



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**DESARROLLO DE HERRAMIENTA DE
SOFTWARE PARA LA
AUTOMATIZACIÓN DEL DISEÑO DE
MUROS PANTALLA**

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Hugo Adrian Meza Vega

ASESORA DE INFORME

M.I. Margarita Muñoz Goncen



Ciudad Universitaria, Cd. Mx., 2024

Contenido

Introducción	8
Objetivo general	8
Objetivos específicos	9
Consideraciones	9
1. Antecedentes	10
2. Marco teórico	13
2.1. Muro de contención para excavaciones profundas: muro pantalla	13
2.2. Ingeniería de software	15
2.3. Producto de software	16
2.4. Ciclo de vida del software	18
2.5. Arquitectura de Software	19
2.5.1. Arquitectura de tubería y filtro	20
2.6. Metodologías de desarrollo de software	21
2.7. Algoritmos	22
2.7.1. Definición de algoritmo	22
2.7.2. Algoritmo Genético	24
2.7.3. PyGAD	26
2.8. Software de diseño y dibujo asistido por computadora	27
2.8.1. Software CAD	27
2.8.2. Rhinoceros 3D	28
2.8.3. BIM	28
2.8.4. Tekla Structures	29
2.9. Lenguajes de programación	29
2.9.1. Lenguaje de Programación Visual (VPL)	29
2.9.1.1. Grasshopper de Rhinoceros	30
2.9.2. Python	30
3. Definición del problema	32
4. Metodología de desarrollo de la herramienta	33
4.1. Análisis	36
4.2. Planeación y definición	37
4.3. Diseño de arquitectura de programa	40
4.4. Desarrollo	42
4.4.1. Ambiente de desarrollo	42
4.4.2. Creación de filtros de la arquitectura	43
4.4.3. Implementación de algoritmo genético para optimización	46
4.4.4. Construcción de algoritmos solucionadores de refuerzos	52
4.5. Pruebas	58

4.5.1. Prueba 1	59
4.5.1.1. Área de acero de diseño - Prueba 1	59
4.5.1.2. Solución automática de bastones - Prueba 1	61
4.5.1.3. Solución automática de estribos - Prueba 1	62
4.5.2. Prueba 2	65
4.5.2.1. Área de acero de diseño - Prueba 2	65
4.5.2.2. Solución automática de bastones - Prueba 2	67
4.5.2.3. Solución automática de estribos - Prueba 2	68
4.6. Liberación a producción	70
4.7. Documentación y mantenimiento	71
5. Resultados	71
5.1. Comparación del tiempo de solución entre la herramienta de software y el método manual	71
5.2. Comparación de resultados entre la herramienta de software y el método manual	74
5.2.1. Áreas de acero: mínima y máxima	76
5.2.2. Propuesta de solución para acero de refuerzo principal	78
5.2.3. Cálculo de momento resistente	83
5.2.4. Búsqueda de bastones como acero de refuerzo adicional para momentos flexionantes.	86
5.2.5. Cálculo de la resistencia al corte: VCR	94
5.2.6. Búsqueda de estribos como acero de refuerzo adicional para fuerzas cortantes.	95
5.3. Mejora en la experiencia de usuario	107
5.4. Cumplimiento de los criterios mínimos de aceptación del programa	108
5.5. Solución final entregada por el programa	108
5.6. Producto final de la integración del programa con las herramientas desarrolladas por el “equipo BIM”	110
6. Participación profesional	114
7. Conclusiones	116
8. Referencias	118

Índice de figuras

Figura 1. Panorama general del proyecto BIM.	11
Figura 2. Proceso de construcción de muro pantalla [2].	13
Figura 3. Producto de software: Características [3].	16
Figura 4. Arquitectura tubería y filtro.	19
Figura 5. Modelo en cascada [5].	20
Figura 6. Vista general del modelo ágil [5].	21
Figura 7. Funcionamiento de los algoritmos genéticos [8].	24
Figura 8. Ejemplo de programación visual: Suma de dos números.	29
Figura 9. Programa Python: Creación de objeto.	30
Figura 10. Diagrama de tiempo de desarrollo del proyecto.	38
Figura 11. Planeación y Definición: Diagrama de flujo del programa.	39
Figura 12. Elementos de entrada de datos (Grasshopper).	39
Figura 13. Diagrama de arquitectura del programa (Tubería y filtro).	40
Figura 14. Plugins de Rhinoceros Grasshopper.	41
Figura 15. Código Python: utilizando Gh_CPython Plugin for Grasshopper.	42
Figura 16. Construcción del primer filtro de la arquitectura de programa.	43
Figura 17. Programa de automatización - Componentes de la arquitectura.	44
Figura 18. Modelo 1 de algoritmo genético - acero principal de muro pantalla.	47
Figura 19. Modelo 1: mejor solución y rendimiento del Fitness.	48
Figura 20. Modelo 2 de algoritmo genético para acero principal.	48
Figura 21. Modelo 2: mejor solución y rendimiento del Fitness por generación.	49
Figura 22. Modelo 3 de algoritmo genético para acero principal.	49
Figura 23. Modelo 3: mejor solución y rendimiento del Fitness.	50
Figura 24. Modelo 4 de algoritmo genético para acero principal.	50
Figura 25. Modelo 4: mejor solución y rendimiento del Fitness por generación.	51
Figura 26. Ejemplo de funcionamiento de muro pantalla [25].	52
Figura 27. Representación de muro pantalla en servicio [2].	52
Figura 28. Representación de muro pantalla en servicio [2].	53
Figura 29. Gráfica de fuerzas cortantes actuantes.	54
Figura 30. Reconocimiento de zonas que requieren bastones como refuerzo adicional para momentos flexionantes.	55
Figura 31. Búsqueda iterativa y colocación de bastones.	56
Figura 32. Búsqueda iterativa y colocación temporal de estribos.	57

Figura 33. Análisis de distribución y colocación final de estribos.	57
Figura 34. Propiedades de muro pantalla para prueba 1.	59
Figura 35. Prueba 1: resultado de área de acero.	59
Figura 36. Prueba 1: Gráfica de rendimiento del parámetro Fitness.	60
Figura 37. Prueba 1: Solución automática de bastones.	60
Figura 38. Prueba 1: Búsqueda de estribos mediante AG.	61
Figura 39. Prueba 1: Rendimiento del Fitness - Estribos lado negativo.	62
Figura 40. Prueba 1: Rendimiento del Fitness - Estribos lado positivo.	63
Figura 41. Prueba 1: Análisis distribución y solución final de estribos.	64
Figura 42. Propiedades de muro pantalla para prueba 2.	65
Figura 43. Prueba 2: resultado de área de acero.	65
Figura 44. Prueba 2: Gráfica de rendimiento del parámetro Fitness.	66
Figura 45. Prueba 2: Solución automática de bastones.	66
Figura 46. Prueba 2: Búsqueda de estribos mediante AG.	67
Figura 47. Prueba 2: Rendimiento del Fitness - Estribos lado negativo.	68
Figura 48. Prueba 2: Rendimiento del Fitness - Estribos lado positivo.	68
Figura 49. Prueba 2: Análisis y solución final de estribos.	69
Figura 50. Comparación de resultados: datos de entrada en herramienta de software.	74
Figura 51. Comparación de resultados: especificación de rutas en herramienta de software.	74
Figura 52. Comparación de resultados: datos de entrada en hoja de excel del método manual.	75
Figura 53. Comparación de resultados: área mínima y máxima de acero proporcionadas por la herramienta de software.	76
Figura 54. Comparación de resultados: área mínima y máxima de acero proporcionadas por método manual.	76
Figura 55. Comparación de resultados: mejor solución de acero de refuerzo principal encontrado por herramienta de software.	78
Figura 56. Comparación de resultados: tabla de posibles soluciones encontradas por herramienta de software.	79
Figura 57. Comparación de resultados: propuesta 1 de solución para acero de refuerzo principal con el método manual.	79
Figura 58. Comparación de resultados: propuesta 2 de solución para acero de refuerzo principal con el método manual.	80

Figura 59. Comparación de resultados: propuesta 3 de solución para acero de refuerzo principal con el método manual.	81
Figura 60. Comparación de resultados: cálculo de momento resistente por herramienta de software.	82
Figura 61. Comparación de resultados: cálculo de momento resistente con el método manual.	83
Figura 62. Comparación de resultados: cálculo de momento resistente por herramienta de software (misma propuesta de acero de refuerzo principal que el método manual).	84
Figura 63. Comparación de resultados: gráficas de elementos mecánicos para momentos en servicio generadas por herramienta de software.	86
Figura 64. Comparación de resultados: solución y colocación de bastones realizado por la herramienta de software.	87
Figura 65. Comparación de resultados: solución final de bastones con el método manual.	89
Figura 66. Comparación de resultados: solución final del arreglo de bastones 1 calculado con el método manual.	90
Figura 67. Comparación de resultados: solución final del arreglo de bastones 3 calculado con el método manual.	91
Figura 68. Comparación de resultados: solución final del arreglo de bastones 4 calculado con el método manual.	91
Figura 69. Comparación de resultados: resumen de arreglos de bastones obtenidos con el método manual.	92
Figura 70. Comparación de resultados: tablas resumen de solución para arreglos de bastones obtenidos por la herramienta de software.	92
Figura 71. Comparación de resultados: VCR calculado por la herramienta de software.	94
Figura 72. Comparación de resultados: VCR calculado con el método manual.	94
Figura 73. Comparación de resultados: gráficas de fuerzas cortantes generadas por la herramienta de software.	95
Figura 74. Comparación de resultados: propuesta preliminar de estribos calculados por la herramienta de software.	97
Figura 75. Comparación de resultados: tabla de estribos calculados por la herramienta de software.	98
Figura 76. Comparación de resultados: Propuesta preliminar de estribos calculados mediante el método manual.	99
Figura 77. Comparación de resultados: solución final del arreglo de estribos 1 calculado con el método manual.	100
Figura 78. Comparación de resultados: solución final del arreglo de estribos 3 calculado con el método manual.	101

Figura 79. Comparación de resultados: solución final del arreglo de estribos 4 calculado con el método manual.	101
Figura 80. Comparación de resultados: solución final del arreglo de estribos 5 calculado con el método manual.	102
Figura 81. Comparación de resultados: solución final del arreglo de estribos 6 calculado con el método manual.	102
Figura 82. Comparación de resultados: Propuesta final de estribos calculados y ordenados por la herramienta de software.	104
Figura 83. Comparación de resultados: tabla final de estribos calculados y acomodados por la herramienta de software.	105
Figura 84. Comparación de resultados: Propuesta final de estribos calculados mediante el método manual y acomodados por el ingeniero de proyecto.	105
Figura 85. Comparación de resultados: tabla final de estribos calculados mediante el método manual y acomodados por el ingeniero de proyecto.	106
Figura 86. Imágenes de las gráficas del programa.	108
Figura 87. Archivo de salida: Hoja con información del refuerzo principal.	108
Figura 88. Archivo de salida: Hoja con información de bastones cara tierra.	109
Figura 89. Archivo de salida: Hoja con información de bastones cara excava.	109
Figura 90. Archivo de salida: Hoja con información de estribos para cortantes.	109
Figura 91. Programa 1: Mejor distribución de paneles de muro pantalla.	110
Figura 92. Obtención de la solución para acero de refuerzo de muro pantalla con programa 2.	111
Figura 93. Programa 3: Representación en 3D del muro Pantalla calculado y diseñado con el programa 1 - Vista aérea.	111
Figura 94. Programa 3: Representación en 3D del muro Pantalla calculado y diseñado con el programa 1 - Vista lateral.	112
Figura 95. Plano de taller del muro pantalla generado en Tekla Structures con la información del programa 1, 2 y 3. Calidad de imagen reducida.	113

Introducción

Durante mis estudios académicos en la Facultad de Ingeniería comencé a trabajar en una empresa constructora especializada en cimentaciones, estructuras subterráneas y obra civil a la que me referiré como “la constructora”. Desempeñé el rol de desarrollador de software, formando parte de un equipo al que llamaré “equipo BIM”, Building Information Modeling (BIM), equipo conformado por 3 células individuales dedicado al desarrollo de herramientas de software, para automatizar procesos constructivos y de diseño en el área de ingeniería, cuya finalidad fue optimizar sus procesos y el uso de materiales.

