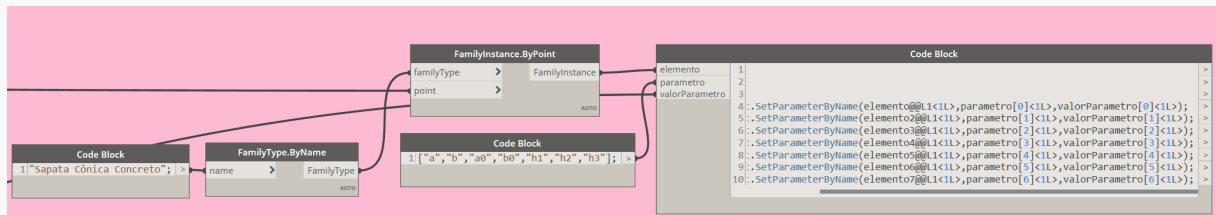


Fonte: Autor

5.5.3 Criação dos elementos de sapata

O grupo apresentado na Figura 5.59 cria os elementos de sapata no modelo. Para isso, é feito uso dos pontos que definem a posição das sapatas e dos valores de dimensões obtidos do pré-dimensionamento.

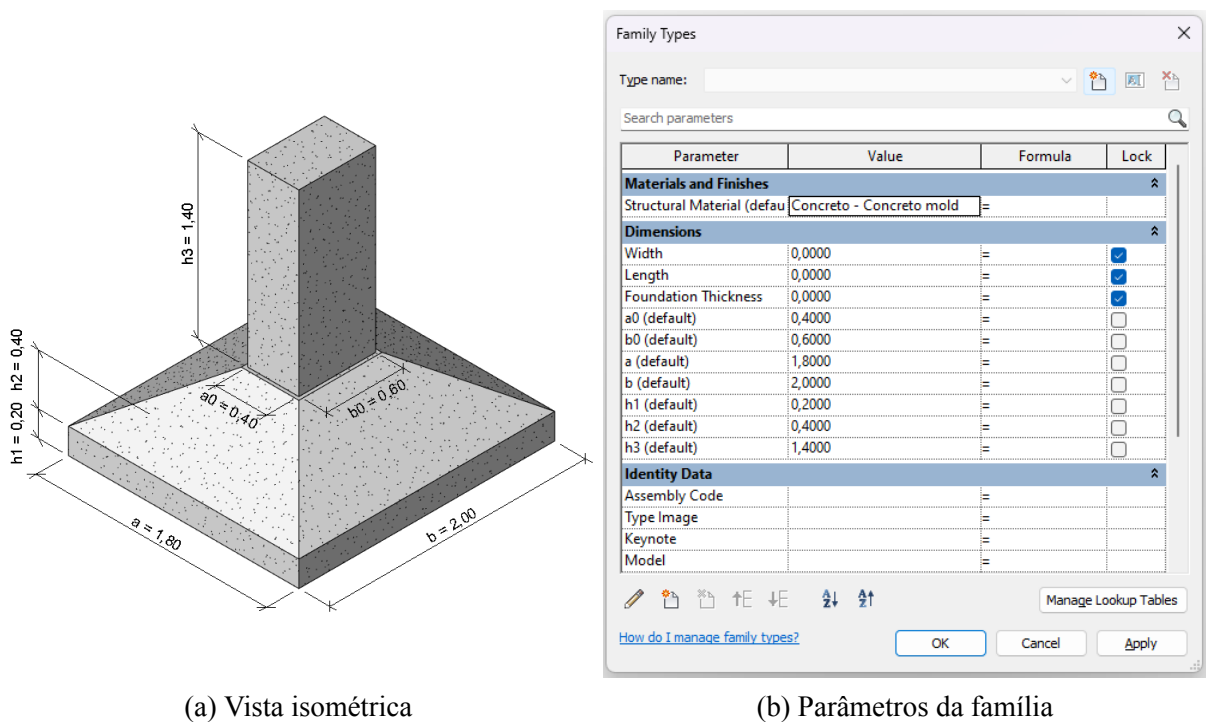
Figura 5.59 – Grupo para criação dos elementos de sapata



Fonte: Autor

Os elementos a serem inseridos no modelo são instâncias da família Sapata Cônica Concreto, exibida na Figura 5.60. As dimensões da geometria desse elemento são governadas pelos parâmetros de instância a , b , $a0$, $b0$, $h1$, $h2$ e $h3$. Essa família tem um único tipo também de nome Sapata Cônica Concreto.

Figura 5.60 – Família Sapata Cônica Concreto



(a) Vista isométrica

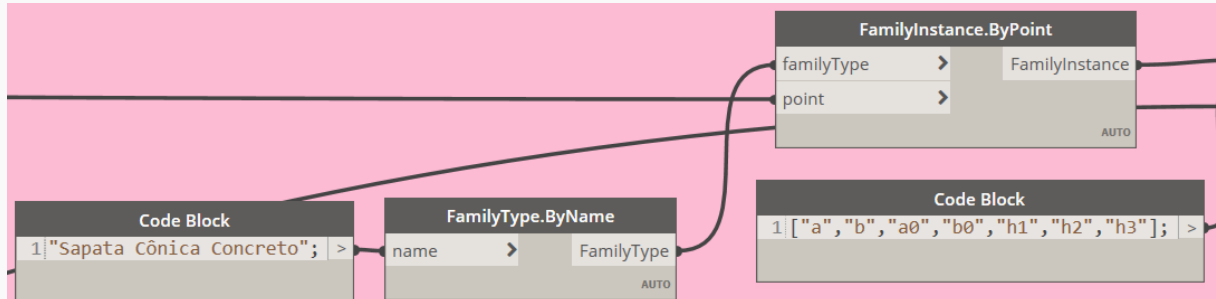
(b) Parâmetros da família

Fonte: Autor

Os trechos do grupo são apresentadas nas Figuras 5.61 e 5.62, abaixo das quais são

fornecidos comentários com as explicações do trecho de código exibido.

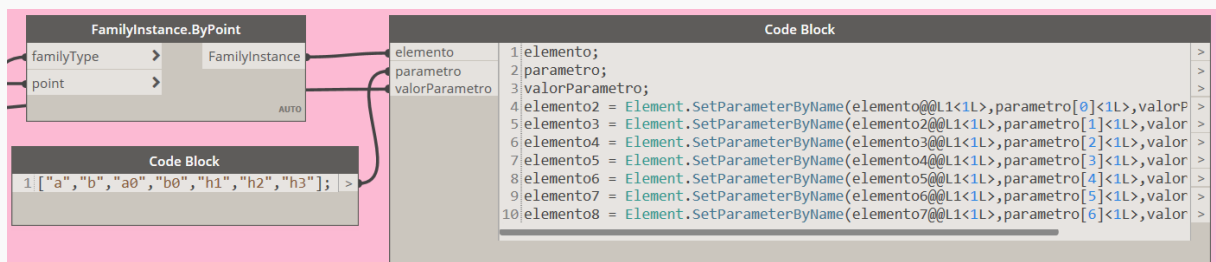
Figura 5.61 – Grupo para criação dos elementos de sapata (trecho 1)



Fonte: Autor

- O *Code Block* e o nó *FamilyType.ByName* selecionam o tipo de família Sapata Cônica Concreto.
- O nó *FamilyInstance.ByPoint* cria uma instância do tipo de família passado na entrada *familyType* no ponto passado em *point*.
- Na entrada *point* é passada a saída do nó *Pontos Sapatas*, exibido na Figura 5.16.

Figura 5.62 – Grupo para criação dos elementos de sapata (trecho 2)



Fonte: Autor

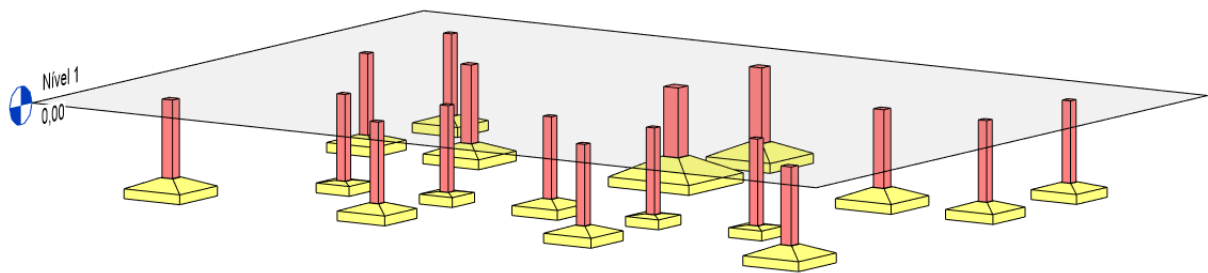
- O nó *FamilyInstance.ByPoint* tem como saída as instâncias da família Sapata Cônica Concreto criadas.
- O *Code Block* à esquerda define uma lista que armazena o nome dos parâmetros da família Sapata Cônica Concreto, como exibido na Figura 5.60.
- No *Code Block* à direita define-se a variável `elemento`, que recebe as instâncias produzidas como saída do nó *FamilyInstance.ByPoint*. A variável `parametro` recebe a lista definida no *Code Block* anterior. A variável `valorParametro` recebe a lista com os va-

lores obtidos do pré-dimensionamento da sapata, provenientes do nó *List Create* exibido na Figura 5.48.

- As demais linhas do *CodeBlock*, nas quais são definidas as variáveis `elemento2`, `elemento3`, `elemento4`, `elemento5`, `elemento6`, `elemento7` e `elemento8` se encarregam de alterar cada um dos parâmetros da família, armazenados em `parametro`, para o correspondente valor que deve assumir, armazenado em `valorParametro`.

Ao final da execução do grupo, tem-se criadas em projeto as instâncias da família Sapata Cônica Concreto. A posição dessas instâncias foi definida ao final da execução da parte Criação das geometrias de referência (seção 5.3) conforme a Figura 5.21. As dimensões das seções das sapatas, determinadas pelos parâmetros a , b , $a0$, $b0$, $h1$, $h2$ e $h3$ (Figura 5.60), foram definidas ao final da execução da parte Pré-dimensionamento dos elementos estruturais (seção 5.4). O resultado é exibido na Figura 5.63.

Figura 5.63 – Vista isométrica das instâncias de sapata

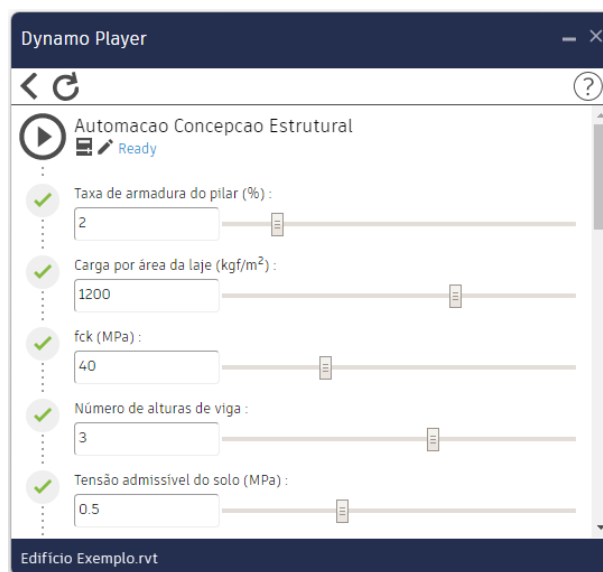


Fonte: Autor

5.6 Execução da rotina

A execução da rotina no ambiente do Revit é feita através do Dynamo Player, que possibilita a execução da rotina e alteração dos valores definidos como de entrada, como exibido na Figura 5.64.

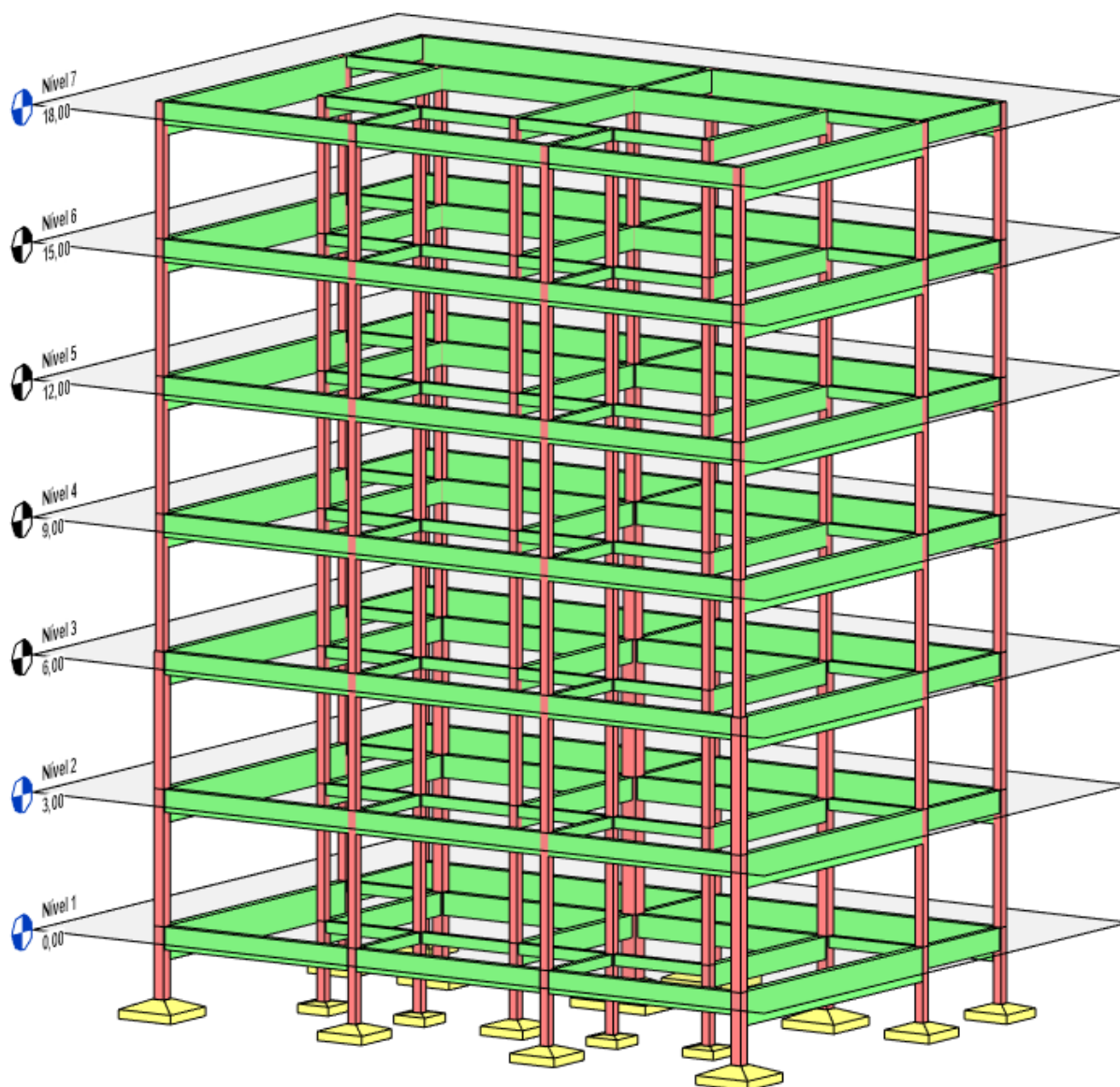
Figura 5.64 – Dynamo Player



Fonte: Autor

A figura Figura 5.65 exibe a saída da execução da rotina, na qual constam todas as instâncias dos elementos estruturais da edificação passada como entrada.

Figura 5.65 – Vista isométrica das instâncias de pilar, viga e sapata



Fonte: Autor

6 APLICAÇÃO DA ROTINA

O presente capítulo se propõe a apresentar um caso de aplicação da rotina, tomando para isso uma arquitetura cuja estrutura foi previamente concebida manualmente por engenheiro. A partir da definição dos demais valores de entrada, a rotina é executada, gerando como saída um novo modelo estrutural. Entre a concepção manual e aquela originada da rotina são feitas comparações quantitativas e qualitativas, a fim de se ter critérios que permitam a avaliação do resultado obtido como fruto dessa pesquisa.

6.1 Modelo arquitetônico

A arquitetura passada como entrada da rotina é adaptada de Queiroz (2023), exibida no Anexo A, ARQUITETURA. Se trata de um estabelecimento comercial que aloca unidades destinadas a *Self Storage* e *Coworking*.

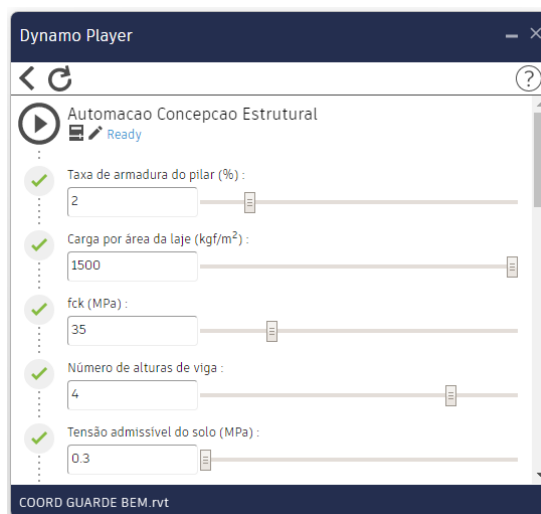
6.2 Modelo estrutural concebido manualmente

O modelo estrutural concebido manualmente é de autoria da TECNCON - Engenharia Estrutural (2023). A solução adotada foi em concreto armado, fazendo uso de laje nervurada, vigas e solução de fundação em sapata. A estrutura é apresentado no Anexo B, ESTRUTURA CONCEBIDA MANUALMENTE.

6.3 Valores passados como entrada da rotina

Além da arquitetura já apresentada, as demais entradas da rotina são exibidas na Figura 6.1.

Figura 6.1 – Valores de entrada da rotina



A captura de tela mostra a interface do 'Dynamo Player' com o título 'Automacao Concepcao Estrutural'. Abaixo do título, há um ícone de play e o status 'Ready'. A interface contém cinco controles deslizantes, cada um com um ícone de checkmark verde à esquerda. Os controles deslizantes são: 'Taxa de armadura do pilar (%)' com o valor 2; 'Carga por área da laje (kgf/m²)' com o valor 1500; 'fck (MPa)' com o valor 35; 'Número de alturas de viga' com o valor 4; e 'Tensão admissível do solo (MPa)' com o valor 0.3. No rodapé da janela, há o texto 'COORD GUARDE BEM.rvt'.

Parâmetro	Valor
Taxa de armadura do pilar (%)	2
Carga por área da laje (kgf/m²)	1500
fck (MPa)	35
Número de alturas de viga	4
Tensão admissível do solo (MPa)	0.3

Fonte: Autor

Os valores de taxa de armadura do pilar e fck são os mesmos usados no modelo estrutural concebido manualmente. A carga por área da laje adotada é o maior dos valores previstos na Equação 3.30, tendo em vista o uso da edificação como depósito. Para número de alturas de viga considerou-se o valor de 4. Por fim, a tensão admissível do solo foi adotada conforme a Tabela 3.10, tomando valor compatível com o solo da localidade em questão.

6.4 Modelo estrutural concebido pela rotina

Uma vez que a rotina é executada obtém-se novo modelo estrutural, conforme exibido no Apêndice D, ESTRUTURA CONCEBIDA PELA ROTINA.

6.5 Avaliação dos resultados da rotina

Comparando o modelo estrutural concebido manualmente por engenheiro com aquele produzido pela rotina, observa-se, quanto aos elementos de pilar, um maior número no segundo modelo do que no primeiro.

A explicação para isso está relacionada à etapa de Criação das geometrias de referência, descrita na seção 5.3, na qual foi feita a criação de linhas de localização dos trechos de pilar coincidentes com todos os pontos de interseção entre linhas de localização de paredes.

Quanto às seções dos elementos de pilar, observa-se a adoção de seções menores na concepção estrutural da rotina, o que se justifica no fato de, havendo maior número de pilares, se ter menores áreas de influência, o que implica em seções reduzidas. O procedimento de pré-dimensionamento dos pilares, segundo o cálculo das áreas de influência, é descrito na seção 5.4. Outra observação pertinente diz respeito à redução da seção dos pilares com a altura, observada

apenas no modelo concebido pela rotina, em razão da carga atuante nos trechos superiores ser menor.

Em termos quantitativos, a solução concebida manualmente apresenta volume de concreto referente aos pilares de 26,11 m³, enquanto a solução concebida pela rotina apresenta volume de 22,15 m³. A proximidade nos valores é justificada no fato de se ter seções reduzidas dos pilares e redução das seções com a altura na concepção feita pela rotina, o que compensa o efeito que o maior número de elementos tem sobre o volume total.

No que diz respeito às vigas, tal qual os pilares, observa-se número maior de elementos na concepção feita pela rotina do que naquela manual. Para essa observação, dá-se a mesma explicação já apresentada para os pilares.

Quanto às seções desses elementos, observa-se uma maior uniformidade nas dimensões tanto de largura como de altura para a concepção manual. Outra observação pertinente é a existência de elementos de viga parede na região do poço do elevador e vigas invertidas na concepção manual, o que não é observado na concepção feita pela rotina.

Em termos quantitativos, observa-se um volume de concreto de viga consideravelmente maior na concepção da rotina: 83,60 m³ contra 57,92 m³ da concepção manual. A explicação para isso se deve ao lançamento de mais vigas na primeira solução.

Ambas as soluções de fundação se deram pelo uso de sapatas, sendo uma para cada pilar. Em decorrência disso, tal qual se observou para os pilares, o número de sapatas é maior na concepção feita pela rotina.

Apesar do número consideravelmente maior de sapatas na concepção feita pela rotina, os volumes de concreto obtidos são próximos: 21,04 m³ para a concepção manual e 23,28 m³ para a concepção da rotina. A explicação para isso é semelhante àquela feita para justificar a proximidade nos volumes referentes aos pilares: um número maior de pilar leva a cargas menores nos mesmos e por vez nas sapatas, o que contribui para a redução no volume das sapatas, compensando sua maior quantidade.

Um resumo dos quantitativos dos dois modelos estruturais abordados é apresentado na Tabela 6.1.

