



**Politécnico
de Viseu**

Escola Superior
de Tecnologia
e Gestão de Viseu

Automação do detalhamento estrutural de consolas curtas em BIM: Um estudo de caso

Miguel Dobrovolski Reghin

Trabalho de Projeto

Mestrado em Construção e Reabilitação

Trabalho efetuado sob a orientação de
Professor Doutor Gilberto Antunes Ferreira Rouxinol
Professora Doutora Daniela Gutstein

Novembro de 2024



**Politécnico
de Viseu**

Escola Superior
de Tecnologia
e Gestão de Viseu

Automação do detalhamento estrutural de consolas curtas em BIM: Um estudo de caso

Miguel Dobrovolski Reghin

Trabalho de Projeto

Mestrado em Construção e Reabilitação

Trabalho efetuado sob a orientação de

Professor Doutor Gilberto Antunes Ferreira Rouxinol
Professora Doutora Daniela Gutstein

Novembro de 2024

DEDICATÓRIA

Dedico este trabalho ao Miguel de 2018, que inúmeras vezes se sentiu perdido e considerou trocar de curso.

AGRADECIMENTOS

A gratidão é um sentimento complexo, cujo significado pode variar de acordo com as pessoas e os contextos. Na verdade, a gratidão é um tema recorrente de estudo na psicologia positiva, que destaca como esse sentimento tende, de maneira geral, a tornar as pessoas mais satisfeitas e felizes. A gratidão pode ser interpretada de várias maneiras: como um sentimento, uma virtude, um traço de caráter, um comportamento ou até mesmo como um mecanismo biológico que fortalece e estreita os laços interpessoais.

Independentemente da perspectiva adotada, sinto a necessidade de expressar, de forma escrita, meus sinceros agradecimentos a todos que, de alguma maneira, fizeram parte da minha trajetória até este momento. Embora não seja possível mencionar individualmente todas as pessoas que influenciaram minha jornada — algo que exigiria outra dissertação — buscarei, de maneira concisa, reconhecer aqueles que compõem meu círculo mais próximo.

Primeiramente, quero agradecer aos meus pais, Ricardo Francis Reghin e Dienes Viliane Dobrovolski Reghin, por sempre me incentivarem e apoiarem nos estudos e sonhos, por estarem presentes em todos os momentos importantes, mesmo os pequenos, e por aliviarem o peso da existência ao longo dos anos. Junto a eles, estendo minha gratidão aos meus irmãos: Davi Dobrovolski Reghin, Beatriz Dobrovolski Reghin e Julia Dobrovolski Reghin, por me manterem atualizado sobre os acontecimentos de nossa cidade natal, por compartilharem fofocas e por demonstrarem genuíno interesse nas minhas vivências aqui, por mais banais que fossem.

Agradeço profundamente aos amigos que vieram comigo a Portugal: Francesca Polydoro, Gabriela Barbosa, Gustavo Bonfim, José Lucas Moraes, Juliane Kuser, Laura Mello e Nicolly Novais, por suportarem meus momentos de estresse, por compartilharem histórias, viagens e tantas outras aventuras. Agradeço também pelas incontáveis ocasiões em que trouxeram alegria e leveza às situações por meio de bolos, jantares e festas que inventamos simplesmente porque queríamos. E, claro, por se prontificarem diversas vezes a ler o que escrevi nesta dissertação.

Gostaria de expressar minha sincera gratidão aos meus orientadores, Daniela Gutstein e Gilberto Rouxinol, pelas orientações, paciência e constante incentivo ao longo desta jornada. Suas contribuições foram fundamentais para a conclusão deste trabalho, e sou imensamente grato por isso.

RESUMO

Esta dissertação aborda o uso da Modelagem da Informação da Construção (BIM) e automação de processos no detalhamento de consolas curtas em projetos de estruturas de betão armado. O estudo é dividido em cinco capítulos principais, nos quais são explorados desde a fundamentação teórica até a apresentação dos resultados e sugestões para trabalhos futuros.

No capítulo Revisão de Literatura é fornecida uma base teórica, abordando os conceitos do BIM, suas aplicações e as ferramentas utilizadas, como o Revit e o Dynamo. A importância da interoperabilidade, o histórico de desenvolvimento do BIM, e as normas regulamentadoras que norteiam sua aplicação são discutidos. Este capítulo também abrange as características das consolas curtas de betão armado, classificando-as e apresentando os modos de ruptura típicos dessas estruturas.

O terceiro capítulo apresenta a metodologia adotada, detalhando o desenvolvimento de um script no Dynamo para automatizar o processo de detalhamento de consolas curtas de concreto armado. Este script foi integrado ao Revit, otimizando o processo de geração de quantitativos e minimizando erros comuns em metodologias tradicionais baseadas em CAD. A metodologia também inclui a manipulação e importação de dados de armaduras, permitindo a automatização completa do processo de detalhamento.

Os resultados obtidos a partir da aplicação do script desenvolvido são apresentados e analisados no Capítulo intitulado Análise e discussão dos resultados. Neste capítulo são feitas comparações entre os quantitativos gerados automaticamente e aqueles obtidos por métodos tradicionais. Também se discutem as implicações dos resultados para a prática da engenharia estrutural.

As conclusões apontam para o sucesso do uso do Dynamo em conjunto com o Revit na automação do detalhamento de consolas, evidenciando ganhos em tempo e precisão. Entretanto, também são abordadas limitações, como a necessidade de adaptação para diferentes tipos de estruturas. O capítulo finaliza com sugestões para pesquisas futuras, que poderiam explorar a expansão da metodologia para outros elementos estruturais e o aprimoramento das técnicas de automação empregadas.

Palavras-chave: Betão armado; BIM; Automatização; Detalhamento de consolas curtas; Detalhamento de betão em BIM.

ABSTRACT

This dissertation addresses the use of Building Information Modeling (BIM) and process automation in the detailing of short reinforced corbels in structural design of concrete. The study is divided into five main chapters, which explore everything from the theoretical foundation to the presentation of results and suggestions for future work.

In the Literature Review chapter, a theoretical basis is provided, addressing the concepts of BIM, its applications, and the tools used, such as Revit and Dynamo. The importance of interoperability, the historical development of BIM, and the regulatory standards that guide its application are discussed. This chapter also covers the characteristics of short reinforced concrete corbels, classifying them and presenting the typical failure modes of these structures.

The third chapter presents the methodology adopted, detailing the development of a script in Dynamo to automate the process of detailing short reinforced concrete corbels. This script was integrated into Revit, optimizing the process of generating quantities and minimizing common errors in traditional CAD-based methodologies. The methodology also includes the manipulation and importation of reinforcement data, allowing for full automation of the detailing process.

The results obtained from the application of the developed script are presented and analyzed in the chapter entitled Analysis and Discussion of Results. In this chapter, comparisons are made between the quantities automatically generated and those obtained by traditional methods. The implications of the results for structural engineering practice are also discussed.

The conclusions point to the success of using Dynamo together with Revit in automating corbel detailing, highlighting gains in time and accuracy. However, limitations are also addressed, such as the need for adaptation to different types of structures. The chapter concludes with suggestions for future research, which could explore expanding the methodology to other structural elements and improving the automation techniques employed.

Keywords: Reinforced concrete; Automation; Short corbel detailing; BIM structural design.

ÍNDICE GERAL

ÍNDICE DE TABELAS	viii
ÍNDICE DE FIGURAS	x
ÍNDICE DE SCRIPTS	xiii
LISTA DE SIGLAS/ABREVIATURAS	xiv
LISTA DE SÍMBOLOS	xv
1 INTRODUÇÃO	1
1.1 Enquadramento	1
1.2 Objetivos	2
1.3 Abordagem metodológica	2
1.4 Estrutura da dissertação	2
2 REVISÃO DE LITERATURA	4
2.1 Definição do BIM	4
2.1.1 Histórico do BIM	4
2.1.2 Interoperabilidade dos softwares	4
2.1.3 Níveis de desenvolvimento	5
2.1.4 As normas do BIM	6
2.2 Revit	6
2.2.1 Famílias e parâmetros	7
2.2.2 Grupos e montagens	8
2.2.3 Criação de tabelas	8
2.3 Dynamo	8
2.4 Python	9
2.5 Consolas curtas de betão	9
2.5.1 Classificação das consolas	10
2.5.2 Modos de ruptura	10
2.5.2.1 Ruptura por deformação excessiva do tirante	11
2.5.2.2 Ruptura por tração diagonal	11
2.5.2.3 Ruptura por corte direto	12
2.5.2.4 Ruptura por compressão na flexão	12
2.5.2.5 Ruptura por arrancamento de borda	13
2.6 Modelo de cálculo de consolas curtas	13
2.6.1 Eurocódigo 2	14
2.6.2 ABNT NBR 6118	14
2.6.3 ABNT NBR 9062	14

2.6.4	Parâmetros de cálculo do modelo de escoras e tirantes	15
2.7	Detalhamento de armaduras de consolas curtas	16
2.7.1	Durabilidade das estruturas de betão	18
2.7.2	Comprimento de amarração	22
2.7.3	Diâmetro do pino de dobramento	22
3	METODOLOGIA DA AUTOMAÇÃO DO DETALHAMENTO DE CONSO-	
	LAS CURTAS	23
3.1	Contextualização do projeto de estudo	23
3.1.1	Fluxo de trabalho do setor de projetos	23
3.1.2	Fluxo de trabalho do detalhamento das peças	24
3.2	Projeto base	25
3.2.1	Padrões de nomenclatura	26
3.2.2	Escolha dos casos	26
3.3	Configuração do Revit	28
3.3.1	Configuração dos varões	30
3.4	Interface do Dynamo	30
3.4.1	Pacote Structural Design	32
3.5	Script desenvolvido	34
3.5.1	Importação dos dados de armaduras	35
3.5.2	Seleção de elementos	38
3.5.3	Criação da armadura	40
3.5.3.1	Armadura de costura	41
3.5.3.2	Armadura transversal	44
3.5.3.3	Tirante	47
3.5.3.4	Barra soldada e armadura complementar	49
4	ANÁLISE E DISCUSSÃO DOS RESULTADOS	52
4.1	Método de utilização do script	52
4.1.1	Cuidados ao utilizar o script	53
4.2	Comparativo do detalhamento das armaduras	54
4.2.1	Armadura de costura	55
4.2.2	Armadura transversal	56
4.2.3	Tirante em laço vertical	57
4.2.4	Tirante com barra soldada na extremidade	57
4.3	Comparativo dos quantitativos de aço	58
4.4	Diagramas de montagem	60
5	CONCLUSÃO	62
5.1	Sugestões para trabalhos futuros	62

ÍNDICE DE TABELAS

2.1	Resistência das escoras pelo métodos das bielas e tirantes, adaptado de Costa [38].	15
2.2	Inclinação das escoras pelo método das bielas e tirantes, adaptado de Costa [38].	15
2.3	Resistência dos nós pelo método das bielas e tirantes, adaptado de Costa [38].	16
2.4	Cálculo das áreas das armaduras de consolas muito curtas, adaptado de Silva e Sampaio [39].	17
2.5	Classes de agressividade ambiental segundo a Normas Técnicas (ABNT) [2].	18
2.6	Correlação entre e a classe de agressividade ambiental com os requisitos de qualidade do betão segundo a Normas Técnicas (ABNT) [2].	19
2.7	Correlação entre e classe de agressividade ambiental e o recobrimento nominal do betão, para $\Delta c = 10mm$, segundo a Normas Técnicas (ABNT) [2].	19
2.8	Classes de exposição devida à corrosão induzida por carbonatação, de acordo com o Standardization [1].	20
2.9	Classes de exposição devida à corrosão induzida por cloretos da água do mar, conforme Standardization [1].	20
2.10	Composição e classe de resistência do betão sob ação de dióxido de carbono, segundo Standardization [1], para vida útil de 50 e 100 anos (os valores em negrito indicam os valores para vida útil de 100 anos, quando esses forem diferentes).	21
2.11	Composição e classe de resistência do betão sob ação de cloretos, segundo Standardization [1], para vida útil de 50 e 100 anos (os valores em negrito indicam os valores para vida útil de 100 anos, quando esses forem diferentes).	21
2.12	Composição e classe de resistência do betão sob ataques químicos, segundo Standardization [1], para vida útil de 50 e 100 anos (os valores em negrito indicam os valores para vida útil de 100 anos, quando esses forem diferentes).	21
2.12	Composição e classe de resistência do betão sob ataques químicos, segundo Standardization [1], para vida útil de 50 e 100 anos (continuação)	22
2.13	Comprimentos de ancoragem de acordo com o Standardization [1] e a Normas Técnicas (ABNT) [2], para $\sigma_s = f_{yd} = 435$ MPa (Vaz Rodrigues e Gutstein [40]).	22
2.14	Diâmetro do pino de dobramento, segundo o Standardization [1].	22
2.15	Diâmetro do pino de dobramento para armaduras de tração em aço CA50, segundo a Normas Técnicas (ABNT) [2].	22
2.16	Diâmetro do pino de dobramento para estribos em aço CA50, segundo a Normas Técnicas (ABNT) [2].	22
3.1	Padrões de nomenclatura adotados no estudo.	26
3.2	Lista das consolas simples na obra base	26
3.2	Lista das consolas simples na obra base (continuação)	27
3.3	Lista das consolas duplas na obra base	27
3.4	Variáveis do nó <i>Create.FromCurves</i> , do pacote <i>Structural Design</i>	32
3.4	Variáveis do nó <i>Create.FromCurves</i> , do pacote <i>Structural Design</i> (continuação)	33
3.5	Variáveis do nó <i>Create.FromCurves</i> , do pacote <i>Structural Design</i>	33
3.6	Atribuição de cores para organização do script	34
3.7	Resumo de armaduras das consolas	35
3.8	Padrão de nomenclatura da empresa para as armaduras das consolas	40

3.9	Valores de entrada do nó Python que cria a geometria da armadura de tirantes . .	48
4.1	Quantitativo da consola C01 feito em CAD	59
4.2	Quantitativo da consola C01 gerado pelo Revit	59
4.3	Quantitativo da consola C02 feito em CAD	59
4.4	Quantitativo da consola C02 gerado pelo Revit	59

ÍNDICE DE FIGURAS

2.1	Diferentes níveis de desenvolvimento em modelos BIM, adaptado de Mekawy e Petzold [13].	6
2.2	Estrutura hierárquica de categorias, famílias, tipos e instâncias (Beraldi [21]).	7
2.3	Representação gráfica da estrutura hierárquica de famílias e tipos no Revit (Santos [22]).	7
2.4	Formas de ligação viga-pilar em elementos de pré-fabricados betão (Van Acker [34]).	9
2.5	Modelo de biela-tirante para consola curta segundo a NBR 9062:2017, adaptado de Normas Técnicas (ABNT) [3].	10
2.6	Modelo de consola curta segundo o Anexo J do Eurocódigo 2 (Standardization [1]).	10
2.7	Ruptura de consola devido à tração na flexão (Kriz e Rathes [36]).	11
2.8	Ruptura de consola devido à tração diagonal (Kriz e Rathes [36]).	11
2.9	Ruptura de consola devido ao corte direto (Kriz e Rathes [36]).	12
2.10	Ruptura de consolas devido à compressão na flexão (Kriz e Rathes [36]).	12
2.11	Arrancamento de borda em consola curta (Kriz e Rathes [36]).	13
2.12	Detalhamento de gancho vertical com raio de curvatura elevado (Kriz e Rathes [36]).	13
2.13	Linha isostática de tensões em consolas que respeitam a relação $a/h = 0,5$ (adaptado de Leonhardt e Mönnig [37]).	14
2.14	Detalhamento das armaduras de consolas curtas, segundo o Standardization [1]: (a) Armaduras para $a_c \leq 0,5h_c$; (b) Armaduras para $a_c > 0,5h_c$, adaptado de Standardization [1]	16
2.15	Armadura típica de consolas curtas, pelo padrão da NBR 9062:2017, adaptado de Normas Técnicas (ABNT) [3].	17
3.1	Fluxograma do setor de projetos definido pela empresa.	23
3.2	Fluxo de processos para detalhamento de elementos pré-fabricados definidos pela empresa.	24
3.3	Fluxo de processos, recomendação após implantação do script em Dynamo, para detalhamento de elementos.	24
3.4	Perspetiva isométrica da estrutura do projeto.	25
3.5	Corte longitudinal da estrutura do projeto.	25
3.6	Consola C01; detalhe em CAD fornecido pela empresa.	27
3.7	Consola C02; detalhe em CAD da fornecido pela empresa.	28
3.8	Interface do Revit, imagem propositadamente sem zoom.	28
3.9	Barra de propriedades do Revit.	29
3.10	Barra de propriedades do Revit não expandido.	29
3.11	Janela de configuração dos varões estruturais	30
3.12	Interface do Dynamo, imagem propositadamente sem zoom.	31
3.13	Geometria destacada pela preferência “Isolar Geometria Seleccionada”.	31
3.14	Nó <i>Create.FromCurves</i> , do pacote <i>Structural Design</i>	32
3.15	Nó <i>Layout.SetLayoutAsFixedNumber</i> , do pacote <i>Structural Design</i>	33
3.16	Esboço da rotina de automatização para detalhamento de armaduras, desenvolvida em Dynamo. ¹	34

3.17	Importação da planilha de resumo de armaduras no Dynamo. ¹	35
3.18	Seleção da linha da planilha de resumo de armaduras correspondente ao tipo de consolo a ser detalhado. ¹	36
3.19	Seleção da quantidade de cada tipo varão. ¹	36
3.20	Extração dos diâmetros de cada conjunto de armadura. ¹	37
3.21	Seleção das faces das consolas. ¹	38
3.22	Seleção das dimensões das consolas. ¹	39
3.23	Seleção da seção transversal do pilar. ¹	40
3.24	Nó <i>Properties.RebarHostCover</i>	40
3.25	Limites da armadura de costura: (a) Processo para criar o limite da armadura de costura; (b) Resultado do limites da armadura de costura criado pelo script.	42
3.26	Ajuste das dimensões da curva base para a criação da armadura de costura. ¹	43
3.27	Criação de um vetor no sentido de aplicação da armadura de costura. ¹	43
3.28	Criação do grupo de armaduras de costura. ¹	44
3.29	Origem interna dos pilares.	46
3.30	Pontos da curva base da armadura de costura utilizados no Script 3.8: (a) Ponto zero; (b) Ponto um.	46
3.31	Processo para identificação da face em que a consola está aplicada. ¹	47
3.32	Seleção da face superior da consola.	50
3.33	Criação da geometria da barra soldada: (a) Curva ajustada; (b) Seleção da face frontal da curva.	50
4.1	Acesso ao reprodutor Dynamo através do menu gerenciar	52
4.2	Interface do reprodutor Dynamo	52
4.3	Dados de entrada para execução do script através do reprodutor Dynamo	53
4.4	Consola C01 detalhada, no Revit, a partir das armaduras geradas pelo script em Dynamo	54
4.5	Consola C02 detalhada, no Revit, a partir das armaduras geradas pelo script em Dynamo	54
4.6	Detalhe da armadura de costura da consola C01: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD pela empresa.	55
4.7	Detalhe da armadura de costura da consola C02: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD (conforme projeto de estudo de implantação BIM cedido por empresa).	55
4.8	Seção transversal das consolas: (a) C01; (b) C02.	56
4.9	Detalhe da armadura transversal da consola C01: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD (conforme projeto de estudo de implantação BIM cedido por empresa).	56
4.10	Detalhe da armadura transversal da consola C02: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD (conforme o projeto de estudo de implantação BIM cedido por empresa).	56
4.11	Detalhe do tirante da consola C01: (a) Armadura de detalhada no Revit a partir do script desenvolvido em Dynamo; (b) Armadura de detalhada em CAD (conforme o projeto de estudo de implantação BIM cedido por empresa).	57

4.12	Detalhe do tirante da consola C02: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD (conforme o projeto de estudo de implantação BIM cedido por empresa).	57
4.13	Detalhe da barra soldada na extremidade da consola C02: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD pela empresa.	58
4.14	Detalhe da barra complementar da consola C02: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD pela empresa.	58
4.15	Diagrama de montagem da consola C01.	60
4.16	Diagrama de montagem da consola C02.	60

ÍNDICE DE SCRIPTS

3.1	Seleção do índice da linha, na folha de resumo de armaduras, correspondente ao tipo de consola a ser detalhada.	36
3.2	Conversão de variáveis do tipo <i>String</i> para variáveis do tipo <i>Integer</i>	37
3.3	Obter atributo <i>length</i> em objetos de classe <i>line</i>	39
3.4	Verificar o paralelismo ente os lados do pilar e da consola.	41
3.5	Calcular a distância entre um vértice da consola e os lados do pilar	41
3.6	Criar os vértices do limite da armadura de costura	41
3.7	Ajusta a curva base para criação da armadura transversal	45
3.8	Identificação da face em que a consola está posicionada	46
3.9	Ajuste do afastamento com base na face em que a consola está aplicada	47
3.10	Identificação da face em que a consola está posicionada	48
3.11	Calculo do comprimento de ancoragem dos tirantes.	49
3.12	Comprimento do laço inferior do tirante.	49

LISTA DE SIGLAS/ABREVIATURAS

IPV	Instituto Politécnico de Viseu
ESTGV	Escola Superior de Tecnologia e Gestão de Viseu
ABNT	Associação Brasileira de Normas Técnicas
NBR	Norma Brasileira
CAD	<i>Computer Aided Design</i>
BIM	<i>Building Information Modeling</i> /Modelagem de Informação da Construção
IFC	<i>Industry Foundation Classes</i>
2D	Duas dimensões
3D	Três dimensões
CSG	<i>Constructive Solid Geometry</i>
B-rep	<i>Boundary Representation</i>
DXF	<i>Drawing Exchange Format</i>
IAI	<i>Industrial Alliance of Interoperability</i>
IDM	Information Delivery Manual/Manuais de Entrega de Informação
LOD	Níveis de Desenvolvimento
a. C.	antes de Cristo
V. S. F. F.	volte, se faz favor
e. g.	<i>exempli gratia</i>
etc.	<i>et cetera</i>
ibid.	<i>ibidem</i>
i. e.	<i>id est</i>
máx.	máximo
mín.	mínimo
fig.	figura
figs.	figuras
n.º	número
n.ºs	números
pág.	página
págs.	páginas
p.	página
pp.	páginas
A.	autor
AA.	autores

LISTA DE SÍMBOLOS

a) Maiúsculas Latinas

A_s	Área da seção de uma armadura para betão armado
F_t	Força de tração
F_{wd}	Valor de cálculo da resistência ao corte da soldadura
F_{Ed}	Esforço de compressão na ligação
F_c	Força de compressão correspondente à força de tração máxima $F_{s,max}$
H_d	Carga horizontal do projeto

b) Minúsculas Latinas

b	Largura do consolo
d	Altura útil do consolo
h	Altura do consolo
z	Braço de alavanca
a	Comprimento da biela de compressão

c) Maiúsculas Gregas

$\sigma_{Rd,max}$	Valor de cálculo máximo da resistência à compressão de uma escora de concreto
-------------------	---

d) Minúsculas Gregas

ν'	Coefficiente de redução da resistência do betão fendilhado por esforço transversal
α_{V2}	Fator de redução utilizado na verificação da compressão diagonal do concreto em bielas comprimidas

e) Características do Betão

f_{ck}	Valor característico da tensão de ruptura do betão à compressão
f_{cd}	Valor de cálculo da tensão de ruptura do betão à compressão
f_{cd1}	Tensão resistente máxima no betão, em verificações pelo método de escoras e tirantes, em regiões com tensões de compressão transversal ou sem tensões de tração transversal e em nós onde confluem somente escoras de compressão
f_{cd2}	Tensão resistente máxima no betão, em verificações pelo método de escoras e tirantes, em regiões com tensões de tração transversal e em nós onde confluem dois ou mais tirantes tracionados
f_{cd3}	Tensão resistente máxima no betão, em verificações pelo método de escoras e tirantes, em nós onde conflui um tirante tracionado

1 INTRODUÇÃO

A presente dissertação aborda a implementação de tecnologias no setor da construção civil, com foco na automatização de processos de detalhamento estrutural em projetos de engenharia. Especificamente, investiga-se o uso do *Building Information Modeling* (BIM) e da programação visual por meio do Dynamo, para a automação do detalhamento de armaduras em consolos de betão armado. A seguir, são apresentados o contexto em que se insere esta pesquisa, os objetivos definidos, a metodologia empregada e a estrutura geral da dissertação.

1.1 Enquadramento

A indústria da construção civil está em constante evolução, impulsionada pela necessidade de otimizar processos, reduzir custos e aumentar a precisão dos projetos. Nesse cenário, a implementação de novas tecnologias tem desempenhado um papel na transformação das práticas tradicionais. O *Building Information Modeling* (BIM) destaca-se como uma das inovações atuais mais importantes nesta indústria, pois permite a criação de modelos digitais ricos em informações que integram diferentes etapas do ciclo de vida de uma edificação. Dessa forma, com o BIM, é possível visualizar, simular, quantificar e analisar aspectos construtivos e operacionais de um projeto antes mesmo de sua execução física, promovendo uma maior eficiência e precisão no setor.

Dentro do escopo do BIM, a automatização de tarefas repetitivas e suscetíveis a erros humanos, como o detalhamento de elementos estruturais, tem se mostrado uma estratégia promissora. O detalhamento de armaduras em consolas curtas de betão armado, por exemplo, é uma tarefa que exige alto grau de precisão e atenção aos detalhes. Historicamente, esse processo foi realizado manualmente, sendo mais tarde transferido para programas computacionais de desenho técnico, denominados de CAD (*Computer Aided Design*), o que representou uma evolução, uma vez que esses softwares facilitaram a criação e modificação de desenhos com maior rapidez e precisão em relação aos métodos manuais. No entanto, o uso do CAD também apresenta limitações, especialmente quando se trata da gestão e coordenação de grandes volumes de dados e a prevenção de inconsistências entre os diferentes desenhos.

Nesse sentido, a transição para um ambiente BIM, onde o detalhamento é realizado de forma integrada e automatizada, representa um avanço significativo. No BIM, todas as informações do projeto são centralizadas em um modelo tridimensional, permitindo que inconsistências, como diferenças nas armaduras ou nas quantidades de materiais entre elementos iguais, sejam facilmente detectadas e corrigidas. Essa centralização de dados resulta em maior precisão, melhor uso dos recursos e uma elevação da qualidade geral dos projetos. Além disso, a automatização de processos dentro do BIM abre portas para a utilização de ferramentas que podem potencializar ainda mais os resultados.

O Dynamo é uma ferramenta de programação visual integrada ao Revit, que pode ser utilizado para automatizar processos dentro do metodologia BIM. O Dynamo permite a criação de scripts que automatizam tarefas repetitivas, como a geração de armaduras em consolas curtas, e garantem a consistência dos resultados. Sendo assim, a integração do Dynamo com o Revit possibilita a modelagem paramétrica, onde alterações no projeto são automaticamente refletidas no detalhamento das armaduras, reduzindo o tempo de resposta a mudanças e minimizando o risco de erros. Por conta disso, essa abordagem geralmente oferece um diferencial competitivo, uma vez que projetos podem ser desenvolvidos com maior precisão e eficiência.

Além disso, a adoção do BIM e de ferramentas como o Dynamo reflete uma tendência na engenharia civil, onde a digitalização e a automação estão modificando as práticas de trabalho. Sendo que, empresas que adotam o BIM de forma abrangente podem obter benefícios como melhor coordenação entre disciplinas, redução de retrabalhos e maior controle sobre o ciclo de vida dos projetos. Nesse sentido, a migração dos processos de detalhamento de CAD para BIM é uma resposta à necessidade de atender a demandas crescentes por maior qualidade e eficiência nos projetos de engenharia.

Entretanto, a implementação dessas tecnologias requer uma adaptação significativa das práticas existentes e o desenvolvimento de novas competências dentro das equipes de projeto. Visto que,

a criação de scripts no Dynamo, por exemplo, exige conhecimentos tanto de programação quanto dos princípios de modelagem estrutural. Dessa forma, este trabalho, busca contribuir para essa transição, propondo uma metodologia que automatiza o detalhamento das armaduras de consolas curtas de betão armado no Revit, utilizando o Dynamo como ferramenta de suporte. Ao fazer isso, pretende-se melhorar a precisão e eficiência dos projetos, e também auxiliar otimização de alguns processos da empresa estudada. Além disso, este trabalho busca estabelecer um paralelismo entre a norma portuguesa para betão armado, Standardization [1], e as normas brasileiras Normas Técnicas (ABNT) [2] e Normas Técnicas (ABNT) [3], tendo como objetivo utilizar as normas brasileiras e demonstrar que as competências para aplicar a norma portuguesa ou a norma brasileira é a mesma.

1.2 Objetivos

O objetivo principal desta dissertação é desenvolver e implementar um script no Dynamo para automatizar o processo de detalhamento das armaduras de consolas curtas retangulares simples de betão armado no software Revit. Essa automatização visa aumentar a precisão e a consistência dos quantitativos gerados, otimizar o tempo necessário para a execução dessa tarefa e reduzir os erros específicos que são comuns no processo manual tradicional. Além disso, o trabalho busca promover a transição dos processos de detalhamento, usualmente realizados em CAD, para um ambiente BIM tomando como base um estudo de caso comparativo realizado para um projeto de uma empresa do setor de projetos estruturais.

1.3 Abordagem metodológica

Para alcançar os objetivos propostos, a metodologia adotada nesta pesquisa envolve várias etapas. (1) Inicialmente, realizou-se uma revisão bibliográfica para entender as melhores práticas e as principais dificuldades no detalhamento de consolas curtas de betão armado e no uso de BIM. Em seguida, (2) foi desenvolvido um script no Dynamo, configurado para automatizar o processo de detalhamento das armaduras no Revit. A eficácia do script foi avaliada por meio de comparações entre os quantitativos gerados pelo Revit e os resultados obtidos através do método tradicional em CAD, considerando critérios de precisão, tempo de execução e otimização de materiais. Os resultados (3) foram analisados para identificar os benefícios e as limitações da abordagem proposta.

1.4 Estrutura da dissertação

A dissertação está organizada em cinco capítulos, cada um focado em uma parte específica do desenvolvimento e aplicação do trabalho.

No **Capítulo 1**, apresenta-se a *Introdução* ao tema, onde são delineados o enquadramento da pesquisa, os objetivos principais, a abordagem metodológica adotada e a estrutura da dissertação. Esse capítulo estabelece as bases para a compreensão do problema estudado e a relevância da implementação do BIM e da automação de processos no contexto da engenharia estrutural.

O **Capítulo 2** é dedicado à *Revisão de Literatura*, onde são explorados os principais conceitos e teorias que fundamentam o trabalho. São discutidos temas como o *Building Information Modeling*, as suas vantagens e os seus desafios, a importância da automação no detalhamento estrutural, e o uso de ferramentas como o Revit e o Dynamo. Este capítulo contextualiza a pesquisa dentro do estado da arte, oferecendo uma visão das tecnologias mais relevantes para o estudo.