Mi trabajo consistió en llevar a cabo el desarrollo de software, así como componentes, módulos y programas de automatización en proyectos que fueron asignados al “equipo BIM”. A lo largo del tiempo que he desempeñado mis actividades laborales, he aplicado los conocimientos adquiridos durante mi trayectoria académica en la Facultad de Ingeniería habiendo cursado materias de los campos de ingeniería de software y tecnologías de la información y comunicaciones.

En este informe presento el desarrollo de mi primer proyecto en “la constructora”, donde desarrollé una herramienta de software con el propósito de automatizar el proceso de diseño estructural de muros pantalla. Proyecto de suma importancia puesto que la automatización de este proceso ahorra en promedio 82% de horas de trabajo, tiempo empleado. Durante el desarrollo del proyecto hice uso de mis conocimientos de ingeniería en computación para dar solución a los retos que se presentaron para lograr la automatización de dicho proceso a través de una herramienta de software.

Objetivo general

Describir el procedimiento, análisis, cálculo y diseño, desarrollo de software, así como los procesos de pruebas, que requirió la construcción de esta herramienta de software.

Dicha herramienta calcula el acero de refuerzo principal de muros pantalla y el acero de refuerzo adicional que los muros requieren para resistir las fuerzas y condiciones a las que estará sometido, y forma parte fundamental de mi “equipo BIM” donde el resultado del trabajo en conjunto de los tres programas, asignados al equipo, permite obtener: un diseño óptimo del acero de refuerzo del muro pantalla, generar el modelo en 3D de dichos elementos y producir de manera rápida los planos de taller para la construcción del muro pantalla en la obra.

Objetivos específicos

Para el cumplimiento del objetivo general se establecieron como objetivos específicos los siguientes:

- Optimización en el uso de acero para el refuerzo del muro pantalla (programa de cálculo y diseño).
- Reducción de la probabilidad de error por el factor humano en el cálculo y diseño del acero de refuerzo de muro pantalla (programa de cálculo y diseño).
- Disminución del tiempo de generación de soluciones óptimas del acero de refuerzo de muro pantalla (programa de cálculo y diseño).
- Construcción de algoritmos para la detección y colocación automática de acero de refuerzo adicional (bastones o refuerzos y estribos para momentos flexionantes y fuerzas cortantes (programa de cálculo y diseño).
- Estandarización de la entrada y salida de datos para la conexión entre programas (programa de cálculo y diseño y las dos herramientas complementarias desarrolladas por el “equipo BIM”).

Consideraciones

Por decisión de “la constructora”, la solución a la problemática a través de una herramienta de software, se desarrolló dentro del software CAD Rhinoceros que, a través de Grasshopper, permite hacer implementaciones con programación visual y programación orientada a objetos con los lenguajes C#, Python y Visual Basic.

La interacción con la herramienta debe ser similar a la de otras herramientas desarrolladas por el “equipo BIM”, mismas que se desarrollaron con Grasshopper de Rhinoceros, por temas de uso y experiencia de usuario para los ingenieros del área. Lo anterior implicó la estandarización de la entrada y salida de datos para lograr una correcta conexión entre la aplicación y los programas desarrollados por el “equipo BIM”.

El desarrollo de la herramienta de software se apegó a los lineamientos que estableció “la constructora”. Diseñando el programa a la medida haciendo uso de elementos de cómputo y programación para ajustar la solución a los requerimientos.

1. Antecedentes

En la actualidad, las empresas y organizaciones se están enfocando en un cambio tecnológico con el objetivo de mejorar sus procesos y optimizar sus recursos. La automatización de procesos y digitalización de la información está provocando que la ingeniería en computación se relacione con otras ingenierías con el objetivo de modernizar los sistemas de trabajo, de acuerdo con lo que he visto en mi experiencia laboral.

En “la constructora” se realizan tareas de cálculo y diseño que no han sido automatizadas, lo cual implica más tiempo de estudio y desarrollo; esto se pretende resolver poniendo en práctica la metodología BIM (Modelado de la Información para la Construcción), el cual tiene como objetivo automatizar los procesos de diseño para la generación de modelos en 3D y de ese modo tener mayor gestión de los proyectos y ofrecer una mejor experiencia de trabajo beneficiando a la organización y a sus clientes.

Respecto a “la constructora” es importante señalar que se dedica a la construcción de cimentaciones profundas, estructuras subterráneas, estructuras portuarias y obra civil. Originalmente la empresa no tenía ningún tipo de implementación BIM, puesto que conserva una forma de trabajo sistémica donde los ingenieros de proyecto realizan sus tareas de cálculo y diseño manualmente con ayuda de algunas herramientas de cálculo analíticas y de software. Recientemente “la constructora” formó un equipo dedicado a la implementación de esta forma de trabajo. En la figura 1 se muestra la planeación para lograr la implementación de la metodología BIM a través del desarrollo de software especializado.

Paso 0, “la constructora” toma la decisión de crear un equipo de desarrollo para automatizar tareas de cálculo y diseño que son realizadas en las áreas de ingeniería y dibujo. Pasos 1 y 1a, consisten en la organización del área de ingeniería y el área de dibujo para generar los diagramas de flujo y definir criterios de diseño.

En los pasos 2, 3 y 4 el área de ingeniería recopila datos de análisis de proyectos realizados por “la constructora”, que servirán de apoyo para el equipo BIM en el desarrollo de las herramientas.

Respecto a los pasos 5, 6 y 7, es asignado el desarrollo de los 3 programas que hacen posible el objetivo general, delegando la responsabilidad a cada integrante del “equipo BIM”.

El paso 8 corresponde al resultado intermedio del objetivo general, se trata del modelo 3D del muro pantalla y finalmente, en el paso 9, se genera el plano a partir del modelo 3D del muro, cumpliéndose así el objetivo general del proyecto.

MODELO BIM-LA CONSTRUCTORA

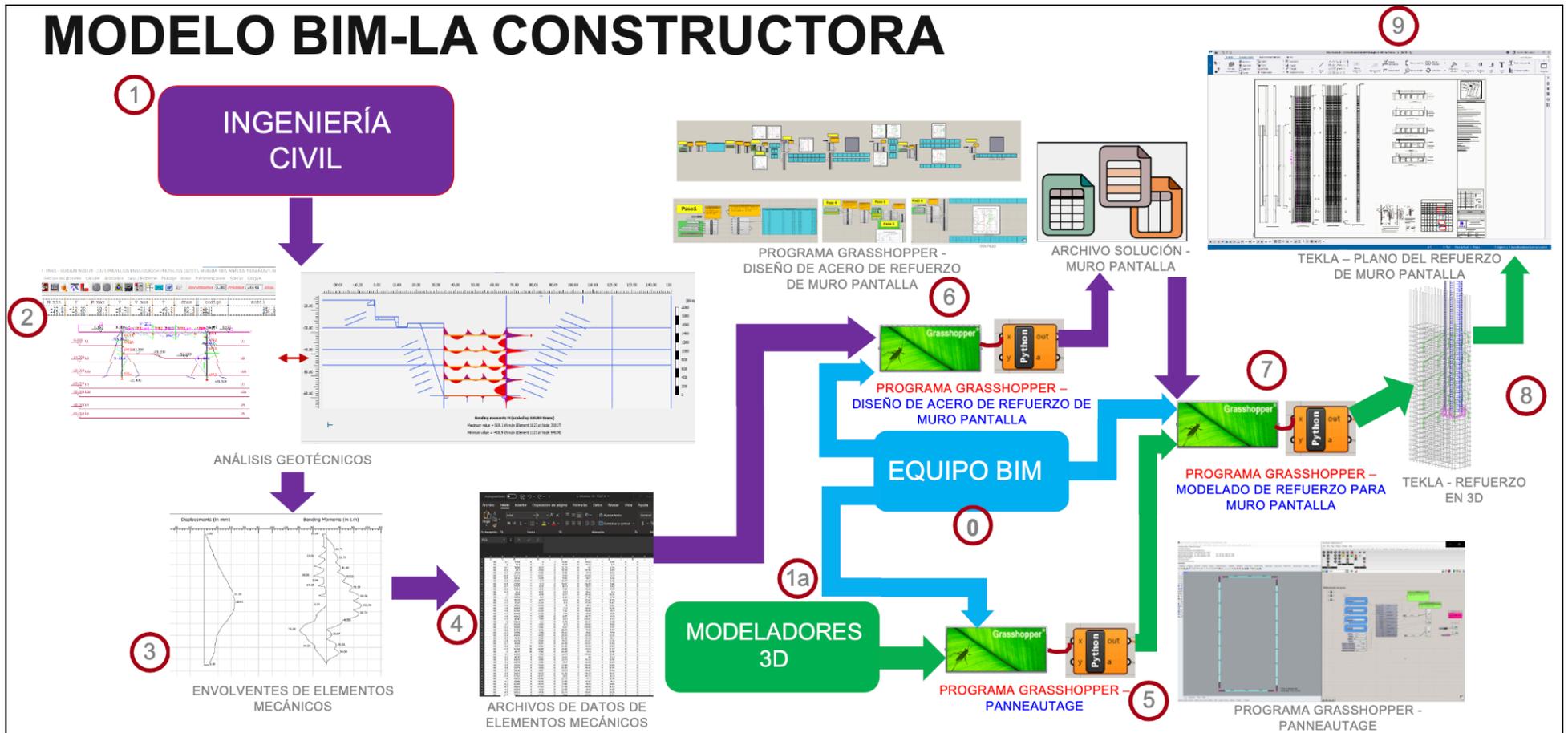


Figura 1. Panorama general del proyecto BIM

2. Marco teórico

Debido al tipo de proyecto, es importante resaltar que de acuerdo a la Universidad Autónoma de Guadalajara [1]: la ingeniería civil “es una rama de la Ingeniería donde mediante el uso de diversas técnicas se solucionan problemas para satisfacer una necesidad, se encarga de la creación de infraestructuras, obras y emprendimientos hidráulicos, también buscando que se provea una solución con el propósito de beneficiar a la sociedad a través de reducción y control de una infraestructura.”

El desarrollo de esta herramienta de software implicó crear una fuerte relación entre los conocimientos del área de ingeniería en computación e ingeniería civil. En este apartado se darán a conocer los elementos teóricos necesarios, de ambas ingenierías, para comprender a detalle el desarrollo del proyecto.

2.1. Muro de contención para excavaciones profundas: muro pantalla

El muro de contención, también conocido como muro pantalla, es una estructura de contención de geometría esbelta que facilita realizar una excavación vertical. La pantalla es una serie de vigas verticales las cuales estarán sometidas a los empujes del terreno y el agua del subsuelo. El material predominante en su construcción es el concreto reforzado [2].

La fijación de la pantalla bajo el fondo de la excavación principalmente permite:

- Equilibrar el empuje pasivo del terreno.
- Soportar cargas verticales.
- Controlar riesgos hidráulicos.

Una de las ventajas más importantes de este sistema de contención es que las pantallas pueden contener elementos verticales de refuerzo, que le proveen de resistencia a la estructura frente a la flexión y se instalan antes de proceder a la excavación. De esta forma, los elementos no interfieren con las excavaciones de la obra.

Los muros de contención están clasificados principalmente en dos tipos, pantallas continuas y pantallas discontinuas:

- Pantallas continuas: Muro pantalla, pantalla prefabricada, pantalla de tablestacas, pantalla de lechada armada, pilotes secantes, etc.

- Pantallas discontinuas: Pantalla berlinesa, pantalla parisina, pantalla luteciana, pantalla moscovita, pilotes tangentes, etc.