Tabela 6.1 – Resumo dos quantitativos dos modelos estruturais

Elemento Estrutural	Volume de Concreto (m ³)		Diferença Relativa
	Concepção Manual	Concepção Rotina	
Pilar	26,11	22,15	-15%
Viga	57,92	83,60	44%
Sapata	21,04	23,28	11%
Total	105,07	129,03	23%

Fonte: Autor

7 CONSIDERAÇÕES FINAIS

A avaliação de diversas alternativas de concepção estrutural passa pela necessidade de atendimento a diversos critérios, com destaque para a satisfação de condições impostas pela arquitetura e aspectos construtivos.

Ainda, destaca-se a participação significativa da execução da estrutura no orçamento de uma obra, motivando que diversas soluções para uma mesma arquitetura sejam concebidas e avaliadas.

Nesse aspecto, se apoiando nos avanços tecnológicos referentes ao BIM, acompanhados do surgimento e aprimoramento de *softwares* como o Revit e o Dynamo, a presente pesquisa se propôs a abordar a automação do processo de concepção de estrutura em concreto armado no Revit através do desenvolvimento de rotina no Dynamo.

Para isso, fez-se necessário o estudo das etapas que envolvem o processo de concepção de estrutura em concreto armado; o entendimento dos conceitos que envolvem o BIM, bem como dos *softwares* Revit e Dynamo; o desenvolvimento propriamente dito da rotina, em concordância com as etapas de concepção estrutural e a avaliação da rotina desenvolvida, através do comparação do modelo estrutural gerado por sua execução com o concebido manualmente para mesma arquitetura.

O resultado final foi uma rotina capaz de cumprir o seu propósito: automatizar o processo de concepção de uma estrutura em concreto armado, bastando para isso que seu usuário tenha o modelo da arquitetura no ambiente do Revit e defina os parâmetros que direcionam a concepção, de modo que a alteração dos mesmos conduz a diferentes alternativas de concepção.

7.1 Recomendações de pesquisas futuras

Se tratando de uma pesquisa cuja etapa principal de desenvolvimento envolve a definição de um algoritmo e sua programação, as recomendações de pesquisas futuras se apoiam no seu aprimoramento. Para isso, são feitas recomendações referentes a cada uma das etapas do algoritmo, apresentadas na seção 5.1, Visão geral da rotina, na Figura 5.4.

Quanto a parte referente a Leitura dos elementos do modelo arquitetônico, recomenda-se:

- Leitura de elementos arquitetônicos que definem restrições no lançamento de elementos estruturais, como portas e janelas, visando a compatibilização da estrutura lançada com a arquitetura existente;

- Leitura de elementos arquitetônicos como telhados, que conferem carga adicional a estrutura e devem ser considerados no pré-dimensionamento dos elementos estruturais.

Quanto a parte referente a Criação das geometrias de referência, recomenda-se:

- Aprimoramento do algoritmo de definição das linhas de localização dos pilares, a fim de permitir que o usuário defina valor de parâmetro que indique o espaçamento entre pilares lançados;
- Aprimoramento do algoritmo de definição das linhas de localização das vigas, a fim de permitir que o usuário defina valor de parâmetro que indique o vão máximo das vigas.

Quanto a parte referente a Pré-dimensionamento dos elementos estruturais, recomenda-se:

- Aprimoramento do critério de pré-dimensionamento das vigas, por meio da estimativa de carga em cada viga através da definição de área de influência;
- Aprimoramento do critério de pré-dimensionamento de pilares, de modo que a posição do pilar relativa à arquitetura seja considerada no seu pré-dimensionamento;

Quanto a parte referente a Criação dos elementos estruturais, recomenda-se:

- Aprimoramento no processo de criação dos elementos de sapata, de maneira que sapatas que venham a apresentar interseção sejam redimensionadas ou seja feita a adoção de sapata associada;
- Aprimoramento no processo de criação dos elementos de fundação, de maneira que a rotina permita a escolha por parte do usuário quanto à solução de fundação desejada;
- Desenvolvimento de algoritmo que permita a criação e pré-dimensionamento de lajes, permitindo ao usuário a escolha entre as diferentes soluções de laje;
- Desenvolvimento de algoritmo que permita ao usuário quanto a escolha pela criação de elementos de contenção.

REFERÊNCIAS

- ARAÚJO, J. de. *Curso de Concreto Armado - Volume 1*. Editora Dunas, 2003. ISBN 9788586717017. Disponível em: <https://books.google.com.br/books?id=ZdfVpM17i_AC>. Citado 3 vezes nas páginas 32, 34 e 36.
- ASBEA. *GUIA AsBEA BOAS PRÁTICAS EM BIM*: Estruturação do escritório de projeto para a implementação do bim. São Paulo, 2013. 20 p. Citado 3 vezes nas páginas 55, 56 e 57.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 6122*: Projeto e execução de fundações. Rio de Janeiro, 1996. 33 p. Citado 3 vezes nas páginas 45, 46 e 47.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 6118*: Projeto de estruturas de concreto — procedimento. Rio de Janeiro, 2014. 238 p. Citado 9 vezes nas páginas 23, 33, 34, 36, 37, 38, 40, 41 e 42.
- AUTODESK. *O Dynamo Primer*. 2019. Disponível em: <<https://primer.dynamobim.org/pt-br/>>. Acesso em: 25 sep. 2023. Citado 6 vezes nas páginas 60, 61, 62, 63, 64 e 65.
- AUTODESK. *Ajuda | Autodesk*. 2023. Disponível em: <<https://help.autodesk.com/view/RVT/2022/PTB/>>. Acesso em: 25 sep. 2023. Citado 3 vezes nas páginas 58, 59 e 60.
- AUTODESK. *VANTAGENS DA BIM*. 2023. Disponível em: <<https://www.autodesk.com.br/solutions/bim/benefits-of-bim>>. Acesso em: 23 sep. 2023. Citado 2 vezes nas páginas 52 e 55.
- BASTOS, P. S. *VIGAS DE CONCRETO ARMADO*. Universidade Estadual Paulista, 2017. Disponível em: <<https://wwwp.feb.unesp.br/pbastos/concreto2/Vigas.pdf>>. Citado na página 40.
- BASTOS, P. S. *Fundamentos do Concreto Armado*. Universidade Estadual Paulista, 2019. Disponível em: <<https://wwwp.feb.unesp.br/pbastos/concreto1/Fundamentos%20CA.pdf>>. Citado na página 35.
- BEER, F. P.; JOHNSTON, J. E. R. *Resistência dos Materiais*. [S.l.]: Pearson, 2008. Citado 2 vezes nas páginas 26 e 30.
- CARVALHO, R.; FILHO, J. de F. *Cálculo e detalhamento de estruturas usuais de concreto armado*. EdUFSCar, 2014. ISBN 9788576003564. Disponível em: <https://books.google.com.br/books?id=ljP_zwEACAAJ>. Citado 5 vezes nas páginas 22, 32, 34, 35 e 36.
- CUNHA, J. *Estruturas de Concreto Armado*. [S.l.]: Universidade Federal de Uberlândia, 2014. 198 p. Citado 14 vezes nas páginas 21, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 49, 50 e 51.
- EASTMAN, C. et al. *Manual de BIM: Um Guia de Modelagem da Informação da Construção para Arquitetos, Engenheiros, Gerentes, Construtores e Incorporadores*. 1. ed. [S.l.]: Bookman Editora, 2014. ISBN 9788582601174. Citado 6 vezes nas páginas 52, 53, 54, 55, 56 e 57.
- HELENE, P. *Estrutura da obra é 30% do custo e 100% da segurança*. Cimento Itambe, 2019. Reportagem com base na palestra “Aprendendo com acidentes”, do professor-doutor da Poli-USP Paulo Helene, dentro do 4º Simposio Paranaense de Patologia

das Construcoes. Disponível em: <<https://www.cimentoitambe.com.br/massa-cinzena/estrutura-da-obra-e-30-do-custo-e-100-da-seguranca/>>. Citado na página 15.

MARTHA, L. *Análise de Estruturas: Conceitos e Métodos Básicos*. Elsevier Brasil, 2010. ISBN 9788535210828. Disponível em: <<https://books.google.com.br/books?id=fn8oEavmGnUC>>. Citado 7 vezes nas páginas 15, 20, 21, 22, 23, 24 e 29.

NETTO, C. C. *Autodesk Revit Architecture 2020*. Saraiva Educação S.A., 2020. ISBN 9788536532912. Disponível em: <<https://books.google.com.br/books?id=urfZDwAAQBAJ>>. Citado 5 vezes nas páginas 52, 53, 58, 59 e 60.

QUEIROZ, F. *Guarde Bem: Projeto arquitetônico*. 2023. Citado na página 110.

REBELLO, Y. C. P. *A concepção estrutural e a arquitetura*. [S.l.]: Zigurate, 2000. Citado 21 vezes nas páginas 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 34, 35, 38, 39, 41, 42, 49, 50 e 51.

REBELLO, Y. C. P. *Fundacoes: Guia pratico de projeto, execucao e dimensionamento*. [S.l.]: Zigurate, 2008. Citado 2 vezes nas páginas 48 e 49.

SACKS, R. et al. *Manual de BIM - 3.ed.: Um Guia de Modelagem da Informação da Construção para Arquitetos, Engenheiros, Gerentes, Construtores e Incorporadores*. Bookman Editora, 2021. ISBN 9788582605523. Disponível em: <<https://books.google.com.br/books?id=e6ggEAAAQBAJ>>. Citado na página 15.

SARMENTO, F. J. *Engenharia Civil e Ambiental: Uma abordagem computacional (Programacao com Visual Basic Livro 1)*. [S.l.]: Francisco Jacome Sarmiento, 2020. ISBN 978-85-924138-2-8. Citado na página 15.

SENA, P. C. P. de. *Automação de processos de projeto e programação em BIM: Dynamo, Python e C#*. 203 p. Dissertação (Mestrado em Arquitetura e Urbanismo) — Instituto de Arquitetura e Urbanismo, Universidade de São Paulo, São Carlos, 2019. Citado na página 15.

SIENGE. *Custo da obra por etapa: conheça os valores e planeje-se*. 2022. Disponível em: <<https://www.sienge.com.br/blog/custo-da-obra-por-etapa/>>. Acesso em: 17 out. 2023. Citado na página 15.

SOARES, F. L. *Fundacoes Diretas*. 2021. Notas de Aula. Citado 3 vezes nas páginas 47, 48 e 49.

SUSSEKIND, J. *Curso de analise estrutural*. Globo, 1981. ISBN 9788525002266. Disponível em: <<https://books.google.com.br/books?id=X7YRtwAACAAJ>>. Citado 4 vezes nas páginas 20, 22, 24 e 25.

TECNCON - ENGENHARIA ESTRUTURAL. *Guarde Bem: Projeto estrutural*. 2023. Citado na página 110.

TEIXEIRA, A. H.; GODOY, N. S. de. Análise, projeto e execução de fundações rasas. In: HACHICH, W. et al. (Ed.). *Fundações: Teoria e Prática*. 2ª. ed. São Paulo: Pini, 1998. cap. 7, p. 227–264. ISBN 9788572660983. Citado 4 vezes nas páginas 46, 47, 48 e 49.

TIMOSHENKO, S. *Meânica dos solidos*. [S.l.]: Livros Técnicos e Científicos, 1983. v. 1. Citado 5 vezes nas páginas 27, 28, 29, 30 e 31.

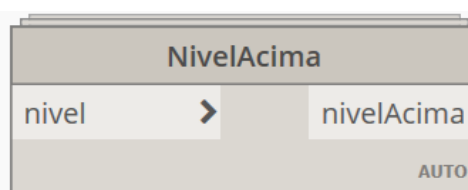
TIMOSHENKO, S. *Meânica dos solidos*. [S.l.]: Livros Técnicos e Científicos, 1984. v. 2. Citado 2 vezes nas páginas 26 e 27.

VELLOSO, D.; LOPES, F. de R. Concepção de obras de fundações. In: HACHICH, W. et al. (Ed.). *Fundações: Teoria e Prática*. 2ª. ed. São Paulo: Pini, 1998. cap. 6, p. 211–226. ISBN 9788572660983. Citado 2 vezes nas páginas 45 e 46.

APÊNDICE A – DESENVOLVIMENTO DO NÓ NIVELACIMA

Esse capítulo tem por propósito explicar as partes componentes do nó NivelAcima e trazer os resultados obtidos da sua execução. No ambiente do Dynamo, esse nó é como exibido na Figura A.1.

Figura A.1 – Nó NivelAcima

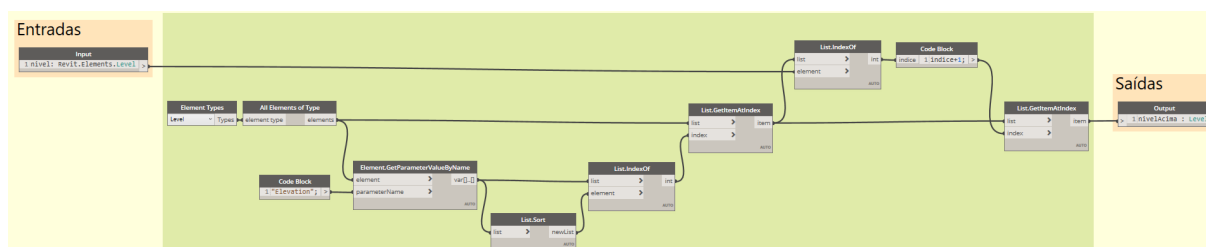


Fonte: Autor

O nó NivelAcima retorna o nível do projeto imediatamente acima daquele passado na sua entrada nivel.

A Figura A.2 fornece uma visão geral do conteúdo do nó, que determina suas entradas, o processamento dos dados e as saídas.

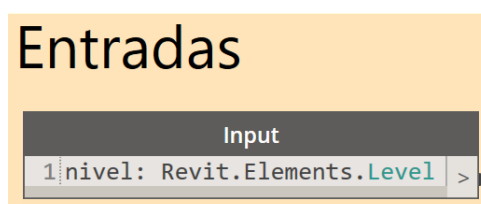
Figura A.2 – Visão geral do nó NivelAcima



Fonte: Autor

As figuras A.3, A.4 e A.5 contêm as partes do nó. Abaixo de cada figura são fornecidos comentários que elucidam o algoritmo desenvolvido.

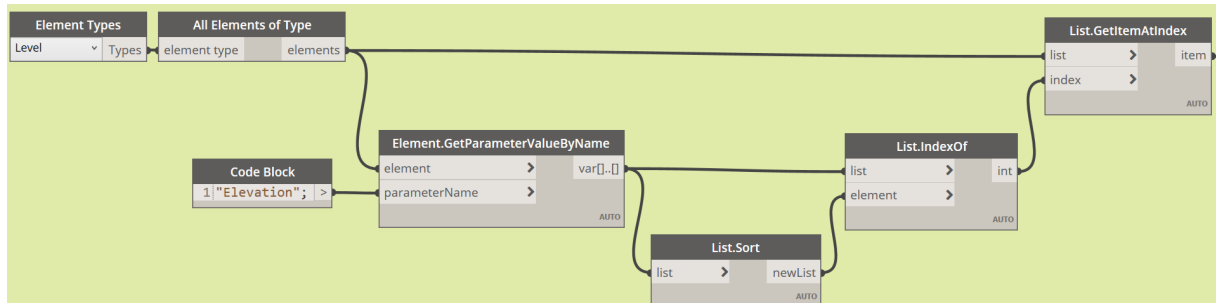
Figura A.3 – Nó NivelAcima (parte 1)



Fonte: Autor

- O nó *Input* define a entrada do nó *NivelAcima*. É fornecido o nome da entrada (nível) e o tipo do valor que deve ser passado (*Level*).

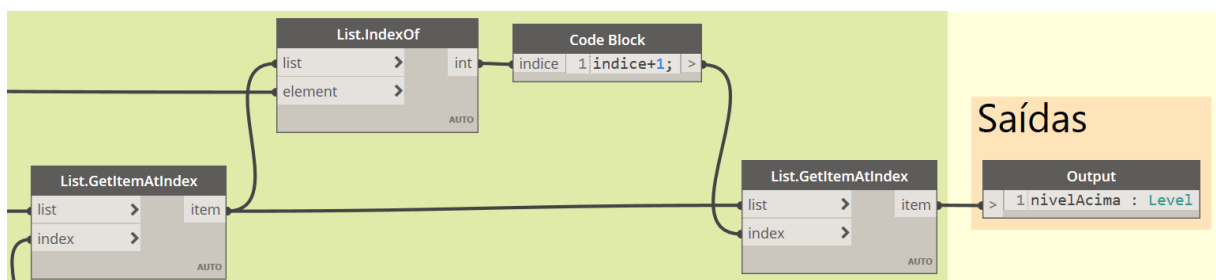
Figura A.4 – Nó NivelAcima (parte 2)



Fonte: Autor

- Os nós *Element Types* e *All Elements of Type* fazem a seleção de todos os níveis do projeto.
- São tomadas as elevações dos níveis através dos nós *Element.GetParameterValueByName* e *Code Block*. A elevação de um nível é armazenada no seu parâmetro *Elevation*.
- O nó *List.Sort* ordena de maneira crescente os valores de elevação obtidos.
- Através dos nós *List.IndexOf* e *List.GetItemAtIndex* os níveis do projeto, retornados como saída do nó *All Elements of Type*, são ordenados de maneira crescente de acordo com o valor da sua elevação.

Figura A.5 – Nó NivelAcima (parte 3)



Fonte: Autor

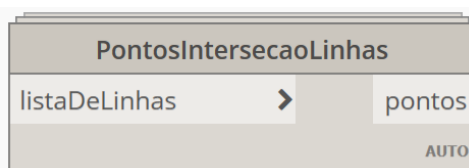
- O nó *List.GetItemAtIndex* à esquerda retorna uma lista com os níveis do projeto em ordem crescente de acordo com o valor da sua elevação.
- O nó *List.IndexOf* retorna o índice do elemento *element* na lista *list*. O nó conectado em *element* é o nível tomado como entrada no nó *Input* (Figura A.3).

- O *Code Block* adiciona uma unidade ao valor do índice. Em seguida, toma-se na lista de níveis ordenados o nível correspondente ao índice dado pelo *Code Block*, através do nó *List.GetItemAtIndex* à direita.
- O nível retornado do nó *List.GetItemAtIndex* é passado como saída do nó *NivelAcima* através do nó *Output*, que define o nome da saída exibida (*nivelAcima*), como exibido na Figura A.1.

APÊNDICE B – DESENVOLVIMENTO DO NÓ PONTOSINTERSECAOLINHAS

Esse capítulo tem por propósito explicar as partes componentes do nó PontosIntersecaoLinhas e trazer os resultados obtidos da sua execução. No ambiente do Dynamo, esse nó é como exibido na Figura B.1.

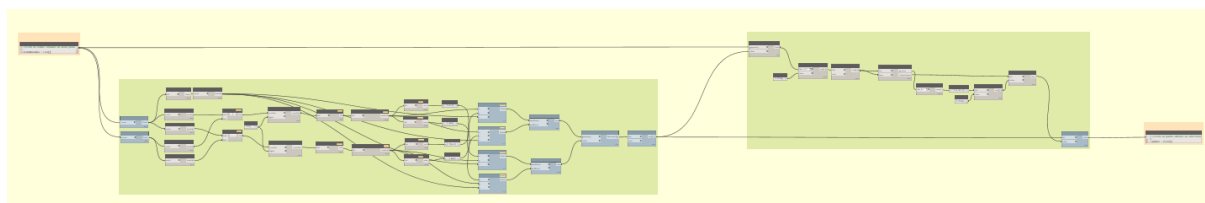
Figura B.1 – Nó PontosIntersecaoLinhas



Fonte: Autor

A Figura B.2 fornece uma visão geral do conteúdo do nó PontosIntersecaoLinhas, que determina suas entradas, o processamento dos dados e as saídas.

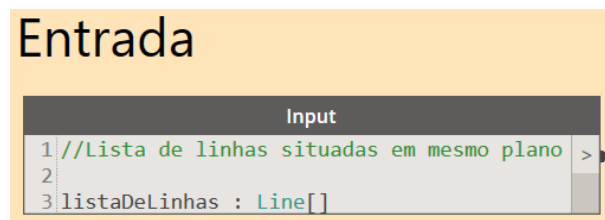
Figura B.2 – Visão geral do nó PontosIntersecaoLinhas



Fonte: Autor

As figuras B.3, B.5, B.7, B.8, B.10, B.13, B.15 e B.16 contêm as partes do nó. Abaixo de cada figura são fornecidos comentários que elucidam o algoritmo desenvolvido.

Figura B.3 – Nó PontosIntersecaoLinhas (parte 1)

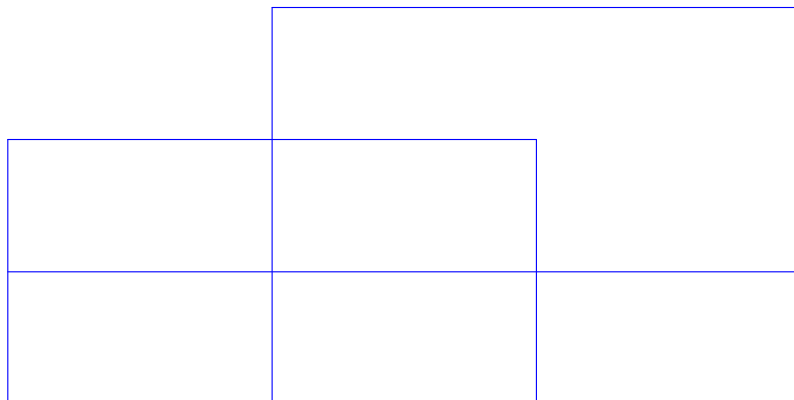


Fonte: Autor

- O nó *Input* define a entrada do nó PontosIntersecaoLinhas. É fornecido o nome da entrada do nó (listaDeLinhas), o tipo do valor que deve ser passado (lista de objetos do tipo *Line*) e um comentário.