No **Capítulo 3**, intitulado *Metodologia da Automação do Detalhamento das Consolas*, detalha-se o processo de desenvolvimento do script no Dynamo e sua integração com o Revit. São descritas as etapas de criação e implementação do script, bem como os critérios utilizados para a avaliação de sua eficácia. Além disso, são apresentados os procedimentos adotados para a coleta e análise dos dados.

O **Capítulo 4** apresenta um *Estudo de Caso*. Nesse capítulo, os resultados obtidos a partir da aplicação do script desenvolvido são analisados. São feitas comparações entre os quantitativos gerados pelo Revit e aqueles obtidos pelo método tradicional em CAD, com foco na precisão, no

tempo de execução e na otimização do uso de materiais. A discussão aborda as implicações dos resultados para a prática da engenharia estrutural e para a adoção do BIM no projeto estudado.

Finalmente, no **Capítulo 5**, são apresentadas as *Conclusões e Sugestões para Trabalhos Futuros*. Esse capítulo sintetiza os principais resultados do estudo, destacando as contribuições da automação para a melhoria da qualidade e da eficiência dos projetos estruturais. Também são discutidas as limitações do trabalho e sugeridas possíveis direções para pesquisas futuras, com o objetivo de expandir e aprofundar os conhecimentos desenvolvidos nesta dissertação.

2 REVISÃO DE LITERATURA

Esta seção tem como objetivo contextualizar e fundamentar teoricamente a pesquisa, apresentando as principais definições, benefícios e aplicações práticas do BIM, bem como os princípios envolvidos no detalhamento de armaduras de consolas pré-fabricados de betão. Além disso, busca fornecer uma base teórica para sustentar a metodologia e as análises subsequentes deste trabalho.

2.1 Definição do BIM

Conforme o *National Institute of Building Sciences* (NIBS) citado por Eastman [4] a Modelagem de Informação da Construção (BIM) é definida como um “processo aprimorado de planejamento, projeto, construção, operação e manutenção de uma edificação. Nesse processo é utilizado um modelo, padronizado e legível em qualquer programa, seja ele antigo ou novo, que contempla todas as informações coletadas durante a vida útil do edifício” (tradução própria).

Segundo Eastman [4], um modelo de construção deve ser consolidado sobre quatro aspectos, sendo eles: (1) objetos paramétricos com atributos gráficos e informação; (2) parâmetros que descrevem o comportamento do objeto; (3) informações consistentes e não redundantes que podem ser acessadas em qualquer vista; e por fim, (4) as informações devem ser atribuídas por coordenadas. Dessa forma, segundo Venâncio [5], softwares/modelos de representação tridimensional, que não seguem tais princípios, não são considerados BIM

Dentre os benefícios observados pelos usuários de BIM, segundo Stanley e Thurnell [6], destacam-se: facilidade de visualização de projetos e de detalhes construtivos, aumento da colaboração entre a equipa e facilidade de identificação de riscos e inconsistência em fases iniciais de projeto. Além disso, conforme Melhado e Pinto [7], o BIM aumenta a capacidade de correção de falhas e de identificação de dificuldades construtivas no projeto, pois o modelo funciona como uma construção virtual do edifício.

2.1.1 Histórico do BIM

De acordo com Venâncio [5], o primórdio do BIM tem sua origem nos anos 1960 com a publicação de *Augmenting Human Intellect*, de Douglas C. Englebart. Nessa publicação o autor introduz uma nova forma de modelar e projetar a partir de objetos parametrizados e integrados a uma base de dados. Entretanto, foi apenas durante a década de 1970, com o desenvolvimento dos métodos de criação de geometria *Constructive Solid Geometry* (CSG) e *Boundary Representation* (B-rep), que Eastman lança o software *Building Description System*, o qual tinha por base a utilização de uma biblioteca de elementos.

O termo *Building Information Modeling* (Modelagem de Informação da Construção) foi utilizado pela primeira vez em um artigo publicado por Nederveen e Tolman em 1992. A partir dele, segundo Venâncio [5], diversas empresas e acadêmicos passaram a adotar o termo para descrever uma variedade de conceitos. Entretanto, foi apenas no início dos anos 2000, com o aprimoramento de softwares de modelagem e de projeto e com o desenvolvimento dos Mandatos BIM, que a adoção da metodologia começou a se intensificar.

2.1.2 Interoperabilidade dos softwares

Segundo Eastman [4], a interoperabilidade refere-se à capacidade dos diferentes softwares em trocar e utilizar informações de maneira eficiente, permitindo que diversas plataformas colaborem sem perda de dados ou funcionalidade. Eastman [4], afirmam que inicialmente a interoperabilidade era limitada à troca de arquivos de geometria, como a extensão DXF, mas que a partir dos anos 1980, modelos de informação foram criados para a troca de modelos de objetos e de produtos entre diferentes indústrias, sendo o principal, o formato IFC (*Industry Foundation Classes*).

O IFC é um formato neutro, não proprietário, desenvolvido pela *Industrial Alliance of Interoperability* (IAI) na metade dos anos 1990. Ele surgiu com o intuito de permitir a interoperabilidade

entre softwares a partir da criação de uma base de dados universal para melhoramento de processos e compartilhamento de informação para o gerenciamento de construções e de indústrias [8]. Esse padrão é amplamente aceito na indústria e ajuda a assegurar que modelos criados em um software possam ser abertos e editados em outro, mantendo a consistência das informações.

Zhang et al. [9] apontam que a interoperabilidade não se limita apenas ao compartilhamento de modelos tridimensionais. Ela também abrange a troca de dados de propriedades dos elementos construtivos, cronogramas, custos e outras informações utilizadas para a gestão do projeto. Além disso Grilo e Jardim-Goncalves [8], propõem que para aplicar a interoperabilidade na era digital as organizações precisam lidar não só com as questões tecnológicas de integração de sistemas e aplicações, como também com a conexão entre os procedimentos e os processos de negócio de cada organização.

Com a necessidade de transmitir a informação sobre os procedimentos internos da organização, de modo a garantir a interoperabilidade total dos sistemas e processos da empresa, surgem os Manuais de Entrega de Informação (IDM), que consistem em construir uma metodologia para auxiliar na implantação do BIM [8]. Dessa forma, o IDM integra os processos de negócio enquanto padroniza, de forma detalhada, os procedimentos necessários para os usuários executarem as tarefas de um projeto [8].

2.1.3 Níveis de desenvolvimento

A maturidade de implantação do BIM, conforme Seidu et al. [10], é avaliada numa escala de zero até três. O nível de 0, chamado de Pré-BIM, que consiste em desenhos 2D, feitos em CAD, nesse nível a informação é armazenada escrita em documentos. O nível 1 consiste na migração do 2D para o 3D e na utilização dos conceitos de modelagem baseada em objetos. Já o nível 2 compreende o estágio de migração da modelação 3D individual para a colaboração e interoperabilidade. E por fim, o nível 3 é a transição da colaboração para a integração.

As especificações dos Níveis de Desenvolvimento (LOD) em BIM representam a quantidade e a precisão das informações contidas em um modelo BIM. Segundo a *BIM Fórum* [11], os LOD são divididos em diferentes graus que vão do LOD 100 até o LOD 500, sendo que cada um especifica um estágio de desenvolvimento do modelo e das informações apresentadas nesse.

- LOD 100: É a representação mais básica e conceitual do modelo, onde apenas a massa e a forma do edifício são indicadas, sem detalhes específicos sobre os elementos construtivos. Geralmente é o nível utilizado nas fases iniciais de planejamento e concepção do projeto [12].
- LOD 200: O modelo inclui elementos genéricos com informações básicas sobre dimensões e localizações aproximadas. Os elementos são representados de forma mais detalhada do que no LOD 100, mas ainda podem não ser precisos ou específicos. Este nível é frequentemente usado durante o desenvolvimento preliminar do projeto [12].
- LOD 300: Possui um aumento significativo na precisão e nos detalhes do modelo. Neste nível, os elementos são representados com tamanhos, formas, localizações e orientações precisas [12]. De acordo com o *BIM Fórum* [11], este nível de detalhe é adequado para a fase de documentação do projeto, onde os desenhos e especificações são finalizados.
- LOD 400: Incorporando detalhes específicos de fabricação e montagem dos elementos construtivos [12]. Este nível é importante para a fase de construção, pois permite que os empreiteiros e fabricantes obtenham as informações necessárias para a produção e montagem dos componentes do edifício [11].
- LOD 500: Representa o modelo “como construído” (as-built), em que documenta o estado final da construção, incluindo todas as modificações e ajustes feitos durante a execução [12]. Este nível é utilizado para a gestão das operações e manutenção da edificação após a conclusão do projeto [11].

Para auxiliar no entendimento dos diferentes níveis de desenvolvimentos de um projeto, eles são representados graficamente na Figura 2.1, onde o LOD 500 não é representando, pois o mesmo possui um enfoque na gestão do projeto e não em seu detalhamento.

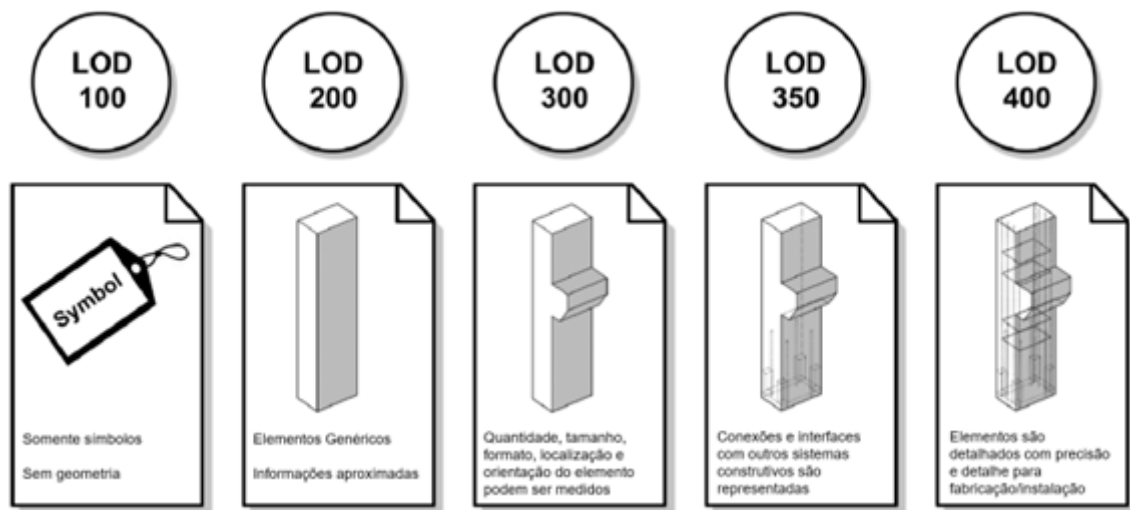


Figura 2.1: Diferentes níveis de desenvolvimento em modelos BIM, adaptado de Mekawy e Petzold [13].

2.1.4 As normas do BIM

As normas regulamentadoras do BIM são diretrizes estabelecidas por organizações de normalização reconhecidas internacionalmente para garantir a consistência e a interoperabilidade na utilização do BIM [14]. Segundo Ibercard [15], a metodologia BIM é regida pelas seguintes normas internacionais:

- ISO 19650-1: 2018 – Esta norma fornece diretrizes para a gestão da informação em projetos BIM, incluindo a organização e digitalização de informações sobre obras de construção e engenharia civil;
- ISO 21597-1: 2020 – Define um formato padronizado para contêineres de informações usados em processos BIM, assegurando a interoperabilidade e troca eficiente de dados entre diferentes softwares;
- ISO 29481-1: 2016 – Esta norma facilita a comunicação entre aplicativos de software durante todo o ciclo de vida de um projeto de construção, promovendo a interoperabilidade e a colaboração eficiente.

Em Portugal, a Comissão Técnica de Normalização BIM Nacional [16], lançou, um documento intitulado “Regras de modelação de objetos BIM”. O documento destaca a necessidade de uma classificação adequada dos objetos BIM e a importância de um manual de utilização com nomenclatura consistente para garantir a interoperabilidade. Além disso, aborda a representação da informação geométrica e não geométrica, visando assegurar que os objetos BIM sejam úteis em diferentes fases do ciclo de vida de edifícios e infraestruturas.

Além disso, conforme Silva [17], existem três iniciativas que visam auxiliar a implementação de processos BIM em Portugal. Sendo elas: Projeto SIGABIM, BIM Fórum Portugal e a Plataforma Tecnológica Portuguesa de Construção.

2.2 Revit

O Revit é uma plataforma de projetos baseados na metodologia BIM. Sendo assim, ela consegue agrupar e extrair todas as informações necessárias para um projeto, como por exemplo: a tabela de quantidades, o faseamento do projeto e o detalhamento das peças [18]. Além disso, segundo Vieira e Figueiredo [19], o Revit facilita a coordenação entre diferentes disciplinas de engenharia, permitindo que arquitetos, engenheiros e construtores colaborem de forma integrada ao longo de todas as fases do projeto.

Segundo Eastman [4] um modelo BIM, deve ser consolidado sobre três princípios, sendo eles: precisão geométrica, dados paramétricos e coordenação multidisciplinar. Dessa forma, o Revit possui recursos computacionais que permite que esses princípios sejam atendidos. Primeiramente, a precisão geométrica é garantida por objetos paramétricos que permitem uma representação fiel dos componentes de construção. Em segundo lugar, os dados paramétricos descrevem o comportamento dos objetos e suas interações dentro do modelo. Finalmente, a coordenação multidisciplinar é facilitada pela interoperabilidade do Revit com outros softwares e formatos de dados, assegurando que todas as informações sejam consistentes e acessíveis em diferentes vistas e documentos.

2.2.1 Famílias e parâmetros

As famílias no Revit são conjuntos de elementos que compartilham características semelhantes e podem incluir elementos como paredes, portas, janelas, móveis, entre outros [20]. Cada família pertence a um grupo e é formada por diversos tipos. Para compreender melhor a hierarquia das famílias no Revit, é apresentado o fluxograma na Figura 2.2 e sua representação gráfica na Figura 2.3.

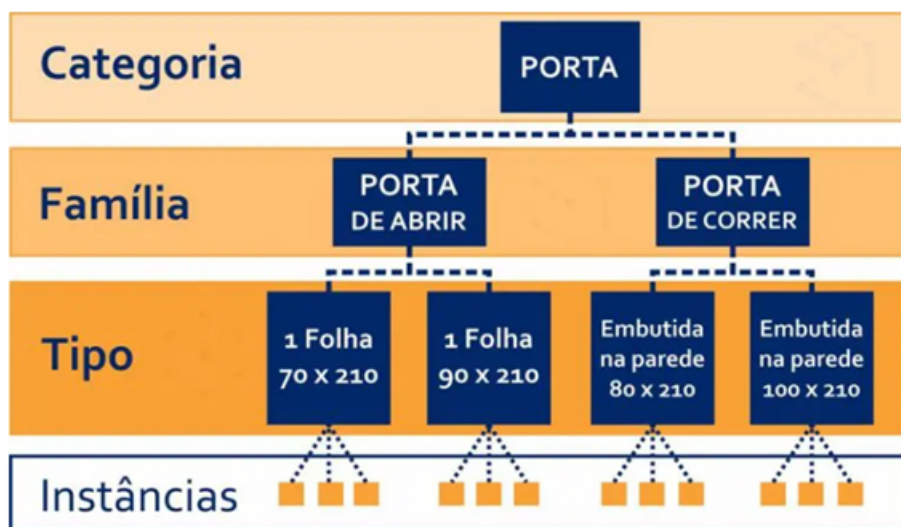


Figura 2.2: Estrutura hierárquica de categorias, famílias, tipos e instâncias (Beraldi [21]).

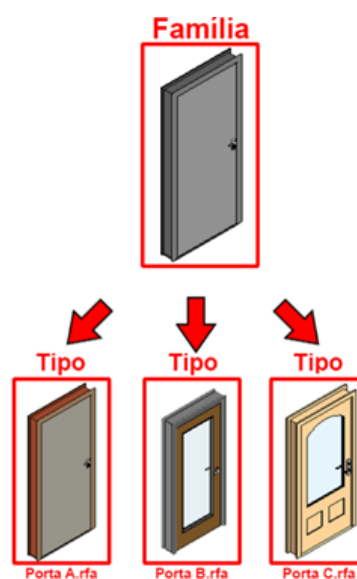


Figura 2.3: Representação gráfica da estrutura hierárquica de famílias e tipos no Revit (Santos [22]).

Os parâmetros, por sua vez, são atributos ajustáveis que controlam o comportamento e a aparência das famílias no Revit. Sendo que eles podem ser de dois tipos principais: parâmetros de tipo e parâmetros de instância. Os parâmetros de tipo afetam todos os elementos de uma determinada família, enquanto os parâmetros de instância permitem ajustes individuais em cada instância da família dentro do modelo.

A utilização eficaz de famílias e parâmetros no Revit oferece uma série de benefícios aos usuários, incluindo maior flexibilidade de design, padronização de elementos recorrentes e redução do tempo de modelagem. Além disso, os parâmetros são fundamentais para que os dados do projeto possam ser convertidos em informações, de forma a permitir o correto fluxo de trabalho em BIM.

2.2.2 Grupos e montagens

Grupos e montagens são ferramentas do Revit que facilitam a organização e a reutilização de elementos do modelo. Essas ferramentas são utilizadas para aumentar a eficiência e garantir a consistência em projetos de construção [23].

Grupos são coleções de elementos que podem ser agrupados para serem manipulados como uma única entidade. Eles são úteis para elementos que se repetem em várias partes do projeto, como conjuntos de móveis, unidades de habitação repetitivas ou elementos estruturais [23]. Uma vez criado, um grupo pode ser instanciado várias vezes no modelo, e qualquer modificação feita em um elemento do grupo será refletida automaticamente em todas as suas instâncias, garantindo consistência e economizando tempo de edição [24].

Montagens, por outro lado, são coleções de elementos que são agrupados para facilitar a documentação do projeto. As montagens são criadas quando há a necessidade de apresentar uma parte específica do projeto como uma unidade única, como um pilar pré-fabricado ou uma seção estrutural específica. As montagens permitem a criação de vistas detalhadas, tabelas de quantidades e desenhos específicos para essa coleção de elementos, sem afetar os elementos fora da montagem [25].

2.2.3 Criação de tabelas

Este processo envolve a seleção dos elementos do modelo que se deseja quantificar e a definição dos parâmetros específicos que serão incluídos na tabela. Os usuários podem personalizar as tabelas de acordo com as necessidades do projeto, especificando filtros, classificações e agrupamentos para organizar os dados de forma coerente e útil. As tabelas de quantidades são geradas automaticamente a partir das informações inseridas no modelo, garantindo que estejam sempre atualizadas e consistentes com quaisquer mudanças no projeto [26].

Para facilitar a extração e utilização dos dados, as informações no Revit devem ser organizadas e padronizadas de maneira consistente. Isso inclui o uso de nomenclaturas padronizadas para elementos e parâmetros, a definição clara de categorias e subcategorias de elementos, e a consistência na aplicação de propriedades e atributos. A padronização dos dados facilita a criação de tabelas de quantidades, e garante que todos os membros da equipa de projeto tenham acesso a informações uniformes [27].

2.3 Dynamo

O Dynamo é uma ferramenta visual de programação que opera dentro do software Revit, utilizado para automatizar processos e criar algoritmos personalizados no desenvolvimento de projetos arquitetônicos e de engenharia [28]. Ele permite aos usuários criar scripts visuais para manipular dados, gerar geometrias complexas e automatizar tarefas repetitivas [28]. O Dynamo foi introduzido pela primeira vez em 2011 como uma aplicação de código aberto, e sua integração com o Revit ocorreu em 2013 [29]. Desde então, ele tem se mostrado útil em diversas etapas do projeto, como na modelagem paramétrica, análise de dados, e interoperabilidade entre diferentes softwares de design.

Dentro do Dynamo é possível utilizar pacotes, de maneira a expandir as capacidades nativas do software [30]. Os pacotes são coleções de nós, scripts, bibliotecas e ferramentas desenvolvidas pela comunidade de usuários ou por desenvolvedores terceiros, disponibilizadas através do Dynamo Package Manager. A instalação de pacotes pode simplificar tarefas complexas, como a criação de geometrias avançadas, a conexão com bancos de dados externos, e a integração com outros softwares de design e análise [30].

2.4 Python

Python é uma linguagem de programação de alto nível, interpretada e de propósito geral, conhecida por sua sintaxe clara e legibilidade [31]. Criada por Guido van Rossum e lançada em 1991, Python é amplamente utilizada em diversas áreas devido à sua gratuidade, portabilidade, potência e facilidade de utilização [31]. Dentro do Dynamo, o Python é utilizado através do nó “Python Script”, que permite aos usuários escrever scripts em Python para executar tarefas que não são facilmente realizadas com os nós padrão do Dynamo [30].

2.5 Consolas curtas de betão

De acordo com El Debs [32], as estruturas pré-fabricadas destacam-se pelo rigor no controle de qualidade e pela facilidade de execução. No entanto, a necessidade de realizar ligações entre os elementos estruturais (como ligações viga-pilar) é um dos principais desafios na aplicação de estruturas pré-fabricadas. Essas ligações têm a função de transmitir os esforços atuantes em todas as fases de utilização e ao longo da vida útil da estrutura. Em particular, as ligações viga-pilar representam um dos maiores problemas na utilização de estruturas pré-fabricadas, pois ligações articuladas tendem a aumentar os momentos fletores nos elementos conectados (como vigas e lajes). Por outro lado, o uso de ligações semirrígidas nas conexões viga-pilar ajuda a reduzir os momentos fletores nos vãos das vigas e proporciona maior rigidez à estrutura, o que é essencial para garantir sua estabilidade global. No entanto, essas ligações também tornam o processo de montagem mais complexo.

Em projetos de ligações viga-pilar é usual a utilização de consolas de betão armado, as quais possuem critérios de dimensionamento e detalhamento semelhantes às consolas dimensionadas em betão moldado *in loco*. Assim, os mesmos métodos para definição de comprimentos de ancoragem e emendas das barras de aço podem ser aplicados, devendo prevalecer o preconizado pelas normas brasileiras Normas Técnicas (ABNT) [2] e Normas Técnicas (ABNT) [3] ou ainda pela norma europeia Standardization [1]. Segundo Cedrim et al. [33], o dimensionamento da ligação pode ser feito através dos componentes básicos envolvidos na transferência de esforços, ou seja, os esforços são transferidos da viga para o dente gerber e em seguida para a consola através da almofada de apoio e do chumbador.

Sendo assim, uma das formas de fazer a ligação viga-pilar entre elementos pré-fabricados é através da utilização de consolas, que servem de apoio para outras peças que compõem a estrutura. Eles geralmente são utilizados nas ligações viga-pilar e viga-viga, mas também podem ser utilizados em ligações piso-parede [34]. Dessa forma, a Figura 2.4, representa algumas formas de ligação viga-pilar em elementos pré-fabricados de betão.

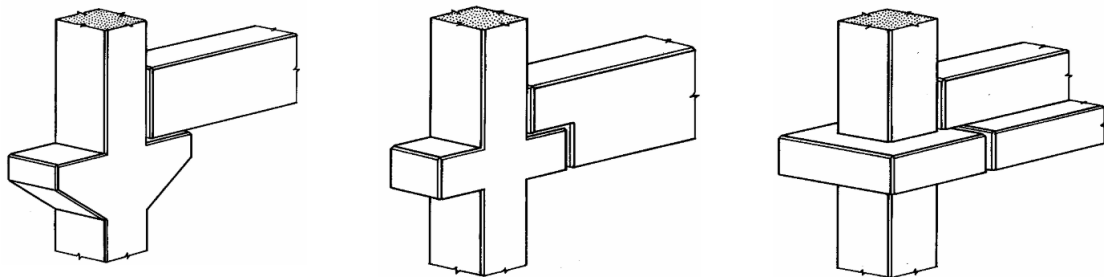


Figura 2.4: Formas de ligação viga-pilar em elementos de pré-fabricados betão (Van Acker [34]).

2.5.1 Classificação das consolas

De acordo com a Normas Técnicas (ABNT) [2], as consolas são elementos em balanço nos quais a distância da carga aplicada (a) até à face do apoio é menor ou igual à altura útil (d) da consola. Além disso, as normas brasileiras Normas Técnicas (ABNT) [2] e Normas Técnicas (ABNT) [3] classificam as consolas em duas categorias: consolas muito curtas (relação $a/d \leq 0,5$) e consolas curtas ($0,5 < a/d \leq 1,0$). Os parâmetros são apresentados na Figura 2.5.

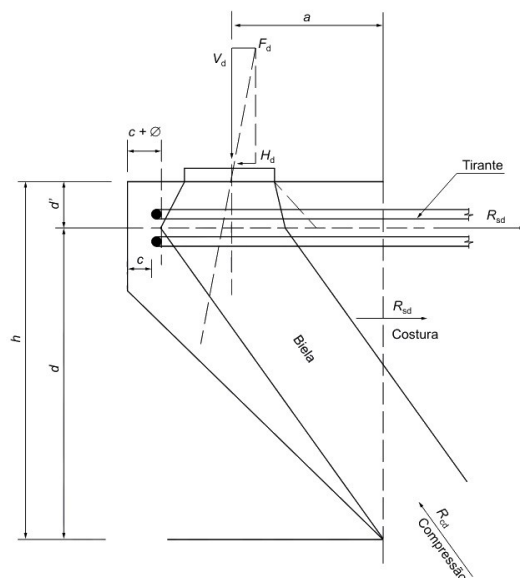


Figura 2.5: Modelo de biela-tirante para consola curta segundo a NBR 9062:2017, adaptado de Normas Técnicas (ABNT) [3].

Em relação à classificação portuguesa, o Standardization [1] define como consolas curtas os elementos em balanço nos quais a distância da carga aplicada até à face do apoio (a_c) é menor que o braço do binário das forças interiores (z_0), ou seja, ($a_c < z_0$). A Figura 2.6 ilustra essa definição.

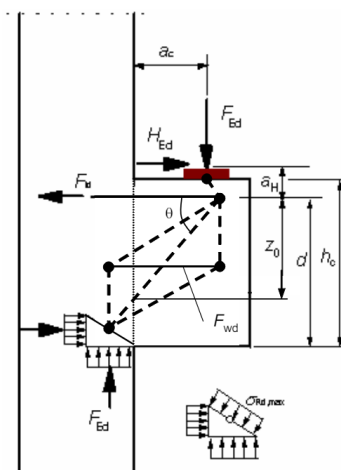


Figura 2.6: Modelo de consola curta segundo o Anexo J do Eurocódigo 2 (Standardization [1]).

2.5.2 Modos de ruptura

De acordo com Santos [35], existem três modos principais de ruptura de consolas curtas: deformação excessiva da armadura do tirante, fissuração diagonal com esmagamento do betão e corte direto. Além desses, o autor menciona outros dois modos de ruptura secundários, ruína de borda

devido à falha de ancoragem e ruína por esforço horizontal não previsto, que devem ser evitados com o detalhamento correto das armaduras.

2.5.2.1 Ruptura por deformação excessiva do tirante

Esse tipo de ruptura ocorre devido ao esmagamento da face inferior da consola, após um grande escoamento da armadura do tirante. Esse modo de ruptura é caracterizado pela abertura de fissuras de flexão, enquanto há controle da abertura de fissuras diagonais [35]. Santos [35] afirma que essa é a melhor forma de ruptura do ponto de vista da segurança, pois o alto grau de fissuração permite a realização de intervenções antes do colapso do elemento. A Figura 2.7, apresenta abertura de fissuras desde modo de ruptura.

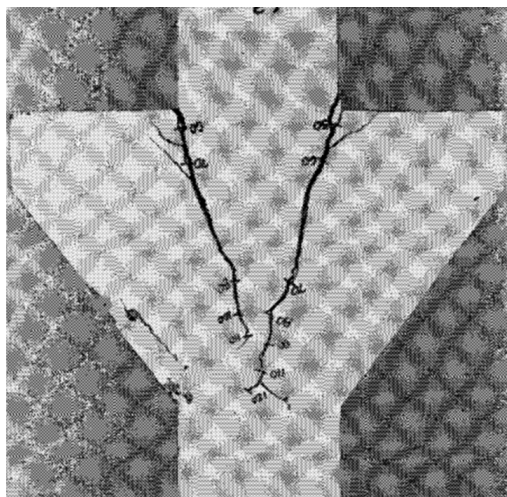


Figura 2.7: Ruptura de consola devido à tração na flexão (Kriz e Rathes [36]).

2.5.2.2 Ruptura por tração diagonal

Em consolas que são armadas sem estribos, geralmente ocorre a formação de uma fissura diagonal, que se inicia na placa de apoio e segue até a interseção da consola com o pilar [35]. Nesses casos, ocorre uma ruptura brusca devido ao cisalhamento nas extremidades dessas fissuras. A Figura 2.8 apresenta o padrão de fissuramento deste tipo de ruptura.

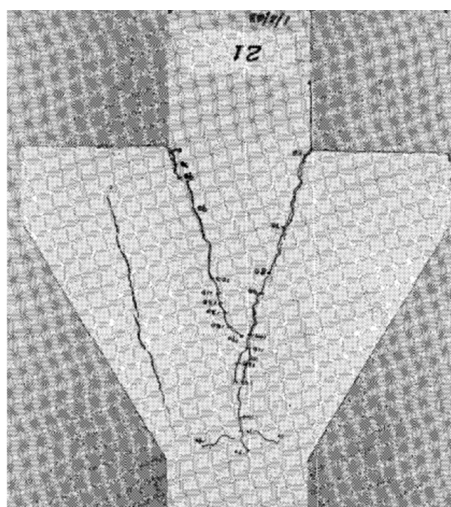


Figura 2.8: Ruptura de consola devido à tração diagonal (Kriz e Rathes [36]).

2.5.2.3 Ruptura por corte direto

A ruptura devido ao corte direto ocorre por cisalhamento ao longo da interface entre a consola e o pilar. Neste modo de ruína surgem fissuras de pequenas dimensões ao longo dessa interface [35]. A Figura 2.9 mostra a abertura de fissuras neste tipo de ruptura.

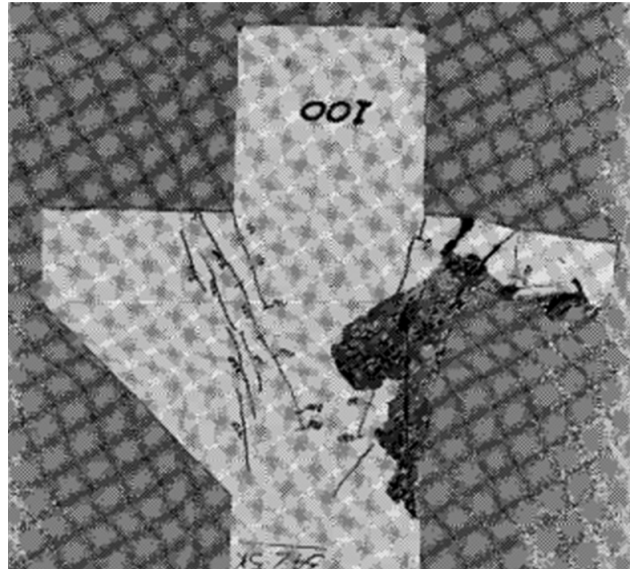


Figura 2.9: Ruptura de consola devido ao corte direto (Kriz e Rathes [36]).

