

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA CIVIL



USO DE PYTHON PARA LA AUTOMATIZACIÓN DEL DISEÑO DE ELEMENTOS
ESTRUCTURALES EN REVIT MEDIANTE DYNAMO.

AUTORES:

JOSE REMIGIO VELEZ BRAVO

CARLOS GUILLERMO ANDRANGO GUERRA

DIRECTOR:

CARLOS ANDRES CELI SANCHEZ

Quito DM, julio DE 2025

Tabla de Contenidos

Tabla de Contenidos	2
Índice de Figuras.....	7
CAPÍTULO I	14
1. INTRODUCCIÓN	14
1.1. Planteamiento del problema.....	14
1.2. Justificación e importancia.....	15
1.3. OBJETIVOS	16
1.3.1. Objetivo general.	16
1.3.2. Objetivos específicos.	16
1.4. ALCANCE	17
CAPÍTULO II	18
2. MARCO TEÓRICO	18
2.1. Propiedades de los materiales estructurales.....	18
2.2. Diseño sismorresistente: principios generales.....	19
2.3. Predimensionamiento estructural.....	19
2.4. Análisis estructural sismorresistente según la NEC 2015 y NEC 2024 (Borrador).....	20
2.5. Asignación de cargas gravitacionales.....	20
2.6. Método de análisis: análisis modal espectral.....	22
2.6.1. Factores de suelo establecidos por la NEC (2024) Borrador.....	22
2.6.2. Importancia sísmica, castigos de planta y elevación.....	24
2.6.3. Periodo aproximado de respuesta.....	25
2.7. Cortante basal.....	27
2.8. Área cooperante.....	28
2.9. Diagrama de interacción.....	28
2.10. Equivalencia de cargas.....	28
2.11. Cálculo de momentos según tabla 6.5.2 ACI.....	29
2.12. Cálculo de espesor de Losa según Tabla ACI 8.3.3.1	29
2.13. Combinaciones de carga según ACI 318-19.....	30
2.14. Componente vertical de sismo según la ASCE 7-16.....	30
2.15. Cortante mínimo según la ASCE 7-16.....	31
2.16. Derivas inelásticas.....	31

2.17.	Diafragmas.....	32
2.18.	Nodos Rígidos.	33
2.19.	Elementos finitos.....	33
2.20.	Software comercial de análisis estructural.	33
2.21.	Modelado estructural en software comercial.....	33
2.22.	Diseño por capacidad.	34
2.23.	Diseño a flexión.	34
2.24.	Python.....	35
2.25.	Librerías de Python.....	35
2.25.1.	Librería: import compytes.client.....	35
2.25.2.	Librería: import numpy as np.	36
2.25.3.	Librería: api_csi as api.....	36
2.25.4.	Librería: matplotlib.pyplot as plt.	36
2.25.5.	Librería: Python-docx.....	36
2.26.	Revit.....	36
2.27.	Dynamo	37
2.27.1.	Nodos de programación.	37
2.27.2.	Nodo number slider.	37
2.27.3.	Nodo Code block.....	38
2.27.4.	Nodos: Lists.....	38
2.27.5.	Nodos: Topology.....	38
2.27.6.	Nodos: Geometry.....	38
2.27.7.	Nodos: REVIT.....	38
2.27.8.	Nodos: Structural Design.	39
CAPÍTULO III.....		39
3.	PRE-DISEÑO DE LA ESTRUCTURA DE HORMIGÓN ARMADO.	39
3.1.	Predimensionamiento.	39
3.2.	Definición de estructura propuesta a análisis.	40
3.3.	Ubicación del proyecto.....	42
3.4.	Características del proyecto.	42
3.5.	Metodología de análisis y normativa utilizada.....	43
3.6.	Espectro de respuesta (UHS).	43
3.7.	Parámetros para el espectro de respuesta y determinación de factores f_a , f_d , f_s y z	44

3.7.1.	Clasificación del suelo.....	44
3.7.2.	Determinación de factor z.	44
3.7.3.	Determinación de zona sísmica.	45
3.7.4.	Determinación de factor Fa.	45
3.7.5.	Determinación de factor Fd.	45
3.7.6.	Determinación de factor Fs.	46
3.8.	Parámetros para el espectro de respuesta: importancia sísmica, castigos y factor R.	46
3.8.1.	Importancia sísmica.	46
3.8.2.	Factor de reducción sísmica R.	47
3.8.3.	Irregularidades estructurales horizontales y verticales.....	47
3.9.	Periodo fundamental aproximado.	48
3.10.	Obtención grafica del espectro de respuesta (UHS) y determinación de SAE.	49
3.11.	Cálculo de cargas no sísmicas.....	50
3.11.1.	Sobrecarga	50
3.11.2.	Carga Viva	50
3.11.3.	Peso Propio (PP).....	51
3.11.4.	Losa	51
3.12.	Predimensionamiento de viga.....	57
3.13.	Predimensionamiento de columna.	60
3.13.1.	Cálculo de carga axial para columna.....	60
3.13.2.	Cálculo de momento para columna.....	61
3.13.3.	Distribución de cortante basal.....	62
3.13.4.	Diagramas de interacción y secciones finales de columna.....	65
3.14.	Resumen de las dimensiones de elementos estructurales.	67
CAPÍTULO IV.....		68
4.	ANÁLISIS DE LA ESTRUCTURA EN SOFTWARE COMERCIAL.	68
4.1.	Modelado de la estructura.	68
4.1.1.	Grillas	68
4.1.2.	Niveles.....	69
4.1.3.	Dibujo Paramétrico	69
4.2.	Definición de materiales y secciones.....	70
4.2.1.	Materiales de diseño.	70
4.2.2.	Ingreso de secciones.....	73

4.3.	Aplicación de inercias efectivas en elementos.....	76
4.4.	Definición de cargas y espectros de respuesta.....	79
4.4.1.	Definición en Load Patterns.....	79
4.4.2.	Coeficientes basales de S_x y S_y estáticas.....	79
4.4.3.	Carga de espectros NEC-2024 y ASCE 7-10.....	80
4.5.	Meshing.....	82
4.6.	Combinaciones de carga.....	83
4.6.1.	Calculo E_v (componente vertical del sismo).....	84
4.6.2.	Combinaciones a ETABS.....	85
4.7.	Primeros resultados.....	86
4.8.	Cortante mínima del ASCE.....	87
4.9.	Resultados igualados.....	88
4.10.	Derivas inelásticas.....	89
4.11.	Desplazamientos máximos.....	89
4.12.	Corte.....	90
4.13.	Fuerzas Laterales.....	90
4.14.	Volcamiento.....	91
4.15.	Rigidez por piso.....	91
CAPÍTULO V.....		92
5.	AUTOMATIZACIÓN DEL DISEÑO ESTRUCTURAL DE VIGAS Y COLUMNAS.....	92
5.1.	Instalación de librerías requeridas para Python.....	92
5.2.	Datos requeridos de ingreso al Python.....	92
5.3.	Extracción de datos desde el software comercial.....	93
5.4.	Diseño longitudinal de vigas.....	94
5.4.1.	Agrupación de datos.....	94
5.4.2.	Filtrado de datos.....	96
5.4.2.1.	Áreas de acero longitudinales.....	97
5.4.3.	Acero de refuerzo para momento negativo.....	99
5.4.4.	Longitud de vigas.....	100
5.4.5.	Longitud de desarrollo de aceros de refuerzo para momento negativo.....	101
5.4.6.	Longitud extra de acero de refuerzo.....	104
5.4.7.	Longitud de acero para momentos negativos total.....	105
5.4.8.	Comprobación de áreas de acero en vigas.....	105

5.5.	Diseño transversal de vigas.	107
5.5.1.	Agrupación de datos.	107
5.5.2.	Cortante sísmico.	108
5.5.3.	Cortante provisto por el concreto.	110
5.5.4.	Espaciamiento entre estribos en zona de confinamiento y en zona media.	111
5.5.5.	Cortante provisto por el acero.	112
5.5.6.	Condición de control.	114
5.5.7.	Acero mínimo transversal.	115
5.5.8.	Cortante último.	116
5.5.9.	Cortante requerido de acero.	117
5.5.10.	Área mínima de acero requerida.	118
5.5.11.	Creación del reporte.	119
5.6.	Diseño longitudinal de columnas.	120
5.6.1.	Agrupación de datos.	120
5.6.2.	Filtrado de datos.	121
5.6.3.	Diseño longitudinal.	122
5.7.	Diseño transversal de columnas.	124
5.7.1.	Agrupación de datos.	124
5.7.2.	Filtrado de datos.	125
5.7.3.	Longitud de zona de confinamiento.	126
5.7.4.	Espaciamiento entre estribos en zona de confinamiento.	127
5.7.5.	Espaciamiento entre estribos en zona media.	128
5.7.6.	Ecuaciones para reforzar por confinamiento.	129
5.7.7.	Creación del reporte.	131
CAPITULO VI.		133
6.	Código visual de en Dynamo.	133
6.1.	Versión de Revit.	133
6.2.	Dynamo Player.	133
6.3.	Código visual para columnas.	134
6.3.1.	Selección de elemento columna.	134
6.3.2.	Creación de estribos columna	135
6.3.3.	Descripción de bloque de nodos para estribos columna.	135
6.3.4.	Creación de aceros longitudinales columna.	138

6.3.5.	Descripción de bloque de nodos para aceros longitudinales columna.	138
6.4.	Código visual para vigas.....	141
6.4.1.	Creación de estribos vigas	141
6.4.2.	Selección de elemento viga	141
6.4.3.	Descripción bloque de nodos para estribos de vigas	142
6.4.4.	Bloque de nodos para refuerzo positivo.....	146
6.4.5.	Bloque de Nodos para refuerzo negativo	148
6.4.6.	Descripción de bloque de nodos para acero negativo Vigas.	148
6.4.7.	Bloque de Nodos para Longitudes de desarrollo.....	149
6.4.8.	Bloque de nodos para ganchos.....	151
6.4.9.	Resultado total de scripts viga y columna.	152
6.5.	Reproductor Dynamo Player.	153
6.6.	Manual de uso Dynamo Player scripts.	153
6.6.1.	Descarga de archivo Dynamo. dyn	153
6.6.2.	Abrir el archivo Dynamo en REVIT 2024.	153
6.6.3.	Uso de Dynamo Player para vigas.....	159
6.6.4.	Manual de uso en Dynamo Player Columnas	163
CAPÍTULO VII		166
7.	CONCLUSIONES Y RECOMENDACIONES.	166
7.1.	Conclusiones.....	166
7.2.	Recomendaciones.....	168
BIBLIOGRAFÍA		169

Índice de Figuras

Figura 2.6-1	22
Figura 2.6-2.....	23
Figura 2.6-3.....	23
Figura 2.6-4.....	24
Figura 2.6-5.....	24

Figura 2.6-6.....	25
Figura 2.6-7.....	25
Figura 2.6-8.....	25
Figura 2.6-9.....	27
Figura 2.13-1.....	30
Figura 2.16-1.....	32
Figura 3.2-1.....	40
Figura 3.2-2.....	41
Figura 3.2-3.....	41
Figura 3.3-1.....	42
Figura 3.6-1.....	43
Figura 3.7-1.....	44
Figura 3.7-2.....	44
Figura 3.7-3.....	45
Figura 3.7-4.....	45
Figura 3.7-5.....	46
Figura 3.7-6.....	46
Figura 3.8-1.....	47
Figura 3.8-2.....	47
Figura 3.8-3.....	47
Figura 3.9-1.....	49
Figura 3.10-1.....	49
Figura 3.11-1.....	50
Figura 3.11-2.....	50
Figura 3.11-3.....	51
Figura 3.11-4.....	52
Figura 3.11-5.....	53
Figura 3.11-6.....	53
Figura 3.11-7.....	54
Figura 3.11-8.....	54
Figura 3.11-9.....	55

Figura 3.11-10.....	55
Figura 3.11-11.....	56
Figura 3.11-12.....	57
Figura 3.12-1.....	57
Figura 3.12-2.....	60
Figura 3.13-1.....	60
Figura 3.13-2.....	63
Figura 3.13-3.....	64
Figura 3.13-4.....	65
Figura 3.13-5.....	65
Figura 3.13-6.....	66
Figura 3.13-7.....	66
Figura 3.13-8.....	67
Figura 3.14-1.....	68
Figura 4.1-1.....	68
Figura 4.1-2.....	69
Figura 4.1-3.....	69
Figura 4.1-4.....	69
Figura 4.2-1.....	70
Figura 4.2-2.....	71
Figura 4.2-3.....	71
Figura 4.2-4.....	72
Figura 4.2-5.....	73
Figura 4.2-6.....	74
Figura 4.2-7.....	75
Figura 4.2-8.....	76
Figura 4.3-1.....	77
Figura 4.3-2.....	78
Figura 4.4-1.....	79
Figura 4.4-2.....	80
Figura 4.4-3.....	80

Figura 4.4-4.....	81
Figura 4.4-5.....	82
Figura 4.5-1.....	83
Figura 4.6-1.....	84
Figura 4.6-2.....	86
Figura 4.7-1.....	86
Figura 4.7-2.....	87
Figura 4.9-1.....	88
Figura 4.9-2.....	88
Figura 4.10-1.....	89
Figura 4.11-1.....	89
Figura 4.12-1.....	90
Figura 4.13-1.....	90
Figura 4.14-1.....	91
Figura 4.15-1.....	91
Figura 5.2-1.....	93
Figura 5.3-1.....	94
Figura 5.3-2.....	94
Figura 5.4-1.....	95
Figura 5.4-2.....	95
Figura 5.4-3.....	96
Figura 5.4-4.....	97
Figura 5.4-5.....	98
Figura 5.4-6.....	98
Figura 5.4-7.....	99
Figura 5.4-8.....	100
Figura 5.4-9.....	101
Figura 5.4-10.....	101
Figura 5.4-11.....	103
Figura 5.4-12.....	103
Figura 5.4-13.....	104

Figura 5.4-14.....	105
Figura 5.4-15.....	105
Figura 5.4-16.....	106
Figura 5.4-17.....	107
Figura 5.5-1.....	108
Figura 5.5-2.....	108
Figura 5.5-3.....	109
Figura 5.5-4.....	110
Figura 5.5-5.....	111
Figura 5.5-6.....	111
Figura 5.5-7.....	112
Figura 5.5-8.....	113
Figura 5.5-9.....	113
Figura 5.5-10.....	114
Figura 5.5-11.....	114
Figura 5.5-12.....	115
Figura 5.5-13.....	116
Figura 5.5-14.....	116
Figura 5.5-15.....	117
Figura 5.5-16.....	117
Figura 5.5-17.....	117
Figura 5.5-18.....	118
Figura 5.5-19.....	118
Figura 5.5-20.....	119
Figura 5.5-21.....	119
Figura 5.6-1.....	120
Figura 5.6-2.....	121
Figura 5.6-3.....	121
Figura 5.6-4.....	122
Figura 5.6-5.....	123
Figura 5.6-6.....	124

Figura 5.7-1	124
Figura 5.7-2	125
Figura 5.7-3	126
Figura 5.7-4	126
Figura 5.7-5	127
Figura 5.7-6	127
Figura 5.7-7	128
Figura 5.7-8	128
Figura 5.7-9	129
Figura 5.7-10	129
Figura 5.7-11	130
Figura 5.7-12	131
Figura 5.7-13	131
Figura 5.7-14	131
Figura 5.7-15	132
Figura 5.7-16	132
Figura 6.3-1	134
Figura 6.3-2	134
Figura 6.3-3	135
Figura 6.3-4	136
Figura 6.3-5	136
Figura 6.3-6	137
Figura 6.3-7	137
Figura 6.3-8	138
Figura 6.3-9	138
Figura 6.3-10	139
Figura 6.3-11	140
Figura 6.3-12	140
Figura 6.4-1	141
Figura 6.4-2	142
Figura 6.4-3	142

Figura 6.4-4.....	143
Figura 6.4-5.....	144
Figura 6.4-6.....	145
Figura 6.4-7.....	146
Figura 6.4-8.....	146
Figura 6.4-9.....	147
Figura 6.4-10.....	147
Figura 6.4-11.....	148
Figura 6.4-12.....	148
Figura 6.4-13.....	149
Figura 6.4-14.....	150
Figura 6.4-15.....	151
Figura 6.4-16.....	152
Figura 6.4-17.....	152
Figura 6.6-1.....	153
Figura 6.6-2.....	153
Figura 6.6-3.....	154
Figura 6.6-4.....	154
Figura 6.6-5.....	155
Figura 6.6-6.....	155
Figura 6.6-7.....	156
Figura 6.6-8.....	157
Figura 6.6-9.....	157
Figura 6.6-10.....	¡Error! Marcador no definido.
Figura 6.6-11.....	159
Figura 6.6-12.....	159
Figura 6.6-13.....	160
Figura 6.6-14.....	160
Figura 6.6-15.....	161
Figura 6.6-16.....	161
Figura 6.6-17.....	161

Figura 6.6-18.....	162
Figura 6.6-19.....	163
Figura 6.6-20.....	164
Figura 6.6-21.....	164
Figura 6.6-22.....	165

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Planteamiento del problema.

La problemática del diseño estructural se da principalmente por causa del dibujante, ya que muchas veces no cumple con la responsabilidad de facilitar la interpretación de planos en obra tanto para el ingeniero residente como para los obreros, por lo tanto es importante hoy en día que en obra se pueda tener acceso a un modelo 3D que permita un entendimiento claro de factores clave dentro del diseño estructural, como espaciamiento entre estribos, detalles de conexión viga columna, longitudes de desarrollo, traslapes y configuración de acero longitudinal. La ejecución correcta de estos parámetros es fundamental para la protección de la vida humana y la prolongación de vida útil de la obra, lo cual impacta directamente con el precio de ejecución del proyecto.

Para el dibujo estructural se requiere cumplir normativas locales de donde se desarrolla el proyecto, en el Ecuador la normativa vigente para la construcción de estructuras de hormigón armado,(Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015c). La cual detalla los lineamientos a seguir para cumplir los estándares mínimos de diseño de elementos estructurales en el Ecuador.

El tiempo empleado en detallar y cumplir con estos lineamientos depende totalmente de una persona más no de un software. Lo cual conlleva un uso significativo de tiempo y una mayor probabilidad de error humano. Además, el entendimiento del personal en obra de los planos estructurales dependerá totalmente de la experiencia con la que el dibujante realiza los planos.

En esta investigación se pretende automatizar el modelado estructural en un software 3D direccionado a la metodología BIM, como Revit (Autodesk Inc., 2024d) a través de Dynamo (Autodesk Inc., 2024b) y Python (Python Software Foundation, 2023), cuidando los lineamientos establecidos por la Norma Ecuatoriana de la Construcción, (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b).

1.2. Justificación e importancia.

Actualmente, en el diseño prevalece el uso del software AutoCAD (Autodesk Inc., 2024a), donde según (Gutiérrez, 2022), un plano estructural puede demorar de 2 a 4 semanas en el software de dibujo de Autodesk, el cual es muy limitado para el diseño de elementos en el área de estructuras, ya que no conlleva una fácil implementación en la nueva metodología en auge: BIM (Autodesk Inc., 2024a).

De acuerdo con la Guía de tramites del Gobierno del Ecuador, como requisitos obligatorios para la aprobación de planos de una construcción, se debe presentar un documento digital en formato .dwg que contenga la información solicitada del proyecto a construir (Gobierno del Ecuador, 2021). Esto supone un atraso respecto a metodologías existentes.

Por otra arista, se tiene programas de dibujo actualizados que nos permiten un modelamiento con mucha información y fácil implementación BIM, tales como Revit (Autodesk

Inc., 2024d) y ArchiCAD (Graphisoft, 2024). Revit (Autodesk Inc., 2024d) es el sucesor BIM del software de diseño AutoCAD (Autodesk Inc., 2024a), este es el software BIM más utilizado por los profesionales que trabajan con esta metodología colaborativa.

Según (Zigurat Global Institute of Technology, 2018) el uso generalizado de Revit (Autodesk Inc., 2024d) ha impulsado el desarrollo de un servicio de soporte muy potente, que incluyen herramientas como Navisworks (Autodesk Inc., 2024c), Dynamo (Autodesk Inc., 2024b), BIM 360, así mismo como dice (Econova Institute, 2022). Dynamo (Autodesk Inc., 2024b) es un software que permite, un diseño impulsado por inteligencia artificial que tiene como objetivo facilitar a los arquitectos, diseñadores e ingenieros el diseño de proyectos enfocados en la construcción.

Desde el punto de vista del diseño estructural, la implementación de Dynamo (Autodesk Inc., 2024b) permite un ahorro significativo de tiempo a la hora de dibujar planos estructurales y si a esto se le añade su función de trabajar en conjunto con Python (Python Software Foundation, 2023) se puede desarrollar varios scripts que permitan la automatización de dibujos con una entrada de datos obtenidos con anterioridad por el ingeniero estructural, se debe tomar en cuenta que el diseño se emplea bajo normativas de diseño y construcción locales.

1.3. OBJETIVOS

1.3.1. Objetivo general.

Programar y codificar un script en Python para detallar elementos estructurales en Revit a través de Dynamo.

1.3.2. Objetivos específicos.

- Plantear una estructura regular aporricada de hormigón armado ubicada en el sector, La Mariscal, Quito, Pichincha, Ecuador.
- Realizar el predimensionamiento de columnas y vigas de la estructura propuesta.
- Diseñar y esquematizar los elementos estructurales en un software comercial.
- Definir las cargas (carga viva, carga muerta) según la normativa vigente ecuatoriana (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015c) en un software comercial.
- Realizar un espectro de respuesta (UHS) según (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b), tomando en cuenta los parámetros de suelo de la zona e introducir sus respuestas en aceleración en un software comercial.
- Analizar la estructura propuesta mediante los métodos de análisis modal espectral y fuerza lateral equivalente a través en un software comercial.
- Realizar un diseño de los elementos estructurales con una envolvente de cargas definida por la (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b).
- Obtener las cuantías de acero a flexión y a corte para vigas y columnas a través del diseño de un software comercial según la (American Concrete Institute, 2019).
- Elaborar scripts en Python (Python Software Foundation, 2023) los cuales permitan procesar los resultados arrojados por el software comercial en función de cuantías de acero y los defina en cantidad y geometría de varillas comerciales, respetando el recubrimiento y acero mínimo especificado por normativa de construcción ecuatoriana. (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b).
- Dibujar la estructura sin acero en Revit (Autodesk Inc., 2024d).
- Elaborar programación visual en Dynamo (Autodesk Inc., 2024b) emparejada con scripts de Python para automatizar el detallado estructural dentro de los elementos de hormigón previamente dibujados en Revit (Autodesk Inc., 2024d).
- Evaluar el detallado y verificar el cumplimiento de los requisitos mínimos de diseño sismorresistente según la norma vigente del Ecuador (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015a).

1.4. ALCANCE

Los scripts elaborados en Python (Python Software Foundation, 2023) aplicado en

Dynamo(Autodesk Inc., 2024b), dibujara en Revit (Autodesk Inc., 2024d)el acero estructural dentro de los elementos de hormigón respetando los lineamientos de detalle sísmicos considerados en la NEC.(Ministerio de Desarrollo Urbno y Vivienda (MIDUVI), 2015b).

CAPÍTULO II

2. MARCO TEÓRICO.

2.1. Propiedades de los materiales estructurales.

En el presente trabajo se ha considerado el uso de hormigón armado, compuesto por concreto estructural y acero de refuerzo, como el sistema constructivo principal. Esta elección responde tanto a su alta disponibilidad en Ecuador como a su comportamiento adecuado ante sollicitaciones sísmicas cuando se encuentra completamente detallado.

Para efectos de análisis y diseño, se ha adoptado un concreto y acero con resistencias de:

$$f'c = 240 \frac{kg}{cm^2}$$

$$fy = 4200 \frac{kg}{cm^2}$$

$$\gamma H = 2400 \frac{kg}{m^3}$$

La adecuada interacción entre estos materiales (hormigón y acero), es fundamental para garantizar un diseño dúctil y no frágil.

2.2. Diseño sismorresistente: principios generales.

El diseño sismorresistente tiene como objetivos garantizar la integridad de una estructura a la hora de recibir esfuerzos inducidos por un sismo de diseño y preservar la vida humana, siendo este último el primordial. El enfoque requiere considerar el comportamiento dinámico de la estructura, su ductilidad y su capacidad de disipar energía.

En Ecuador, el diseño sismorresistente está regulado por la Norma Ecuatoriana de la Construcción (NEC) (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b), cuyas versiones 2015 y 2024 (Borrador)(Ministerio de Desarrollo Urbano y Vivienda, 2024) establecen los criterios de análisis, los espectros de respuesta y los factores de reducción que permiten dimensionar correctamente las edificaciones en función de su uso, zona sísmica y tipo de suelo. Estas normas se complementan a su vez con normativas internacionales como los son la ACI 318-19 (American Concrete Institute, 2019) y la ASCE 7-16 (American Society of Civil Engineers, 2017).

2.3. Predimensionamiento estructural.

El predimensionamiento estructural es la primera aproximación técnica para definir las dimensiones básicas de los elementos que componen una edificación. Este proceso no busca aun un diseño detallado, sino establecer proporciones razonables y funcionales que permitan generar un modelo estructural coherente desde el inicio del proyecto.

En el caso de esta tesis, el predimensionamiento se realiza en función de la arquitectura propuesta, las luces estructurales dominantes y los parámetros sísmicos del sitio. A partir de esos datos se calcula el espectro de respuesta (UHS) y se definen las dimensiones iniciales de los elementos estructurales. Estas dimensiones son luego introducidas en el software comercial de

análisis para verificar su desempeño.

2.4. Análisis estructural sismorresistente según la NEC 2015 y NEC 2024 (Borrador).

El análisis sismorresistente es una fase crítica en el diseño estructural, ya que permite evaluar cómo responde una edificación frente a solicitaciones dadas por el sismo de diseño. Este análisis se basa en la determinación de fuerzas laterales, deformaciones, derivas y desplazamientos máximos, y su correcta implementación garantiza que la estructura cumpla con los niveles de seguridad definidos por la NEC 2024 (Borrador) (Ministerio de Desarrollo Urbano y Vivienda, 2024).

En el presente trabajo, el análisis estructural se realiza en función de lo establecido por la Norma Ecuatoriana de la Construcción (NEC)(Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b) en sus versiones 2015 y 2024 (borrador) (Ministerio de Desarrollo Urbano y Vivienda, 2024), las cuales establecen los parámetros sísmicos obligatorios para el territorio ecuatoriano.

2.5. Asignación de cargas gravitacionales.

Antes de cualquier análisis es importante la asignación y determinación de las cargas de sollicitación gravitacionales, que comprenden el peso propio de la estructura que por lo general recae en una estimación a base de experiencia, una sobrecarga calculada o estimada en función de los elementos no estructurales permanentes (acabados, muros de separación y maquinaria permanente), y la carga viva normativa definida según la NEC-SE-CG (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015a), todas estas cargas son necesarias para calcular el peso de la estructura y su masa sísmica, así como la sollicitación gravitacional de los elementos

estructurales.

En el presente trabajo se determina las cargas vivas a través de la tabla 4.2.1 establecida por la NEC-SE-CG (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015a).

4.2 Carga viva: sobrecargas mínimas

4.2.1. Sobrecargas mínimas uniformemente distribuidas L_o y concentradas P_o En la **Tabla 9** se muestran los valores de cargas (uniforme y/o concentrada) de acuerdo con la ocupación o los usos:

Ocupación o Uso	Carga uniforme (kN/m ²)	Carga concentrada (kN)
...

Nota: Tabla tomada de NEC-SE-CG.

De la misma forma los pesos específicos de los materiales para cálculo de sobrecarga se determinan a través de la tabla 4.1 Peso muerto de los materiales establecida por la NEC-SE-CG (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015a).

4. Apéndice: valores de las cargas muertas y vivas

4.1. Carga muerta: pesos de los materiales

En la **Tabla 8** se muestran los valores de los pesos para los materiales de uso más frecuente:

Material	Peso Unitario (kN/m ³)
...	...

Nota: Tabla tomada de NEC-SE-CG.

2.6. Método de análisis: análisis modal espectral.

Para evaluar de manera más precisa la respuesta sísmica se utiliza análisis modal espectral el cual conlleva una consideración de múltiples modos de vibración a través de un espectro de respuesta unificado (UHS o Uniform Hazard Specter) el cual consiste en dar las respuestas máximas en aceleración estadísticas a distintos periodos de vibración.

Para el presente trabajo se utiliza el espectro definido por la normativa NEC 2024 (BORRADOR) (Ministerio de Desarrollo Urbano y Vivienda, 2024), la normativa define el espectro a través de distintas ecuaciones establecidas por los factores de suelo, factor de importancia sísmica y factores de castigo definidos a continuación.

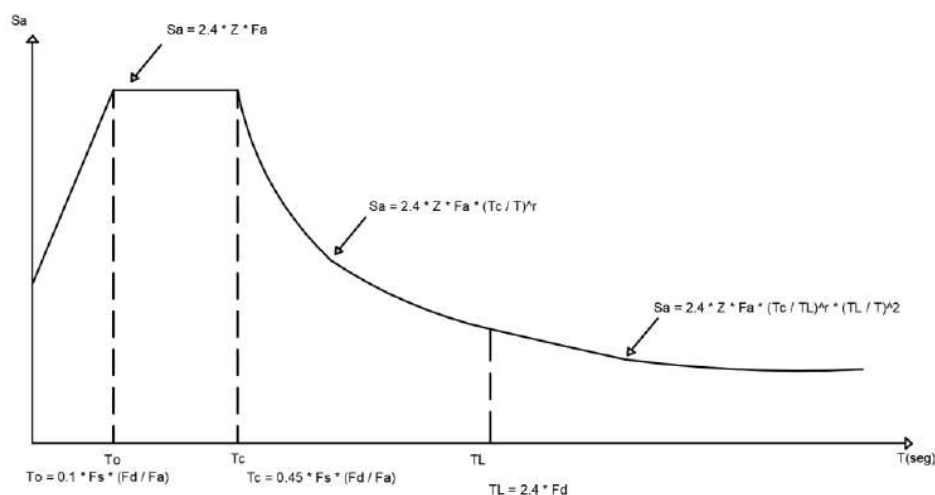


Figura 2.6-1

Nota: imagen creada a partir de NEC-SE-CG.

2.6.1. Factores de suelo establecidos por la NEC (2024) Borrador.

En el presente trabajo se realiza la estimación del perfil del suelo a través de la tabla 3.2 proporcionada por la NEC 2024 (borrador) (Ministerio de Desarrollo Urbano y Vivienda, 2024).

Tabla 3.2. Perfiles de suelo para diseño Sísmico.

Tipo de perfil	Descripción	Definición
...

Nota: Tabla tomada de NEC-SE-DS.

De igual manera los factores de suelo se tomarán de las siguientes tablas especificadas por la NEC 2024 (borrador) (Ministerio de Desarrollo Urbano y Vivienda, 2024).

Tabla 3.1. Zonas Sísmicas (BORRADOR-NEC-2024)

Zona Sísmica	Coefficiente Z (fracción de la gravedad)
I	$0.1 \leq Z < 0.2$
II	$0.2 \leq Z < 0.3$
III	$0.3 \leq Z < 0.4$
IV	$0.4 \leq Z < 0.5$
V	$0.5 \leq Z < 0.65$

Figura 2.6-2

Nota: tabla tomada de NEC-SE-DS.

Tabla 3.3. Coeficiente Fa (BORRADOR-NEC-2024)

Tipo de Perfil de Suelo	I	II	III	IV	V
A	0.9	0.9	0.9	0.9	0.9
B	1	1	1	1	1
C	1.4	1.3	1.23	1.19	1.13
D	1.6	1.4	1.25	1.14	1
E	1.8	1.4	1.1	0.9	0.62

Figura 2.6-3

Nota: tabla tomada de NEC-SE-DS.

Tabla 3.4. Coeficiente Fd (BORRADOR-NEC-2024)

Tipo de Perfil de Suelo	I	II	III	IV	V
A	0.9	0.9	0.9	0.9	0.9
B	1	1	1	1	1
C	1.36	1.28	1.15	1.08	1
D	1.62	1.45	1.28	1.15	1
E	2.1	1.75	1.65	1.52	1.36

Figura 2.6-4

Nota: tabla tomada de NEC-SE-DS.

Tabla 3.5. Coeficiente Fs (BORRADOR-NEC-2024)

Tipo de Perfil de Suelo	I	II	III	IV	V
A	0.75	0.75	0.75	0.75	0.75
B	0.75	0.75	0.75	0.75	0.75
C	0.85	0.94	1.06	1.17	1.28
D	1.02	1.06	1.19	1.32	1.44
E	1.5	1.6	1.8	1.94	2.09

Figura 2.6-5

Nota: tabla tomada de NEC-SE-DS.

2.6.2. Importancia sísmica, castigos de planta y elevación.

La importancia sísmica es un factor de mayoración que depende de la ocupación de la estructura y de su función de diseño, así mismo los castigos de planta y elevación son de necesaria aplicación y se calculan en función de la arquitectura y geometría estructural del proyecto.

En el presente trabajo se calcula estos factores a través de las tablas especificadas proporcionadas por la NEC 2024 (Borrador) (Ministerio de Desarrollo Urbano y Vivienda, 2024).

Tabla 1.3 del Capítulo NEC-SE-CG.

Tipo de Edificación	Ejemplos	Categoría de Riesgo
Todas las edificaciones, excepto aquellas clasificadas con categorías de riesgo I, III y IV	<ul style="list-style-type: none"> Residencias unifamiliares y bifamiliares que no clasifican dentro de NEC-SE-VIVIENDA Edificios de oficinas, apartamentos, centros comerciales y otras estructuras con capacidad menor a 300 personas 	II

Figura 2.6-6

Nota: tabla tomada de NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024).

Tabla C.4. Sección de la Tabla 1.4 del Capítulo NEC-SE-CG

Categoría de Riesgo	Factor de Importancia Sísmica, I_e
I	1
II	1
III	1.25
IV	1.5

Figura 2.6-7

Nota: tabla tomada de NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024).

Tabla 4.4. Sistemas Estructurales Permitidos (BORRADOR-NEC-2024)

	Referencia normativa	R						
Pórticos especiales resistentes a momento en acero	NEC-SE-AC: 5.3, AISC 341-16: E3	8	3	5.5	SL	SL	SL	
Pórticos intermedios resistentes a momento en acero	NEC-SE-AC: 5.2, AISC 341-16: E2	4.5	3	4	SL	12	NP	
Pórticos ordinarios resistentes a momento en acero	NEC-SE-AC: 5.1, AISC 341-16: E1	3.5	3	3	10	NP*	NP	
Pórticos especiales resistentes a momento en hormigón	ACI 318-19: 18.6, 18.7, 18.8	8	3	5.5	SL	SL	SL	
Pórticos con vigas de acero y columna tubular rellena de hormigón*	AISC 341-16: Cap. G, Lai et al. (2020)	3.5	3	3	10	NP*	NP	

Figura 2.6-8

Nota: tabla tomada de NEC-SE-DS.

2.6.3. Periodo aproximado de respuesta.

El periodo fundamental aproximado está definido por una expresión dada por la NEC-2024 (Borrador) (Ministerio de Desarrollo Urbano y Vivienda, 2024), lo cual permite tener una aproximación del periodo de una estructura en función de la altura y los factores C_t y x , esto con el fin de poder obtener una aceleración espectral de predimensionamiento.

En el presente trabajo se utiliza la siguiente expresión para el cálculo del periodo fundamental aproximado:

6.2.1.1. Periodo Fundamental Aproximado, T_a (BORRADOR-NEC-2024) (Ministerio de Desarrollo Urbano y Vivienda, 2024).

$$T_a = C_t * h_n^x$$

Donde:

T_a = es el periodo de vibración aproximado, (s)

C_t = Factor dado por la tabla 6.2

h_n = es la altura de la estructura, (m), dada por la distancia vertical desde la base hasta el nivel más alto del sistema resistente a carga sísmica de la estructura.

X = Factor dado por la tabla 6.2

De igual manera los factores C_t y x se calculan a través de la siguiente tabla

6.2.1.1. Tabla 6.2.

Parámetros C_t y x para el Cálculo del Periodo de Vibración Aproximado (BORRADOR-NEC-2024) (Ministerio de Desarrollo Urbano y Vivienda, 2024).

Tipo de Estructura	Ct	x
Pórticos resistentes a momento de acero ^a	0.0724	0.8
Pórticos resistentes a momento de hormigón ^a	0.0466	0.9
Pórticos arriostrados excéntricamente de acero	0.0731	0.75
Pórticos con arriostramiento con pandeo restringido	0.0731	0.75
Todos los otros sistemas estructurales	0.0488	0.75

Figura 2.6-9

Nota: imagen tomada de NEC-SE-DS.

2.7. Cortante basal.

El cortante basal es la fuerza horizontal sísmica que actúa en la base de una estructura, producto de la inercia generada por el movimiento sísmico sobre la masa del edificio. Representa el esfuerzo total que la cimentación debe resistir y transmitir al suelo durante un sismo, y es una de las principales magnitudes a considerar en el diseño sismorresistente.

“El cortante basal es el punto de partida para distribuir las fuerzas sísmicas entre los diferentes niveles del edificio, y su correcta estimación es esencial para un diseño estructural seguro.”

(NEC-SE-DS 2024, Sección 7.2) (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b).

Para el presente análisis se estima la fuerza de cortante basal con la siguiente expresión establecida por la NEC-2024 (Borrador) (Ministerio de Desarrollo Urbano y Vivienda, 2024).

$$V = \frac{I_e \cdot S_{aT}}{R} \cdot W$$

Nota: expresión tomada de NEC-SE-DS.

2.8. Área cooperante.

El área cooperante es el valor numérico en m² que carga teóricamente un elemento estructural, que se calcula a través de diferentes criterios.

En este trabajo el área cooperante se calcula mediante la suposición que los elementos estructurales cargan la mitad de la luz que comprende la separación entre un elemento y el próximo.

2.9. Diagrama de interacción.

El diagrama de interacción son los valores numéricos de Pu (Carga axial última) y Mu (Momento último) representados gráficamente en un bulbo. El diagrama de interacción es principalmente usado para determinar la capacidad a flexo-compresión de un elemento sometido a esfuerzos tanto axiales como de flexión.

2.10. Equivalencia de cargas.

La equivalencia de cargas es la transformación matemática de un área a otra, con fin de mantener los mismos valores y simplificar el diagrama de cargas en una figura conocida mas sencilla para el cálculo.