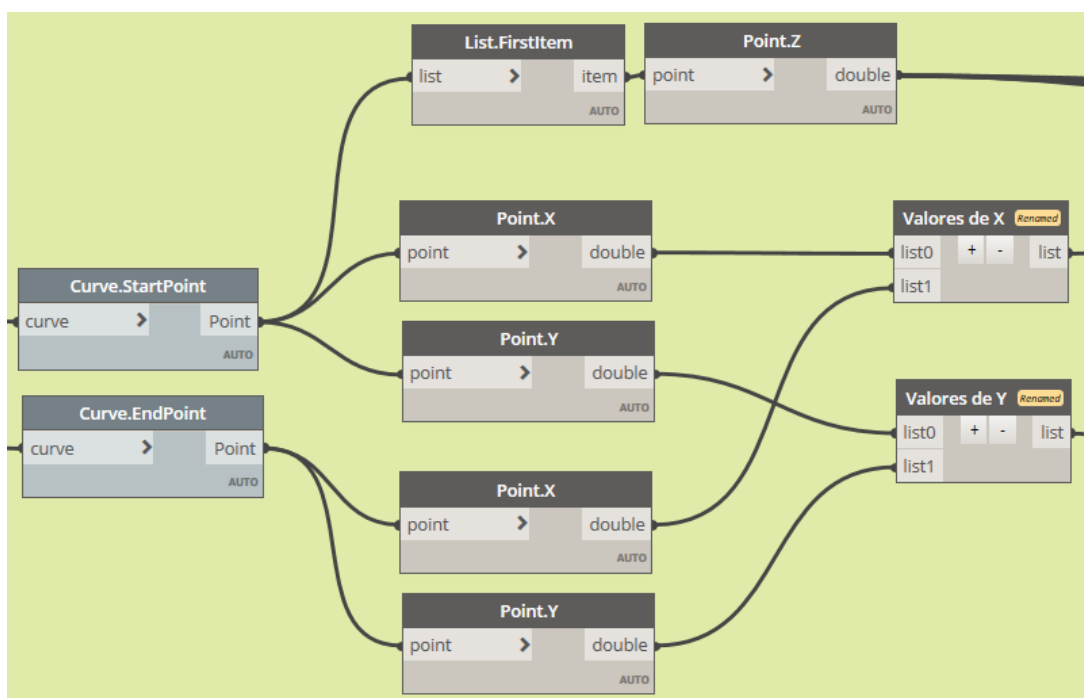
A Figura B.4 exibe um conjunto de linhas tomado para exemplo e ilustração das etapas do algoritmo. É esse conjunto de linhas o passado na entrada do nó em questão.

Figura B.4 – Lista de linhas passada como entrada



Fonte: Autor

Figura B.5 – Nó PontosIntersecaoLinhas (parte 2)



Fonte: Autor

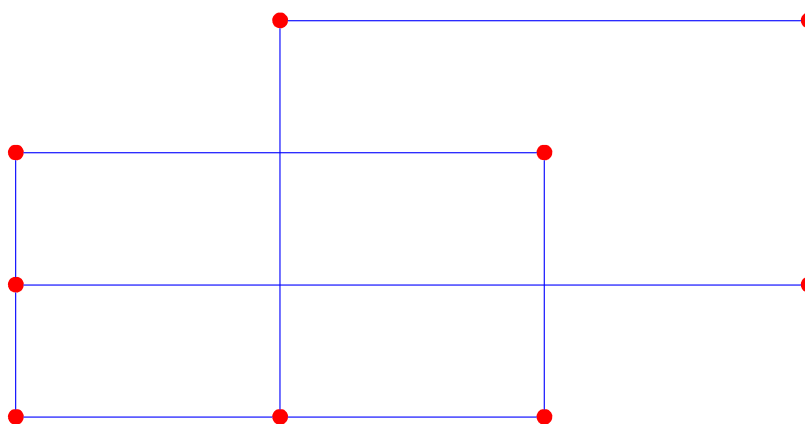
- Da lista passada na entrada do nó são extraídos os pontos de início e ponto de fim de cada linha, através dos nós *Curve.StartPoint* e *Curve.EndPoint* respectivamente.
- Desses pontos são extraídas as coordenadas x, y e z, através dos nós *Point.X*, *Point.Y* e *Point.Z*, respectivamente. É tomada a coordenada z de apenas um dos pontos dos fornecidos na lista, o que é garantido pelo nó *List.FirstItem*, tendo em vista que se assume que

a lista de linhas passada na entrada seja coplanar, levando todos os pontos a ter mesma coordenada z.

- Os valores de x e y são agrupados em listas únicas por meio dos nós “Valores de X” e “Valores de Y”, que tratam de nós *List.Join* que foram renomeados.

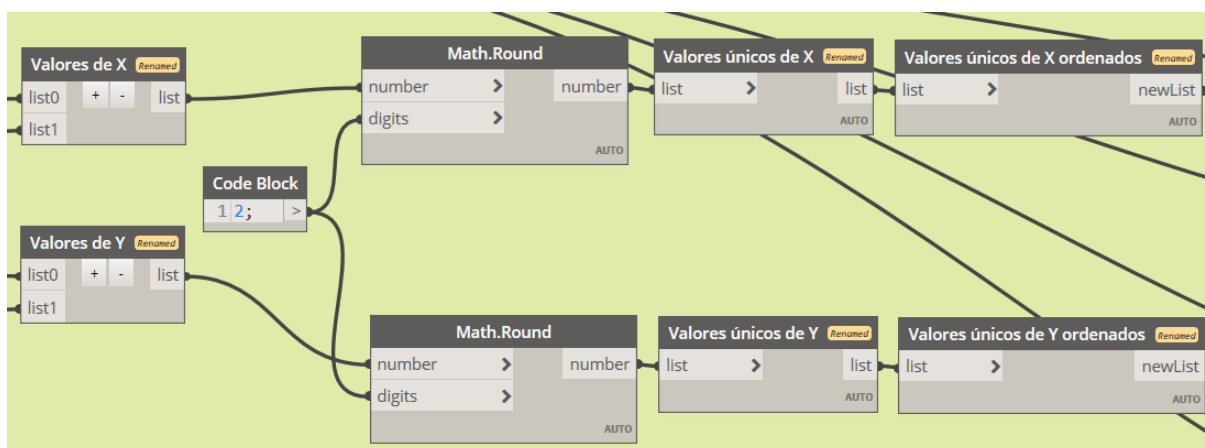
A seleção dos pontos iniciais e finais de cada linha é ilustrada na Figura B.6.

Figura B.6 – Seleção de pontos iniciais e finais



Fonte: Autor

Figura B.7 – Nó PontosIntersecaoLinhas (parte 3)

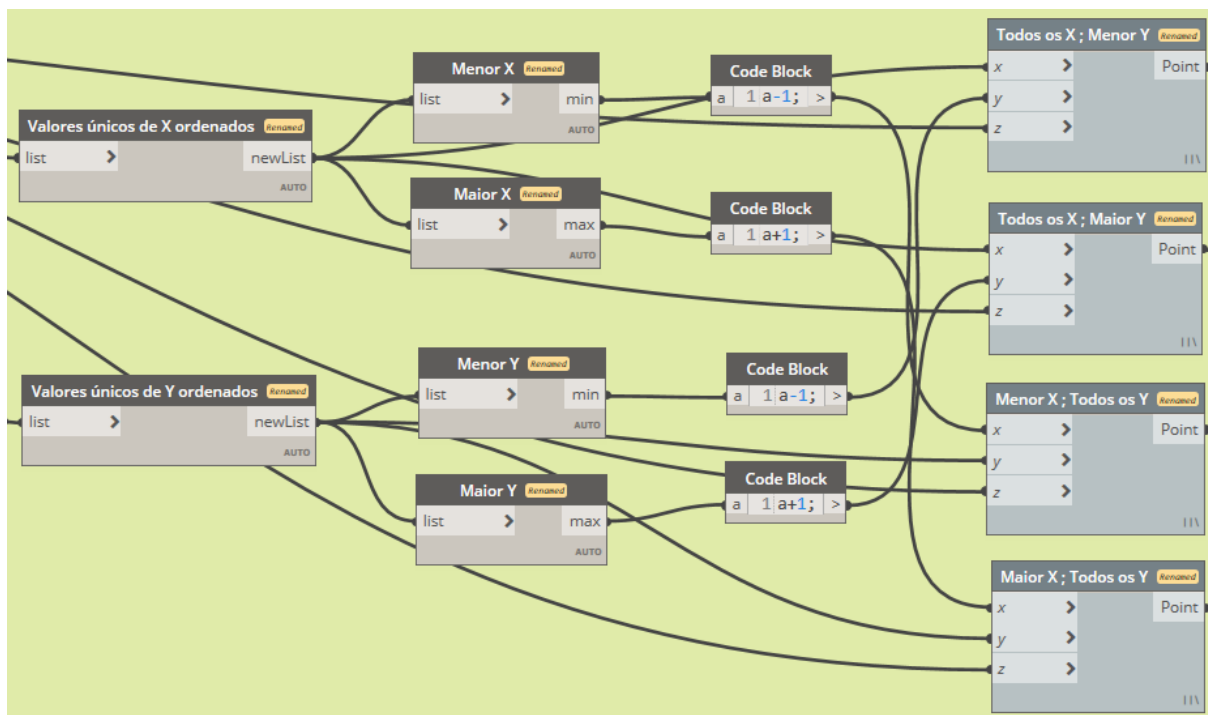


Fonte: Autor

- Através do nó *Math.Round* as listas com os valores de x e y tem seus valores arredondados para duas casas decimais, o que garante que eventuais diferenças entre os valores, consideradas desprezíveis, sejam ignoradas.
- As listas são reduzidas a seus valores únicos através dos nós *List.UniqueItems*, aqui chamados de “Valores únicos de X” e “Valores únicos de Y”.

- As listas de valores únicos são ordenadas de maneira crescente através do nó *List.Sort*, aqui chamado de “Valores únicos de X ordenados” e “Valores únicos de Y ordenados”.

Figura B.8 – Nó PontosIntersecaoLinhas (parte 4)

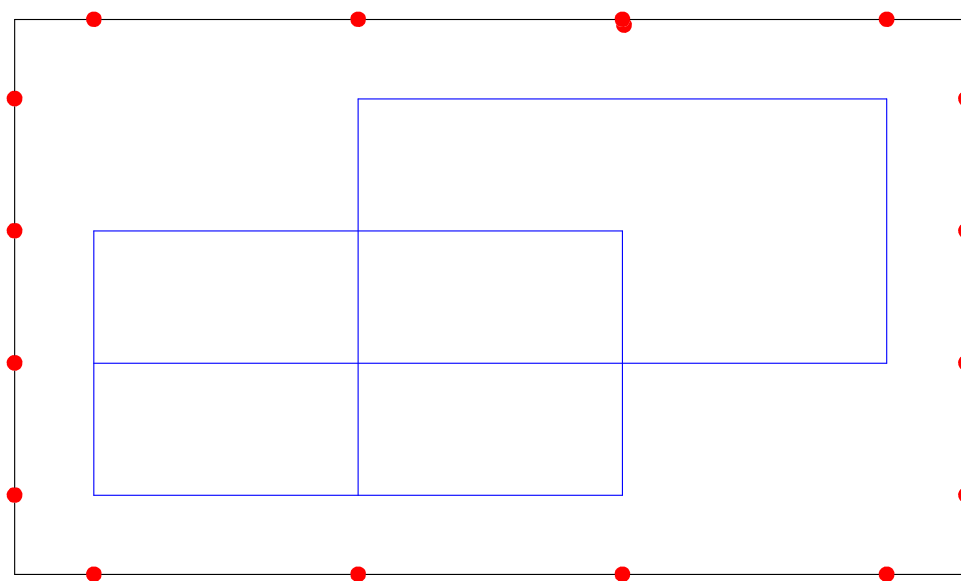


Fonte: Autor

- Dos nós “Valores únicos de X ordenados” e “Valores únicos de Y ordenados” são extraídos o menor e maior valor, por meio dos nós *List.MinimumItem* e *List.MaximumItem*, chamados de “Menor X”, “Maior X”, “Menor Y” e “Maior Y”.
- Do cruzamento dos valores de x com o mínimo e máximo valor de y e dos valores de y com o mínimo e máximo de x é obtido um conjunto de pontos com essas coordenadas. Para todos esses pontos é feito uso da coordenada z extraída no nó *Point.Z* (Figura B.5). A criação dos pontos é feita por meio de nós *Point.ByCoordinates*, renomeados para “Todos os X; Menor Y”, “Todos os X; Maior Y”, “Menor X; Todos os Y” e “Maior X; Todos os Y”.
- Observar que dos valores mínimos e máximos extraídos são retirados e adicionados uma unidade respectivamente, por meio de nós *Code Block*.

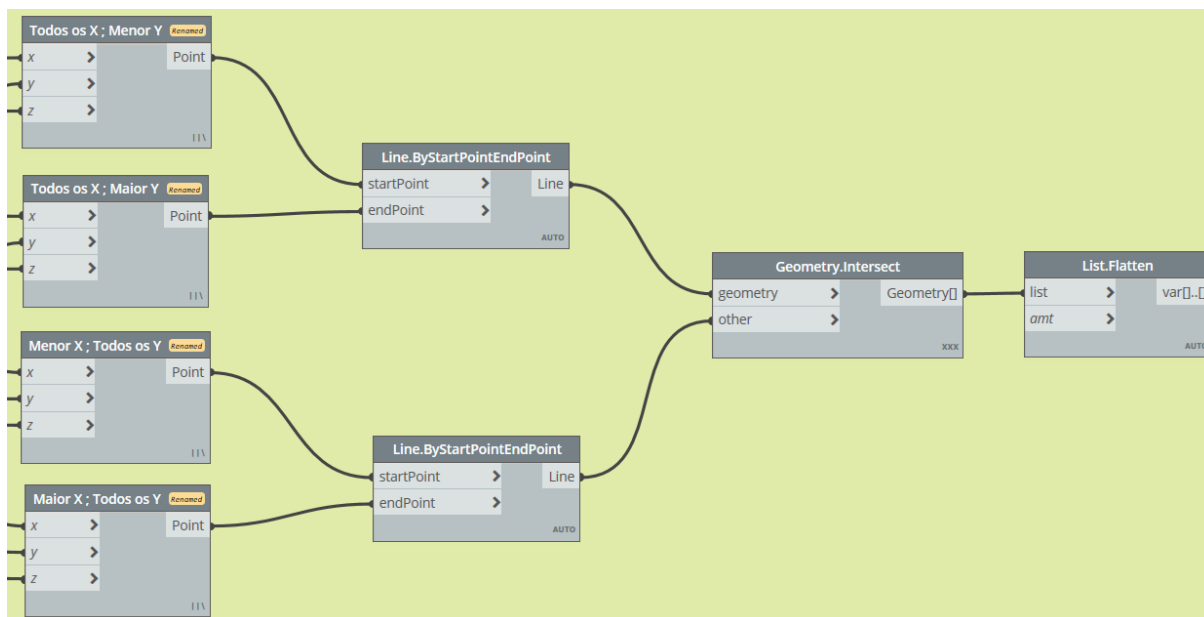
Os pontos obtidos são ilustrados na Figura B.9. Esses pontos consistem da projeção dos pontos iniciais e finais nas coordenadas mínimas e máximas, nas duas direções. Vale destacar que a adição e remoção de uma unidade dos valores mínimos e máximos foi feita para garantir o afastamento do conjunto de pontos das linhas originais.

Figura B.9 – Pontos obtidos por projeção



Fonte: Autor

Figura B.10 – Nó PontosIntersecaoLinhas (parte 5)



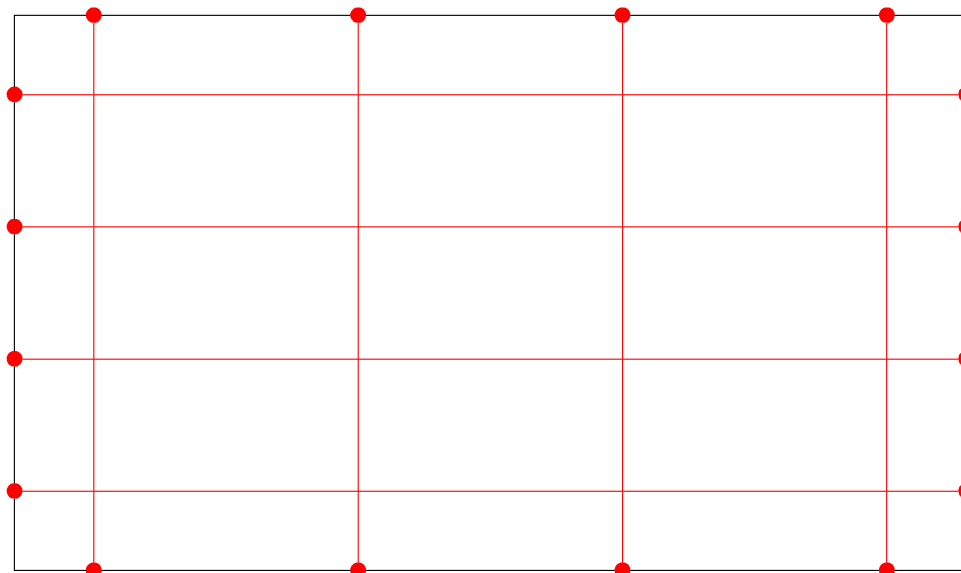
Fonte: Autor

- Por meio de nós *Line.ByStartPointEndPoint* são traçados dois conjuntos de linhas. O primeiro a partir dos pontos gerados pelos nós “Todos os X; Menor Y” e “Todos os X; Maior Y”. O segundo a partir dos pontos gerados pelos nós “Menor X, Todos os Y” e “Maior X; Todos os Y”.
- O nó *Geometry.Intersect* retorna os pontos obtidos a partir da interseção dos dois conjuntos de linhas gerados pelos nós *Line.ByStartPointEndPoint*.

- O nó *List.Flatten* aplaina a lista de pontos criada pelo nó *Geometry.Intersect*.

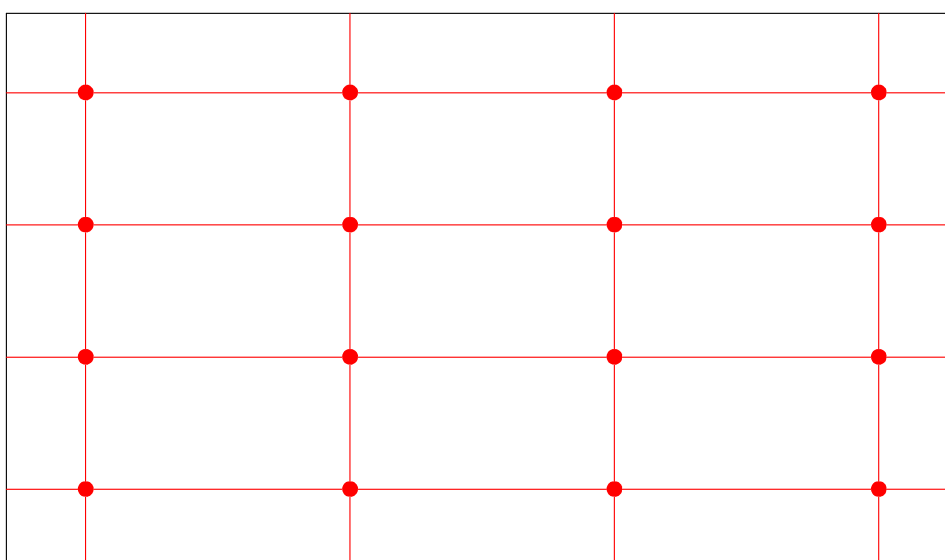
As linhas traçadas são representadas na Figura B.11. Os pontos obtidos da interseção dessas linhas são ilustrados na Figura B.12.

Figura B.11 – Linhas traçadas



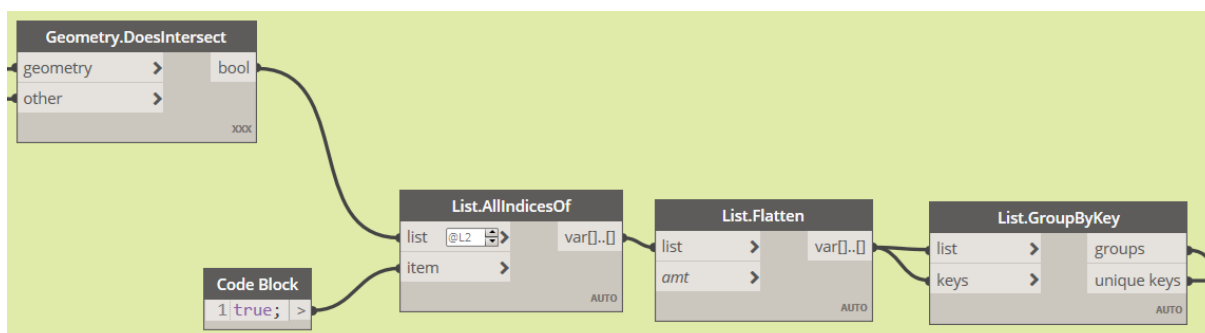
Fonte: Autor

Figura B.12 – Interseção das linhas traçadas



Fonte: Autor

Figura B.13 – Nó PontosIntersecaoLinhas (parte 6)

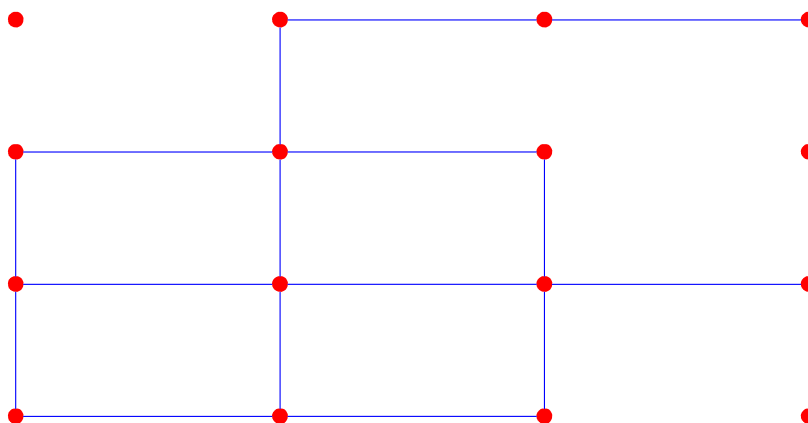


Fonte: Autor

- O nó *Geometry.DoesIntersect* verifica a interseção entre duas geometrias, retornando o booleano *true* quando há interseção e *false* quando não há. Essa interseção é verificada entre as linhas passadas como entrada na Figura B.3 e os pontos obtidos do nó *List.Flatten* exibido na Figura B.10, passados nas entradas *geometry* e *other* respectivamente. Esse nó produz uma sublista para cada linha.
- Por meio do nó *List.AllIndicesOf* obtém-se os índices dos valores *true* nas sublists criadas no nó *Geometry.DoesIntersect*. O mesmo índice em sublists diferentes corresponde ao mesmo ponto.
- A lista de índices criado pelo nó *List.AllIndicesOf* é aplainada através do nó *List.Flatten*. Da lista obtida são criadas sublists que agrupam os mesmos índices em listas separadas através do nó *List.GroupByKey*.