2.5.2.4 Ruptura por compressão na flexão

A ruptura por compressão na flexão ocorre quando o betão é esmagado antes do escoamento da armadura do tirante, desenvolvendo-se fissuras de flexão. No entanto, essas fissuras apresentam uma abertura de forma menos significativa do que na ruptura por tração na flexão, tornando-a um modo de ruptura menos seguro que o anterior [35]. A Figura 2.10 representa a abertura de fissuras devido à ruptura por compressão na flexão.

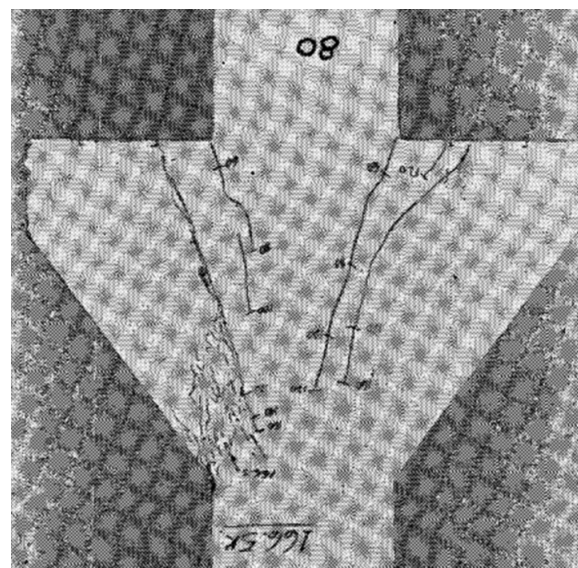


Figura 2.10: Ruptura de consolas devido à compressão na flexão (Kriz e Rathes [36]).

2.5.2.5 Ruptura por arrancamento de borda

De acordo com Santos [35], a ruptura por arrancamento de borda ocorre devido à falta de armadura na extremidade da consola, sendo caracterizada pelo arrancamento do betão na ponta da consola, conforme mostrado na Figura 2.11. Isso geralmente acontece devido à ausência de barras de aço na ponta do consolo, uma vez que a utilização de ganchos verticais em varões de diâmetros muito elevados exige um raio de curvatura demasiadamente grande na execução da dobra, como exemplificado na Figura 2.12 [35]. Por conta disso, a norma brasileira Normas Técnicas (ABNT) [3] recomenda a utilização de laços horizontais ou barras horizontais soldadas quando o diâmetro do varão é igual ou superior a $16mm$, conforme indicado na Figura 2.15.

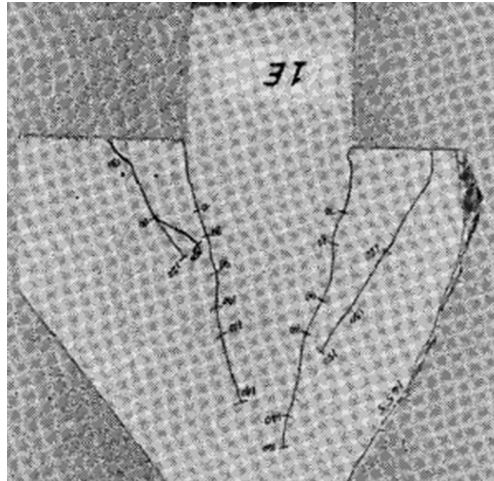


Figura 2.11: Arrancamento de borda em consola curta (Kriz e Rathes [36]).

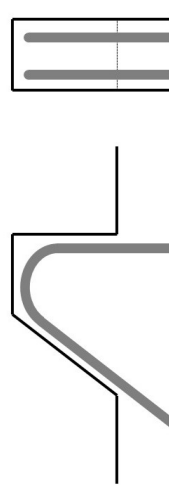


Figura 2.12: Detalhamento de gancho vertical com raio de curvatura elevado (Kriz e Rathes [36]).

2.6 Modelo de cálculo de consolas curtas

O modelo de cálculo de consolas curtas fundamenta-se no método de escoras e tirantes, uma abordagem utilizada para analisar e dimensionar regiões de descontinuidade em elementos estruturais. Esse modelo idealiza o comportamento estrutural por meio das trajetórias de tensões principais de compressão (escoras) e tração (tirantes). As primeiras noções sobre o comportamento de consolas curtas foram apresentadas por Leonhardt e Mönnig [37], que analisaram as trajetórias de tensões em consolas com relação $a/h = 0,5$, conforme ilustrado na Figura 2.13.

Leonhardt e Mönnig [37] chegaram a algumas conclusões ao analisar a trajetória de tensões em consolas curtas: a região inferior da consola é pouco solicitada, de forma que a presença de um chanfro não afeta sua resistência; na parte superior da consola, as linhas isostáticas são praticamente horizontais e apresentam tensão constante, o que torna necessário o uso de armaduras próximas a essa face; a biela de compressão origina-se no ponto de aplicação da força e se estende até a base da consola; e, por fim, as tensões de tração remanescentes devem ser absorvidas por estribos horizontais, pois estribos verticais são ineficazes nesse contexto.

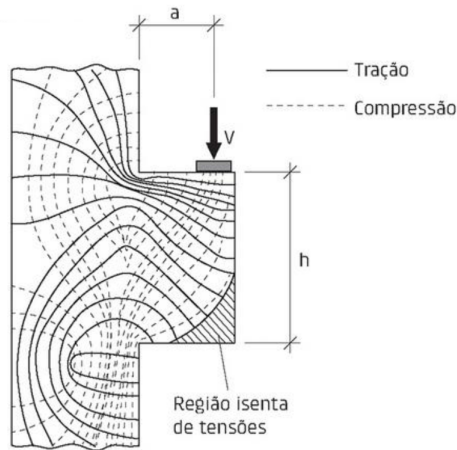


Figura 2.13: Linha isostática de tensões em consolas que respeitam a relação $a/h = 0,5$ (adaptado de Leonhardt e Mönnig [37]).

2.6.1 Eurocódigo 2

O Eurocódigo 2 é uma norma europeia de projeto de estruturas de betão que estabelece diretrizes para o dimensionamento e detalhamento de elementos estruturais. Ele foi publicado pela primeira vez em 1992 e foi revisado com o decorrer dos anos, sendo sua revisão mais recente publica em 2010. O Eurocódigo 2 é adotado em diversos países da União Europeia, entre eles Portugal [1]. Em relação ao dimensionamento de consolas curtas, o Standardization [1], recomenda a utilização de modelos de bielas e tirantes, conforme indicado na Figura 2.6.

2.6.2 ABNT NBR 6118

A ABNT NBR 6118 é a norma brasileira que estabelece os critérios para o projeto de estruturas de betão armado e pré-esforçado. Em relação aos modelos de cálculo que devem ser adotados para o dimensionamento de consolas curtas, a Normas Técnicas (ABNT) [2] sugere a utilização de modelos planos lineares, modelos de bielas e tirantes, ou modelos de atrito-cisalhamento, devendo respeitar os campos de aplicação de cada modelo. Além disso, a Normas Técnicas (ABNT) [2] informa que os modelos de bielas e tirantes são usualmente utilizados no dimensionamento de consolas curtas, enquanto os modelos de atrito-cisalhamento são geralmente utilizados para o cálculo de consolas muito curtas, sendo que, independentemente do modelo utilizado, o comportamento estrutural da consola precisa seguir o que é especificado na Seção 22.5.1.2 da Normas Técnicas (ABNT) [2].

2.6.3 ABNT NBR 9062

A Normas Técnicas (ABNT) [3] é a norma brasileira que estabelece os critérios para o projeto e a execução de estruturas de betão pré-moldado, funcionando como um complemento da Normas Técnicas (ABNT) [2]. Tal como a Normas Técnicas (ABNT) [2], a Norma Brasileira 9062 recomenda que consolas curtas sejam dimensionadas pelos modelos de bielas e tirantes, enquanto que para consolas muito curtas, o dimensionamento deve ser feito supondo a ruptura entre o plano de ligação da consola com o suporte, ou seja, conforme os modelos de atrito-cisalhamento.

2.6.4 Parâmetros de cálculo do modelo de escoras e tirantes

Tanto no Standardization [1] quanto na Normas Técnicas (ABNT) [2], os valores de resistência das escoras são calculados considerando fatores como a resistência do betão à compressão. A determinação adequada desses parâmetros é essencial para evitar a ruptura prematura da consola, garantindo que as escoras sejam dimensionadas para suportar as cargas aplicadas. Os parâmetros de resistência das bielas podem variar entre as normas, dessa forma, a Tabela 2.1 apresenta uma comparação entre os parâmetros apresentados no Standardization [1] e na Normas Técnicas (ABNT) [2].

Tabela 2.1: Resistência das escoras pelo métodos das bielas e tirantes, adaptado de Costa [38].

	Eurocódigo 2	NBR 6118
Valor de cálculo da resistência das escoras de betão na ausência de trações transversais	$\sigma_{Rd,max} = f_{cd}$	$f_{cd1} = 0,85 \cdot \alpha_{V2} \cdot f_{cd}$ Sendo: $\alpha_{V2} = 1 - \frac{f_{ck}}{250}$ p/ escoras prismáticas
Valor de cálculo da resistência das escoras de betão sujeitas a tração transversal	$\sigma_{Rd,max} = 0,6 \cdot \nu' \cdot f_{cd}$ $\nu' = 1 - \frac{f_{ck}}{250}$	$f_{cd2} = 0,60 \cdot \alpha_{V2} \cdot f_{cd}$ p/ escoras atravessadas por mais de um tirante $f_{cd3} = 0,72 \cdot \alpha_{V2} \cdot f_{cd}$ p/ escoras atravessadas por um tirante único

Onde:

$\sigma_{Rd,max}$	Valor de cálculo máximo da resistência à compressão de uma escora de concreto
ν'	Coefficiente de redução da resistência do betão fendilhado por esforço transversal
α_{V2}	Fator de redução utilizado na verificação da compressão diagonal do concreto em bielas comprimidas
f_{ck}	Valor característico da tensão de ruptura do betão à compressão
f_{cd}	Valor de cálculo da tensão de ruptura do betão à compressão;
f_{cd1}	Tensão resistente máxima no betão, em verificações pelo método de escoras e tirantes, em regiões com tensões de compressão transversal ou sem tensões de tração transversal e em nós onde confluem somente escoras de compressão
f_{cd2}	Tensão resistente máxima no betão, em verificações pelo método de escoras e tirantes, em regiões com tensões de tração transversal e em nós onde confluem dois ou mais tirantes tracionados
f_{cd3}	Tensão resistente máxima no betão, em verificações pelo método de escoras e tirantes, em nós onde conflui um tirante tracionado

Costa [38] menciona que outro parâmetro usualmente diferente entre as normas e que pode afetar a aplicabilidade do modelo é a inclinação das escoras. A Tabela 2.2, indica o intervalo de inclinação admitido pelo Standardization [1] e pela Normas Técnicas (ABNT) [2].

Tabela 2.2: Inclinação das escoras pelo método das bielas e tirantes, adaptado de Costa [38].

	Eurocódigo 2	NBR 6118
Inclinação das escoras	$26 \leq \theta \leq 45$	$30 \leq \theta \leq 63$

Conforme mencionado por Costa [38], no modelo de bielas e tirantes, os nós representam mudanças na direção ou interseções de forças, sendo uma simplificação da realidade. Assim como os parâmetros de resistência e inclinação das escoras, a resistência dos nós pode variar conforme a norma utilizada. Neste sentido, a Tabela 2.3 compara os valores de resistência dos nós segundo o Standardization [1] e a Normas Técnicas (ABNT) [2].

Tabela 2.3: Resistência dos nós pelo método das bielas e tirantes, adaptado de Costa [38].

	Eurocódigo 2	NBR 6118
Nós comprimidos, sem tirantes amarrados no nó	$\sigma_{Rd,max} = 1,00 \cdot \nu' \cdot f_{cd}$	$f_{cd1} = 0,85 \cdot \alpha_{V2} \cdot f_{cd}$
Nós sujeitos a compressão e tração, com tirantes amarrados em uma direção	$\sigma_{Rd,max} = 0,85 \cdot \nu' \cdot f_{cd}$	$f_{cd2} = 0,60 \cdot \alpha_{V2} \cdot f_{cd}$
Nós sujeitos a compressão e tração, com tirantes amarrados em mais de uma direção	$\sigma_{Rd,max} = 0,75 \cdot \nu' \cdot f_{cd}$	$f_{cd3} = 0,72 \cdot \alpha_{V2} \cdot f_{cd}$

2.7 Detalhamento de armaduras de consolas curtas

De acordo com o Standardization [1] e com suplemento da Normas Técnicas (ABNT) [3], as armaduras das consolas curtas devem ser detalhadas de modo a garantir a sua integridade e segurança estrutural. O Standardization [1] recomenda o uso de armaduras distribuídas de forma a resistir às forças de tração e compressão, com especial atenção à ancoragem adequada das armaduras, conforme indicado na Figura 2.14. A Normas Técnicas (ABNT) [3] também ressalta a importância de utilizar barras de aço soldadas ou com laços horizontais quando o diâmetro do varão for superior a 16 mm, para garantir a ancoragem na borda e evitar o arrancamento do betão, conforme indicado na Figura 2.15.

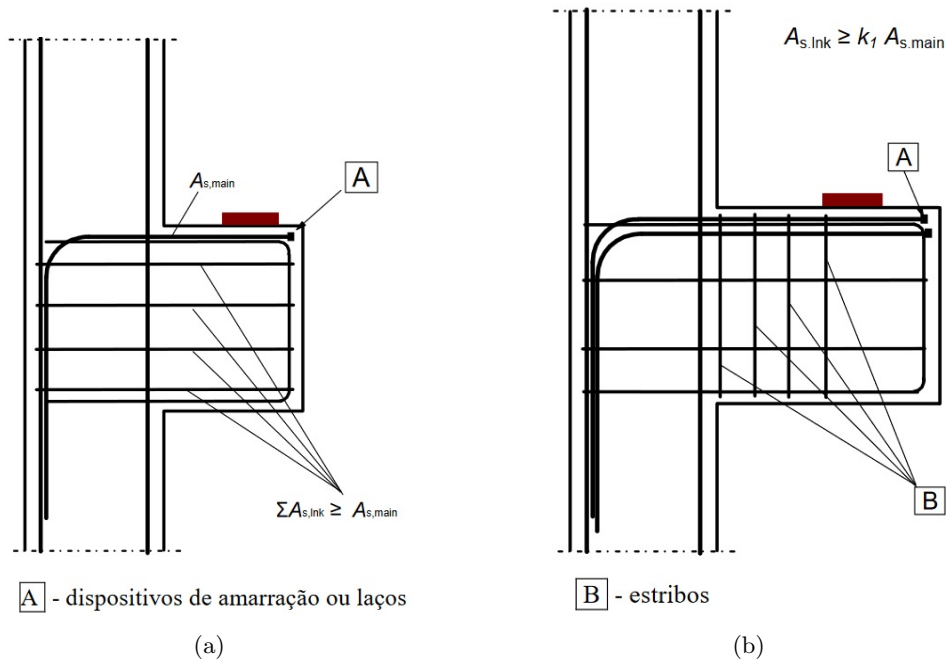


Figura 2.14: Detalhamento das armaduras de consolas curtas, segundo o Standardization [1]: (a) Armaduras para $a_c \leq 0,5h_c$; (b) Armaduras para $a_c > 0,5h_c$, adaptado de Standardization [1]

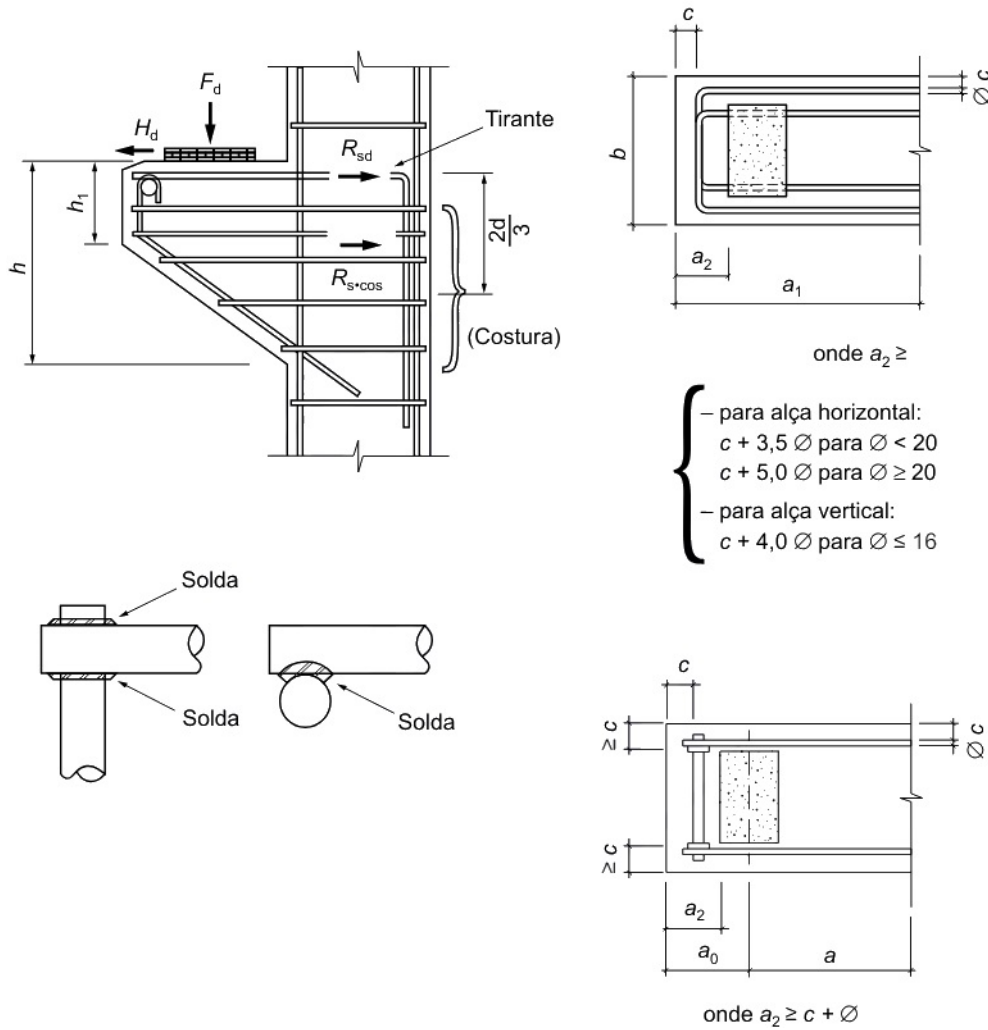


Figura 2.15: Armadura típica de consolas curtas, pelo padrão da NBR 9062:2017, adaptado de Normas Técnicas (ABNT) [3].

Além disso, tanto o Standardization [1] quanto a Normas Técnicas (ABNT) [3] recomendam a utilização de três tipos principais de armaduras: tirantes, costura e transversal (estribos). Entretanto, essas normas indicam formulações diferentes para o cálculo da quantidade de aço necessária em cada posição. Dessa forma, a Tabela 2.4 apresenta uma comparação entre os métodos de cálculo especificados na Normas Técnicas (ABNT) [3] e no Standardization [1].

Tabela 2.4: Cálculo das áreas das armaduras de consolas muito curtas, adaptado de Silva e Sampaio [39].

	Eurocódigo 2	NBR 9062
Armadura do tirante (cm^2)	$A_s = \frac{F_t}{f_{yd}}$	$A_{s,tir} = A_{sv} + \frac{H_d}{f_{yd}}$
Armadura de costura (cm^2)	$F_{wd} = \frac{2 \cdot \frac{z}{a} - 1}{3 + F_{Ed}/F_c} \cdot F_c$	$(A_s/s)_{cost} \geq 0,5 \cdot \frac{A_{sv}}{d}$
Armadura transversal (cm^2)	$A_{sw} = \frac{F_{wd}}{f_{yd}}$	$(A_s/s)_{estr,min} = 0,0015 \cdot b \cdot h$

Onde:

A_s Área da seção de uma armadura para betão armado

f_{yd}	Valor de cálculo da tensão de cedência à tração do aço das armaduras para betão armado
F_t	Força de tração
F_{wd}	Valor de cálculo da resistência ao corte da soldadura
F_{Ed}	Esforço de compressão na ligação
F_c	Força de compressão correspondente à força de tração máxima $F_{s,max}$
H_d	Carga horizontal do projeto
d	Altura útil do consolo
h	Altura do consolo
b	Largura do consolo
z	Braço de alavanca
a	Comprimento da biela de compressão

2.7.1 Durabilidade das estruturas de betão

De acordo com Vaz Rodrigues e Gutstein [40], a durabilidade de estruturas de betão é abordada pela Normas Técnicas (ABNT) [2] através da identificação dos mecanismos de deterioração. Os mecanismos são definidos a partir da classificação da agressividade ambiental, baseada nas condições de exposição da estrutura, conforme indicado na Tabela 2.5. A partir da definição da agressividade ambiental, a Normas Técnicas (ABNT) [2] correlaciona os requisitos mínimos de resistência e composição do betão, conforme indicado na Tabela 2.6, bem como os recobrimentos mínimos, conforme indicado na Tabela 2.7 [40].

Tabela 2.5: Classes de agressividade ambiental segundo a Normas Técnicas (ABNT) [2].

Classe de agressividade ambiental	Agressividade	Classificação geral do tipo de ambiente para efeito de projeto	Risco de deterioração da estrutura
I	Fraca	Rural Submersa	Insignificante
II	Moderada	Urbana ^{a,b}	Pequeno
III	Forte	Marinha ^a Industrial ^{a,b}	Grande
IV	Muito forte	Industrial ^{a,c} Respingos de maré	Elevado

^a Pode-se admitir um microclima com uma classe de agressividade mais branda (uma classe acima) para ambientes internos secos (salas, dormitórios, banheiros, cozinhas e áreas de serviço de apartamentos residenciais e conjuntos comerciais ou ambientes com betão revestido com argamassa e pintura).

^b Pode-se admitir uma classe de agressividade mais branda (uma classe acima) em obras em regiões de clima seco, com umidade média relativa do ar menor ou igual a 65%, partes da estrutura protegidas de chuva em ambientes predominantemente secos ou regiões onde raramente chove.

^c Ambientes quimicamente agressivos, tanques industriais, galvanoplastia, branqueamento em indústrias de celulose e papel, armazéns de fertilizantes, indústrias químicas.

Tabela 2.6: Correlação entre a classe de agressividade ambiental com os requisitos de qualidade do betão segundo a Normas Técnicas (ABNT) [2].

Betão	Tipo	Classe de agressividade			
		I	II	III	IV
Relação água / cimento em massa	CA	$\leq 0,65$	$\leq 0,60$	$\leq 0,55$	$\leq 0,45$
	CP	$\leq 0,60$	$\leq 0,55$	$\leq 0,50$	$\leq 0,45$
Classe de betão	CA	$\geq C20$	$\geq C25$	$\geq C30$	$\geq C40$
	CP	$\geq C25$	$\geq C30$	$\geq C35$	$\geq C40$

Notas:

1. O betão empregado na execução das estruturas deve cumprir com os requisitos estabelecidos na ABNT NBR 12655.
2. CA corresponde a componentes e elementos estruturais de betão armado.
3. CP corresponde a componentes e elementos estruturais de betão protendido.

Tabela 2.7: Correlação entre a classe de agressividade ambiental e o recobrimento nominal do betão, para $\Delta c = 10mm$, segundo a Normas Técnicas (ABNT) [2].

Tipo de estrutura	Componente ou elemento	Classe de agressividade ambiental			
		I	II	III	IV ^c
Betão armado	Laje ^b	20	25	35	45
	Viga / pilar	25	30	40	50
	Elementos estruturais em contato com o solo ^d	30	35	45	55
Betão pré-esforçado ^a	Laje	25	30	40	50
	Viga / pilar	30	35	45	55

Notas:

- a. Recobrimento nominal da bainha ou dos fios, cabos e cordoalhas. O recobrimento da armadura passiva deve respeitar os recobrimentos para betão armado.
- b. Para a face superior de lajes e vigas que serão revestidas com argamassas de contrapiso, com revestimentos finais secos tipo carpete e madeira, com argamassas de revestimento e acabamento, como pisos de elevado desempenho, pisos cerâmicos, pisos asfálticos e outros, as exigências desta Tabela podem ser substituídas pelas de 7.4.7.5, respeitando um recobrimento nominal ≥ 15 mm.
- c. Nas superfícies expostas a ambientes agressivos, como reservatórios, estações de tratamento de água e esgoto, condutas de esgoto, canaletas de efluentes e outras obras em ambientes químico e intensamente agressivo, devem ser respeitados os recobrimentos da classe de agressividade IV.
- d. No troço de pilares em contato com o solo junto com elementos de fundação, a armadura deve ter recobrimento nominal ≥ 45 mm.

Em relação ao especificado pelo Standardization [1], Vaz Rodrigues e Gutstein [40] explicam que ele define classes de exposição com base no tipo de degradação envolvida. São três classes associadas à ação de cloretos, dióxido de carbono ou outras origens (XC, XS e XD), outras duas associadas à deterioração do betão pela ação de gelo/degelo ou por ataques químicos (XF e XA, respectivamente) e, por fim, uma classe quando não há risco de corrosão das armaduras ou de ataques químicos no betão (X0) [40], conforme indicado nas Tabelas 2.8 e 2.9. De forma semelhante ao especificado na Normas Técnicas (ABNT) [2], o Standardization [1] correlaciona os requisitos mínimos de resistência e composição do betão com a classe de agressividade do ambiente. Além disso, ele também relaciona esses parâmetros com a vida útil esperada para a estrutura, conforme indicado nas Tabelas 2.10, 2.11 e 2.12 [40].

Tabela 2.8: Classes de exposição devida à corrosão induzida por carbonatação, de acordo com o Standardization [1].

Classe	Descrição do ambiente	Exemplos informativos
XC1	Seco ou permanentemente húmido	Betão armado no interior de edifícios ou estruturas, com exceção das áreas com humidade elevada. Betão armado permanentemente submerso em água não agressiva.
XC2	Úmido, raramente seco	Betão armado enterrado em solo não agressivo. Betão armado sujeito a longos períodos de contato com água não agressiva. Superfícies exteriores de betão armado protegidas da chuva transportada pelo vento.
XC3	Moderadamente húmido	Betão armado no interior de estruturas com moderada ou elevada humidade do ar (v.g., cozinhas, casas de banho).
XC4	Ciclamente húmido e seco	Betão armado exposto a ciclos de molhagem/seca-gem. Superfícies exteriores de betão armado expostas à chuva ou fora do âmbito da XC2

Tabela 2.9: Classes de exposição devida à corrosão induzida por cloretos da água do mar, conforme Standardization [1]

Classe	Descrição do ambiente	Exemplos informativos
XS1	Ar transportando sais marinhos mas sem contato direto com água do mar	Betão armado em ambiente marítimo saturado de sais. Betão armado em áreas costeiras perto do mar, diretamente exposto e a menos de 200 m do mar; esta distância pode ser aumentada até 1 km nas costas planas e foz de rios.
XS2	Submersão permanente	Betão armado permanentemente submerso.
XS3	Zona de marés, de arrebentação e de salpicos	Betão armado sujeito às marés ou aos salpicos, desde 10 m acima do nível superior das marés (5 m na costa Sul de Portugal Continental) até 1 m abaixo do nível inferior das marés. Betão armado em que uma das superfícies está imersa em água do mar e a outra exposta ao ar (v.g., túneis submersos ou abertos em rocha ou solos permeáveis no mar ou em estuário de rios). Esta exposição exigirá muito provavelmente medidas de protecção suplementares.

Tabela 2.10: Composição e classe de resistência do betão sob ação de dióxido de carbono, segundo Standardization [1], para vida útil de 50 e 100 anos (os valores em negrito indicam os valores para vida útil de 100 anos, quando esses forem diferentes).

<i>Tipo de cimento</i>	CEM I (Referência); CEM II/A ⁽¹⁾				CEM II/B ⁽¹⁾ ; CEMIII/A ⁽²⁾ ; CEMIV ⁽²⁾ ; CEMV/A ⁽²⁾			
	XC1	XC2	XC3	XC4	XC1	XC2	XC3	XC4
<i>Classe de exposição</i>	XC1	XC2	XC3	XC4	XC1	XC2	XC3	XC4
<i>Recobrimento nominal mínimo (mm)</i>	25 (35)	35 (45)	35 (45)	40 (50)	25 (35)	35 (45)	35 (45)	40 (50)
<i>Relação água / cimento máxima</i>	0,65	0,65	0,60	0,60	0,65	0,65	0,55	0,55
<i>Consumo mínimo de cimento C (kg/m³)</i>	240	240	280	280	260	260	300	300
<i>Classe de resistência mínima</i>	C25/30	C25/30	C30/37	C30/37	C25/30	C25/30	C30/37	C30/37

⁽¹⁾ Não aplicável aos cimentos II/A e II/A-W e aos cimentos II/B-T e II/B-W, respectivamente.

⁽²⁾ Não aplicável aos cimentos com percentagem inferior a 50% a clínquer portland, em massa.

Tabela 2.11: Composição e classe de resistência do betão sob ação de cloretos, segundo Standardization [1], para vida útil de 50 e 100 anos (os valores em negrito indicam os valores para vida útil de 100 anos, quando esses forem diferentes).

<i>Tipo de cimento</i>	CEMIV/A (Referência); CEMIV/B; CEMIII/A; CEMIII/B; CEM V; CEMII/B ⁽¹⁾ ; CEMII/A-D			CEMI; CEMII/A ⁽¹⁾		
	XS1/XD1	XS1/XD1	XS1/XD1	XC1	XC2	XC3
<i>Classe de exposição</i>	XS1/XD1	XS1/XD1	XS1/XD1	XC1	XC2	XC3
<i>Recobrimento nominal mínimo (mm)</i>	45 (55)	50 (65)	55 (65)	45 (55)	50 (60)	55 (65)
<i>Relação água / cimento máxima</i>	0,55	0,55	0,45	0,45	0,45	0,40
<i>Consumo mínimo de cimento C (kg/m³)</i>	320	320	340	360	360	380
<i>Classe de resistência mínima</i>	C30/37	C30/37	C35/45	C40/50	C40/50	C50/60

⁽¹⁾ Não aplicável aos cimentos II-T, II-W, II/B-L e II/B-LL.

Tabela 2.12: Composição e classe de resistência do betão sob ataques químicos, segundo Standardization [1], para vida útil de 50 e 100 anos (os valores em negrito indicam os valores para vida útil de 100 anos, quando esses forem diferentes).