En este trabajo se utiliza la siguiente equivalencia de cargas para pasar de una carga trapezoidal a una carga distribuida cuadrada.

$$W_L = \frac{q * s}{3} * \left(\frac{3 - m^2}{2}\right)$$

Siendo:

W_L : Es la carga rectangular equivalente

q: carga por m^2

s: Lado menor

L: Lado mayor

m: Relación entre lado mayor y el lado menor $m = s/L$

2.11. Cálculo de momentos según tabla 6.5.2 ACI.

Para el cálculo de momentos en los vanos respectivos de las vigas la ACI permite el uso de distintas ecuaciones según las condiciones de estas.

Para este trabajo se utiliza la siguiente tabla proporcionada por la ACI, para calcular el momento negativo máximo de la cara exterior del primer apoyo exterior en dos vanos:

$$\frac{w_u \cdot l_n^2}{10}$$

Nota: ecuación tomada de ACI 318-19 Tabla 6.5.2.

2.12. Cálculo de espesor de Losa según Tabla ACI 8.3.3.1

Para el predimensionamiento de losa en el presente trabajo se aprovechó de la fórmula proporcionada por la ACI 318-19 Tabla 8.3.3.1 para espesor mínimo de Losas no preesforzadas en dos direcciones sin vigas interiores (mm) (American Concrete Institute, 2019).

$$\frac{l_n}{33}$$

Nota: ecuación tomada de ACI 318-19 Tabla 8.3.3.1.

2.13. Combinaciones de carga según ACI 318-19.

Las combinaciones de carga son expresiones que agrupan distintas acciones que puede experimentar una estructura, esto incluye cargas muertas, vivas, sísmicas, de viento, etc. Su finalidad es garantizar que el diseño estructural considere la interacción simultánea de distintas sollicitaciones.

Según la ACI 318-19 (American Concrete Institute, 2019) “La resistencia requerida U debe ser por lo menos igual al efecto de las cargas mayoradas de la tabla 5.31, con las excepciones y adiciones de 5.3.3 hasta 5.3.13.”

En este trabajo para el diseño en el software comercial se ingresan las combinaciones de cargas especificadas por la ACI (American Concrete Institute, 2019) en la siguiente tabla:

Tabla 5.3.1- Combinaciones de carga

Combinación de carga	Ecuación	Carga primaria
$U = 1.4D$	(5.3.1a)	D
$U = 1.2D + 1.6L + 0.5(Lr \text{ o } S \text{ o } R)$	(5.3.1b)	L
$U = 1.2D + 1.6(Lr \text{ o } S \text{ o } R) + (1.0L \text{ o } 0.5W)$	(5.3.1c)	Lr o S o R
$U = 1.2D + 1.0W + 1.0L + 0.5(Lr \text{ o } S \text{ o } R)$	(5.3.1d)	W
$U = 1.2D + 1.0E + 1.0L + 0.2S$	(5.3.1e)	E
$U = 0.9D + 1.0W$	(5.3.1f)	W
$U = 0.9D + 1.0E$	(5.3.1g)	E

Figura 2.13-1

Nota: ecuación tomada de ACI 3-18 Tabla 5.3.1. (American Concrete Institute, 2019)

2.14. Componente vertical de sismo según la ASCE 7-16.

La componente vertical de sismo, denotada como E_v , es la parte de la acción sísmica que actúa perpendicular al plano del terreno, es decir es la descomposición del sismo en su componente normal al plano.

En este trabajo se calcula la componente vertical del sismo a través de la expresión 12.4-4 facilitada por la ASCE 7-16 (American Society of Civil Engineers, 2017).

Ecuación 12.4.4a

$$E_v = 0.2 \cdot S_{DS} \cdot D$$

Nota: ecuación tomada de ASCE 7-16 eq 12.4-4a.

2.15. Cortante mínimo según la ASCE 7-16.

El cortante sísmico mínimo es el valor mínimo permitido del cortante basal (V) que debe considerarse en el diseño sismorresistente de una estructura, este valor es proporcionado por la ASCE 7-16 (American Society of Civil Engineers, 2017) en sus ecuaciones pertenecientes a la sección 12.8

En este trabajo se utiliza específicamente 12.8-3

Para T menor o igual a TL

$$C_s = \frac{S_{D1}}{T \left(\frac{R}{I_e} \right)}$$

Nota: ecuación tomada de ASCE 7-16 eq 12.8-3

2.16. Derivas inelásticas.

Las derivas inelásticas representan el desplazamiento relativo de entre piso ya teniendo en cuenta la devolución del factor R multiplicado 0.75 siempre y cuando se asegure ductilidad en el diseño, y estas derivas mantiene un valor máximo en función de porcentaje de deriva de entrepiso, la cual en el borrador de la NEC 2024 (Borrador) (Ministerio de Desarrollo Urbano y Vivienda,

2024) estos valores se encuentran determinados en la tabla 4.3 Limite de la deriva inelástica de piso:

Estructura	Categoría de Riesgo I o II	Categoría de Riesgo III	Categoría de Riesgo IV
Sistemas de resistencia a carga sísmica en edificaciones con paredes interiores flexibles, constituidas por perfiles livianos de acero o madera y recubiertos con paneles de yeso, fibrocemento, o similar.	0.018	0.015	0.012
Sistemas de resistencia a carga sísmica en edificaciones con paredes interiores rígidas de bloque de cemento, hormigón, o de ladrillo de arcilla.	0.015	0.012	0.01
Pórticos resistentes a momento constituidos por vigas de acero y columnas tubulares de acero rellenas de hormigón, con cualquier tipo de divisiones interiores.	0.01	N.P.	N.P.

Figura 2.16-1

Nota: Tabla tomada de NEC 2024 (borrador)

2.17. Diafragmas

En una edificación, como tal los diafragmas son los elementos estructurales que se encargan de distribuir las fuerzas laterales como puede ser el viento o un sismo hacia otros elementos estructurales de la estructura como muros, columnas y pórticos, de aquí se desprenden dos conceptos: Diafragma Rígido y Diafragma flexibles, siendo la principal diferencia que en un diafragma rígido se considera que se distribuyen las fuerzas proporcionalmente a los otros elementos estructurales gracias a la rigidez del diafragma, mientras que en un diafragma flexible se permiten deformaciones significativas y no distribuye las fuerzas uniformemente.

En esta tesis debido a la Rigidez considerada de la losa calculada se considera diafragma rígido.

2.18. Nodos Rígidos.

Un nodo rígido en ingeniería estructural es un punto de unión entre elementos estructurales, es decir, es la conexión de los elementos donde se considera que no hay deformación relativa entre elementos, el correcto funcionamiento de este dependerá del correcto diseño de las conexiones.

2.19. Elementos finitos.

Los elementos finitos son porciones pequeñas y simples en las que se divide una estructura compleja para analizar matemáticamente, esto se suele hacer con elementos estructurales donde se complica el modelado matemático, como por ejemplo losas.

2.20. Software comercial de análisis estructural.

Debido a la complejidad e iteraciones que se requiere a la hora de realizar un análisis estructural se han desarrollado distintos métodos computacionales para agilizar el análisis de estructuras.

En el presente trabajo se utiliza un software estructural para realizar la determinación de distintos parámetros de análisis y diseño, exportar esos datos para su posterior procesamiento y optimizar el diseño proporcionado por el programa.

2.21. Modelado estructural en software comercial.

El modelado estructural constituye una de las etapas fundamentales en el proceso de análisis y diseño sísmico de edificaciones. Su propósito es representar de forma idealizada la geometría, condiciones de soporte, materiales, cargas y comportamiento estructural de una edificación, permitiendo realizar simulaciones numéricas bajo diferentes combinaciones de carga y demandas sísmicas.

En el contexto de esta tesis, se emplea un software comercial de análisis estructural para desarrollar el modelo tridimensional de la edificación. Este entorno permite definir elementos y solicitaciones tales como:

- Niveles o plantas
- Grillas de ejes de dibujo
- Elementos estructurales
- Cargas
- Parámetros de los materiales

Y muchas más condiciones de análisis.

2.22. Diseño por capacidad.

Diseñar por capacidad significa especialmente para columnas, que más allá de resistir los esfuerzos de corte demandados por el diagrama de esfuerzos cortante, se debe diseñar para garantizar el confinamiento del elemento estructural.

En el presente trabajo los elementos estructurales verticales (columnas) se calcula bajo el criterio de diseño de capacidad a través de las siguientes expresiones definidas por la ACI 318-19 (American Concrete Institute, 2019)

2.23. Diseño a flexión.

El diseño a flexión es el proceso mediante el cual se dimensionan los elementos de refuerzo estructural que están encargados de resistir los esfuerzos a flexión generados por el diagrama de

momentos calculado.

En este trabajo el refuerzo a flexión es el acero grado 60, y se calcula usando las siguientes expresiones de la ACI 318-19 (American Concrete Institute, 2019)

2.24. Python

Python(Python Software Foundation, 2023) es un lenguaje de programación de alto nivel, interpretado y de propósito general, ampliamente utilizado en ingeniería, ciencia de datos, automatización y desarrollo de software. Se caracteriza por una sintaxis clara y legible, lo que facilita su aprendizaje y aplicación incluso en entornos complejos como el diseño estructural automatizado.

En esta tesis, Python (Python Software Foundation, 2023)es la base para procesar automáticamente los resultados del análisis estructural, realizar el diseño por flexión y corte, generar hojas de armado, y usar esta información en Dynamo para su visualización en Revit, optimizando así el flujo de trabajo desde el cálculo hasta el modelado. (Autodesk Inc., 2024b)

2.25. Librerías de Python.

Las librerías de Python (Python Software Foundation, 2023) son conjuntos de módulos y funciones predefinidas que extienden la funcionalidad del lenguaje y permiten resolver tareas específicas sin necesidad de programar todo desde cero. Estas librerías contienen código reutilizable que ha sido diseñado para simplificar operaciones complejas.

2.25.1. Librería: import compytes.client.

Permite a Python (Python Software Foundation, 2023) comunicarse con aplicaciones Windows que tienen interfaces COM, en este trabajo se utiliza la librería import compytes.client para tener conectividad directa con el software comercial.

2.25.2. Librería: import numpy as np.

Esta librería matemática en Python (Python Software Foundation, 2023) permite el manejo de arreglos, vectores, matrices y operaciones numéricas de forma rápida y eficiente, esencial para el proceso matemático de datos.

2.25.3. Librería: api_csi as api.

Es una librería personalizada creada para encapsular funciones del software comercial a partir de parámetros de entrada como el nombre de tablas, esta librería permite automatización en la recolección de datos de este software.

2.25.4. Librería: matplotlib.pyplot as plt.

Es una librería que permite generar graficas profesionales en Python (Python Software Foundation, 2023): curvas, espectros, histogramas, etc.

2.25.5. Librería: Python-docx.

Esta es una librería que te permite crear, modificar y dar formato a archivos .docx (Microsoft Word) usando Python, sin necesidad de abrir Word manualmente.

2.26. Revit

Revit (Autodesk Inc., 2024d) es un software de modelado de información para la construcción, (BIM, por sus siglas en inglés) que permite diseñar, documentar y coordinar digitalmente edificaciones desde sus etapas más tempranas. A diferencia de los programas de dibujo en 2D, Revit (Autodesk Inc., 2024d) trabaja con un enfoque tridimensional e inteligente, en el que cada elemento del modelo (como muros, columnas, losas o instalaciones) forma parte de un sistema interconectado. Esto significa que cualquier modificación en el diseño se actualiza automáticamente en todos los planos, vistas y tablas, garantizando coherencia y eficiencia.

En este trabajo para el desarrollo de la programación visual se utiliza Revit 2024. (Autodesk Inc., 2024d)

2.27. Dynamo

Dynamo es una plataforma de programación visual que permite automatizar procesos, generar geometrías complejas y manipular datos dentro de entornos de modelado como Revit (Autodesk Inc., 2024d). A través de un sistema de nodos interconectados entre si – similar a un diagrama de flujo – los usuarios pueden definir las reglas, condiciones y relaciones sin necesidad de escribir código tradicional, aunque también ofrece soporte para lenguajes como Python (Python Software Foundation, 2023) para funciones avanzadas.

Dynamo de por si es complemento interno del Revit a partir del Revit 2021 (Autodesk Inc., 2024d), es decir desde esa versión del software viene incluido de por si Dynamo.

En el presente trabajo para el desarrollo de la programación visual parte de la automatización de diseño se utiliza Revit 2024 y el Dynamo que viene instalado de por si en el programa. (Autodesk Inc., 2024b)

2.27.1. Nodos de programación.

Los nodos de programación representan el corazón de la programación visual, son cajas representadas visualmente que cumplen con funciones específicas programadas, estas cajas se conectan con otras para transmitir información procesada de una a otra y así sucesivamente, es decir, es una manera grafica de unir paquetes de códigos preprogramados para que realicen funciones y tareas específicas en función de la información brindada.

2.27.2. Nodo number slider.

Este nodo contiene un deslizador donde se configuran los rangos de valores y se

seleccionan a través de un desplazamiento de barra, ideal para datos de entrada.

2.27.3. Nodo Code block.

Este nodo permite escribir código en formato Python (Python Software Foundation, 2023), también es usado para definir repeticiones de procesos a través de intervalos deseados, cualquier nodo se puede representar en el code block a través de código escrito.

2.27.4. Nodos: Lists.

Esta familia de nodos entrega varios tipos de listas, las listas son nodos que almacena, agrupa, ordena y extrae información deseada de otros nodos, es decir en estos nodos se pueden concatenar diferentes parámetros de información que se necesite para realizar una tarea que dependa de varios datos proporcionados por diferentes nodos.

2.27.5. Nodos: Topology.

Esta familia de nodos da como resultado parámetros topológicos de alguna estructura hecha en dynamo (Autodesk Inc., 2024b), es decir en esta familia de nodos podemos encontrar aquellos bloques que pueden definir, extraer, modificar y crear topologías.

2.27.6. Nodos: Geometry.

Los nodos Geometry comprenden la familia más grande de nodos en Dynamo (Autodesk Inc., 2024b) teniendo más de 100 bloques de programación visual, los nodos geometry sirven para definir, dibujar, extruir, alterar, crear, duplicar, modificar y unir formas geométricas de todo tipo, Dynamo (Autodesk Inc., 2024b) como tal no puede dibujar con un proceso visual como Revit (Autodesk Inc., 2024d), todos los dibujos se logran a través de unión y extrusiones de puntos, líneas y áreas definidas en un plano cartesiano.

2.27.7. Nodos: REVIT.

La familia de nodos REVIT (Autodesk Inc., 2024d) lo que permite es asociar dynamo

(Autodesk Inc., 2024b) con elementos ya especificados y dibujados en Revit (Autodesk Inc., 2024d), es decir extrae la información deseada del plano de dibujo actual del software, donde se pueden obtener elementos, familias, planos y toda la información de REVIT(Autodesk Inc., 2024d).

2.27.8. Nodos: Structural Design.

Estos nodos son nodos personalizados en un paquete diseñado para el diseño estructural muy puntualmente lo que realiza es dar la propiedad deseada de material (acero estructural) a las formas deseadas ya previamente dibujadas en Dynamo (Autodesk Inc., 2024b), es decir este bloque tiene como entrada líneas y como salida acero de refuerzo que siguen la geometría de las líneas dibujadas.

CAPÍTULO III

3. PRE-DISEÑO DE LA ESTRUCTURA DE HORMIGÓN ARMADO.

3.1. Predimensionamiento.

El predimensionamiento es la etapa preliminar del proceso de análisis donde se define, las dimensiones aproximadas de los elementos estructurales tales como: vigas, columnas, losas, etc. Con base a criterios normativos, experiencia profesional y requerimientos del proyecto. El principal objetivo de esta etapa es evitar la mayor cantidad de iteraciones de análisis posibles y tener un buen punto de partida.

“El predimensionamiento permite establecer la factibilidad estructural del proyecto en

función del uso previsto, tipo de estructura, altura, luces y características del suelo.” (NEC 2015, Introducción al Diseño Estructural) (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015c).

3.2. Definición de estructura propuesta a análisis.

La construcción será 2 pisos con uso residencial ubicada en sector La Mariscal con un método constructivo de hormigón armado, contando con un hormigón de 240 kg/cm^2 como resistencia a la compresión y un acero grado 60 con un F_y de 4200 kg/cm^2 , se contó preliminarmente con 240 m^2 de construcción (120 m^2 por planta), teniendo su luz más larga de 5 metros en el eje Y vista desde planta, primera altura de entre piso cuenta con 4 metros siguiéndole un “piso 2” a 3 metros más arriba.

PLANTA

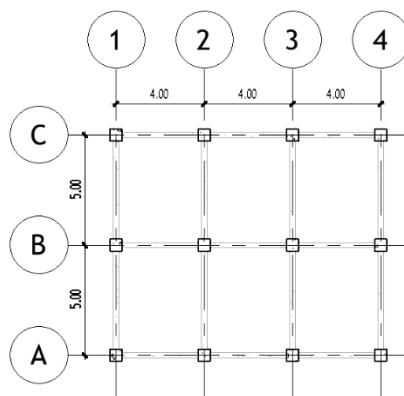


Figura 3.2-1

ELEVACIÓN ESTE-OESTE

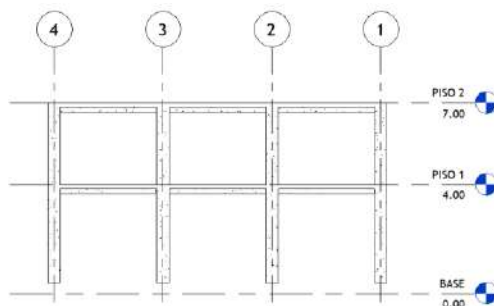


Figura 3.2-2

Nota: Tomado de REVIT 2024 (Autodesk Inc., 2024d) (autoría propia).
ELEVACION NORTE-SUR

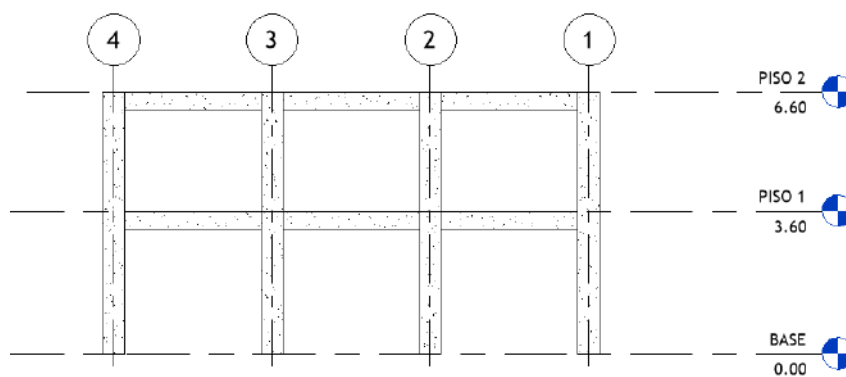


Figura 3.2-3

Nota: Tomado de REVIT 2024 (Autodesk Inc., 2024d)(autoría propia).

3.3. Ubicación del proyecto.

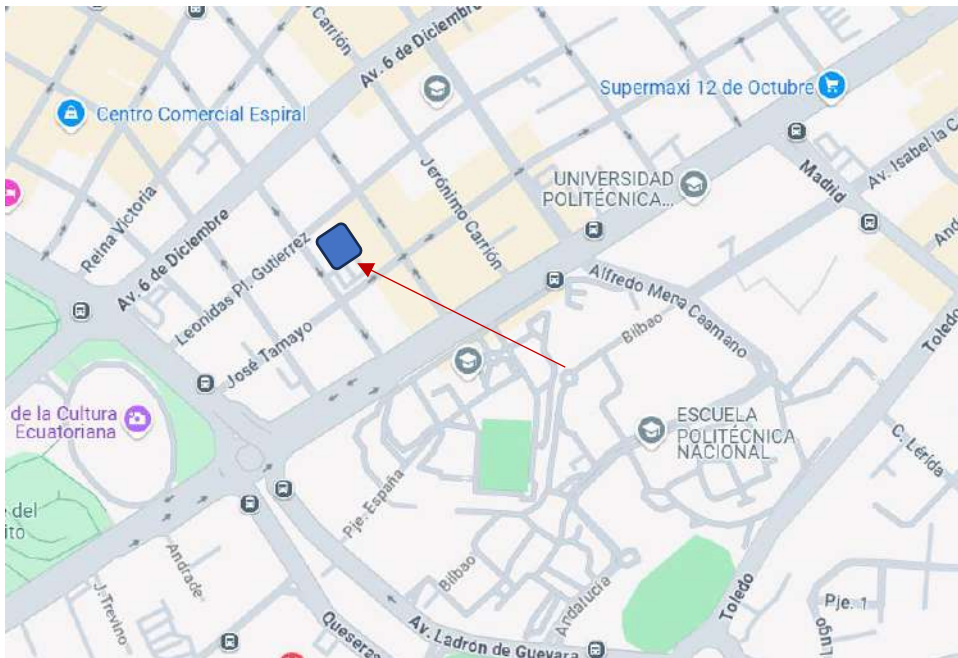


Figura 3.3-1

Nota: Imagen tomada de Google Maps.

El proyecto se ubica en el sector El mariscal precisamente en las calles: Leonidas Pl. Gutierrez y Francisco Robles.

3.4. Características del proyecto.

El proyecto planifica su construcción con el método constructivo: hormigón armado (H.A), teniendo en cuenta los siguientes materiales con sus definidas resistencias especificadas en kg/cm^2 :

Hormigón ($240 \text{ kg}/\text{cm}^2$)

Acero grado 60 ($4200 \text{ kg}/\text{cm}^2$)

3.5. Metodología de análisis y normativa utilizada.

La estructura propuesta se analizará utilizando la normativa NEC 2024 (Borrador) (Ministerio de Desarrollo Urbano y Vivienda, 2024) para los criterios sismorresistentes (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b), y la NEC-SE-CG (2015) (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015a) para las cargas no sísmicas, como metodología de análisis sísmico se utilizará el método análisis modal espectral lo cual conlleva a una definición de un espectro de respuesta relacionado con los parámetros del suelo y factores brindados por la (NEC-SE-DS). (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b).

3.6. Espectro de respuesta (UHS).

El UHS contiene las respuestas en aceleración máximas estadísticas para sistemas con distintos periodos de vibración (T).

Para el análisis del proyecto se utilizará un espectro de respuesta con un periodo de retorno de 475 años establecido por la (NEC-SE-DS). (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b).

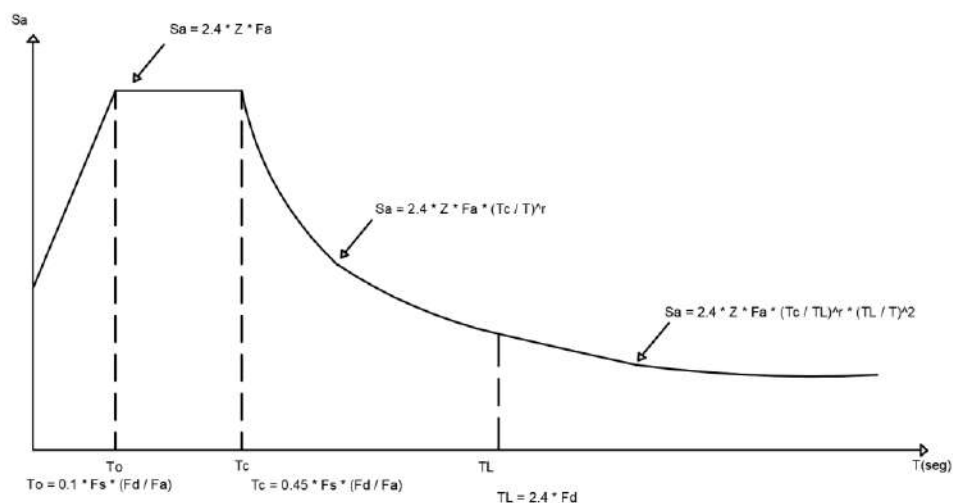


Figura 3.6-1

Nota: imagen tomada de NEC-SE-DS

3.7. Parámetros para el espectro de respuesta y determinación de factores f_a , f_d , f_s y z .

3.7.1. Clasificación del suelo.

Se realizó un estudio preliminar del sector y se concluyó que el proyecto se establecerá en un tipo de suelo **D**.

D	Perfiles de suelos rígidos que cumplan con el criterio de velocidad de la onda de cortante, o	$360 \text{ m/s} > V_s \geq 180 \text{ m/s}$	
	Perfiles de suelos rígidos que cumplan cualquiera de las dos condiciones	$50 > N \geq 15.0$	$100 \text{ kPa} > S_u \geq 50 \text{ kPa}$

Figura 3.7-1

Nota: Tabla tomada de NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024) tabla Tabla 3.2.

3.7.2. Determinación de factor z .

Seismic Disaggregation Tool for Ecuador 2024 (Celi et al., 2024) (SDTE - 2024)

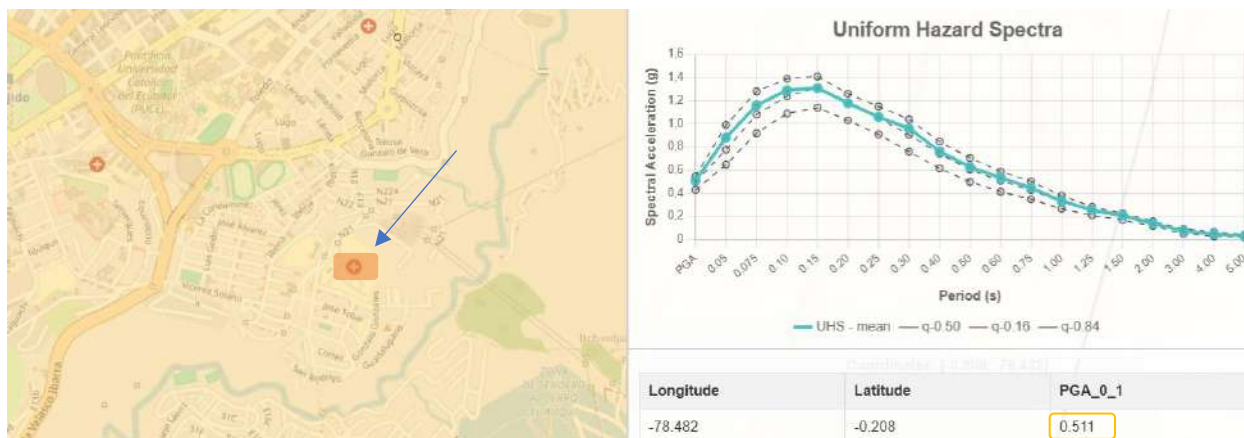


Figura 3.7-2

nota: imagen tomada de <https://sdte-torrefuerte.streamlit.app/#seismic-disaggregation-tool-for-ecuador-2024-sdte-2024> (Celi et al., 2024)

Se determina un factor z en función de su PGA (PEAK GROUND ACCELERATION): 0.511

3.7.3. Determinación de zona sísmica.

Zona Sísmica	Coficiente Z (fracción de la gravedad)
V	$0.5 \leq Z < 0.65$

Figura 3.7-3

Nota: Tabla tomada de NEC-SE-DS Tabla 3.1. (Ministerio de Desarrollo Urbano y Vivienda, 2024)

Debido a que el factor z se determinó con un valor de 0.511 se tiene en la tabla que se recae en una zona sísmica tipo V.

3.7.4. Determinación de factor Fa.

Tipo de Perfil de Suelo	ZONA SISMICA
	V
D	1

Figura 3.7-4

Nota: Tabla tomada de NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024)
Tabla 3.3.2.1.

3.7.5. Determinación de factor Fd.

Tipo de Perfil de Suelo	ZONA SISMICA
D	1

Figura 3.7-5

Nota: Tabla tomada de NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024)

Tabla 3.4.

3.7.6. Determinación de factor F_s .

Tipo de Perfil de Suelo	ZONA SISMICA
D	1.44

Figura 3.7-6

Nota: Tabla tomada de NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024)

Tabla 3.5.

3.8. Parámetros para el espectro de respuesta: importancia sísmica, castigos y factor R.

La importancia sísmica es un factor importante para el cálculo del UHS, este mismo este dictado por la NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015b), y es determinado a través de la ocupación y uso planeado de la estructura, así mismo los castigos de planta y elevación son factores de penalidad calculados a partir de las irregularidades estructurales que presente el proyecto.

3.8.1. Importancia sísmica.

Características de la edificación	Ejemplos	Categoría de Riesgo
Todas las edificaciones, excepto aquellas clasificadas con categorías de riesgo I, III y IV	<ul style="list-style-type: none"> Residencias unifamiliares y bifamiliares que no clasifican dentro de NEC-SE-VIVIENDA Edificios de oficinas, apartamentos, centros comerciales y otras estructuras con capacidad menor a 300 personas 	II

Figura 3.8-1

Nota: Tabla tomada de NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024)

Tabla 1.3 d

Categoría de Riesgo	Factor de Importancia Sísmica, I_e
II	1

Figura 3.8-2

Nota Tabla tomada de NEC-SE-DS Tabla C4.1 (Ministerio de Desarrollo Urbano y Vivienda, 2024)

3.8.2. Factor de reducción sísmica R.

		R					
PORTICOS RESISTENTES A MOMENTOS							
Pórticos especiales resistentes a momento en hormigón	ACI 318-19 (18.6, 18.7, 18.8)	8	3	5.5	SL	SL	SL

Figura 3.8-3

Nota: Tabla tomada de NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024)

Tabla 4.4.

Se determina un factor de R (Factor de reducción sísmica) de: 8.

3.8.3. Irregularidades estructurales horizontales y verticales.

Para los castigos de planta y elevación debido a la regularidad estructural de la estructura

planteada no se recae en ninguno de los casos marcados por la Tabla 5.1. Irregularidades Estructurales Horizontales (BORRADOR-NEC-2024) (Ministerio de Desarrollo Urbano y Vivienda, 2024) ni a su vez en ninguno de los casos de Tabla 5.2. Irregularidades Estructurales Verticales (BORRADOR-NEC-2024) (Ministerio de Desarrollo Urbano y Vivienda, 2024) lo cual nos permite mantener un factor ϕ_e (castigo en elevación) y ϕ_p (castigo en planta) en ambos casos de:

$$\phi_p = 1$$

$$\phi_e = 1$$

$$\therefore \phi_p * \phi_e = 1$$

3.9. Periodo fundamental aproximado.

$$T_a = C_t * h_n^x$$

Donde:

T_a = es el periodo de vibración aproximado, (s)

C_t = Factor dado por la tabla 6.2

h_n = es la altura de la estructura, (m), dada por la distancia vertical desde la base hasta el nivel más alto del sistema resistente a carga sísmica de la estructura.

X = Factor dado por la tabla 6.2

- Parámetros C_t y x para el Cálculo del Periodo de Vibración Aproximado

Tipo de Estructura	Ct	x
Pórticos resistentes a momento de hormigón ^a	0.0466	0.9

Figura 3.9-1

Nota: imagen tomada de NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024)

Tabla 6.2.

Obteniendo así:

$$T_a = 0.0466 * 7^{0.9}$$

Un periodo aproximado de $T_a = 0.268518$

3.10. Obtención grafica del espectro de respuesta (UHS) y determinación de SAE.

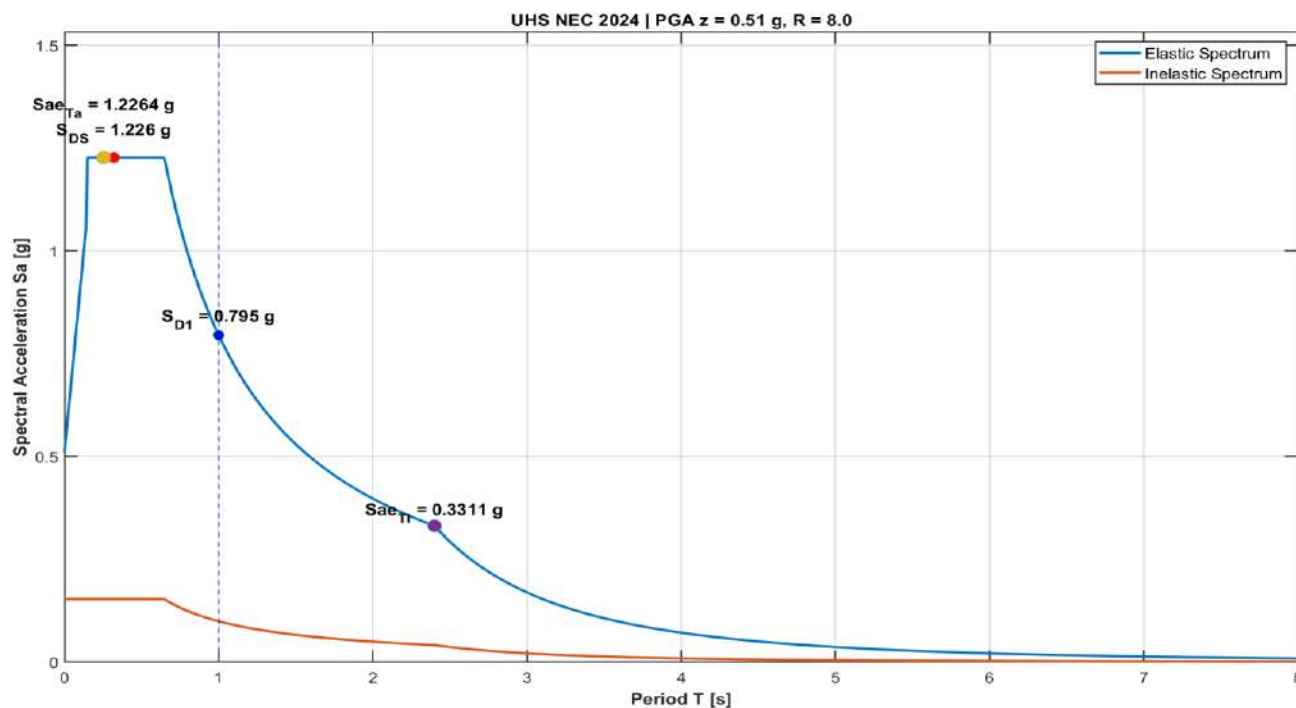


Figura 3.10-1

Nota: imagen de autoría propia tomada de código elaborado en MatLab

Se calcula una aceleración en función de la gravedad de $Sae = 1.226 g$

3.11. Cálculo de cargas no sísmicas.

3.11.1. Sobrecarga

Sobrecarga		
Enlucido y masillado:	0.035	ton/m ²
Recubrimiento de piso:	0.015	ton/m ²
Peso Mampostería (asumido)	0.2	ton/m ²
Sobrecarga Total	0.25	ton/m ²

Figura 3.11-1

Nota: Tabla de autoría propia

El enlucido y masillado se calculó tomando en cuenta un espesor de **2 cm** y un peso específico de 1.75 ton/m^3 , el recubrimiento de piso se calculó tomando en cuenta un espesor de **1.50 cm** y un peso específico de 1 ton/m^3 .

3.11.2. Carga Viva

Categoría	Subcategoría	Valor
Residencias	Viviendas (unifamiliares y bifamiliares)	2

Figura 3.11-2

Nota: Tabla tomada de NEC-SE-CG Tabla 9 (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015a)

La tabla se encuentra en valores de Kn/m^2 . lo que equivale a una carga viva de **0.2 t/m^2** .

3.11.3. Peso Propio (PP)

3.11.4. Losa

Para el cálculo de peso propio se empezó por el espesor de Losa en función de la luz máxima de arquitectura propuesta, en este caso tenemos:

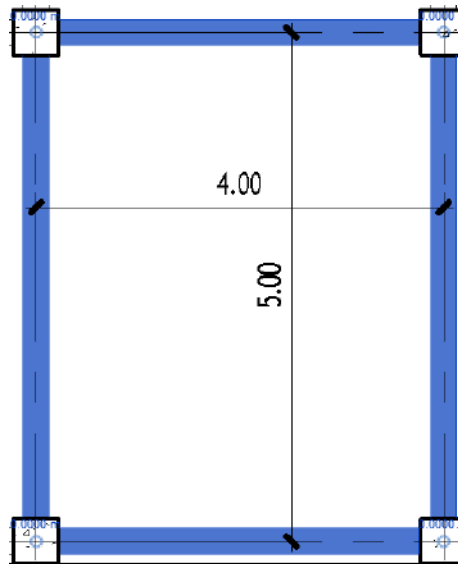


Figura 3.11-3

Nota : imagen de autoría propia extraída de REVIT (Autodesk Inc., 2024d)

La losa de mayor paño comprende un área casi cuadrada por lo tanto Según el ACI 318-19 (American Concrete Institute, 2019), una losa se considera bidireccional cuando la relación entre la luz mayor y la luz menor no excede 2.0. Este criterio se encuentra en el comentario R13.1.3.1, donde se indica que losas con proporciones menores o iguales a este valor deben diseñarse como sistemas de dos direcciones, ya que distribuyen cargas significativamente en ambas direcciones principales.

American Concrete Institute. (2019). *Building code requirements for structural concrete (ACI 318-19) and commentary* (p. 395, R13.1.3.1). Farmington Hills, MI: ACI.

Esto nos permite remitirnos a la tabla 8.3.1.1:

fy (MPa)	Sin abacos	
	Paneles exteriores	
	Sin vigas de borde	Con vigas de borde
280	$l_n / 33$	$l_n / 36$
420	$l_n / 30$	$l_n / 33$
520	$l_n / 28$	$l_n / 31$

Figura 3.11-4.

Nota: Tabla sacada de la ACI 318-19 Tabla 8.3.1.1 (American Concrete Institute, 2019)

Debido a las condiciones geométricas y constructivas se recae en el caso marcado, donde la estimación de la altura de losa puede ser considerada como $l_n/33$ siendo l_n la luz libre del tramo

Como de l_n (Luz libre) en el caso de la estructura planteada se tiene 4.40 m, la losa calculada sería:

$$\frac{4400}{33} = 13.333 \approx 15 \text{ cm}$$

Por lo tanto, teniendo en cuenta un $\gamma_H = 2400 \frac{\text{kg}}{\text{m}^3}$ se tiene que por un metro cuadrado de losa el peso propio se define por la siguiente expresión:

$$Peso_{Losa} = 0.15\text{m} \times 1\text{m} \times 1\text{m} \times 2.4 \frac{\text{Ton}}{\text{m}^3} = 0.36 \text{ Ton}$$

$$PP_{Losa} = 0.36\text{Ton} / 1\text{m}^2 = 0.36 \text{ Ton/m}^2$$



Figura 3.11-5

Nota: Imagen de autoría propia tomada del software REVIT (Autodesk Inc., 2024d)

PP de Columnas y Vigas

Para las columnas y vigas en este trabajo se estimaron dimensiones esperadas de acuerdo con la sollicitación, uso de la estructura y la experiencia ingenieril representada en obras cercanas al lugar de construcción: El mariscal, cabe aclararse que este no es un prediseño como tal, solo son suposiciones iniciales para el cálculo del peso propio de los materiales, estas dimensiones pueden variar en el prediseño definitivo.