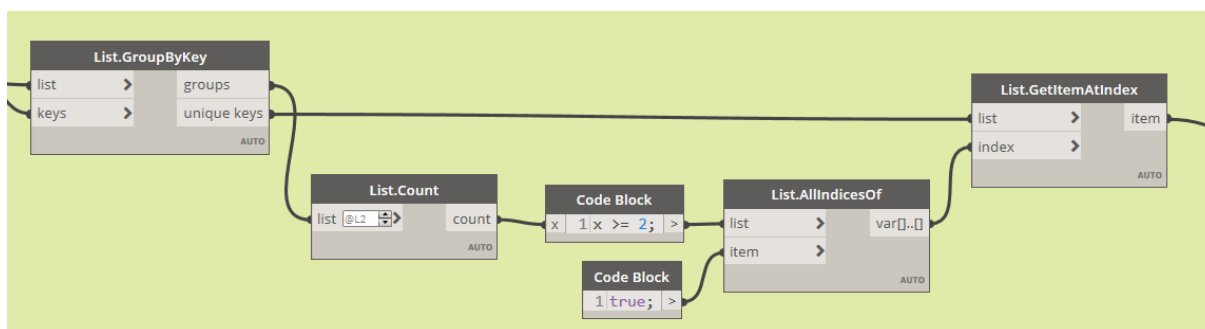
Os pontos e linhas entre os quais vai se verificar a interseção são os ilustrados na Figura B.14.

Figura B.14 – Verificação da interseção entre pontos e linhas



Fonte: Autor

Figura B.15 – Nó PontosIntersecaoLinhas (parte 7)

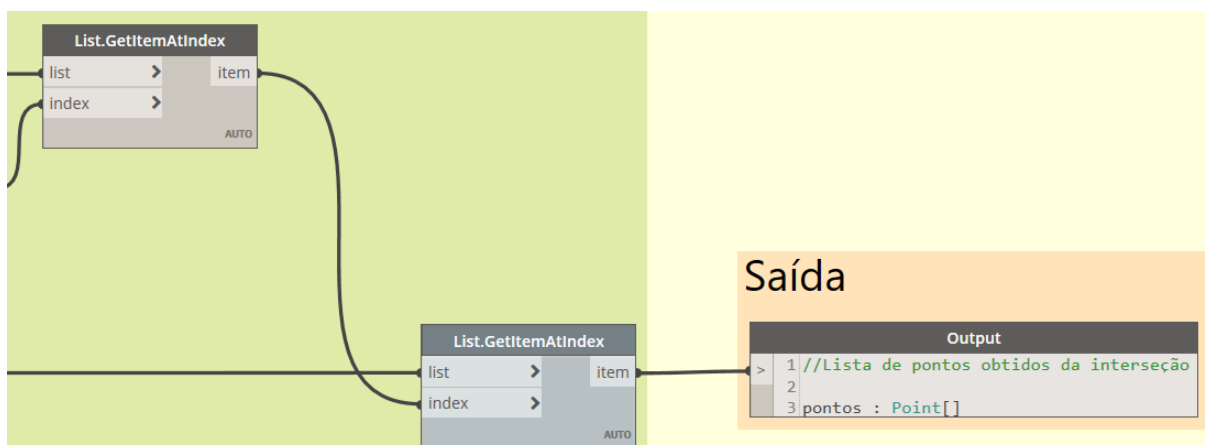


Fonte: Autor

- As sublistas obtidas do nó *List.GroupByKey* agrupam os mesmos índices em listas separadas. Como cada índice corresponde a um ponto passado na entrada *other* do nó *Geometry.DoesIntersect* (Figura B.13), o número de itens da sublista de um índice indica em quantas linhas aquele ponto faz interseção.
- Por meio do nó *List.Count* faz-se a contagem dos elementos em cada sublista.
- Verificam-se as listas com dois ou mais itens através do *Code Block* e do nó *List.AllIndicesOf*.
- Ao fim, dos índices produzidos na saída *unique keys* no nó *List.GroupByKey*, são filtrados aqueles cujas sublistas correspondentes possuem dois ou mais elementos por meio do nó *List.GetItemAtIndex*.
- Dessa forma, os índices obtidos nesse processo são aqueles correspondentes aos pontos gerados do nó *List.Flatten* (Figura B.10) que fazem duas ou mais interseções com as linhas passadas como entrada na Figura B.3.

Dessa forma, o trecho de código retratado na Figura B.15 tem por finalidade verificar quais dos pontos exibidos na Figura B.14 interseccionam ao menos duas linhas.

Figura B.16 – Nó PontosIntersecaoLinhas (parte 8)

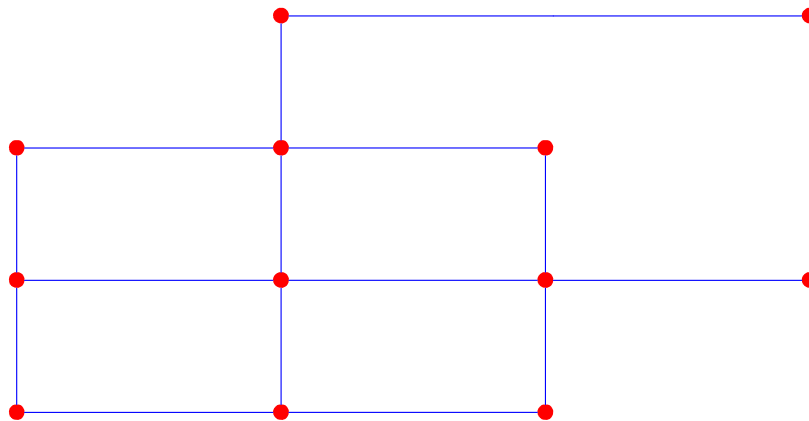


Fonte: Autor

- Esses pontos são acessados através do nó *List.GetItemAtIndex* passando como entrada *index* os índices dos pontos e *list* a lista de todos os pontos, gerada em *List.Flatten* (Figura B.10).
- Essa lista de ponto é a saída produzido pelo nó *PontosIntersecaoLinhas* através do nó *Output*.
- O nó *Output* define o nome da saída (pontos), o tipo de dado retornado (lista de objetos do tipo *Point*) e um comentário.

Ao final do nó desenvolvido, serão retornados os pontos resultantes da interseção das linhas, como retratado na Figura B.17.

Figura B.17 – Pontos resultantes da interseção das linhas

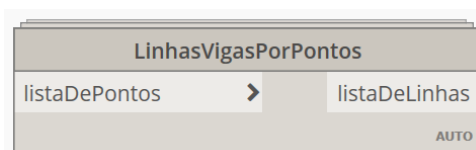


Fonte: Autor

APÊNDICE C – DESENVOLVIMENTO DO NÓ LINHASVIGASPORPONTOS

Esse capítulo tem por propósito explicar as partes componentes do nó LinhasVigasPorPontos e trazer os resultados obtidos da sua execução. No ambiente do Dynamo, esse nó é como exibido na Figura C.1.

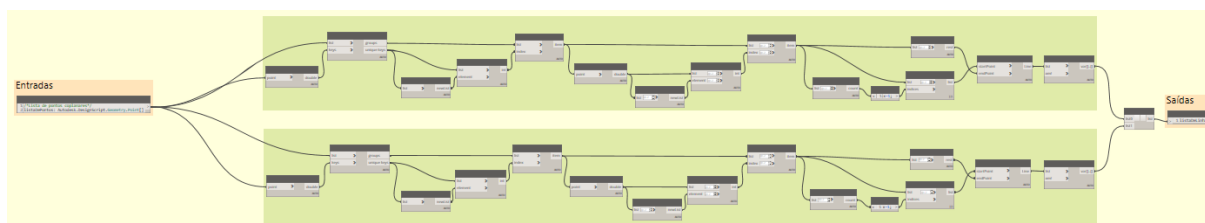
Figura C.1 – Nó LinhasVigasPorPontos



Fonte: Autor

A Figura C.2 fornece uma visão geral do conteúdo do nó, que determina suas entradas, o processamento dos dados e as saídas.

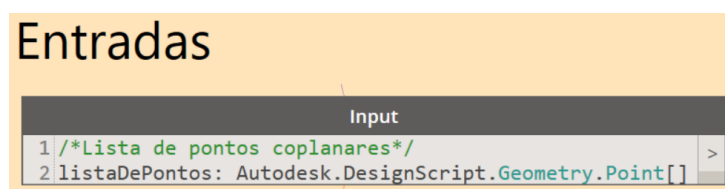
Figura C.2 – Visão geral do nó LinhasVigasPorPontos



Fonte: Autor

As figuras C.3, C.5, C.7, C.9 e C.11 contêm as partes do nó. Abaixo de cada figura são fornecidos comentários que elucidam o algoritmo desenvolvido.

Figura C.3 – Nó LinhasVigasPorPontos (parte 1)

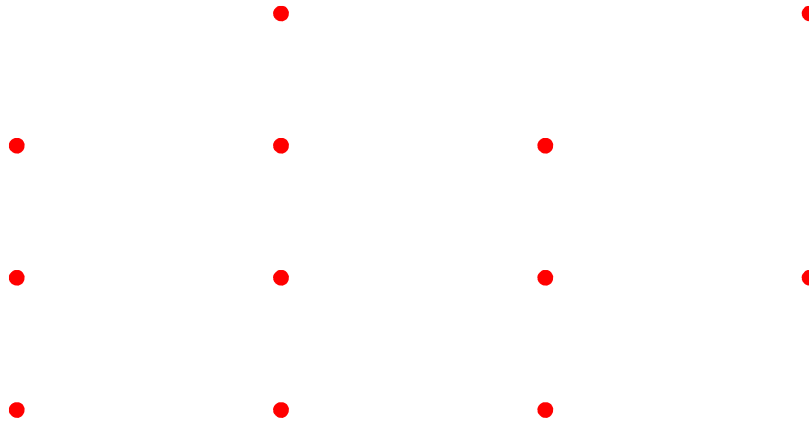


Fonte: Autor

- O nó *Input* define a entrada do nó *LinhasVigasPorPontos*. É fornecido o nome da entrada (*listaDePontos*), o tipo do valor que deve ser passado (lista de *Point*) e um comentário.

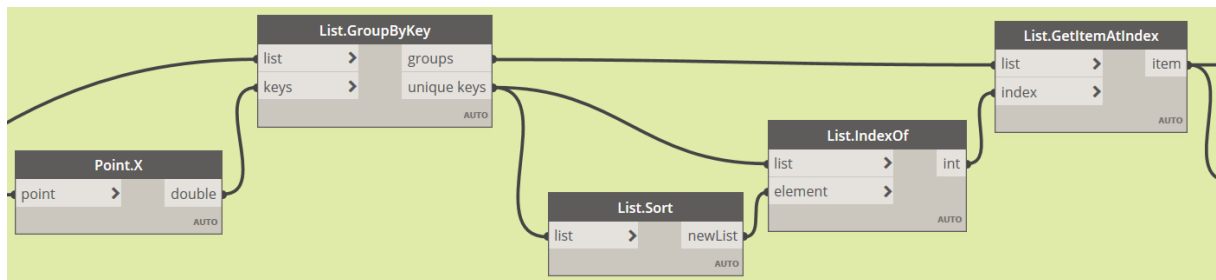
A Figura C.4 apresenta um conjunto de pontos que será tomado para ilustrar os resultados intermediários da execução da rotina. É esse conjunto de pontos o passado na entrada do nó em questão.

Figura C.4 – Lista de pontos passada como entrada



Fonte: Autor

Figura C.5 – Nó LinhasVigasPorPontos (parte 2)



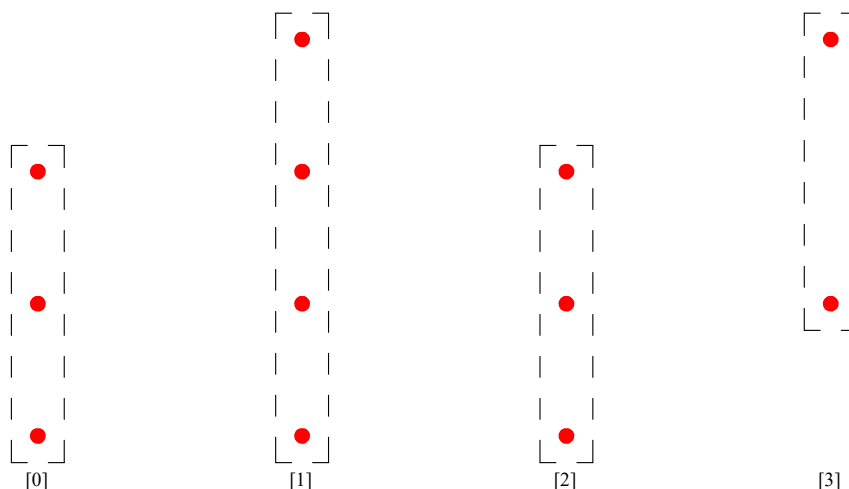
Fonte: Autor

- Da lista de pontos passada como entrada do nó são extraídos os valores da coordenada x, através do nó *Point.X*.
- Os valores de x dos pontos são tomados para agrupá-los em listas, que isolam pontos com a mesma coordenada x. Esse processo é feito por meio do nó *List.GroupByKey*.
- Os valores retornados pelo nó *List.GroupByKey* são as listas de pontos, na saída *groups*, e os valores únicos de x que correspondem a cada uma das listas, na saída *unique keys*. Tais valores únicos de x são ordenados de forma crescente, por meio do nó *List.Sort*.
- Com o nó *List.IndexOf* e *List.GetItemAtIndex* obtém-se as listas da saída *groups* ordenadas de maneira crescente em função do valor da coordenada x dos pontos que armazenam.

O resultado obtido até o momento é ilustrado na figura Figura C.6. O pontilhado preto representa as listas criadas. O valor sob o pontilhado é o índice de cada lista. Vale observar que

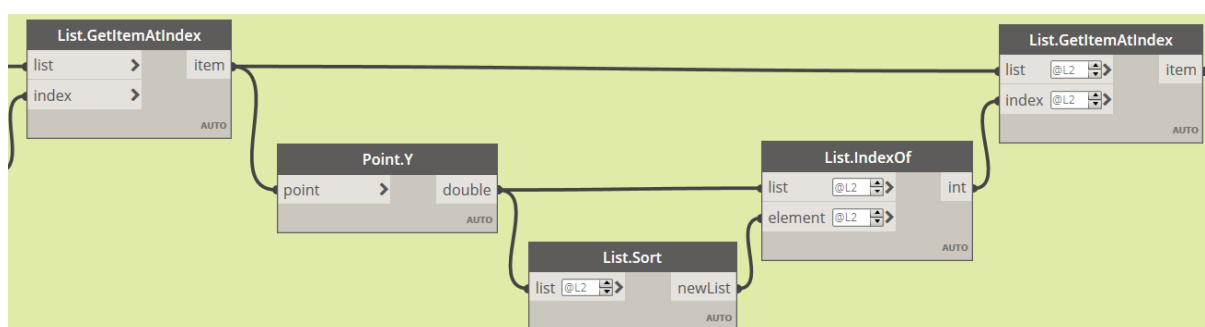
os índices são dispostos em ordem crescente em função do valor da coordenada x dos pontos armazenados em cada lista.

Figura C.6 – Pontos agrupados em listas



Fonte: Autor

Figura C.7 – Nó LinhasVigasPorPontos (parte 3)

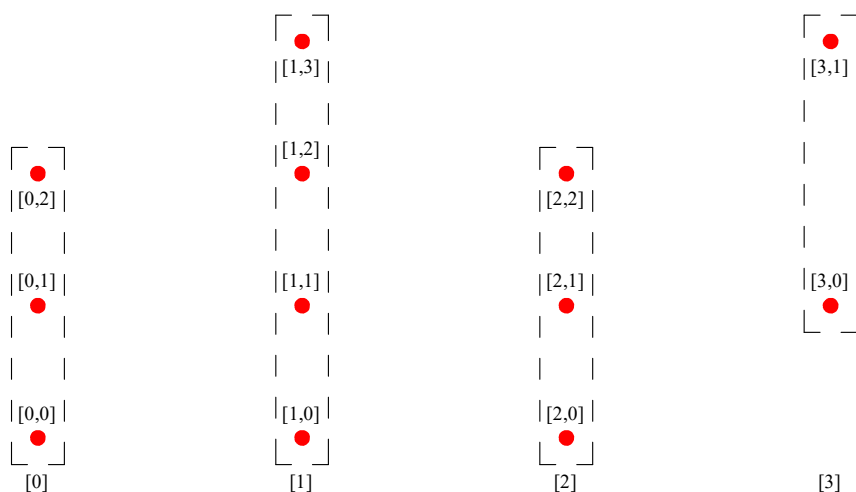


Fonte: Autor

- Das listas ordenadas, geradas no nó *List.GetItemAtIndex*, são tomados os valores de y dos pontos que armazenam, por meio do nó *Point.Y*.
- É feita então a ordenação dos pontos de cada lista em ordem crescente da coordenada y. Para isso, faz-se uso dos nós *List.Sort*, *List.IndexOf* e *List.GetItemAtIndex*, de maneira semelhante ao realizado na Figura C.5.

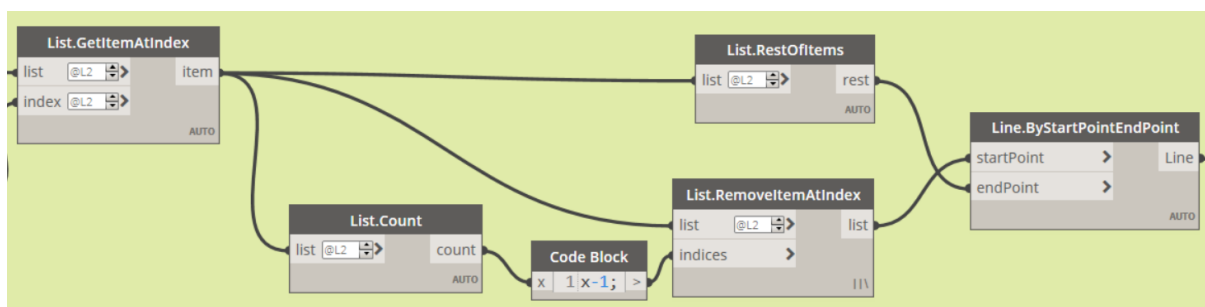
Nesse momento, tem-se a ordenação dos pontos de cada lista gerada, de maneira que os índices em cada lista são atribuídos de maneira crescente com o valor de y, como é ilustrado na Figura C.8.

Figura C.8 – Ordenação dos pontos das listas



Fonte: Autor

Figura C.9 – Nó LinhasVigasPorPontos (parte 4)

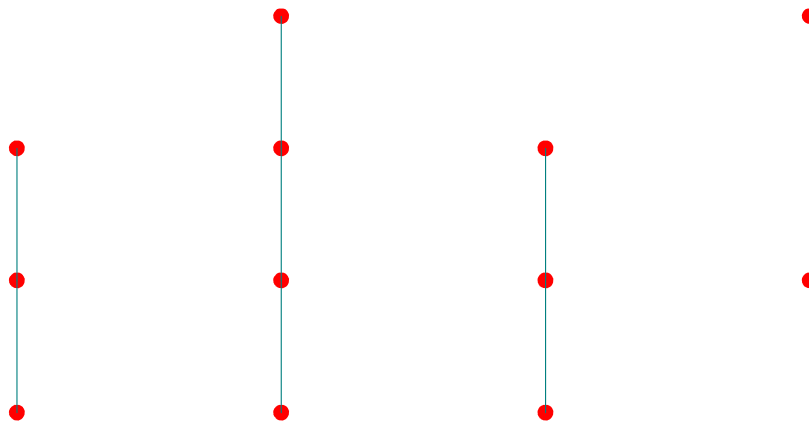


Fonte: Autor

- Das listas ordenadas segundo os valores das coordenadas x e y dos pontos que armazenam é removido o primeiro item de cada lista, por meio do nó *List.RestOfItems*.
- Também é feita remoção do último item de cada lista, por meio dos nós *List.Count*, *Code Block* e *List.RemoveItemAtIndex*.
- Dos novos conjuntos de listas, sem o primeiro e último elemento, são traçadas linhas, tomando esses conjuntos como ponto de início e fim das linhas geradas, por meio do nó *Line.ByStartPointEndPoint*.