<i>Tipo de cimento</i>	CEM IV/A (Referência); CEM IV/B; CEM III/A; CEM III/B; CEM V; CEM II/B ⁽¹⁾ ; CEM II/A-D			CEM I; CEM II/A ⁽¹⁾		
	XA1	XA2 ⁽²⁾	XA3 ⁽²⁾	XA1	XA2 ⁽²⁾	XA3 ⁽²⁾
<i>Classe de exposição</i>	XA1	XA2 ⁽²⁾	XA3 ⁽²⁾	XA1	XA2 ⁽²⁾	XA3 ⁽²⁾
<i>Relação água / cimento máxima</i>	0,55 (0,50)	0,50 (0,45)	0,45 (0,40)	0,50 (0,45)	0,45 (0,40)	0,45 (0,40)
<i>Consumo mínimo de cimento C (kg/m³)</i>	320 (340)	340 (360)	360 (380)	340 (360)	360 (380)	380 (400)

Tabela 2.12: Composição e classe de resistência do betão sob ataques químicos, segundo Standardization [1], para vida útil de 50 e 100 anos (continuação)

<i>Classe de resistência mínima</i>	C30/37 (C40/50)	C35/45 (C45/50)	C35/45 (C45/50)	C35/45 (C45/50)	C40/50 (C50/60)	C40/50 (C50/60)
-------------------------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------

⁽¹⁾ Não aplicável aos cimentos II-T, II-W, II/B-L e II/B-LL.

⁽²⁾ Quando a agressividade resultar da presença de sulfatos, os cimentos devem satisfazer os requisitos mencionados na secção 5.3, nomeadamente no Quadro 10, aplicando-se ao betão as exigências estabelecidas neste quadro para o CEM IV.

2.7.2 Comprimento de amarração

De acordo com Vaz Rodrigues e Gutstein [40], os comprimentos de amarração calculados pelo Standardization [1] e pela Normas Técnicas (ABNT) [2] possuem a mesma ordem de grandeza, sendo os valores da NBR ligeiramente inferiores aos do Eurocódigo. Os resultados desses cálculos são apresentados na Tabela 2.13 [40].

Tabela 2.13: Comprimentos de ancoragem de acordo com o Standardization [1] e a Normas Técnicas (ABNT) [2], para $\sigma_s = f_{yd} = 435$ MPa (Vaz Rodrigues e Gutstein [40]).

<i>Betão f_{ck} em (MPa)</i>	Norma	20	25	30	35	40	45	50
<i>Condições de boa aderência</i>	NBR	44 ϕ	38 ϕ	33 ϕ	30 ϕ	28 ϕ	25 ϕ	25 ϕ
	EC2	48 ϕ	40 ϕ	36 ϕ	33 ϕ	30 ϕ	27 ϕ	25 ϕ
<i>Outras condições de aderência</i>	NBR	62 ϕ	54 ϕ	48 ϕ	43 ϕ	39 ϕ	36 ϕ	34 ϕ
	EC2	69 ϕ	57 ϕ	52 ϕ	47 ϕ	43 ϕ	38 ϕ	36 ϕ

2.7.3 Diâmetro do pino de dobramento

De acordo com Standardization [1], as armaduras devem possuir um diâmetro de dobragem mínimo para evitar danos, conforme indicado na Tabela 2.14. De forma similar, a Normas Técnicas (ABNT) [2] especifica diâmetros mínimos de dobragem para armaduras de tração (ver Tabela 2.15) e para estribos (ver Tabela 2.16).

Tabela 2.14: Diâmetro do pino de dobramento, segundo o Standardization [1].

Diâmetro do varão	Diâmetro mínimo do pino de dobramento
$\phi \leq 16mm$	4 ϕ
$\phi > 16mm$	7 ϕ

Tabela 2.15: Diâmetro do pino de dobramento para armaduras de tração em aço CA50, segundo a Normas Técnicas (ABNT) [2].

Diâmetro do varão	Diâmetro mínimo do pino de dobramento
$\phi < 20mm$	5 ϕ
$\phi \geq 20mm$	8 ϕ

Tabela 2.16: Diâmetro do pino de dobramento para estribos em aço CA50, segundo a Normas Técnicas (ABNT) [2].

Diâmetro do varão	Diâmetro mínimo do pino de dobramento
$\phi \leq 10mm$	3 ϕ
$10 < \phi < 20mm$	5 ϕ
$\phi \geq 20mm$	8 ϕ

3 METODOLOGIA DA AUTOMAÇÃO DO DETALHAMENTO DE CONSOLAS CURTAS

Neste capítulo, serão apresentados os principais métodos e procedimentos utilizados no desenvolvimento de uma rotina de automatização para o detalhamento de consolas curtas em projetos de estruturas de betão armado. Inicialmente, serão descritos os fluxos de trabalho relevantes, seguidos por uma análise dos softwares empregados e das técnicas de parametrização aplicadas ao detalhamento.

3.1 Contextualização do projeto de estudo

O projeto utilizado como base para o desenvolvimento do script em Dynamo foi fornecido por uma empresa brasileira de projetos de engenharia, que está estudando a possibilidade de implementar a metodologia BIM em seus processos. Este projeto piloto busca explorar as potencialidades do BIM no projeto de estruturas pré-fabricadas e averiguar sua integração aos fluxos de trabalho da empresa, como parte de um esforço para aprimorar a parametrização e a automatização no detalhamento estrutural.

A partir deste estudo, todas as diferenças identificadas entre os resultados gerados pela abordagem metodológica adotada neste trabalho e as premissas e processos definidas pela empresa para esse estudo inicial da implementação BIM, serão transmitidas à empresa cedente. O objetivo disso é fornecer informações que permitam ajustes e melhorias necessárias nessa fase inicial de implementação da metodologia BIM, auxiliando assim na padronização de processos.

3.1.1 Fluxo de trabalho do setor de projetos

Com o intuito de compreender o fluxo de informações, dentro do setor de projetos, definidos para esse estudo de implementação BIM, e assim identificar a melhor abordagem para o desenvolvimento do script em Dynamo, foi desenvolvido o fluxograma apresentado na Figura 3.1. A análise do fluxograma revelou a necessidade de desenvolver uma rotina específica para o detalhamento dos elementos, pois foi observado que o detalhamento das armaduras não é necessário durante o desenvolvimento do pré-projeto, uma vez que a taxa de aço, para orçamentação do projeto, nessa primeira fase, é estimada de outra forma.

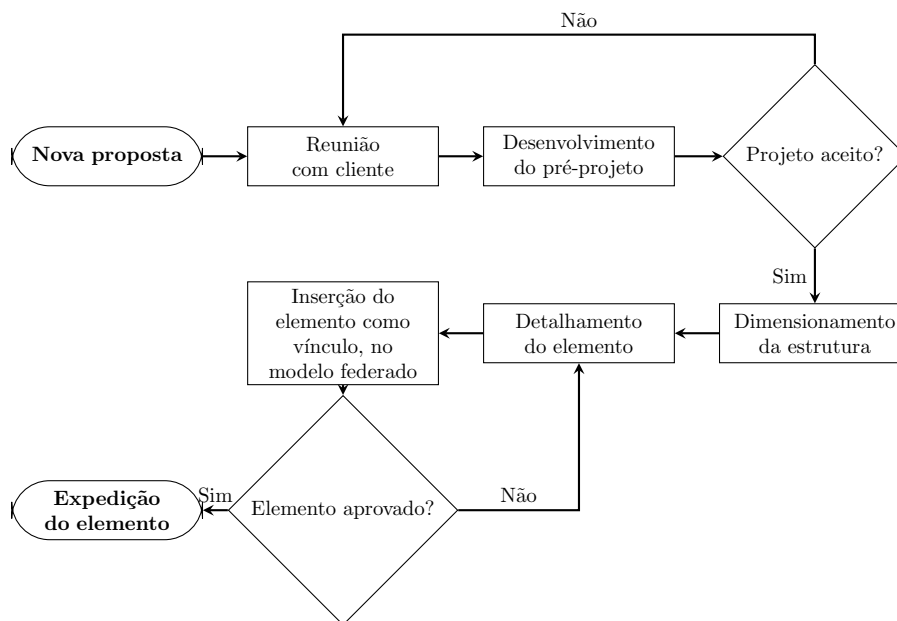


Figura 3.1: Fluxograma do setor de projetos definido pela empresa.

3.1.2 Fluxo de trabalho do detalhamento das peças

Assim como foi necessário compreender o fluxo de trabalho definido para o setor de projetos, também foi importante entender os procedimentos específicos de detalhamento dos elementos pré-fabricados contidos no projeto de estudo, uma vez que era necessário identificar a melhor fase para a execução do script, de forma a minimizar retrabalhos. Dessa forma, a Figura 3.2 apresenta a ordem dos processos de detalhamento de estruturas pré-fabricadas de betão, ilustrando a sequência de atividades a ser seguida.

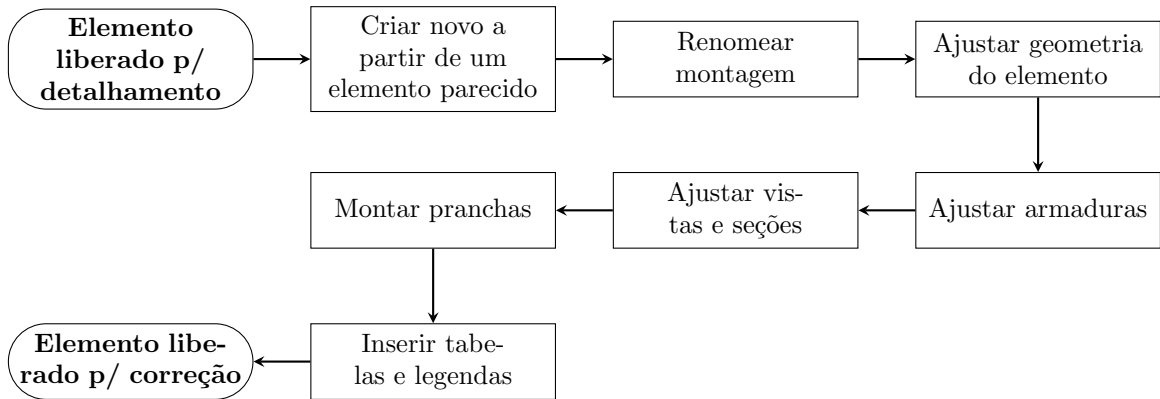


Figura 3.2: Fluxo de processos para detalhamento de elementos pré-fabricados definidos pela empresa.

A análise do fluxograma apresentado na Figura 3.2 revelou que a rotina Dynamo poderia ser executada em dois momentos distintos: após o ajuste da geometria do elemento ou juntamente com a inserção de tabelas e legendas, uma vez que, nesse primeiro estudo, a empresa optou por detalhar as consolas curtas manualmente em CAD e inseri-las no projeto como figuras vinculadas. No entanto, recomenda-se executar a rotina logo após os ajustes de geometria, pois isso permite ajustar as armaduras das consolas, se necessário, na mesma etapa. Sendo assim, a Figura 3.3 apresenta o fluxo recomendado após a implementação do script em Dynamo.

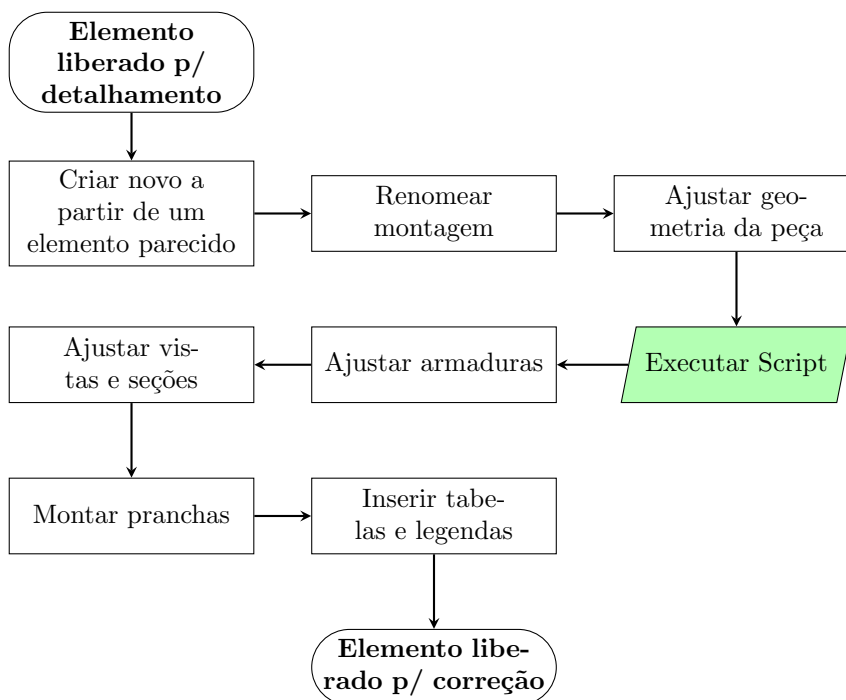


Figura 3.3: Fluxo de processos, recomendação após implantação do script em Dynamo, para detalhamento de elementos.

3.2 Projeto base

Como referido anteriormente, o projeto usado como base para a criação da rotina de automatização (script em Dynamo) foi cedido por uma empresa brasileira. Tratando-se de um artefato que segue as prescrições e normas brasileiras, a escrita do script segue essas mesmas normas, nomeadamente, no que diz respeito, e.g., às distâncias entre armaduras, taxas de armaduras máximas e mínimas, etc. O aço e o betão também seguem os requisitos preconizados nos seus documentos de homologação do Brasil. A decisão foi tomada pelo fato de se acreditar que a competência para aplicar a norma de Portugal para betão armado, particularmente o Standardization [1], ou a norma do Brasil para betão armado, designadamente, as normas Normas Técnicas (ABNT) [2] e Normas Técnicas (ABNT) [3], é a mesma.

O projeto consiste numa entrustrutura pré-fabricada de betão armado, conforme indicado na Figura 3.4. A estrutura possui 10 níveis (como representado na Figura 3.5), 92 pilares, 476 vigas e 1177 lajes.

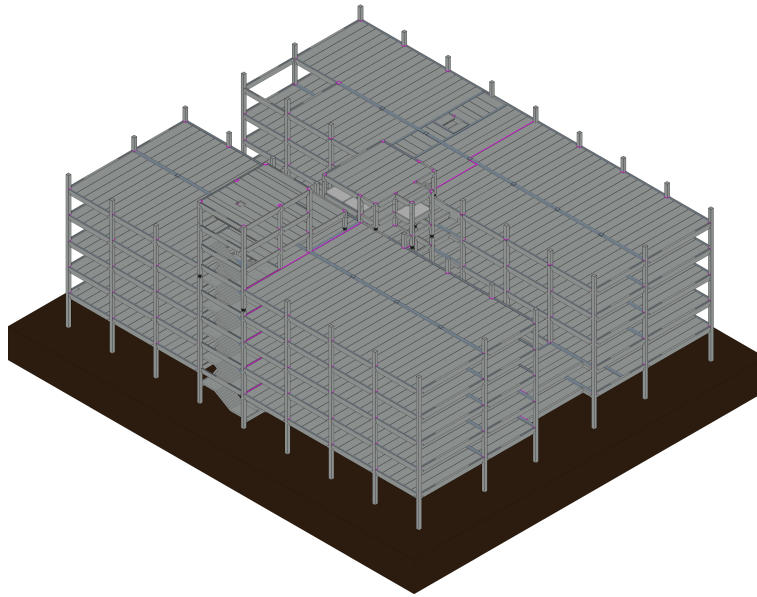


Figura 3.4: Perspetiva isométrica da estrutura do projeto.

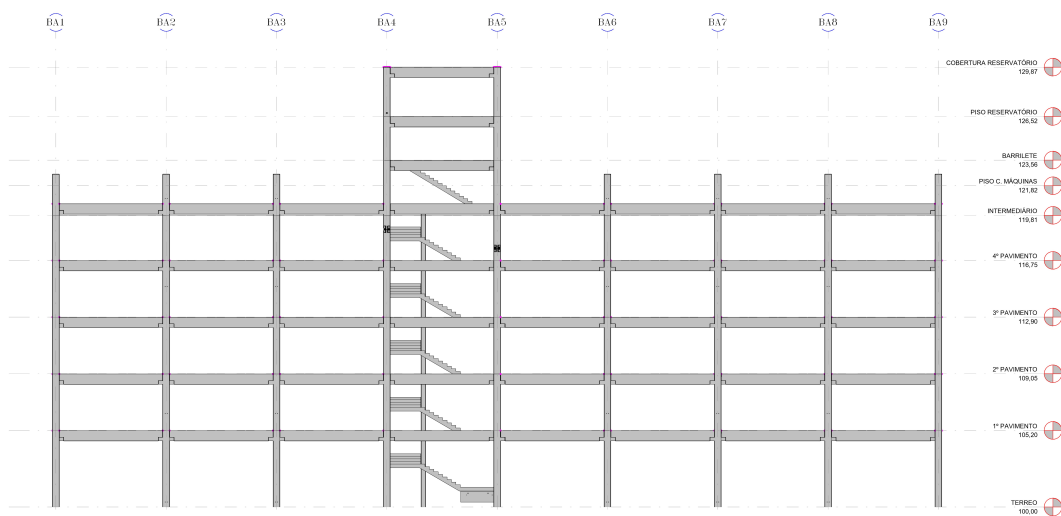


Figura 3.5: Corte longitudinal da estrutura do projeto.

3.2.1 Padrões de nomenclatura

A empresa responsável pelo projeto, desenvolveu o estudo a partir de um modelo próprio do Revit e definiu que qualquer elemento criado deve seguir uma nomenclatura pré-definida. Esses padrões são essenciais, pois facilitam a criação de filtros, a extração de informações, medições, além do controle e gerenciamento dos dados presentes no projeto. Os principais padrões de nomenclatura, necessários para o desenvolvimento do script, são apresentados na Tabela 3.1.

Tabela 3.1: Padrões de nomenclatura adotados no estudo.

Categoria do elemento	Nomenclatura	Explicação da nomenclatura
Armadura estrutural	EMPRESA-ARM- ϕ XXmm	EMPRESA: Família própria ARM: Família de armaduras ϕ XXmm: Indicação do diâmetro da armadura, deve ser sempre em milímetros
Consola	EMPRESA-PREO-Consolo-CP00-00x00cm.C00	PREO: Elemento pré-fabricado CPXX: Consola pré-fabricada, o 00 indica a altura da consola 00x00cm: Dimensões da consola, largura e profundidade C00: Tipo de consola
Tipo de consola	C00	C: Indica que é uma consola 00: Numeração da consola. Valores menores que 100 são destinados a consolas simples, e maiores ou iguais a 100, a consolas duplas
Montagem	PP0000	PP: Tipo de peça, PP indica um pilar pré-fabricado 0000: Numeração do elemento

3.2.2 Escolha dos casos

Devido à alta repetitividade das consolas, a empresa optou, nesse primeiro estudo, por detalhar as consolas curtas manualmente em CAD inserindo-as no projeto por meio de tabelas com figuras vinculadas. Essa escolha considerou que a modelagem manual de armaduras para elementos com alta repetitividade poderia dificultar a aceitação inicial da metodologia BIM, visto que não é prático modelar as armaduras manualmente, pois demanda muito tempo e pode resultar em problemas de inconsistência entre elementos idênticos. Por esse motivo, se pretende apresentar à empresa um script para detalhar as armaduras das consolas pelo método que será explicado posteriormente na Seção 3.5. Assim, foi identificada a necessidade de automatizar esse processo para eliminar o uso do CAD e prevenir inconsistências.

Ao todo, no projeto em estudo, foram utilizadas 735 consolas, sendo 459 destas consolas simples (ver Tabela 3.2) e as restantes (276) consolas duplas (ver Tabela 3.3). Como a quantidade de consolas simples na obra é significativamente maior do que a de consolas duplas, optou-se por fazer a automatização com base nos dois tipos mais frequentes de consolas simples: a consola C01, com 44 repetições, e a consola C02, com 70 repetições.

Tabela 3.2: Lista das consolas simples na obra base

Tipo de consola	Quantidade	Tipo de Consola	Quantidade
C01	44	C18	2
C02	70	C19	8
C03	31	C20	5
C04	1	C21	2
C05	3	C22	12
C06	32	C23	8

Tabela 3.2: Lista das consolas simples na obra base (continuação)

Tipo de consolo	Quantidade	Tipo de Consolo	Quantidade
C07	41	C24	1
C08	4	C25	4
C09	27	C26	34
C10	17	C27	1
C11	29	C28	1
C12	26	C29	1
C13	5	C30	1
C14	9	C31	1
C15	3	C32	-
C16	25	C33	2
C17	7	C34	24
TOTAL			459

Tabela 3.3: Lista das consolas duplas na obra base

Tipo de consola	Quantidade	Tipo de Consola	Quantidade
C101	2	C112	2
C102	77	C114	18
C103	17	C115	1
C105	6	C116	1
C106	67	C117	7
C107	6	C118	2
C108	2	C119	1
C109	6	C120	1
C110	55	C122	2
C111	1	C123	2
TOTAL			276

A partir da análise do detalhamento das consolas realizado em CAD pela empresa, especialmente das consolas C01 (Figura 3.6) e C02 (Figura 3.7), notou-se uma grande recorrência de tirantes detalhados como laços verticais ou como barras soldadas. Por conta disso, esse trabalho será destinado a automatização do detalhamento das armaduras de consolas retangulares simples, com armadura soldada na transversal ou em laço vertical.

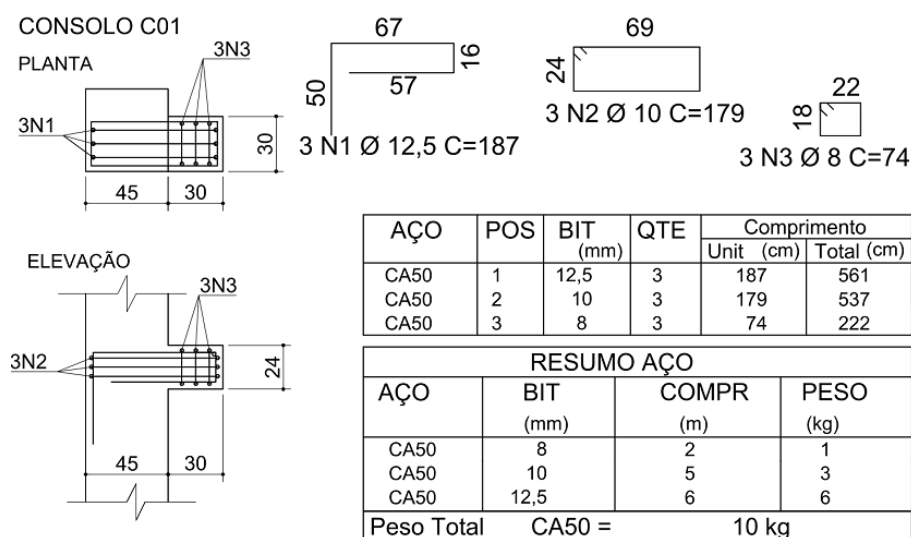


Figura 3.6: Consola C01; detalhe em CAD fornecido pela empresa.

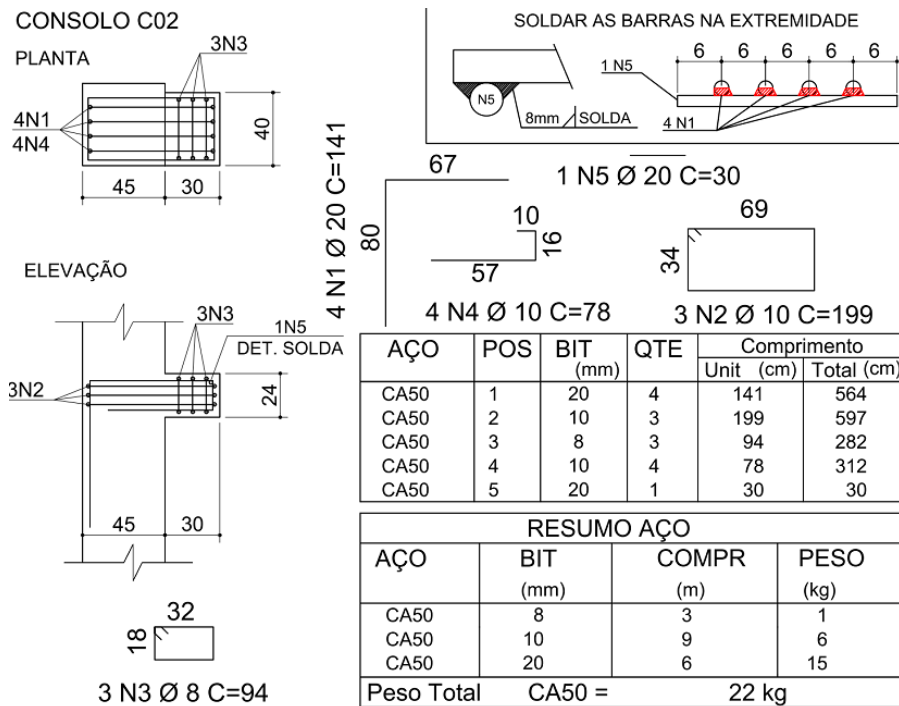


Figura 3.7: Consola C02; detalhe em CAD da fornecido pela empresa.

3.3 Configuração do Revit

A interface do Revit é organizada em abas e painéis, de maneira a promover o fácil acesso e navegação ao projeto. A área de interação pode ser dividida em quatro partes distintas, conforme indicado na Figura 3.8, sendo elas: Menus de abas (1), barra de propriedades (2), navegador de projetos (3) e a área de trabalho (4). Destaca-se que o script foi desenvolvido a partir da versão do Revit de 2023, e algumas funcionalidades e configurações podem não estar presentes em versões anteriores.

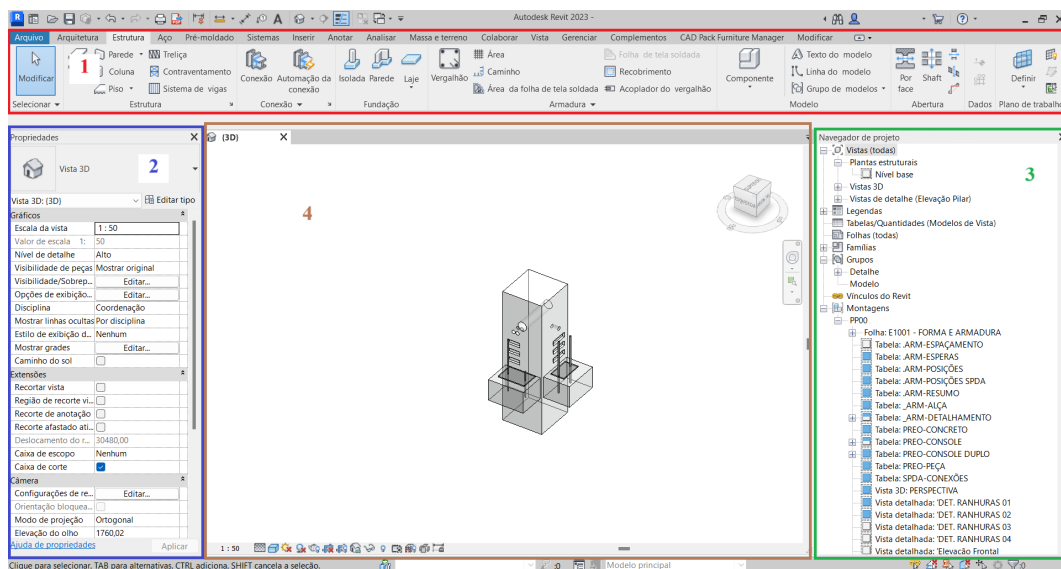


Figura 3.8: Interface do Revit, imagem proposadamente sem zoom.

O menu de abas agrupa as ferramentas por categorias, de modo a otimizar a utilização do software. O Revit possui, por padrão, 14 abas nesse menu, sendo as principais: Arquitetura, Estrutura, Vista e Gerenciar. A aba Arquitetura é destinada ao desenvolvimento de elementos arquitetônicos

do projeto, incluindo ferramentas para a criação de paredes, portas, janelas, pisos, tetos e telhados. Já a aba Estrutura é voltada para a modelagem e detalhamento de componentes estruturais, como vigas, pilares e lajes. Nessa aba também está localizada a ferramenta “Armadura”, que serve para a inserção de armaduras em elementos de betão armado.

Por sua vez, a aba Vista agrupa as ferramentas para a criação de plantas, de cortes, de elevações, de vistas 3D e de detalhes. Além disso, nela também estão localizadas ferramentas para a criação, a aplicação e o gerenciamento de modelos de vistas, que controlam os estilos visuais, as escalas e os níveis de detalhe das vistas do modelo. Por fim, a aba Gerenciar reúne ferramentas para a administração e a configuração do projeto, como ferramentas para a definição de fases, a importação e a exportação de dados, a gestão de materiais e os estilos de objeto. Além disso, o Dynamo é acessado através dessa aba.

A Barra de Propriedades, Figura 3.9, exibe as propriedades dos elementos selecionados no projeto, sendo que, quando não há elementos selecionados, a barra mostra as propriedades da vista ativa. A primeira linha desta barra mostra qual é a vista ou o componente selecionado. Logo abaixo, são exibidas as propriedades e os parâmetros dessa seleção, sendo possível editá-los diretamente. Através dessa barra também é possível editar e criar novos tipos, dentro das famílias, a partir do comando “Editar tipo”.

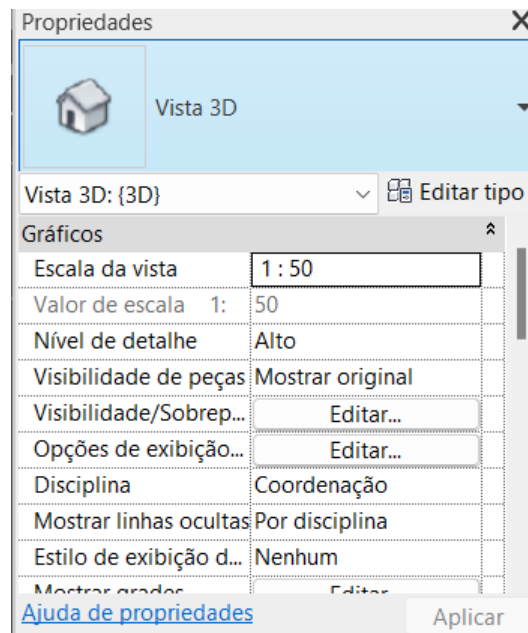


Figura 3.9: Barra de propriedades do Revit.

O Navegador de Projetos, Figura 3.10, organiza hierarquicamente todas as vistas, folhas, pranchas, tabelas e outros elementos do projeto. Os usuários podem expandir e colapsar categorias para acessar as diferentes partes do projeto. Além disso, o Navegador de Projetos oferece recursos para a criação, duplicação de vistas, folhas e pranchas.

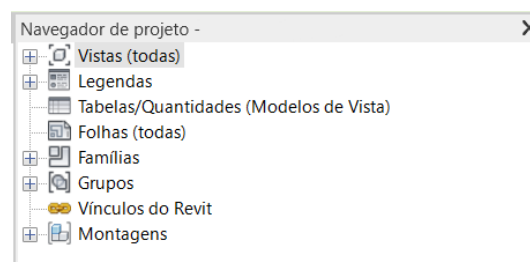


Figura 3.10: Barra de propriedades do Revit não expandido.