Muro Pantalla

Es un muro de tipo continuo, compuesto de una estructura de concreto y armado de acero, que es colado bajo el subsuelo. Su principal tarea es delimitar la periferia del terreno mediante un sistema de apuntalamiento para así poder realizar una excavación profunda en todo el predio y con ello evitar un colapso durante el proceso de excavación y construcción.

Para la construcción del muro pantalla es necesario perforar (excavar secuencialmente una trinchera lineal) en la periferia del terreno, empleando lodo para estabilizar las paredes de la excavación el tiempo suficiente para poder colocar el acero de refuerzo y el concreto. Para garantizar la estabilidad de la trinchera durante la colocación de armado de acero y concreto, se utiliza un lodo de perforación o lodo bentonítico. Este elemento se fabrica con bentonita para formar una capa estanca en las paredes del suelo, también nombrada como “cake”, con el objetivo de asegurar que el fluido de perforación no pueda colarse en el terreno y de ese modo garantizar la estabilidad de la trinchera abierta.

En la figura 2, se muestra el proceso de construcción del muro pantalla durante la ejecución en la obra civil:

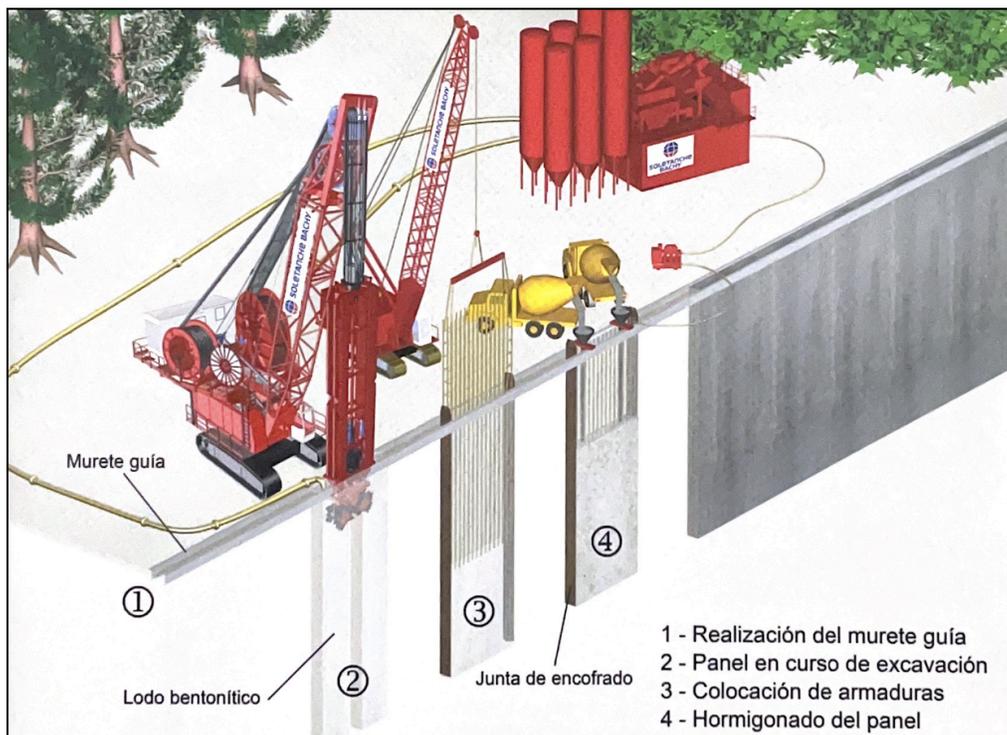


Figura 2. Proceso de construcción de muro pantalla [2].

Lo primero es realizar el murete guía o brocal en superficie. Este elemento está conformado de dos muretes paralelos de concreto armado con un ancho aproximado de 20 cm y de entre 80 y 120 cm de profundidad, los cuales permiten ensamblar las diferentes secciones del muro pantalla definiendo su trazado con precisión, también de guiar la máquina de perforación, de soporte y posicionamiento al colocar las armaduras de acero durante el vaciado del concreto.

Observaciones importantes a tener en cuenta para asegurar la estabilidad de la excavación: la perforación se realiza por etapas mediante paneles o tableros de longitudes limitadas (de 2.7 metros o bien paneles de alrededor de los 6 a 7 metros), esta longitud depende del tipo de equipo, entorno, subsuelo y la propia geometría del proyecto.

La armadura de acero, también conocida como acero de refuerzo, es diseñada por el área de ingeniería. Este proceso consiste en definir, a partir de las características del terreno y la secuencia constructiva, las cantidades de acero necesarias para que el muro resista las fuerzas y empujes a los que será sometido durante la construcción en obra civil y durante su etapa de servicio (trabajo a lo largo de su vida útil). Se estudia el diseño que debe tener la armadura o acero de refuerzo tomando en cuenta elementos mecánicos, separaciones, diámetros de varilla y refuerzo de acero adicional en zonas específicas del muro en caso de requerir mayor resistencia, tomando en consideración el propio proceso constructivo que garantice la calidad de cada sección de la pantalla.

2.2. Ingeniería de software

El software es un elemento no tangible que se encuentra en dispositivos electrónicos (computadoras, celulares inteligentes, relojes inteligentes, televisores, sistemas embebidos, etc.) destinado a realizar tareas específicas o funcionar como contenido de consumo. Principalmente es un programa de computadora que se ejecuta siguiendo una serie de algoritmos para cumplir un objetivo. [3]

Hoy en día, forma parte fundamental en las actividades diarias y ejercicio de profesiones de muchas personas. Desde sus inicios siempre existió la necesidad de gestionar el proceso de creación y su mantenimiento, pues es una herramienta en constante evolución.

Se requiere de todo un campo de estudio dedicado al manejo, gestión y mantenimiento de los productos de software.

La ingeniería de software, área de la ingeniería especializada en el diseño y desarrollo sistemáticos de productos de software, tiene como principales ideales la realización de programas informáticos que cumplan con especificaciones, estándares, metodologías que avalan buenas prácticas para garantizar un producto confiable que cumpla con su objetivo, tiempo y presupuesto.

La organización IEEE¹ define a esta rama de la ingeniería como una aplicación sistemática, disciplinada y confiable para el desarrollo y mantenimiento de los sistemas de software.

Se vuelve un campo de conocimiento muy importante en el desarrollo de nuevas tecnologías, ya que cualquier avance en la computación regularmente va acompañado del desarrollo de algún tipo de software. Para garantizar un buen producto, es necesario seguir las buenas prácticas que la ingeniería de software indica para obtener un producto de calidad.

En todo proyecto de software es necesario definir los requerimientos y objetivos del producto. Diseñar y construir la solución mientras se verifica su desarrollo.

En la ingeniería de software, existen diversos modelos y metodologías para desarrollar software que permiten llevar un control en su producción. Estos modelos están respaldados por años de trabajo en investigación, organizaciones especialistas en estándares, empresas de gran prestigio en la industria y los buenos resultados que han entregado a lo largo de su historia. Permiten llevar a cabo la división de tareas para tener una mejor gestión en la producción del producto, permitiendo medir el avance y rendimiento del proceso durante el desarrollo.

2.3. Producto de software

La naturaleza del software juega un rol dual. Es un producto y a la vez es el medio para entregar un producto de software.

- El software como producto, se encarga de proporcionar la potencia de cómputo incorporada en el hardware del sistema ya sea en un dispositivo o una red de dispositivos para producir, administrar, adquirir, modificar, desplegar o transmitir información en sus diferentes formas (imágenes, textos, videos, audio, etc.). Se considera un transformador de información

¹ Institute of Electrical and Electronics Engineers, por sus siglas en inglés, es una organización profesional, de orden mundial con el objetivo de beneficiar a la humanidad a través de conocimiento en forma de publicaciones científicas, estándares tecnológicos y actividades profesionales [4].

que tiene como entrada información y como salida información procesada a través de su programación [3].

- Como medio para entregar un software actúa principalmente como la base para tener el control de los dispositivos electrónicos. Recordando que el hardware es inutilizable sin la existencia de un software que permita controlar al sistema. Uno de los productos de software más importantes para esta tarea son los sistemas operativos. Estos permiten la comunicación de información, creación y gestión de otros programas de software [3].

Existe una serie de características esenciales que todo software debe tener para ser considerado un buen software profesional. En la figura 3 se profundizan 5 características esenciales de un producto de software.

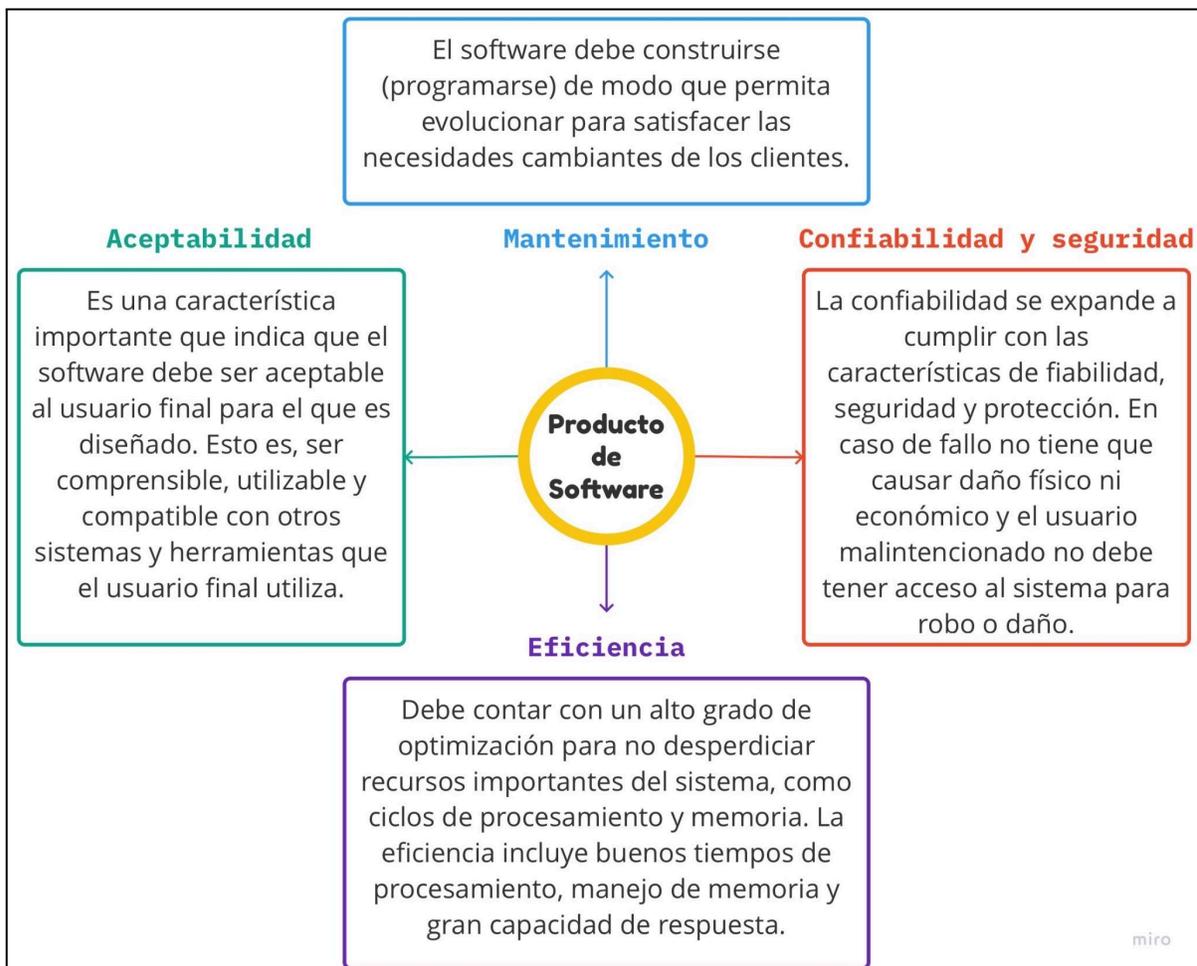


Figura 3. Producto de software: Características [3].