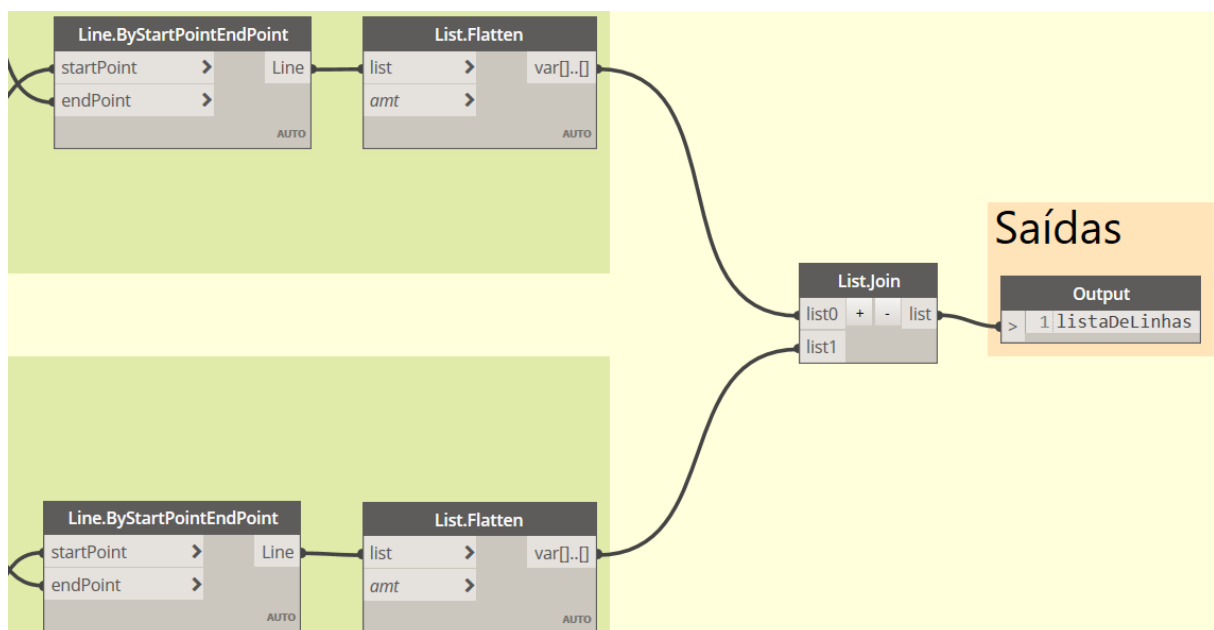
O resultado obtido é apresentado na Figura C.10. Observa-se que o trecho de código apresentado na Figura C.9 se encarrega de criar linhas que unem pontos consecutivos de cada lista.

Figura C.10 – Criação de linhas nas listas



Fonte: Autor

Figura C.11 – Nó LinhasVigasPorPontos (parte 5)



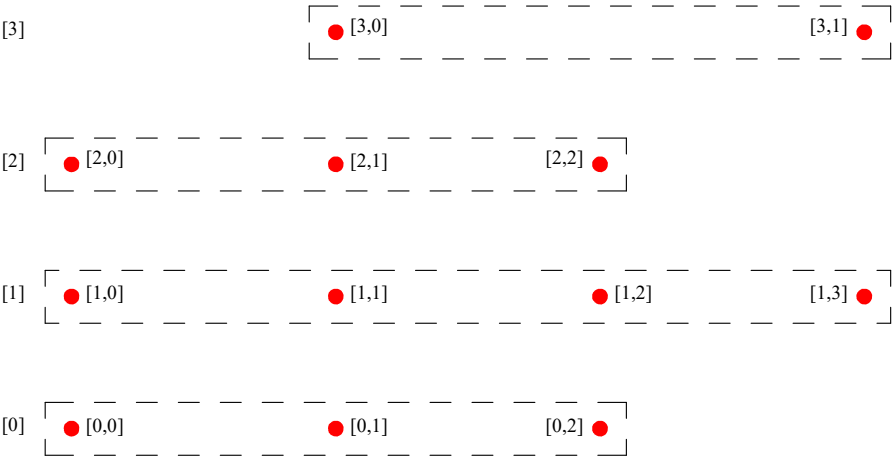
Fonte: Autor

- O procedimento que teve início na Figura C.5 é repetido agrupando os pontos em listas cujos pontos têm mesma coordenada y e em seguida são ordenados em valor crescente da coordenada x (Figura C.12). Por fim são traçadas linhas que unem dois pontos consecutivos dentro de cada lista, resultado do nó *Line.ByStartPointEndPoint* exibido na parte inferior.
- As listas de linhas produzidas pelos dois nós *Line.ByStartPointEndPoint* são aplainadas por meio dos nós *List.Flatten* e em seguida unidas em uma só por meio do nó *List.Join*.

- Por fim, a lista final produzida é passada como saída do nó LinhasVigasPorPontos através do nó *Output*, que contém o nome da saída exibida (listaDeLinhas), como pode ser visto na Figura C.1.

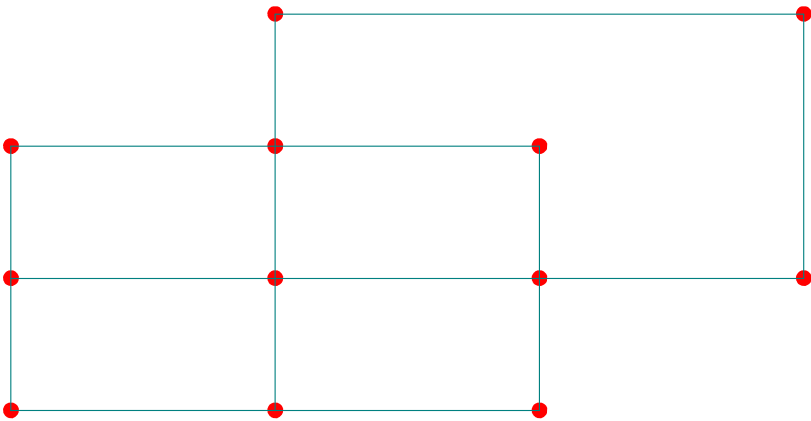
A lista de linhas final passada como saída do nó LinhasVigasPorPontos é a exibida na Figura C.13.

Figura C.12 – Lista de pontos agrupados pelo valor da coordenada y



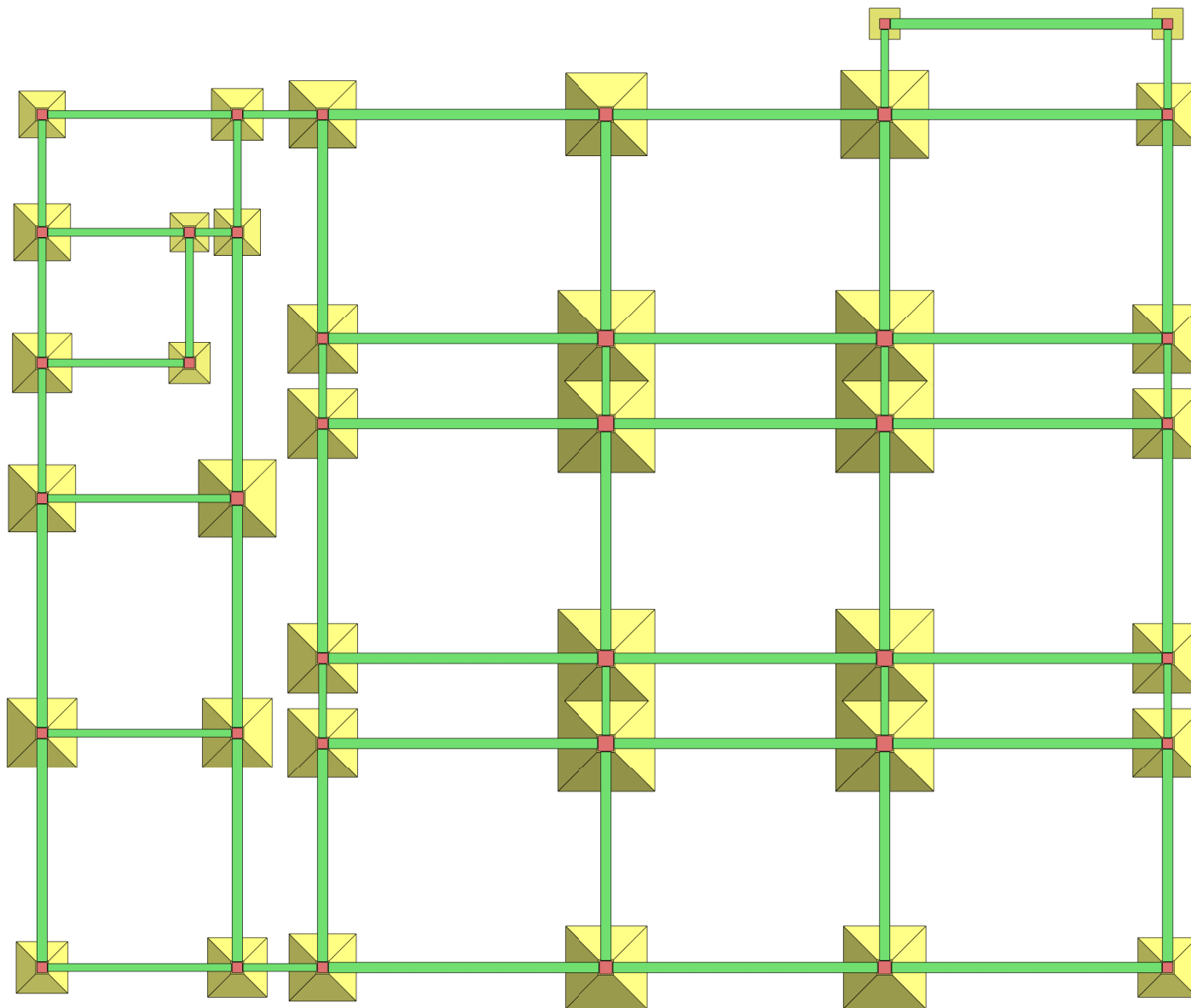
Fonte: Autor

Figura C.13 – Lista de linhas passada como saída



Fonte: Autor

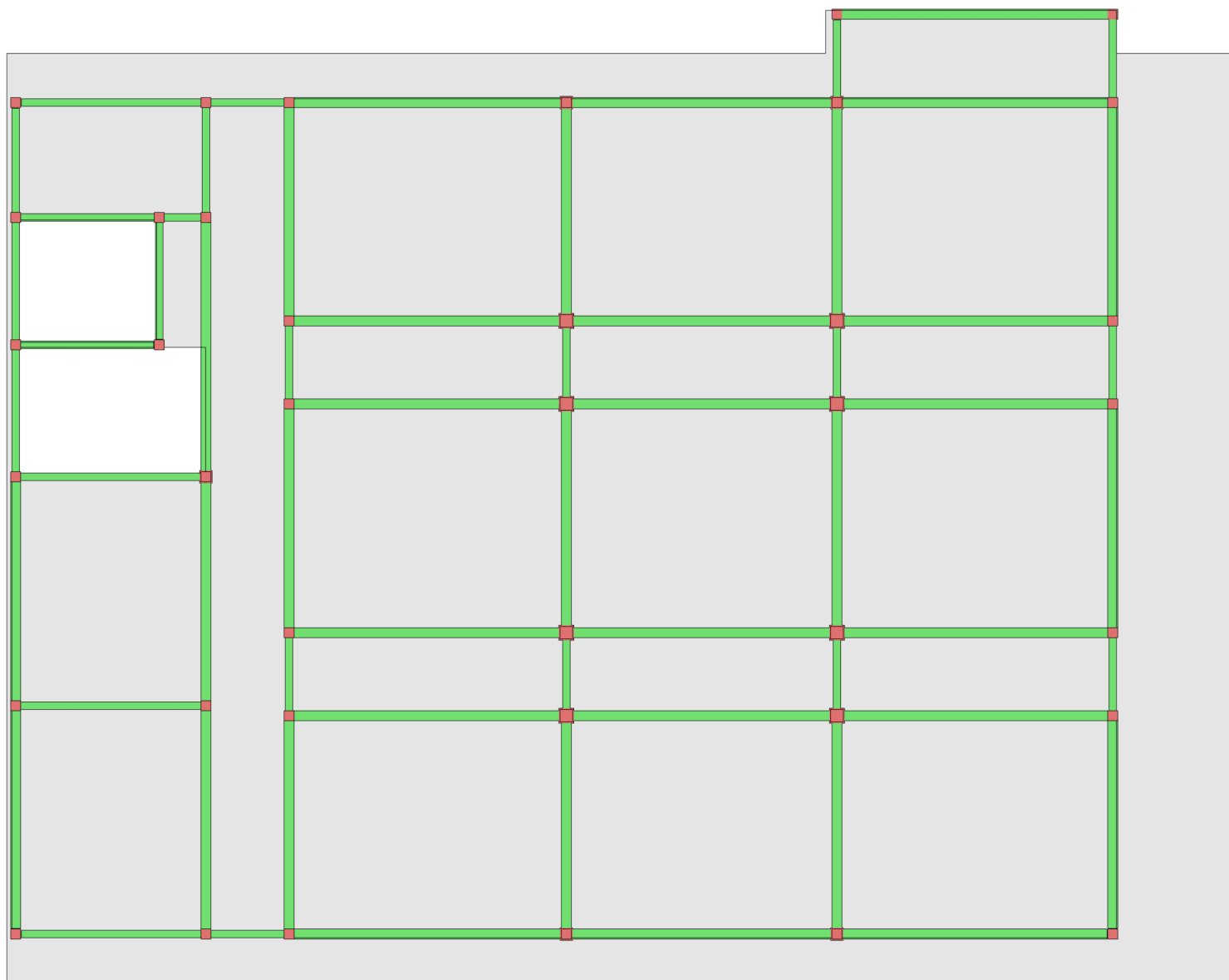
APÊNDICE D – ESTRUTURA CONCEBIDA PELA ROTINA