3.3.1 Configuração dos varões

Por padrão, o Revit já possui uma categoria de varões estruturais, com parâmetros próprios que permitem a personalização dessas barras. Entre esses parâmetros estão o diâmetro da barra e o diâmetro do mandril, também chamado pino de dobragem. A edição desses parâmetros é feita através da janela “Propriedades do Tipo”, conforme indicado na Figura 3.11. Vale destacar que o template utilizado já possuía os varões de $\phi 6, 3mm$, $\phi 8mm$, $\phi 10mm$, $\phi 12, 5mm$, $\phi 16mm$ e $\phi 20mm$ configurados, e que os diâmetros do pino de dobragem foram conferidos e estavam de acordo com o especificado na Normas Técnicas (ABNT) [2], conforme discutido anteriormente na Seção 2.7.3.

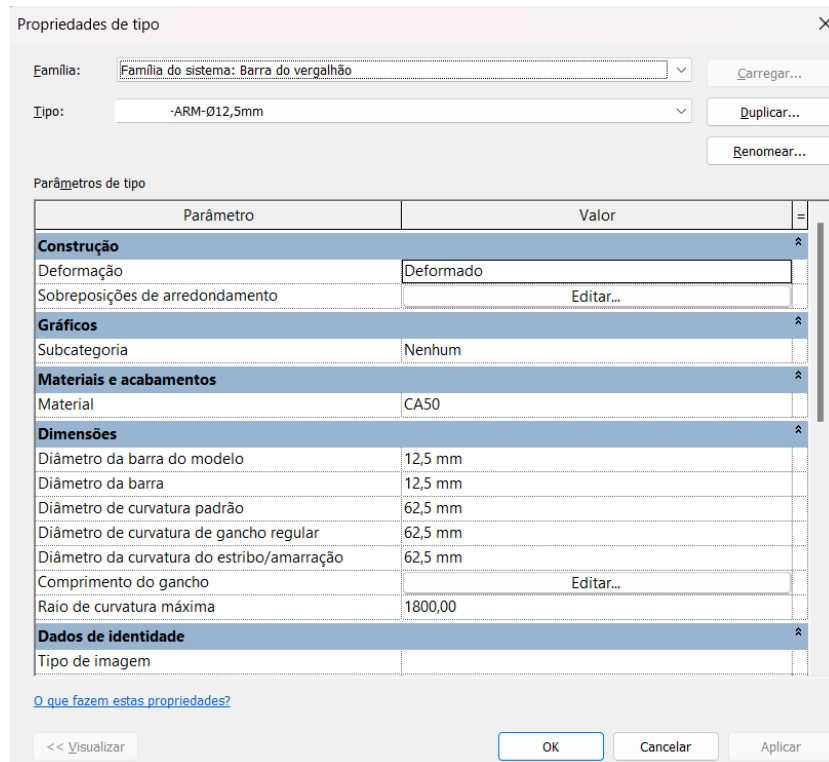


Figura 3.11: Janela de configuração dos varões estruturais

O Revit também permite criar conjuntos de armaduras, que podem ser distribuídas de cinco formas distintas: (1) único; (2) número fixo; (3) espaçamento máximo; (4) número com espaçamento; e (5) espaço mínimo livre. A distribuição único posiciona uma única barra num local específico. A distribuição por número fixo permite que um número determinado de barras seja distribuído uniformemente ao longo de uma linha ou caminho. A distribuição por espaçamento máximo ajusta as barras para garantir que o espaçamento entre elas não exceda um valor máximo definido. A distribuição por número com espaçamento combina os métodos anteriores, especificando tanto o número de barras quanto o espaçamento desejado. Por fim, a distribuição por espaço mínimo livre garante que um espaço mínimo entre barras seja mantido.

3.4 Interface do Dynamo

A interface do Dynamo, dentro do Revit, pode ser separada em três partes distintas, conforme indicado na Figura 3.12, sendo elas: Biblioteca (1), painel de visualização (2) e menu superior (3). Além dessas áreas padrão, é possível exibir uma caixa de diálogo (4), que ajuda a identificar e localizar os erros que aparecem durante o desenvolvimento de rotinas. Vale destacar que o script foi desenvolvido com a versão 2.10.3.1.3887 do Dynamo e, assim como no Revit, algumas funcionalidades e configurações podem não estar presentes em versões anteriores.

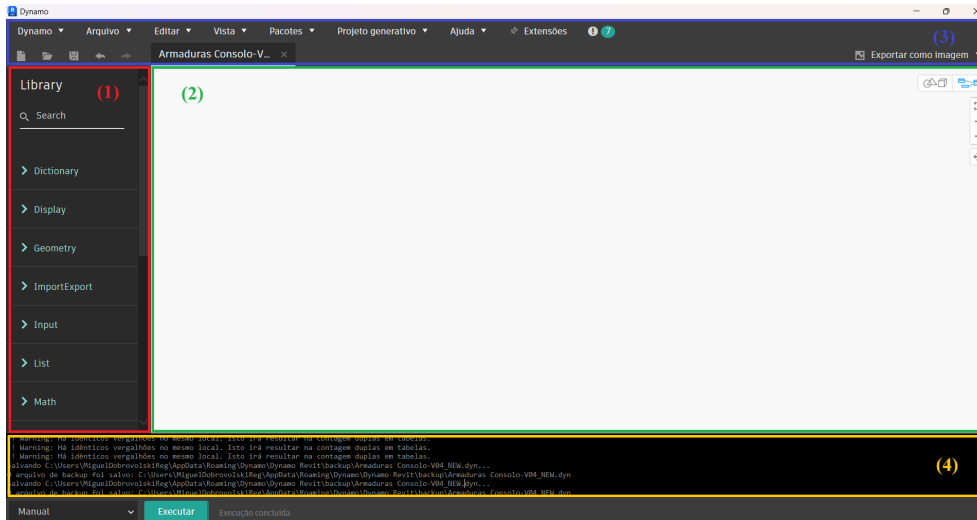


Figura 3.12: Interface do Dynamo, imagem propositalmente sem zoom.

O menu superior do Dynamo inclui diversas opções para a administração do ambiente de script, como abrir, salvar, importar e exportar scripts, além de configurações de preferências e acesso a pacotes adicionais. Ele também oferece ferramentas para depuração e execução de scripts, assim como opções para ajustar as configurações de visualização do espaço de trabalho. Uma dessas configurações de visualização que foi bastante útil é a opção “isolar geometria selecionada”, acessível a partir da aba de preferências. Com essa opção ativa, é possível visualizar com mais clareza o elemento selecionado pelo script, conforme demonstrado na Figura 3.13.

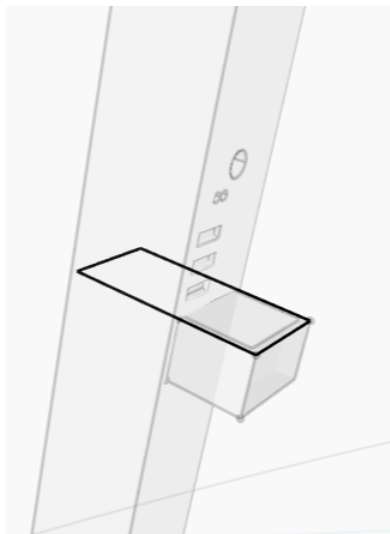


Figura 3.13: Geometria destacada pela preferência “Isolar Geometria Selecionada”.

A Biblioteca agrupa os nós por categorias, facilitando a busca e a inserção de elementos na área de trabalho. As principais categorias incluem Geometry e Revit, cada uma contendo nós específicos para diferentes tipos de operações e funções. A categoria Geometry agrupa os nós para criação e alteração de elementos geométricos, enquanto a categoria Revit reúne as ferramentas para a interação com os elementos do Revit, como criação, modificação e análise de componentes do modelo.

A área de trabalho é o espaço onde os nós são inseridos e conectados para criar os scripts. Nessa área, os nós da Biblioteca são arrastados e conectados para definir o fluxo de trabalho. Cada nó inserido pode ser configurado individualmente, permitindo a personalização dos parâmetros e das operações. Além disso, a área de trabalho exibe uma representação gráfica dos resultados dos

scripts, permitindo a visualização de geometrias e outros elementos criados ou modificados pelo script em tempo real, sem a necessidade de retornar à área de trabalho do Revit.

3.4.1 Pacote Structural Design

Os pacotes do Dynamo são coleções de nós adicionais, criados por usuários ou desenvolvedores, que estendem as funcionalidades do Dynamo. Eles cobrem uma ampla gama de aplicações, desde modelagem avançada e análise estrutural até automação de tarefas repetitivas e integração com outras ferramentas e plataformas. Esses pacotes são disponibilizados no Dynamo Package Manager, onde podem ser facilmente baixados e instalados, o que permite a personalização do ambiente de trabalho do Dynamo.

Para o desenvolvimento do script, foi utilizado apenas um pacote adicional: o *Structural Design* versão 2024.1.18. Esse pacote oferece nós para a criação e manipulação de elementos estruturais, como vigas, colunas e fundações, permitindo que os usuários definam parâmetros detalhados e ajustem as propriedades desses elementos. Além disso, ele possui ferramentas específicas para a criação de armaduras, facilitando a geração de varões e conjuntos de armaduras. Entre os nós adicionados por esse pacote, destacam-se dois principais: *Create.FromCurves* e *Layout.SetLayoutAsFixedNumber*, que permitem, respectivamente, a criação de varões a partir de curvas e a formação de conjuntos de armaduras com um número fixo de barras.

Para criar corretamente o varão com o nó *Create.FromCurves* (ver Figura 3.14), é necessário preencher obrigatoriamente nove parâmetros: *curve*, *rebarStyle*, *rebarBarType*, *startHookType*, *endHookType*, *startHookOrientation*, *endHookOrientation*, *hostElement* e *vector*. Cada um desses parâmetros deve receber variáveis de tipos específicos, caso contrário, o nó não será executado corretamente. A Tabela 3.4 correlaciona o nome do parâmetro com o tipo de variável esperado, juntamente com uma descrição do parâmetro.

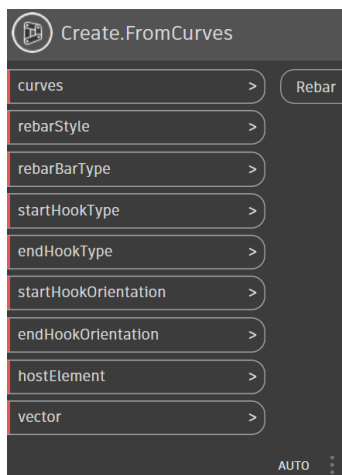


Figura 3.14: Nó *Create.FromCurves*, do pacote *Structural Design*.

Tabela 3.4: Variáveis do nó *Create.FromCurves*, do pacote *Structural Design*

Nome da variável	Tipo da variável	Descrição
curves	Curve[]	Curva que define o contorno do varão estrutural.
rebarStyle	String	Define-se é uma barra de forma livre, estribo ou gancho. Pode ser preenchido com “Standard”, para barras de forma livre ou com “StirrupTie”, para estribos e ganchos
rebarBarType	Element	Tipo de varão. Precisa ser uma categoria existente no modelo.

Tabela 3.4: Variáveis do nó *Create.FromCurves*, do pacote *Structural Design* (continuação)

Nome da variável	Tipo da variável	Descrição
starHookType	Element	Define a orientação do gancho na extremidade inicial da curva. Pode ser preenchido com <i>none</i> , quando não existem ganchos.
endHookType	Element	Define a orientação do gancho na extremidade final da curva. Pode ser preenchido com <i>none</i> , quando não existem ganchos.
startHookOrientation	String	Define se o gancho na extremidade é desenhado a direita ou esquerda da extremidade inicial da curva. Pode ser preenchido com as opções <i>Right</i> ou <i>Left</i> .
endHookOrientation	String	Define se o gancho na extremidade é desenhado a direita ou esquerda da extremidade final da curva. Pode ser preenchido com as opções <i>Right</i> ou <i>Left</i> .
hostElement	Element	Define qual o elemento hospedeiro do varão.
vector	Vector[]	Vetor normal ao plano em que a curva foi desenhada.

De modo semelhante, o nó *Layout.SetLayoutAsFixedNumber* (ver Figura 3.15) também precisa ser preenchido com alguns valores para ser executado corretamente. Ao todo, ele pode receber seis valores, sendo dois obrigatórios (*rebar* e *arrayLength*) e quatro opcionais (*numberOfBarPositions*, *barsOnNormalSide*, *includeFirstBar* e *includeLastBar*). A descrição de cada um dos parâmetros é apresentada na Tabela 3.5. Vale destacar que, no desenvolvimento do script, apenas os três primeiros parâmetros foram preenchidos, pois os valores padrão dos demais estavam de acordo com o necessário.

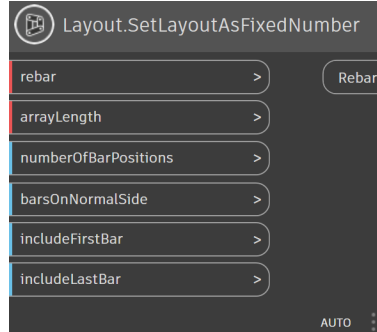


Figura 3.15: Nó *Layout.SetLayoutAsFixedNumber*, do pacote *Structural Design*.

Tabela 3.5: Variáveis do nó *Create.FromCurves*, do pacote *Structural Design*

Nome da variável	Tipo da variável	Descrição
rebar	Element	Varão estrutural
arrayLength	Double	Comprimento para distribuir os varões.
numberOfBarPositions	Integer	Quantidade de varões a serem posicionados do conjunto. Valor padrão: 2.
BarsOnNormalSide	Boleana	Verifica se o conjunto de varões está distribuído no plano normal ao traçado desse. Valor padrão: Verdadeiro.
IncludeFirstBar	Boleana	Defini se a primeira barra do conjunto é exibida. Valor padrão: Verdadeiro.
IncludeLastBar	Boleana	Defini se a última barra do conjunto é exibida. Valor padrão: Verdadeiro.

3.5 Script desenvolvido

A automatização do processo de detalhamento de consolas, foi feita pelo criação de um script em Dynamo, como mencionado anteriormente. A rotina desenvolvida pode ser dividida em três partes distintas, conforme mostrado na Figura 3.16, sendo elas: importação dos dados de armaduras (1), seleção de elementos (2) e criação da armadura (3). Além disso, para manter o código organizado, foram criados grupos de diferentes cores para identificar facilmente a função deles, como detalhado na Tabela 3.6.

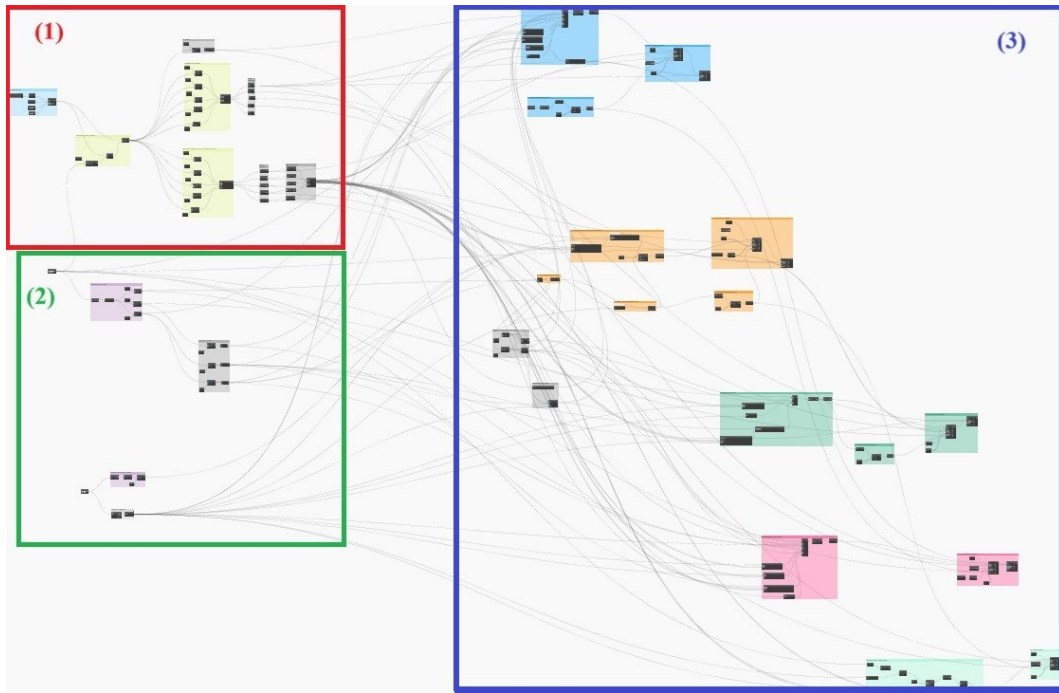


Figura 3.16: Esboço da rotina de automatização para detalhamento de armaduras, desenvolvida em Dynamo.¹

Tabela 3.6: Atribuição de cores para organização do script

Cor	Descrição
Amarelo	Manipulação de dados
Ciano	Importação de dados
Roxo	Seleção de geometria
Cinza	Dados prontos para uso
Azul	Criação do Tirante
Laranja	Criação da armadura de costura
Verde	Criação da armadura transversal
Rosa	Criação da armadura complementar
Verde claro	Criação da barra soldada

Para facilitar a manutenção e a atualização do script, foi implementada uma estrutura modular, onde cada grupo corresponde a uma função específica dentro do código, o que simplifica a identificação e a modificação de partes individuais do código. Além disso, todos os grupos foram nomeados, e suas funcionalidades e observações importantes foram explicadas na descrição de cada grupo. Essa documentação adicional garante que qualquer pessoa que utilize ou altere o script tenha uma compreensão propósito e funcionamento de cada seção, minimizando o risco de erros.

¹A imagem tem como objetivo mostrar a globalidade da etapa, de forma que o conteúdo escrito não é relevante.

3.5.1 Importação dos dados de armaduras

No projeto piloto da empresa, o dimensionamento das consolas curtas de pilares se realiza por meio de folhas do Excel. Por esse motivo, foi decidido agrupar todos os dados de armaduras de consolas em um único local. A partir desse resumo, a rotina importa e manipula os dados para que as consolas curtas sejam detalhadas automaticamente. É importante salientar que, para o correto funcionamento da rotina, a folha de resumo deve estar estruturada conforme apresentado na Tabela 3.7, com a primeira célula posicionada na célula A1. Além disso, a célula com a informação sobre o detalhamento do tirante, como o laço vertical ou com a barra soldada, pode ser automatizada conforme o diâmetro do varão, conforme discutido anteriormente na Seção 2.7.

Tabela 3.7: Resumo de armaduras das consolas

Consola	N1			N2		N3		N4		N5	
	Quant.	Ø	L.V?	Quant.	Ø	Quant.	Ø	Quant.	Ø	Quant.	Ø
C01	3	12,5	Sim	3	10	3	8				
C02	4	20	Não	3	10	3	8	4	10	1	20
C03	4	20	Não	3	10	3	8	4	10	1	20

L.V = Laço Vertical

A importação da folha de resumo para o ambiente Dynamo foi realizada através do nó *Data.ImportExcel*. Esse nó requer o preenchimento obrigatório de dois campos: *file* e *sheetName*, que indicam, respectivamente, o caminho do arquivo e o nome da planilha a ser importada. Além disso, foi necessário definir o parâmetro *readAsString* como verdadeiro, pois, para as manipulações subsequentes, era essencial que as células fossem lidas como variáveis do tipo texto, e não como variáveis numéricas. Esta primeira etapa da importação dos dados da planilha é apresentada na Figura 3.17.

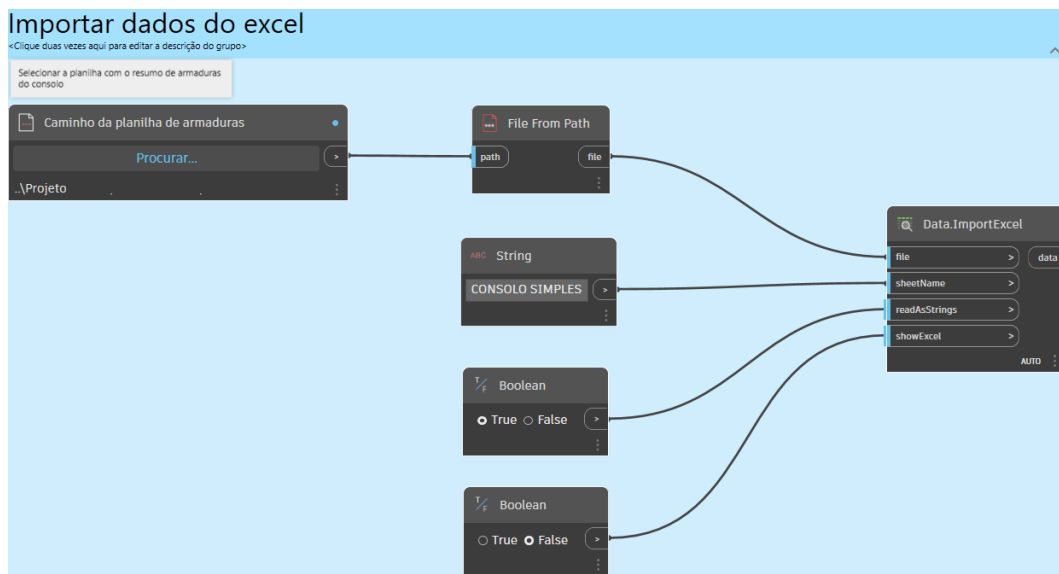


Figura 3.17: Importação da planilha de resumo de armaduras no Dynamo.¹

A próxima etapa foi selecionar a linha da tabela de resumo que corresponde à consola a ser detalhada, conforme indicado na Figura 3.18. Para fazer isso de forma automatizada, foi utilizado o nó *Element.GetParameterValueByName*, que retorna o valor de um parâmetro com base no seu nome. Nesse caso, o parâmetro pedido foi “CONSOLO-Tipo”, pois é ele que define o tipo de consola. Além disso, foi criado um código em Python para retornar o índice da linha correspondente à consola selecionada, como mostrado no Script 3.1. É importante salientar que o parâmetro

¹A imagem tem como objetivo mostrar a globalidade da etapa, de forma que o conteúdo escrito não é relevante.

“CONSOLE-Tipo” foi acrescentado à família de consoles e definido como igual ao parâmetro já existente “CONSOLE-Tipo”. Isso foi necessário porque o nó *Element.GetParameterValueByName* lê apenas parâmetros de instância, enquanto o parâmetro original precisa ser um parâmetro de tipo.

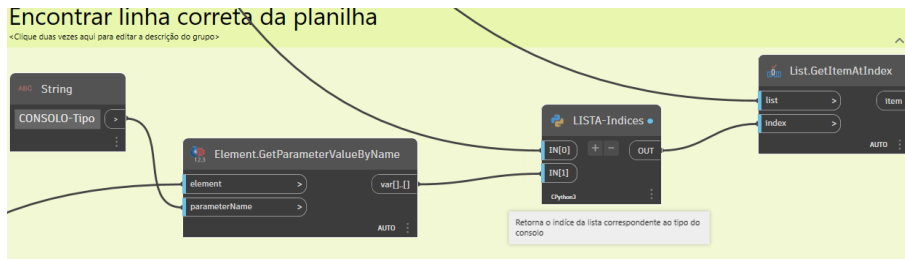


Figura 3.18: Seleção da linha da planilha de resumo de armaduras correspondente ao tipo de console a ser detalhado.¹

```

1 # Carregar as bibliotecas DesignScript e padrao do Python
2 import sys
3 import clr
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6
7 # Funcao para encontrar o indice da sublista onde o primeiro item e igual ao valor
  do usuario
8 def encontrar_indice(data, target):
9     for i, sublist in enumerate(data):
10        if sublist and sublist[0] == target:
11            return i
12    return -1
13
14 # Lista fornecida (aninhada) e valor alvo
15 lista_entrada = IN[0]
16 valor_procurado = IN[1]
17
18 # Encontrar o indice
19 index = encontrar_indice(lista_entrada, valor_procurado)

```

Script 3.1: Seleção do índice da linha, na folha de resumo de armaduras, correspondente ao tipo de console a ser detalhada.

A próxima etapa foi a extração das quantidades de cada tipo de armadura, conforme indicado na Figura 3.19. Como essas quantidades são numéricas e não textuais, foi necessário converter o tipo de variável de *string* para *integer*. Essa conversão foi realizada através de um nó personalizado em Python, conforme mostrado no Script 3.2.

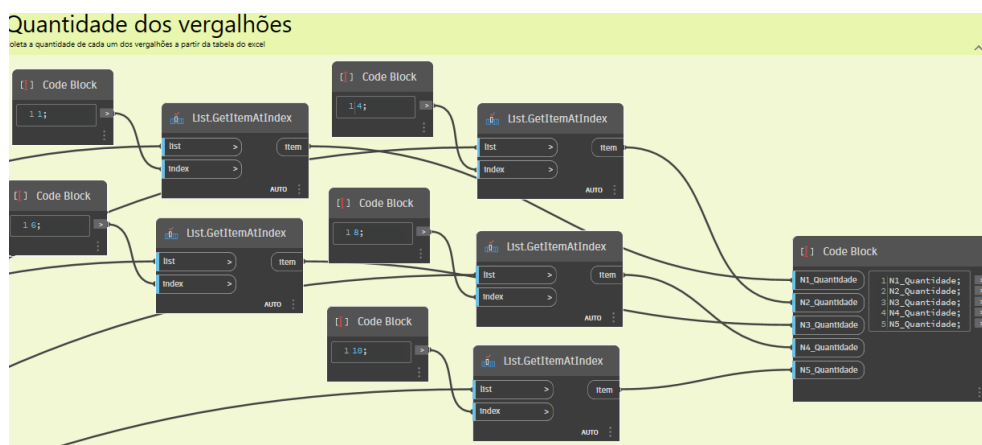


Figura 3.19: Seleção da quantidade de cada tipo varão.¹

¹A imagem tem como objetivo mostrar a globalidade da etapa, de forma que o conteúdo escrito não é relevante.

```

1 import sys
2 import clr
3 clr.AddReference('ProtoGeometry')
4 from Autodesk.DesignScript.Geometry import *
5
6 # Funcao para converter uma string em inteiro
7 def convert_to_integer(input_string):
8     try:
9         int_value = int(input_string)
10    except ValueError:
11        int_value = None # Ou algum valor padrao ou logica de erro
12    return int_value
13
14 # Inputs do Dynamo
15 input_string = IN[0] # Entrada da string que representa um numero
16
17 # Chamar a funcao e obter o valor convertido
18 result = convert_to_integer(input_string)
19
20 # Saida para o Dynamo
21 OUT = result

```

Script 3.2: Conversão de variáveis do tipo *String* para variáveis do tipo *Integer*.

Por fim, a última etapa da importação de dados foi a extração do diâmetro de cada conjunto de armaduras. Para isso, foi criado um nó que gera a nomenclatura dos varões conforme explicado anteriormente na Seção 3.2.1. No entanto, também é necessário associar um elemento a esse nome, para isso, foi utilizado o nó “*ElementType.ByName*”, que consulta um elemento com o nome especificado. Essa última etapa é ilustrada na Figura 3.20.

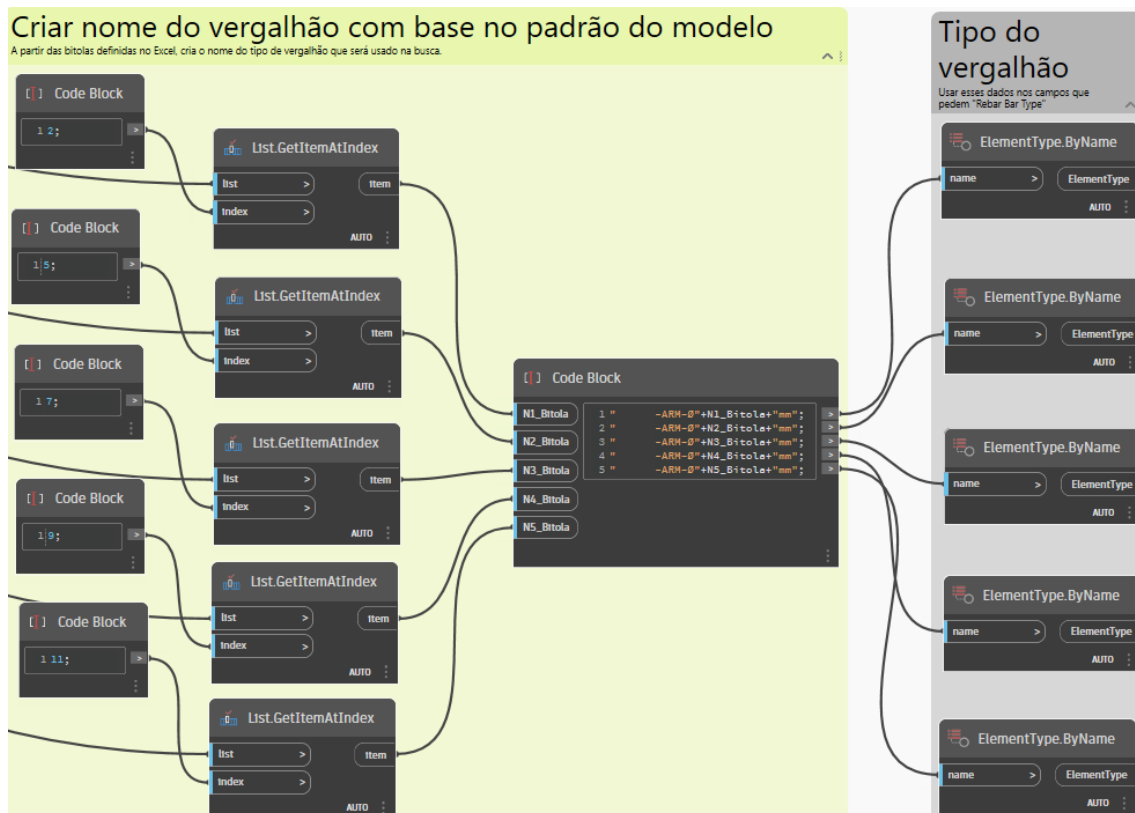


Figura 3.20: Extração dos diâmetros de cada conjunto de armadura.¹

¹A imagem tem como objetivo mostrar a globalidade da etapa, de forma que o conteúdo escrito não é relevante.

3.5.2 Seleção de elementos

A etapa subsequente à importação dos dados de armadura, foi a seleção dos elementos. Para traçar o formato das armaduras, foi necessário selecionar planos a partir dos quais as geometrias são criadas. No caso, foi necessário selecionar três faces das consolas: superior, frontal e lateral, conforme indicado na Figura 3.21. É importante salientar que os índices das faces são definidos de acordo com a forma de criação da família, de forma que esses índices sejam fixos e independentes de fatores, como o posicionamento no projeto, o que tornou viável selecionar as faces a partir deles.