2.4. Ciclo de vida del software

El ciclo de vida del software fue el primer modelo para el proceso de desarrollo de software. Es un proceso en donde cada una de las actividades a realizar deben ser planeadas y programadas [5]. Este modelo se sigue adaptando a la mayoría de los proyectos de desarrollo y se basa en etapas (revisar Tabla 1) que reflejan directamente las actividades fundamentales del desarrollo de software.

Etapa	Descripción
Análisis y definición de requerimientos	Proceso de consulta a los usuarios del sistema para definir los servicios, objetivos y restricciones del producto. Se definen los requerimientos y especificaciones del sistema.
Diseño del sistema y del software	En función de los requerimientos, se realiza el proceso de diseño asignando requerimientos de hardware y software. En esta fase se diseña una arquitectura general del proyecto. El diseño del software consiste en identificar y especificar las funciones fundamentales del sistema.
Implementación y prueba de unidad	En esta etapa, se realiza el diseño de software a nivel programación. Ya sea como un paquete de programas o componentes de programa. El proceso de prueba de unidad consiste en la verificación del cumplimiento de especificaciones de cada unidad de programa.
Integración y prueba de sistema	Los programas individuales o unidades de programa se fusionan y prueban como un producto completo para verificar el cumplimiento de requerimientos de software. Después de ser testeado y verificado, se le proporciona el producto al cliente.
Operación y mantenimiento	Esta etapa puede considerarse la parte más extensa del ciclo de vida del software, donde se despliega el producto de software, se instala en el equipo del usuario final y se pone en funcionamiento. La parte del mantenimiento abarca la corrección de errores no detectados en etapas anteriores, optimizar la implementación de las unidades del sistema y evolucionar el software conforme surgen nuevos requerimientos.

Tabla 1. Desarrollo de software: etapas [5].

2.5. Arquitectura de Software

Dentro del desarrollo de software, la primera fase para dar inicio a la construcción del producto de software es el diseño de la arquitectura de software.

Comenzando por la definición de arquitectura, de acuerdo con Jerrold Grochow [5]: “La arquitectura de un sistema es un marco general que describe su forma y estructura: sus componentes y la manera en la que se ajustan entre sí”. La arquitectura refleja una serie de decisiones que son tomadas para integrar elementos de un sistema con el objetivo de formar un sistema completo y funcional.

La arquitectura de software no es más que la estructura, o conjunto de estructuras, que conforman los componentes del software, abarcando propiedades y relaciones entre ellos.

Es importante mencionar que la arquitectura no es en sí el software operativo. La arquitectura del software tiene como principales objetivos:

1. Verificar si el diseño es efectivo para cumplir con los requerimientos establecidos.
2. Tener en cuenta alternativas arquitectónicas en etapas tempranas del diseño que no dificulten su implementación.
3. Reducir los riesgos que involucran el desarrollo del software.

El diseño arquitectónico se enfoca en definir la organización de un sistema, centrándose en el diseño de la estructura global del mismo. Es la unión fundamental entre el diseño y el levantamiento de requerimientos de un software, debido a que identifica los componentes estructurales fundamentales del sistema y su relación.

Un componente de software se considera desde algo simple como un módulo de programa o una clase, en el caso de programación orientada a objetos, hasta elementos más complejos como las bases de datos.

En cambio, las relaciones entre componentes van desde una simple llamada de procedimiento o función de algún módulo hasta protocolos de acceso a bases de datos.

En la ingeniería de software existe el concepto de patrones arquitectónicos. Este concepto fue introducido en 1990 bajo el nombre de “estilos arquitectónicos” [3].

Un patrón arquitectónico se puede definir como una descripción abstracta que fue puesta a prueba, ensayada y verificada en diferentes sistemas y entornos. Dentro de un patrón arquitectónico se incluye información sobre cuándo es y cuando no es adecuado utilizar el patrón, así como sus ventajas y desventajas

Existen diversas arquitecturas enfocadas a diferentes aplicaciones en la industria del software. A continuación me centraré en la arquitectura de tubería y filtro, que es la arquitectura que se utilizó para el desarrollo del proyecto debido a que todo el proceso depende de los datos de elementos mecánicos.

2.5.1. Arquitectura de tubería y filtro

Es una arquitectura muy utilizada en el tratamiento de datos, ya que las entradas son procesadas en etapas separadas y como resultado genera salidas relacionadas [3].

El principio de funcionamiento consiste en descomponer el problema y dividirlo en componentes, ver figura 4. Estos componentes son llamados filtros, mismos que se encargan de realizar la transformación y el procesamiento de los datos. Los datos son transportados de un componente a otro a través de una tubería.

Cada elemento, o filtro, trabaja de manera independiente de aquellos filtros que se encuentran arriba o abajo del flujo; se diseña para esperar una entrada de datos de cierta forma y produce datos de salida para los filtros siguientes de una forma específica.

Su principal ventaja es que es fácil de comprender, soporta la reutilización de código y los cambios. Además, se adapta adecuadamente a muchos procesos empresariales y puede implementarse para sistemas secuenciales o concurrentes.

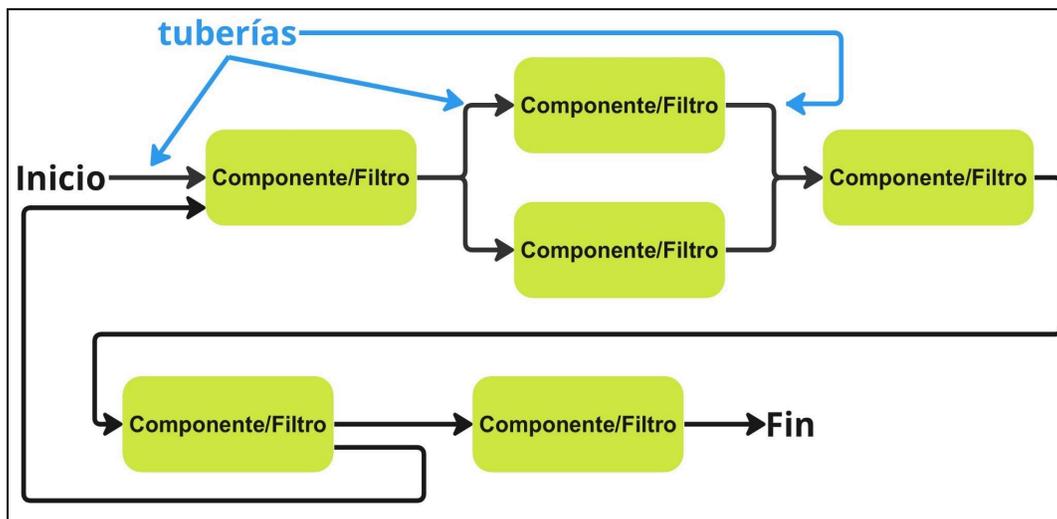


Figura 4. Arquitectura de tubería y filtro.

2.6. Metodologías de desarrollo de software

Actualmente y desde hace un par de años, las metodologías de desarrollo de software están ampliamente agrupadas en 3 categorías principales, estas son: tradicionales, ágiles e híbridas [4].

Metodologías tradicionales: este grupo se enfoca principalmente en los modelos en cascada y espiral, recordando que son pesados en sus procesos debido a que son modelos formales y resisten al cambio. Son etapas definidas que se desarrollan secuencialmente y uno de los principales retos es la implementación de cambios al proyecto.

La industria del desarrollo de software inició con el uso de metodologías tradicionales con el objetivo de garantizar entregas exitosas.

Los elementos importantes de los modelos tradicionales son el liderazgo, políticas estratégicas, indicadores de desempeño, gestión del conocimiento, comunicación efectiva y aprendizaje organizacional, siendo el más importante la gestión de las partes interesadas. Un ejemplo de metodología tradicional es el modelo en cascada (revisar figura 5).

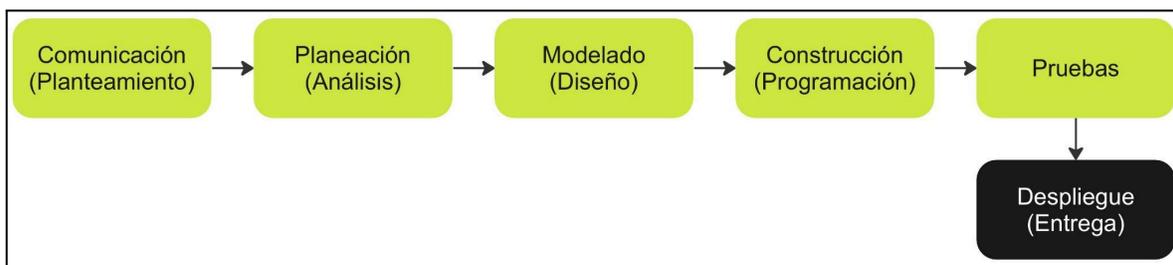


Figura 5. Modelo en cascada [5].

Metodologías ágiles: son un conjunto de procesos adaptativos e iterativos, se consideran ligeros por ser informales y no resistentes al cambio (ver figura 6). Se apoyan en su mayoría en el enfoque incremental para la especificación, desarrollo y entrega del producto de software.

Consideran 4 valores clave para motivar el surgimiento de mejores formas de desarrollar softwares, especialmente para empresas distribuidas mundialmente.

1. Interacción entre individuos, procesos y herramientas.
2. Software de trabajo y documentación.
3. Comunicación con el cliente sobre negociación del proyecto.
4. Respuesta al cambio para el siguiente plan.

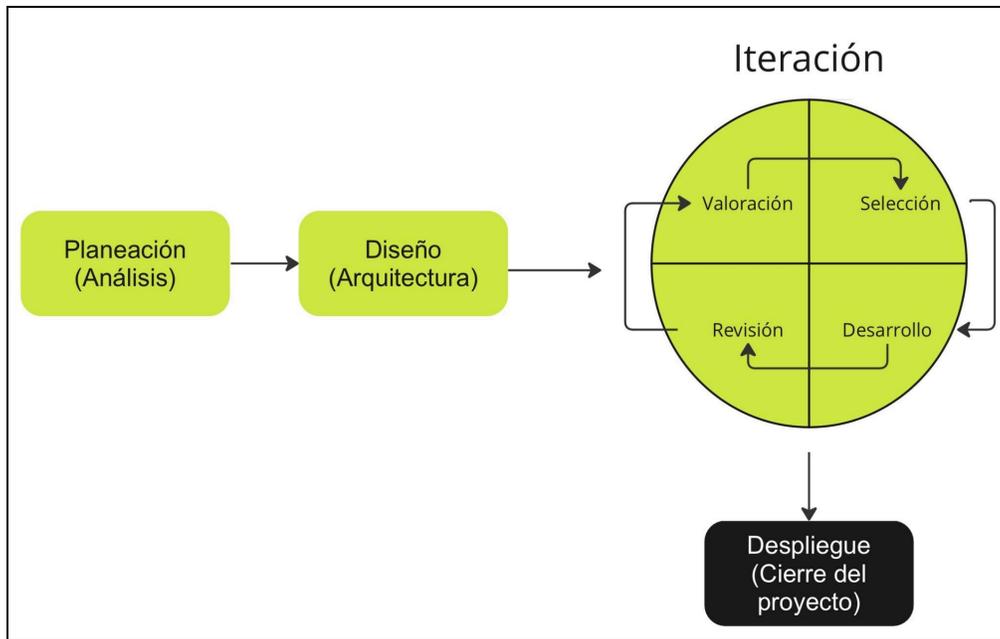


Figura 6.Vista general del modelo ágil [5].

Metodologías híbridas: Combinación de metodologías tradicionales y ágiles que coexisten como una sola metodología. Nuevas áreas de investigación que analizan estas metodologías, confirman que los desarrolladores de software muestran oposición a respaldar las promesas de entrega de metodologías ágiles cuando se trata de proyectos grandes y complejos [4].