1

FÔRMA - TÉRREO

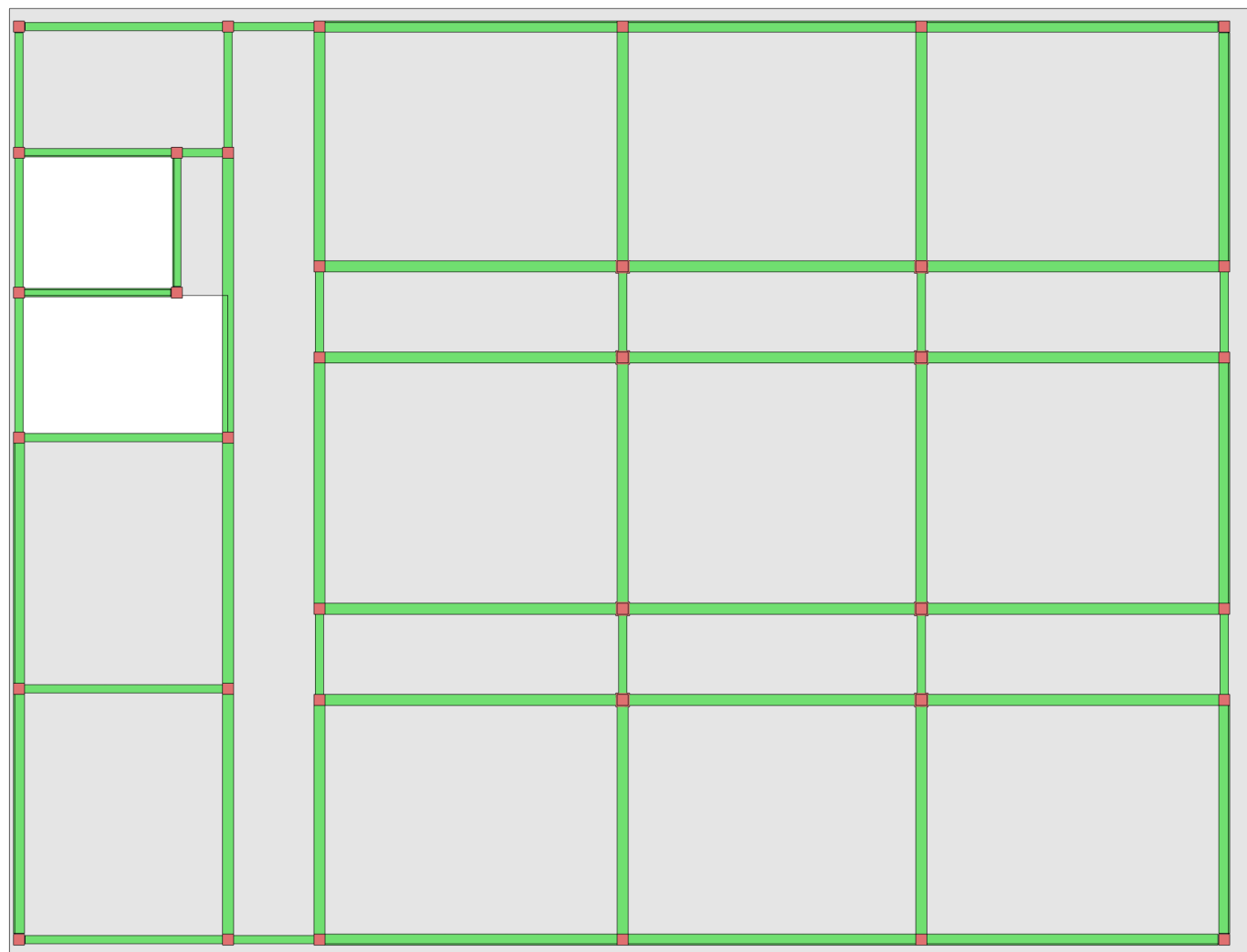
ESCALA 1 : 125



1

FÔRMA - PAVIMENTO 1

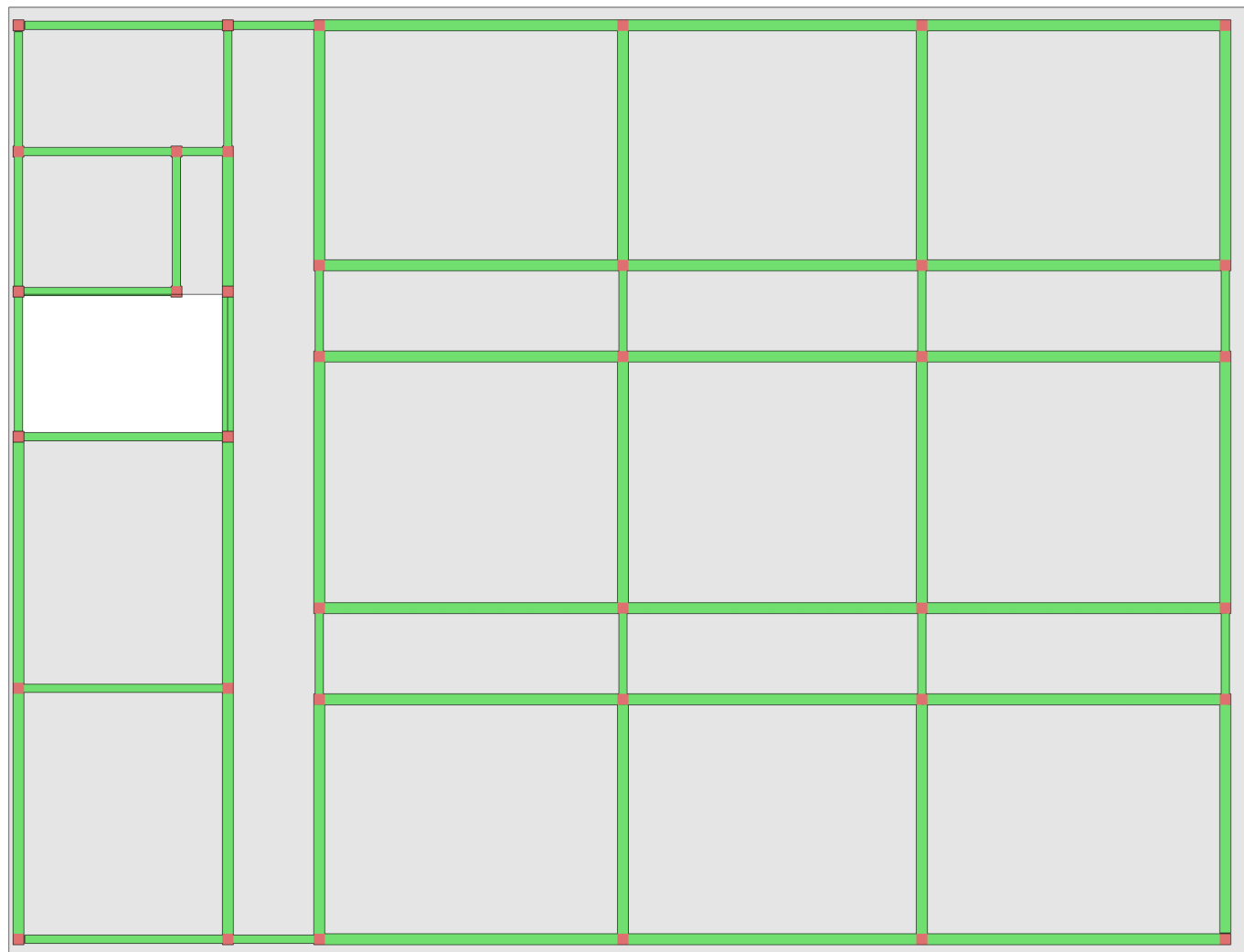
ESCALA 1 : 125



1

FÔRMA - PAVIMENTO 2

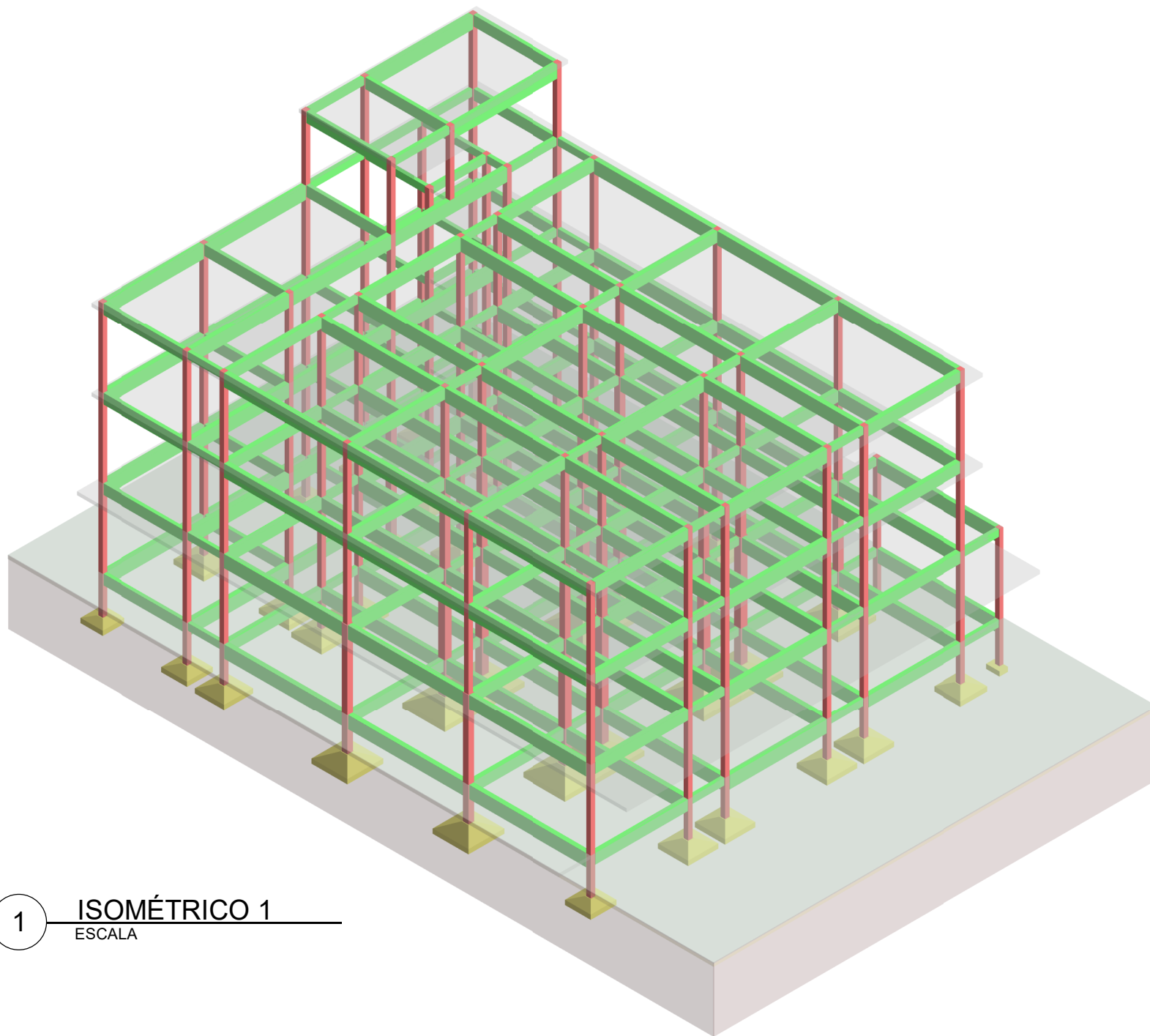
ESCALA 1 : 125



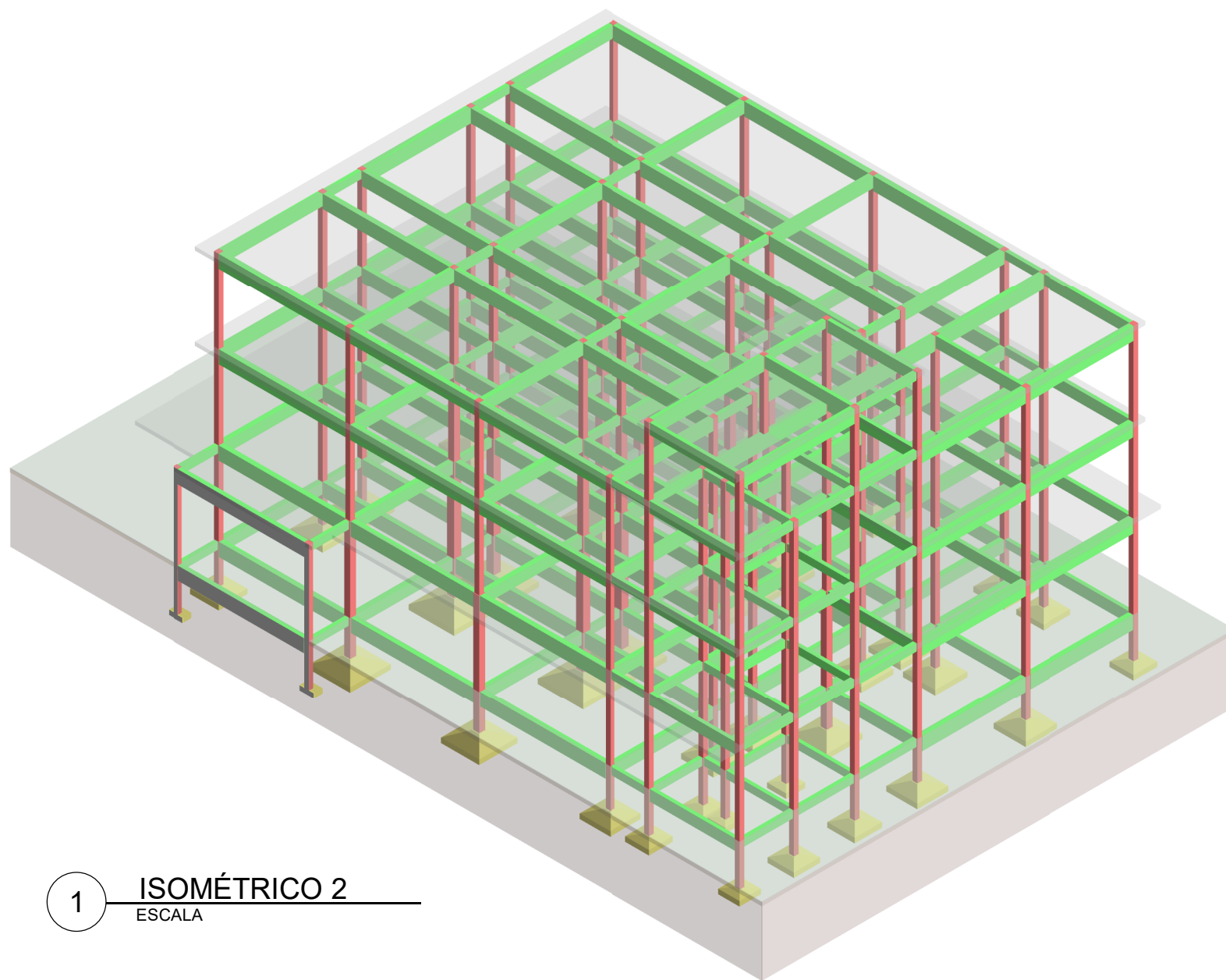
1

FÔRMA - COBERTA

ESCALA 1 : 125



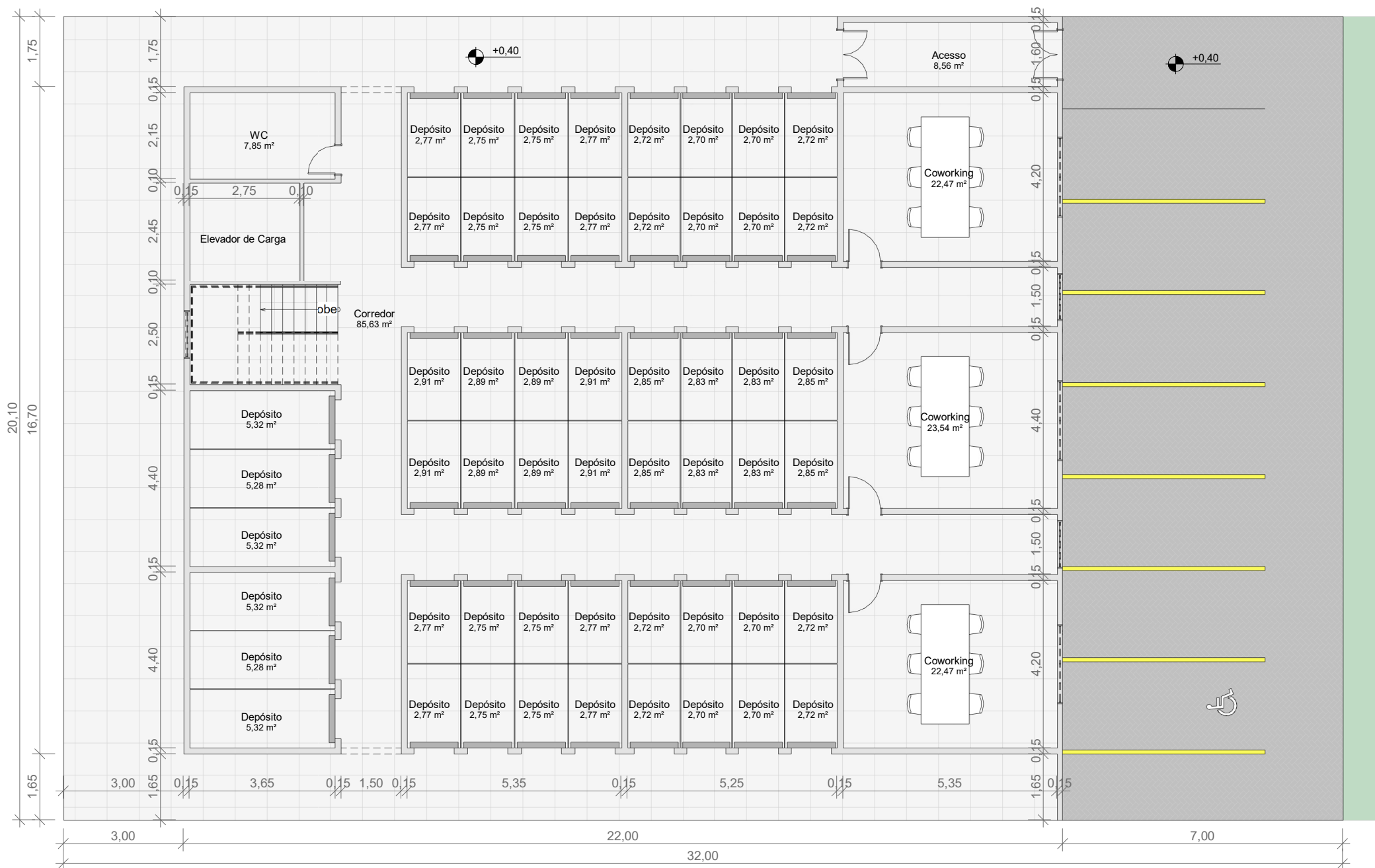
1 ISOMÉTRICO 1
ESCALA

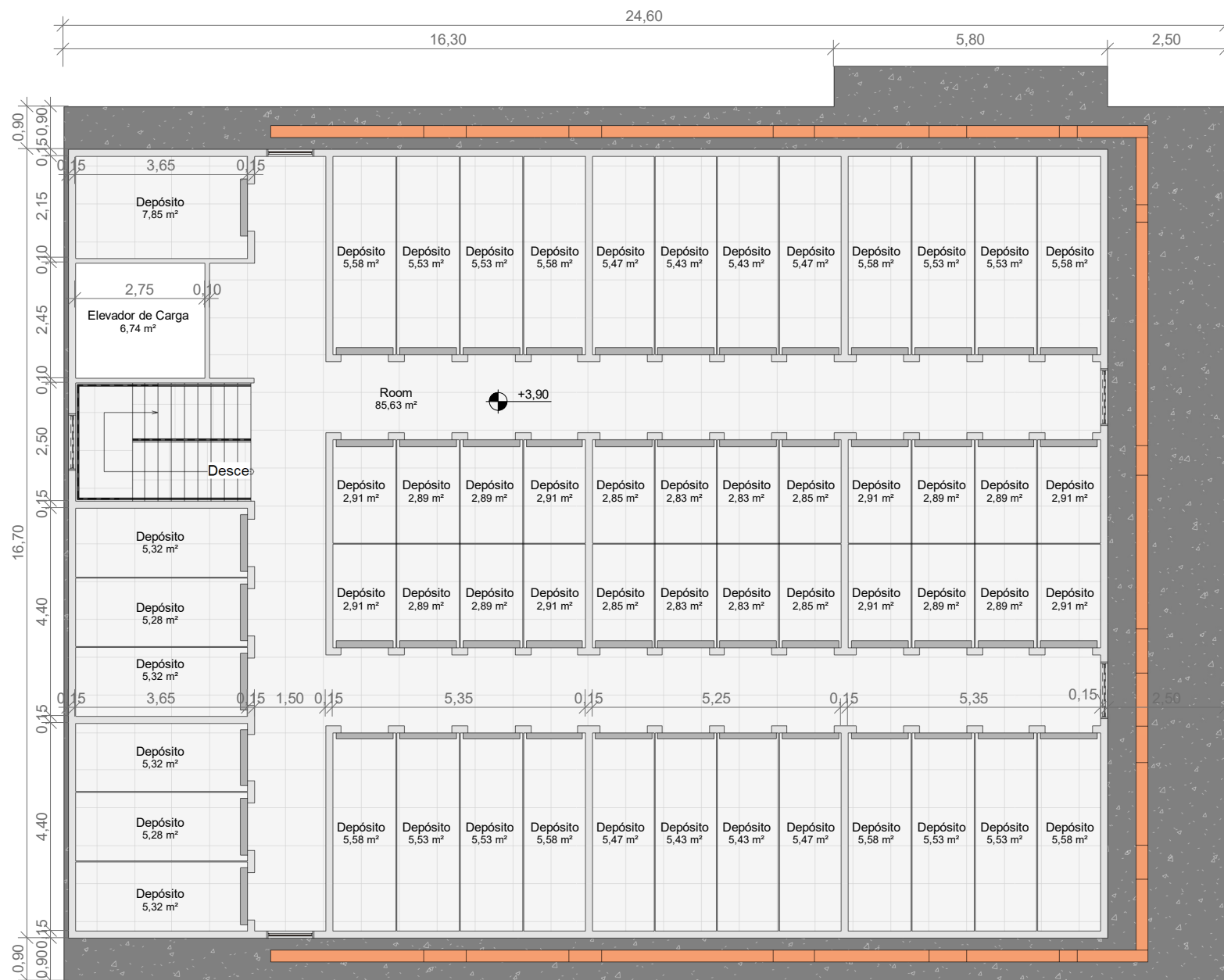


1

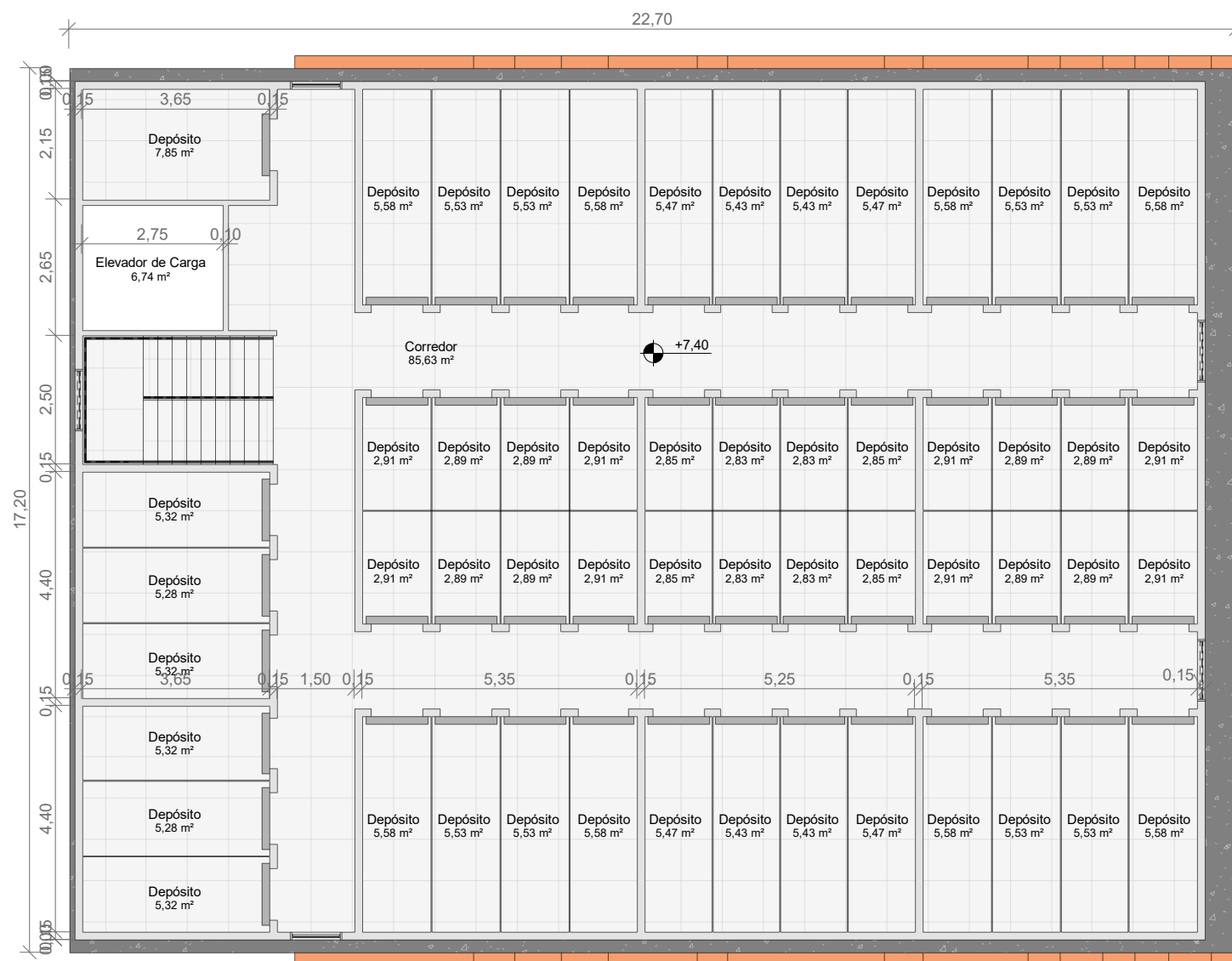
ISOMÉTRICO 2
ESCALA

ANEXO A – ARQUITETURA

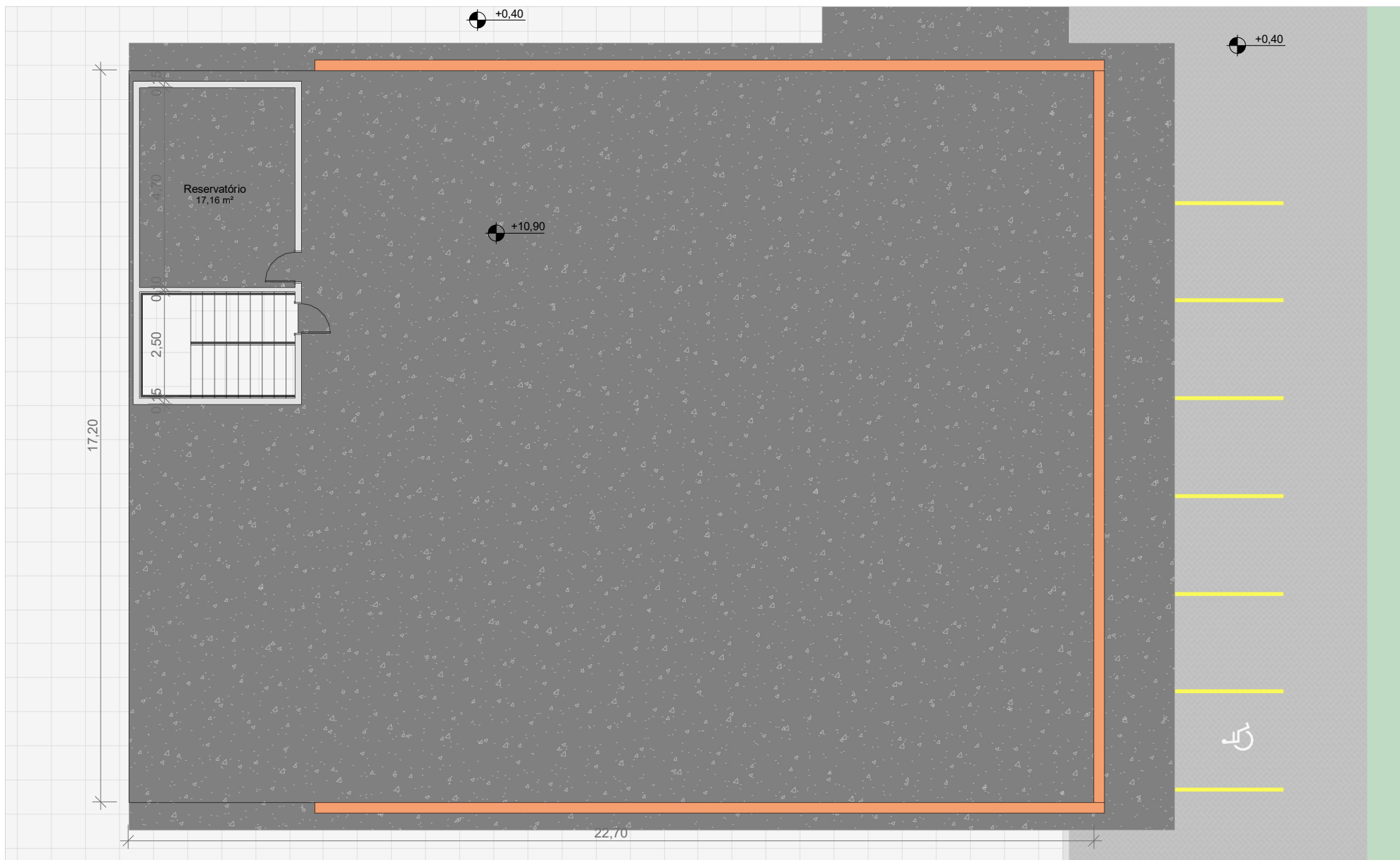




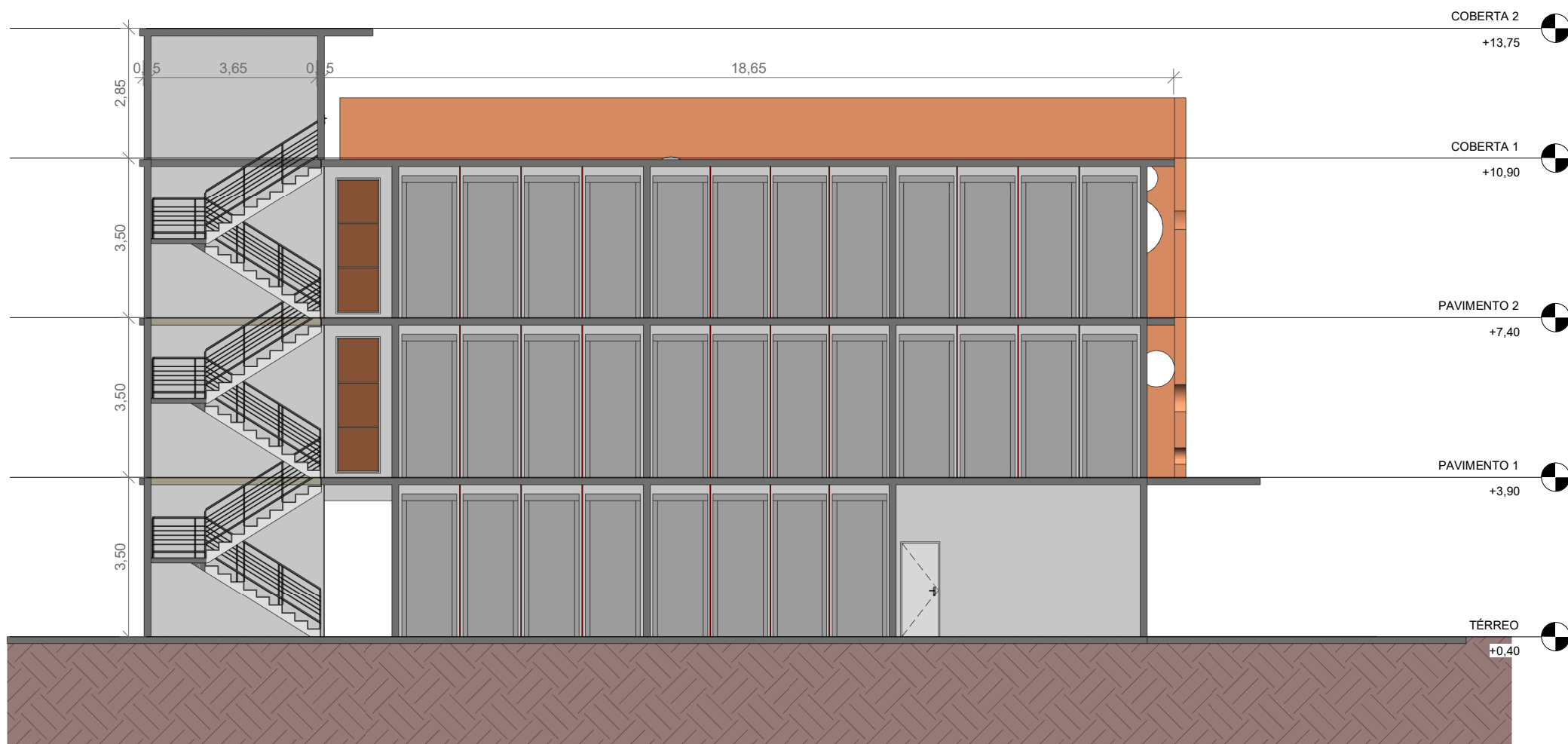
1 **PLANTA BAIXA - PAVIMENTO 1**
ESCALA 1 : 125