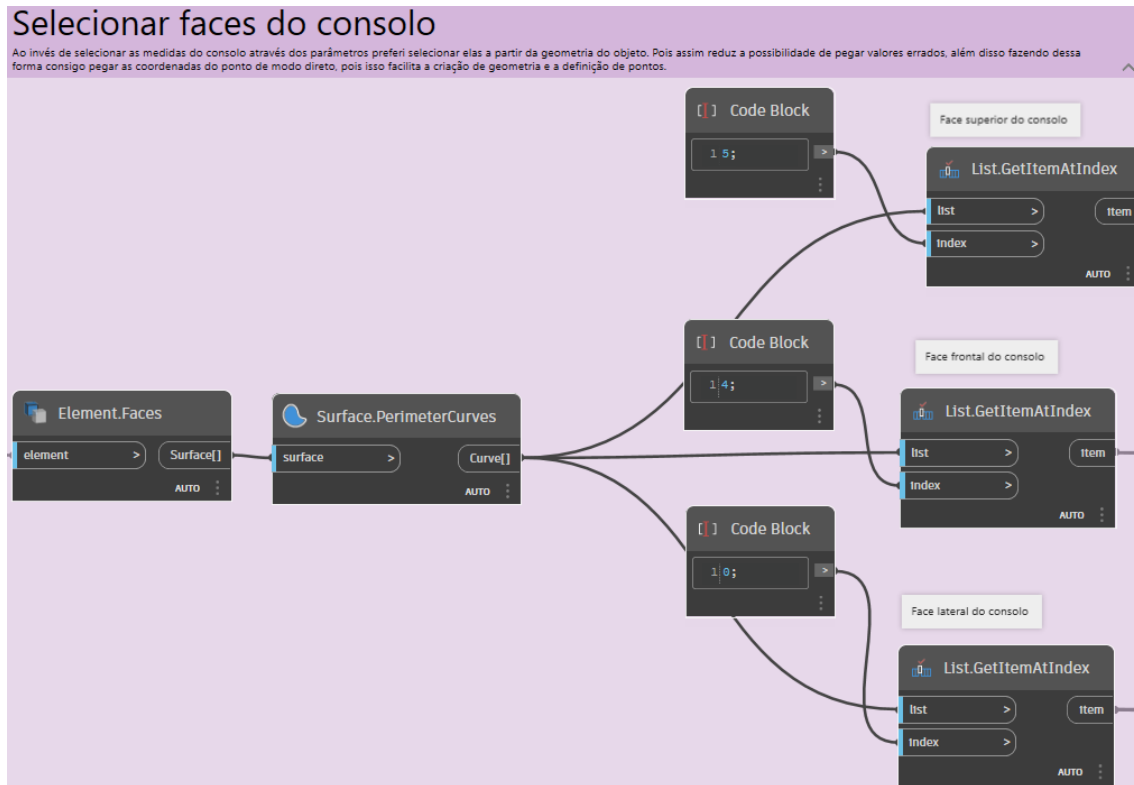


Figura 3.21: Seleção das faces das consolas.¹

Além disso, foi necessário selecionar os parâmetros de altura, largura e profundidade da consola. Essa tarefa poderia ser executada de duas formas: a partir dos valores preenchidos pelo usuário ou a partir da geometria selecionada. Para reduzir a possibilidade de erros, foi decidido seguir pelo segundo caminho. De modo que foram selecionadas as arestas que definem essas três dimensões para então obter o atributo *length*, presente em todos os objetos da classe *Line*, através de um nó personalizado em Python (Script 3.3), conforme indicado na Figura 3.22.

¹A imagem tem como objetivo mostrar a globalidade da etapa, de forma que o conteúdo escrito não é relevante.

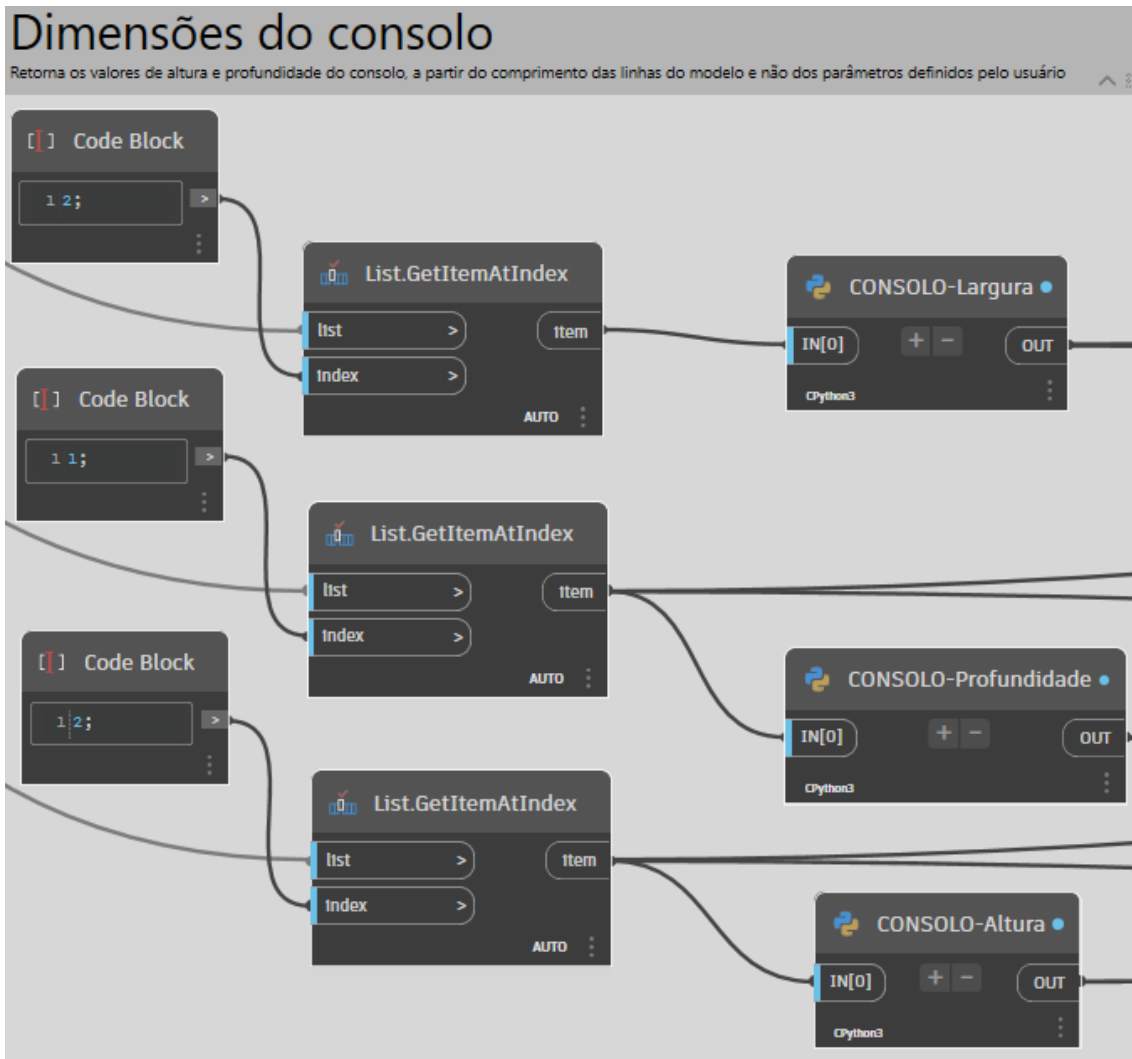


Figura 3.22: Seleção das dimensões das consolas.¹

```

1 # Carregar as bibliotecas DesignScript e padrao do Python
2 import sys
3 import clr
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6
7 # As entradas para este no serao armazenadas como uma lista nas variaveis IN.
8 entradaDados = IN[0]
9
10 # Atribua a sua saida para a variavel OUT.
11 OUT = entradaDados.Direction.Length

```

Script 3.3: Obter atributo *length* em objetos de classe *line*.

A seleção do pilar, elemento hospedeiro das consolas, foi feita de modo semelhante, diferenciando-se apenas nos elementos que precisavam ser escolhidos. Para o detalhamento das armaduras das consolas, é importante conhecer a seção do pilar, pois o tirante e a armadura de costura devem se estender até a face mais distante do pilar. Assim como nas consolas, os índices das faces dos pilares são definidos de acordo com a forma de criação da família, sendo que a face de índice 0 é a face acima do encastramento, que, por convenção, é a seção do pilar. Esse processo é ilustrado na Figura 3.23.

¹A imagem tem como objetivo mostrar a globalidade da etapa, de forma que o conteúdo escrito não é relevante.

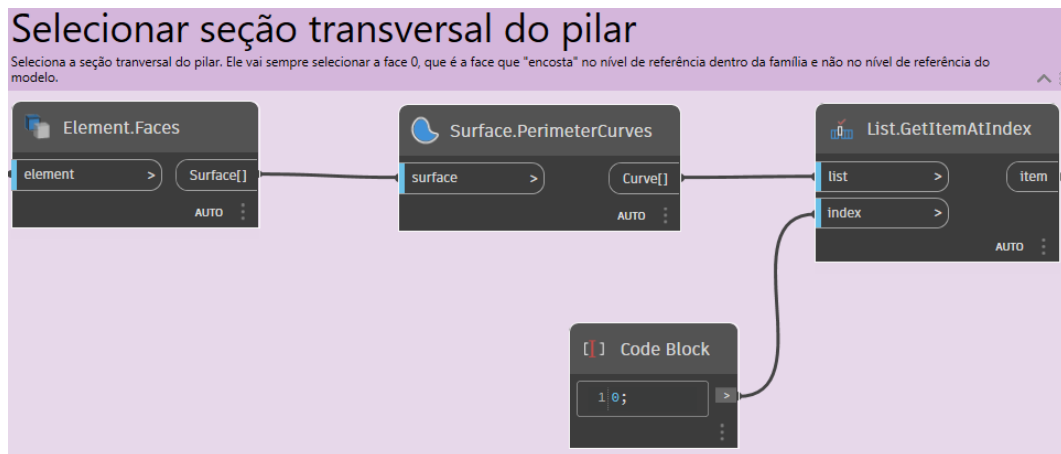


Figura 3.23: Seleção da seção transversal do pilar.¹

Além disso, a partir da seleção do pilar, foi coletado o parâmetro de cobrimento usando o nó *Properties.RebarHostCover*, ver Figura 3.24. Vale destacar que esse nó não pode ser aplicado à consola, pois a categoria em que a família de consolas foi modelada (Conexões estruturais) não possui o parâmetro de cobrimento de forma nativa. Como o cobrimento da consola deve ser o mesmo que o do pilar, optou-se por obter essa informação diretamente do pilar.

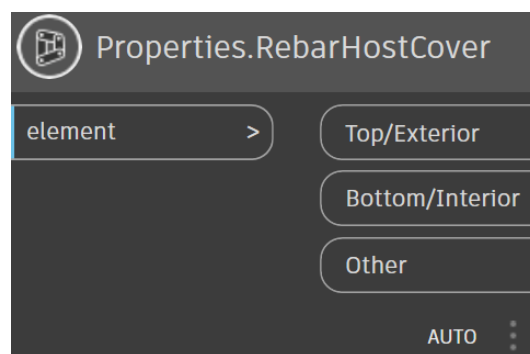


Figura 3.24: Nó *Properties.RebarHostCover*.

3.5.3 Criação da armadura

Ao analisar o detalhamento das consolas feitas em CAD pela equipa da empresa, observou-se que as armaduras seguem um padrão de nomenclatura, conforme indicado na Tabela 3.8. Para facilitar a compreensão e manter a consistência, o script foi desenvolvido com base nesse padrão de nomenclatura.

Tabela 3.8: Padrão de nomenclatura da empresa para as armaduras das consolas

Armadura	Designação da empresa
Tirante	N1
Costura	N2
Armadura Transversal	N3
Armadura complementar	N4
Barra soldada	N5

¹A imagem tem como objetivo mostrar a globalidade da etapa, de forma que o conteúdo escrito não é relevante.

De modo geral, a criação das armaduras envolve três etapas: traçar o varão, criar um vetor com o sentido de repetição dos varões e, finalmente, formar o grupo de varões. Nos tópicos subsequentes, esses procedimentos serão explicados de forma mais detalhada, ressaltando as peculiaridades na criação de cada um dos cinco grupos de armaduras.

3.5.3.1 Armadura de costura

A armadura de costura (armadura N2) foi a primeira a ser criada e é modelada como um estribo fechado que conecta a consola ao pilar. Para criá-la, foi necessário identificar as faces paralelas do pilar e da consola que estão mais distantes entre si. Para isso, foram desenvolvidas duas funções dentro de um nó Python personalizado.

A primeira função (Script 3.4), verifica o paralelismo entre os lados do pilar e a aresta superior da face frontal da consola, usando, para isso, as coordenadas dos vértices dessas seções. Vale destacar que foi necessário usar a função *round* com três casas decimais para arredondar os valores nas verificações. Isso é importante porque, independentemente da unidade utilizada no projeto, o Revit armazena os valores de comprimento utilizando o sistema imperial. Como o projeto foi desenvolvido com base no sistema métrico, podem ocorrer problemas de arredondamento, resultando em pequenas diferenças nas casas decimais. Embora essas diferenças sejam insignificantes para o projeto em si, elas afetam algumas comparações realizadas pela rotina de automação.

```

1 def ladosParalelos(vetor, secao_consolo):
2     # Lista para armazenar os indices dos vetores paralelos entre si
3     indices_vetores_paralelos = []
4
5     # Verifica se o vetor e paralelo a cada vetor na lista
6     for i, v in enumerate(secao_consolo):
7         if (round(vetor.X,3) != 0 and round(v.Direction.X,3) != 0 and round(vetor.Y
8             ,3) == 0 and round(v.Direction.Y,3) == 0) or \
9             (round(vetor.X,3) == 0 and round(v.Direction.X,3) == 0 and round(vetor.Y
10                ,3) != 0 and round(v.Direction.Y,3) != 0):
11             indices_vetores_paralelos.append(v)
12         else:
13             proporcao_x = round(vetor.X, 6) / round(v.Direction.X, 6)
14             proporcao_y = round(vetor.Y, 6) / round(v.Direction.Y, 6)
15             if proporcao_x == proporcao_y:
16                 indices_vetores_paralelos.append(v)
17
18     return indices_vetores_paralelos

```

Script 3.4: Verificar o paralelismo ente os lados do pilar e da consola.

A segunda função (Script 3.5) é executada após a função que encontra os lados paralelos e calcula a distância, usando o método euclidiano, entre o vértice superior direito da consola e as extremidades dos lados paralelos do pilar identificados anteriormente. Para garantir que a face mais distante do pilar foi corretamente identificada, essa função compara a soma das distâncias, entre o vértice da consola e as extremidades inicial e final da face do pilar.

```

1 def distancia(consolo, pilar):
2     # Calcula a distancia entre dois pontos usando a formula de distancia
3     euclidiana
4     return (((consolo.StartPoint.X - pilar.StartPoint.X)**2 + (consolo.StartPoint.Y
5         - pilar.StartPoint.Y)**2)**0.5) + \
6         (((consolo.StartPoint.X - pilar.EndPoint.X)**2 + (consolo.StartPoint.Y -
7         pilar.EndPoint.Y)**2)**0.5)

```

Script 3.5: Calcular a distância entre um vértice da consola e os lados do pilar

Essas duas funções são acionadas dentro do script que cria os vértices que delimitam o perímetro máximo da armadura de costura. Esses pontos são gerados a partir das coordenadas da consola e da face mais distante do pilar, conforme mostrado no Script 3.6. Com esses pontos, utiliza-se o nó *Rectangle.ByCornerPoints* para atribuir uma forma retangular ao limite. É importante salientar que os pontos precisam estar ordenados para que o nó possa criar a forma corretamente. Para facilitar o entendimento, a Figura 3.25 apresenta o perímetro da forma criada por esse processo.

```

1 secao_pilar = IN[0]
2 secao_consolo = IN[1]

```

```

3
4 # Seleciona linhas da secao da consola, a partir da verificacao de paralelismo
   entre os pontos iniciais e finais da linha
5 if round(secao_consolo[3].EndPoint.X, 2) - round(secao_consolo[3].StartPoint.X, 2)
   == 0:
6     lado = 1
7     consolo_X = secao_consolo[3]
8     consolo_Y = secao_consolo[1]
9 else:
10    lado = 0
11    consolo_X = secao_consolo[1]
12    consolo_Y = secao_consolo[3]
13
14 vetores_paralelos = ladosParalelos(consolo_X.Direction, secao_pilar)
15 d1 = distancia(consolo_X, vetores_paralelos[0])
16 d2 = distancia(consolo_X, vetores_paralelos[1])
17 lados_distantes = vetores_paralelos[0] if d1 > d2 else vetores_paralelos[1]
18
19 #cria os vertices do limite maximo do ferro N2
20 if lado == 1:
21     retangulo = []
22     retangulo.append(Point.ByCoordinates(consolo_X.StartPoint.X, lados_distantes.
23     EndPoint.Y, consolo_X.StartPoint.Z))
24     retangulo.append(Point.ByCoordinates(consolo_X.StartPoint.X, consolo_X.EndPoint
25     .Y, consolo_X.EndPoint.Z))
26     retangulo.append(Point.ByCoordinates(consolo_Y.StartPoint.X, consolo_X.EndPoint
27     .Y, consolo_X.StartPoint.Z))
28     retangulo.append(Point.ByCoordinates(consolo_Y.StartPoint.X, lados_distantes.
29     EndPoint.Y, consolo_X.EndPoint.Z))
30 else:
31     retangulo = []
32     retangulo.append(Point.ByCoordinates(lados_distantes.StartPoint.X, consolo_Y.
33     StartPoint.Y, consolo_X.EndPoint.Z))
34     retangulo.append(Point.ByCoordinates(consolo_X.StartPoint.X, consolo_Y.
35     StartPoint.Y, consolo_X.StartPoint.Z))
36     retangulo.append(Point.ByCoordinates(consolo_X.StartPoint.X, consolo_X.EndPoint
37     .Y, consolo_X.EndPoint.Z))
38     retangulo.append(Point.ByCoordinates(lados_distantes.StartPoint.X, consolo_X.
39     EndPoint.Y, consolo_X.StartPoint.Z))

```

Script 3.6: Criar os vértices do limite da armadura de costura

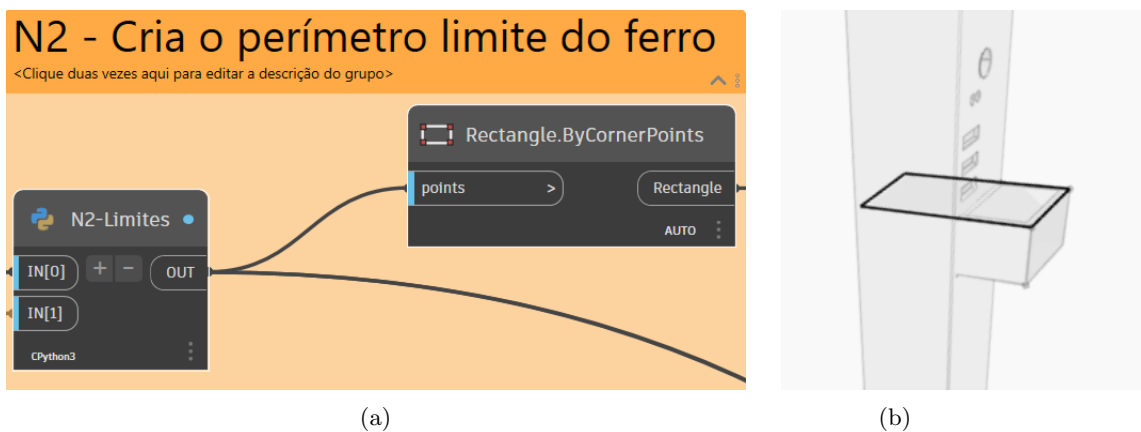


Figura 3.25: Limites da armadura de costura: (a) Processo para criar o limite da armadura de costura; (b) Resultado do limites da armadura de costura criado pelo script.

A partir desse limite, foram aplicados deslocamentos e afastamentos para criar uma curva que respeita os preceitos definidos pelas normas Normas Técnicas (ABNT) [2] e Normas Técnicas (ABNT) [3], conforme discutido no Capítulo 2. De modo geral, esse processo envolveu três etapas: a primeira foi aplicar um *offset* à curva no valor do cobrimento; em seguida, foi calculada a altura livre da consola (Equação 3.1), que seria a altura em que a armadura de costura pode ser aplicada de forma a respeitar o cobrimento e evitar conflitos com demais armaduras; e, por fim, a partir da altura livre, foi calculado o afastamento vertical (Equação 3.2). A Figura 3.26 mostra esse

procedimento de forma mais detalhada. Além disso, destaca-se que, nesse processo, foi necessário converter unidades de centímetros para milímetros e vice-versa; por isso, algumas variáveis estão sendo divididas ou multiplicadas por dez.

$$alturaLivre = (CONSOLO_Altura \cdot 10) - 2 \cdot (cobrimento + N3_Diametro + N1_Diametro + N5_Diametro) \quad (3.1)$$

$$afastamentoVerticalN2 = -\frac{CONSOLO_Altura - (CONSOLO_alturaLivre/10)}{2} - (N5_Diametro/10) \quad (3.2)$$



Figura 3.26: Ajuste das dimensões da curva base para a criação da armadura de costura.¹

A partir da curva ajustada, foi criado um vetor normal ao plano dessa geometria (Ver procedimento na Figura 3.27) para indicar o sentido de aplicação do grupo de armaduras. Por fim, a última etapa para a criação das armaduras de costura foi conectar essas informações aos nós *Create.FromCurves* e *SetLayout.AsFixedNumber*, conforme explicado na Seção 3.4.1. Nessa etapa, é importante destacar que foi utilizado um recurso semelhante ao descrito na Seção 3.5.1 para atribuir um elemento ao nome do varão. No entanto, nesse caso, foi necessário associar um elemento existente a um tipo de gancho, evitando que esse parâmetro fixo seja resetado ao selecionar outra consola. Esse processo é detalhado na Figura 3.28.

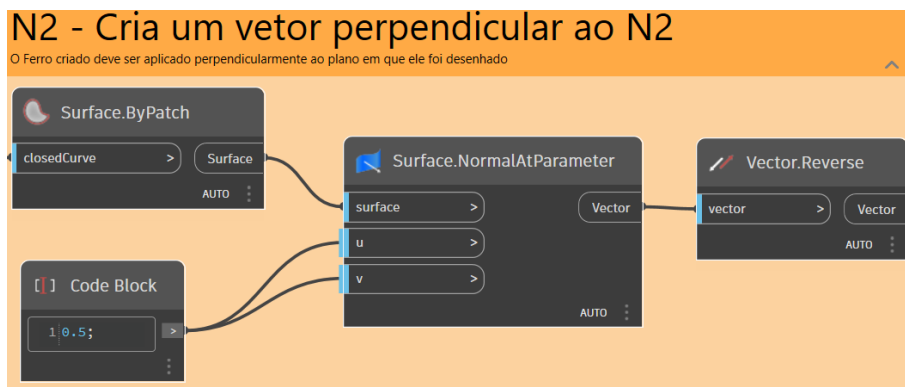


Figura 3.27: Criação de um vetor no sentido de aplicação da armadura de costura.¹

¹A imagem tem como objetivo mostrar a globalidade da etapa, de forma que o conteúdo escrito não é relevante.

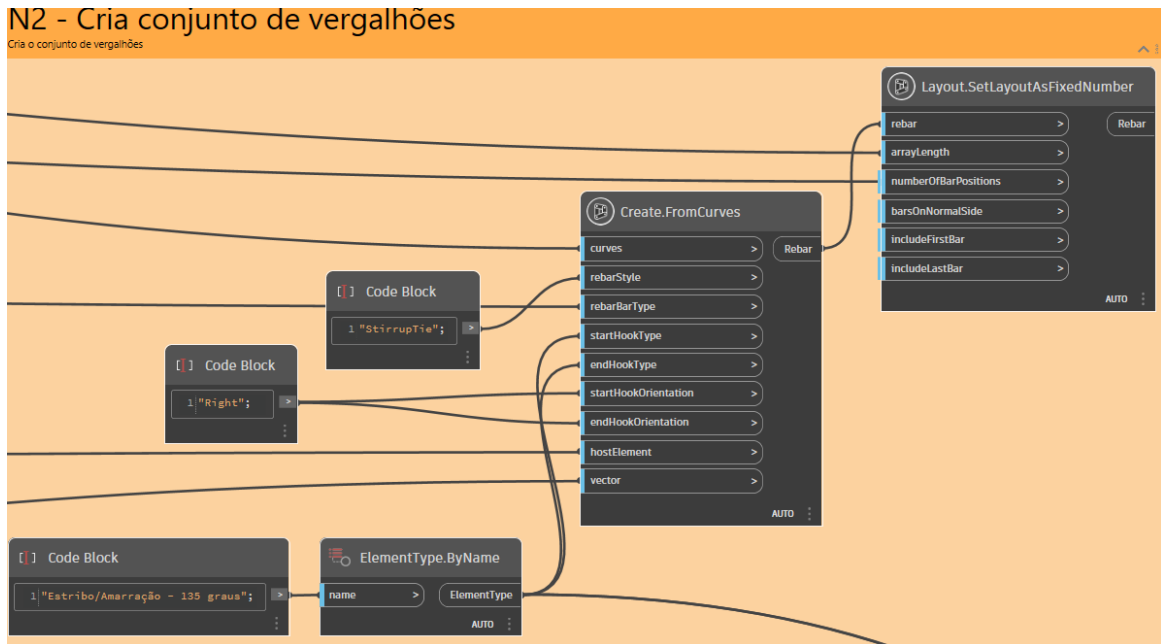


Figura 3.28: Criação do grupo de armaduras de costura.¹

3.5.3.2 Armadura transversal

A armadura transversal foi detalhada utilizando procedimentos semelhantes aos empregados na modelagem da armadura de costura (ver Seção 3.5.3.1), tais como a criação de curvas base, aplicação de deslocamentos conforme as normas Normas Técnicas (ABNT) [2] e Normas Técnicas (ABNT) [3], e criação dos grupos de armaduras a partir dos nós *Create.FromCurves* e *SetLayout.AsFixedNumber*. Por conta disso, nesta seção serão abordadas apenas as diferenças na abordagem em relação à armadura de costura.

Devido ao foco desta rotina ser a criação de consolas retangulares, a modelagem da armadura transversal foi uma tarefa simples, pois essa armadura possui dimensões fixas. Como a armadura transversal fica completamente contida dentro da geometria da consola, foi decidido usar o perímetro da face frontal da consola como curva base. Assim como na armadura de costura, foi calculada a profundidade livre, que é a medida da profundidade disponível dentro da consola após considerar o cobrimento de betão (Equação 3.3), e a partir dela, o afastamento frontal, que determina a posição da armadura dentro da consola (Equação 3.4).

$$\begin{aligned} \text{profundidadeLivre} = & (\text{CONSOLO_Profundidade} \cdot 10) \\ & - 2 \cdot (\text{Cobrimento} + N5_Diametro) + N1_Diametro + N2_Diametro \end{aligned} \quad (3.3)$$

$$\text{afastamentoFrontal} = - \frac{\text{CONSOLO_Profundidade} - (\text{CONSOLO_Profundidade_Util}/10)}{2} \quad (3.4)$$

Além disso, como a armadura possui afastamentos verticais e horizontais diferentes devido à sua posição no interior da armadura de costura, foi decidido criar um script personalizado (ver Script 3.7) dentro de um nó Python. Diferentemente da armadura de costura, todas essas transformações são feitas por esse nó Python.

¹A imagem tem como objetivo mostrar a globalidade da etapa, de forma que o conteúdo escrito não é relevante.

```

1 import sys
2 import clr
3 clr.AddReference('ProtoGeometry')
4 from Autodesk.DesignScript.Geometry import *
5
6 # Inputs do Dynamo
7 curva_retangular = IN[0] # Curva retangular original
8 offset_lateral = IN[1] # Offset lateral
9 offset_vertical = IN[2] # Offset vertical
10 offset_frontal = IN[4] # Deslocamento frontal
11 plano_curva = IN[3] # Verifica se a curva esta no plano XZ ou YZ
12 direcao_curva = IN[5] # Face do pilar a qual a consola esta aplicada
13
14 inferior_esquerdo = curva_retangular[0].StartPoint
15 inferior_direito = curva_retangular[1].StartPoint
16 superior_direito = curva_retangular[2].StartPoint
17 superior_esquerdo = curva_retangular[3].StartPoint
18
19
20 if plano_curva == 0:
21     if direcao_curva == 0:
22         retangulo = []
23         retangulo.append(Point.ByCoordinates(inferior_esquerdo.X - offset_lateral,
24 inferior_esquerdo.Y - offset_frontal, inferior_esquerdo.Z - offset_vertical))
25         retangulo.append(Point.ByCoordinates(inferior_direito.X + offset_lateral,
26 inferior_direito.Y - offset_frontal, inferior_direito.Z - offset_vertical))
27         retangulo.append(Point.ByCoordinates(superior_direito.X + offset_lateral,
28 superior_direito.Y - offset_frontal, superior_direito.Z + offset_vertical))
29         retangulo.append(Point.ByCoordinates(superior_esquerdo.X - offset_lateral,
30 superior_esquerdo.Y - offset_frontal, superior_esquerdo.Z + offset_vertical))
31
32     elif direcao_curva == 1:
33         retangulo = []
34         retangulo.append(Point.ByCoordinates(inferior_esquerdo.X + offset_lateral,
35 inferior_esquerdo.Y + offset_frontal, inferior_esquerdo.Z - offset_vertical))
36         retangulo.append(Point.ByCoordinates(inferior_direito.X - offset_lateral,
37 inferior_direito.Y + offset_frontal, inferior_direito.Z - offset_vertical))
38         retangulo.append(Point.ByCoordinates(superior_direito.X - offset_lateral,
39 superior_direito.Y + offset_frontal, superior_direito.Z + offset_vertical))
40         retangulo.append(Point.ByCoordinates(superior_esquerdo.X + offset_lateral,
41 superior_esquerdo.Y + offset_frontal, superior_esquerdo.Z + offset_vertical))
42
43     else:
44         if direcao_curva == 2:
45             retangulo = []
46             retangulo.append(Point.ByCoordinates(inferior_esquerdo.X - offset_frontal,
47 inferior_esquerdo.Y + offset_lateral, inferior_esquerdo.Z - offset_vertical))
48             retangulo.append(Point.ByCoordinates(inferior_direito.X - offset_frontal,
49 inferior_direito.Y - offset_lateral, inferior_direito.Z - offset_vertical))
50             retangulo.append(Point.ByCoordinates(superior_direito.X - offset_frontal,
51 superior_direito.Y - offset_lateral, superior_direito.Z + offset_vertical))
52             retangulo.append(Point.ByCoordinates(superior_esquerdo.X - offset_frontal,
53 superior_esquerdo.Y + offset_lateral, superior_esquerdo.Z + offset_vertical))
54
55         elif direcao_curva == 3:
56             retangulo = []
57             retangulo.append(Point.ByCoordinates(inferior_esquerdo.X + offset_frontal,
58 inferior_esquerdo.Y - offset_lateral, inferior_esquerdo.Z - offset_vertical))
59             retangulo.append(Point.ByCoordinates(inferior_direito.X + offset_frontal,
60 inferior_direito.Y + offset_lateral, inferior_direito.Z - offset_vertical))
61             retangulo.append(Point.ByCoordinates(superior_direito.X + offset_frontal,
62 superior_direito.Y + offset_lateral, superior_direito.Z + offset_vertical))
63             retangulo.append(Point.ByCoordinates(superior_esquerdo.X + offset_frontal,
64 superior_esquerdo.Y - offset_lateral, superior_esquerdo.Z + offset_vertical))
65
66 # Saida para o Dynamo
67 OUT = retangulo

```

Script 3.7: Ajusta a curva base para criação da armadura transversal

Neste caso, serão explicadas algumas abordagens feitas no Script 3.7. Durante a análise dos arquivos, especialmente os arquivos de pilares, notou-se que estes sempre possuem as faces laterais paralelas ao plano XZ ou YZ. Para ilustrar essa observação, ver a Figura 3.29, onde os eixos são

indicados nas seguinte cores: Vermelho eixo X, verde eixo Y e azul eixo Z.