Aún cuando se sabe que las metodologías tradicionales funcionan bien, existe la necesidad de abordar proyectos complejos que pueden necesitar estrategias de desarrollo tanto adaptativas como lineales.

Son metodologías relativamente nuevas, altamente personalizables y están cobrando impulso por los buenos resultados que han tenido y se recomiendan como nuevas corrientes de investigación.

2.7. Algoritmos

2.7.1. Definición de algoritmo

La palabra algoritmo, surge en honor al matemático persa del siglo IX Abu Muhammad Ibn Musa Al-Khwarizmi, también conocido como Al-Juarismi². Se

² Estudioso persa responsable de recopilar y traducir grandes obras matemáticas de las civilizaciones griega, india y persa, ampliando este conocimiento y creando el álgebra. Expuso completamente el sistema de numeración Indio, comprobando que nuestro sistema de numeración actual no es de origen Árabe. Este sistema se conoció como Al-Juarismi y el nombre se deformó después de múltiples traducciones en algoritmo, palabra que se asoció a los distintos pasos a seguir de las reglas de cálculo que podían efectuarse con sus sistema de numeración [6].

pueden encontrar una gran variedad de definiciones en distintas fuentes bibliográficas y dependen mucho de la perspectiva de cada autor.

De manera informal, un algoritmo es cualquier proceso computacional bien definido que percibe como entrada un valor o valores, y produce como respuesta de salida un valor, o conjunto de valores. Son una herramienta fundamental para resolver problemas siguiendo una serie de instrucciones lógicas bien definidas, que se pueden trasladar a una computadora [7].

Es relevante mencionar que los algoritmos no dependen de los numerosos lenguajes de programación con los que pueden ser implementados. En la mayoría de las problemáticas es posible construir un algoritmo, mismo que puede implementarse y ejecutarse en diferentes lenguajes de programación.

Para ser considerados como tal, el algoritmo debe ser preciso, definido y finito.

- Preciso: El algoritmo debe ser creado de manera estricta, dejando de lado las ambigüedades en las indicaciones que lo conforman.
- Definido: Si el algoritmo se ejecuta dos o más veces con la misma entrada, omitiendo los algoritmos que involucren la aleatoriedad, se debe obtener el mismo resultado de salida en todas las ejecuciones.
- Finito: Significa que debe tener un inicio y un final, sin importar la cantidad de instrucciones a seguir.

Se pueden clasificar en dos grandes grupos, algoritmos cualitativos y algoritmos cuantitativos.

En el caso de los algoritmos cualitativos, son algoritmos cuyos pasos o instrucciones no involucran cálculos numéricos, pueden ser: instrucciones para realizar actividades físicas, buscar palabras en el diccionario, beber un té, etcétera.

En cambio, los algoritmos cuantitativos si involucran cálculos numéricos en sus pasos o instrucciones, por ejemplo los pasos para solucionar el determinante de una matriz, el cálculo de la velocidad de un cuerpo, calcular la resistencia total de un circuito, etcétera. Estos últimos son los más comunes en la computación.

Diferencia entre algoritmo y programa de cómputo

Es importante conocer la diferencia entre ambos conceptos. De antemano, no son lo mismo. Un programa de cómputo es un conjunto de instrucciones ordenadas, codificadas en algún lenguaje de programación que interpreta uno o varios algoritmos y estas instrucciones son ejecutadas por la computadora.

A diferencia de los algoritmos, que únicamente son los pasos a seguir para dar solución a alguna problemática, siendo independientes del lenguaje de programación. El algoritmo es parte fundamental de un programa de cómputo.

2.7.2. Algoritmo Genético

Son un tipo de algoritmos, considerados técnicas, de optimización metaheurísticas. Fueron propuestos en los años 60's por Jonh Holland y comenzaron a aplicarse a problemas de ciencia e ingeniería reales algunas décadas atrás. Principalmente debido a que son computacionalmente demandantes y por lo tanto requieren de una capacidad de cómputo elevada para llevar a cabo un gran número de operaciones en poco tiempo [7].

Este tipo de algoritmos están basados en la teoría de la evolución de Charles Darwin. Recordando la teoría evolutiva, los individuos que se adapten mejor al medio, tienen mayor probabilidad de dejar descendencia y sus genes pasarán a las siguientes generaciones de individuos, de este modo permiten la evolución de los individuos.

Este tipo de algoritmo está pensado para desempeñar una búsqueda guiada a partir de un conjunto inicial de posibles soluciones, se le conoce como población inicial, la cual va mejorando en cada iteración del algoritmo.

El algoritmo genético se rige por iteraciones que son conocidas como generaciones y, regularmente, la última generación proporciona la mejor o las mejores soluciones al problema de optimización.

A cada una de las posibles soluciones del problema de optimización se les conocen como individuos y cada individuo representa a las variables independientes del problema.

Para los problemas numéricos de optimización, los individuos son potenciales soluciones al problema. La adaptación, en este caso a los problemas de optimización, se mide mediante la función a optimizar, también llamada función objetivo o Fitness function.

Un individuo se adaptará bien a la solución del problema si se produce un Fitness alto en caso de maximización o si se produce un rendimiento bajo en caso de que se trate de un problema de minimización.

En cada generación del algoritmo se producen nuevos individuos, descendientes y offsprings. Son creados mediante operaciones genéticas, produciendo así nuevas poblaciones.

Las operaciones genéticas presentes en los algoritmos genéticos son selección de padres, cruce y mutación.

En la figura 7 se aprecia de manera general el funcionamiento de los algoritmos genéticos.

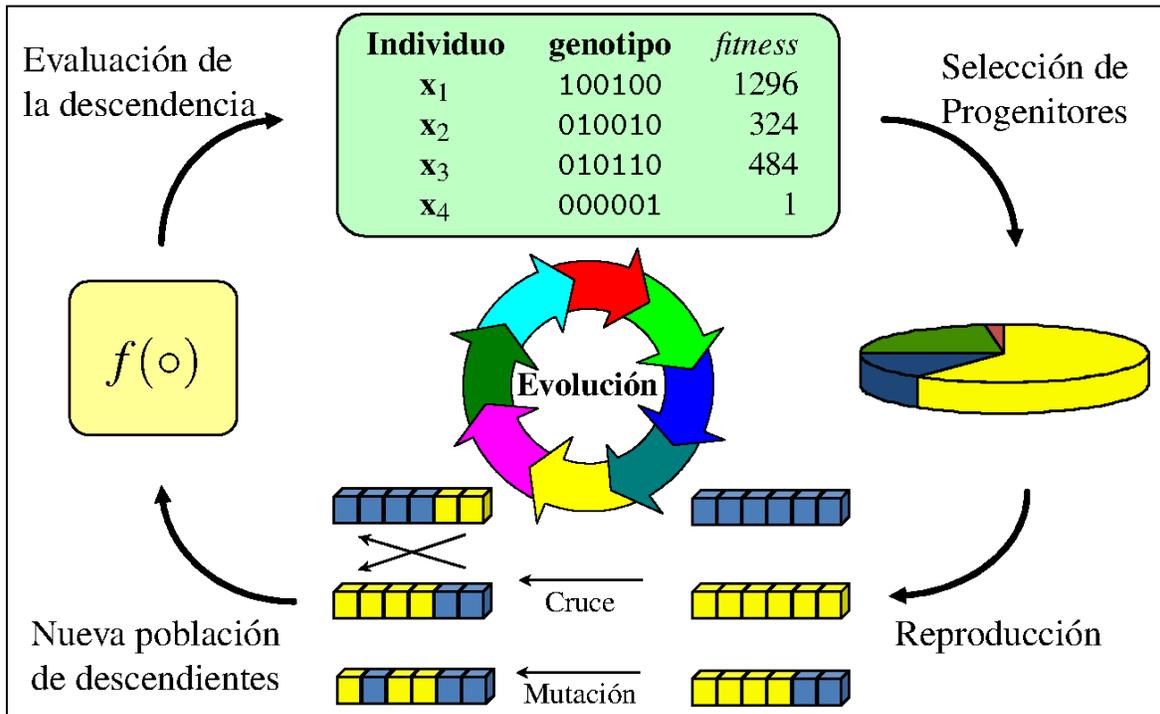


Figura 7. Funcionamiento de los algoritmos genéticos. [8]

La **selección**, como operación genética del algoritmo, es el proceso de elección de progenitores para llevar a cabo el cruce. Se eligen los individuos de la población para crear la nueva población para la siguiente generación.

El objetivo principal de esta operación es elegir de la mejor manera a los progenitores para que la descendencia tenga mejor fitness.

La operación de selección debe ser equilibrada con la mutación y el cruce. Si se realiza una selección muy fuerte o estricta, el algoritmo puede caer en el sobreajuste de soluciones, reduciendo la evolución y la diversidad necesaria para el cambio en siguientes generaciones.

Existen diversas operaciones de selección, a continuación se mencionan las operaciones más representativas y comunes [9].

- Selección por ruleta "roulette wheel selection".
- Selección aleatoria "random selection".

- Selección por rango “rank selection”.
- Selección estocástica universal “stochastic universal selection”.
- Selección de estado estacionario “steady state selection”.

El **cruce** es otra operación genética que tiene lugar después de la selección de progenitores. El propósito del cruce es intercambiar la información genética de los individuos para crear descendencia artificial con características prometedoras para un mejor Fitness.

A continuación, se mencionan los tipos de cruce más utilizados en algoritmos genéticos.

- Cruce de punto simple “single point crossover”.
- Cruce de punto doble “two points crossover”.
- Cruce uniforme.

Siguiendo el flujo del proceso evolutivo, la operación que se ejecuta tras el cruce es la **mutación**. Esta es una operación muy importante debido a que impide que el algoritmo permanezca en una zona de optimización local. Trabaja bajo el fenómeno aleatorio para abarcar la mayoría de las soluciones posibles haciendo que la población de individuos sea diversa.

Los tipos de mutación más relevantes, se enlistan a continuación:

- Mutación aleatoria “Random mutation”.
- Mutación por intercambio “swap mutation”.
- Mutación de revuelta “scramble mutation”.
- Mutación de inversión “Inversion mutation”.

2.7.3. PyGAD

PyGAD es una biblioteca Python de código abierto, útil para la construcción de algoritmos genéticos altamente personalizables y para optimizar algoritmos de aprendizaje automático. Se integra bien con las bibliotecas de uso científico para machine learning. Actualmente la biblioteca sigue en desarrollo activo y recibe actualizaciones de manera recurrente [10].

Recordando una de las principales características de los algoritmos, donde se dice que son independientes del lenguaje de programación y pueden codificarse en diversos lenguajes. Se optó por utilizar la biblioteca PyGAD debido a que el desarrollo de este trabajo se realizó utilizando el lenguaje de programación Python.

La biblioteca tiene implementadas distintas operaciones de cruce, mutación y selección de padres. Tales como:

- **Selección:** steady state selection, rank selection, random selection, roulette wheel selection y stochastic universal selection.
- **Cruce:** single point crossover, two points crossover y uniform.
- **Mutación:** random mutation, scramble mutation, swap mutation, inversion mutation y adaptive mutation.

Esta biblioteca permite una fácil implementación de algoritmos genéticos que requieren poco ajuste para funcionar. La instancia, por sí sola, tiene sus hiper parámetros ajustados por defecto de modo que ofrezca buenos resultados y rendimiento en la mayoría de los casos.

El buen rendimiento y funcionamiento del algoritmo implementado utilizando esta instancia, depende mucho del programador y de lo claro que tenga el proceso a optimizar.

Otra de las ventajas que ofrece esta biblioteca es que permite diseñar e implementar las funciones para cada una de las operaciones genéticas. No se limita a utilizar las que tiene implementadas la biblioteca. Esto implica un alto grado de personalización del algoritmo genético y hace posible adaptarlo a muchos problemas. Esto influyó en la decisión de trabajar con esta biblioteca para implementar el algoritmo.

La documentación completa, así como ejemplos programados, se encuentra en la página oficial de la biblioteca PyGAD [10].