Vigas

Se supuso inicialmente una viga de 25 cm x 35 cm de alto como se muestra a continuación:

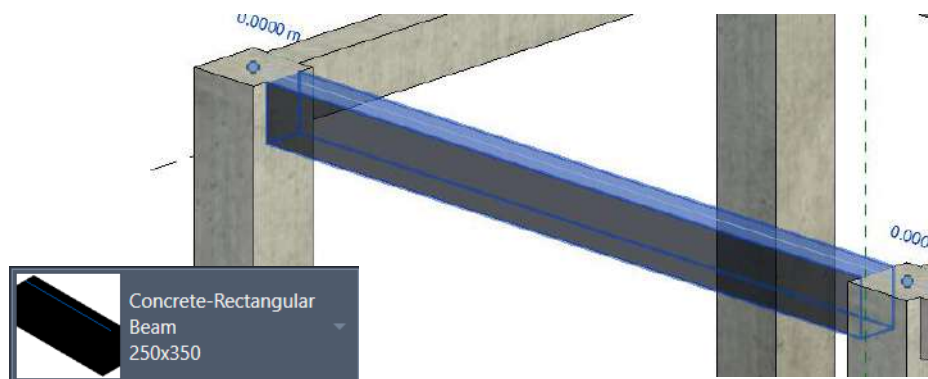


Figura 3.11-6

Nota: Imagen de autoría propia tomada del software REVIT (Autodesk Inc., 2024d)

Teniendo en cuenta las dimensiones asumidas se extrae del software Revit a través de una tabla de planificación el volumen de hormigón de todas las vigas del proyecto (Structural framing), obteniendo así:

Elemento	Parámetro	Valor
Vigas	Volume	10.13 m ³

Figura 3.11-7

Nota: Tabla de autoría propia transcrita de resultados arrojados por REVIT.

Considerando este volumen y un $\gamma_H = 2400 \frac{kg}{m^3}$ se tiene que el peso de todas las vigas actuante en la estructura es:

$$Peso_{vigas} = 10.13m^3 \times 2.4 \frac{Ton}{m^3} = 24.312 \text{ Ton}$$

Este es el peso actuante de las vigas en todo el edificio, para facilidad de cálculo debido a la simetría estructural se asumirá que ese peso se distribuye por igual en todo el área de planta del edificio, la cual extraída de una tabla de REVIT (Autodesk Inc., 2024d) es la siguiente:

Elemento	Parámetro	Valor
AREA DE LOSA	Area	260.21 m ²

Figura 3.11-8

Nota: Tabla de autoría propia transcrita de resultados arrojados por REVIT.

Por lo tanto, se divide el valor de peso calculado para el área bruta de planta teniendo así un peso propio de vigas de:

$$PP_{Viga} = 24.312Ton / 260.21m^2 = 0.0934 \text{ Ton}/m^2$$

Columnas

Se supuso inicialmente una columna de 45 cm x 45 cm de alto como se muestra a continuación:

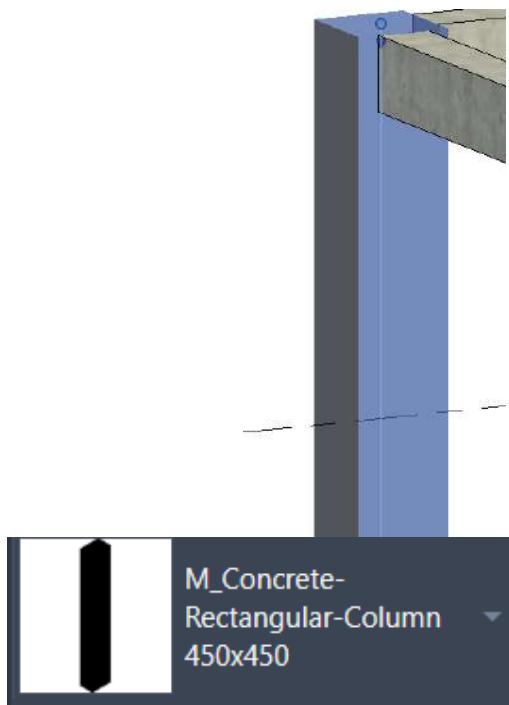


Figura 3.11-9

Teniendo en cuenta las dimensiones asumidas se extrae del software Revit (Autodesk Inc., 2024d) a través de una tabla de planificación el volumen de hormigón de todas las columnas del proyecto (Structural columns), obteniendo así:

Elemento	Parámetro	Valor
Columnas	Volume	15.86 m ³

Figura 3.11-10.

Nota: Tabla de autoría propia transcrita de resultados arrojados por REVIT (Autodesk Inc.,

2024d).

Considerando este volumen y un $\gamma_H = 2400 \frac{kg}{m^3}$ se tiene que el peso de todas las columnas actuante en la estructura es:

$$Peso_{Columnas} = 15.86m^3 \times 2.4 \frac{Ton}{m^3} = 38.064 \text{ Ton}$$

Por lo tanto, al igual que en vigas se asume rigidez y simetría en la planta, por lo tanto, se divide el valor de peso calculado para el área bruta de planta teniendo así un peso propio de columnas de:

$$PP_{Columna} = 38.064Ton / 260.21m^2 = 0.1462 \text{ Ton}/m^2$$

Teniendo los efectos del PP en m^2 de cada elemento estructural podemos sumarlos y obtener el peso total por m^2 debido a Peso Propio:

	Peso por m2
Vigas	0.0934
Columnas	0.1463
Losa	0.3600
Total	0.5997

Figura 3.11-11

Sin embargo, para fines de cálculo posteriores en este trabajo también se desglosará el peso propio que cargan las Vigas, solo separando PPlosa y PPvigas, teniendo así una carga de peso propio para columnas (Losa, vigas y losa) y otro para vigas (Losa y vigas).

	COLUMNAS	VIGAS	LOSAS	PESO PROPIO TON/m ²
COLUMNAS	0.1463	0.0934	0.3600	0.5997
VIGAS		0.0934	0.3600	0.4534

Figura 3.11-12

Es decir, en los cálculos posteriores donde se utilice Peso propio dependerá de esta tabla.

3.12. Predimensionamiento de viga.

Para el predimensionamiento de vigas primero se buscó la viga más desfavorable de la estructura, es decir la de luz más larga, después se supuso una distribución de cargas tipo sobre como se muestra a continuación:

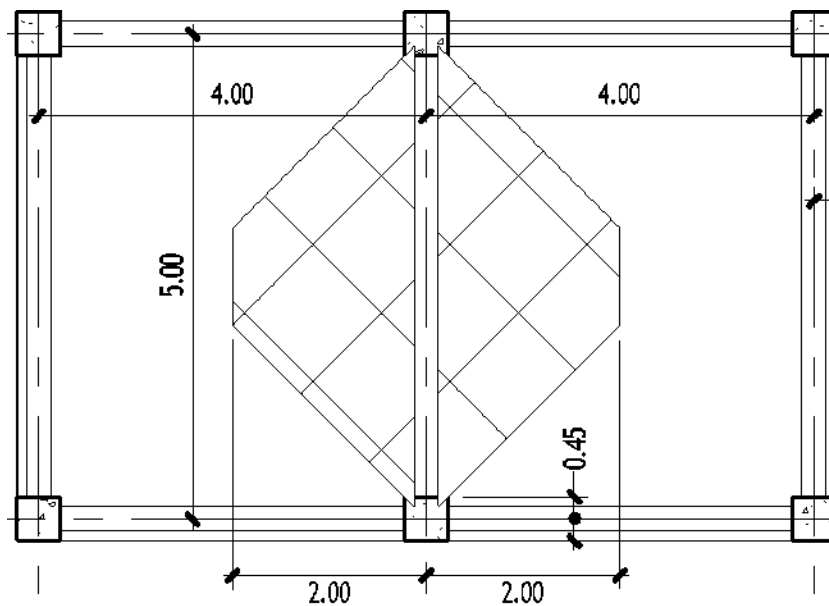


Figura 3.12-1

Nota: imagen de autoría propia extraída de software REVIT (Autodesk Inc., 2024d).

Se aprecia la luz de 5m cargando dos vanos simétricos con altura de 2m en el área cargada, esta distribución conforma un área cargada de tipo trapezoidal, por lo cual para fines de facilidad de cálculo se transforma a una carga cuadrada uniformemente distribuida a través de la siguiente expresión:

$$W_L = \frac{q * s}{3} * \left(\frac{3 - m^2}{2}\right)$$

Siendo:

W_L : Es la carga rectangular equivalente

q: carga por m^2

s: Lado menor

L: Lado mayor

m: Relación entre lado mayor y el lado menor $m = s/L$

Se considero un factor de amplificación sísmica $f=1.60$ para el cálculo de W_L y una resistencia última nominal a flexión $R_u = 45.38$ que se obtuvo en función del $f'c = 210 \text{ kg/cm}^2$ para el cálculo del peralte de viga.

En el caso de este trabajo tenemos:

$$q = (1.2 (\text{sobrecarga} + \text{PesoPropio}) + 1.6(\text{garga viva}) = 1.2 (0.25 + 0.4534) + 1.6(1.2)) * f$$

$$q = 1.3561 \text{ Ton/m}^2$$

$$s = 4 \text{ m}$$

$L = 5 \text{ m}$

$m = 1$

Por lo tanto, reemplazando tenemos que $WL = 2.1336 \text{ Ton/m}$, como son dos paños o trapecios cargados por la viga se multiplica este valor para obtener un $WL_{TOTAL} = 2.1336 \times 2 = 4.2671 \text{ TON/m}$

Para obtener el momento actuante de la viga se utiliza la siguiente expresión brindada por la ACI (American Concrete Institute, 2019):

$$\frac{w_u \cdot l_n^2}{10}$$

Nota: ecuación tomada de ACI 3-18 Tabla 6.5.2. (American Concrete Institute, 2019).

Siendo $l_n = L$ en este caso debido a que aún no se tiene dimensiones de columnas.

Se obtiene que: $M_u = 10.6678 \text{ Ton.m}$

Por último, se calcula el d en función de la siguiente expresión:

$$d_{\text{viga}} = \sqrt{\frac{M_u \cdot 10^5}{0.9 \cdot R_u \cdot (b \cdot 100)}} + \text{Recubrimiento}$$

Nota: ecuación obtenida de (Guerra Avendaño & Guerra Valladares, 2020)

Suponiendo un b (ancho de viga) de 25 cm y un R_u calculado de 39.72 se obtiene un peralte de:

$d_{\text{viga}} = 40 \text{ cm}$

La viga predimensionada queda de la siguiente manera:

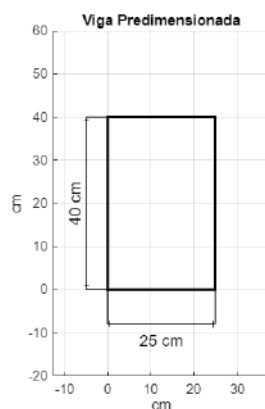


Figura 3.12-2

Nota: Imagen de autoría propia extraída de código en MatLab.

3.13. Predimensionamiento de columna.

3.13.1. Cálculo de carga axial para columna.

Para el predimensionamiento de columnas primero se buscó la columna más desfavorable de la estructura, es decir la columna con más área portante.

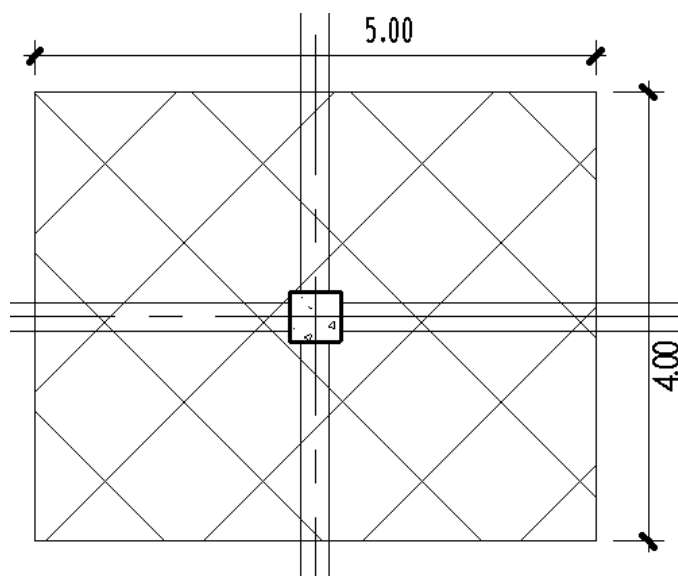


Figura 3.13-1

Nota: imagen de autoría propia extraída de software REVIT (Autodesk Inc., 2024d).

Se puede apreciar que la columna más cargada (columna central) mantiene un área portante de $20m^2$ producto de la multiplicación $4 \times 5m$.

Para el cálculo de la carga axial a la cual está sometida la columna debido a un piso se calcula la combinación de cargas siguiente.

$$W = (1.2 \times (PP+ CM) \times \text{Área portante}) + (1.6 \times CV \times \text{área portante})$$

Teniendo todos los datos calculados nos queda:

$$W = (1.2 \times (0.5997+ 0.25) \times 20) + (1.6 \times 0.2 \times 20) = 26.79 \text{ ton}$$

Teniendo la carga axial que llega a la columna por un piso, resta multiplicarlo por el número de pisos para calcular la carga axial a la columna de la primera planta (más desfavorable).

$$W \text{ TOTAL} = 26.79 \times 2 \text{ pisos} = 53.5856 \text{ Ton}$$

3.13.2. Cálculo de momento para columna.

Para el cálculo de momento es necesario primero calcular el peso de la estructura, así como el cortante basal, para lo cual se utilizó el espectro de respuesta calculado anteriormente.

El cortante se calculó utilizando la siguiente expresión facilitada por la NEC

$$V = \frac{I_e \cdot S_{aT}}{R} \cdot W$$

Nota: expresión tomada de NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024).

Para lo cual se calcula el peso de la estructura utilizando la siguiente expresión:

$$W = (1.2 \times (PP+ CM) \times \text{Area Total}) + (1.6 \times CV \times \text{Area Total})$$

$$W = (1.2 \times (0.5997+ 0.25) \times 260.21) + (1.6 \times 0.2 \times 260.21)$$

$$W = 348.587 \text{ Toneladas}$$

Nota: Para el cálculo de: área total, el peso propio y la sobrecarga ver 3.10. Cálculo de cargas no sísmicas.

Teniendo los datos previamente calculados tenemos:

$$V = \frac{1 \cdot 1.2264}{8} \cdot 348.547$$

$$V = 53.43 \text{ Ton}$$

Nota: Para el cálculo de: importancia, SaT y R ver 3.7 PARÁMETROS PARA EL ESPECTRO DE RESPUESTA: IMPORTANCIA SÍSMICA, CASTIGOS Y FACTOR R

3.13.3. Distribución de cortante basal.

Para la distribución de cortante basal se tomó el sentido más desfavorable, en este caso el sentido Y: logrando así 3 pórticos equivalentes. También se asume la distribución del cortante en los dos pisos de la estructura y la descomposición en X y Y del mismo.

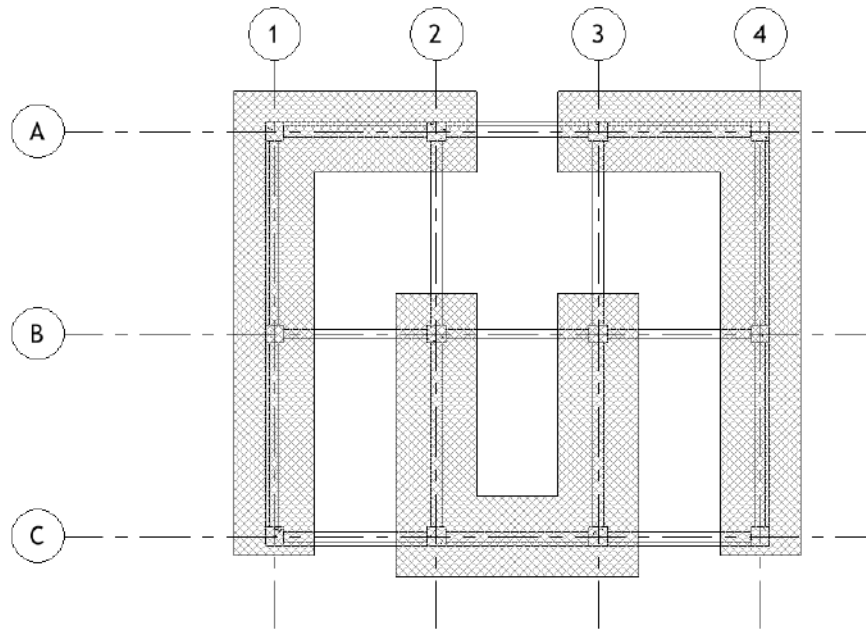


Figura 3.13-2

Por lo tanto, el cortante Basal quedaría de la siguiente manera:

$$V_{columna} = \frac{53.43 \text{ ton}}{3 \text{ porticos} \times 2 \text{ pisos} \times 2 \text{ direcciones}}$$

$$V_{columna} = 4.45 \text{ Ton}$$

Teniendo de esta manera los siguientes datos de corte en la columna:

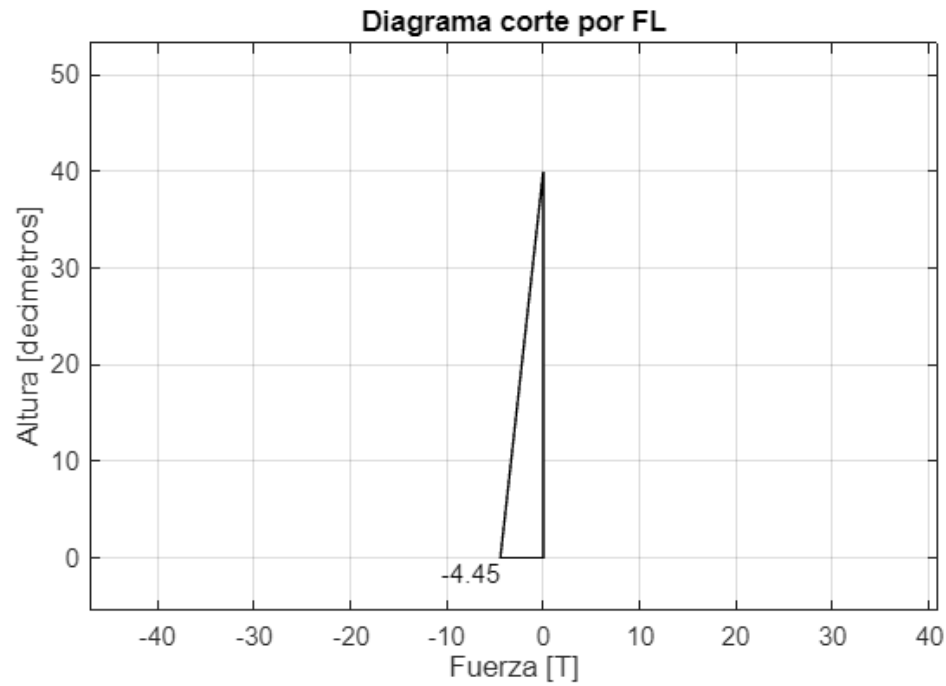


Figura 3.13-3

Nota: Imagen de autoría propia extraída de código en MatLab. (MathWorks, 2024)

Sabiendo la altura del primer piso podemos sacar el diagrama de momentos que afectara a la columna a través de una simple operación:

$$M_{columna} = \frac{4.45 \times 4}{2} = 8.9 \text{ Ton.m}$$

Teniendo el siguiente diagrama de momentos:

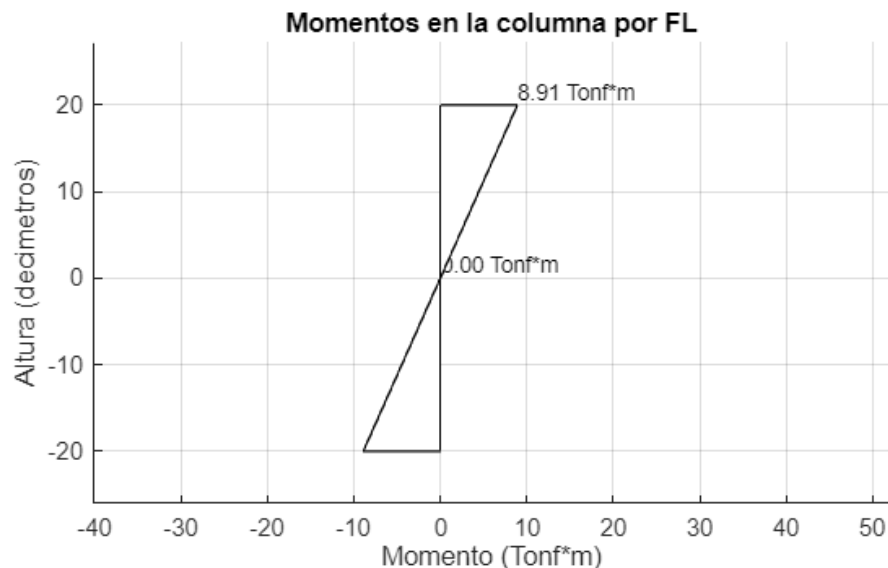


Figura 3.13-4

Nota: Imagen de autoría propia extraída de código en MatLab. (MathWorks, 2024)

A manera de resumen:

Mucol	Wcolumna
8.9064	53.5856

Figura 3.13-5

3.13.4. Diagramas de interacción y secciones finales de columna.

Con fin de comprobar las secciones finales iteramos áreas de columnas en el diagrama de interacción incluyendo un armado del 1% requerido como mínimo por la norma para columnas, las secciones son arbitrarias y se va iterando hasta lograr que la demanda entre dentro de la

envolvente de capacidad de la columna.

Primera interacción COLUMNA 30X30 con 8 varillas de 12 mm (uno por ciento de armado)

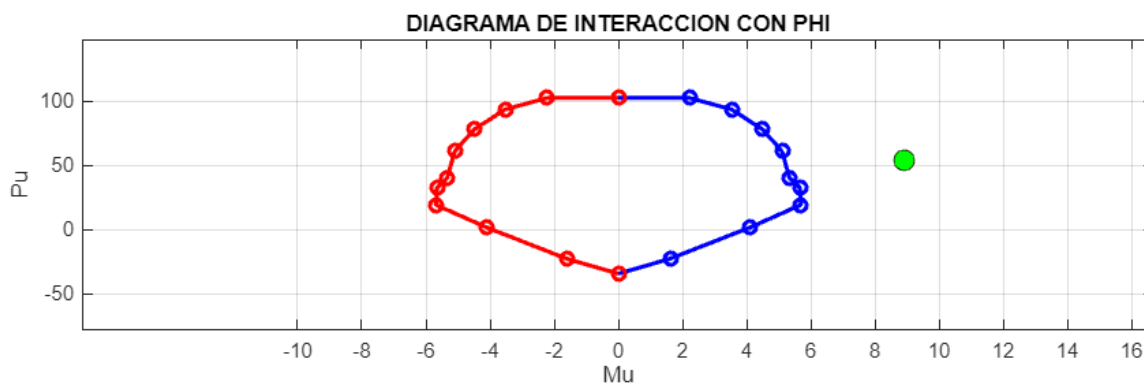


Figura 3.13-6

Nota: Imagen de autoría propia extraída de código en MatLab. (MathWorks, 2024)

Como se aprecia la demanda supera la capacidad en esta iteración por lo que se procede a subir sección.

Segunda interacción COLUMNA 40x40 8 varillas de 16 (uno por ciento de armado)

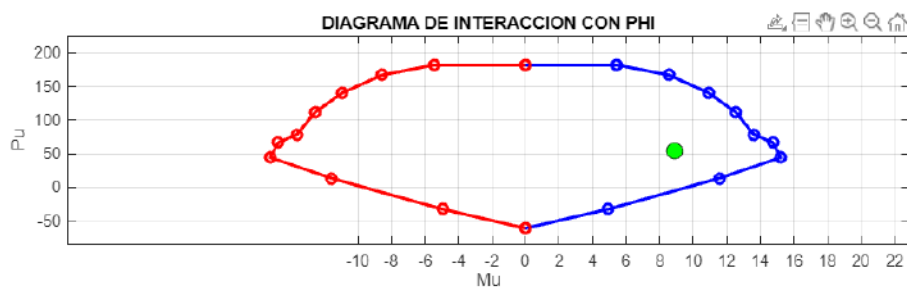


Figura 3.13-7

Nota: Imagen de autoría propia extraída de código en MatLab. (MathWorks, 2024)

Como se aprecia la demanda quedo dentro de la envolvente de capacidad de la columna por lo que se definen las secciones finales de la columna a continuación.

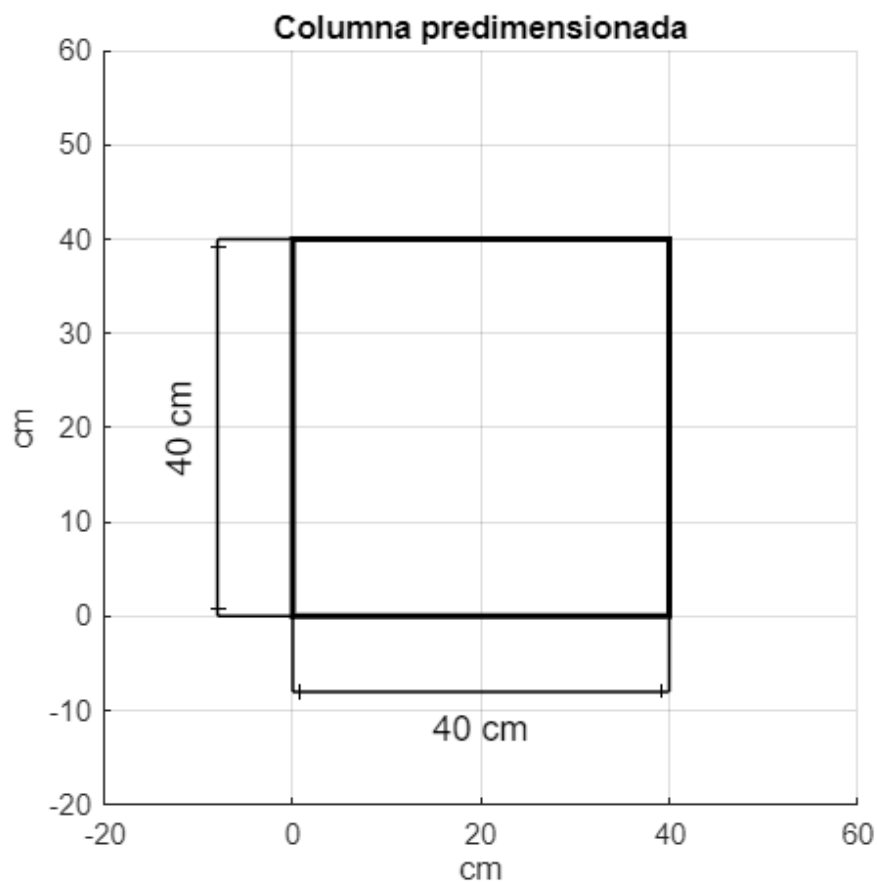


Figura 3.13-8

Nota: Imagen de autoría propia extraída de código en MatLab. (MathWorks, 2024)

3.14. Resumen de las dimensiones de elementos estructurales.

ELEMENTO	INFORMACION
Columnas	40cm x 40cm
Viga	25cm x 40cm
Losa	maciza-15 cm

Figura 3.14-1

CAPÍTULO IV

4. ANALISIS DE LA ESTRUCTURA EN SOFTWARE COMERCIAL.

4.1. Modelado de la estructura.

Para el modelado de una estructura existen varios softwares de interoperabilidad de programas para exportar una estructura de un software de dibujo tal como REVIT (Autodesk Inc., 2024d) y pasarlo directamente a un software de cálculo estructural como lo es el ETABS (Computers and Structures, 2023), debido a la simplicidad geométrica de la arquitectura propuesta en este trabajo se optó por el dibujo manual dentro del mismo software de cálculo, empezando de esta manera con la definición de grillas y niveles.

4.1.1. Grillas

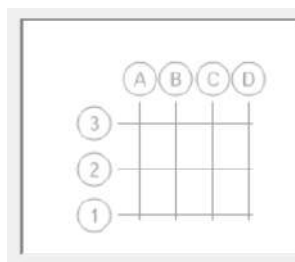


Figura 4.1-1

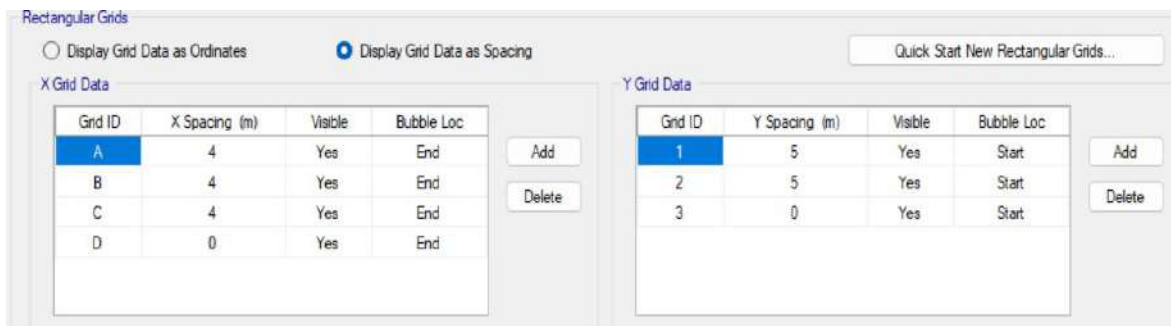


Figura 4.1-2

Nota: Imagen de autoría propia extraída de software ETABS.(Computers and Structures, 2023)

4.1.2. Niveles

	Story	Height m	Elevation m	Master Story	Similar To	Splice Story	Splice Height m	Story Color
▶	Story2	3	7	No	Story1	No	0	Yellow
	Story1	4	4	Yes	None	No	0	Grey
	Base		0					White

Figura 4.1-3

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

4.1.3. Dibujo Paramétrico

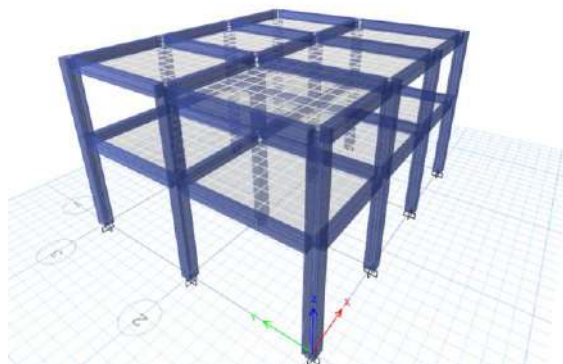


Figura 4.1-4

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

4.2. Definición de materiales y secciones.

Una vez dibujada la estructura paramétrica pasamos a definir materiales y secciones, esta información se cambia automáticamente en el modelo. Los materiales y secciones serán definidos de acuerdo con el predimensionamiento calculado anteriormente, así como los parámetros de materiales definidos en un principio del trabajo.

4.2.1. Materiales de diseño.

Para la determinación del módulo de elasticidad se utiliza la siguiente expresión definida por la NEC-SE-HM en la sección 3.3.3. (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015c)

$$E_c = 4.77\sqrt{f'_c}$$

E_c : Módulo de elasticidad del concreto (en GPa)

f'_c : Resistencia a la compresión del concreto (en MPa)

Nota: Ecuación tomada de NEC-SE-HM sección 3.3.3 (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015c)

De esta manera considerando una resistencia del hormigón a la compresión de 240 kg/cm^2 , tenemos que:

f'_c (kg/cm ²)	f'_c (Mpa)	E_c (Gpa)	E_c (Kg/cm ²)
240	23.5359648	22.8015232	232510.8294

Figura 4.2-1

Una vez calculado el módulo de elasticidad del hormigón procedemos a definir los materiales de hormigón y acero en el software.

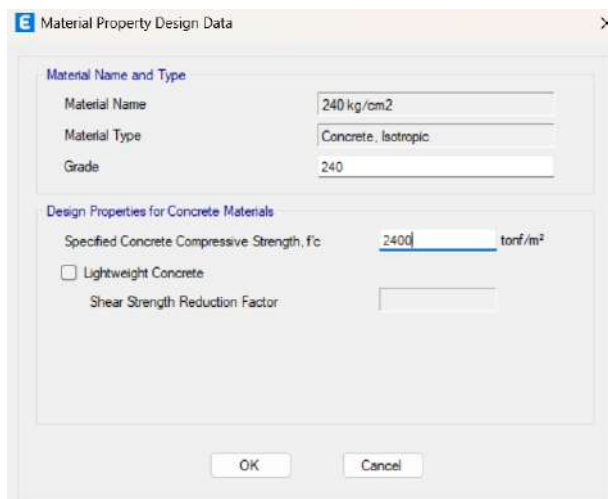


Figura 4.2-2

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)



Figura 4.2-3

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

Material Property Data

General Data

Material Name: Acero tesis

Material Type: Rebar

Directional Symmetry Type: Uniaxial

Material Display Color: Change...

Material Notes: Modify/Show Notes...

Material Weight and Mass

Specify Mass Density

Weight per Unit Volume: 7.849 tonf/m³

Mass per Unit Volume: 0.80038 tonf-s²/m⁴

Mechanical Property Data

Modulus of Elasticity, E: 20389019.16 tonf/m²

Coefficient of Thermal Expansion, A: 0.0000117 1/C

Design Property Data

Modify/Show Material Property Design Data...

Advanced Material Property Data

Nonlinear Material Data...

Material Damping Properties...
Time Dependent Properties...

OK Cancel

Figura 4.2-4

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

Material Name and Type	
Material Name	Acero tesis
Material Type	Rebar, Uniaxial
Grade	Grade 60

Design Properties for Rebar Materials	
Minimum Yield Strength, Fy	42184.18 tonf/m ²
Minimum Tensile Strength, Fu	63276.27 tonf/m ²
Expected Yield Strength, Fye	46402.6 tonf/m ²
Expected Tensile Strength, Fue	69603.89 tonf/m ²

Figura 4.2-5

Nota: Imagen de autoría propia extraída de software ETABS (Computers and Structures, 2023).

4.2.2. Ingreso de secciones.

Ingresamos las dimensiones antes predimensionadas de los elementos estructurales.

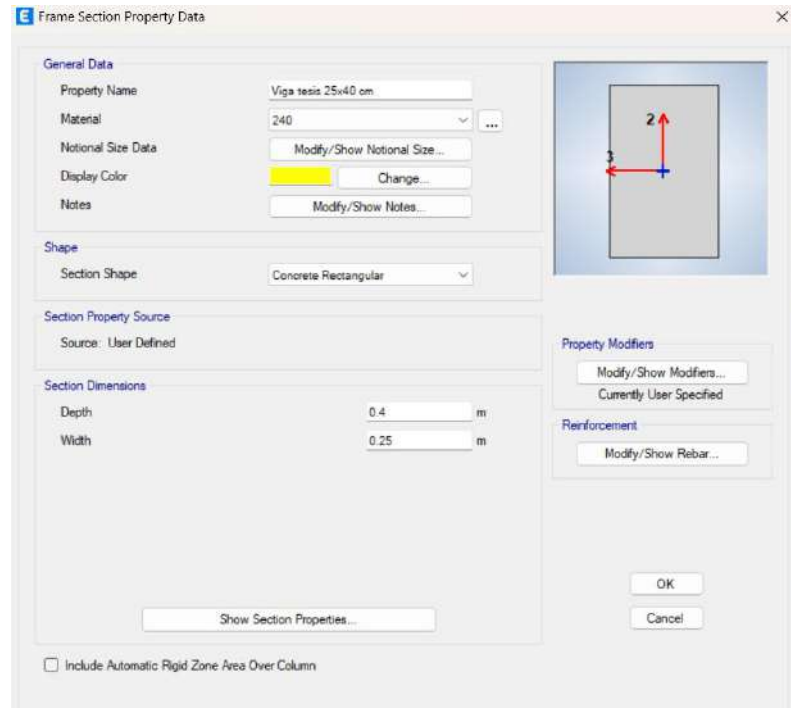


Figura 4.2-6

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023).

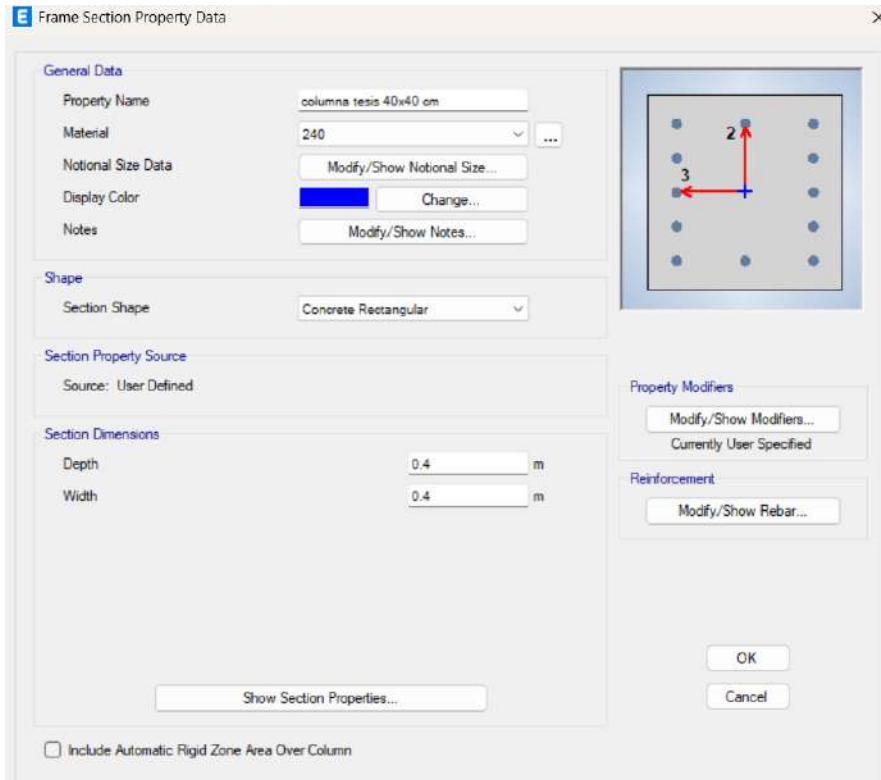


Figura 4.2-7

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023).