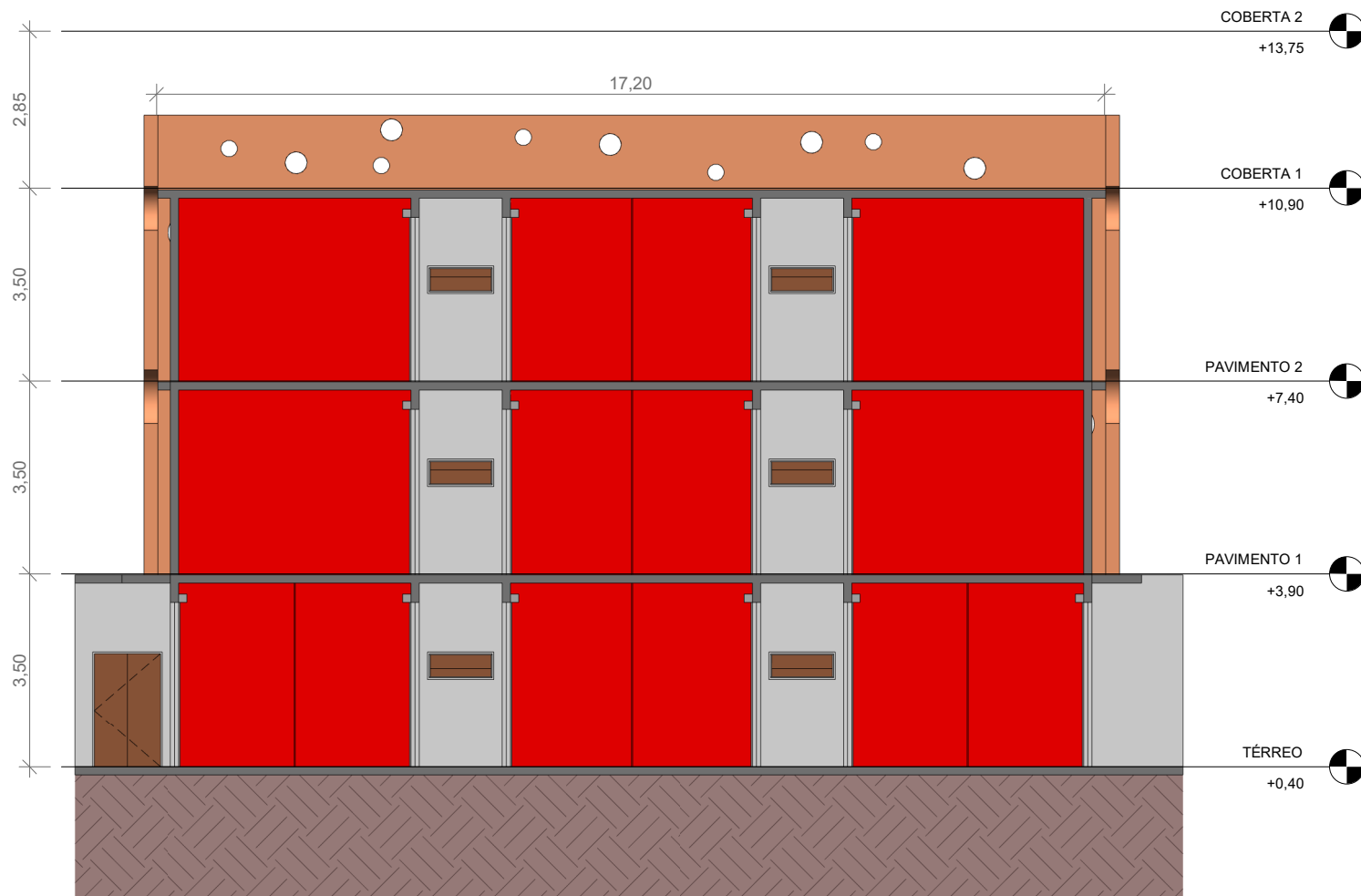
1 PLANTA BAIXA - PAVIMENTO 2
ESCALA 1 : 125