2.8. Software de diseño y dibujo asistido por computadora

2.8.1. Software CAD

El diseño asistido por computadora (CAD) se trata de un software destinado al diseño, modificación y documentación técnica de representaciones gráficas bidimensionales o tridimensionales (2D o 3D) de objetos físicos que pretende sustituir el dibujo manual por un proceso automatizado [11].

El CAD es muy utilizado en procesos de ingeniería, desde el diseño hasta la definición del método de construcción. Es una herramienta que permite a los ingenieros probar de manera interactiva las variantes de diseño sin tener que implementar físicamente su modelo [12].

Los objetivos principales del software CAD son:

- Reducir los tiempos de lanzamiento al mercado (tiempos de diseño, fabricación, implementación, etc.).
- Reducir costos de implementación de los productos.
- Mejorar la productividad.
- Optimizar tiempos.

2.8.2. Rhinoceros 3D

Entre un gran número de herramientas de software CAD existentes en el mercado. Rhinoceros 3D³ es un software que permite crear, editar, analizar, documentar, renderizar y animar superficies, sólidos, nubes de puntos y mallas poligonales por mencionar algunos.

Ofrece un gran número de herramientas de modelado 3D, así como una plataforma de desarrollo que permite crear cualquier forma imaginable de manera gráfica en una computadora.

Una de las principales ventajas que ofrece Rhinoceros 3D, dentro de su plataforma de desarrollo, es la posibilidad de implementar algoritmos para automatizar el dibujo y diseño de elementos a partir de programación visual o programación orientada a objetos, ya que permite la implementación de código de programación Python y C#.

2.8.3. BIM

Es una metodología para la construcción. El modelado de información para la construcción (BIM) es el proceso que abarca la creación y administración de la información de un activo construido. Se apoya principalmente en modelos inteligentes impulsados por plataformas en la nube e interactúa con datos estructurados y multidisciplinarios para generar una representación digital (modelos 3D) de un activo durante su ciclo de vida, esto es desde el origen y planeación, hasta su implementación en la construcción y puesta en marcha [14].

El modelado de información para la construcción (BIM) es la base de la transformación digital en la industria de la arquitectura, la ingeniería y la construcción (AEC). (Autodesk, 2023).

Este proceso es útil para proyectos de edificaciones, infraestructura y construcción, ya que a diferencia del modelado CAD, BIM va más allá de las representaciones geométricas para contener las relaciones, los metadatos y

³ Rhinoceros 3D software asistido por computadora creado por Robert McNeel & Associates. Para más información, consultar la referencia [13].

comportamientos propios de los componentes de construcción reales. Este tipo de modelado se complementa con CAD. Al combinar ambas tecnologías se genera como resultado la optimización de procesos de diseño y construcción, obteniendo beneficios como el control y administración total del proyecto y cada una de sus fases.

2.8.4. Tekla Structures

Es un software BIM, perteneciente a Trimble⁴, para gestionar el diseño, detallado, despiece, fabricación, planeación de la producción y el montaje de todas las estructuras de acero de una construcción.

Entre las características a destacar de este software es que permite modelar la estructura a construir de una manera rápida y eficiente, teniendo la posibilidad de modelar automáticamente la estructura a partir de la interconexión con herramientas de software CAD como Rhinoceros 3D, permitiendo la automatización de generación del diseño.

Tekla proporciona plug-ins desarrollados para Rhinoceros que hace posible la comunicación entre ambas plataformas con el objetivo de automatizar el proceso de diseño y modelado.

Una vez creado el modelo de la estructura a construir, Tekla tiene la posibilidad de generar todo tipo de planos generales, de despiece y fabricación, así como listado de materiales y de piezas.

2.9. Lenguajes de programación

2.9.1. Lenguaje de Programación Visual (VPL)

El lenguaje de programación visual es un tipo de programación fácil de entender y realizar, permite a los usuarios construir programas de software utilizando expresiones visuales. Hace uso de componentes prefabricados no editables, conectados entre sí para realizar tareas específicas [16].

Permite implementar algoritmos sin la necesidad de tener conocimientos de código de programación.

Entre sus ventajas más destacables está la facilidad de uso, son fáciles de aprender, las funciones de los componentes ya están predefinidas y permite una agilidad de diseño. El usuario solo debe preocuparse por utilizar los componentes

⁴ Trimble es una empresa tecnológica que ofrece soluciones de modelado 3D, análisis de datos y conectividad. Para más información sobre Trimble y Tekla Structures, consultar referencia [15].

correctos para la acción que desea realizar, sin importar la cantidad de elementos necesarios para implementar su algoritmo.

La desventaja más importante es que no es posible editar de manera interna el funcionamiento de los componentes, esto limita a la hora de diseñar algoritmos complejos, lo que obliga a incrementar el uso de componentes para realizar tareas, en ocasiones sencillas. No obstante, algunos lenguajes de programación visual permiten crear elementos personalizados, mismos que pueden ser reutilizados las veces que sean necesarias.

2.9.1.1. Grasshopper de Rhinoceros

Es un editor gráfico que permite realizar programación visual. Se encuentra incluido en las herramientas de modelado 3D de Rhinoceros 3D [17] y [18].

A diferencia de otros lenguajes de programación con texto, Grasshopper no requiere de conocimientos de programación o secuencias de comandos, pero esto no impide que los diseñadores puedan construir algoritmos complejos.

A continuación, en la figura 8, se muestra un pequeño programa para sumar dos números, realizado con programación visual en Grasshopper.



Figura 8. Ejemplo de programación visual: Suma de dos números.

Por otra parte, Grasshopper permite diseñar componentes personalizados a partir de programación con lenguajes C#, Visual Basic y Python. Esto permite al usuario implementar cualquier algoritmo y proceso, dentro del entorno de Grasshopper.

2.9.2. Python

Es un lenguaje de programación que fue creado a principios de la década de 1990 por Guido Van Rosum, de Países Bajos. Este lenguaje pretendía reemplazar al lenguaje ABC [19].

Se trata de un lenguaje de programación interpretado y multiparadigma que entre sus ventajas permite el paradigma orientado a objetos. Hoy en día, Python es ampliamente utilizado para desarrollo de software, aplicaciones web, ciencia de datos y machine learning.

En el paradigma orientado a objetos, los objetos definidos en Python tienen las siguientes características [20]:

- Identidad: Cada objeto se distingue uno del otro.
- Estado: Cada objeto es capaz de guardar su estado a través de sus atributos (variables de instancias y campos.)
- Comportamiento: Cada objeto tiene la capacidad de modificar su estado a través de métodos.

Que Python sea un lenguaje interpretado significa que el código se traduce a código máquina a medida que se ejecuta cada instrucción. No pasa por un proceso de compilación a diferencia de lenguajes como C o C++. Al no requerir de un compilador, se dice que es un lenguaje multiplataforma lo que significa que permite ser ejecutado en diferentes sistemas y dispositivos, requiriendo únicamente el intérprete del lenguaje.

En la figura 9, se muestra un ejemplo de un programa en lenguaje Python, creando un objeto y llamando a sus métodos.

```
1 class Carro(): #Objeto carro
2     def __init__(self,marca,modelo,year,transmision):
3         #Campos o atributos
4         self.marca=marca
5         self.modelo=modelo
6         self.year=year
7         self.transmision=transmision
8
9     #Métodos/funciones
10    def Encender(self):
11        print("Carro encendido")
12
13    def Apagar(self):
14        print("Carro apagado")
15
16    def obtener_marca(self):
17        print("Marca del carro: {}".format(self.marca))
18
19    def obtener_modelo(self):
20        print("Modelo del carro: {}".format(self.modelo))
21
22    def obtener_year(self):
23        print("Año del carro: {}".format(self.year))
24
25 if __name__ == "__main__": #Programa principal
26
27     Camioneta=Carro("Ford","Ecosport",2018,"Automática")#Creación de instancia del objeto carro
28     #Llamada a los métodos del objeto carro.
29     Camioneta.obtener_marca()
30     Camioneta.obtener_modelo()
31     Camioneta.obtener_year()
32     Camioneta.Encender()

```

Marca del carro: Ford
Modelo del carro: Ecosport
Año del carro: 2018
Carro encendido

Figura 9. Programa Python: Creación de objeto.

3. Definición del problema

El diseño de un muro pantalla requiere numerosas verificaciones apegadas a las normas técnicas complementarias para diseño y construcción de cimentaciones y normas técnicas complementarias para estructuras de concreto (Referencias [11] y [12]), normativas aplicables para el diseño y construcción para la ciudad de México.

Una de las prioridades a tomar en consideración en el diseño, es la definición del área de acero mínima, en función de las características del muro; el área de acero mínima es la cantidad de acero que permite garantizar un integridad estructural del muro incluso por cambios volumétricos. Es necesario que el ingeniero se encargue de seleccionar el diámetro de varilla y la separación de refuerzo principal para verificar el cumplimiento del área mínima de acero. También debe analizar los datos de elementos mecánicos, para determinar si se vuelve necesario colocar refuerzos adicionales donde los momentos flexionantes rebasan la capacidad del momento resistente correspondiente al área de acero propuesta y si las fuerzas cortantes rebasan las resistencias, en conjunto, del concreto y acero.

En caso de requerir modificar los diámetros de varilla o separación entre varillas del refuerzo principal, el ingeniero de proyecto debe realizar nuevamente todo el proceso de cálculo y diseño. Este tipo de cambios pueden deberse a problemas constructivos en la obra, problemas de inventario e incluso a temas de presupuesto. Los tiempos de cálculo y diseño aumentan cuando se presentan estas situaciones.

De lo anterior, “la constructora” tuvo la iniciativa de automatizar los procesos de diseño para el muro pantalla, con el objetivo de reducir el tiempo que le toma al ingeniero de proyecto realizar el diseño de este sistema de contención y al área de dibujo el generar planos de taller. Además, lograr la optimización en el uso del acero, con el apoyo de una herramienta de software que permita agilizar estas actividades proponiendo soluciones óptimas en una menor cantidad de tiempo.

El proyecto general, que consistió en la automatización del diseño de muro pantalla, se dividió en 3 partes principales:

1. Creación de un programa en Grasshopper, al cual nos referiremos como “Programa 1”, capaz de proponer de manera automática la mejor distribución de paneles de muro en el perímetro del terreno, utilizando la menor cantidad de tipos de paneles (tipos de panel se refiere a segmentos de muro pantalla de diferentes anchos) tomando en cuenta las restricciones constructivas que existen para este proceso.

2. La creación de un programa, al que nos referiremos como “Programa 2”, que tomando los datos de elementos mecánicos y características del muro pantalla, se encargue de calcular de manera óptima el acero de refuerzo para cada tipo de muro pantalla indicado por el “Programa 1”. Esto incluye el refuerzo principal, los bastones para el caso de momentos flexionantes y los estribos para el caso de fuerzas cortantes.
3. Desarrollo de un programa en Grasshopper, “Programa 3”, que utiliza los datos de cálculo y diseño del muro pantalla “Programa 2”, para generar el modelo en 3D en un software especializado que permite obtener los planos de taller de manera simplificada y en menor tiempo.

El presente trabajo se enfoca en detallar la solución que implementé en su totalidad del “Programa 2”, encargado de calcular de manera automatizada el acero de refuerzo de muro pantalla. Esta herramienta funciona en conjunto con el “Programa 1” y “Programa 3” que fueron desarrollados por los otros dos integrantes del “equipo BIM”, ofreciendo una solución eficiente y sólida.

4. Metodología de desarrollo de la herramienta

En este capítulo se describe todo el proceso de desarrollo de la herramienta. Con el objetivo de respetar la privacidad y confidencialidad de la empresa, este informe está limitado a mostrar de manera global el desarrollo de la herramienta de software, cuidando la información de la empresa contenida en los algoritmos y códigos de programación.