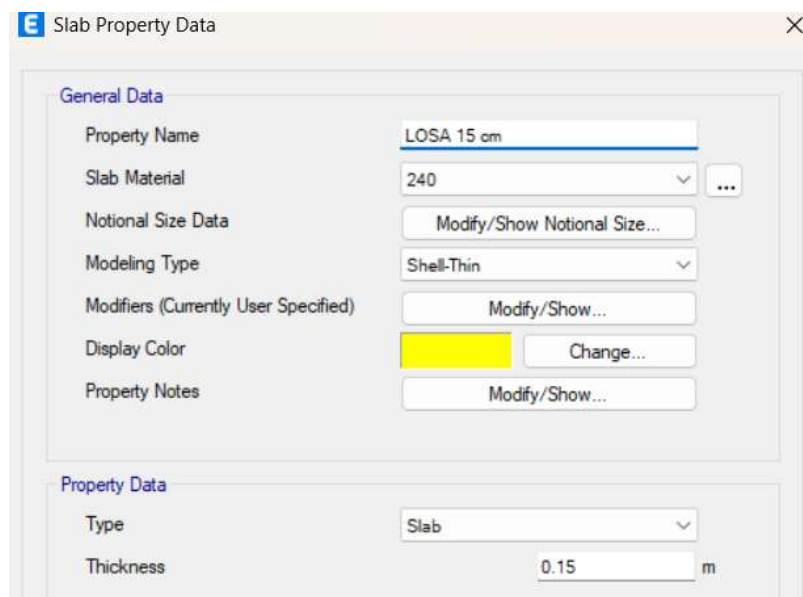


Figura 4.2-8

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

4.3. Aplicación de inercias efectivas en elementos.

Para las inercias efectivas se siguen los valores establecidos por la NEC-SE-DS (Ministerio de Desarrollo Urbano y Vivienda, 2024) a continuación:

Estructuras de hormigón armado

“En este caso, en el cálculo de la rigidez y de las derivas máximas se deberán utilizar los valores de las inercias agrietadas de los elementos estructurales, de la siguiente manera:”

0.5 I_g para vigas (considerando la contribución de las losas, cuando fuera aplicable)

0.8 I_g para columnas

0.6 I_g para muros estructurales

Norma Ecuatoriana de la Construcción – Diseño Sísmico, **NEC-SE-DS** (Ministerio de Desarrollo Urbano y Vivienda, 2024) , Sección **6.1.6(b)**.

Columnas

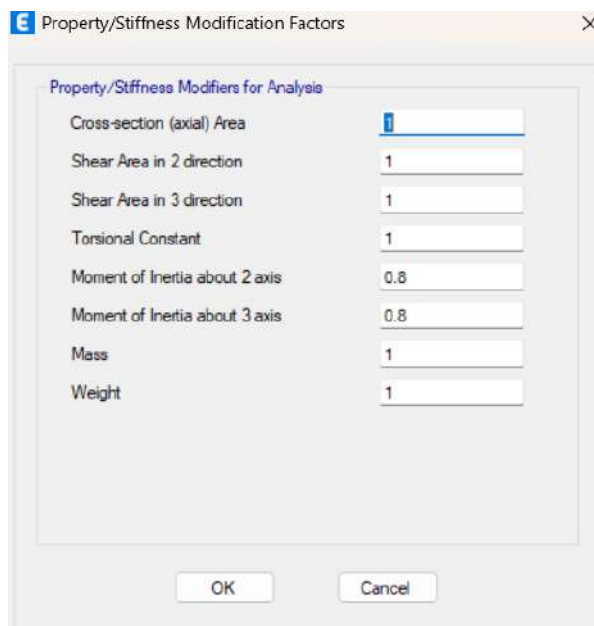


Figura 4.3-1

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023).

Vigas

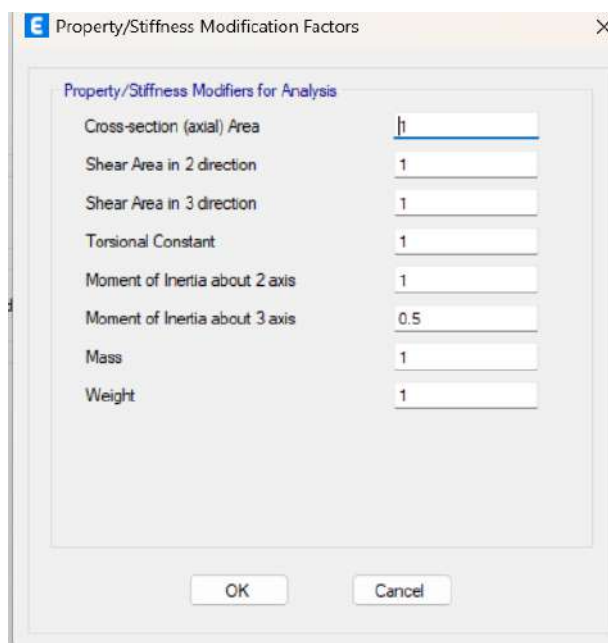
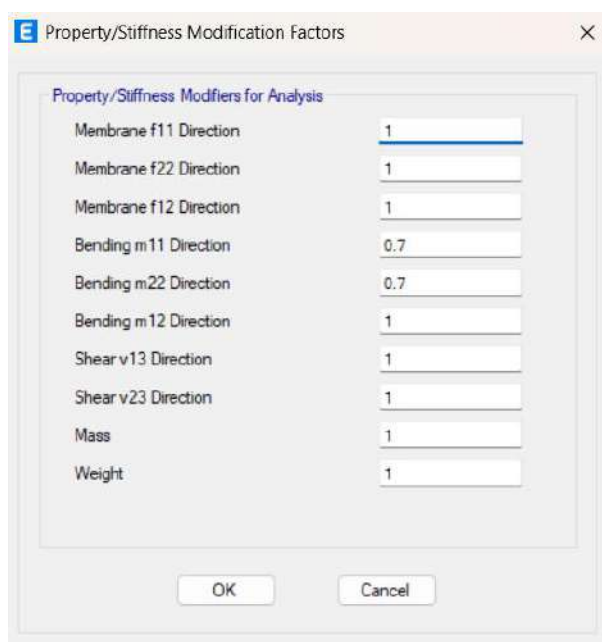


Figura 4.3-2

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023).

Losa



Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023).

4.4. Definición de cargas y espectros de respuesta.

4.4.1. Definición en Load Patterns.

Load	Type	Self Weight Multiplier	Auto Lateral Load
Dead	Dead	0	
Dead	Dead	0	User Coefficient
Live	Live	0	User Coefficient
SismoX	Seismic	0	
SismoY	Seismic	0	
PP	Dead	1	

Figura 4.4-1

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

4.4.2. Coeficientes basales de Sx y Sy estáticas.

Para el coeficiente basal se toma los datos de cortante basal y peso de la estructura del predimensionamiento en el capítulo III y realizamos la relación.

$$\text{coeficiente} = \frac{V}{W}$$

$$\text{coeficiente} = \frac{53.43}{348.57} = 0.1532$$

Nota: Para el cálculo de V (cortante basal) y W (peso de la estructura) ver 3.12.2

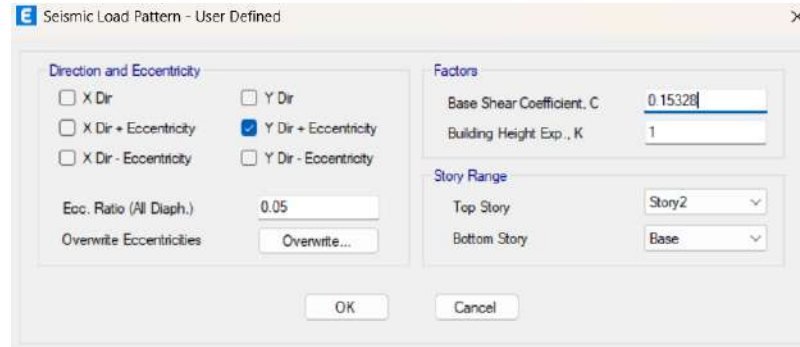


Figura 4.4-2

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

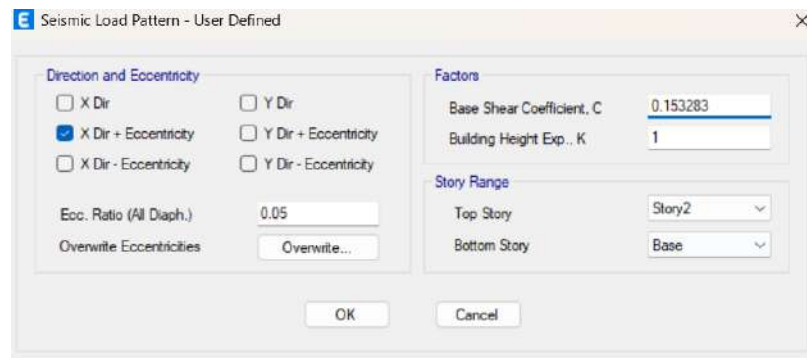


Figura 4.4-3

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

4.4.3. Carga de espectros NEC-2024 y ASCE 7-10.

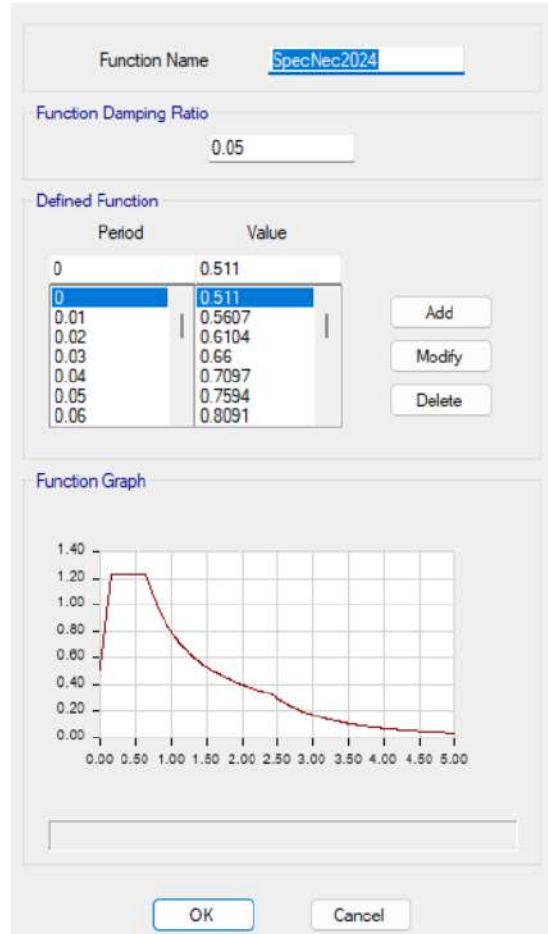


Figura 4.4-4

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

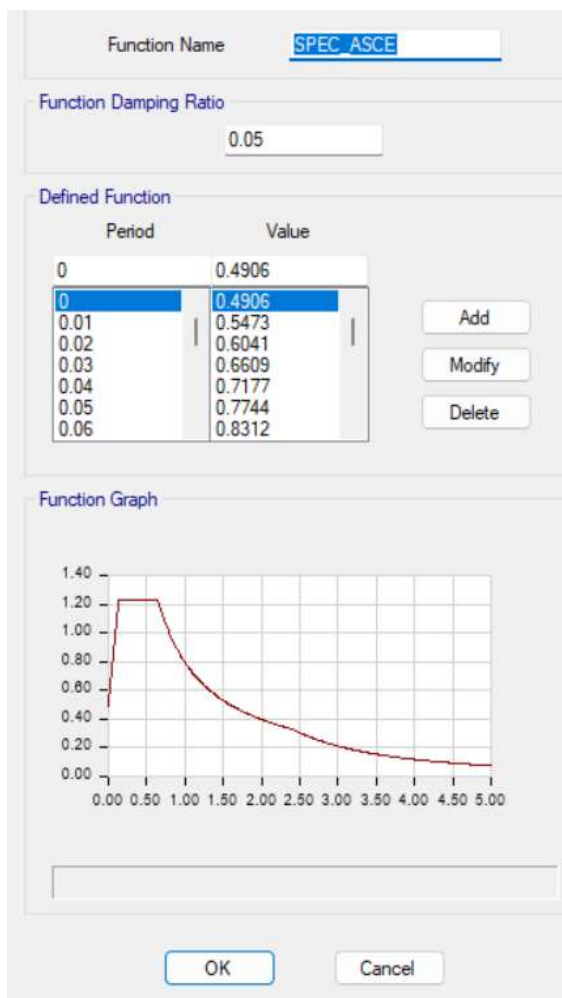


Figura 4.4-5

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

La igualación de espectros de la NEC (Ministerio de Desarrollo Urbano y Vivienda, 2024) y de la ASCE se realiza internamente en el ETABS (Computers and Structures, 2023) a través de los factores S_{d1} y S_{ds} .

4.5. Meshing

En el programa se realiza un meshing en los elementos tipo losa para una correcta interacción y transferencia de esfuerzos entre la losa y los pórticos, esto se realiza automáticamente

intentando cumplir las 3 reglas del meshing: polígonos de 3 y 4 lados máximo, relaciones máximas de 2 a 1 y conectar los elementos en las esquinas de los polígonos.

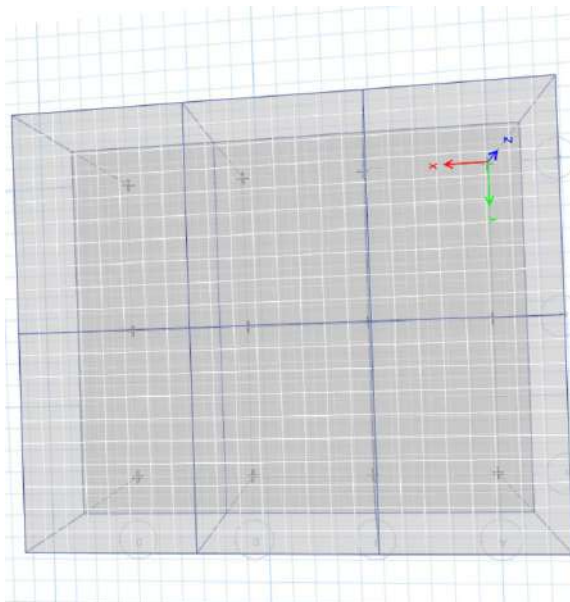


Figura 4.5-1

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

4.6. Combinaciones de carga.

Para las combinaciones de carga utilizamos las brindadas por la ACI:(American Concrete Institute, 2019)

Tabla 5.3.1- Combinaciones de carga

Combinación de carga	Ecuación	Carga primaria
U = 1.4D	(5.3.1a)	D
U = 1.2D + 1.6L + 0.5(Lr o S o R)	(5.3.1b)	L
U = 1.2D + 1.6(Lr o S o R) + (1.0L o 0.5W)	(5.3.1c)	Lr o S o R
U = 1.2D + 1.0W + 1.0L + 0.5(Lr o S o R)	(5.3.1d)	W
U = 1.2D + 1.0E + 1.0L + 0.2S	(5.3.1e)	E
U = 0.9D + 1.0W	(5.3.1f)	W
U = 0.9D + 1.0E	(5.3.1g)	E

Figura 4.6-1

Nota: ecuación tomada de ACI 3-18 Tabla 5.3.1. (American Concrete Institute, 2019)

Sin embargo, para las combinaciones de cargas con sismo se decide hacer un análisis ortogonal tomando en cuenta el 100% de sismo principal en una dirección y el 30% de sismo opuesto en otra dirección, adicional a esto se descompone la componente vertical del sismo que se sumó a la carga muerta de las combinaciones.

4.6.1. Cálculo E_v (componente vertical del sismo).

Para este cálculo se utiliza la siguiente expresión facilitada por la ASCE 7-16 (American Society of Civil Engineers, 2017)

Ecuación 12.4.4a

$$E_v = 0.2 \cdot S_{DS} \cdot D$$

Nota: ecuación tomada de ASCE 7-16 eq 12.4-4a.

Siendo:

E_v = Componente Vertical del sismo.

D = La carga muerta

S_{DS} = Aceleración espectral de diseño para periodos cortos.

Por lo tanto se tiene que:

$$E_v = 0.2 \cdot 1.2264 \cdot 0.8497$$

$$E_v = 0.2084$$

Nota: Para el cálculo de D (carga muerta) ver 3.10 y para Sds ver (Aceleración espectral de diseño para periodos cortos) ver 3.9

4.6.2. Combinaciones a ETABS.

Una vez obtenido E_v se procede a explicar las permutaciones posibles a realizar para las combinaciones de carga:

$$(1.2 \pm E_v) \cdot D + L \pm SismoX \pm 0.3 \cdot SismoY$$

$$(1.2 \pm E_v) \cdot D \pm SismoX \pm 0.3 \cdot SismoY$$

$$(0.9 \pm E_v) \cdot D \pm SismoX \pm 0.3 \cdot SismoY$$

Como se puede apreciar se tienen 3 ecuaciones con 3 signos permutables entre + y -, lo que nos da 8 posibles combinaciones para cada ecuación, dando un total de 24 posibles combinaciones por sismo principal, dando un total de 96 combinaciones (24 para principal Sx, 24 para principal Sy, 24 para principal Nec x y 24 para principal Nec y), si se suma a estas combinaciones las dos permanentes que no contienen sismo se tiene un total de 98 combinaciones de carga.

Nota: No se consideró ASCE_x y ASCE_y, ya que las aceleraciones mantienen el mismo valor que NEC. (Ministerio de Desarrollo Urbano y Vivienda, 2024)

Con ayuda de un código programado en MatLab (MathWorks, 2024) se pasan las 98 combinaciones a ETABS.



Figura 4.6-2

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

4.7. Primeros resultados.

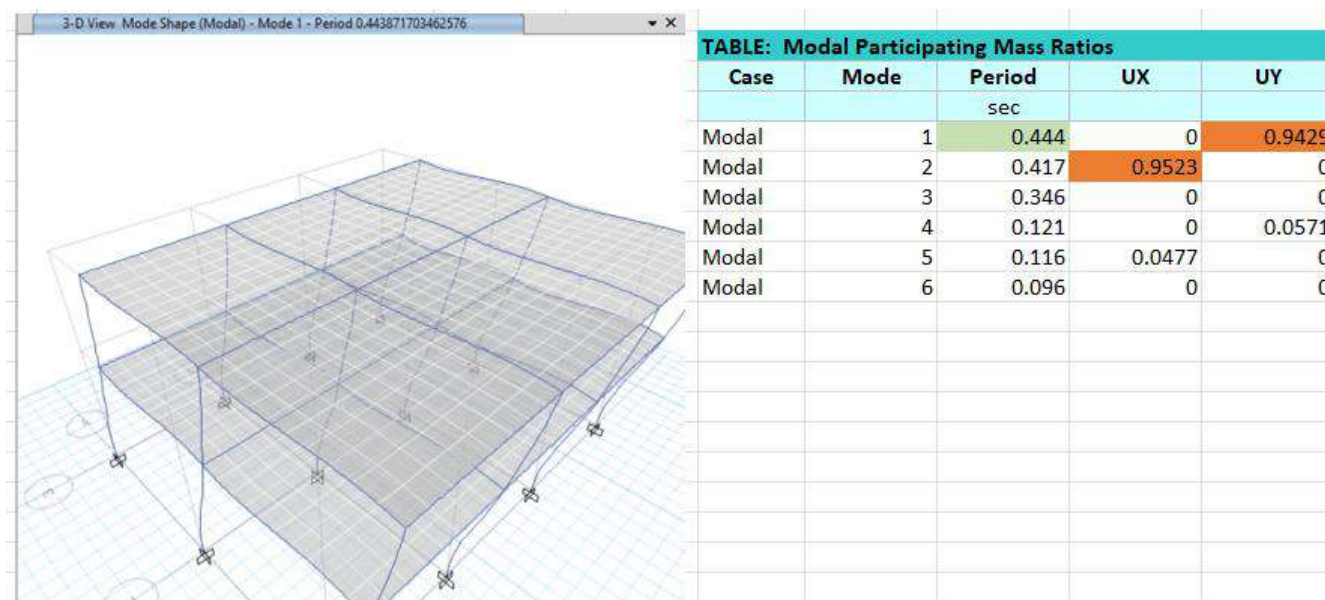


Figura 4.7-1

					PESO EDIFICIO		CORTANTE BASAL	CORTANTE %
Análisis estático	Sismo X	0.05	Story51	Base	0.0526	196.82063	30.1693	15.32832204 %
	Sismo Y	0.05	Story51	Base	0.0526	196.82063	30.1693	15.32832204 %
							CORTANTE BASAL Y	CORTANTE %
	Nec 2024 X					196.82063	28.7464	14.60537953 %
Análisis Modal	Nec 2024 Y					196.82063	28.4805	14.4702819 %
	ASCE 16 X					196.82063	28.75	14.6072086 %
	ASCE 16 Y					196.82063	28.486	14.47307632 %

Figura 4.7-2

Nota: Tabla de datos de autoría propia extraída de software Excel.

4.8. Cortante mínima del ASCE.

En este trabajo se utiliza específicamente 12.8-3

Para T menor o igual a TL

$$C_s = \frac{S_{D1}}{T \left(\frac{R}{I_e} \right)}$$

Nota: ecuación tomada de ASCE 7-16 eq 12.8-3 (American Society of Civil Engineers, 2017)

Reemplazando Tenemos:

$$C_s = \frac{0.7947}{0.444 \left(\frac{8}{1} \right)}$$

$$C_s = 22.37 \%$$

Como el cortante mínimo de la ASCE (American Society of Civil Engineers, 2017) sale mayor a todos los coeficientes basales calculados en los primeros resultados se procede a igualar todos los coeficientes basales a 22.37 %.

4.9. Resultados igualados.

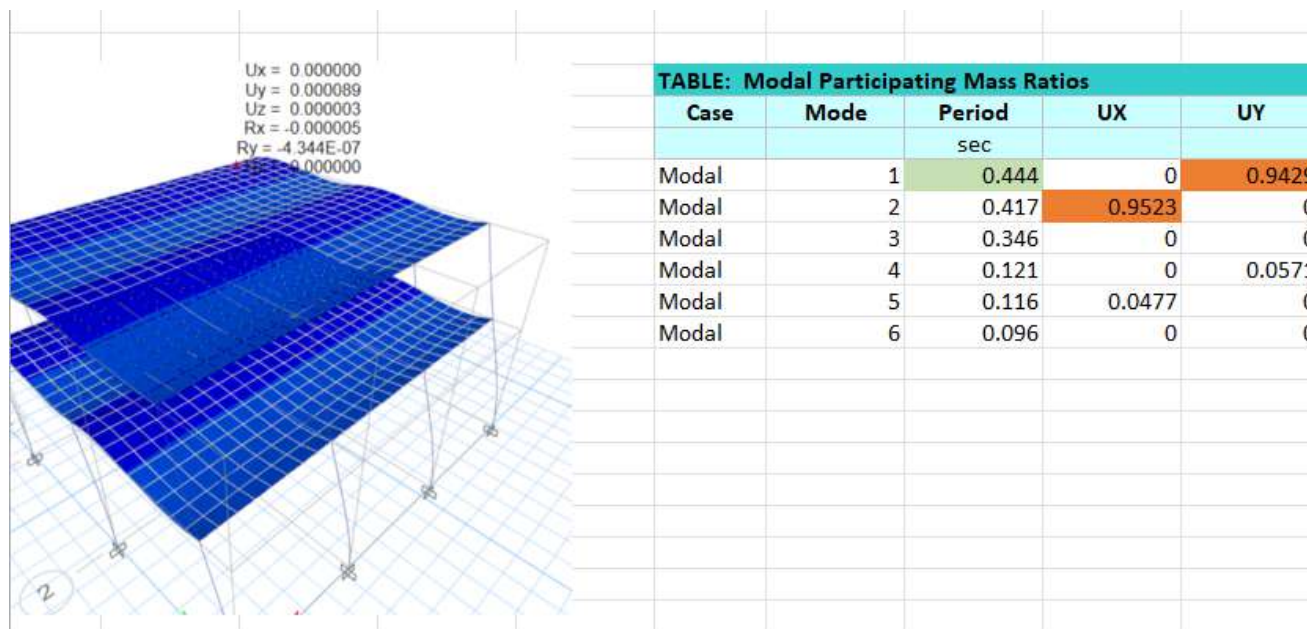


Figura 4.9-1

					PESO EDIFICIO		CORTANTE BASAL	CORTANTE %	
Analysis estatico	Sismo X	0.05	Story51	Base	0.0526	192.66383	44.0288	22.8526548 %	
	Sismo Y	0.05	Story51	Base	0.0526	192.66383	44.0288	22.8526548 %	
							CORTANTE BASAL Y	CORTANTE BASAL X	CORTANTE %
Analysis Modal	Nec 2024 X					192.66383	44.0348	22.85576904 %	
	Nec 2024 Y					192.66383	44.0343	22.85550952 %	
	ASCE 16 X					192.66383	44.0333	22.85499048 %	
	ASCE 16 Y					192.66383	44.0335	22.85509429 %	

Figura 4.9-2

Nota: Tabla de datos de autoría propia extraída de software excel.

4.10. Derivas inelásticas.

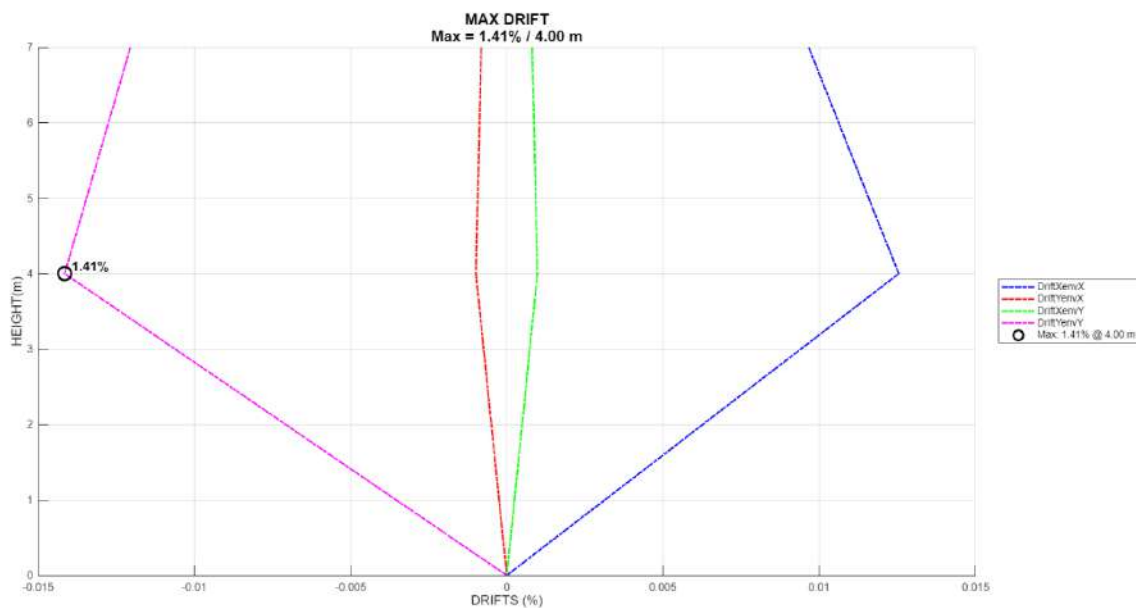


Figura 4.10-1

Nota: Imagen de autoría propia extraída de código en MatLab. (MathWorks, 2024)

4.11. Desplazamientos máximos.

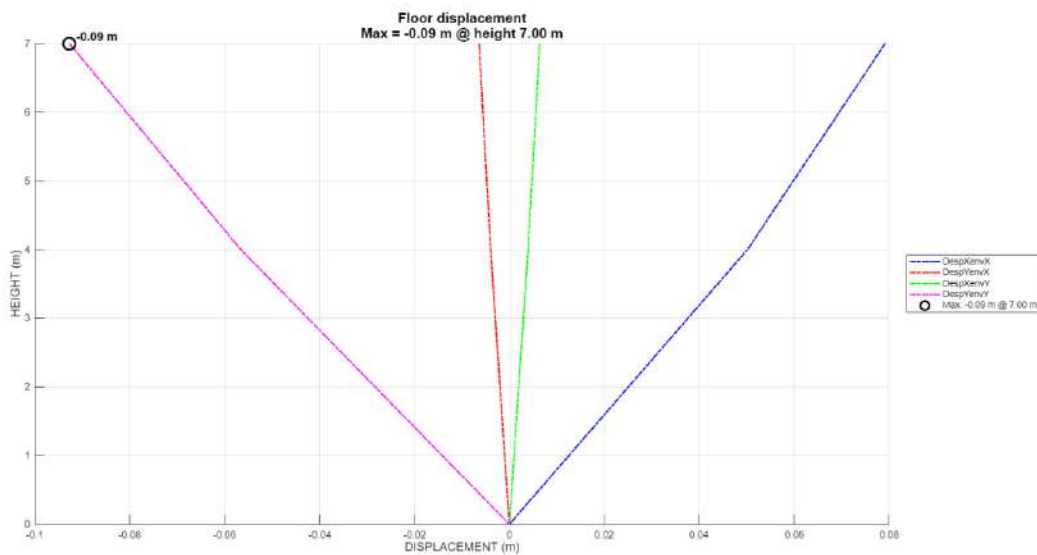


Figura 4.11-1

Nota: Imagen de autoría propia extraída de código en MatLab. (MathWorks, 2024)

4.12. Corte

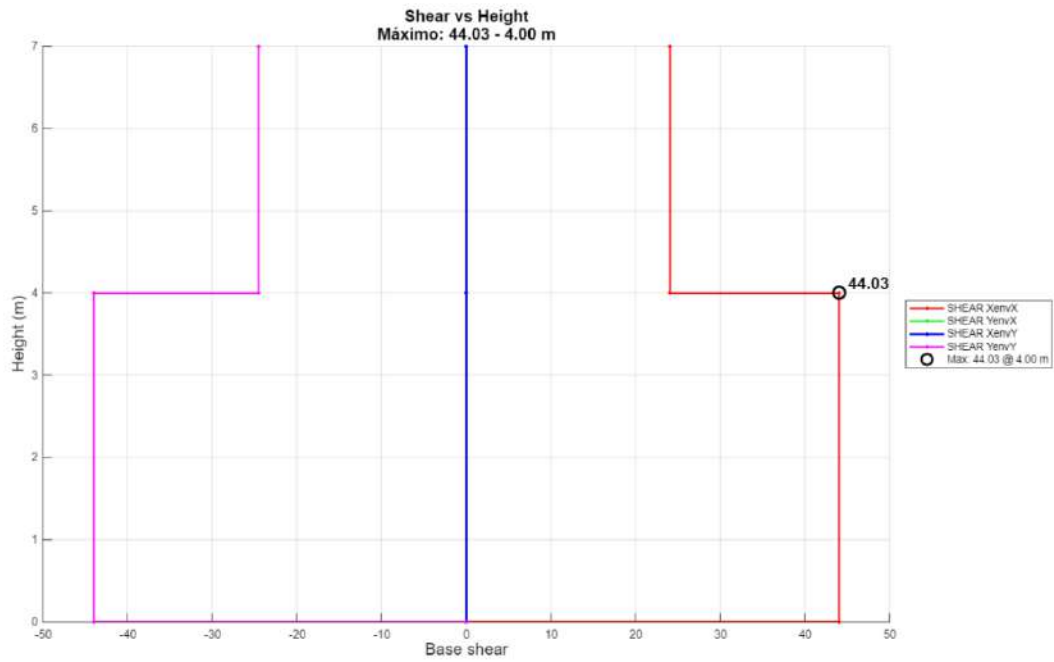


Figura 4.12-1

Nota: Imagen de autoría propia extraída de código en MatLab. (MathWorks, 2024)

4.13. Fuerzas Laterales.

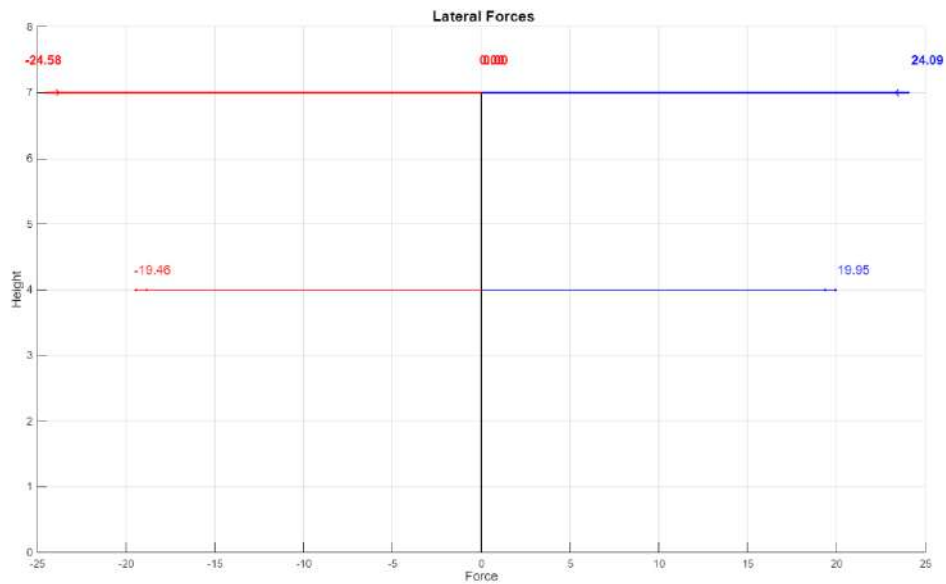


Figura 4.13-1

Nota: Imagen de autoría propia extraída de código en MatLab. (MathWorks, 2024)

4.14. Volcamiento

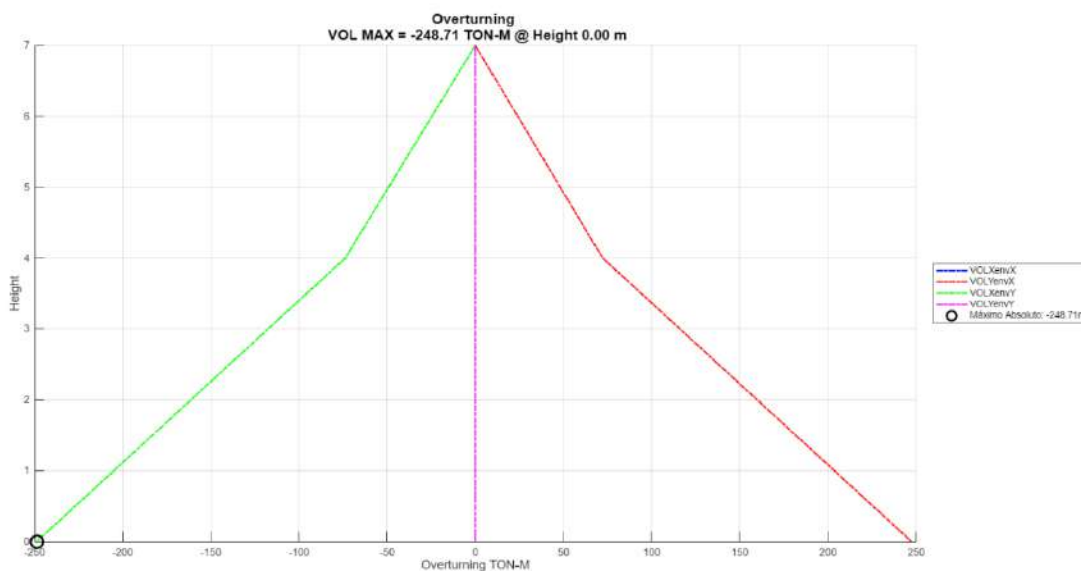


Figura 4.14-1

Nota: Imagen de autoría propia extraída de código en MatLab. (MathWorks, 2024)

4.15. Rigidez por piso.

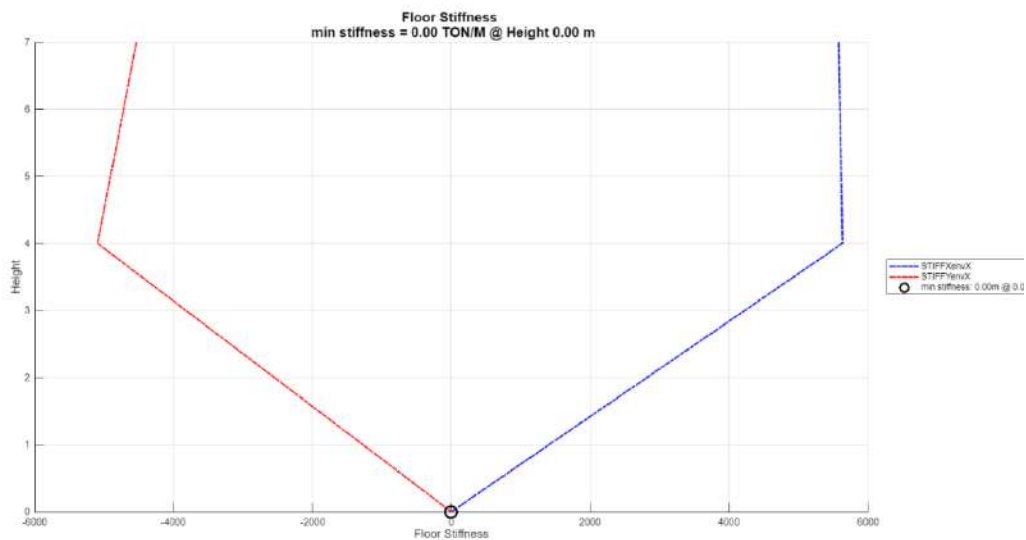


Figura 4.15-1

Nota: Imagen de autoría propia extraída de código en MatLab.

CAPÍTULO V

5. AUTOMATIZACIÓN DEL DISEÑO ESTRUCTURAL DE VIGAS Y COLUMNAS

5.1. Instalación de librerías requeridas para Python.

Para correr el código se debe instalar librerías en el entorno de Python (Python Software Foundation, 2023). Estas permiten el correcto funcionamiento de las operaciones numéricas, manipular datos, conexión con el software comercial. El usuario debe instalar:

```
pip install numpy pandas matplotlib comtypes openpyxl python – docx
```

5.1.1. Instrucciones

Las instrucciones del uso de Python se encuentra el en script.