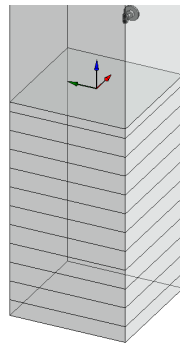


Figura 3.29: Origem interna dos pilares.

Devido a essa característica, foi criado um nó em Python para verificar em qual face a consola está aplicada (ver Script 3.8). Nesse script, os pontos zero e um são definidos como o vértice posterior esquerdo e o vértice frontal esquerdo da curva limite da armadura de costura, respectivamente, conforme indicado na Figura 3.30. Essa configuração garante que a armadura transversal seja posicionada corretamente, independentemente da orientação da consola.

```
1 import sys
2 import clr
3 clr.AddReference('ProtoGeometry')
4 from Autodesk.DesignScript.Geometry import *
5
6 # As entradas para este nó serão armazenadas como uma lista nas variáveis IN.
7 ponto1 = IN[0]
8 ponto2 = IN[1]
9
10 if ponto1.X - ponto2.X == 0:
11     if ponto1.Y - ponto2.Y > 0:
12         booleana = 0;
13     else:
14         booleana = 1;
15 else:
16     if ponto1.X - ponto2.X > 0:
17         booleana = 2;
18     else:
19         booleana = 3;
20
21 OUT = booleana
```

Script 3.8: Identificação da face em que a consola está posicionada

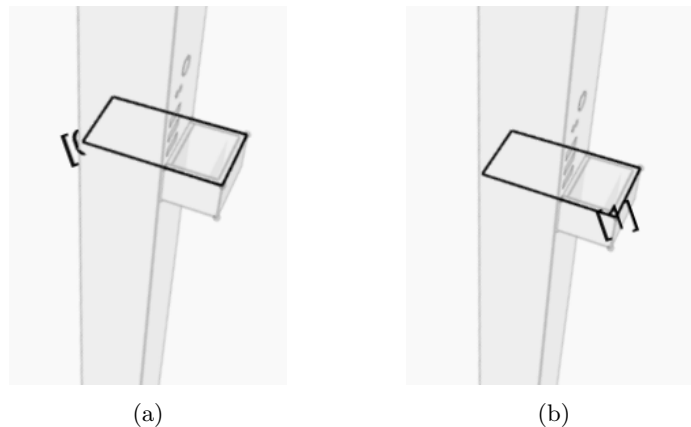


Figura 3.30: Pontos da curva base da armadura de costura utilizados no Script 3.8: (a) Ponto zero; (b) Ponto um.

Esse nó Python realiza a verificação das faces analisando as coordenadas dos vértices e determinando sua relação com os planos XZ e YZ. Isso assegura que a armadura transversal se ajuste adequadamente à geometria da consola e se alinhe com a armadura de costura. Esse processo completo é indicado na Figura 3.31.

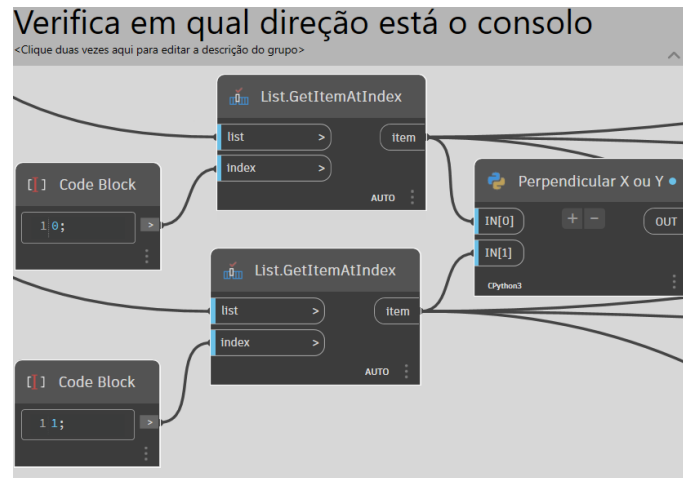


Figura 3.31: Processo para identificação da face em que a consola está aplicada.¹

3.5.3.3 Tirante

Os tirantes foram detalhados seguindo procedimentos muito parecidos aos seguidos no detalhamento da armadura de costura (3.5.3.1) e da armadura transversal (3.5.3.2). Por conta disso, nesta seção serão abordadas apenas as diferenças de abordagem mais relevantes. Assim como nas armaduras anteriores, os afastamentos foram calculados a partir da definição da largura útil (Equação 3.5). Com base nessa largura útil, foi calculado o afastamento lateral utilizando a Equação 3.6.

$$larguraLivre = CONSOLO_Largura / (N1_Quantidade + 1) \quad (3.5)$$

$$afastamentoLateral = N1_Espaçamento \cdot (N1_Quantidade - 1) \cdot 10 \quad (3.6)$$

Devido à complexidade geométrica da armadura de tirante, optou-se por desenvolver um nó personalizado em Python para sua criação. A geração da curva da armadura é baseada nos pontos zero e um da curva base da armadura de costura, conforme mostrado na Figura 3.30. Além disso, o script implementado ajusta os valores dos afastamentos e offsets, de acordo com a face onde a consola está aplicada, para manter a consistência das fórmulas, conforme detalhado no Script 3.9.

```

1 # Identifica a face em que a consola esta aplicada inverte o valor para manter
  consistencia da formula
2 if (perpendicular == 1 or perpendicular == 3):
3     offset_horizontal = (-1) * offset_horizontal
4     offset_lateral = (-1) * offset_lateral
5     acrescimo = (-1) * acrescimo
6     meio_N1_diametro = (-1) * meio_N1_diametro

```

Script 3.9: Ajuste do afastamento com base na face em que a consola está aplicada

Ao contrário das armaduras transversais e de costura, a criação da geometria do tirante exige um maior número de parâmetros devido à sua complexidade. O nó Python desenvolvido requer 12 valores de entrada específicos para gerar corretamente sua geometria. Dessa forma, esses valores são necessários para definir os afastamentos, a ancoragem e outros aspectos geométricos do tirante. A Tabela 3.9 apresenta e explica cada um desses valores de entrada em detalhe.

¹A imagem tem como objetivo mostrar a globalidade da etapa, de forma que o conteúdo escrito não é relevante.

Tabela 3.9: Valores de entrada do nó Python que cria a geometria da armadura de tirantes

Valores	Nome	Descrição
IN[0]	ponto1	Vértice posterior esquerdo da curva base da armadura de costura
IN[1]	ponto2	Vértice frontal esquerdo da curva base da armadura de costura
IN[2]	ferro_direto	Define se o varão será armado como laço vertical ou barra soldada
IN[3]	ancoragem	Comprimento de ancoragem do tirante
IN[4]	linha_altura_consolo	Linha que define a altura da consola
IN[5]	linha_profundidade_consolo	Linha que define a profundidade da consola
IN[6]	acrescimo	Comprimento do excedente da armadura na base da consola
IN[7]	offset_vertical	Offset vertical
IN[8]	offset_horizontal	Offset horizontal
IN[9]	offset_lateral	Afastamento lateral da armadura
IN[10]	perpendicular	Face de aplicação da consola
IN[11]	meio_N1_diametro	Raio da armadura do tirante

A partir desses valores de entrada, a criação dos vértices foi realizada utilizando comandos aninhados. A primeira verificação identifica a qual plano a profundidade da consola é perpendicular. Dentro dessa verificação, outros comandos aninhados determinam se a armadura deve ser gerada como um laço vertical ou como uma barra soldada em sua extremidade. Essa estrutura lógica garante que a armadura seja modelada de acordo com as especificações geométricas estipuladas pela Normas Técnicas (ABNT) [3] e Normas Técnicas (ABNT) [2]. O código completo dessa implementação pode ser observado no Script 3.10.

```

1 # Cria os vertices da curva que define o tirante
2 if ponto1.X - ponto2.X == 0:
3     profundidade_consolo = linha_profundidade_consolo.StartPoint.Y -
4     linha_profundidade_consolo.EndPoint.Y
5
6     if ferro_direto == 1:
7         linha = []
8         linha.append(Point.ByCoordinates(ponto1.X + offset_lateral, ponto1.Y -
9         offset_horizontal, ponto1.Z - ancoragem - offset_vertical))
10        linha.append(Point.ByCoordinates(ponto1.X + offset_lateral, ponto1.Y -
11        offset_horizontal, ponto1.Z - offset_vertical))
12        linha.append(Point.ByCoordinates(ponto2.X + offset_lateral, ponto2.Y +
13        offset_horizontal, ponto2.Z - offset_vertical))
14        linha.append(Point.ByCoordinates(ponto2.X + offset_lateral, ponto2.Y +
15        offset_horizontal, ponto2.Z - altura_consolo + offset_vertical))
16        linha.append(Point.ByCoordinates(ponto2.X + offset_lateral, ponto2.Y +
17        profundidade_consolo + acrescimo + offset_horizontal, ponto2.Z - altura_consolo
18        + offset_vertical))
19    else:
20        linha = []
21        linha.append(Point.ByCoordinates(ponto1.X + offset_lateral, ponto1.Y -
22        offset_horizontal, ponto1.Z - ancoragem - offset_vertical))
23        linha.append(Point.ByCoordinates(ponto1.X + offset_lateral, ponto1.Y -
24        offset_horizontal, ponto1.Z - offset_vertical))
25        linha.append(Point.ByCoordinates(ponto2.X + offset_lateral, ponto2.Y +
26        offset_horizontal - meio_N1_diametro, ponto2.Z - offset_vertical))
27    else:
28        profundidade_consolo = linha_profundidade_consolo.StartPoint.X -
29        linha_profundidade_consolo.EndPoint.X
30
31    if ferro_direto == 1:
32        linha = []

```

```

24     linha.append(Point.ByCoordinates(ponto1.X - offset_horizontal, ponto1.Y -
25     offset_lateral, ponto1.Z - ancoragem - offset_vertical))
26     linha.append(Point.ByCoordinates(ponto1.X - offset_horizontal, ponto1.Y -
27     offset_lateral, ponto1.Z - offset_vertical))
28     linha.append(Point.ByCoordinates(ponto2.X + offset_horizontal, ponto2.Y -
29     offset_lateral, ponto2.Z - offset_vertical))
30     linha.append(Point.ByCoordinates(ponto2.X + offset_horizontal, ponto2.Y -
31     offset_lateral, ponto2.Z - altura_consolo + offset_vertical))
32     linha.append(Point.ByCoordinates(ponto2.X + profundidade_consolo +
33     acrescimo + offset_horizontal, ponto2.Y - offset_lateral, ponto2.Z -
34     altura_consolo + offset_vertical))
35
36     else:
37         linha = []
38         linha.append(Point.ByCoordinates(ponto1.X - offset_horizontal, ponto1.Y -
39         offset_lateral, ponto1.Z - ancoragem - offset_vertical))
40         linha.append(Point.ByCoordinates(ponto1.X - offset_horizontal, ponto1.Y -
41         offset_lateral, ponto1.Z - offset_vertical))
42         linha.append(Point.ByCoordinates(ponto2.X + offset_horizontal -
43         meio_N1_diametro, ponto2.Y - offset_lateral, ponto2.Z - offset_vertical))
44
45 # Atribua a sua saída para a variável OUT.
46 OUT = linha

```

Script 3.10: Identificação da face em que a consola está posicionada

É importante salientar que o comprimento de amarração foi calculado respeitando os valores mínimos apresentados na Seção 2.7.2. Todavia, para seguir o padrão estabelecido pela empresa, esse valor foi arredondado para o múltiplo de dez mais próximo. O código que realiza esse arredondamento é apresentado no Script 3.11.

```

1 (Math.Round(((N1_Diametro / 10) * fatorDeMultiplicao) / 10) * 10) + 10 - (
2     N1_Diametro / 20)

```

Script 3.11: Cálculo do comprimento de ancoragem dos tirantes.

Outro ponto relevante é que o comprimento da dobra inferior do laço foi calculado por meio de um nó Python personalizado (Script 3.12). Para seguir o padrão da empresa, esse comprimento foi definido como 10 cm menor que o lado da armadura de costura. Nesse script, as variáveis “ponto1” e “ponto2” indicam, respetivamente, os pontos posterior esquerdo e frontal esquerdo da curva base da armadura de costura.

```

1 import sys
2 import clr
3 clr.AddReference('ProtoGeometry')
4 from Autodesk.DesignScript.Geometry import *
5
6 # As entradas para este no serao armazenadas como uma lista nas variaveis IN.
7 ponto1 = IN[0]
8 ponto2 = IN[1]
9 consolo_profundidade = IN[2]
10 comprimento = (IN[3] / 10)
11
12 if ponto1.X - ponto2.X == 0:
13     comprimento = round(abs(ponto1.Y - ponto2.Y) - consolo_profundidade - 2 *
14     comprimento - 10, 0);
15 else:
16     comprimento = round(abs(ponto1.X - ponto2.X) - consolo_profundidade - 2 *
17     comprimento - 10, 0);
18
19 # Comprimento do laço
20 OUT = comprimento

```

Script 3.12: Comprimento do laço inferior do tirante.

3.5.3.4 Barra soldada e armadura complementar

A criação da barra soldada (N5) foi realizada por meio de um processo bastante simplificado. Inicialmente, foi selecionada a face superior da consola, conforme indicado na Figura 3.32. Em

seguida, foi aplicado um offset com valor igual ao cobrimento, conforme especificado pela norma brasileira Normas Técnicas (ABNT) [3].

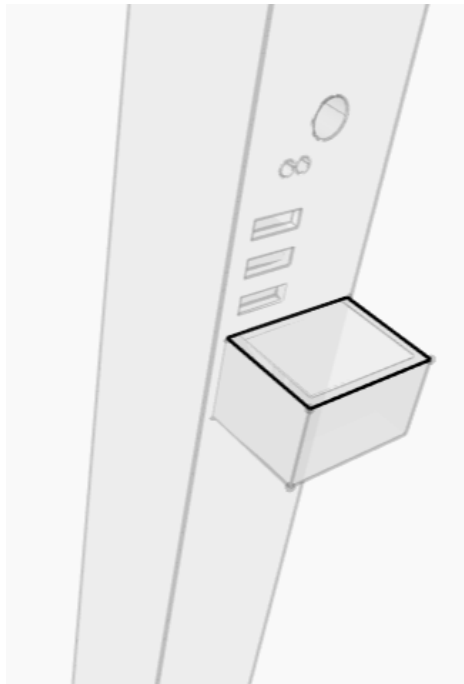


Figura 3.32: Seleção da face superior da consola.

Posteriormente, foi aplicado um afastamento vertical, conforme indicado na Equação 3.7, para posicionar a barra corretamente, cde forma semelhante ao procedimento utilizado para a armadura de costura, . Por fim, como a barra N5 é reta, foi selecionada a aresta frontal do retângulo ajustado, essas etapas são indicadas na Figura 3.33.

$$a_{fastamentoVerticalN5} = -[\text{cobrimento} + N1_Diametro + N3_Diametro + (N5_Diametro/2)]/10 \quad (3.7)$$

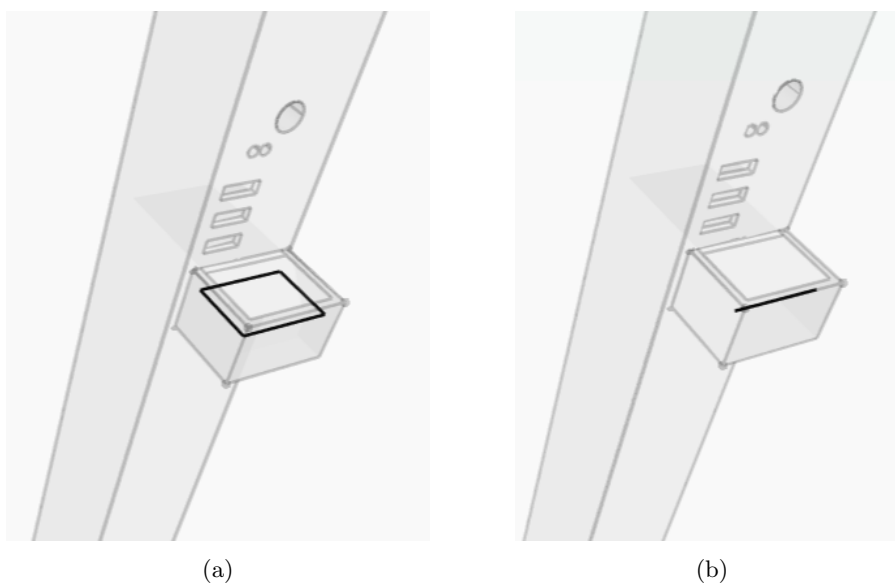


Figura 3.33: Criação da geometria da barra soldada: (a) Curva ajustada; (b) Seleção da face frontal da curva.

A metodologia para a criação da armadura complementar (N4) seguiu o mesmo procedimento utilizado na criação da armadura dos tirantes. Sendo que, os vértices da curva foram os três últimos pontos quando a armadura é detalhada como um laço vertical. Devido a essa similaridade, não se viu necessidade em explicar esse processo.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Neste capítulo, serão analisados e discutidos os resultados obtidos ao longo do desenvolvimento e implementação das metodologias descritas nos capítulos anteriores. Inicialmente, são explicados os procedimentos necessários para a correta execução do script desenvolvido em Dynamo. Em seguida, são apresentados e avaliados os resultados obtidos.

4.1 Método de utilização do script

Para facilitar a execução de scripts, o Revit possui uma ferramenta chamada “Reprodutor Dynamo”, acessível através da aba Gerenciar do Revit, conforme indicado na Figura 4.1. Nessa interface, o usuário não tem acesso direto ao código, podendo apenas definir elementos que foram marcados como “entrada” pelo autor do script. Além da agilidade proporcionada por essa função, ela também evita alterações equivocadas no código, o que aumenta a confiabilidade dos scripts executados.

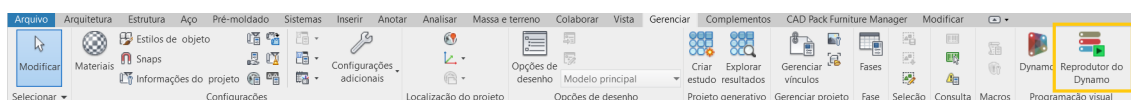


Figura 4.1: Acesso ao reprodutor Dynamo através do menu gerenciar

A interface do Reprodutor Dynamo é bastante simplificada e intuitiva. Para executar um script por meio dela, o usuário precisa, primeiramente, selecionar a pasta onde o código se encontra. O programa então retorna uma lista com todos os scripts presentes nessa pasta, conforme indicado na Figura 4.2.



Figura 4.2: Interface do reprodutor Dynamo

Em seguida, o usuário deve selecionar o script desejado através de um duplo clique sobre o título deste. Neste caso, o script desenvolvido foi nomeado como “Armaduras Consolo-V04_NEW”. Ao fazer isso, ao usuário é apresentada uma janela onde é possível selecionar os elementos de entrada. No script desenvolvido, o usuário precisa fornecer apenas três informações: o diretório da folha Excel, o elemento hospedeiro da consola que se deseja detalhar (neste caso, o pilar), e, por fim, o próprio elemento a ser detalhado (a consola). Essas etapas são exemplificadas na Figura 4.3. Após a seleção desses elementos, basta pressionar o botão intitulado “Executar” para o script fazer o detalhamento automático da armadura da consola selecionada.

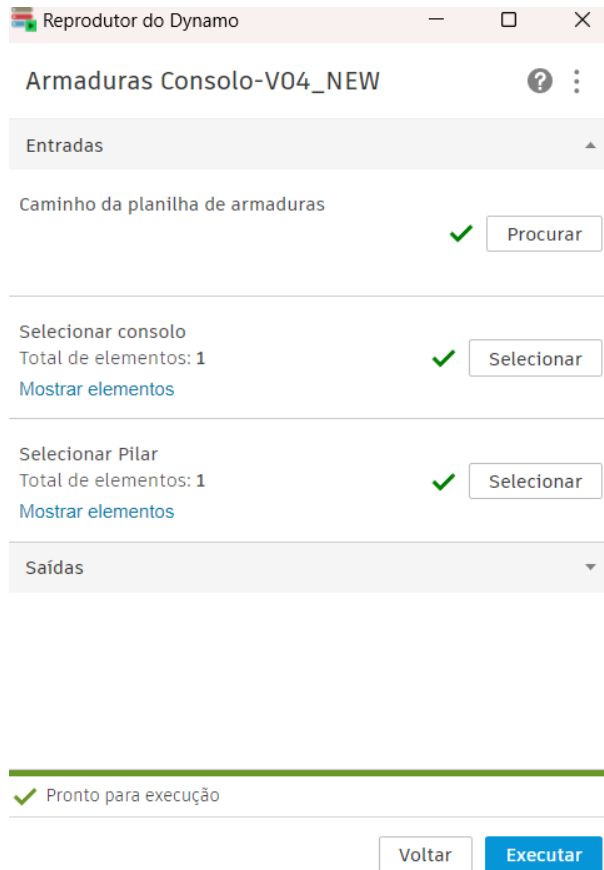


Figura 4.3: Dados de entrada para execução do script através do reprodutor Dynamo

4.1.1 Cuidados ao utilizar o script

Assim como em qualquer programa, é necessário adotar certos cuidados para garantir que ele funcione conforme o esperado. Portanto, alguns cuidados específicos em relação à utilização deste script são descritos a seguir.

- A geometria da consola e do pilar não podem estar na montagem. Quando um elemento é selecionado a partir do reprodutor Dynamo, não é possível selecionar elementos filhos. Por conta disso, a consola e o pilar não podem fazer parte da montagem no momento da execução do script e devem ser inseridos na montagem após o seu detalhamento ter sido concluído.
- A folha Excel precisa estar organizada conforme explicado na Seção 3.5.1. Caso contrário, o script não conseguirá importar os dados corretamente e, conseqüentemente, não realizará o detalhamento das armaduras da consola.
- O parâmetro “*CONSOLO-Tipo*” deve existir e estar configurado conforme explicado na Seção 3.5.1. Assim como a disposição da folha excel, este parâmetro é responsável por selecionar os dados corretamente. Como este parâmetro foi criado durante o desenvolvimento deste script, é necessário atualizar a família de consolas em todo o projeto.

4.2 Comparativo do detalhamento das armaduras

A partir das armaduras geradas pelo script desenvolvido, foram criados os detalhamentos das consolas C01 e C02, conforme indicado nas Figuras 4.4 e 4.5, respectivamente. Vale ressaltar que, para fins de comparação, são apresentados apenas os resultados obtidos com o detalhamento dessas duas consolas, pois elas são mais representativas no projeto fornecido, conforme discutido na Seção 3.2.2. Todavia, destaca-se que o script foi testado em outras consolas retangulares simples e demonstrou funcionar de maneira consistente.

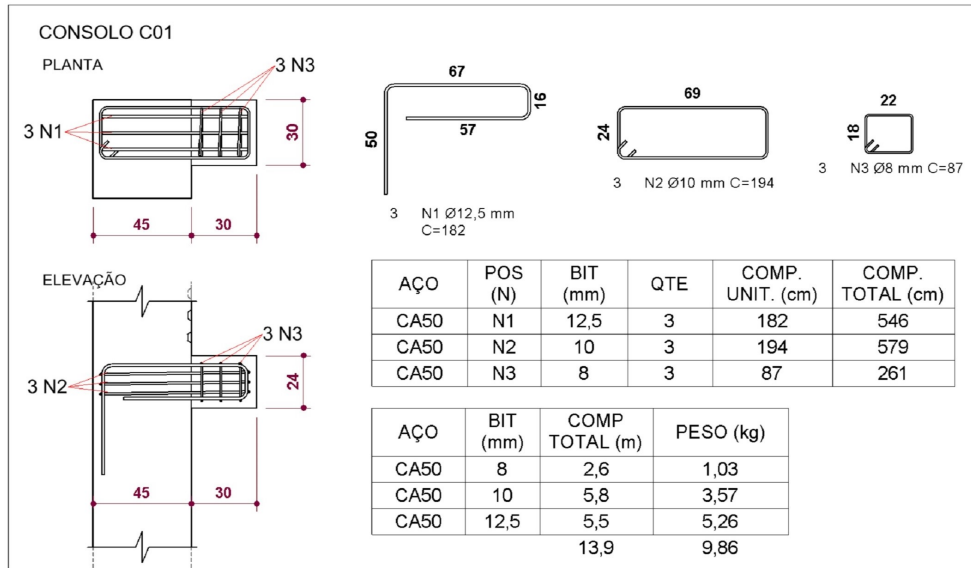


Figura 4.4: Consola C01 detalhada, no Revit, a partir das armaduras geradas pelo script em Dynamo

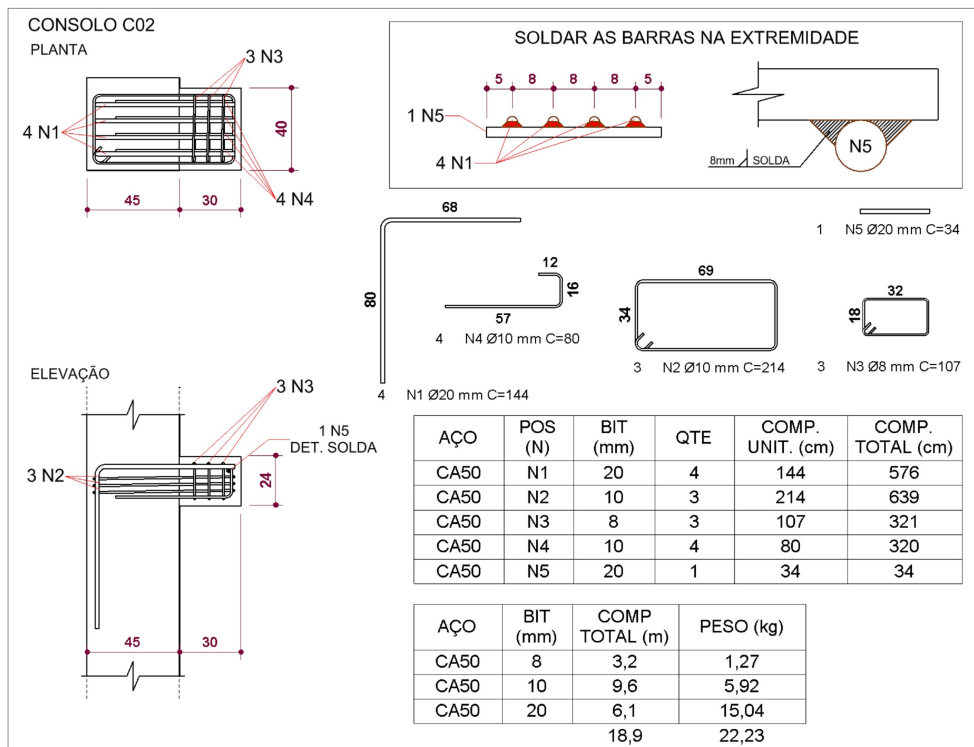


Figura 4.5: Consola C02 detalhada, no Revit, a partir das armaduras geradas pelo script em Dynamo

4.2.1 Armadura de costura

Para a consola C01, a Figura 4.6a representa o detalhamento da armadura de costura feito no Revit a partir das armaduras criadas pelo script desenvolvido no Dynamo. Em contraste, a Figura 4.6b mostra o detalhamento da armadura elaborado em CAD (conforme o projeto de estudo de implantação BIM cedido por empresa). Ao comparar os dois detalhamentos, observa-se que o comprimento total da armadura detalhada em CAD é 15 cm inferior ao comprimento da armadura gerada pelo script no Dynamo. Esta diferença no comprimento da armadura de costura pode ter ocorrido devido a maior precisão no quantitativo dos comprimentos com relação aos raios de curvatura. Além disso, também se observou que no detalhamento em CAD não estava sendo considerada a adição do comprimento da ponta reta do gancho na extremidade do estribo.

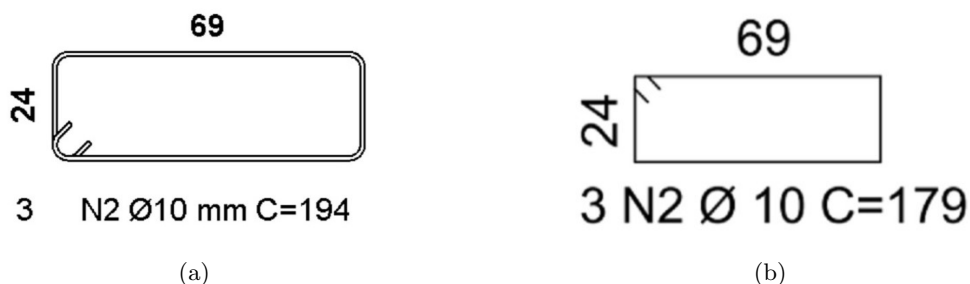


Figura 4.6: Detalhe da armadura de costura da consola C01: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD pela empresa.

Assim como observado na consola C01, a consola C02 apresenta uma diferença de 15 cm no comprimento total da armadura de costura detalhada pelo Revit, utilizando as armaduras criadas pelo script no Dynamo (Figura 4.7a), em comparação com o detalhamento feito em CAD (Figura 4.7b). Esta diferença pode ser atribuída a dois fatores principais: primeiro, o detalhamento em CAD não considera o comprimento da ponta reta do gancho ao calcular o comprimento total da barra; segundo, não foi descontado o comprimento devido a raio de curvatura de duas dobras do estribo.