En la tabla 2, se muestra una descripción del desarrollo del proyecto que realicé en el periodo de agosto de 2021 a abril de 2022. Durante este periodo de tiempo, mi reto principal fue abstraer el conocimiento existente en la documentación de la empresa para generar los algoritmos y desarrollar el proyecto en cada una de las etapas, con el objetivo de cumplir con la funcionalidad y reglas de negocio del proyecto.

Etapa	Fecha	Descripción de actividades
Planeación y análisis de requerimientos	30 Ago 2021 al 17 de Sep 2021.	<ul style="list-style-type: none"> ● Del 30 de Agosto al 3 de Septiembre de 2021. Primer acercamiento entre el equipo de ingeniería y yo como desarrollador, a través de reuniones virtuales. El objetivo de dichas reuniones fue conocer los requerimientos, expectativas y necesidades del área de ingeniería y diseñar el diagrama de flujo del programa. ● Del 6 al 10 de Septiembre de 2021 Se designó un responsable del área de ingeniería encargado de recopilar los requerimientos del proyecto para generar una estructura, basada en las diferentes reuniones realizadas. ● Del 13 al 17 de Septiembre de 2021 El encargado del proyecto analizó y verificó la información de requerimientos, definiendo el alcance y estructura en un documento de especificación de requerimientos de software (SRS).
Definición de los requerimientos	20 Sep 2021 al 30 Sep 2021.	Una vez completado el levantamiento y análisis de requerimientos, en este periodo el director de ingeniería aprobó la información de especificación de requisitos.
Diseño arquitectónico del producto.	4 Oct 2021 al 15 Oct 2021	<ul style="list-style-type: none"> ● Tomando en cuenta las especificaciones del proyecto por parte del área de ingeniería, diseñé la arquitectura de la herramienta definiendo la cantidad de componentes, la forma en la que estarían conectados y las tareas delegadas a cada uno. ● En cuanto al apartado gráfico, debido al requerimiento de utilizar el entorno Grasshopper, se simplificó la parte

		<p>gráfica de la herramienta, pues el área de ingeniería ya se encontraba familiarizada con la operatividad de Grasshopper y de los programas implementados en dicho entorno por lo que se optó por continuar trabajando bajo el entorno de Grasshopper.</p>
Desarrollo de la herramienta de software.	18 Oct 2021 al 18 Mar de 2022.	<p>En este punto del ciclo de vida del desarrollo de software, comenzó la construcción real y se desarrolló el producto de software, en este caso la herramienta de automatización de diseño de muros pantalla.</p>
Etapa de Pruebas de la herramienta.	21 Mar 2022 al 22 Abr 2022.	<ul style="list-style-type: none"> • Del 21 al 30 de Marzo de 2022 El programa completo en su versión beta se sometió a pruebas con la persona que tuvo el rol de “tester” con el objetivo de identificar defectos, problemas e inconsistencias para reportarlas y corregirlas y así cumplir con los estándares de calidad definidos en el documento de especificación de requerimientos. • Del 31 de Marzo al 15 de Abril de 2022 La herramienta de software en su versión beta se liberó al área de ingeniería en concepto de prueba para verificar su funcionamiento y validación de cálculos y soluciones. • Del 18 al 22 de Abril de 2022 Periodo destinado a corrección de errores y ajustes al producto de software. En esta etapa se corrigieron errores importantes, detalles de funcionamiento y experiencia de usuario.
		<p>Finalmente, una vez que la herramienta de software se probó y</p>

Despliegue y mantenimiento.	28 Abr 2022	verificó, se realizó el despliegue del producto de software. El mantenimiento del producto se realizó de manera mensual, durante 6 meses, verificando el correcto funcionamiento de la herramienta de software.
-----------------------------	-------------	--

Tabla 2. Desarrollo del proyecto: Etapas.

4.1. Análisis

Con el objetivo de realizar la implementación de la automatización del cálculo y diseño de muros pantalla, se realizó el análisis sobre el procedimiento de cálculo y diseño de muros. Uno de los primeros pasos de este análisis fue entender el proceso e interactuar con las personas involucradas, a fin de comprender sus actividades e identificar las áreas de oportunidad para la propuesta de automatización.

Mediante reuniones virtuales con el equipo de ingeniería y el resto del “equipo BIM” se recopilaron y detallaron los requisitos de la herramienta de automatización, identificamos las mejoras necesarias con respecto al proceso manual y se definió el alcance del proyecto.

Las reuniones se realizaron principalmente en conjunto con el director de ingeniería y la ingeniera representante del “equipo BIM”, ya que son los ingenieros con mayor experiencia en el proceso de cálculo y diseño de muros pantalla. Posteriormente, otros miembros del área de ingeniería y del “equipo BIM” se integraron a las reuniones con el propósito de comprender el proyecto y aportar ideas.

Como resultado, en la tabla 3, se presenta la comparación del proceso tradicional de cálculo y diseño con la herramienta de automatización acotando el alcance del proyecto, definiendo los requerimientos funcionales principales de la automatización del proceso de cálculo y diseño de muros pantalla.

Proceso manual	Proceso con propuesta de mejora
<ul style="list-style-type: none"> Se utiliza como herramienta a Microsoft Excel. El ingeniero de proyecto toma los datos resultado de análisis 	<ul style="list-style-type: none"> Utilizar el editor de programas visuales Grasshopper del software CAD Rhinoceros y programación orientada a objetos con Python.

<p>geotécnico del muro y los ingresa a una hoja de cálculo de Excel.</p> <ul style="list-style-type: none"> ● Se elige, de manera manual e iterativa, un diámetro de varilla y separación entre varillas para verificar que el área de acero propuesta, cumpla con lo establecido en el reglamento de construcciones [21]. ● El ingeniero, de manera manual e iterativa, define la colocación de acero adicional para momentos y cortantes en las zonas de mayor demanda, proponiendo nuevamente diámetro y separación de varillas hasta cubrir los esfuerzos máximos a los que será sometido el muro. ● El uso de hojas de cálculo no estandarizadas son susceptibles a tener errores. ● El proceso de cálculo y diseño toma alrededor de 2 días hasta que el ingeniero de proyecto genera la información necesaria para que el área de dibujo realice los planos de taller. 	<ul style="list-style-type: none"> ● Automatizar la búsqueda de acero principal del muro, seleccionando diámetro de varilla y separación entre varillas, analizando los datos resultado del análisis geotécnico del muro y garantizando el cumplimiento de las Normas técnicas complementarias para diseño y construcción de estructuras de concreto [22]. ● Optimización de la cantidad de acero utilizando algoritmos heurísticos para garantizar el menor uso de acero que cumpla con las Normas técnicas complementarias para diseño y construcción de estructuras de concreto [22]. ● Revisión y solución automática de refuerzo de acero adicional del muro para momentos flexionantes y fuerzas cortantes propuestas por el programa. ● Optimización del tiempo empleado para calcular y diseñar el muro pantalla, reduciendo aproximadamente un 82 % el tiempo necesario para este proceso.
---	---

Tabla 3. Proceso manual y propuesta de mejora.

4.2. Planeación y definición

Después de haber identificado las necesidades del área de ingeniería y crear la lista de requerimientos mínimos de la herramienta de cálculo y diseño de muros pantalla (ver tabla 4), se construyó un plan de desarrollo con el “equipo BIM” donde se definieron fechas y periodos de revisión de los avances del proyecto.

Nº	Requerimientos mínimos
1	Uso del software Rhinoceros 3D y Grasshopper.
2	Solución automática para acero principal de muros pantalla.
3	Optimización del uso de acero para muros pantalla.
4	Análisis y solución automática de acero de refuerzo adicional.
5	Formato estándar de datos de salida para comunicarlo con el software de dibujo en Tekla.

Tabla 4. Requerimientos mínimos para herramienta de cálculo y diseño de muros pantalla.

La metodología de desarrollo de software empleada para la creación de la herramienta fue un sistema tradicional. Se utilizó el ciclo de vida clásico o modelo en cascada, con una peculiaridad, durante el desarrollo se realizaron tareas iterativas en diferentes puntos, esto debido a cambios y modificaciones en el proceso. Ocasionalmente requerían resolverse de manera rápida puesto que la gerencia de “la constructora” solicitaba avances de los desarrollos del “equipo BIM”.

Esta metodología de software tiene cierta flexibilidad con los cambios en la etapa de construcción y garantiza la obtención de un producto de software de calidad, construido bajo buenas prácticas de desarrollo de software.

Dado que en su momento me encontraba cursando mis últimos semestres en la facultad de ingeniería, se estableció un itinerario de trabajo para dar seguimiento al avance del proyecto. Realizamos reuniones virtuales cada semana para revisar los avances, el funcionamiento de la herramienta, y sobre todo, para verificar que el desarrollo avanzaba correctamente.

Adicionalmente, tenía reuniones con la encargada del proyecto para aclarar dudas y profundizar en el tema del proceso de cálculo y diseño de muros pantalla. En la figura 10 se observa el cronograma de actividades del desarrollo de la herramienta de software bajo esta metodología de desarrollo de software.

Periodo	Fecha inicio de periodo semanal 2021	Periodo	Fecha de inicio de periodo semanal 2022
1	30 de agosto	19	3 de enero
2	6 de septiembre	20	10 de enero
3	13 de septiembre	21	17 de enero
4	20 de septiembre	22	24 de enero
5	27 de septiembre	23	31 de enero
6	4 de octubre	24	7 de febrero
7	11 de octubre	25	14 de febrero
8	18 de octubre	26	21 de febrero
9	25 de octubre	27	28 de febrero
10	1 de noviembre	28	7 de marzo
11	8 de noviembre	29	14 de marzo
12	15 de noviembre	30	21 de marzo
13	22 de noviembre	31	28 de marzo
14	29 de noviembre	32	4 de abril
15	6 de diciembre	33	11 de abril
16	13 de diciembre	34	18 de abril
17	20 de diciembre	35	25 de abril
18	27 de diciembre	Despliegue del software	29 de abril

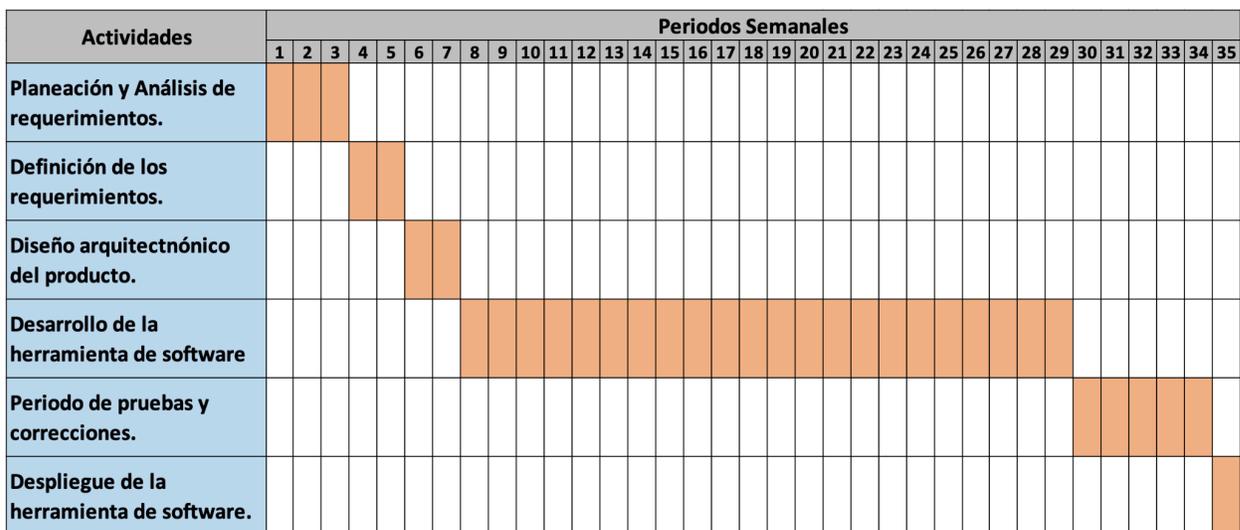


Figura 10. Cronograma de actividades del proyecto.