5.2. Datos requeridos de ingreso al Python.

Es necesario ingresar parámetros geométricos y mecánicos. En dos pasos se ingresa los datos requeridos, en función al elemento estructural y condiciones de diseño.

```

# ----- P A S O 1 -----
# === Parámetros Geométricos "V I G A" ===
b_viga = 25 # cm
h_viga = 40 # cm
recubrimiento = 4 # cm
diam_estribo = 1 # cm

# === ¿Cuántas varillas deseas para armar la "V I G A"? ===
num_varillas = 4 # u

# === Parámetros Geométricos "C O L U M N A" ===
h_columna = 40 # cm
b_columna = 40 # cm

# --- Parámetros Mecánicos ---
f_c = 24 # MPa
f_y = 420 # MPa
lambda_conc = 1.0 # Concreto normal - liviano
psi_e = 1.0 # Recubrimiento
psi_g = 1.0 # Confinamiento
alpha = 1.25 # Coeficiente de momento probable

# --- ¿Cuántas varillas deseas para armar la "C O L U M N A"? ---
opcion1 = 4 # u
opcion2 = 8 # u
opcion3 = 10 # u
opcion4 = 12 # u

# ----- P A S O 2 -----

# Ingresar Diámetro Longitudinal "Elegido en Paso 1"
diametro_long_mm = 18 # mm (diámetro menor de varilla longitudinal)

```

Figura 5.2-1

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

5.3. Extracción de datos desde el software comercial.

Para enlazar Python (Python Software Foundation, 2023) con el software comercial de análisis estructural, se crea una conexión directa con la función Conect2Etabs, la cual permite conectarse a una instancia abierta del software comercial, esto es posible por medio de un script creado en Python (Python Software Foundation, 2023), usando la librería compytes.client. Establecida la conexión se procede a la extracción de tablas y se guarda el documento en formato xlsx.(Python Software Foundation, 2023)

```

# Conexión a software comercial
SapModel, _, _ = api.Connect2Etabs()

# Nombre de la tabla
table_name = "Concrete Beam Design Summary - ACI 318-19"
table = SapModel.DatabaseTables.GetTableForDisplayArray(table_name, GroupName="")

# Procesar datos
cols, noOfRows = table[2], table[3]

if noOfRows > 0:
    df = pd.DataFrame(np.array_split(table[4], noOfRows))
    df.columns = cols

    # Convertir automáticamente columnas numéricas
    df = df.apply(pd.to_numeric, errors='ignore')

    # Mostrar todo sin recortar
    pd.set_option('display.max_rows', None)
    pd.set_option('display.max_columns', None)
    pd.set_option('display.width', None)
    pd.set_option('display.colheader_justify', 'center')
    pd.set_option('display.float_format', '{:.4f}'.format)

    # Guardar en Excel
    df.to_excel("A)tabla_completa_diseno_vigas.xlsx", index=False)
    print("✅ EXTRACCION ETABS COMPLETA: 'tabla_completa_diseno_vigas.xlsx'")

```

Figura 5.3-1.

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Label	UniqueName	DesignSect	Station	Status	sTopCom	AsMinTop	AsTop	sBotCom	AsMinBot	AsBot	VCombo	VRebar	TLngCombo	TLngRebar	
Story2	B1	2 Viga tesis 25x40 cm		200	No Messa	C78		300	314	C98		300	300 C82	0,43	C98	0
Story2	B1	2 Viga tesis 25x40 cm		500	No Messa	C98		300	300	C98		300	300 C82	0,42	C98	0
Story2	B1	2 Viga tesis 25x40 cm		500	No Messa	C98		300	300	C98		300	300 C82	0,37	C98	0
Story2	B1	2 Viga tesis 25x40 cm		1000	No Messa	C98		300	300	C98		300	300 C82	0,36	C98	0
Story2	B1	2 Viga tesis 25x40 cm		1000	No Messa	C98		300	300	C98		300	300 C82	0,32	C98	0
Story2	B1	2 Viga tesis 25x40 cm		1500	No Messa	C98		300	300	C98		300	300 C98	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		1500	No Messa	C98		300	300	C98		300	300 C98	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		2000	No Messa	C98		300	300	C98		300	300 C98	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		2000	No Messa	C98		300	300	C98		300	300 C78	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		2500	No Messa	C98		300	300	C98		300	300 C98	0	C98	0
Story2	B1	2 Viga tesis 25x40 cm		2500	No Messa	C98		300	300	C98		300	300 C86	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		3000	No Messa	C98		300	300	C98		300	300 C94	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		3000	No Messa	C98		300	300	C98		300	300 C98	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		3500	No Messa	C98		300	300	C98		300	300 C98	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		3500	No Messa	C98		300	300	C98		300	300 C98	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		4000	No Messa	C98		300	300	C98		300	300 C98	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		4000	No Messa	C98		300	300	C98		300	300 C98	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		4500	No Messa	C98		300	300	C98		300	300 C98	0,21	C98	0
Story2	B1	2 Viga tesis 25x40 cm		4500	No Messa	C98		300	300	C98		300	300 C2	0,21	C98	0

Figura 5.3-2

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.4. Diseño longitudinal de vigas.

5.4.1. Agrupación de datos.

Se extrae la tabla “Concrete Beam Design Summary – ACI 318-19”, se agrupa las

columnas Story, Label, Station, AsTop, AsBot, y crea el nuevo archivo con datos listos para proceder al cálculo.(American Concrete Institute, 2019)

```
# Cargar archivo Excel
archivo = "A)tabla_completa_diseno_vigas.xlsx"

# Columnas específicas a extraer
columnas_deseadas = ["Story", "Label", "Station", "AsTop", "AsBot"]

# Leer solo esas columnas
datos = pd.read_excel(archivo, usecols=columnas_deseadas)

# Reiniciar el índice por limpieza (opcional)
datos = datos.reset_index(drop=True)

# Guardar archivo nuevo con esas columnas
datos.to_excel("01)vigas_diseño_filtrado_rebar.xlsx", index=False)
print("✅ Archivo Generado: '01)vigas_diseño_filtrado_rebar.xlsx'")
```

Figura 5.4-1

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Label	Station	AsTop	AsBot
Story2	B1	200	314	300
Story2	B1	500	300	300
Story2	B1	500	300	300
Story2	B1	1000	300	300
Story2	B1	1000	300	300
Story2	B1	1500	300	300
Story2	B1	1500	300	300
Story2	B1	2000	300	300
Story2	B1	2000	300	300
Story2	B1	2500	300	300
Story2	B1	2500	300	300
Story2	B1	3000	300	300
Story2	B1	3000	300	300
Story2	B1	3500	300	300
Story2	B1	3500	300	300
Story2	B1	4000	300	300

Figura 5.4-2

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.4.2. Filtrado de datos.

Se carga el documento agrupado, y se filtra los resultados de AsTop y AsBot, en los máximos por estación inicial, estación final, y máximos de las estaciones del centro de la luz, el for hace este procedimiento por cada número de viga y por cada número de piso, agrupa las tres estaciones, y crea el nuevo archivo.

```
# Cargar archivo Excel
archivo = "01)vigas_diseño_filtrado_rebar.xlsx"
df = pd.read_excel(archivo)

# Agrupar por Story y Label
agrupado = df.groupby(["Story", "Label"])

filas_finales = []

for (story, label), grupo in agrupado:
    grupo_ordenado = grupo.sort_values("Station").reset_index(drop=True)

    # Estaciones de inicio y final
    estacion_min = grupo_ordenado.iloc[0]
    estacion_max = grupo_ordenado.iloc[-1]

    # Estaciones del centro (4 valores si existen)
    centro_idx = len(grupo_ordenado) // 2
    inicio = max(0, centro_idx - 2)
    fin = centro_idx + 2
    rango_medio = grupo_ordenado.iloc[inicio:fin]

    # Estación con mayor totTopRebar + totBotRebar (sin crear columna)
    estacion_media_max = rango_medio.loc[
        (rango_medio["AsTop"] + rango_medio["AsBot"]).idxmax()
    ]

    # Añadir las tres estaciones seleccionadas
    filas_finales.extend([estacion_min, estacion_media_max, estacion_max])

# Crear DataFrame final
df_final = pd.DataFrame(filas_finales).reset_index(drop=True)
df_ordenado = df_final.sort_values(by=["Story", "Label", "Station"]).reset_index(drop=True)

# Guardar a archivo Excel FILTRADO A 3 ESTACIONES POR VIGA
df_ordenado.to_excel("02)vigas_inicio_media_max_final.xlsx", index=False)
print("✅ Archivo Generado: '02)vigas_inicio_media_max_final.xlsx'")
```

Figura 5.4-3

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Label	Station	AsTop	AsBot
Story1	B1	200	258	258
Story1	B1	2000	258	258
Story1	B1	4800	258	258
Story1	B10	200	325	263
Story1	B10	1500	258	258
Story1	B10	3800	325	263

Figura 5.4-4

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.4.2.1. Áreas de acero longitudinales.

Se crea una tabla de varillas comerciales de Ecuador con número, diámetro y área, el código selecciona el área de varilla que cubra el AsBot máximo de cada viga, con la cantidad de varillas que el usuario ingresa, el primer for selecciona el AsBot máximo por viga y el segundo

for busca el área mínima de la varilla para cubrir la cantidad de área del AsBot máximo.

```
# === Tabla de varillas comerciales Ecuador (mm²) ===
df_varillas = pd.DataFrame({
    "Numero": [10, 12, 14, 16, 18, 20, 22, 25, 28, 32],
    "Diametro_mm": [10, 12, 14, 16, 18, 20, 22, 25, 28, 32],
    "Area_mm2": [78.54, 113.09, 153.94, 201.06, 254.47, 314.16, 380.13, 490.87, 615.75, 804.25]
})

# === Cargar los datos filtrados ===
df = pd.read_excel("02)vigas_inicio_media_max_final.xlsx")

# Calcular refuerzo base para la cantidad deseada de varillas
grupo = df.groupby(["Story", "Label"])
resultados_base = []

for (story, label), subgrupo in grupo:
    as_bot_max = subgrupo["AsBot"].max()
    mejor_opcion = None

    for _, fila in df_varillas[df_varillas["Numero"] >= 12].iterrows():
        area_total = num_varillas * fila["Area_mm2"]
        if area_total >= as_bot_max:
            mejor_opcion = {
                "Story": story,
                "Label": label,
                "AsBot_max (mm2)": as_bot_max,
                "Varilla #": fila["Numero"],
                "Diametro (mm)": fila["Diametro_mm"],
                "Área/varilla (mm2)": fila["Area_mm2"],
                f"Total {num_varillas} barras (mm2)": area_total,
                "Cantidad de barras": num_varillas
            }
            break

    resultados_base.append(mejor_opcion)

df_base = pd.DataFrame(resultados_base)
df_base.to_excel("03)refuerzo_longitudinal.xlsx", index=False)
print("✅ Archivo Generado: '03)refuerzo_longitudinal.xlsx'")
```

Figura 5.4-5

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Label	pt_max (m)	Varilla #	Diametro (mm)	Área/varilla (mm2)	Total 4 barras (mm2)	Cantidad de barras
Story1	B1	300	12	12	113,09	452,36	4
Story1	B10	300	12	12	113,09	452,36	4
Story1	B11	300	12	12	113,09	452,36	4
Story1	B12	300	12	12	113,09	452,36	4

Figura 5.4-6

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.4.3. Acero de refuerzo para momento negativo.

Se lee los datos filtrados por estaciones inicial y final, para poder calcular el área de varilla necesaria para contrarrestar el momento negativo en el AsTop, se llama al área colocada en las varillas longitudinales y se resta la diferencia que falta para cubrir el AsTop, el for recorre en los valores filtrados de la estación inicial y final por viga y evalúa si necesita un refuerzo adicional, el if indica si no requiere acero adicional, el else indica si, si se necesita acero adicional busca una varilla con un área mayor al déficit.

```

adicionales = []
for _, fila in df_base.iterrows():
    story = fila["Story"]
    label = fila["Label"]
    area_base = fila[f"Total {num_varillas} barras (mm2)"]

    grupo_viga = df[(df["Story"] == story) & (df["Label"] == label)].sort_values("Station")
    extremos = grupo_viga.iloc[[0, -1]] # inicio y final

    for idx, extremo in extremos.iterrows():
        deficit = extremo["AsTop"] - area_base
        if deficit <= 0:
            adicional = {
                "Story": story,
                "Label": label,
                "Station": extremo["Station"],
                "AsTop": extremo["AsTop"],
                "Deficit": 0,
                "Varilla_adicional": "NO REQUIERE",
                "Diametro_varilla_adicional": np.nan,
                "Numero_varilla_adicional": np.nan
            }
        else:
            varilla_extra = df_varillas[df_varillas["Area_mm2"] >= deficit].iloc[0]
            adicional = {
                "Story": story,
                "Label": label,
                "Station": extremo["Station"],
                "AsTop": extremo["AsTop"],
                "Deficit": round(deficit, 2),
                "Varilla_adicional": f"#{varilla_extra['Numero']} ({varilla_extra['Area_mm2']} mm2)",
                "Diametro_varilla_adicional": varilla_extra["Diametro_mm"],
                "Numero_varilla_adicional": varilla_extra["Numero"]
            }
        adicionales.append(adicional)

df_adicionales = pd.DataFrame(adicionales)
df_adicionales.to_excel("04)Refuerzo Adicional.xlsx", index=False)
print("🟢 Archivo Generado: '04)Refuerzo Adicional.xlsx'")

```

Figura 5.4-7

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Label	Station	AsTop	Deficit	Varilla_adicional	Diametro_varilla_adicional	Numero_varilla_adicional
Story1	B1	200	488	35,64	#10.0 (78.54 mm ²)	10	10
Story1	B1	4800	502	49,64	#10.0 (78.54 mm ²)	10	10
Story1	B10	200	384	0	NO REQUIERE		
Story1	B10	3800	384	0	NO REQUIERE		
Story1	B11	200	418	0	NO REQUIERE		
Story1	B11	3800	418	0	NO REQUIERE		

Figura 5.4-8

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.4.4. Longitud de vigas.

Con la estación inicial y la estación final, se calcula la longitud total de la viga, y se calcula la longitud libre de viga. En el código la función “.sort_values” ordena los datos según columnas, piso, etiqueta y estaciones, con. “reset_index(drop=true)” no conserva índices con datos anteriores, el for se realiza el ciclo en cada viga y por grupo de estaciones. Finalmente se crea la columna luz libre, y la columna centro a centro.

```

# Cargar archivo con estaciones inicial, media y final por viga
df = pd.read_excel("02)vigas_inicio_media_max_final.xlsx")
df = pd.read_excel(archivo)

# Asegurar que está ordenado por estación
df_ordenado = df.sort_values(by=["Story", "Label", "Station"]).reset_index(drop=True)

# Agrupar por viga
grupos = df_ordenado.groupby(["Story", "Label"])

# Calcular longitudes por viga
resultados = []
for (story, label), grupo in grupos:
    estacion_inicial = grupo.iloc[0]["Station"]
    estacion_final = grupo.iloc[-1]["Station"]

    luz_libre = estacion_final # de cara a cara de columna
    centro_a_centro = estacion_inicial + estacion_final # desde centro de columna a centro de columna

    resultados.append({
        "Story": story,
        "Label": label,
        "Estación Inicial": estacion_inicial,
        "Estación Final": estacion_final,
        "Luz libre (mm)": round(luz_libre, 3),
        "Centro a centro (mm)": round(centro_a_centro, 3)
    })

# Crear DataFrame con los resultados
df_longitudes = pd.DataFrame(resultados)

df_longitudes.to_excel("05)Longitudes_vigas.xlsx", index=False)
print("✅ Archivo Generado: '05)Longitudes_vigas.xlsx'")

```

Figura 5.4-9

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Label	Estación Inicial	Estación Final	Luz libre (mm)	Centro a centro (mm)
Story1	B1	200	4800	4800	5000
Story1	B10	200	3800	3800	4000
Story1	B11	200	3800	3800	4000
Story1	B12	200	3800	3800	4000
Story1	B13	200	3800	3800	4000
Story1	B14	200	3800	3800	4000
Story1	B15	200	3800	3800	4000

Figura 5.4-10

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.4.5. Longitud de desarrollo de aceros de refuerzo para momento negativo.

En la ACI 318-19 (American Concrete Institute, 2019) figura R9.7.3.2 específica para

colocar la longitud del refuerzo negativo es mayor o igual a la longitud de desarrollo “ld”.

< 20 mm	> 22 mm
$\left(\frac{f_y \Psi_t \Psi_e \Psi_g}{2.1 \lambda \sqrt{f'_c}}\right) d_b$	$\left(\frac{f_y \Psi_t \Psi_e \Psi_g}{1.7 \lambda \sqrt{f'_c}}\right) d_b$

Nota: Tabla 25.4.2.3 extraída de ACI 318-19. (American Concrete Institute, 2019)

Factores de modificación:

f_y = Resistencia a la fluencia del acero.

Ψ_t = Factor por ubicación del refuerzo.

Ψ_e = Factor de recubrimiento.

Ψ_g = Factor de grado del refuerzo.

λ = Factor de concreto peso liviano.

En Python (Python Software Foundation, 2023) el usuario introduce los valores de factores de modificación establecidos por la ACI 318-19 (American Concrete Institute, 2019), la función es aplicada para cada diámetro de acero, con función a su ubicación.

En el código se define la función que llama a todos los factores para el cálculo de “ld”, el for recorre las varillas comerciales del ecuador con sus diámetros, define la ecuación para usar por su el tipo de diámetro, además calcula “ld” para refuerzo interior y superior.

```

# === Función para calcular Ld/db ===
def calcular_ld_db(f_c, f_y, lambda_conc, psi_t, psi_e, psi_g, divisor):
    return (f_y * psi_t * psi_e * psi_g) / (divisor * lambda_conc * np.sqrt(f_c))

# === Cálculo con df_varillas ya cargado ===
datos = []
for _, fila in df_varillas.iterrows():
    numero = fila["Numero"]
    d_b = fila["Diametro_mm"]

    # Elegir divisor según el diámetro
    divisor = 2.1 if d_b <= 19 else 1.7

    # Inferior
    ld_db_inf = int(round(calcular_ld_db(f_c, f_y, lambda_conc, psi_t=1.0, psi_e=psi_e, psi_g=psi_g, divisor=divisor)))
    ld_inf = round(ld_db_inf * d_b, 2)

    # Superior
    ld_db_sup = int(round(calcular_ld_db(f_c, f_y, lambda_conc, psi_t=1.3, psi_e=psi_e, psi_g=psi_g, divisor=divisor)))
    ld_sup = round(ld_db_sup * d_b, 2)

    datos.append({
        "Varilla #": numero,
        "Diámetro (mm)": d_b,
        "Divisor": divisor,
        "Ld/db INF": ld_db_inf,
        "Ld INF (mm)": ld_inf,
        "Ld/db SUP": ld_db_sup,
        "Ld SUP (mm)": ld_sup
    })

# === Resultado en DataFrame ===
df_ld_resultado = pd.DataFrame(datos)
#print(df_ld_resultado)
df_ld_resultado.to_excel("06)longitud_de_desarrollo.xlsx", index=False)
print("📄 Archivo Generado: '06)longitud_de_desarrollo.xlsx'")

```

Figura 5.4-11

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Varilla #	Diámetro (mm)	Divisor	Ld/db INF	Ld INF (mm)	Ld/db SUP	Ld SUP (mm)
10	10	2,1	41	410	53	530
12	12	2,1	41	492	53	636
14	14	2,1	41	574	53	742
16	16	2,1	41	656	53	848
18	18	2,1	41	738	53	954
20	20	1,7	50	1000	66	1320
22	22	1,7	50	1100	66	1452
25	25	1,7	50	1250	66	1650
28	28	1,7	50	1400	66	1848
32	32	1,7	50	1600	66	2112

Figura 5.4-12.

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.4.6. Longitud extra de acero de refuerzo.

Según la ACI 318-19 (American Concrete Institute, 2019), describe que el refuerzo se debe extender más allá del punto en el que ya no es necesario para resistir flexión, y se establece tres condiciones.

$$L_2 = \begin{cases} d \\ 12d_b \\ \frac{ln}{16} \end{cases}$$

El código carga los datos necesarios para calcular las tres condiciones, se elige el mayor diámetro de varillas longitudinales y de varillas de refuerzo con “.merge()” la cual une los dos dataframe, los mismo se realiza para longitudes, pisos y etiquetas. La función “def buscar_ld(row)” devuelve el mayor de los tres.

```
# === Cargar archivos de entrada ===
df_base = pd.read_excel("03)refuerzo_longitudinal.xlsx")
df_adicionales = pd.read_excel("04)Refuerzo_Adicional.xlsx")
df_longitudes = pd.read_excel("05)Longitudes_vigas.xlsx")
df_id = pd.read_excel("06)longitud_de_desarrollo.xlsx")

# Refuerzo base: 4 varillas
# Refuerzo adicional en extremos
# Longitudes: Luz Libre y Longitud Total
# Longitud de desarrollo por varilla

# === Asegurar que la columna 'Diametro_varilla_adicional' sea numérica ===
df_adicionales["Diametro_varilla_adicional"] = pd.to_numeric(df_adicionales["Diametro_varilla_adicional"], errors="coerce")

# === Combinar datos ===
# Tomar el mayor diámetro entre las varillas base y adicionales por viga
df_comb = df_base.merge(
    df_adicionales.groupby(["Story", "Label"])["Diametro_varilla_adicional"].max().reset_index(),
    on=["Story", "Label"],
    how="left"
)

df_comb = df_comb.merge(df_longitudes, on=["Story", "Label"], how="left")

# === Función para calcular L2 ===
def calcular_L2(row):
    if pd.isna(row["Diametro_varilla_adicional"]):
        return 0 # No hay refuerzo adicional + L2 = 0

    db_base = row["Diametro (mm)"]
    db_adic = row["Diametro_varilla_adicional"]
    db_max = max(db_base, db_adic if not pd.isna(db_adic) else db_base)
    doce_db = 12 * db_max
    ln_16 = row["Luz libre (mm)"] / 16
    return max(d, doce_db, ln_16)

# === Calcular L2 por fila ===
df_comb["L2 (mm)"] = df_comb.apply(calcular_L2, axis=1)
```

Figura 5.4-13

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

5.4.7. Longitud de acero para momentos negativos total.

La longitud total del acero para momento negativo es una sumatoria de la longitud de desarrollo con la longitud extra. La función “buscar_ld(row)” obtiene el valor que se calcula de “ld” solo si la viga tiene refuerzo adicional, el if realiza si no hay varilla adicional, retorna un valor de 0.

```
# Cargar longitudes de desarrollo
df_ld = pd.read_excel("06)longitud_de_desarrollo.xlsx")

# Limpiar espacios invisibles por si acaso
df_ld.columns = df_ld.columns.str.strip()

# === Agregar Ld SUP solo si tiene refuerzo adicional ===
def buscar_ld(row):
    if pd.isna(row["Diámetro_varilla_adicional"]):
        return 0
    fila = df_ld[df_ld["Diámetro (mm)"] == row["Diámetro_varilla_adicional"]]
    if not fila.empty:
        return fila.iloc[0]["Ld SUP (mm)"] # convertir a cm
    return 0

df_comb["Ld (mm)"] = df_comb.apply(buscar_ld, axis=1)

df_comb["Ld + L2 (mm)"] = np.ceil((df_comb["Ld (mm)"] + df_comb["L2 (mm)"]) / 5) * 5

# === Exportar resultado ===
archivo_salida = "07)longitudes_varilla_LD+L2.xlsx"
df_comb.to_excel(archivo_salida, index=False)
print("✅ Archivo Generado: '07)longitudes_varilla_LD+L2.xlsx'")
```

Figura 5.4-14

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Label	AsBot_max (mm2)	Varilla #	Diámetro (mm)	Área/varilla (mm2)	4 barras	cantidad de b	varilla	posición	Inic	posición	Fin	libre (m)	a centro	L2 (mm)	Ld (mm)	Ld + L2 (mm)
Story1	B1	300	12	12	113,09	452,36	4	10	200	4800	4800	5000	300	530	830		
Story1	B10	300	12	12	113,09	452,36	4		200	3800	3800	4000	0	0	0		
Story1	B11	300	12	12	113,09	452,36	4		200	3800	3800	4000	0	0	0		
Story1	B12	300	12	12	113,09	452,36	4	12	200	3800	3800	4000	237,5	636	875		

Figura 5.4-15

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.4.8. Comprobación de áreas de acero en vigas.

Comprobación manual de área de acero de una viga. Citando a (Guerra Avendaño & Guerra Valladares, 2020), se usa las siguientes ecuaciones para determinar el área de acero requerida para

la sección transversal prevista.

$$k = \frac{0.85 \cdot f'_c \cdot b \cdot d}{f_y}$$

$$A_s = k \cdot \left(1 - \sqrt{1 - \frac{2 \cdot M_u}{\phi \cdot k \cdot d \cdot f_y}} \right)$$

El acero mínimo para flexión en vigas se especifica en la ACI 318-19 (American Concrete Institute, 2019) en el capítulo 9.6.1.2.

$$(a) A_{s,\min} = \frac{0.8\sqrt{f'_c}}{f_y} \cdot b \cdot d$$

$$(b) A_{s,\min} = \frac{14}{f_y} \cdot b \cdot d$$

Se realiza el cálculo manual para área de acero "As":

V I G A "1" - P I S O "1"						
h cm	d cm	b cm	f'c kg/cm2	f'y kg/cm2	k	As cm2
35	31	25	210	4200	183600	4.97

Figura 5.4-16

Nota: Tabla de datos de autoría propia extraída de software Excel.

Área de acero calculada en software comercial:

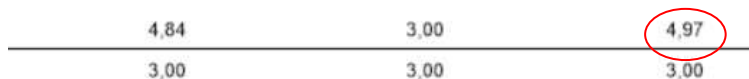


Figura 5.4-17

Nota: Imagen de autoría propia extraída de software ETABS. (Computers and Structures, 2023)

5.5. Diseño transversal de vigas.

5.5.1. Agrupación de datos.

Se extrae la tabla “Element Forces - Beams”, se agrupa las columnas Story, Beam, y se identifica los valores máximos de cortante en los extremos de las vigas es decir en la estación inicial y final de cada viga, además se calcula un cortante promedio para el siguiente paso.

La función “def resumenV2(grupo)”, permite encontrar los valores máximos positivos y negativos referente a la agrupación Story, y Beam. Además, con “abs” se convierte a valor absoluto ya que nos interesa la magnitud de cada valor. Finalmente, el “if pd.notnull” permite el promedio de ambos valores.

```

# Extraer solo columnas 1, 2, 3, 9 y 11 + indices 0,1,2,8,10
df_filtrado = df.iloc[:, [0, 1, 2, 8, 10]].copy()
df_filtrado.columns = ['Story', 'Beam', 'UniqueName', 'Station', 'V2']

# Asegurar que V2 sea numérico
df_filtrado['V2'] = pd.to_numeric(df_filtrado['V2'], errors='coerce')

# Agrupar por piso y viga, calcular máximos y promedio (valor absoluto del negativo)
def resumen_v2(grupo):
    max_pos = grupo[grupo['V2'] > 0]['V2'].max()
    max_neg = grupo[grupo['V2'] < 0]['V2'].min()
    max_neg_abs = abs(max_neg) if pd.notnull(max_neg) else None
    promedio = None
    if pd.notnull(max_pos) and pd.notnull(max_neg_abs):
        promedio = (max_pos + max_neg_abs) / 2
    return pd.Series({
        'V2_max_positivo': max_pos,
        'V2_max_negativo': max_neg_abs,
        'V2_promedio': promedio
    })

# Aplicar agrupamiento y función
resumen_v2_final = df_filtrado.groupby(['Story', 'Beam']).apply(resumen_v2).reset_index()

# Reordenar columnas
resumen_v2_final = resumen_v2_final[['Story', 'Beam', 'V2_max_positivo', 'V2_promedio', 'V2_max_negativo']]

# Exportar a Excel si deseas
resumen_v2_final.to_excel("0)Vv_max_promedio_negativo_vigas.xlsx", index=False)
print("✅ Archivo Generado: '0)Vv_max_promedio_negativo_vigas.xlsx' guardado correctamente.")

```

Figura 5.5-1

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Beam	V2_max_positivo	V2_promedio	V2_max_negativo
Story1	B1	6,426830729	6,490677656	6,554524583
Story1	B10	5,587697943	5,587697943	5,587697943
Story1	B11	6,170043995	6,147197128	6,124350261
Story1	B12	9,962299686	9,957872663	9,95344564

Figura 5.5-2

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.5.2. Cortante sísmico.

Para calcular el diseño por capacidad, la ACI 318-19 (American Concrete Institute, 2019) considera la formación de rótulas plásticas en los extremos de la viga, en un evento sísmico establecido en el capítulo 18.7.6.1.1, la fuerza en los nudos se determina usando resistencias a flexión máximas probables.

$$M_{pr} = \alpha \cdot f_y \cdot A_s \cdot \left(d - \alpha \frac{f_y \cdot A_s}{1.7 \cdot f'_c \cdot b} \right)$$

$$V_{sismico} = \frac{M_{pr1} + M_{pr2}}{ln}$$

```

# CARGA DE ARCHIVOS
df_base = pd.read_excel("03)refuerzo_longitudinal.xlsx")
df_adic_raw = pd.read_excel("04)Refuerzo_Adicional.xlsx")
df_longitudes = pd.read_excel("05)Longitudes_vigas.xlsx")

# 1. CALCULAR As_base_total_cm2
df_base['As_base_total_cm2'] = df_base['Total (num_varillas) barras (mm2)'] / 100 # mm² a cm²
df_base = df_base[['Story', 'Label', 'As_base_total_cm2']]

# 2. CALCULAR As_adicional EN CM2 (una sola varilla por fila)
df_adic_raw['As_adic_cm2'] = (
    (np.pi * (df_adic_raw['Diametro_varilla_adicional'] ** 2) / 4)
) / 100 # mm² a cm²

# Separar por extremos usando Station
df_izq = df_adic_raw.sort_values('Station').groupby(['Story', 'Label']).first().reset_index()
df_der = df_adic_raw.sort_values('Station').groupby(['Story', 'Label']).last().reset_index()

# Renombrar columnas
df_izq = df_izq[['Story', 'Label', 'As_adic_cm2']].rename(columns={'As_adic_cm2': 'As_adic_izq_cm2'})
df_der = df_der[['Story', 'Label', 'As_adic_cm2']].rename(columns={'As_adic_cm2': 'As_adic_der_cm2'})

# 3. UNIÓN Y SUMA DE ACEROS
df = df_base.merge(df_izq, on=['Story', 'Label'], how='left')
df = df.merge(df_der, on=['Story', 'Label'], how='left')

# Reemplazar NaN (si no hay refuerzo adicional)
df['As_adic_izq_cm2'] = df['As_adic_izq_cm2'].fillna(0)
df['As_adic_der_cm2'] = df['As_adic_der_cm2'].fillna(0)

# Calcular total de acero por extremo
df['As_total_izq_cm2'] = df['As_base_total_cm2'] + df['As_adic_izq_cm2']
df['As_total_der_cm2'] = df['As_base_total_cm2'] + df['As_adic_der_cm2']

# 4. CÁLCULO DE MOMENTOS PROBABLES
def calcular_mpr(As_cm2):
    I = alpha * f_y * 10 * As_cm2
    Mpr = I * (d - (I / (1.7 * f_c * 10 * b_viga))) # kg·cm
    return Mpr

df['Mpr_izq_kgcm'] = df['As_total_izq_cm2'].apply(calcular_mpr)
df['Mpr_der_kgcm'] = df['As_total_der_cm2'].apply(calcular_mpr)

# Convertir a tonf·m
df['Mpr_izq_tonfm'] = df['Mpr_izq_kgcm'] * 1e-5
df['Mpr_der_tonfm'] = df['Mpr_der_kgcm'] * 1e-5

# 5. LUZ LIBRE Y CALCULO DE Vs
df_longitudes['luz_libre_m'] = df_longitudes['luz libre (mm)'] / 1000
df_luz = df_longitudes[['Story', 'Label', 'luz_libre_m']]

df_vs = df.merge(df_luz, on=['Story', 'Label'], how='left')
df_vs['Vs_tonfm'] = (df_vs['Mpr_izq_tonfm'] + df_vs['Mpr_der_tonfm']) / df_vs['luz_libre_m']

# 6. RESULTADO FINAL
df_resultado = df_vs[['Story', 'Label', 'As_total_izq_cm2', 'As_total_der_cm2',
                    'Mpr_izq_tonfm', 'Mpr_der_tonfm', 'luz_libre_m', 'Vs_tonfm']]

# Exportar a Excel si deseas
df_resultado.to_excel("1)resultado_cortante_sismico_vs.xlsx", index=False)
print("📄 Archivo Generado: '1)resultado_cortante_sismico_vs.xlsx'")

```

Figura 5.5-3

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Label	As_total_izq_cm2	As_total_der_cm2	Mpr_izq_tonfm	Mpr_der_tonfm	luz_libre_m	Vs_tonf
Story1	B1	5,308998163	5,308998163	9,27237733	9,27237733	4,8	3,863490554
Story1	B10	4,5236	4,5236	7,996652773	7,996652773	3,8	4,208764618
Story1	B11	4,5236	4,5236	7,996652773	7,996652773	3,8	4,208764618
Story1	B12	5,308998163	5,654573355	9,27237733	9,823134933	3,8	5,025134806

Figura 5.5-4

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.5.3. Cortante provisto por el concreto.

Se determina el cortante que aporta el concreto descrito en la ACI 318-19 (American Concrete Institute, 2019) en el capítulo 22.5.5.1 para miembros no preesforzadas, además cuando se calcula el cortante provisto por el concreto “ V_c ”, las cargas axiales no se consideran para el cálculo por que puede causar un valor negativo. Finalmente se calcula las dos ecuaciones y se elige la mayor.

$$V_{c(a)} = \left(0.53 \cdot \lambda \cdot \sqrt{f'_c} + \frac{N_u}{6 \cdot A_g} \right) \cdot b_w \cdot d$$

$$V_{c(b)} = \left(2.1 \cdot p_w^{\frac{1}{3}} \cdot \sqrt{f'_c} + \frac{N_u}{6 \cdot A_g} \right) \cdot b_w \cdot d$$

$$p_w = \frac{A_s}{b_w \cdot d}$$

$$\lambda = 1.0 \text{ Hormigón de peso normal } 2.4 \frac{\text{Tonf}}{\text{m}^3}$$

El script identifica el valor mayor del acero longitudinal por medio del “df_vc=.max(axis=1)” esto devuelve el valor máximo por fila en un dataframe, posterior el código calcula los cortantes a y b y elige el valor mayor.

```

# 1. CALCULAR As_mayor (el mayor entre izq y der)
df_vs['As_mayor_cm2'] = df_vs[['As_total_izq_cm2', 'As_total_der_cm2']].max(axis=1)

# 2. CALCULAR pw^(1/3)
df_vs['pw_1_3'] = df_vs['As_mayor_cm2'] / (b_viga * d)

# 3. CALCULAR Vc(a)
df_vs['Vc_a_kgf'] = 0.53 * lambda_conc * np.sqrt(f_c*10) * b_viga * d
df_vs['Vc_a_tonf'] = df_vs['Vc_a_kgf'] / 1000

# 4. CALCULAR Vc(b)
df_vs['Vc_b_kgf'] = 2.1 * lambda_conc * (df_vs['pw_1_3'] ** (1/3)) * np.sqrt(f_c*10) * b_viga * d
df_vs['Vc_b_tonf'] = df_vs['Vc_b_kgf'] / 1000

# 5. CREAR COLUMNA FINAL Vc = max(Vc_a, Vc_b)
df_vs['Vc_max_tonf'] = df_vs[['Vc_a_tonf', 'Vc_b_tonf']].max(axis=1)

# 6. RESULTADO FINAL
resultado_vc = df_vs[['Story', 'Label', 'As_mayor_cm2', 'pw_1_3', 'Vc_a_tonf', 'Vc_b_tonf', 'Vc_max_tonf']]

# Exportar a Excel si deseas
resultado_vc.to_excel("2)resultado_Vc.xlsx", index=False)
print("📄 Archivo Generado '2)resultado_Vc.xlsx'")

```

Figura 5.5-5

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Label	As_mayor_cm2	pw_1_3	Vc_a_tonf	Vc_b_tonf	Vc_max_tonf
Story1	B1	5,308998163	0,005898887	7,389652225	5,290427659	7,389652225
Story1	B10	4,5236	0,005026222	7,389652225	5,015504838	7,389652225
Story1	B11	4,5236	0,005026222	7,389652225	5,015504838	7,389652225
Story1	B12	5,654573355	0,006282859	7,389652225	5,402812169	7,389652225

Figura 5.5-6

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.5.4. Espaciamiento entre estribos en zona de confinamiento y en zona media.

La normativa ACI 318-19 (American Concrete Institute, 2019) detalla la longitud de la zona de confinamiento de 2h. En el capítulo 18.6.4.4, la norma indica los respectivos espaciamientos entre estribos, menciona:

Zona de confinamiento: *El valor menor de* $\left((a) = \frac{d}{4} ; (b) = 150mm ; (c) 6d_b \right)$

Zona media: $(a) = \frac{d}{2} ; (b) \text{ máximo } 35cm$

El script realiza el cálculo con las condiciones de la normativa, se comienza en busca del

dataframe que corresponde a cada viga por piso, para buscar el diámetro menor de la varilla longitudinal, en “diam_mm” se calcula las 3 opciones y se toma la menor.

```
# Cargar la tabla con los 3 valores por viga
df_vv_long = pd.read_excel("2.1)Valores_ordenados.xlsx")

# Cargar la tabla de refuerzo longitudinal para obtener el diámetro
df_refuerzo = pd.read_excel("03)refuerzo_longitudinal.xlsx")

# Ajustamos nombres para trabajar más fácilmente
df_refuerzo = df_refuerzo.rename(columns={
    df_refuerzo.columns[0]: "Story",
    df_refuerzo.columns[1]: "Label",
    df_refuerzo.columns[4]: "Diametro_mm"
})

df_refuerzo["Diametro_mm"] = pd.to_numeric(df_refuerzo["Diametro_mm"], errors="coerce")

# ZONA MEDIA: se asigna solo a tipo 'promedio'
S_media_valor = math.floor(min(d / 2, 3S))

# Asegúrate que exista una columna que indique el tipo (tú podrías renombrarla si es diferente)
# Por ejemplo, si la columna se llama 'Tipo' y tiene 'positivo', 'promedio', 'negativo':
df_vv_long["S_media (cm)"] = df_vv_long["Tipo"].apply(
    lambda tipo: S_media_valor if tipo == "promedio" else None
)

# ZONA DE CONFINAMIENTO: se asigna a extremos (positivo y negativo)

def calcular_s_confinamiento(story, label):
    fila = df_refuerzo[(df_refuerzo["Story"] == story) & (df_refuerzo["Label"] == label)]
    if not fila.empty:
        diam_mm = fila["Diametro_mm"].values[0]
        opciones = [d / 4, 6 * (diam_mm / 18), 1S]
        return math.floor(min(opciones))
    return None

df_vv_long["S_confinamiento (cm)"] = df_vv_long.apply(
    lambda row: calcular_s_confinamiento(row["Story"], row["Beam"]) if row["Tipo"] in ["positivo", "negativo"] else None,
    axis=1
)

# Crear columna final unificada
df_vv_long["S_final (cm)"] = df_vv_long.apply(
    lambda row: row["S_media (cm)"] if row["Tipo"] == "promedio" else row["S_confinamiento (cm)"],
    axis=1
)

# Exportar
df_vv_long.to_excel("3)Valores_con_S_final.xlsx", index=False)
print("📄 Archivo generado '3)Valores_con_S_final.xlsx'")
```

Figura 5.5-7

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

5.5.5. Cortante provisto por el acero.

La resistencia al corte provisto por el acero está establecida en la ACI318-19 (American Concrete Institute, 2019) capítulo 22.5.8.5.3, corresponde la resistencia que aporta el estribo para contrarrestar el cortante.

$$V_s = \frac{Av \cdot f'_y \cdot d}{S}$$

S = separación en zona de confinamiento o zona media.