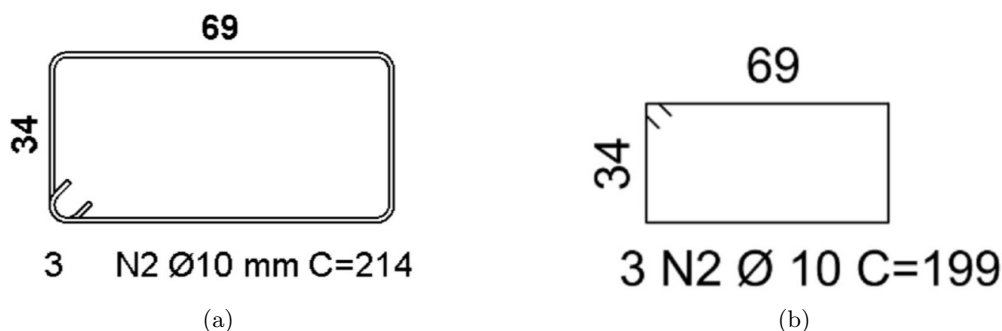


Figura 4.7: Detalhe da armadura de costura da consola C02: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD (conforme projeto de estudo de implantação BIM cedido por empresa).

Adicionalmente, conforme ilustrado na Figura 4.8, verifica-se que o posicionamento da armadura de costura está correto em ambos os casos: consolas retangulares simples com tirantes em laço vertical e consolas retangulares com tirantes com barras soldadas. Uma vez que toda a armadura de costura está contida em dois terços da altura útil da consola, atendendo às especificações da Normas Técnicas (ABNT) [3].

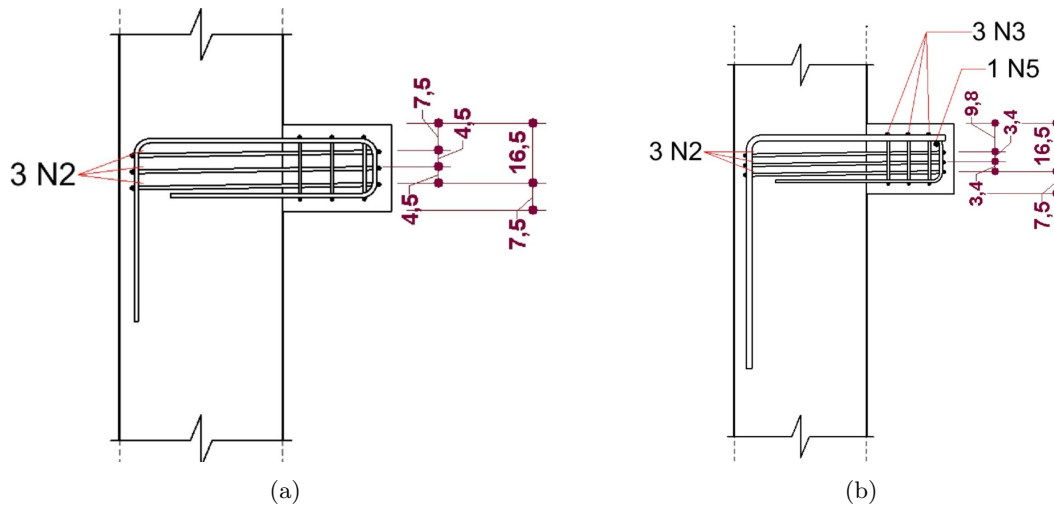


Figura 4.8: Seção transversal das consolas: (a) C01; (b) C02.

4.2.2 Armadura transversal

As Figuras 4.9 e 4.10 apresentam, respectivamente, os detalhes das armaduras transversais das consolas C01 e C02, destacando duas abordagens diferentes para o detalhamento: uma utilizando o Revit com um script desenvolvido no Dynamo e a outra feita em CAD (conforme projeto de estudo de implantação BIM cedido por empresa).



Figura 4.9: Detalhe da armadura transversal da consola C01: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD (conforme projeto de estudo de implantação BIM cedido por empresa).

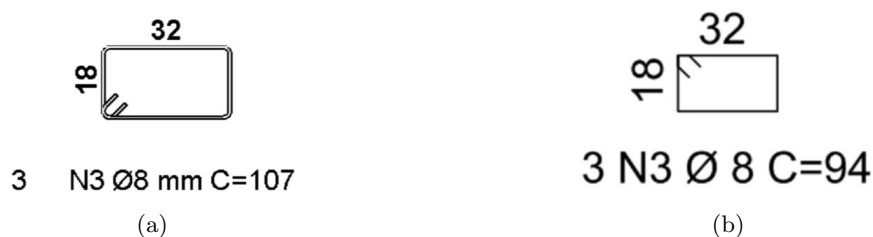


Figura 4.10: Detalhe da armadura transversal da consola C02: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD (conforme o projeto de estudo de implantação BIM cedido por empresa).

Em ambos os casos, foi observada uma diferença de 13 cm no comprimento total entre as duas abordagens. Essa diferença pode ser atribuída a diferentes métodos de consideração do raio de curvatura dos varões, à precisão nos cálculos das dimensões e consideração do comprimento dos ganchos, conforme explicado no Seção 4.2.1. Destaca-se que a diferença de 2 cm, quando comparada com a armadura de costura, deve-se ao diâmetro inferior dos varões da armadura transversal.

4.2.3 Tirante em laço vertical

A Figura 4.11 apresenta o detalhamento do tirante da consola C01 em duas abordagens distintas: (a) utilizando o script desenvolvido no Dynamo e detalhado no Revit, e (b) utilizando o detalhamento em CAD (conforme o projeto de estudo de implantação BIM cedido por empresa).



Figura 4.11: Detalhe do tirante da consola C01: (a) Armadura de detalhada no Revit a partir do script desenvolvido em Dynamo; (b) Armadura de detalhada em CAD (conforme o projeto de estudo de implantação BIM cedido por empresa).

Ao observar a Figura 4.11a, é possível notar que a armadura detalhada no Revit possui um comprimento total de 182 cm. Esta armadura é composta por três segmentos principais: um segmento horizontal de 67 cm, um segmento vertical de 50 cm e um segmento horizontal inferior de 57 cm, além de um gancho de 16 cm na extremidade inferior direita. Por outro lado, a Figura 4.11b apresenta o detalhamento da armadura em CAD, que possui um comprimento total de 187 cm. A diferença de 5 cm em relação ao detalhamento no Revit é atribuída a diferentes métodos de arredondamento e à consideração do raio de curvatura dos varões, como discutido na Seção 4.2.1.

É importante salientar que a armadura em CAD também apresenta os mesmos segmentos e comprimentos que a gerada pelo script em Dynamo. Isso era esperado, uma vez que foram considerados os padrões de projeto da empresa no desenvolvimento do script, conforme detalhado no Capítulo 3. Portanto, as discrepâncias observadas estão relacionadas à precisão do cálculo do diâmetro do raio de curvatura da armadura.

4.2.4 Tirante com barra soldada na extremidade

Na Figura 4.12, referente ao tirante da consola C02, que foi detalhado com uma barra soldada na extremidade. O comprimento total da armadura detalhada no Revit (Figura 4.12a) é de 144 cm, enquanto a armadura detalhada em CAD (Figura 4.12b) apresenta um comprimento ligeiramente menor, de 141 cm. Essa discrepância de 3 cm é atribuída a diferenças na consideração do raio de curvatura dos varões. É importante destacar que, embora as dimensões individuais dos segmentos da armadura (68 cm e 80 cm) sejam praticamente idênticas em ambas as representações, a soma desses segmentos resulta em comprimentos totais distintos devido a essas pequenas variações metodológicas.



Figura 4.12: Detalhe do tirante da consola C02: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD (conforme o projeto de estudo de implantação BIM cedido por empresa).

A Figura 4.13 apresenta o detalhamento da barra soldada na extremidade da consola C02. Observa-se que a armadura detalhada no Revit (Figura 4.13a) possui um comprimento total de 34 cm, enquanto a armadura detalhada em CAD (Figura 4.13b) apresenta um comprimento de 30 cm, resultando em uma diferença de 4 cm. Essa discrepância é atribuída ao fato de que, no desenvolvimento do script em Dynamo, o comprimento da barra de ancoragem soldada à extremidade do tirante foi considerado como o máximo possível, de forma a respeitar apenas os requisitos de recobrimento mínimo do projeto e o modelo de detalhamento especificado pela Normas Técnicas (ABNT) [3]. A equipe da empresa, por outro lado, pode ter optado por um comprimento ligeiramente inferior, em conformidade com procedimentos internos ou considerações práticas específicas, como limitações do processo fabril, que não foram acessíveis no desenvolvimento do script. Portanto, para garantir a aderência total às diretrizes do projeto e evitar inconsistências, optou-se por adotar o comprimento máximo no script em Dynamo.



Figura 4.13: Detalhe da barra soldada na extremidade da consola C02: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD pela empresa.

Por fim, a Figura 4.14 apresenta o detalhamento da barra complementar da consola C02. Observa-se que a armadura gerada no Revit (Figura 4.14a) possui um comprimento total de 80 cm, enquanto a armadura detalhada em CAD (Figura 4.14b) apresenta um comprimento total de 78 cm. Essa diferença de 2 cm decorre do fato de que o gancho na armadura detalhada no Revit foi dimensionado com 12 cm, em vez de 10 cm, como no detalhamento em CAD. Essa escolha foi feita porque o script utiliza as configurações de projeto padrão no detalhamento das armaduras, e uma dessas configurações define um comprimento mínimo de 12 cm para ganchos em armaduras de 10 mm de diâmetro. Considerou-se mais apropriado manter esse padrão estabelecido no projeto, em vez de ajustá-lo apenas para coincidir com as medidas adotadas no detalhamento em CAD.



Figura 4.14: Detalhe da barra complementar da consola C02: (a) Armadura no Revit a partir do script desenvolvido em Dynamo; (b) Armadura em CAD pela empresa.

4.3 Comparativo dos quantitativos de aço

As Tabelas 4.1 e 4.2 apresentam os quantitativos de aço para a consola C01, detalhados em CAD e gerados pelo Revit, respectivamente. Observa-se que o comprimento total das barras detalhadas em CAD é de 13 metros, enquanto no Revit esse comprimento aumenta para 13,9 metros, representando uma diferença de aproximadamente **6,9%** em relação ao valor calculado em CAD. Esse aumento no comprimento se reflete também nos pesos totais: 10 kg em CAD e 9,86 kg no Revit, uma diferença percentual de **1,4%**.

A diferença mais significativa ocorre na bitola de 10 mm, onde o Revit indica um comprimento de 5,8 metros, enquanto o CAD indica 5 metros, resultando em uma variação de **16%** a favor do Revit. Esta variação é atribuída aos possíveis aspectos mencionados em cada caso do item 4.2.

Tabela 4.1: Quantitativo da consola C01 feito em CAD

Aço	Bitola (mm)	Comprimento (m)	Peso (kg)
CA50	8	2	1
CA50	10	5	3
CA50	12,5	6	6
Total		13	10

Tabela 4.2: Quantitativo da consola C01 gerado pelo Revit

Aço	Bitola (mm)	Comprimento (m)	Peso (kg)
CA50	8	2,6	1,03
CA50	10	5,8	3,57
CA50	12,5	5,5	5,26
Total		13,9	9,86

Nas Tabelas 4.3 e 4.4, que se referem à consola C02, observa-se uma diferença similar. O comprimento total das barras no CAD é de 18 metros, enquanto no Revit é de 18,9 metros, uma diferença de **5%**. Em termos de peso total, o CAD apresenta 22 kg e o Revit 22,23 kg, uma variação de apenas **1%**.

Para a bitola de 10 mm, o comprimento reportado pelo Revit é de 9,6 metros, comparado aos 9 metros no CAD, resultando em uma diferença de **6,7%**. Esse padrão de diferença evidencia como o Revit costuma proporcionar um nível de detalhe superior, prestando maior atenção às dobras e curvaturas ao calcular o comprimento das barras.

Tabela 4.3: Quantitativo da consola C02 feito em CAD

Aço	Bitola (mm)	Comprimento (m)	Peso (kg)
CA50	8	3	1
CA50	10	9	6
CA50	20	6	15
Total		18	22

Tabela 4.4: Quantitativo da consola C02 gerado pelo Revit

Aço	Bitola (mm)	Comprimento (m)	Peso (kg)
CA50	8	3,2	1,27
CA50	10	9,6	5,92
CA50	20	6,1	15,04
Total		18,9	22,23

A diferença percentual mais alta registrada foi de **16%** na bitola de 10 mm para a consola C01. Essa variação é considerável e é significativa no contexto de um projeto de construção. Um aumento de 16% no comprimento de uma barra implica em um uso adicional de material, o que pode influenciar diretamente o custo do projeto, o planeamento de aquisições de materiais e a logística de execução. Em projetos onde grandes volumes de aço são utilizados, essa diferença pode resultar em quantidades substancialmente maiores de material, aumentando custos e complexidade de armazenamento. A diferença de **6,9%** no comprimento total da consola C01 gerado pelo Revit em comparação com o CAD, e a diferença de **5%** na consola C02, embora menores do que a variação de 16%, também podem ser significativas dependendo da precisão exigida pelo projeto.

Por outro lado, as diferenças menores, como os **1,4%** observados no peso total do aço para a consola C01 e os **1%** no peso total para a consola C02, podem ser consideradas não significativas no contextos de projeto. Visto que, essas pequenas variações dificilmente impactariam de maneira relevante os custos ou a execução do projeto. Em projetos de larga escala, onde pequenas diferenças podem se acumular, essas variações ainda seriam vistas como dentro da margem de erro

aceitável, especialmente quando se considera a precisão necessária nas estimativas de peso total para planejamento e orçamento.

4.4 Diagramas de montagem

A partir das armaduras geradas pelo script em Dynamo, foram criados diagramas de montagem das consolas curtas C01 e C02, conforme indicado na Figuras 4.15 e 4.16. Esses diagramas foram desenvolvidos para facilitar a compreensão e identificação da sequência correta de montagem pelos montadores, pois muitos profissionais têm dificuldade em visualizar a geometria tridimensional das estruturas utilizando apenas desenhos de plantas e seções bidimensionais.

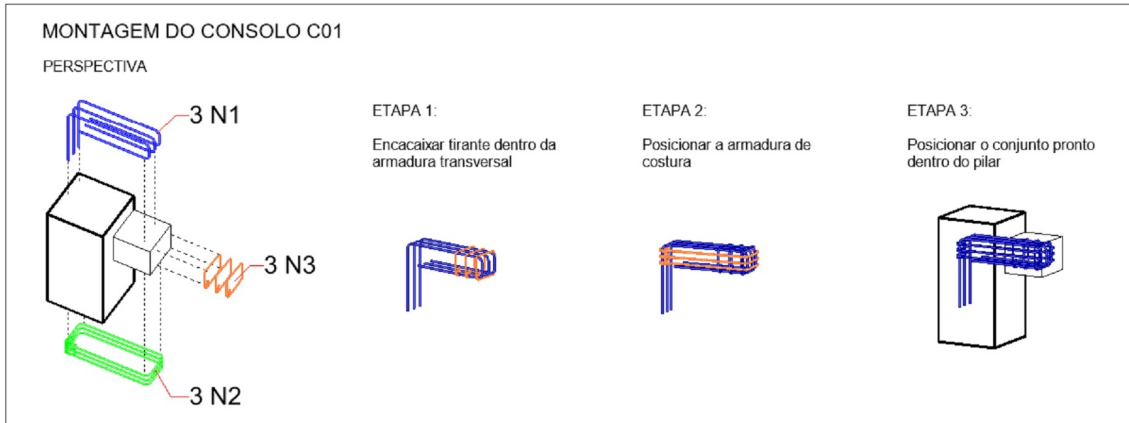


Figura 4.15: Diagrama de montagem da consola C01.

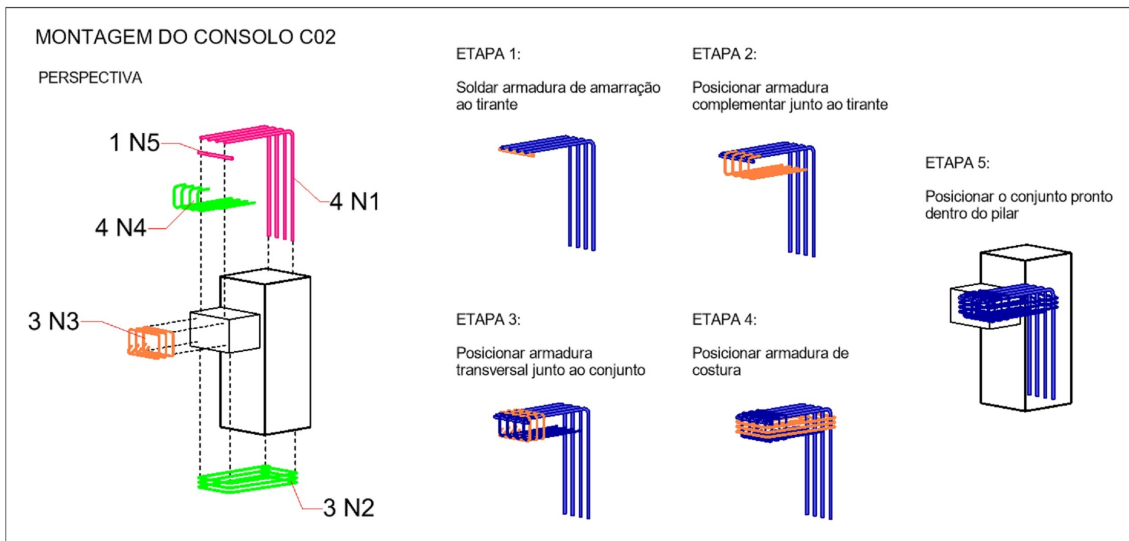


Figura 4.16: Diagrama de montagem da consola C02.

Nas vistas em perspectiva das consolas, cada diâmetro de barra foi representado por uma cor distinta, com o objetivo de fornecer ao montador uma ferramenta prática para verificar se as armaduras estão posicionadas conforme especificado no projeto. As cores utilizadas para representar os diferentes diâmetros são as seguintes:

- **Rosa:** Barras com diâmetro de 20 mm.
- **Verde:** Barras com diâmetro de 10 mm.
- **Azul:** Barras com diâmetro de 12,5 mm.

- **Laranja:** Barras com diâmetro de 8 mm.

Além disso, os diagramas apresentam a sequência de montagem dividida em etapas. Essa abordagem foi desenvolvida para auxiliar no correto posicionamento das armaduras, minimizando assim possíveis erros durante a execução do projeto.

5 CONCLUSÃO

Primeiramente, foi realizada uma revisão de literatura que estabeleceu as bases teóricas do trabalho. O conceito de Modelagem de Informação da Construção (BIM) foi explorado, com ênfase em suas definições, benefícios e aplicações práticas no contexto de projetos de engenharia. Além disso, foram discutidos aspectos como a interoperabilidade entre softwares e os níveis de desenvolvimento no BIM. Sendo, também introduzidas as ferramentas Revit e Dynamo. Além disso, foi feita uma análise comparativa destacando as diferenças entre o Eurocódigo 2 e as normas NBR 6118 e NBR 9062, de modo a justificar a utilização das normas brasileiras, uma vez que a competência para aplicar a norma portuguesa ou a norma brasileira é a mesma.

Em seguida, foi apresentada a metodologia desenvolvida para a automação do detalhamento de consolas curtas. Esse processo envolveu o desenvolvimento de um script no Dynamo, integrado ao Revit, que possibilitou a criação automatizada de armaduras estruturais. A metodologia incluiu a importação de dados, seleção de elementos e criação de geometrias das armaduras, visando precisão e eficiência na geração de quantitativos e minimização de erros. Foram também abordadas as dificuldades encontradas e as soluções aplicadas para garantir a consistência e confiabilidade dos resultados obtidos.

A partir desses procedimentos, é possível concluir que a implementação do script desenvolvido em Dynamo, integrado ao Revit, trouxe avanços no detalhamento das armaduras de consolas, promovendo maior precisão e eficiência no processo de modelagem estrutural. A automação desse processo, além de reduzir a probabilidade de erros humanos, também melhorou a consistência dos resultados, o que é fundamental para a qualidade dos projetos estruturais. A migração do detalhamento tradicionalmente realizado em CAD para um ambiente totalmente integrado ao BIM mostrou-se uma estratégia eficaz para aumentar a precisão e padronizar as práticas de modelagem dentro da empresa.

Os resultados obtidos demonstram que o Revit, quando comparado ao CAD, oferece maior precisão no cálculo dos quantitativos de barras de aço, especialmente no que diz respeito ao comprimento das armaduras. Essa precisão, embora tenha gerado variações pequena no peso total estimado do material, pode ser relevante em projetos de grande escala, onde pequenas diferenças podem impactar os custos e o uso de materiais. Dessa forma, a capacidade de prever e controlar essas variações contribui diretamente para a otimização dos recursos e a eficiência econômica dos projetos.

Além disso, a integração do Dynamo com o Revit facilita a implementação completa do BIM na empresa. Sendo que, a automatização do detalhamento e a precisão oferecida pela metodologia adotada permitem que a equipe de engenharia foque em aspectos mais complexos dos projetos, aumentando a produtividade e a qualidade geral dos trabalhos. Essa transição para um ambiente totalmente digital também proporciona novas possibilidades para a análise e o gerenciamento de projetos.

5.1 Sugestões para trabalhos futuros

- Ampliar a abrangência do script desenvolvido: Da forma que está o script detalha apenas consolas retangulares simples, é interessante ampliar os casos de abrangência de modo a detalhar outros casos de consolas, como: consolas trapezoidais e consolas duplas (retangulares e trapezoidais);
- Aprimorar a automatização de detalhamentos com o Dynamo: Explorar o desenvolvimento de scripts adicionais no Dynamo para automatizar outras etapas do processo de detalhamento estrutural no Revit. Isso incluiria, outros elementos estruturais, como vigas e pilares;
- Estudos de Custo-Benefício em Grandes Escalas: Dado que as diferenças de quantitativos geradas pelo Dynamo/Revit foram consideradas pequenas neste estudo, uma análise aprofundada em projetos de maior escala poderia investigar como essas pequenas diferenças se acumulam e seu impacto financeiro. Isso seria útil para validar ou ajustar as práticas de automação conforme a complexidade e o tamanho dos projetos aumentam.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] European Committee for Standardization. *EN 1992-1-1: Eurocode 2: Design of Concrete Structures - Part 1-1: General rules and rules for buildings*. EN 1992-1-1:2004. Brussels, 2010.
- [2] Associação Brasileira de Normas Técnicas (ABNT). *ABNT NBR 6118:2023 - Projeto de Estruturas de Concreto: Procedimento*. Rio de Janeiro, 2023.
- [3] Associação Brasileira de Normas Técnicas (ABNT). *ABNT NBR 9062:2017 - Projeto e Execução de Estruturas de Concreto Pré-moldado*. Rio de Janeiro, 2017.
- [4] Charles M Eastman. *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons, 2011.
- [5] M Venâncio. «Avaliação da Implementação de BIM em Portugal». Dissertação de Mestrado. Faculdade de Engenharia da Universidade do Porto, 2015.
- [6] Ryan Stanley e Derek Thurnell. «The benefits of, and barriers to, implementation of 5D BIM for quantity surveying in New Zealand». Em: *Australasian Journal of Construction Economics and Building*, The 14.1 (2014), pp. 105–117.
- [7] Silvio Melhado e Ana Carolina Pinto. «Benefícios e desafios da utilização do BIM para extração de quantitativos». Em: *IX Simpósio Brasileiro de Gestão e Economia da Construção (SIBRAGEC) e VI Encuentro Latinoamericano de Gestión y Economía de la Construcción (ELAGEC)* (2015), pp. 511–518.
- [8] António Grilo e Ricardo Jardim-Goncalves. «Value proposition on interoperability of BIM and collaborative working environments». Em: *Automation in construction* 19.5 (2010), pp. 522–530.
- [9] Sijie Zhang et al. «Building information modeling (BIM) and safety: Automatic safety checking of construction models and schedules». Em: *Automation in construction* 29 (2013), pp. 183–195.
- [10] Rafiu Dimeji Seidu et al. «Innovative changes in quantity surveying practice through BIM, big data, artificial intelligence and machine learning». Em: *Applied Science University Journal of Natural Science*. 4.1 (2020), pp. 37–47.
- [11] *Level of Development Specification 2018 Part I*. Disponível em: <https://bimforum.org/1od/>. Acesso em: 11 ago. 2024. BIM Forum. 2018.
- [12] American Institute of Architects. *AIA Document G202-2013: Project Building Information Modeling Protocol Form*. Ed. por American Institute of Architects. American Institute of Architects. Washington, D.C., 2013.
- [13] Mohammed Mekawy e FRANK Petzold. «BIM-based model checking in the early design phases of precast concrete structures». Em: *Learning, Prototyping and Adapting-Proceedings of the 23rd International Conference on Computer-Aided Architectural Design Research in Asia*. 2018, pp. 71–80.
- [14] SPBIM. *Normas de BIM*. SPBIM - ARQUITETURA DIGITAL. 14 de dez. de 2023. URL: <https://spbim.com.br/normas-de-bim/> (acedido em 21/05/2024).
- [15] Ibercard. *Padrões e Normas BIM em Portugal - Ibercad, Lda - Ibercad, software cad*. 2024. URL: <https://www.ibercad.pt/padroes-e-normas-bim-em-portugal.html> (acedido em 21/05/2024).
- [16] Comissão Técnica de Normalização BIM Nacional. *Regras de modelação de objetos BIM*. 2021. URL: <https://www.eeagrants.gov.pt/media/5775/actividade-a4-relatorio-regras-de-objetos.pdf> (acedido em 21/05/2024).
- [17] Jorge Miguel Santos Silva. «Princípios para o desenvolvimento de projetos com recurso a ferramentas BIM». Em: *Repositório aberto da Universidade do Porto* (2013).
- [18] Autodesk. *Ajuda: Conceitos básicos do Revit*. 2024. URL: <https://help.autodesk.com/view/RVT/2023/PTB/?guid=GUID-03565843-BB48-4707-B54C-39D6E8E51880> (acedido em 23/05/2024).

- [19] Tayna Teixeira Vieira e Karoline Vieira Figueiredo. «Vantagens de planejar uma obra com a plataforma BIM, REVIT.» Em: *Boletim do Gerenciamento* 17.17 (2020), pp. 10–19.
- [20] Autodesk. *Ajuda: Sobre as famílias*. 2024. URL: <https://help.autodesk.com/view/RVT/2024/PTB/?guid=GUID-6DDC1D52-E847-4835-8F9A-466531E5FD29> (acedido em 23/05/2024).
- [21] Mateus Beraldi. *13 dicas para dominar as Famílias do Revit*. Estúdio BIM - Engenharia Digital BIM. 2 de mai. de 2022. URL: <https://estudiobim.com.br/13-dicas-dominar-familias-revit/> (acedido em 23/05/2024).
- [22] Luis André Santos. *Revit: Família, Tipo e Instância*. Section: REVIT. QualifiCAD. 14 de mai. de 2018. URL: <https://qualificad.com.br/revit-familia-tipo-e-instancia/> (acedido em 23/05/2024).
- [23] Luis André Santos. *Criar grupos no Revit*. Section: REVIT. QualifiCAD. 6 de ago. de 2018. URL: <https://qualificad.com.br/criar-grupos-no-revit/> (acedido em 26/05/2024).
- [24] Autodesk. *Ajuda: Grupos de elementos*. 2024. URL: <https://help.autodesk.com/view/RVT/2022/PTB/?guid=GUID-52612B0F-43AA-47AF-A76C-BB0E3DD24E34> (acedido em 26/05/2024).
- [25] Autodesk. *Ajuda: Sobre as montagens*. 2024. URL: <https://help.autodesk.com/view/RVT/2024/PTB/?guid=GUID-EBB5B040-BEA8-4867-B245-8C1308EE3AE4> (acedido em 26/05/2024).
- [26] Luis André Santos. *Tabelas, dados e parâmetros no Revit - Qualificad*. Section: REVIT. QualifiCAD. 13 de mai. de 2019. URL: <https://qualificad.com.br/tabelas-dados-e-parametros-no-revit/> (acedido em 28/05/2024).
- [27] *Como a padronização das famílias Revit pode agilizar e gerenciar projetos complexos em larga escala no revit?* Martins de Barros Aquitetura. 26 de mar. de 2023. URL: <https://mdbarquitetura.com.br/wppress/2023/03/26/como-a-padronizacao-das-familias-revit-pode-agilizar-e-gerenciar-projetos-complexos-em-larga-escala-no-revit/> (acedido em 28/05/2024).
- [28] KM Kensek. «Integration of Environmental Sensors with BIM: case studies using Arduino, Dynamo, and the Revit API». Em: (2014).
- [29] Zach Kron. *Q&A about Dynamo*. Dynamo BIM. 3 de mai. de 2014. URL: <https://dynamobim.org/qa-about-dynamo/> (acedido em 09/08/2024).
- [30] Felipe Bueno Brandão Simões. «Automação da Pormenorização de Armaduras em Bim e Modelação Numérica do Emboquilhamento de um Túnel». Dissertação de Mestrado. Instituto Politecnico do Porto (Portugal), 2020.
- [31] Fábio Louzada Colodette. «Desenvolvendo um aplicativo móvel de assistência à leitura para deficientes visuais através de visão computacional e OCR». Em: (2023).
- [32] Mounir Khalil El Debs. *Concreto pré-moldado: fundamentos e aplicações*. Oficina de Textos, 2000.
- [33] Matheus Barbosa Moreira Cedrim et al. «Aplicação do método das bielas para o dimensionamento de consolos curtos com suporte da otimização topológica». Em: (2019).
- [34] Arnold Van Acker. «Manual de sistemas pré-fabricados de concreto». Em: *Traduzido por Marcelo de Araújo Ferreira*. São Paulo, SP: Associação Brasileira da Construção Industrializada de Concreto (2002).
- [35] Vinícius Carvalho Santos. «Análise de consolos de concreto armado à luz dos códigos de projeto». Em: (2022).
- [36] Ladislav Bohumer Kriz e Charles H Raths. «Connections in precast concrete structures: strength of corbels». Em: *Portland Cement Association, Research and Development Laboratories Skokie ...* (1965).
- [37] Fritz Leonhardt e Eduard Mönnig. «Construções de concreto: casos especiais de dimensionamento de estruturas de concreto armado». Em: *Rio de Janeiro, Interciências* (1978).
- [38] Carlos Daniel da Silva Costa. «Aplicação do modelo escoras e tirantes a vigas parede de acordo com o Eurocódigo 2: 2010». Dissertação de Mestrado. Instituto Politécnico do Porto, 2016.

- [39] Gabriel Costa da Silva e Silva e Maria Socorro Martins Sampaio. «Dimensionamento de Consoles em Concreto Armado: Estudo de caso na Ponte do Educandos». Em: *XIV Congresso Brasileiro de Pontos e Estruturas* (2023). URL: <https://www.studocu.com/pt-br/document/universidade-federal-de-minas-gerais/patologia-das-edificacoes/id-008-dimensionamento-de-console-em-concreto-armado-estudo-de-caso-na-ponte-do-educandos/79584134> (acedido em 27/06/2024).
- [40] Rui Vaz Rodrigues e Daniela Gutstein. «Estudo comparativo do dimensionamento de betão de acordo com Eurocódigo 2 e a ABNT NBR6118:2014». Em: 2 de nov. de 2016.