1 PLANTA BAIXA - COBERTA
ESCALA 1 : 125



1 CORTE 1
ESCALA 1 : 125



1 **CORTE 2**
ESCALA 1 : 125



COBERTA 2

+13,75



COBERTA 1

+10,90



PAVIMENTO 2

+7,40



PAVIMENTO 1

+3,90



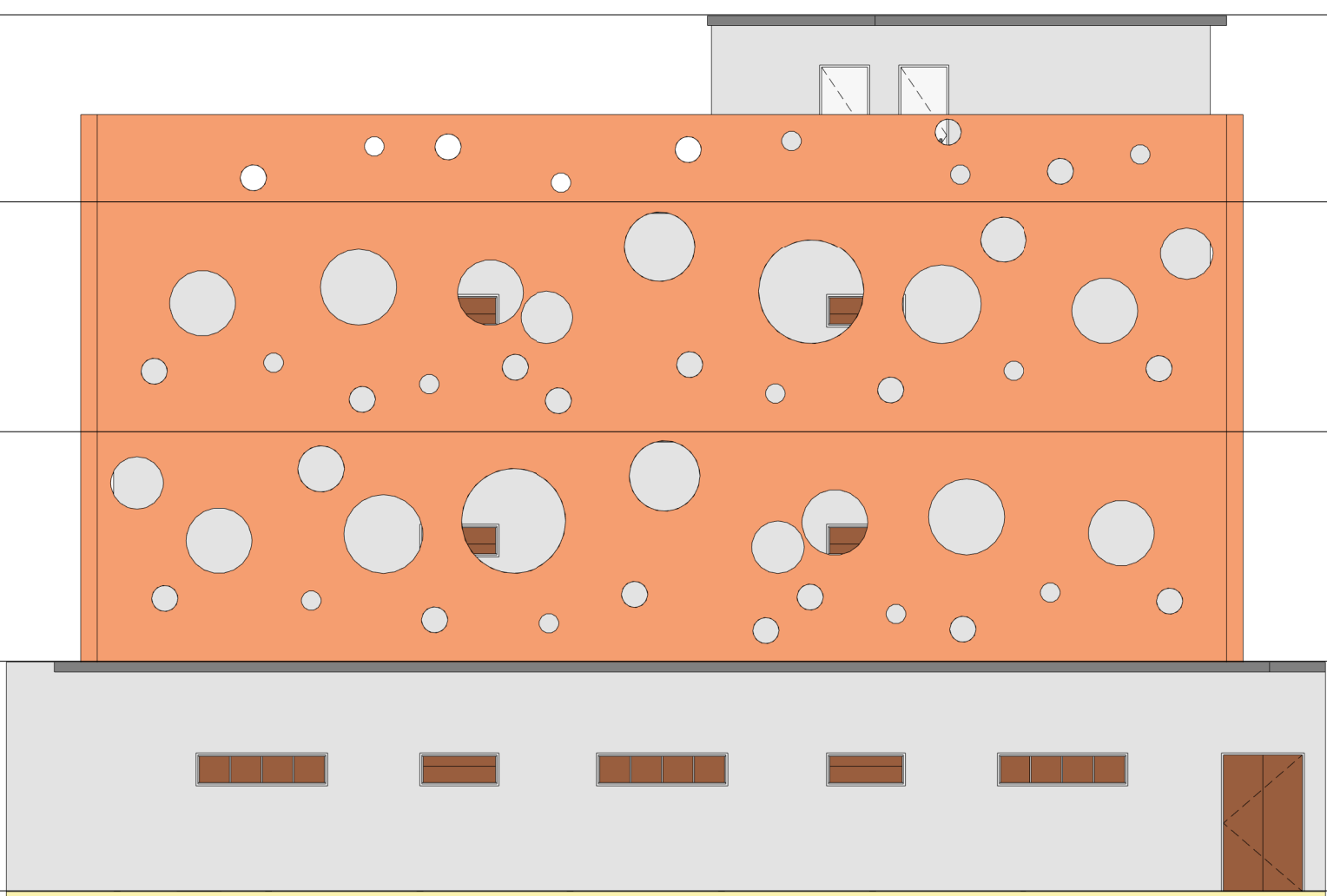
TÉRREO

+0,40

1

FACHADA LESTE

ESCALA 1 : 100

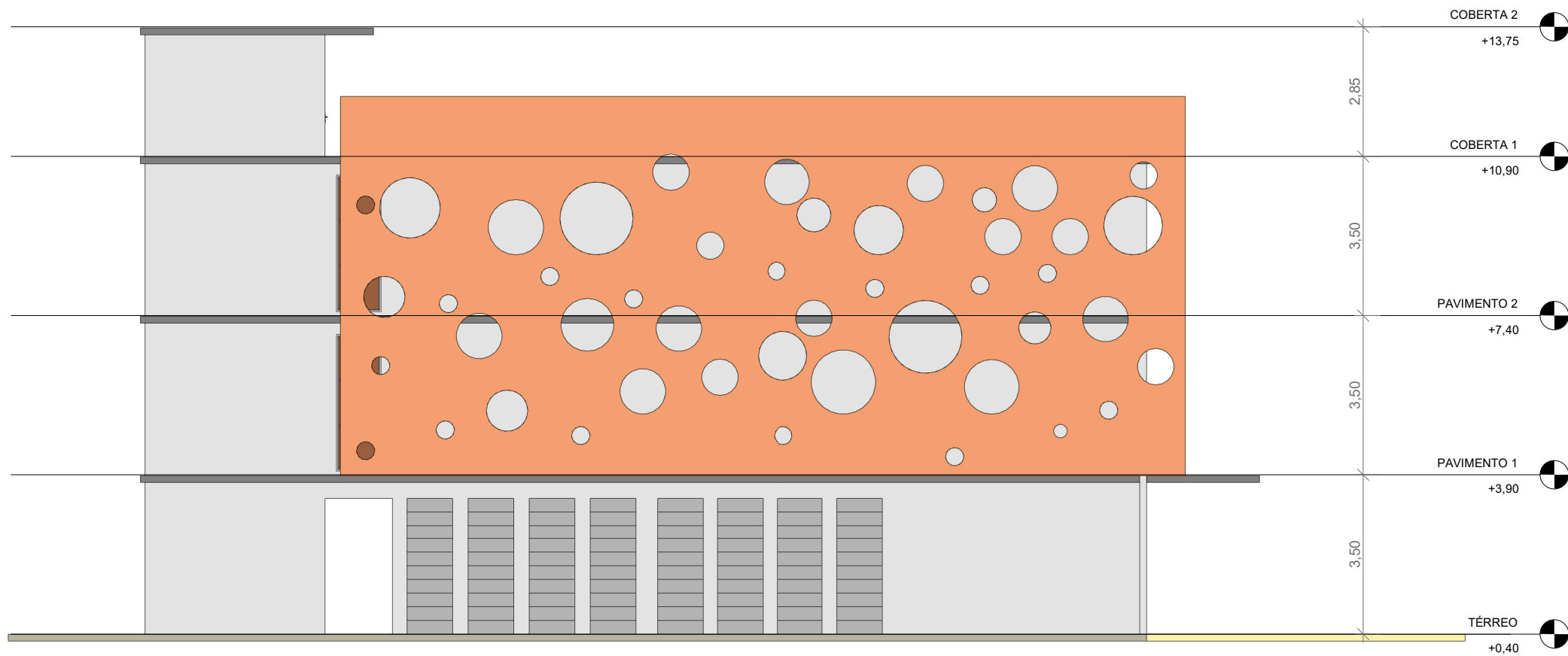


2,85

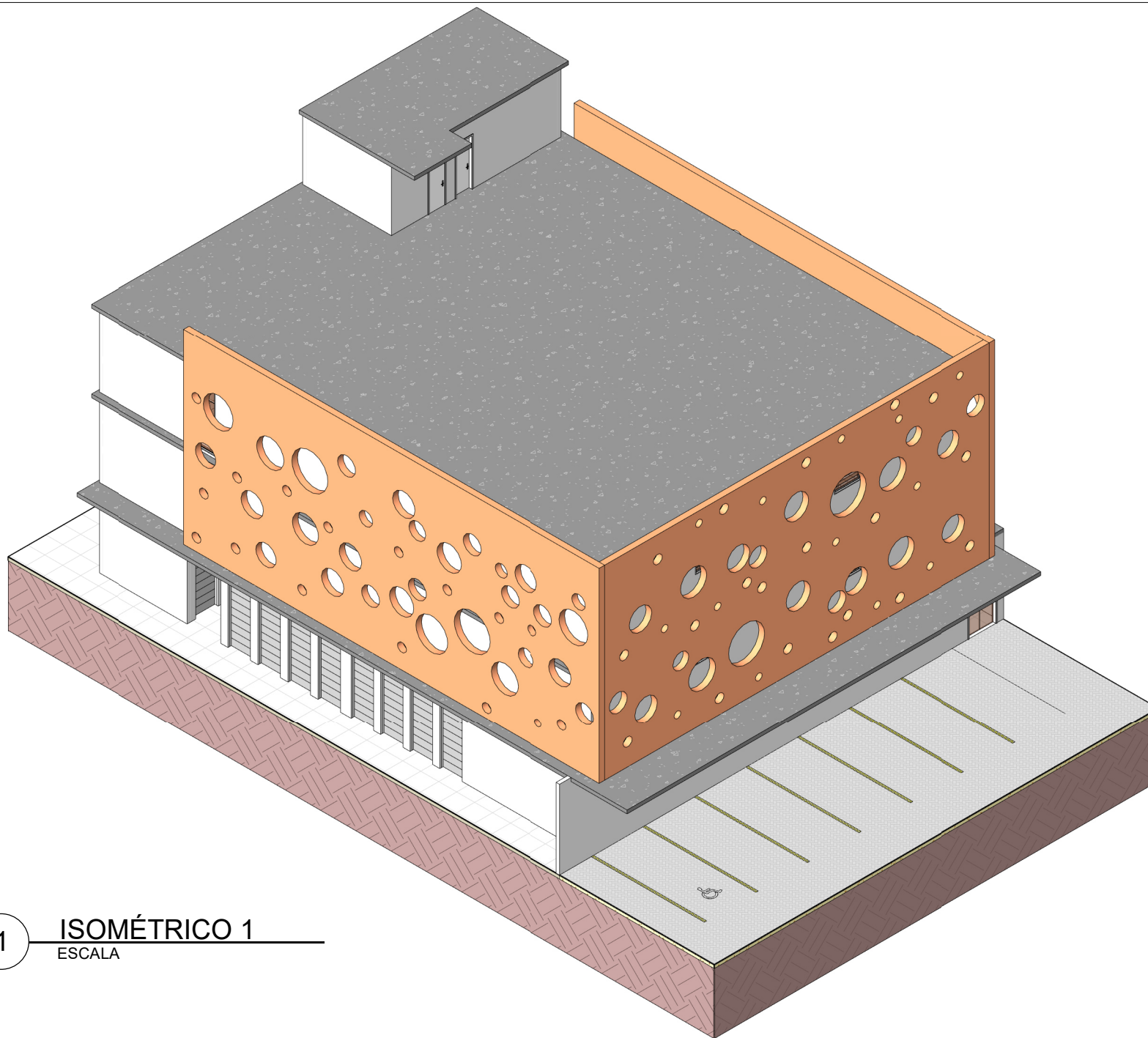
3,50

3,50

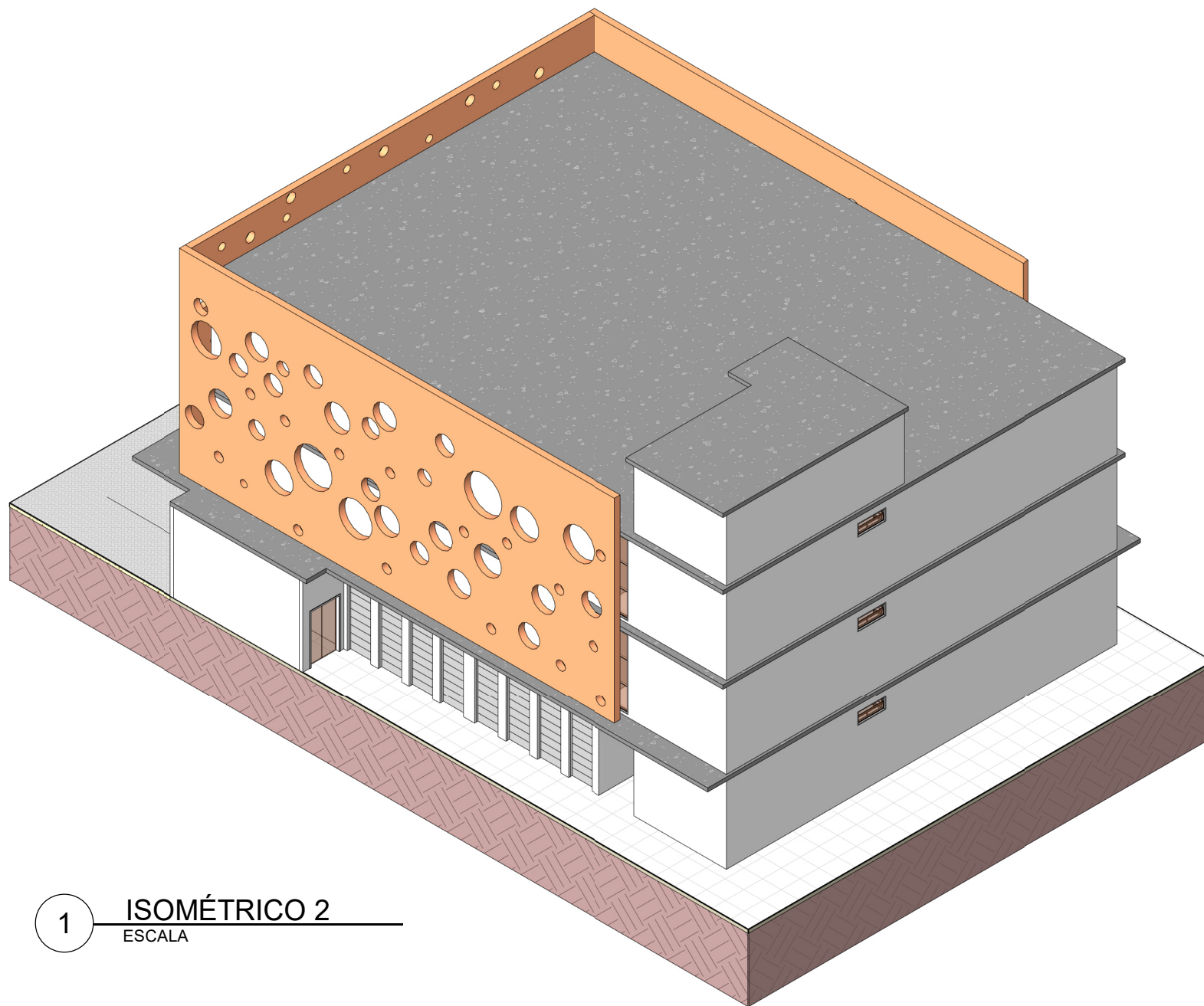
3,50



1 FACHADA SUL
ESCALA 1 : 125

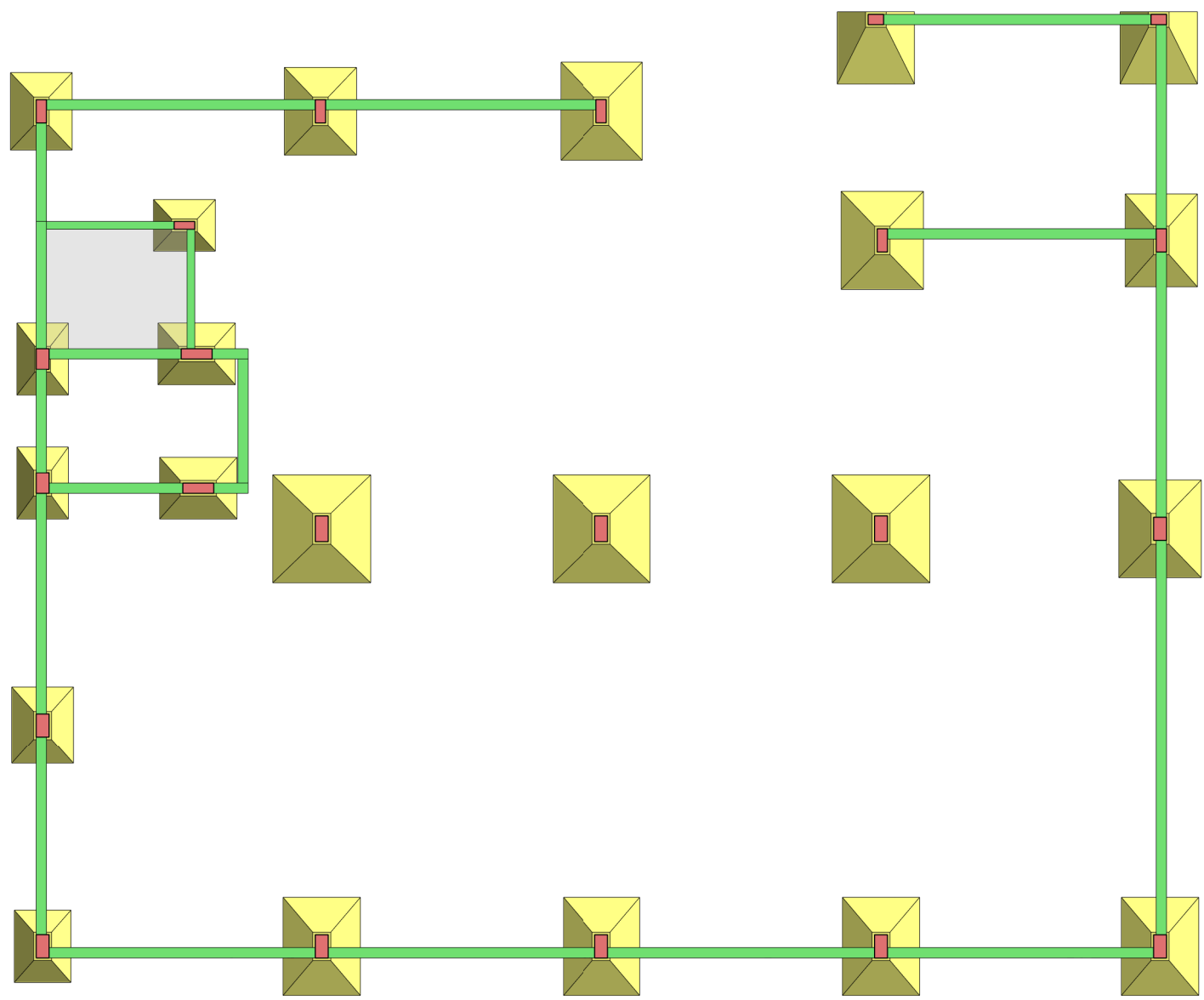


1 ISOMÉTRICO 1
ESCALA

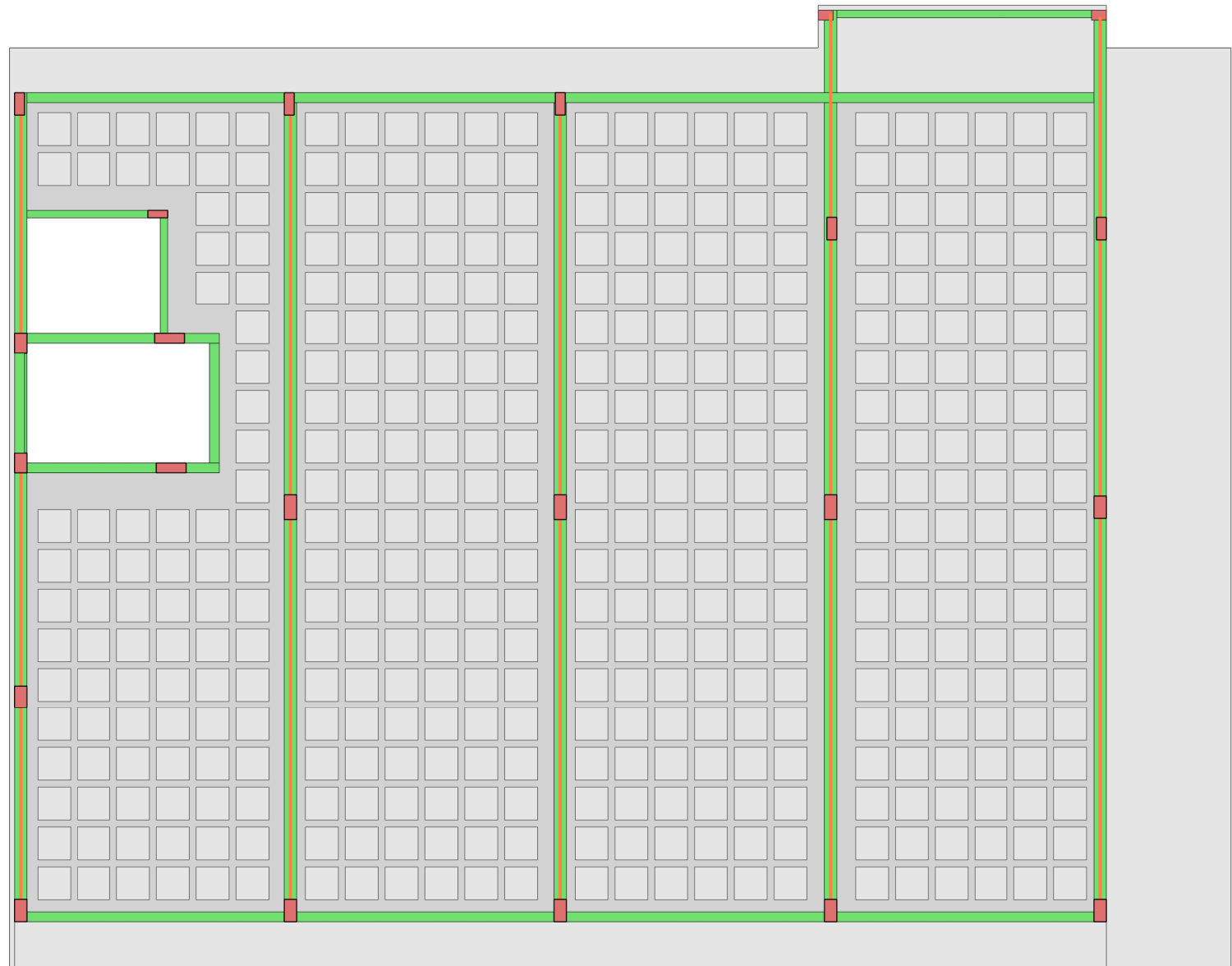


1 ISOMÉTRICO 2
ESCALA

ANEXO B – ESTRUTURA CONCEBIDA MANUALMENTE



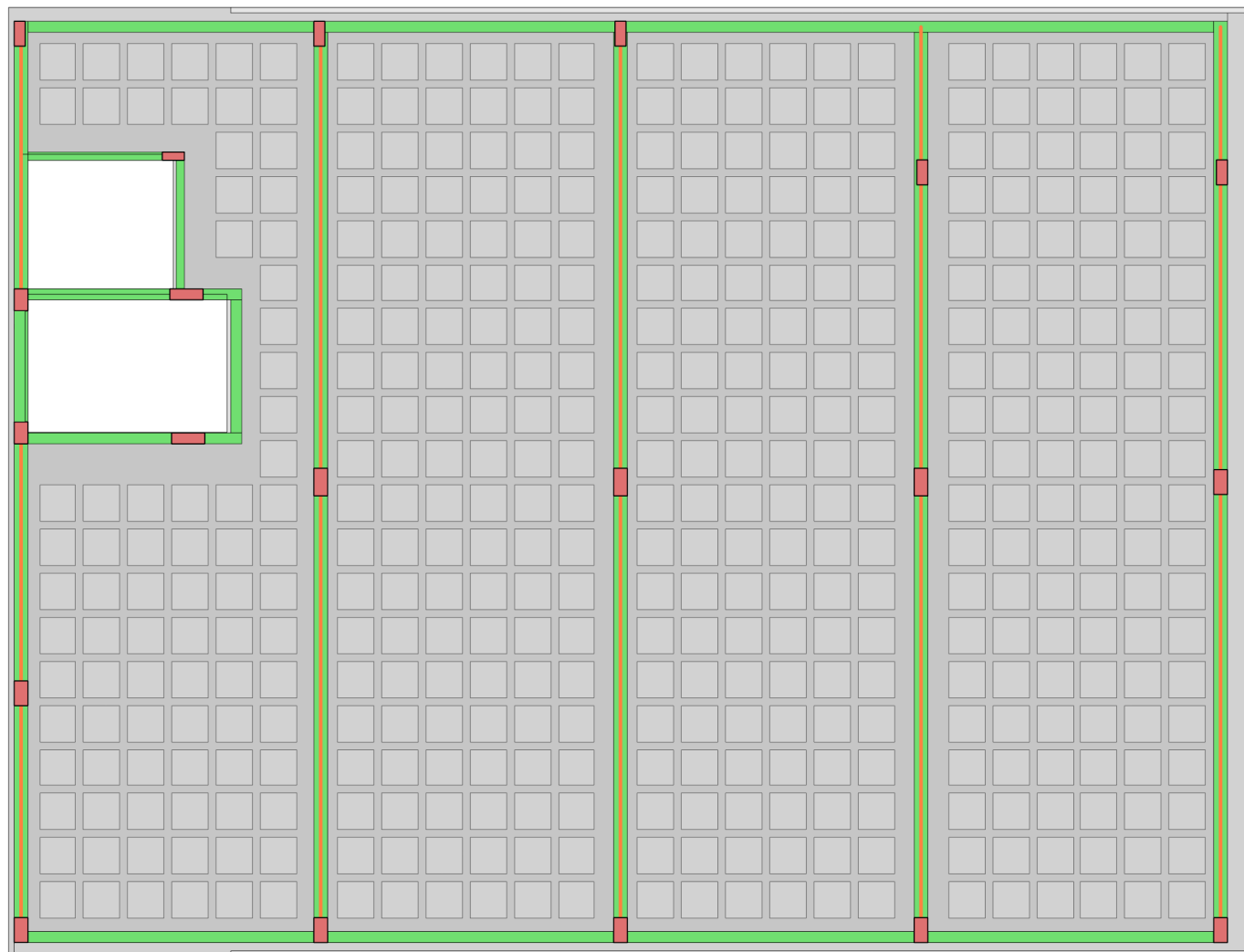
1 FÔRMA - TÉRREO
ESCALA 1 : 125



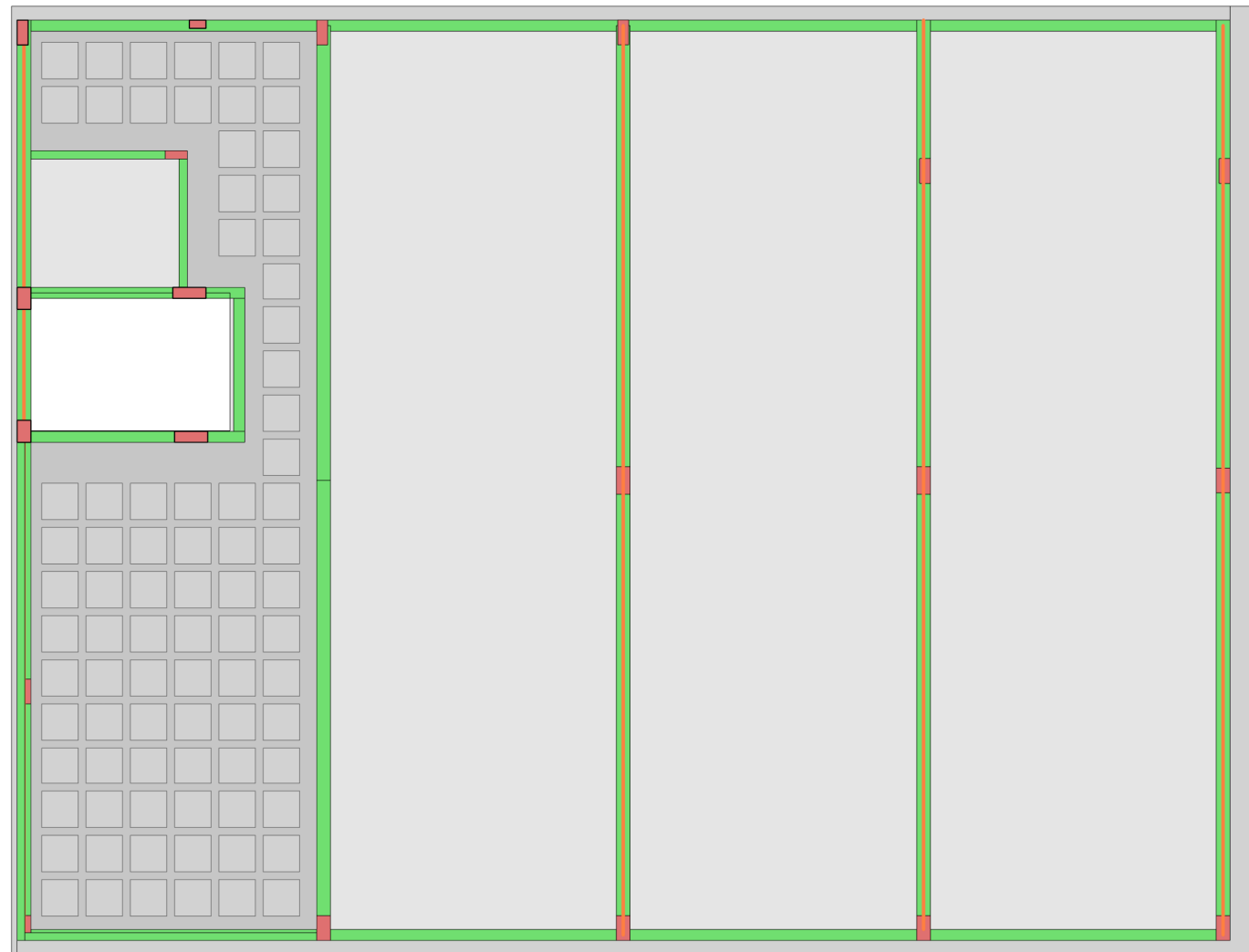
1

FÔRMA - PAVIMENTO 1

ESCALA 1 : 125



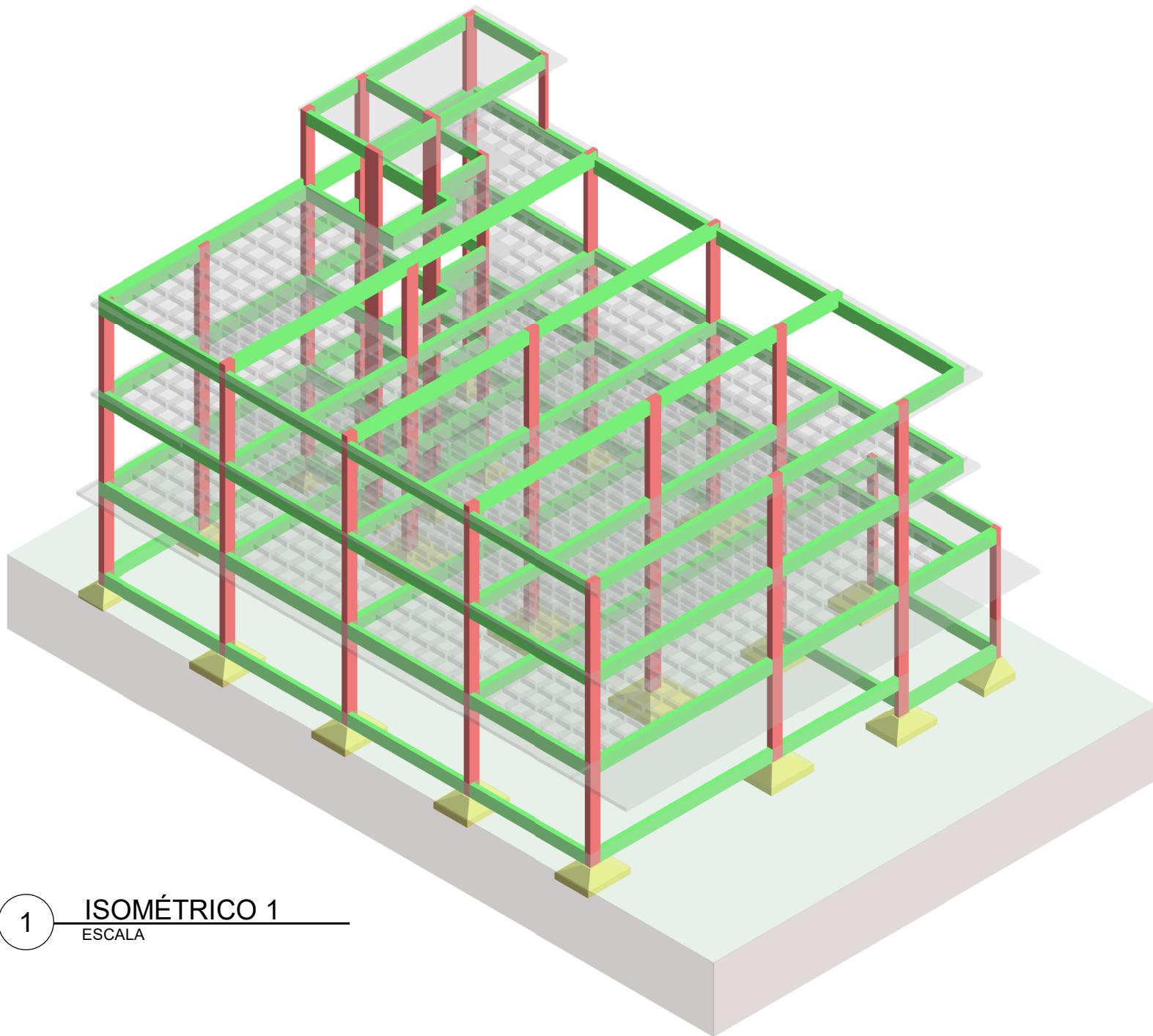
1 FÔRMA - PAVIMENTO 2
ESCALA 1 : 125



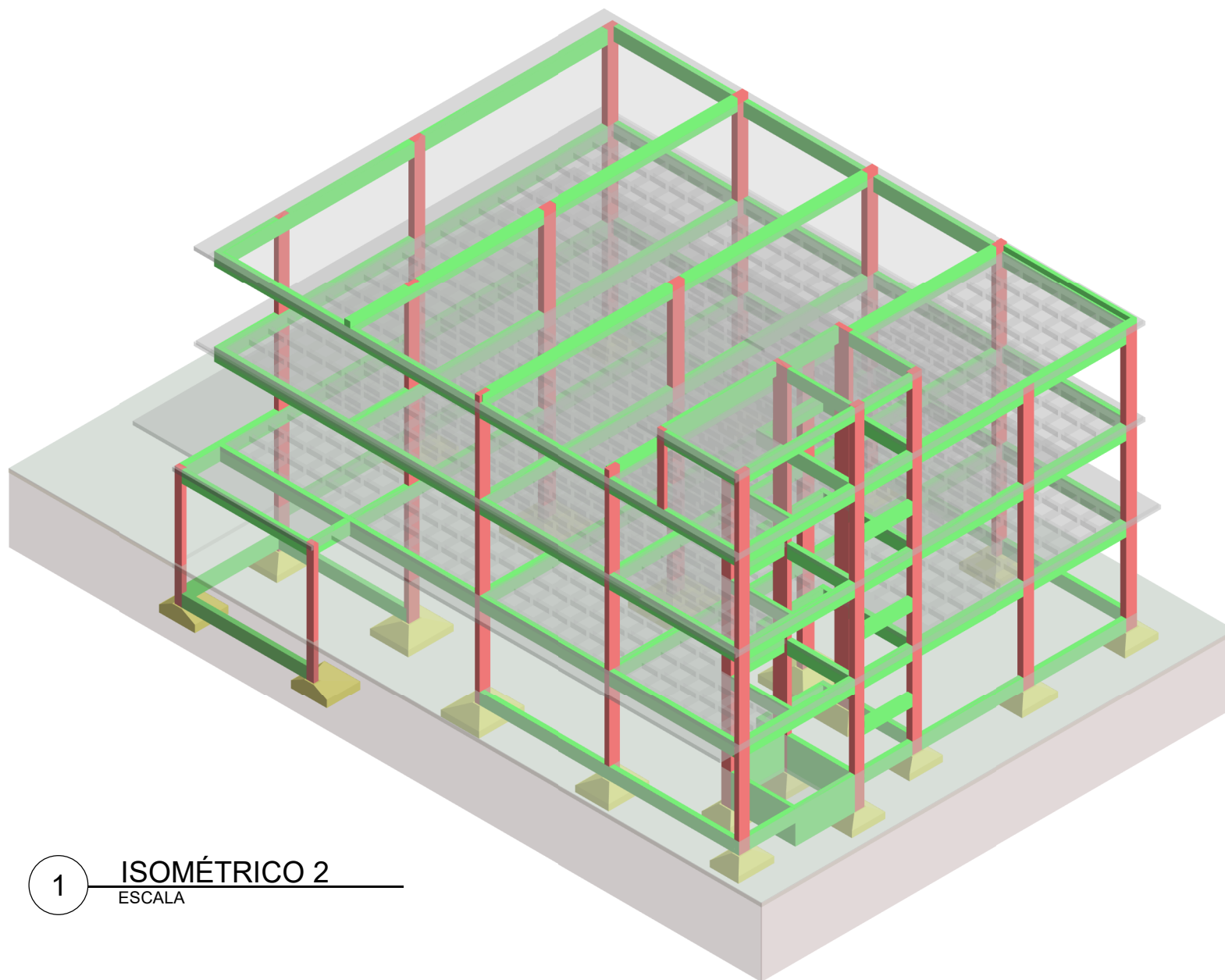
1

FÔRMA - COBERTA

ESCALA 1 : 125



1 ISOMÉTRICO 1
ESCALA



1 ISOMÉTRICO 2
ESCALA