El script calcula en función de las separaciones en zona de confinamiento y en zona media, asigna tipos para cada fila, la función “apply()” recorre por cada fila con la ayuda de “axis=1” realiza la operación fila a fila, la función anónima “lambda” crea en una sola línea un condicional que permite la selección del espaciamiento adecuado según el tipo de zona.

```
# Cargar archivo con separaciones ya calculadas
df = pd.read_excel("3)Valores_con_S_final.xlsx")

# Añadir columna Tipo manualmente (cada viga tiene 3 filas: positivo, promedio, negativo)
tipos = ["positivo", "promedio", "negativo"]
df["Tipo"] = tipos * (len(df) // 3)

# Función para calcular Vs en Tonf
def calcular_vs(s):
    if pd.notna(s) and s != 0:
        return ((Av * f_y*10 * d) / s) / 1000 # Resultado en Tonf
    return None

# Aplicar fórmula según el tipo de zona
df["Vs_acero (Tonf)"] = df.apply(
    lambda row: calcular_vs(row["S_confinamiento (cm)"]) if row["Tipo"] in ["positivo", "negativo"]
    else calcular_vs(row["S_media (cm)"]) if row["Tipo"] == "promedio"
    else None,
    axis=1
)

# Exportar si deseas
df.to_excel("4)Vs_acero_resultado.xlsx", index=False)
print("📄 Archivo generado '4)Vs_acero_resultado.xlsx'")
```

Figura 5.5-8

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Beam	Vv (Tonf)	Vsismo (Tonf)	Vc (Tonf)	Tipo	S_media (cm)	S_confinamiento (cm)	S_final (cm)	Vs_acero (Tonf)
Story1	B1	6,426830729	3,863490554	7,389652225	positivo		7	7	33,92920066
Story1	B1	6,490677656	0	0	promedio	18		18	13,19468915
Story1	B1	6,554524583	3,863490554	7,389652225	negativo		7	7	33,92920066
Story1	B10	5,587697943	4,208764618	7,389652225	positivo		7	7	33,92920066
Story1	B10	5,587697943	0	0	promedio	18		18	13,19468915
Story1	B10	5,587697943	4,208764618	7,389652225	negativo		7	7	33,92920066

Figura 5.5-9

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.5.6. Condición de control.

La norma ACI 318-19 (American Concrete Institute, 2019) , establece en el capítulo 22.5.1.2, verificar que el cortante actuante no supere el límite de cuatro veces el cortante del concreto. El mal uso de este aparato provoca agotamiento, falla frágil.

$$Vv \leq 4Vc$$

El script define la función evaluar_condicion con “row” para presenta filas del dataframe el contiene valores de cortante actuante y de cortante provisto por el concreto. Se crea la condición, y se usa “apply()” con “axis=1” para llamar la función fila por fila

```
# Cargar el archivo
df = pd.read_excel("4)Vvs_acero_resultado.xlsx")

# Evaluar condición con manejo de Vc = 0
def evaluar_condicion(row):
    Vv = row["Vv (Tonf)"]
    Vc = row["Vc (Tonf)"]
    if Vc == 0 or pd.isna(Vc):
        return "ok"
    return "ok" if Vv <= 4 * Vc else "no cumple"

df["Cumple Vv ≤ 4Vc"] = df.apply(evaluar_condicion, axis=1)

# Exportar resultados
df.to_excel("5)Vvs_revisión_4Vc.xlsx", index=False)
print("✅ Archivo generado '5)Vvs_revisión_4Vc.xlsx'")
```

Figura 5.5-10

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Beam	Vv (Tonf)	Vsismo (Tonf)	Vc (Tonf)	Tipo	S_media (cm)	S_confinamiento (cm)	S_final (cm)	Vs_acero (Tonf)	Cumple Vv ≤ 4Vc
Story1	B1	6,426831	3,863490554	7,389652	positivo			7	7	33,92920066 ok
Story1	B1	6,490678	0	0	promedio	18		18	13,19468915	ok
Story1	B1	6,554525	3,863490554	7,389652	negativo			7	7	33,92920066 ok
Story1	B10	5,587698	4,208764618	7,389652	positivo			7	7	33,92920066 ok
Story1	B10	5,587698	0	0	promedio	18		18	13,19468915	ok

Figura 5.5-11

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.5.7. Acero mínimo transversal.

Colocar siempre el área de acero mínimo transversal mejora el comportamiento ante un sismo, evita grietas además beneficia a la ductilidad del elemento. La ACI 318-19 (American Concrete Institute, 2019) las fórmulas se encuentran en el capítulo 9.6.3.4 para elementos no preesforzados, con elementos de torsión despreciados.

$$A_{v_{min}} \begin{cases} \frac{0.2 \cdot \sqrt{f'c}}{f_y} \cdot b \cdot S \\ \frac{3.5 \cdot b \cdot S}{f'y} \end{cases}$$

El código define la función que recibe los espaciamientos por zonas con el “(s, tipo):”, se validan datos, si no existen datos no procesa, además con “return max()” elije el mayor de las dos ecuaciones. Finalmente se aplica la función “apply()” para que se aplique en cada fila.

```
# Fórmulas generales
def calcular_av_min(s, tipo):
    if pd.isna(s):
        return None
    av_1 = (0.2 * np.sqrt(f_c*10) / f_y*10) * b_viga * s
    av_2 = (3.5 * b_viga * s) / f_y*10
    return max(av_1, av_2)

# Aplicar por fila
df["Av_min_conf (cm²)"] = df["S_confinamiento (cm)"].apply(lambda s: calcular_av_min(s, "conf"))
df["Av_min_medio (cm²)"] = df["S_medio (cm)"].apply(lambda s: calcular_av_min(s, "medio"))

# Exportar resultados
df.to_excel("6)Av_min_resultado.xlsx", index=False)
print("✅ Archivo generado '6)Av_min_resultado.xlsx'")
```

Figura 5.5-12

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Beam	Vv (Tonf)	Vsismo (Tonf)	Vc (Tonf)	Tipo	S_media (cm)	S_confinamiento (cm)	S_final (cm)	Vs_acero (Tonf)	Cumple Vv ≤ 4Vc	Av_min_conf (cm²)	Av_min_medio (cm²)
Story1	B1	6,426830729	3,863490554	7,389652	positivo			7	33,92920066	ok	14,58333333	
Story1	B1	6,490677656		0	promedio	18		18	13,19468915	ok		37,5
Story1	B1	6,554524583	3,863490554	7,389652	negativo			7	33,92920066	ok	14,58333333	
Story1	B10	5,587697943	4,208764618	7,389652	positivo			7	33,92920066	ok	14,58333333	
Story1	B10	5,587697943		0	promedio	18		18	13,19468915	ok		37,5
Story1	B10	5,587697943	4,208764618	7,389652	negativo			7	33,92920066	ok	14,58333333	

Figura 5.5-13

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.5.8. Cortante último.

El procedimiento aplicado es un enfoque conservador para el diseño por capacidad se debe considerar las fuerzas generadas en los momentos probables, con los cortantes actuantes.

$$Vu = V_{análisis} + V_{sismo}$$

La “función asignar_vs_por_tipo” recorre por el dataframe de piso, viga y tipo, con el “==” busca los valores correspondientes a cada piso y viga, además si no hay valor de corte sísmico en zona media retorna como cero, para sumar los cortantes.

```
# Primero, cargamos los valores de Vs_tonf
vsismo_df = pd.read_excel("1)resultado_cortante_sismico_vs.xlsx")
vsismo_df = vsismo_df[["Story", "Label", "Vs_tonf"]]

# Volvemos a cargar la tabla donde tenemos las 3 filas por viga
df = pd.read_excel("6)Av_min_resultado.xlsx")

# Función corregida
def asignar_vs_por_tipo(row):
    story = row["Story"]
    label = row["Beam"]
    tipo = row["Tipo"]

    vs_row = vsismo_df[(vsismo_df["Story"] == story) & (vsismo_df["Label"] == label)]

    if vs_row.empty:
        return 0

    vs = vs_row["Vs_tonf"].values[0]

    return 0 if tipo == "promedio" else vs

# Aplicar la función corregida
df["Vsismo (Tonf)"] = df.apply(asignar_vs_por_tipo, axis=1)

# Calcular Vu (Tonf)
df["Vu (Tonf)"] = df["Vv (Tonf)"] + df["Vsismo (Tonf)"]

# Exportar a Excel
df.to_excel("7)Vu_resultado.xlsx", index=False)
print("✅ Archivo generado '7)Vu_resultado.xlsx'")
```

Figura 5.5-14

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Beam	Vv (Tonf)	Vismo (Tonf)	Vc (Tonf)	Tipo	S_media (cm)	S_confinamiento (cm)	S_final (cm)	Vs_acero (Tonf)	Cumple Vv ≤ 4Vc	Av_min_conf (cm²)	in_medio	Vu (Tonf)
Story1	B1	6,426831	3,863491	7,389652	positivo			7	7	33,92920066	ok	14,58333333	10,29032128
Story1	B1	6,490678	0	0	promedio	18		18	13,19468915	ok		37,5	6,490677656
Story1	B1	6,554525	3,863491	7,389652	negativo			7	7	33,92920066	ok	14,58333333	10,41801514
Story1	B10	5,587698	4,208765	7,389652	positivo			7	7	33,92920066	ok	14,58333333	9,79646256
Story1	B10	5,587698	0	0	promedio	18		18	13,19468915	ok		37,5	5,587697943
Story1	B10	5,587698	4,208765	7,389652	negativo			7	7	33,92920066	ok	14,58333333	9,79646256

Figura 5.5-15

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.5.9. Cortante requerido de acero.

La ACI 318-19 (American Concrete Institute, 2019) en el capítulo 22.5.8.1, brinda el criterio de cortante nominal la cual debe ser igual al cortante que aporta el concreto más el cortante que aporta el acero, deduciendo se tiene la siguiente ecuación.

$$V_{REQ}^{ACERO} = \frac{Vu - \phi Vc}{\phi}$$

ϕ : factor de reducción de resistencia a corte 0.75

En el script se define ϕ y se crea una nueva columna “df[“V_requerido (Tonf)”]”, y se calcula el cortante requerido.

```
# Definir phi
phi = 0.75

# Calcular V_requerido
df["V_requerido (Tonf)"] = (df["Vu (Tonf)"] - phi * df["Vc (Tonf)"]) / phi

# Exportar resultado
df.to_excel("8)V_requerido_resultado.xlsx", index=False)
print("✅ Archivo generado '8)V_requerido_resultado.xlsx'")
```

Figura 5.5-16

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Beam	Vv (Tonf)	Vismo (Tonf)	Vc (Tonf)	Tipo	S_media (cm)	S_confinamiento (cm)	S_final (cm)	Vs_acero (Tonf)	Cumple Vv ≤ 4Vc	Av_min_conf (cm²)	Av_min_medio (cm²)	Vu (Tonf)	V_requerido (Tonf)
Story1	B1	6,426830729	3,863490554	7,389652225	positivo			7	7	33,92920066	ok	14,58333333	10,29032128	6,330776154
Story1	B1	6,490677656	0	0	promedio	18		18	13,19468915	ok		37,5	6,490677656	8,654236875
Story1	B1	6,554524583	3,863490554	7,389652225	negativo			7	7	33,92920066	ok	14,58333333	10,41801514	6,501034625
Story1	B10	5,587697943	4,208764618	7,389652225	positivo			7	7	33,92920066	ok	14,58333333	9,79646256	5,672297856
Story1	B10	5,587697943	0	0	promedio	18		18	13,19468915	ok		37,5	5,587697943	7,450263924
Story1	B10	5,587697943	4,208764618	7,389652225	negativo			7	7	33,92920066	ok	14,58333333	9,79646256	5,672297856

Figura 5.5-17

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.5.10. Área mínima de acero requerida.

El área mínima transversal se necesita para el soporte del acero a la demanda, se usa la siguiente ecuación, deducida de la ACI 318-19. (American Concrete Institute, 2019)

$$Av_{\min} = \frac{V_{REQ} \cdot S}{fy \cdot d}$$

Con el script se realiza el cálculo de conversión de unidades del cortante requerido a unidades de kgf, el “pd.isna()” comprueba si un valor está vacío o es un “NaN”, entonces en el dataframe se coloca “none” el cual es sin valor, todo esto realiza en zona de confinamiento y en zona media, al final crea la columna de área mínima requerida y con “round(row() , 2)” se usa el valor de confinamiento con dos decimales.

```
# Convertir Vu requerido a kgf (si está en Tonf)
df["Vrequerido_kgf"] = df["V_requerido (Tonf)"] * 1000

# Calcular Av requerido en zona de confinamiento
df["Av_req_conf (cm²)"] = df.apply(
    lambda row: (row["Vrequerido_kgf"] * row["S_confinamiento (cm)"]) / (f_y*10 * d)
    if not pd.isna(row["S_confinamiento (cm)"]) else None,
    axis=1
)

# Calcular Av requerido en zona media
df["Av_req_medio (cm²)"] = df.apply(
    lambda row: (row["Vrequerido_kgf"] * row["S_medio (cm)"]) / (f_y*10 * d)
    if not pd.isna(row["S_medio (cm)"]) else None,
    axis=1
)

# Crear columna final unificada con 2 decimales
df["Av_requerido_final (cm²)"] = df.apply(
    lambda row: round(row["Av_req_conf (cm²)"], 2) if row["Tipo"] in ["positivo", "negativo"]
    else round(row["Av_req_medio (cm²)"], 2) if row["Tipo"] == "promedio"
    else None,
    axis=1
)

# Guardar el resultado
df.to_excel("9)Av_requerido_resultado.xlsx", index=False)
print("✅ Archivo generado '9)Av_requerido_resultado.xlsx'")
```

Figura 5.5-18

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Beam	Vu (Tonf)	ismo (Tonf)	Vc (Tonf)	Tipo	dia	lami	S_final (cm)	Vs_acero (Tonf)	Cumple Vv s 4Vc	Av_min_conf (cm²)	Av_min_medio (cm²)	Vu (Tonf)	V_requerido (Tonf)	Vrequerido_kgf	req_conf	dt_medio	Av_requerido_final (cm²)
Story1	B1	6,426831	3,863491	7,389652225	positivo	7	7	33,92920066	ok		14,58333333		10,29032128	6,330776154	6330,776154	0,2930915		0,29
Story1	B1	6,490678	0	0	promedio	18	18	13,19468915	ok			37,5	6,490677656	8,654236875	8654,236875	1,0303		1,03
Story1	B1	6,554525	3,863491	7,389652225	negativo	7	7	33,92920066	ok		14,58333333		10,41801514	6,501034625	6501,034625	0,3009738		0,3
Story1	B10	5,587698	4,208765	7,389652225	positivo	7	7	33,92920066	ok		14,58333333		9,79646256	5,672297856	5672,297856	0,2626064		0,26
Story1	B10	5,587698	0	0	promedio	18	18	13,19468915	ok			37,5	5,587697943	7,450263924	7450,263924	0,8869		0,89
Story1	B10	5,587698	4,208765	7,389652225	negativo	7	7	33,92920066	ok		14,58333333		9,79646256	5,672297856	5672,297856	0,2626064		0,26

Figura 5.5-19

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.5.11. Creación del reporte.

Para que el usuario, visualice de mejor manera los resultados, se realiza dos reportes el primero reporte es el diseño longitudinal, el segundo reporte es el diseño transversal, los cuales son datos necesarios para continuar al Dynamo en Revit. (Autodesk Inc., 2024b)

En el primer reporte se visualiza el diámetro de acero necesario para contrarrestar el momento negativo para cada viga, además el armado de la viga con sus respectivos diámetros, finalmente el reporte se realiza al ejecutar el Python. (Python Software Foundation, 2023)

Story	Label	Refuerzo - ϕ Izq (mm)	Refuerzo - ϕ Der (mm)	Armado de Viga
Story1	B1	10.0 mm	10.0 mm	4 ϕ 12

Figura 5.5-20

Nota: Tabla de datos de autoría propia extraída de software Word.

En el segundo reporte se visualiza las separaciones en zona de confinamiento y en zona media, además el área requerida de acero transversal, finalmente nos menciona si es necesario un estribo o requiere refuerzo.

Story	Beam	S_final (cm)	Av_requerido_final (cm ²)	Verificación final	Detalle
Story1	B1	7	0.29	Estribo	1E ϕ de 10mm @ 7cm
Story1	B1	18	1.03	Estribo	1E ϕ de 10mm @ 18cm
Story1	B1	7	0.3	Estribo	1E ϕ de 10mm @ 7cm

Figura 5.5-21

Nota: Tabla de datos de autoría propia extraída de software Word.

5.6. Diseño longitudinal de columnas.

5.6.1. Agrupación de datos.

Para el diseño de aceros longitudinales en columnas, se extrae la tabla “Concrete Column Desing Summary – ACI 318-19”, se agrupa las columnas Story, Label, DesingSect, Station y As, en la extracción los datos tienen unidades de longitud en mm, en el script se crea una nueva columna con sus unidades en el encabezado “As (mm2)” y se procede a eliminar la anterior columna de “As”, finalmente se reinicia el índice y se guarda la nueva columna con “.reset_index(drop=True)”.

```
# Ruta del archivo original
archivo = "C:\tabla_completa_diseno_columnas.xlsx"

# Columnas específicas a extraer
columnas_deseadas = ["Story", "Label", "DesignSect", "Station", "As"]

# Leer solo esas columnas
datos = pd.read_excel(archivo, usecols=columnas_deseadas)

# Poner unidades a la columna
datos["As (mm2)"] = datos["As"]

# Eliminar la columna original As
datos.drop(columns=["As"], inplace=True)

# Reiniciar índice
datos = datos.reset_index(drop=True)

# Guardar archivo nuevo con esas columnas ya procesadas
datos.to_excel("001columnas_diseño_filtrado_rebar.xlsx", index=False)
print("✅ Archivo Generado: '001columnas_diseño_filtrado_rebar.xlsx'")
```

Figura 5.6-1

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023).

Story	Label	DesignSect	Station	As (mm2)
Story2	C1	columna tesis 40x40 cm	0	1600
Story2	C1	columna tesis 40x40 cm	1300	1600
Story2	C1	columna tesis 40x40 cm	2600	1600
Story2	C2	columna tesis 40x40 cm	0	1600
Story2	C2	columna tesis 40x40 cm	1300	1600
Story2	C2	columna tesis 40x40 cm	2600	1775

Figura 5.6-2

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.6.2. Filtrado de datos.

Obtenidas las columnas deseadas, se procede a obtener los valores máximos de cada columna con su respectiva identificación por piso, columna y su debida estación. Se implementa un for el cual permite cambiar de forma automática el nombre de la columna y colocar cm2, además la función de nuevo nombre divide para 100 los valores de áreas en mm2, con el objetivo de colocar en unidades de cm2.

```
# === Cargar archivo original ===
archivo = "001)columnas_diseño_filtrado_rebar.xlsx"
df = pd.read_excel(archivo)

# === Agrupar por Story y Label y obtener máximos numéricos ===
df_maximos = df.groupby(["Story", "Label"], as_index=False).max(numeric_only=True)

# === Identificar columnas que contienen "As" numéricas ===
columnas_As = [col for col in df_maximos.columns if "As" in col and df_maximos[col].dtype in ['int64', 'float64']]

# === Convertir y renombrar correctamente ===
for col in columnas_As:
    nuevo_nombre = col.replace("As", "As").replace("_cm2", "").replace("(mm2)", "").strip()
    nuevo_nombre = nuevo_nombre + " cm2"
    df_maximos[nuevo_nombre] = df_maximos[col] / 100

# === Guardar el nuevo archivo corregido ===
df_maximos.to_excel("002)Col_As_max_cm2.xlsx", index=False)
print("✅ Archivo corregido guardado: 002)Col_As_max_cm2.xlsx")
```

Figura 5.6-3

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023).

Story	Label	Station	As (mm2)	As cm2
Story1	C1	3600	1600	16
Story1	C10	3600	1600	16
Story1	C11	3600	1600	16
Story1	C12	3600	1600	16

Figura 5.6-4

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.6.3. Diseño longitudinal.

El script comienza importando un módulo de “collections”, el cual se llama “defaultdict”, esto no permite crear claves para tener muchas combinaciones en una misma columna con diferentes diámetros, los cuales el usuario debe ingresar al inicio. Adicional se programa una tabla de diámetros y áreas de varillas comerciales del Ecuador, además se crea un parámetro en el cual el usuario puede ingresar hasta 4 combinaciones que se crea convenientes para el diseño, para hacer posible, se codifica con bucles “for”, el primero por todas las columnas del archivo y evalúa cada columna por separado, el segundo recorre por todos los diámetros de las varillas comerciales, el tercer calcula las combinaciones posibles para cada combinación, el ultimo for recorre grupos de opciones y selecciona la mejor alternativa, finalmente todo esto se realiza en función de las área requerida para no sobredimensionar y tener varias alternativas de combinaciones de diseño longitudinal para columnas.

```

from collections import defaultdict
# === 1. Cargar archivo de columnas con As requerido ===
archivo = "002)Col_As_max_cm2.xlsx"
df = pd.read_excel(archivo)
# === 2. Datos de varillas: área en cm² ===
tabla_varillas = {
    12: 1.13,
    14: 1.54,
    16: 2.01,
    18: 2.54,
    20: 3.14,
    22: 3.80,
    25: 5.07,
    28: 6.36,
    30: 7.07,
    32: 8.04
}
# === 3. Parámetros columna ===
num_varillas_opciones = [Opcion1, Opcion2, Opcion3, Opcion4]
area_columna = b_columna * h_columna
columna_as = "As cm2"
# === 4. Calcular TODAS las combinaciones posibles que cumplan As_req ===
todas_opciones = []

for _, fila in df.iterrows():
    story = fila["Story"]
    label = fila["Label"]
    As_req = fila[columna_as]

    for diam, area_unitaria in tabla_varillas.items():
        for n in num_varillas_opciones:
            As_prov = n * area_unitaria
            if As_prov >= As_req:
                cuantia = As_prov / area_columna
                todas_opciones.append({
                    "Story": story,
                    "Label": label,
                    "As_req (cm2)": round(As_req, 2),
                    "Varilla (mm)": diam,
                    "Nº Varillas": n,
                    "As_prov (cm2)": round(As_prov, 2),
                    "Cuantia (%)": round(cuantia * 100, 2)
                })

# === 5. Agrupar y seleccionar la mejor opción por cantidad de varillas ===
grupo_columnas = defaultdict(list)

for fila in todas_opciones:
    key = (fila["Story"], fila["Label"], fila["Nº Varillas"])
    grupo_columnas[key].append(fila)

seleccion_final = []

for (story, label, n_var), opciones in grupo_columnas.items():
    mejor_opcion = sorted(opciones, key=lambda x: abs(x["As_prov (cm2)"] - x["As_req (cm2)"])[0])
    seleccion_final.append(mejor_opcion)

# === 6. Crear DataFrame final ===
df_resultado = pd.DataFrame(seleccion_final)
df_resultado = df_resultado.sort_values(by=["Story", "Label", "Nº Varillas"]).reset_index(drop=True)
# === 7. Guardar en Excel ===
df_resultado.to_excel("003)Diseño_Columnas_Varillas_Optimas.xlsx", index=False)
print("📄 Archivo generado: 003)Diseño_Columnas_Varillas_Optimas.xlsx")

```

Figura 5.6-5

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023).

Story	Label	As_req (cm2)	Varilla (mm)	N° Varillas	As_prov (cm2)	Cuantía (%)
Story1	C1	16	16	8	16.08	1
Story1	C1	16	16	10	20.1	1.26
Story1	C1	16	14	12	18.48	1.16
Story1	C1	16	14	14	21.56	1.35

Figura 5.6-6

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.7. Diseño transversal de columnas.

5.7.1. Agrupación de datos.

Con la extracción de datos de la tabla de “Element Forces - Columns”, desde el software comercial se agrupa las siguientes columnas Story, Column, Station, V_2 y P una vez ordenado se transforma las estaciones en cm, para facilitar las operaciones geométricas en los siguientes cálculos, finalmente los valores de la columna V_2 en valor absoluto ya que nos interesa la magnitud de cada valor referente a cada columna.

```
#Cargar el archivo original exportado desde ETABS
df = pd.read_excel("D:\tabla_columna_cortantes.xlsx")

#Crear columna 'Station (cm)' si no existe
if 'Station (cm)' not in df.columns and 'Station' in df.columns:
    df['Station (cm)'] = df['Station'] / 10

#Determinar el nombre correcto de la columna que identifica las columnas (Column o Element)
columna_nombre = 'Column' if 'Column' in df.columns else 'Element'

#Seleccionar columnas necesarias (agregamos P)
df_filtrado = df[['Story', columna_nombre, 'Station (cm)', 'V2', 'P']].copy()
df_filtrado.rename(columns={columna_nombre: 'Column'}, inplace=True)

#Calcular valor absoluto de V2
df_filtrado['|V2|'] = df_filtrado['V2'].abs()

#Obtener todas las filas donde el valor absoluto de V2 es el máximo por columna
valores_maximos = df_filtrado.groupby(['Story', 'Column'])['|V2|'].max().reset_index()
df_maximos = df_filtrado.merge(valores_maximos, on=['Story', 'Column', '|V2|'])

#Eliminar columna auxiliar y dejar V2 como positivo
df_maximos.drop(columns='|V2|', inplace=True)
df_maximos['V2'] = df_maximos['|V2|'].abs()

#Resetear el índice (sin mostrar columna extra)
df_maximos.reset_index(drop=True, inplace=True)

#Guardar a Excel
df_maximos.to_excel("0001cortantes_maximos_col.xlsx", index=False)
print("✅ Archivo guardado: '0001cortantes_maximos_col.xlsx'")
```

Figura 5.7-1

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Column	tation (cm)	V2	P
Story2	C1	0	2.572358	-8.4758
Story2	C1	130	2.572358	-7.77193
Story2	C1	260	2.572358	-7.06806
Story2	C2	0	4.26421	-13.936
Story2	C2	130	4.26421	-13.2322
Story2	C2	260	4.26421	-12.5283

Figura 5.7-2

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.7.2. Filtrado de datos.

Para realizar las operaciones respecto a cada columna el usuario debe correr el programa una vez para poder elegir la opción que crea conveniente para el diseño del armado longitudinal de la columna, una vez que tenga elegida la opción para el diseño, debe ingresar en el paso 2 el diámetro seleccionado de diámetro longitudinal, nuevamente correr el programa.

El script filtra los datos y selecciona la fila con la mayor estación en cm, referente a cada piso y columna, para usar los datos de la columna P se coloca en valor absoluto, con la función “.abs()” esta es un método de la Librería de pandas para aplicar todos los valor de una columna en valor absoluto.

```

#Cargar el archivo con los valores máximos de V2 por columna
df = pd.read_excel("0001)cortantes_maximos_col.xlsx") # Cambia si el archivo tiene otro nombre

#Seleccionar la fila con la mayor Station (cm) por cada Story y Column
df_mayor_estacion = df.loc[df.groupby(['Story', 'Column'])['Station (cm)'].idxmax()]

#Convertir P a valor absoluto
df_mayor_estacion['P'] = df_mayor_estacion['P'].abs()

#Ordenar y resetear indice
df_mayor_estacion = df_mayor_estacion.sort_values(by=['Story', 'Column']).reset_index(drop=True)

#Mostrar o guardar el resultado
print(df_mayor_estacion)

#Guardar como Excel
df_mayor_estacion.to_excel("0002)cortantes_maximos_mayor_station.xlsx", index=False)
print("✅ Archivo guardado: '0002)cortantes_maximos_mayor_station.xlsx")

```

Figura 5.7-3

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Column	tation (cm)	V2	P
Story1	C1	360	3.790645	18.11377
Story1	C10	360	3.790645	0.572966
Story1	C11	360	4.194965	5.450126
Story1	C12	360	3.964853	0.466819

Figura 5.7-4

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.7.3. Longitud de zona de confinamiento.

En columnas la normativa ACI 318-19, describe en el capítulo 18.7.5.1 la longitud en la zona de confinamiento “lo” se calcula en función de tres condiciones y se elige la mayor de las tres, las cuales son las siguientes:

$$Lo = \begin{cases} \text{Dimension mayor de la columna (h o b)} \\ \frac{1}{6} \text{ de la altura libre entre los piso} \\ 450 \text{ mm} \end{cases}$$

El script permite calcular las tres condiciones y toma la mayor con la función “.max(axis=1)”, con la función “np.round()” redondea al múltiplo 5 más cercano.

```

#Cargar el archivo
df = pd.read_excel("0002)cortantes_maximos_mayor_station.xlsx")
lo_minimo = 45 # (c) Mínimo Lo = 45 cm

#Cálculo de las tres condiciones
df['Lo_a'] = max(h_columna, b_columna) # (a) Altura de la sección (cara del nudo)
df['Lo_b'] = df['Station (cm)] / 6 # (b) Un sexto de la luz libre
df['Lo_c'] = lo_minimo # (c) 45 cm fijo

#Tomar la mayor de las tres
df['Lo_sin_redondear'] = df[['Lo_a', 'Lo_b', 'Lo_c']].max(axis=1)

#Redondear al múltiplo de 5 más cercano
df['Lo (cm)'] = (np.round(df['Lo_sin_redondear'] / 5) * 5).astype(int)

#Elegir columnas finales
df_resultado = df[['Story', 'Column', 'Station (cm)', 'V2', 'P', 'Lo (cm)']]

#Mostrar o guardar
print(df_resultado)

#Guardar a Excel
df_resultado.to_excel("0003)longitud_confinamiento_Lo.xlsx", index=False)
print("✅ Archivo guardado: '0003)longitud_confinamiento_Lo.xlsx'")

```

Figura 5.7-5

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Column	tation (cm)	V2	P	Lo (cm)
Story1	C1	360	3.790645	18.11377	60
Story1	C10	360	3.790645	0.572966	60
Story1	C11	360	4.194965	5.450126	60
Story1	C12	360	3.964853	0.466819	60

Figura 5.7-6

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.7.4. Espaciamiento entre estribos en zona de confinamiento.

El espaciamiento de estribos en columnas se determinó en base a los criterios establecidos por la NEC-15 (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015c) de hormigón armado como de la ACI 318-19 (American Concrete Institute, 2019). Las tres condiciones que usamos son las siguientes:

$$\begin{cases} \frac{b}{4} \\ 6 d_b \text{ refuerzo longitudinal} \\ S_o = 100 + \left(\frac{350 - h_x}{3} \right) \end{cases}$$

El script calcula las 3 condiciones, “.min(axis=1)” con esta función obtiene el mínimo de

las tres, y redondea con él “.round() / 2” para obtener 2 decimales.

```
#Cargar archivo original con Station, V2 y Lo
df = pd.read_excel("0003)longitud_confinamiento_Lo.xlsx")

#(a) Cuarta parte de la dimensión menor
df['S_a (mm)'] = (b_columna / 4) * 10

#(b) Seis veces el diámetro de la varilla longitudinal
df['S_b (mm)'] = 6 * diametro_long_mm

#(c) Fórmula So = 100 + ((350 - hx) / 3)
varilla_cm = diametro_long_mm / 10
hx = (h_columna - recubrimiento - recubrimiento - varilla_cm - varilla_cm) / 2 - varilla_cm / 2 # en cm
df['S_c (mm)'] = 100 + ((350 - hx * 10) / 3)

#Obtener mínimo entre las 3 condiciones
df['S_min (mm)'] = df[['S_a (mm)', 'S_b (mm)', 'S_c (mm)']].min(axis=1)

#Convertir a cm y redondear al 0.5 cm más cercano
df['Espaciamento Zona Confinada S (cm)'] = (df['S_min (mm)'] / 10)
df['Espaciamento Zona Confinada S (cm)'] = (df['Espaciamento Zona Confinada S (cm)'] * 2).round() / 2

#Tabla final conservando todo lo necesario
df_resultado = df[['Story', 'Column', 'Station (cm)', 'V2', 'P', 'Lo (cm)', 'Espaciamento Zona Confinada S (cm)']]

#guardar
df_resultado.to_excel("0004)espaciamento_zona_confinada_final.xlsx", index=False)
print("✅ Archivo guardado como '0004)espaciamento_zona_confinada_final.xlsx")
```

Figura 5.7-7

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Column	Station (cm)	V2	P	Lo (cm)	Espaciamento Zona Confinada S (cm)
Story1	C1	360	3.790645	18.11377	60	10
Story1	C10	360	3.790645	0.572966	60	10
Story1	C11	360	4.194965	5.450126	60	10
Story1	C12	360	3.964853	0.466819	60	10

Figura 5.7-8

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.7.5. Espaciamento entre estribos en zona media.

Para desarrollar los espaciamentos entre estribos de columnas, se obtiene con las dos ecuaciones que se encuentra en la norma NEC-15 (Ministerio de Desarrollo Urbano y Vivienda (MIDUVI), 2015c) de hormigón armado, capítulo 4.3.4. Donde las 2 condiciones son las siguientes:

$$S \leq \begin{cases} 6 d_b \text{ refuerzo longitudinal.} \\ 150_{mm} \end{cases}$$

El script contiene las dos condiciones calcula y elige la menor con la función “min()”, después con un “if” condicionamos si la separación del centro es menor a 15 cm entonces redondea a 10 cm para facilitar la colocación en obra.

```
#Cargar el archivo con columnas
df = pd.read_excel("0004)espaciamento_zona_confinada_final.xlsx")

#Condición (a): 6 veces el diámetro
s_a = 6 * diametro_long_mm
#Condición (b): 150 mm
s_b = 150

#Menor en mm → convertir a cm
s_medio_cm = min(s_a, s_b) / 10

#Redondear:
if s_medio_cm < 15:
    espaciamento_cm = 10.0
else:
    espaciamento_cm = int(s_medio_cm * 2) / 2

#Asignar el valor a todas las filas
df['Espaciamento Zona Media S (cm)'] = espaciamento_cm

#Seleccionar columnas finales
df_resultado = df[['Story', 'Column', 'Station (cm)', 'V2', 'P', 'Lo (cm)',
                  'Espaciamento Zona Confinada S (cm)', 'Espaciamento Zona Media S (cm)']]
#Guardar en Excel
df_resultado.to_excel("0005)espaciamientos_completo.xlsx", index=False)
print("✅ Archivo guardado como '0005)espaciamientos_completo.xlsx'")
```

Figura 5.7-9

Nota: Imagen de autoría propia extraída de software Python. (Python Software Foundation, 2023)

Story	Column	Station (cm)	V2	P	Lo (cm)	Espaciamento Zona Confinada S (cm)	Espaciamento Zona Media S (cm)
Story1	C1	360	3.790645	18.11377	60	10	10
Story1	C10	360	3.790645	0.572966	60	10	10
Story1	C11	360	4.194965	5.450126	60	10	10
Story1	C12	360	3.964853	0.466819	60	10	10

Figura 5.7-10

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.7.6. Ecuaciones para reforzar por confinamiento.

Para el cálculo de refuerzo transversal para columnas en pórticos especiales resistentes a momento la normativa ACI 318-19 (American Concrete Institute, 2019) detalla fórmulas para estribos cerrados de confinamiento, en el capítulo 18.7.5.4 la cual es la siguiente tabla:

Refuerzo Transversal	Condición	Refuerzo Transversal	Refuerzo Transversal
$\frac{A_{sh}}{sb_c}$ para estribos cerrados de confinamiento rectilíneos.	$P_u \leq 0.30 A_g * f_c$ y $f_c \leq 70 \text{ Mpa}$	Mayor de a y b	$a) 0.30 * \left(\frac{A_g}{A_{ch}} - 1 \right) * \frac{f_c}{f_y}$

Por lo siguiente en el programa se realiza la condición para que elija el caso, con las condiciones si P es menor que $0.30 * A_g * f_c$, y si f_c es menor o igual al límite permitido, con la función “df.apply(lambda)” la cual es una función anónima que nos permite realizar si cumple o no cumple los dos criterios establecidos.

```

#Cargar el archivo que contiene la columna P
df = pd.read_excel("0005)espaciamientos_completo.xlsx")

Ag = b_columna*h_columna
limite_fc_mpa = 70 # Límite máximo permitido [MPa]

#Cálculo del límite de carga axial permisible
df["0.30Agf'c"] = (0.30 * Ag * f_c*10)/1000 # En tonf

#Evaluar condiciones
df['Cumple_Pu'] = df['P'] <= df["0.30Agf'c"]
df['Cumple_fc'] = f_c <= limite_fc_mpa

#Evaluar si ambas condiciones se cumplen
df['Resultado'] = df.apply(lambda x: "Cumple" if x['Cumple_Pu'] and x['Cumple_fc'] else "No cumple", axis=1)

#Seleccionar columnas relevantes
df_resultado = df[['Story', 'Column', 'P', "0.30Agf'c", 'Cumple_Pu', 'Cumple_fc', 'Resultado']]

#Guardar resultado a Excel
df_resultado.to_excel("0006)verificacion_Pu_fc.xlsx", index=False)
print("✅ Archivo guardado: '0006)verificacion_Pu_fc.xlsx'")

```

Figura 5.7-11

Story	Column	P	0.30Agf'c	Cumple_Pu	Cumple_fc	Resultado
Story1	C1	18.11377	115.2	VERDADERO	VERDADERO	Cumple
Story1	C10	0.572966	115.2	VERDADERO	VERDADERO	Cumple
Story1	C11	5.450126	115.2	VERDADERO	VERDADERO	Cumple
Story1	C12	0.466819	115.2	VERDADERO	VERDADERO	Cumple

Figura 5.7-12

Nota: Tabla de datos de autoría propia extraída de software Excel.

Aplicamos las expresiones aplicables en el script de Python (Python Software Foundation, 2023) para calcular las áreas requeridas para el corte.

```
#Cargar el archivo con datos completos por columna
df = pd.read_excel("0005)espaciamientos_completo.xlsx")

# === Geometría efectiva
Ag = h_columna * b_columna
bc1 = h_columna - 2 * recubrimiento
bc2 = b_columna - 2 * recubrimiento
bc_102 = max(bc1, bc2)
Ach = bc1 * bc2

#Calcular Ash por fórmula (a)
df['Ash_a (cm2)'] = (0.3 * bc_102 * ((Ag / Ach) - 1) * (f_c*10 / f_y*10))/100

#Calcular Ash por fórmula (b)
df['Ash_b (cm2)'] = (0.09 * bc_102 * (f_c*10 / f_y*10))/100

#Elegir el mayor entre a y b
df['Ash requerido (cm2)'] = df[['Ash_a (cm2)', 'Ash_b (cm2)']].max(axis=1)

#Filtrar solo la fila de mayor Ash por columna
df_max_ash = df.loc[df.groupby(['Story', 'Column'])['Ash requerido (cm2)'].idxmax()].reset_index(drop=True)

#Guardar resultado
df_max_ash.to_excel("0007)Ash_refuerzo_confinamiento.xlsx", index=False)
print("✅ Archivo guardado: '0007)Ash_refuerzo_confinamiento.xlsx'")
```

Figura 5.7-13

Story	Column	Station (cm)	V2	P	Lo (cm)	Zona Cor	Espaciamiento	Zona Media S (cm)	Ash_a (cm2)	Ash_b (cm2)	Ash requerido (cm2)
Story1	C1	360	3.790645	18.11377	60	10		10	0.308571429	0.164571429	0.308571429
Story1	C10	360	3.790645	0.572966	60	10		10	0.308571429	0.164571429	0.308571429
Story1	C11	360	4.194965	5.450126	60	10		10	0.308571429	0.164571429	0.308571429

Figura 5.7-14

Nota: Tabla de datos de autoría propia extraída de software Excel.

5.7.7. Creación del reporte.

Para que el usuario, visualice de mejor manera los resultados, se realiza dos reportes el

primero reporte es el diseño longitudinal, el segundo reporte es el diseño transversal, los cuales son datos necesarios para continuar al Dynamo en Revit. (Autodesk Inc., 2024b)

En el primer reporte se visualiza el diámetro de acero necesario para cada columna, además el armado de la columna con sus respectivos diámetros, finalmente el reporte se realiza al ejecutar el Python. (Python Software Foundation, 2023).

Story	Label	As_req (cm2)	Varilla (mm)	N° Varillas	As_prov (cm2)	Cuantía (%)
Story2	C11	18.69	18	8	20.32	1.27

Figura 5.7-15

Nota: Tabla de datos de autoría propia extraída de software Word.

En el segundo reporte se visualiza las separaciones en zona de confinamiento y en zona media, además el área requerida de acero transversal, finalmente nos menciona si es necesario un estribo o requiere refuerzo.

Story	Column	Separación Zona Confinada	Separación Zona Media	Av_requerido_final (cm ²)	Verificación final	Av_colocado
Story 1	C1	10	10	0.309	Estribo	1E φ de 10mm

Figura 5.7-16

Nota: Tabla de datos de autoría propia extraída de software Word.

CAPITULO VI

6. Código visual de en Dynamo.

En el presente trabajo se realizaron dos códigos visuales separados en Dynamo, uno para columnas y otro para vigas, el propósito de separarlos fue que cada código tiene concatenado ciertos parámetros de diseño para el tipo de elemento estructural, vigas y columnas al tener distintos elementos físicos (acero negativo en vigas, zonas de confinamiento, refuerzo positivo, etc.) se tuvo la necesidad de separar los códigos. Sin embargo la “idea” es la misma : Crear geometrías que dependan de valores input (valores de entrada) para después asignarles una familia deseada, en este caso una familia de refuerzo estructural (acero).

6.1. Versión de Revit

Revit desde el año 2020 integró Dynamo como parte de su paquete de instalación, antes de esta versión era necesario instalar Dynamo for Revit por separado al software de dibujo, el presente código se encuentra hecho en Revit 2024, una versión que viene con Dynamo por defecto, debido a las actualizaciones constantes por parte de AutoDesk a su software ningún archivo se puede abrir en una versión anterior al año del programa con el que fue creado

6.2. Dynamo Player

Dynamo Player es un add de Dynamo que de igual manera viene instalado por defecto con el Revit desde el año 2020, esta extensión lo que permite es reproducir un código visual previamente creado en Dynamo a través de una ventana emergente, sin necesidad de abrir el código en si, solo depende de los valores de entrada asignados que el usuario quiera.

6.3. Código visual para columnas.

Este script visual desarrollado en Dynamo para Revit (Autodesk Inc., 2024b) automatiza la colocación de acero de refuerzo en columnas estructurales, optimizando el proceso de modelado y asegurando coherencia con los parámetros de diseño. Conformado por 83 nodos distribuidos por funciones específicas, el gráfico integra geometría, filtros, parámetros estructurales y asignación de familias de refuerzo, ofreciendo una solución eficiente y precisa para la documentación de proyectos estructurales.

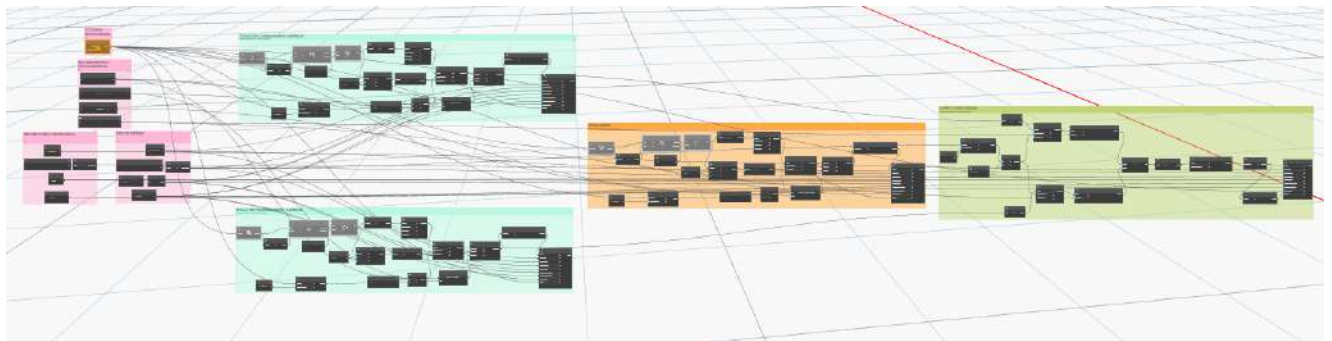


Figura 6.3-1

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

6.3.1. Selección de elemento columna.

Primeramente, se llama al nodo “Select Element”, cuya función principal es pasar a Dynamo un elemento seleccionado desde la interfaz de Revit, en este caso tendrá como salida la columna seleccionada desde la cual se empezará a desarrollar el código:

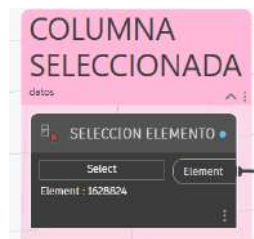


Figura 6.3-2

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

6.3.2. Creación de estribos columna

Para la creación de estribos es necesario separar 3 bloques de códigos con la misma funcionalidad, un bloque generara los estribos desde la base del elemento hasta la longitud deseada (zona de confinamiento inicial), otro bloque se encargara de generar estribos desde el fin del primer bloque hasta el comienzo del tercer bloque (zona media) y por último el bloque generara geometrías desde el fin del segundo bloque hasta la longitud total del elemento (zona de confinamiento final), para practicidad de este trabajo solo se describirá un bloque de los 3 mencionados, ya que los 3 funcionan exactamente con los mismos nodos, solo cambiando las coordenadas de inicio y de fin.

6.3.3. Descripción de bloque de nodos para estribos columna.

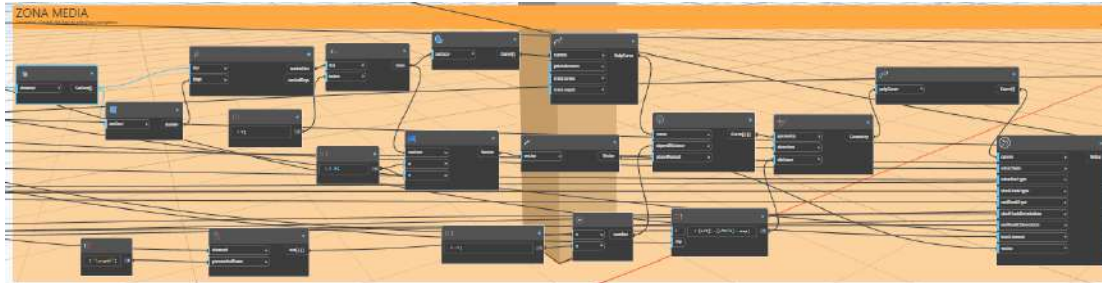


Figura 6.3-3

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Una vez seleccionado el elemento se procede a aislar las caras de este mismo con el nodo “Element Faces”, que da como salida las caras de la geometría seleccionada, en el caso de la columna 6 caras, después con un nodo informativo llamado “Area Surface” calculamos las áreas de todas las caras extraídas, esto con el fin de seleccionar el área menor, pues el área menor será la cara de la sección transversal de la columna:

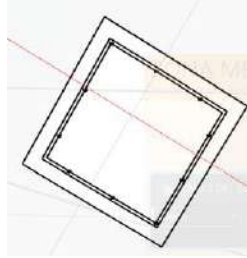


Figura 6.3-4

Nota: Imagen de autoría propia extraída de código en Dynamo. (Autodesk Inc., 2024b)

De esta manera se tiene la cara donde se quiere dibujar el contorno del estribo, después con un nodo llamado “Perimeter Surface” se extrae el perímetro de la cara dando como resultado el contorno del área transversal de la columna. Una vez aislado este perímetro se lo extruye con un offset negativo para que el contorno quede dentro de la columna, este offset será el recubrimiento, en el caso de este trabajo 4 cm de cara de columna. Una vez extruido el contorno se lo duplica con un “code block” cada cierta distancia hasta cierta longitud.

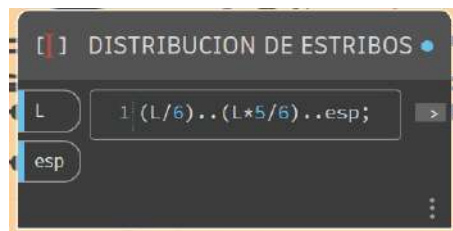


Figura 6.3-5

Nota: Imagen de autoría propia extraída de código en Dynamo. (Autodesk Inc., 2024b)

En este específico code block se está dando la condición de que el perímetro extraído se duplique desde $L/6$ (siendo L la longitud del elemento seleccionado) hasta $5/6 * L$ cada cierto espacio, en este caso “esp” (esp esta concatenado al espaciamiento dispuesto por el usuario), de esta manera se duplica el perímetro en la zona media de columna definido por un espaciamiento (estribos de zona media). Para la duplicación también nos valemos de un nodo llamado “normal

vector” que saca el vector normal a una superficie, en este caso la superficie es la cara de la columna y la dirección del duplicado es el vector normal a la misma, de este modo no afecta la inclinación del elemento a la hora de duplicar los contornos.

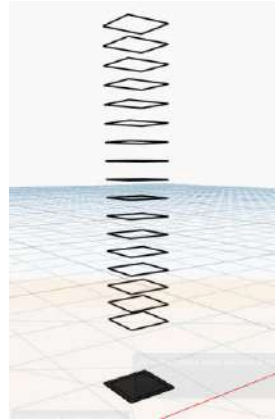


Figura 6.3-6

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Por último se asigna esta familia de curvas a un nodo personalizado llamado Structural Rebar donde se especifican como datos de entrada el diámetro que queremos asignar, el tipo de refuerzo, el tipo de gancho y la orientación de este.

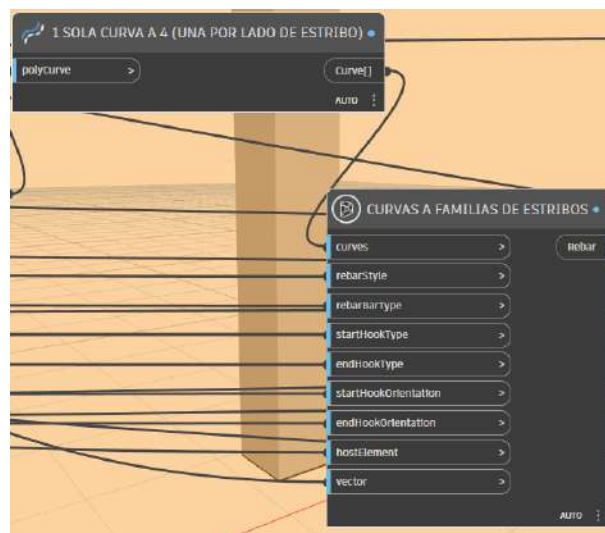


Figura 6.3-7

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Una vez asignadas las curvas al nodo estructural dentro de Revit se podrá apreciar la creación de los estribos dentro del elemento.

6.3.4. Creación de aceros longitudinales columna.

Los aceros longitudinales se crean a partir de los 3 bloques de estribos creados anteriormente, tomando como referencia el primer y último estribo



Figura 6.3-8

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

6.3.5. Descripción de bloque de nodos para aceros longitudinales columna.

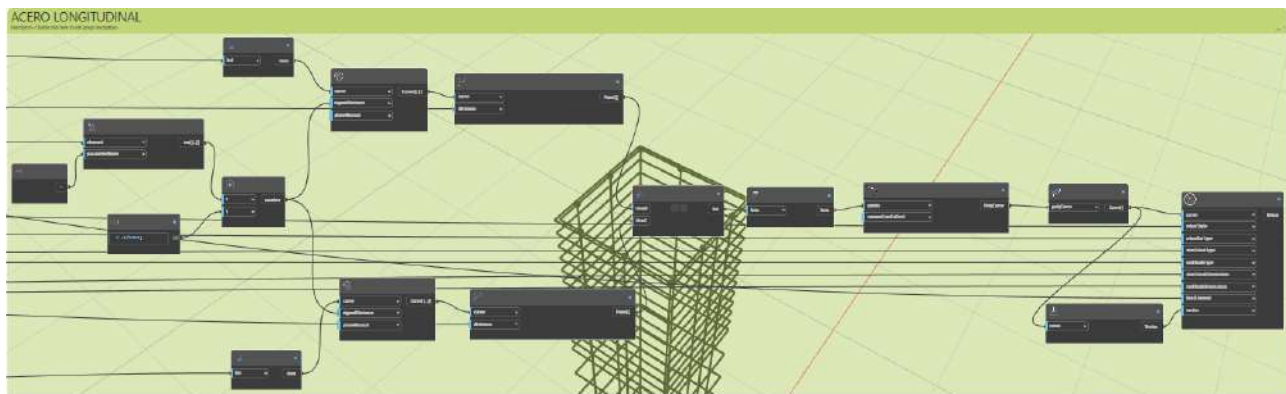


Figura 6.3-9

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Primero se divide el primer y último estribo con el nodo “Curve.OffsetMany”, el cual pide como parámetro de entrada el número de divisiones a realizar en la curva seleccionada, ese número queda como input en Dynamo (Autodesk Inc., 2024b) player para seleccionar la cantidad de varillas longitudinales que se van a colocar en la columna, luego con un nodo tipo lista se agrupan las líneas segmentadas y se aíslan los puntos de corte antes generados, al tener esta lista de punto, a través del nodo line.by.points se unen los puntos segmentados del primer estribo con el ultimo generando una línea que atraviesa toda la sección transversal del elemento.

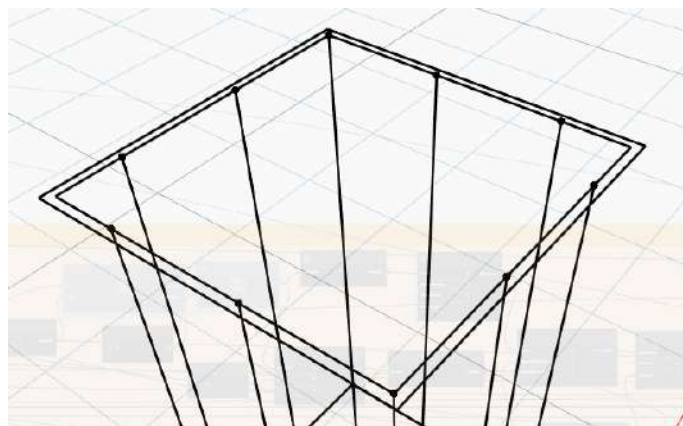


Figura 6.3-10

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Estas líneas se extruyen hacia dentro de la sección una distancia igual al diámetro del estribo, esto con fin de que los aceros longitudinales queden por adentro del estribo, esto se logra con un grupo de nodos que extrae información del estribo antes hecho y calculando sus parámetros (información como longitud, área, perímetro, etc.)

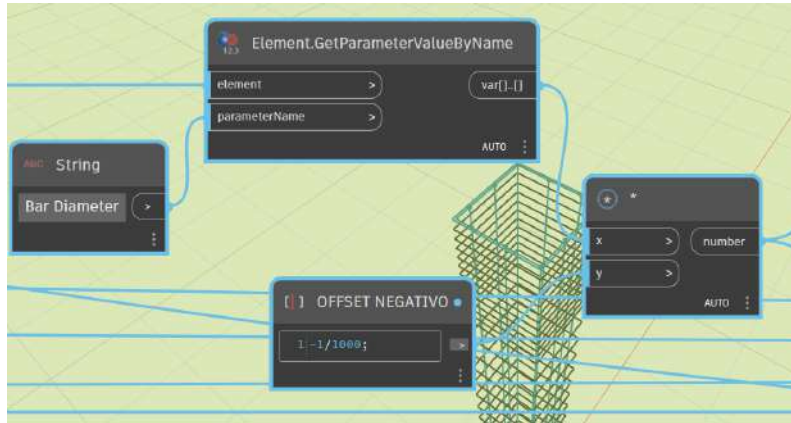


Figura 6.3-11

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Al igual que en los estribos en el siguiente fragmento de bloque de nodos se observa como estas curvas se van a un nodo personalizado para convertirse de curvas a elementos solidos (acero de refuerzo).



Figura 6.3-12

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

6.4. Código visual para vigas.

Este script visual en Dynamo (Autodesk Inc., 2024b), está compuesto por 220 nodos, ha sido desarrollado para automatizar de forma integral la colocación del acero de refuerzo en vigas dentro de un entorno. Dada la complejidad que implica modelar acero positivo, acero negativo, ganchos, zonas de confinamiento y longitudes de desarrollo, el código organiza de manera estructurada cada una de estas etapas, permitiendo generar modelos precisos y coordinados directamente en Revit (Autodesk Inc., 2024d).

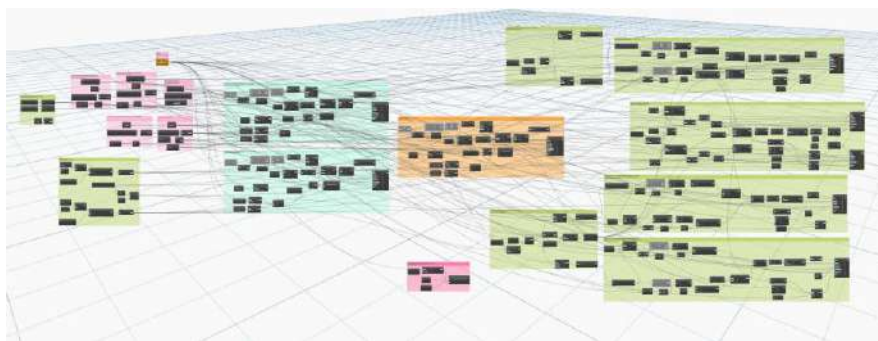


Figura 6.4-1

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

6.4.1. Creación de estribos vigas

Al igual que las columnas, las vigas necesitan 3 paquetes diferenciados de estribos, uno para la zona de confinamiento inicial, otro para la zona de confinamiento final, y un último para la zona de separación media, estos 3 paquetes manejan el mismo funcionamiento por lo tanto por practicidad se explicará solo uno a continuación.

6.4.2. Selección de elemento viga

Primeramente, se llama al nodo “Select Element”, cuya función principal es pasar a Dynamo (Autodesk Inc., 2024b) un elemento seleccionado desde la interfaz de Revit, en este caso tendrá como salida la viga seleccionada desde la cual se empezará a desarrollar el código:



Figura 6.4-2

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

6.4.3. Descripción bloque de nodos para estribos de vigas

Una vez seleccionado el elemento se procede a aislar las caras de este mismo con el nodo “Element Faces”, que da como salida las caras de la geometría seleccionada, en el caso de la viga 6 caras, después con un nodo informativo llamado “Area Surface” calculamos las áreas de todas las caras extraídas, esto con el fin de seleccionar el área menor, pues el área menor será la cara de la sección transversal de la Viga:



Figura 6.4-3

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

De esta manera se tiene la cara donde se quiere dibujar el contorno del estribo, después con un nodo llamado “Perimeter Surface” se extrae el perímetro de la cara dando como resultado el contorno del área transversal de la columna. Una vez aislado este perímetro se lo extruye con un

offset negativo para que el contorno quede dentro de la columna, este offset será el recubrimiento, en el caso de este trabajo 4 cm de cara de columna. Una vez extruido el contorno se lo duplica con un “code block” cada cierta distancia hasta cierta longitud.

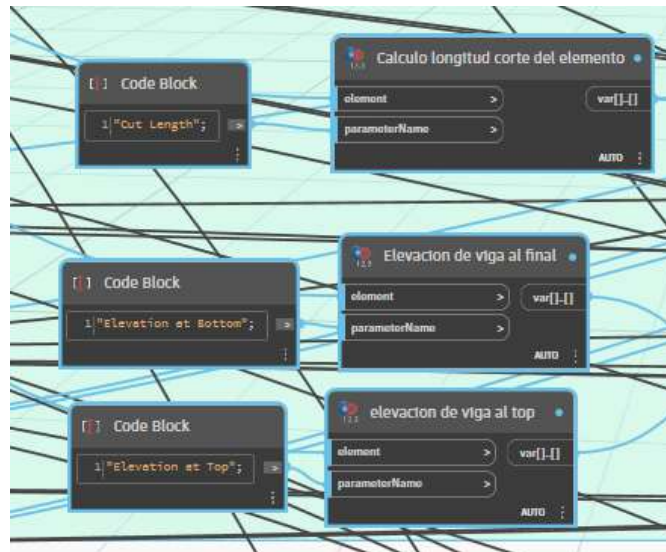


Figura 6.4-4

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

En los estribos para viga se empieza aislando parámetros del material a través del nodo “Get.element.Parameters”, esto con finalidad de trabajar con la longitud libre del elemento, mas no la completa a diferencia de las columnas, “Cut Length” es el nombre que debemos poner en un code block para determinar esta longitud, (El nombre del parámetro puede variar según la versión e idioma de Revit(Autodesk Inc., 2024d)) , una vez extraída la longitud libre de la viga seleccionada extraemos sus cotas en el plano z, esto con fin de calcular el h de la viga.

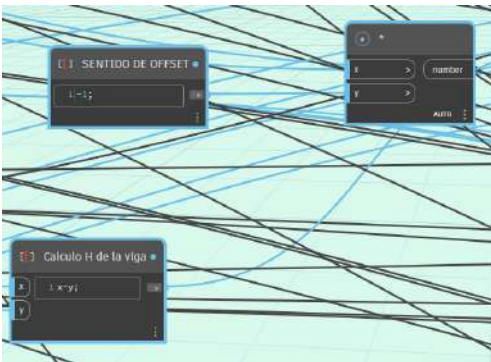


Figura 6.4-5

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Para zona de confinamiento y separación del primer estribo nos apoyamos de lo dictado por ACI 318-19(American Concrete Institute, 2019):

18.6.4.4 El primer estribo cerrado de confinamiento debe estar situado a no más de 50 mm de la cara de la columna de apoyo. El espaciamiento de los estribos cerrados de confinamiento no debe exceder el menor de (a) hasta (d).

American Concrete Institute. (2019). *Building code requirements for structural concrete (ACI 318-19) and commentary* (pp. 313). ACI.

18.6.4.1 Deben colocarse estribos cerrados de confinamiento en las siguientes regiones de las vigas:

(a) En una longitud igual a dos veces la altura de la viga, medida desde la cara de la columna de apoyo hacia el centro de la luz, en ambos extremos de la viga.

(b) En longitudes iguales a dos veces la altura de la viga a ambos lados de una sección donde puede ocurrir fluencia por flexión debido a los desplazamientos laterales más allá del rango elástico de comportamiento.

American Concrete Institute. (2019). *Building code requirements for structural concrete (ACI 318-19) and commentary* (p. 313). ACI.

Donde se define que el primer estribo de la viga debe estar separado 5 cm de la cara de apoyo y que su longitud de confinamiento será 2 veces H, siendo H el peralte de la viga.

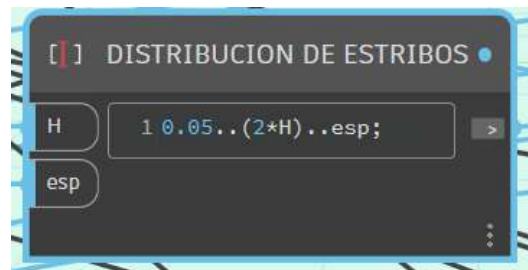


Figura 6.4-6

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Una vez extraído la altura de la viga extruimos los contornos del estribo creado con un code block que especifica que en este paquete (zona confinamiento inicio) empezara la extrusión a partir de los primeros 0.05 m de la cara de columna hasta 2 veces H (peralte de la viga antes calculado), cada esp, en este caso esp es un factor de entrada definido por el usuario para el espaciamiento de estribos en la zona de confinamiento.

Los otros 2 bloques cumplen la misma función solo cambiando el code block para alterar la extrusión en las dimensiones de la viga.

Al igual que todas las curvas creadas, estas curvas se van a un nodo personalizado para

convertirse de curvas a elementos solidos (acero de refuerzo).

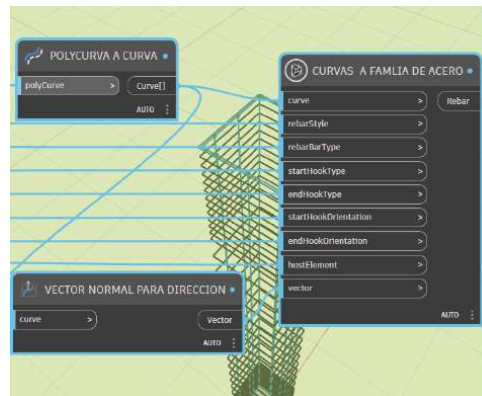


Figura 6.4-7

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

6.4.4. Bloque de nodos para refuerzo positivo

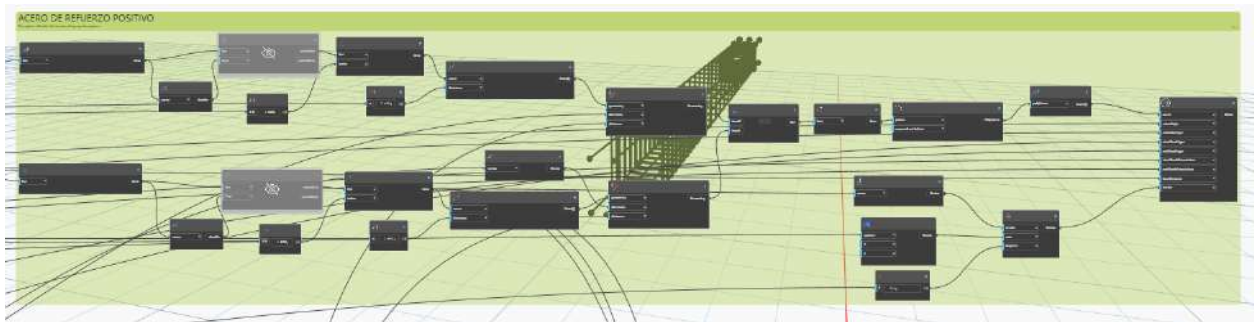


Figura 6.4-8

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Para el acero positivo se descompone los estribos del inicio y del final, pero a diferencia de columnas en el caso de vigas solo se requiere la parte de abajo del estribo, es decir la curva inferior como se muestra a continuación.



Figura 6.4-9

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b)

Una vez seleccionado la curva inferior del estribo se divide en segmentos, el número de segmentos a dividir estará como dato de entrada para dibujar el número de varillas longitudinales positivas. Después con el nodo Line.ByPoints se unen los segmentos del primer y del último estribo formando una curva que paralela al vector normal de la cara transversal de la viga.

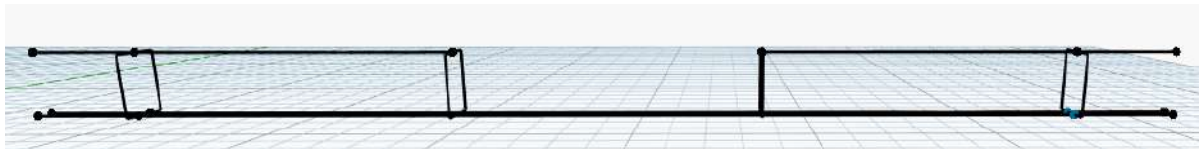


Figura 6.4-10

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Al igual que todas las curvas creadas, estas se van a un nodo personalizado para convertirse de curvas a elementos sólidos (acero de refuerzo).

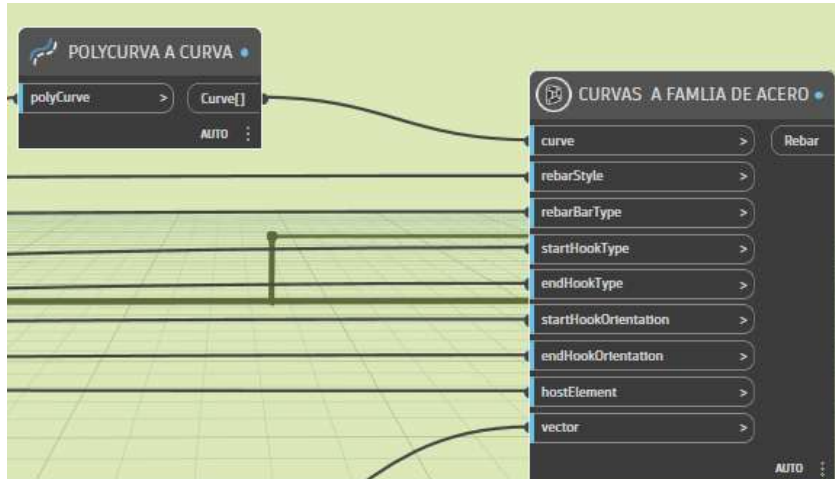


Figura 6.4-11

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b)

6.4.5. Bloque de Nodos para refuerzo negativo

El acero negativo se separa en dos bloques de nodos, uno para lado derecho de la viga y otro para lado izquierdo, por simplicidad y similitud de los dos códigos, en este trabajo se explicará solo uno.

6.4.6. Descripción de bloque de nodos para acero negativo Vigas.

Para el acero negativo se descompone el estribo superior del inicio en puntos como se ve a continuación.

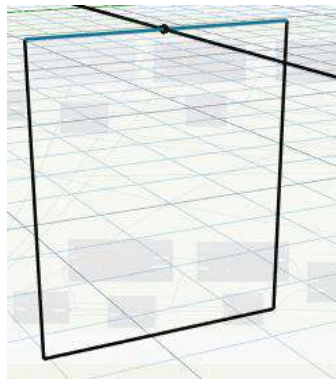


Figura 6.4-12

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Luego este estribo se divide en segmentos, los cuales serán el numero de varillas negativas que requiera el usuario, estos puntos se extruyen a una longitud de $L/3$ hacia dentro de la viga con un vector negativo como se muestra a continuación.

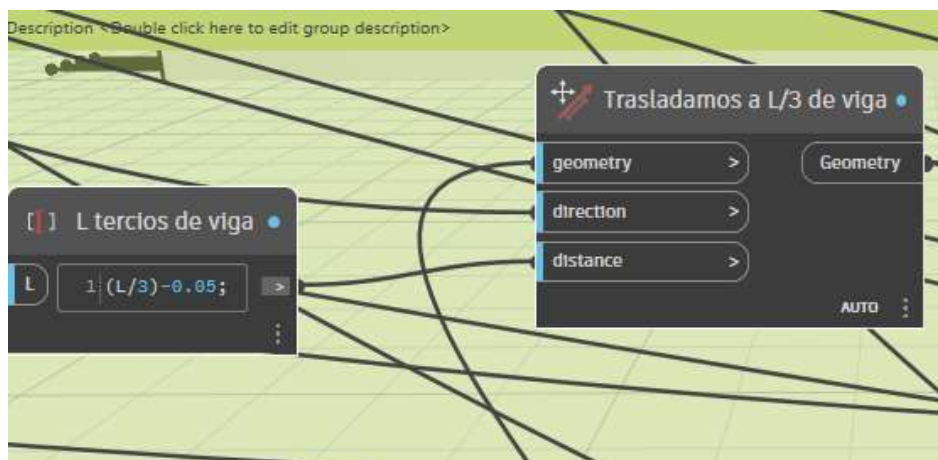


Figura 6.4-13

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

La distancia se marca con un code block que en función de la longitud libre de la viga de la distancia correcta a $L/3$.

6.4.7. Bloque de Nodos para Longitudes de desarrollo.

Los ganchos y longitudes de desarrollo se calcularon en base a las disposiciones de la ACI (American Concrete Institute, 2019)

Especificadas en la sección 5.4.1 de este trabajo.

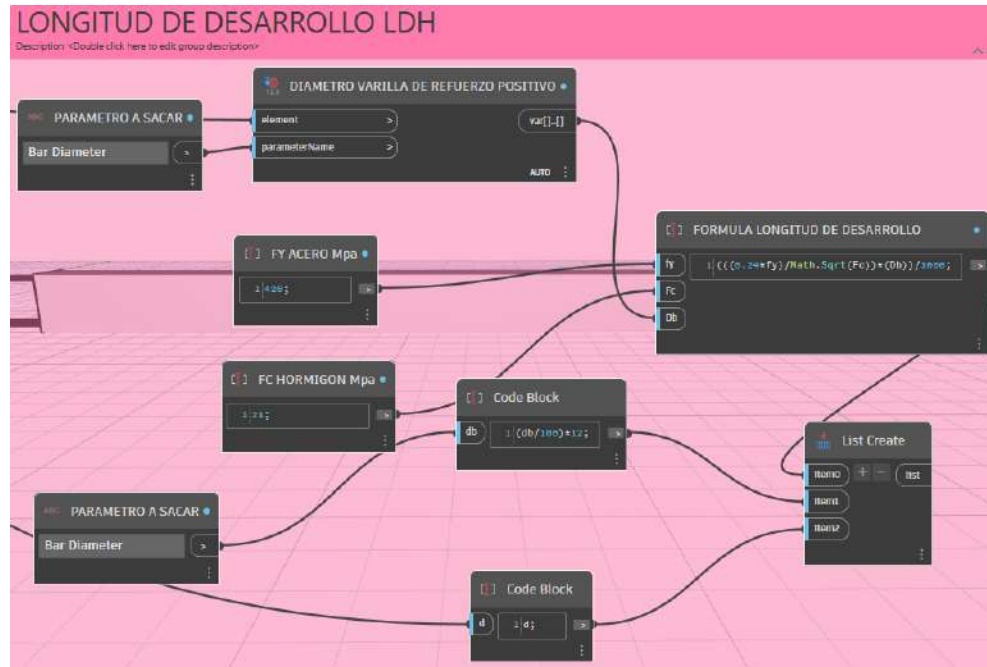


Figura 6.4-14

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

En este bloque de nodos podemos observar que se crea una lista con los 3 valores especificados por la ACI (American Concrete Institute, 2019) siendo, 12 veces el diámetro de la varilla, H de la viga o la fórmula de longitud de desarrollo especificada en la sección 5.4.1 de este trabajo.

$$\left(\frac{f_y \cdot \psi_t \cdot \psi_e \cdot \psi_s}{2.1 \cdot \lambda \cdot \sqrt{f'_c}} \right) d_b$$

American Concrete Institute. (2019). *Building code requirements for structural concrete (ACI 318-19) and commentary* (p. 308). ACI.

Después a través de un condicional seleccionamos el valor mayor y lo asignamos a una extrusión a todas las barras creadas para que su punto de unión sea dentro de la columna.



Nota: Imagen de autoría propia extraída de código en Dynamo

6.4.8. Bloque de nodos para ganchos.

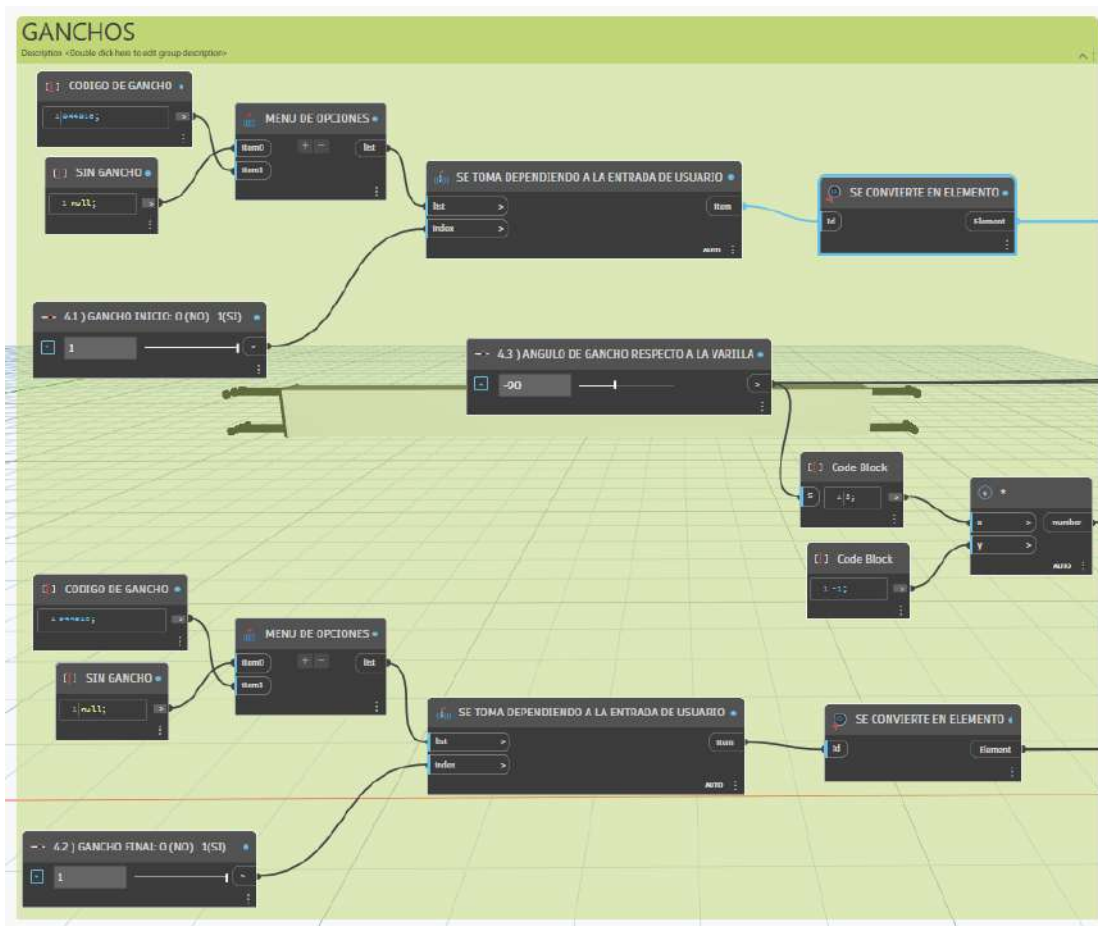


Figura 6.4-15

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

Para los ganchos se toma la información de id del gancho deseado, en el caso del gancho a noventa grados es “944918”, esto se convierte en elemento y se lo rota con el nodo rótese, con un valor angular deseado por el usuario, una vez el gancho rotado se añade a la familia estructural para hacerlo sólido.

6.4.9. Resultado total de scripts viga y columna.

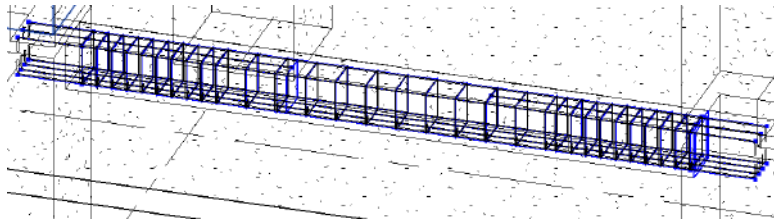


Figura 6.4-16

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

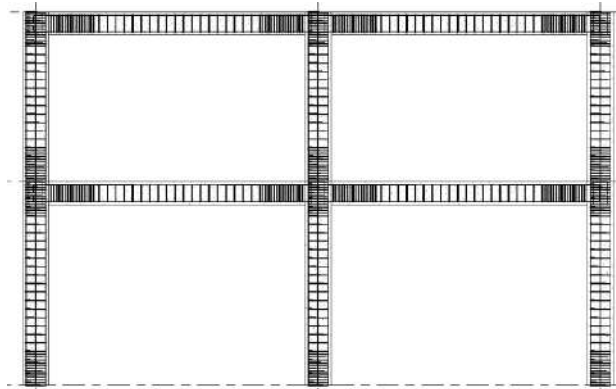
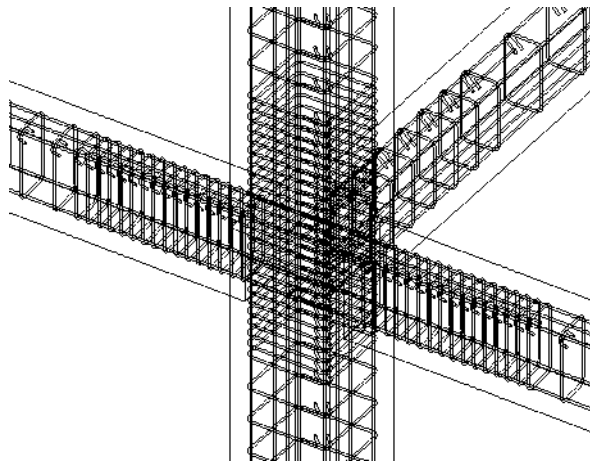


Figura 6.4-17

Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).



Nota: Imagen de autoría propia extraída de código en Dynamo (Autodesk Inc., 2024b).

6.5. Reproductor Dynamo Player.

En Dynamo (Autodesk Inc., 2024b) 2024 incluido en REVIT(Autodesk Inc., 2024d) se añade automáticamente todos los scripts realizados en la interfaz de Dynamo a Dynamo player (Autodesk Inc., 2024b), dejando como entradas (in put) los valores especificados por el creador del código visual.

6.6. Manual de uso Dynamo Player scripts.

6.6.1. Descarga de archivo Dynamo. dyn

Como primer paso esta descargar el archivo. dyn en la computadora del usuario, aparece de la siguiente manera:

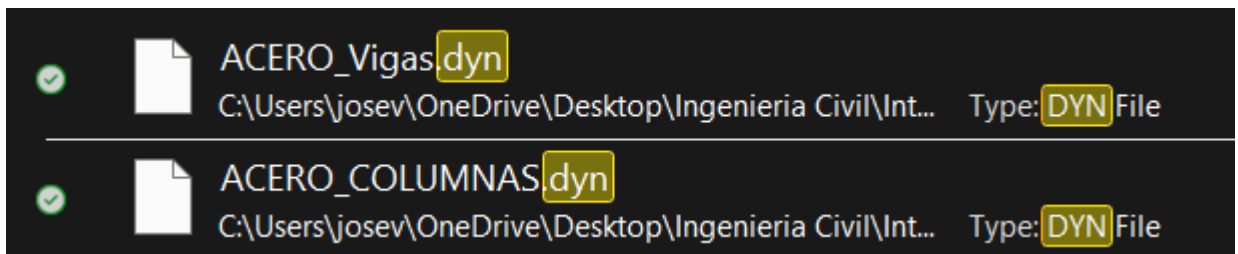


Figura 6.6-1

Nota: Imagen de autoría propia

6.6.2. Abrir el archivo Dynamo en REVIT 2024.

Una vez descargado el archivo, se pulsa en la ventana Manage ubicada en la parte superior del programa.

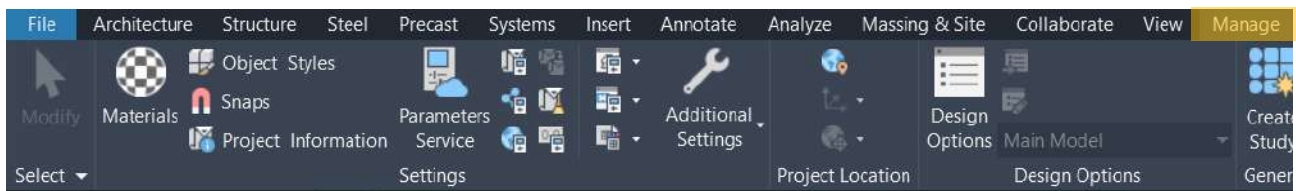


Figura 6.6-2

Nota: Imagen de autoría propia.

Una vez en manage se pulsa en la parte izquierda de la pantalla donde dice Dynamo (Autodesk Inc., 2024b).

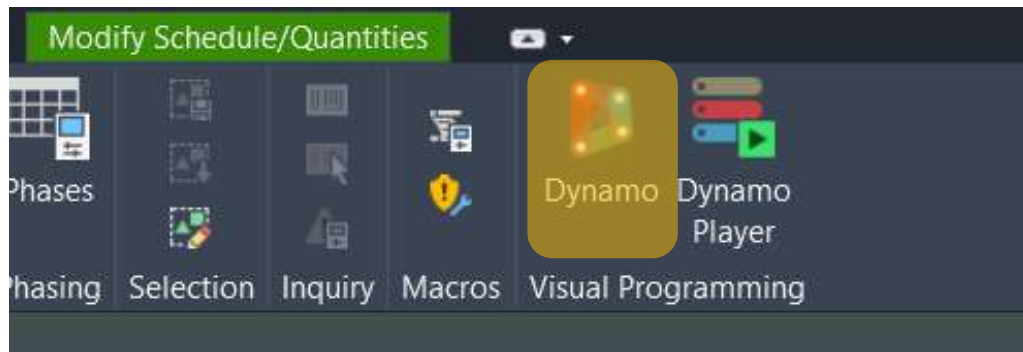


Figura 6.6-3

Nota: Imagen de autoría propia

Una vez abierto el Dynamo (Autodesk Inc., 2024b) se desplegará la siguiente ventana, donde se deberá seleccionar la opción Open.

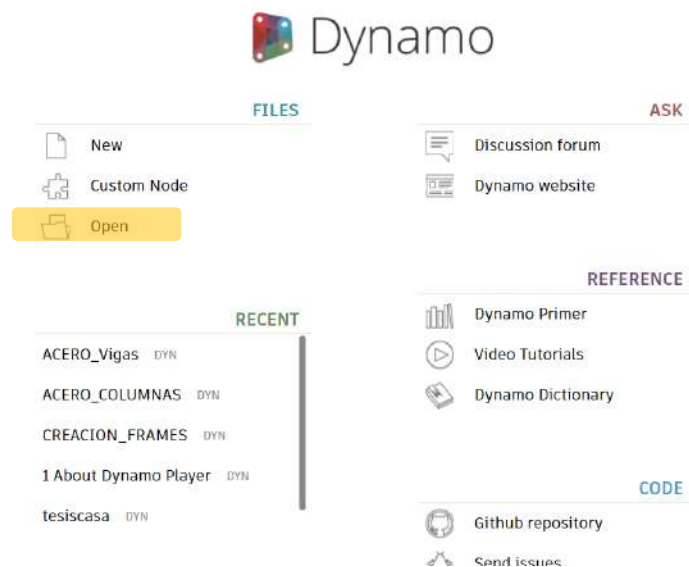


Figura 6.6-4

Nota: Imagen de autoría propia

Después se abre el archivo en la carpeta donde se haya guardado, esto abrirá el código visual de Dynamo (Autodesk Inc., 2024b) directamente y se desplegará de la siguiente manera:

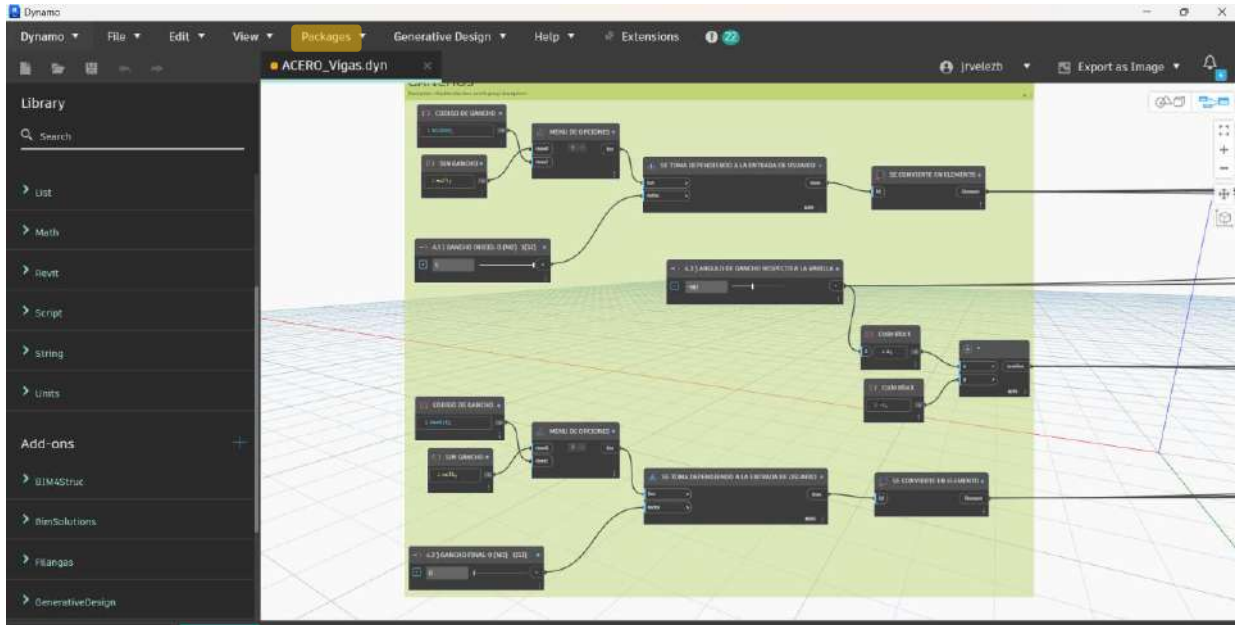


Figura 6.6-5

Nota: Imagen de autoría propia

En esta parte es NECESARIO instalar el paquete estructural que convierte las polycurvas a aceros de la siguiente manera: 1) vamos a la pestaña superior donde dice “Packages”, de donde se desplegará la siguiente ventana:

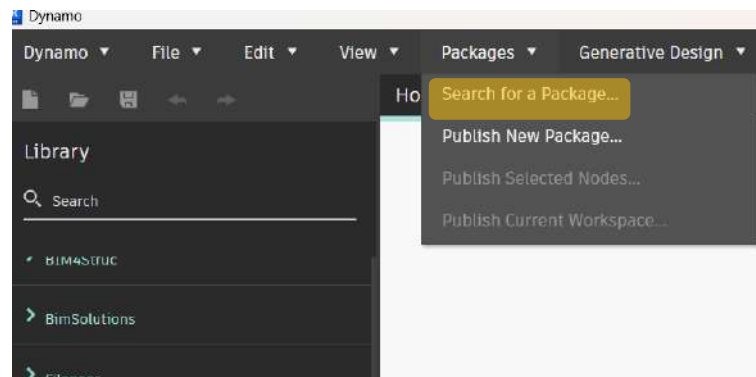


Figura 6.6-6

Nota: Imagen de autoría propia

Después en la ventana desplegada buscamos el siguiente paquete de nodos y lo instalamos:

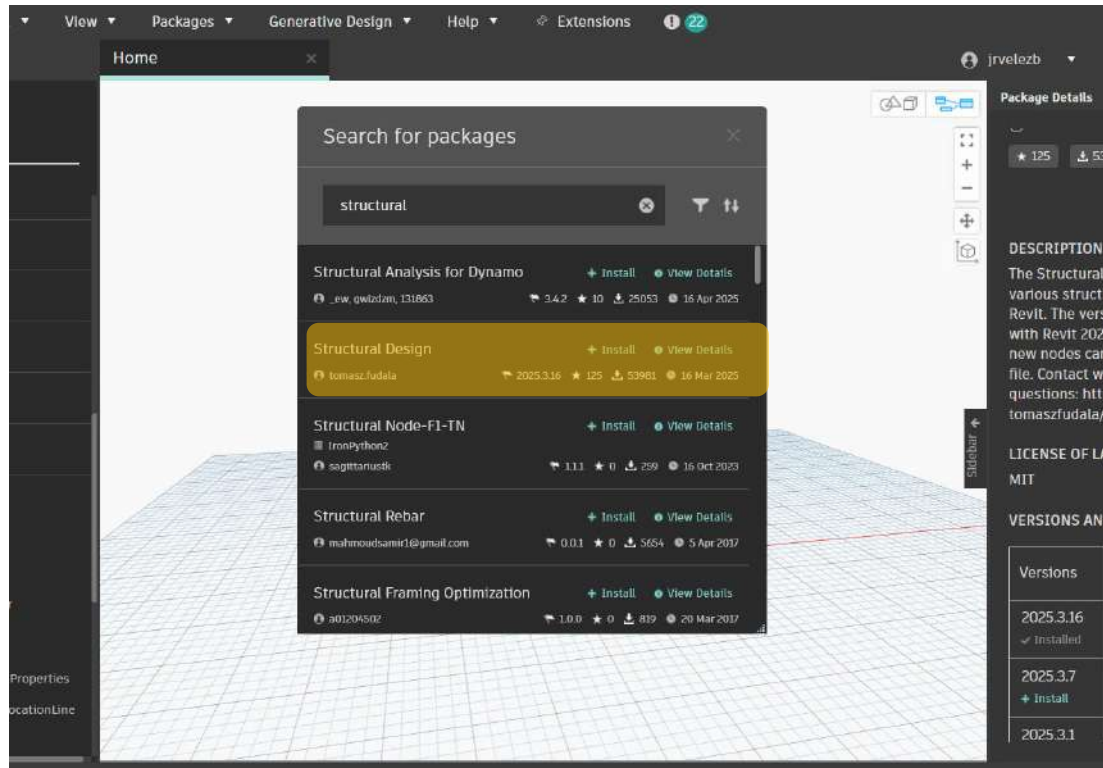


Figura 6.6-7

Nota: Imagen de autoría propia

El paquete se llama Structural Design y el nombre del autor es tomasz.fudala, una vez ubicados damos en install y lo instalamos. Una vez instalado no se necesita modificar nada más del código o realizar alguna otra instalación, se continua con el siguiente paso.

Nota: Imagen de autoría propia

Para que el interpretador lea los scripts en Dynamo Player el usuario debe ingresar los scripts de viga y columna, en la carpeta seleccionada:

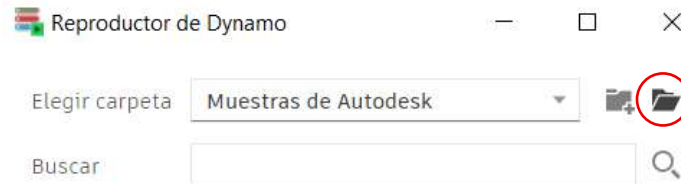


Figura 6.6-10

Nota: Imagen de autoría propia

Ejemplo:

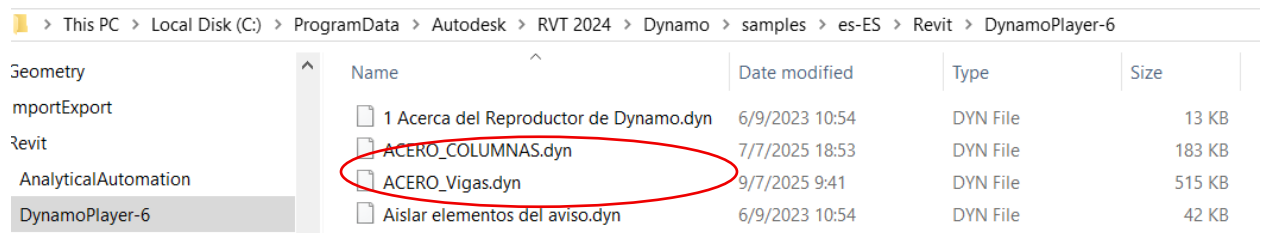


Figura 6.6-11

Nota: Imagen de autoría propia

Una vez guardadas los scripts se desplegarán automáticamente en la pantalla:

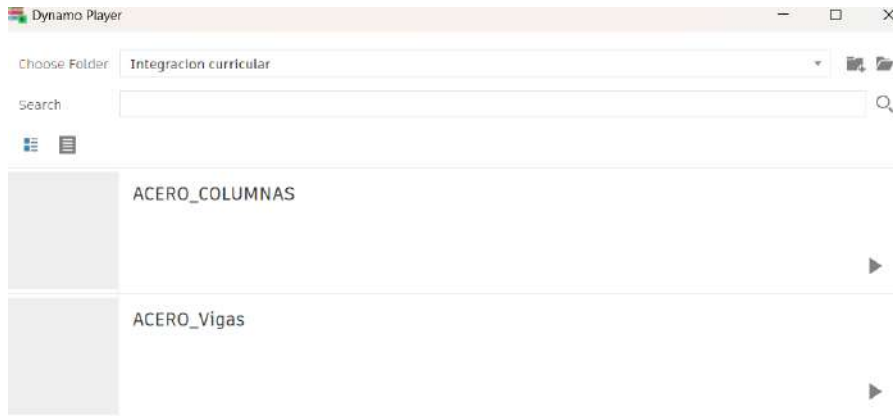


Figura 6.6-12

Nota: Imagen de autoría propia

6.6.4. Uso de Dynamo Player para vigas.

DESCRIPCION DE DATOS DE ENTRADA:

Ejes Locales

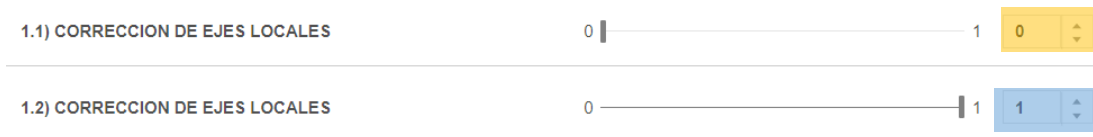


Figura 6.6-13

Nota: Imagen de autoría propia

En el Dynamo (Autodesk Inc., 2024b) para generar el acero negativo y positivo se usan los ejes locales del elemento, lo cual depende de cómo se dibujó los elementos en Revit (Autodesk Inc., 2024d), por lo tanto, hay la posibilidad de que el elemento tenga ejes contrarios a los programados, por lo cual el acero negativo se dibujara en la cara inferior de la viga y el acero positivo en el superior, si sucede esto se debe alterar el orden de los ejes locales, es decir intercalar los valores de 0 y 1 a 1 y 0. (Es poco común necesitar la corrección).

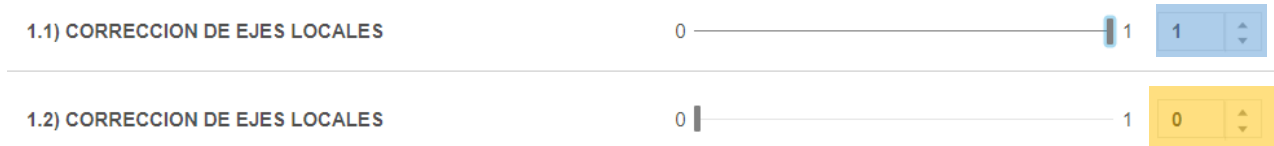


Figura 6.6-14

Nota: Imagen de autoría propia

Definición de familias y cantidades de acero longitudinal.

2.1) DIAMETRO LONGITUDINAL EN 4 ESQUINAS (ESCRITO TAL CUAL)	10M
2.2) DIAMETRO REFUERZO NEGATIVO (ESCRITO TAL CUAL)	10M
2.3) N BARRAS DE REFUERZO NEGATIVO	0 4 1
2.4) DIAMETRO REFUERZO POSITIVO (ESCRITO TAL CUAL)	10M
2.5) N BARRAS DE REFUERZO POSITIVO	0 4 2

Figura 6.6-15

Nota: Imagen de autoría propia

Los diámetros de aceros se deben escribir tal cual se llama la familia dentro del Revit (Autodesk Inc., 2024d) en este ejemplo esta es una varilla de 10M y se puede observar que es el mismo nombre escrito tal cual como se llama en Revit (Autodesk Inc., 2024d).

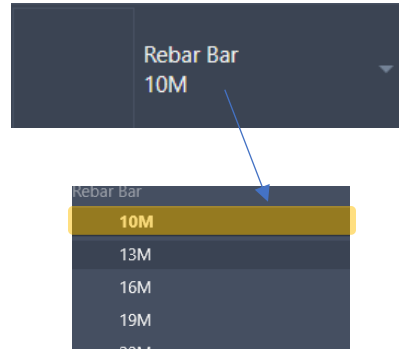


Figura 6.6-16

Nota: Imagen de autoría propia

Después de tener definido el diámetro que se requiere en el number slider se pone el valor de varillas que se desean poner tanto para refuerzo positivo, como para negativo.



Figura 6.6-17

Nota: Imagen de autoría propia

Definición de familias y cantidades de acero Transversal.

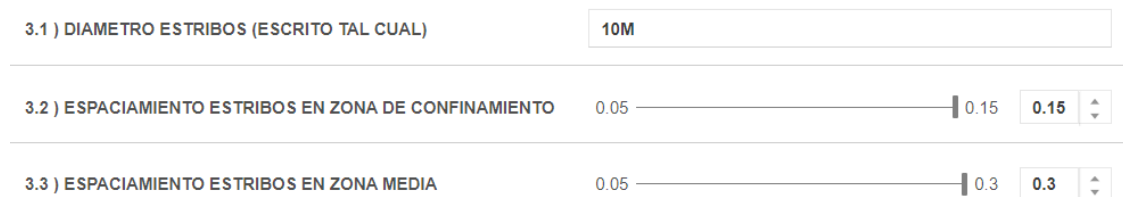


Figura 6.6-18

Nota: Imagen de autoría propia

De igual manera como en el acero longitudinal se tiene que definir la familia del diámetro de estribo tal cual se llame dentro del Revit (Autodesk Inc., 2024d) en este caso también de 10M.

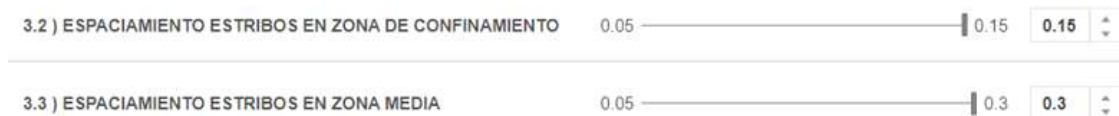


Figura 6.6-19

Nota: Imagen de autoría propia

En el Number Slider definimos el espaciamiento en metros tanto de la zona de confinamiento como de la separación de la Zona media de la viga.

Definición de ganchos y recubrimiento



Nota: Imagen de autoría propia

Para los ganchos definimos si queremos que se haga al inicio, al final, ambos lados o ningún lado en caso de que sea viga de vano medio. Como se define en el nombre de la entrada colocar el significado de los números 0 NO y 1 SI, así mismo como en el caso de ejes locales puede que el inicio y el final se intercalen desde nuestra perspectiva de vista en Revit por lo cual es recomendable revisar e intercalar valores de ser necesario.

El ángulo de gancho tiene un number slider que va de -360 a 360 grados respecto a la normal del plano de la cara de viga por lo cual se puede poner el ángulo deseado de los ganchos al final del acero.

Selección de Elemento

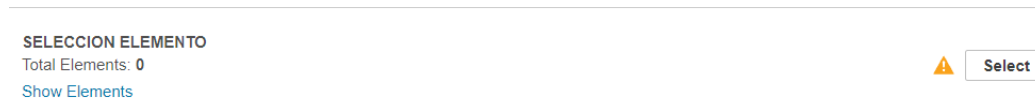


Figura 6.6-20

Nota: Imagen de autoría propia

Por último, se selecciona el elemento en Revit a donde quedamos aplicar todos los parámetros especificados, se aprieta el botón select, se puede minimizar el dynamo player, seleccionamos con cuidado la viga en el modelo de Revit, se vuelve al Dynamo Player y se pone Run.

6.6.5. Manual de uso en Dynamo Player Columnas


1.1) DIAMETRO LONGITUDINAL (ESCRITO TAL CUAL)	<input type="text" value="10M"/>
1.2) NUMERO DE VARILLAS LONGITUDINALES	4 <input type="range" value="10"/> 22 <input type="text" value="10"/>
2.1) DIAMETRO TRANSVERSAL (ESCRITO TAL CUAL)	<input type="text" value="10M"/>
2.2) ESPACIAMIENTO EN ZONA DE CONFINAMIENTO (m)	0.05 <input type="range" value="0.05"/> 0.15 <input type="text" value="0.05"/>
2.3) ESPACIAMIENTO ZONA MEDIA (m)	0.05 <input type="range" value="0.15"/> 0.3 <input type="text" value="0.15"/>
3.1) Recubrimiento (m)	0.02 <input type="range" value="0.04"/> 0.1 <input type="text" value="0.04"/>
SELECCION ELEMENTO Total Elements: 0 Show Elements	
 <input type="button" value="Select"/>	

Figura 6.6-21

Nota: Imagen de autoría propia

DESCRIPCION DE DATOS DE ENTRADA

Definición de familias y cantidades de acero longitudinal.

1.1) DIAMETRO LONGITUDINAL (ESCRITO TAL CUAL)	<input type="text" value="10M"/>
1.2) NUMERO DE VARILLAS LONGITUDINALES	4 <input type="range" value="10"/> 22 <input type="text" value="10"/>

Figura 6.6-22

Nota: Imagen de autoría propia

De igual manera como en vigas en el acero longitudinal se tiene que definir la familia del diámetro de estribo tal cual se llame dentro del Revit en este caso también de 10M y el número de varillas longitudinales se especifican en el number slider

2.1) DIAMETRO TRANSVERSAL (ESCRITO TAL CUAL)	<input type="text" value="10M"/>
2.2) ESPACIAMIENTO EN ZONA DE CONFINAMIENTO (m)	0.05 <input type="range" value="0.05"/> 0.15 <input type="text" value="0.05"/>
2.3) ESPACIAMIENTO ZONA MEDIA (m)	0.05 <input type="range" value="0.15"/> 0.3 <input type="text" value="0.15"/>

Nota: Imagen de autoría propia

De igual manera como en vigas en el acero transversal se tiene que definir la familia del diámetro de estribo tal cual se llame dentro del Revit en este caso también de 10M, los espaciamientos se especifican en el number slider.

Selección de Elemento

SELECCION ELEMENTO

Total Elements: 0

Show Elements



Select

Figura 6.6-23

Nota: Imagen de autoría propia

Por último, se selecciona el elemento en Revit a donde quedamos aplicar todos los parámetros especificados, se aprieta el botón select, se puede minimizar el dynamo player, seleccionamos con cuidado la columna en el modelo de Revit, se vuelve al Dynamo Player (Autodesk Inc., 2024b) y se pone Run.

Comentarios y criterios de diseño

-La zona de confinamiento en vigas está definida como $2 \cdot h$ de la viga, siendo h la altura de viga.

-La longitud de acero negativo en la viga está definida desde el centro de columna hasta $L_n/3$ de viga, siendo L_n la luz libre de viga.

-La zona de confinamiento de columna está definida como $L/6$ de columna siendo L la longitud total de columna.

-Los estribos empiezan a 5cm de cara de apoyo de la viga.

-Los aceros NO son continuos, es decir en vigas continuas los aceros terminan en mitad de columna lo que da aspecto de una varilla continua, pero en realidad son dos elementos.

-Los ejes locales y parámetros de gancho pueden alterar su orden, en ese caso será necesario ajustar.

CAPÍTULO VII

7. CONCLUSIONES Y RECOMENDACIONES.

7.1. Conclusiones

Para el desarrollo de esta tesis se planteó una edificación de dos niveles con sistema estructural a porticado de hormigón armado, ubicada en el sector La Mariscal, Quito. El proceso de análisis y diseño se ejecutó bajo los lineamientos de las normativas NEC 2015 y NEC 2024 (borrador), complementadas por los criterios técnicos del ACI 318-19, integrando herramientas digitales como ETABS, Python, Dynamo y Revit.

La metodología seguida abarcó desde el predimensionamiento manual, la definición del espectro de respuesta elástica (UHS), hasta el modelado estructural tridimensional y el análisis mediante el método modal espectral y fuerza lateral equivalente. Posteriormente, se desarrollaron scripts en Python para automatizar el diseño de acero a flexión y corte, los cuales fueron integrados en Revit mediante programación visual en Dynamo, logrando así la generación automática del detallado estructural, respetando los requisitos mínimos normativos.

Al finalizar este proceso de diseño integral, se concluye que:

- El uso combinado de Python y Dynamo en Revit permite automatizar el detallado estructural de elementos de hormigón armado de manera eficiente, reduciendo tiempos de dibujo y errores humanos.

- Las cuantías de refuerzo obtenidas para vigas y columnas, tanto a flexión como a corte, cumplen con las exigencias establecidas en la NEC y el ACI, garantizando el desempeño sísmico esperado para la zona sísmica tipo V.

- El espectro de respuesta generado con parámetros de suelo tipo D y un valor de PGA de 0.511g proporcionó una base adecuada para el análisis sísmico, permitiendo una evaluación realista del comportamiento dinámico de la estructura.

- El diseño por capacidad aplicado a columnas permitió asegurar el mecanismo de falla deseado en estructuras dúctiles, ubicando la disipación de energía en las vigas, tal como exigen los sistemas de “marcos especiales sismorresistentes”.

- La automatización del diseño permitió generar configuraciones de armado reales con varillas comerciales, respetando recubrimientos y separaciones mínimas, listas para su implementación directa en obra o taller.

- El flujo de trabajo propuesto demuestra que es posible integrar análisis, diseño y documentación dentro de un entorno digital completamente automatizado, aplicable a estructuras sencillas en contexto ecuatoriano, con posibilidades claras de escalar a proyectos de mayor complejidad.

La incorporación de metodología BIM en conjunto con automatización por código representa un avance sustancial para el diseño estructural en Ecuador, ya que permite mantener el cumplimiento normativo mientras se optimizan procesos repetitivos como el detallado de refuerzo.

Este enfoque no solo mejora la productividad del ingeniero, sino que también facilita la comprensión del proyecto por parte del personal de obra, al ofrecer una representación visual clara y precisa del armado estructural.

7.2. Recomendaciones

Se recomienda continuar el desarrollo de scripts personalizados en Python, ampliando su funcionalidad para incluir otros elementos estructurales como muros, losas, escaleras o conexiones, con el objetivo de lograr una automatización más completa y aplicable a proyectos reales.

Es fundamental que la automatización del diseño estructural no reemplace el criterio técnico del ingeniero. Aunque los scripts agilizan procesos, siempre se debe realizar una validación manual de los resultados, especialmente en zonas críticas o estructuras con geometría compleja.

Se sugiere estandarizar un flujo de trabajo entre el software comercial, Revit, Dynamo y Python, de forma que se pueda replicar en oficinas de diseño estructural como parte de una metodología moderna de trabajo, facilitando la documentación, coordinación y revisión.

Para futuros proyectos se recomienda incluir un análisis comparativo de tiempos y costos entre el proceso automatizado y el proceso tradicional, para cuantificar con datos concretos el beneficio real de implementar estas herramientas en el diseño estructural.

También se recomienda validar los resultados arrojados por los scripts mediante hojas de cálculo estructurales o software alternativo, para asegurar que los valores generados se ajusten a los criterios normativos de diseño sismorresistente vigentes en el país.

BIBLIOGRAFÍA

- American Concrete Institute. (2019). *ACI 318-19: Building Code Requirements for Structural Concrete and Commentary*.
https://www.concrete.org/store/productdetail.aspx?ItemID=318U19&Language=English&Units=US_Units
- American Society of Civil Engineers. (2017). *Minimum Design Loads and Associated Criteria for Buildings and Other Structures* (ASCE/SEI 7-16). American Society of Civil Engineers.
- Autodesk Inc. (2024a). *AutoCAD* (2024). Autodesk Inc.
<https://www.autodesk.com/products/autocad/overview>
- Autodesk Inc. (2024b). *Dynamo for Revit* (2.19). Autodesk Inc. <https://dynamobim.org/>
- Autodesk Inc. (2024c). *Navisworks*. Autodesk Inc.
<https://www.autodesk.com/products/navisworks/overview>
- Autodesk Inc. (2024d). *Revit* (2025). Autodesk Inc. <https://www.autodesk.com/products/revit/overview>
- Celi, C., Palacios, P., & Poveda, J. (2024, April 15). *Seismic Disaggregation Tool for Ecuador 2024 (SDTE-2024)*. TorreFuerte.Ec.
- Computers and Structures, Inc. (2023). *ETABS (Version 20.3.0)* (20.3.0). Computers and Structures, Inc.
- Econova Institute of Architecture and Engineering. (2022). *Qué es y para qué sirve Dynamo*.
<https://econova-institute.com/que-es-y-para-que-sirve-dynamo/>
- Gobierno del Ecuador. (2021, June 29). *Aprobación de planos de una construcción | Ecuador - Guía Oficial de Trámites y Servicios*. Guía Oficial de Trámites y Servicios - Gobierno del Ecuador.
<https://www.gob.ec/gadmg/tramites/aprobacion-planos-construccion>
- Graphisoft. (2024). *ArchiCAD*. Graphisoft SE.
- Guerra Avendaño, M., & Guerra Valladares, M. D. (2020). *Diseño Sismo Resistente de Edificios de Hormigón Armado Utilizando ETABS* (Primera edición). Proyectos Estructurales.
- Gutiérrez, B. R. (2022, October 15). *Conoce las fases del proyecto arquitectónico y los plazos para cada una*. <https://www.blancarosagutierrez.com/conoce-las-fases-del-proyecto-arquitectonico-y-los-plazos-para-cada-una/>
- MathWorks. (2024). *MATLAB (Version R2024a)* (R2024a). The MathWorks, Inc.
- Ministerio de Desarrollo Urbano y Vivienda. (2024). *Borrador NEC-SE-DS 2024 - Diseño Sísmico: Vol. Borrador*. MIDUVI.
- Ministerio de Desarrollo Urbano y Vivienda (MIDUVI). (2015a). Norma Ecuatoriana de la Construcción NEC-SE-CG: Cargas (No Sísmicas). In *Ministerio de desarrollo Urbano y vivienda*.
<https://www.habitatyvivienda.gob.ec/documentos-normativos-nec-norma-ecuatoriana-de-la-construccion/>
- Ministerio de Desarrollo Urbano y Vivienda (MIDUVI). (2015b). *Norma Ecuatoriana de la Construcción NEC-SE-DS: Peligro Sísmico y Diseño Sismorresistente*.
<https://www.habitatyvivienda.gob.ec/documentos-normativos-nec-norma-ecuatoriana-de-la-construccion/>
- Ministerio de Desarrollo Urbano y Vivienda (MIDUVI). (2015c). Norma Ecuatoriana de la Construcción NEC-SE-HM: Diseño y construcción en hormigón armado. In M. Dirección de Comunicación Social (Ed.), *Norma Ecuatoriana de la Construcción*. MIDUVI.
<https://www.habitatyvivienda.gob.ec/documentos-normativos-nec-norma-ecuatoriana-de-la-construccion/>
- Python Software Foundation. (2023). *Python 3.8* (3.8). Python Software Foundation.
<https://www.python.org/>
- Zigurat Global Institute of Technology. (2018, July 17). *Cuál software BIM debo utilizar en proyectos de arquitectura?* <https://www.e-zigurat.com/es/blog/que-software-bim-debo-utilizar/>