Durante la etapa de planeación y definición, en las reuniones con el área de ingeniería, se construyó un diagrama de flujo donde se define el funcionamiento de la herramienta de software.

Cuando me integré al proyecto, el área de ingeniería ya tenía un gran avance en el diagrama de flujo, mi aporte fue revisar y concluir el diagrama de flujo que ya se había trabajado. En la figura 11, se muestra una captura de pantalla del diagrama de flujo generado en su momento.

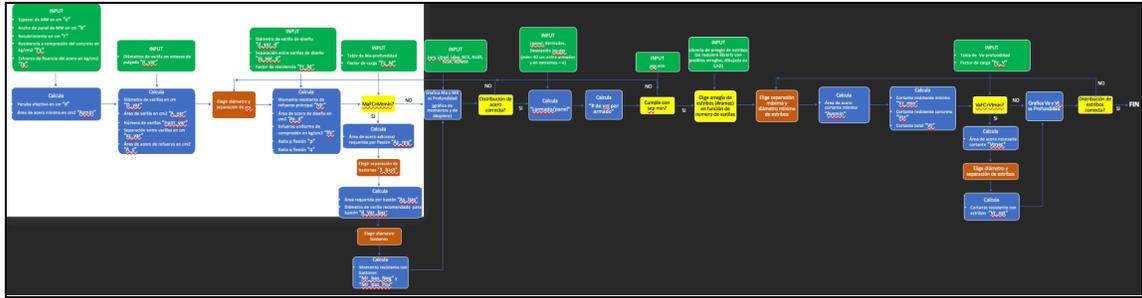


Figura 11 .Planeación y Definición: Diagrama de flujo del programa.

4.3. Diseño de arquitectura de programa

El proceso de diseño y cálculo de muro pantalla surge del análisis de datos de elementos mecánicos. Dichos datos son resultado de análisis geotécnicos previos realizados con software especializado. Sabiendo lo anterior y teniendo como requerimiento implementar la herramienta de software en el editor de programas visuales Grasshopper de Rhinoceros, propuse implementar la arquitectura de tubería y filtro.

Una ventaja notable que proporciona Grasshopper es que no debemos preocuparnos por el diseño de una interfaz gráfica. Por sí solo, el editor permite la implementación de software a través de programación visual. Esto se traduce en la utilización de elementos visuales fáciles de manipular y entender para el usuario.

La manera en la que el usuario interactúa con la herramienta, es a través de elementos visuales proporcionados por Grasshopper. Entre ellos se destacan por funcionar como inputs: Panel, Number slider, Value list y file path que permite seleccionar archivos mediante el explorador de archivos de windows, esto por mencionar algunos. En la figura 12, se pueden apreciar ejemplos de estos elementos que le permiten al usuario ingresar valores al programa.

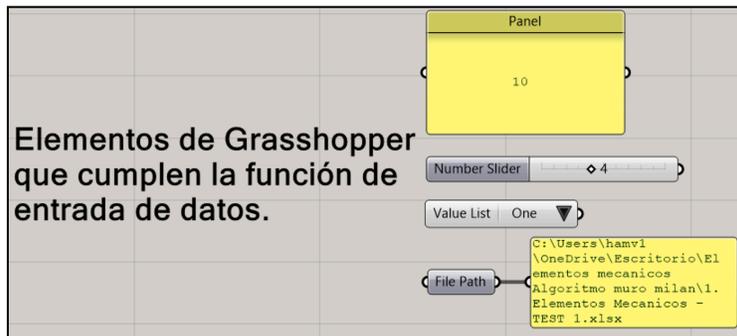


Figura 12. Elementos de entrada de datos (Grasshopper).

El diseño de la arquitectura consistió en realizar un análisis del proceso completo de cálculo y diseño de muros pantalla. Identifiqué todos los filtros de la arquitectura, definiendo la tarea que le corresponde a cada uno.

Como resultado, generé el diagrama de la figura 13, que corresponde a una vista general de la arquitectura del programa. Entiéndase “AG” como Algoritmo Genético.

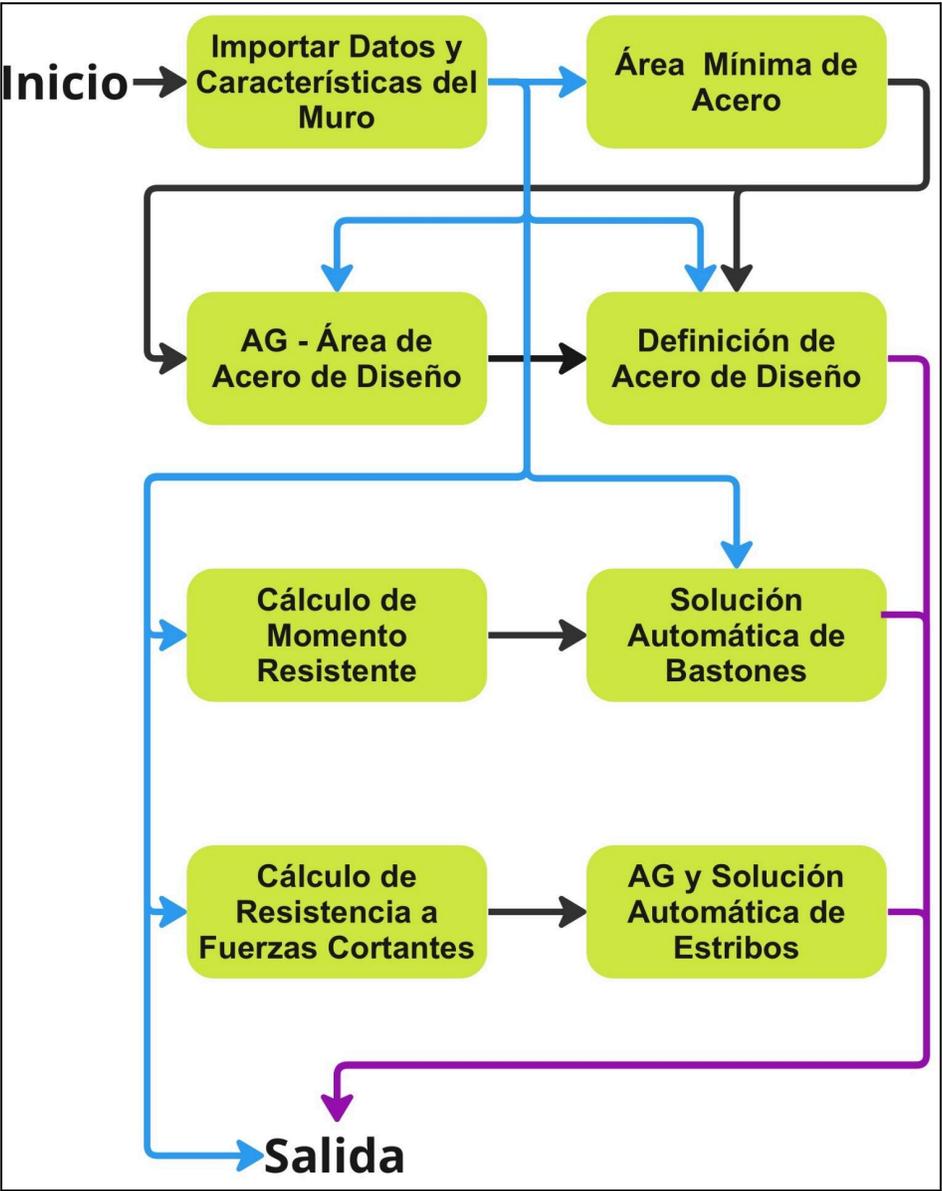


Figura 13. Diagrama de arquitectura del programa (tubería y filtro). Las flechas de cada color corresponden a los datos procesados en el filtro del que provienen, mismos que sirven como datos de entrada para otros filtros o en todo caso para construir la solución final.mi

4.4. Desarrollo

Este capítulo habla sobre el desarrollo de la herramienta de automatización. Cuidando la privacidad y propiedad intelectual de “la constructora” presentaré los aspectos y características generales del desarrollo del programa evitando compartir el código interno de los componentes y algoritmos, de modo que no se comprometa el secreto industrial de la empresa.

Se explica el ambiente de desarrollo y el flujo de trabajo que seguí para la creación de la herramienta de software, como lo es:

- El diseño de los filtros de la arquitectura.
- Creación y configuración de los algoritmos genéticos para el tema de optimización.
- El desarrollo de los algoritmos encargados de la búsqueda y colocación de bastones para momentos flexionantes y estribos para fuerzas cortantes.

Posterior al diseño de la arquitectura, se definió la forma en la que se alimentará la herramienta. Esto es a través del primer filtro, con ayuda de un archivo excel que contendrá los datos de elementos mecánicos de momentos flexionantes y fuerzas cortantes del muro, mismos que serán importados por el componente. El primer elemento de la arquitectura, también recibirá las características del muro por parte del usuario.

4.4.1. Ambiente de desarrollo

Como ya se ha mencionado previamente, el programa se implementó en el entorno de desarrollo de Grasshopper. Grasshopper tiene, de manera nativa, elementos genéricos que se pueden utilizar para la creación de programas de forma visual. Además, proporciona la posibilidad de manera nativa el crear elementos personalizados utilizando programación con lenguajes como Visual Basic, C# y Python. En la figura 14 se muestran algunos de los componentes y plugins disponibles en Grasshopper.

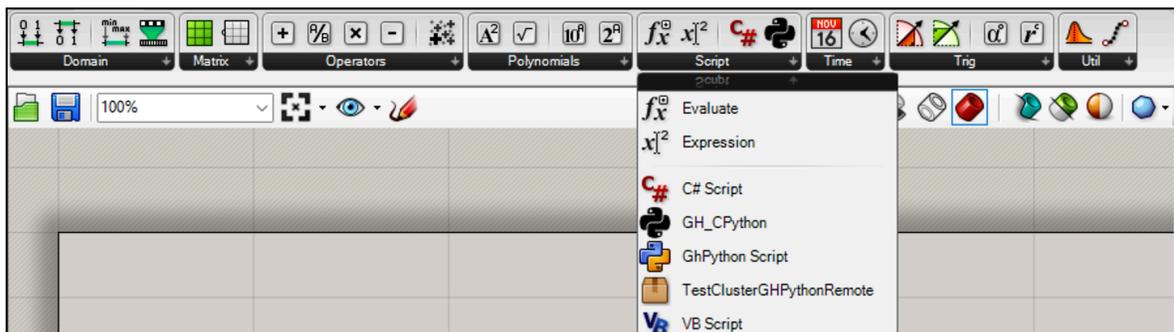


Figura 14. Plugins de Rhinoceros Grasshopper.

Es importante mencionar que para el lenguaje de programación Python, Grasshopper únicamente proporciona de manera nativa el plugin “GhPython Script” el cual permite crear scripts de Python en su versión 2.7. Esto por sí solo representa una limitante significativa a la hora de querer desarrollar programas complejos con Python dado que la mayoría de bibliotecas científicas actuales no son compatibles con la versión 2.7 del lenguaje.

Para resolver el obstáculo que representaba Python 2.7, se utilizó el trabajo de Mahmoud Abdel Rahman, “GH_CPython” [23], el cual consiste en un plugin para Grasshopper que permite utilizar un intérprete externo de Python, pudiendo utilizar versiones actuales del lenguaje y así, construir scripts Python 3 para elementos personalizados dentro de Grasshopper.

Gracias al plug-in de Gh_CPython, se pudo emplear Python en su versión 3.8, lo que permitió utilizar bibliotecas científicas como Pandas, NumPy, PyGAD, Matplotlib y Scikit-learn.

El funcionamiento del plugin Gh_CPython es muy similar al que incluye Grasshopper para crear scripts de Python. Como se muestra en la figura 15, el componente contiene un editor de código, mismo que recibe y entrega variables, que puede comunicarse con otros componentes, ya sean nativos o creados a través de Gh_CPython.

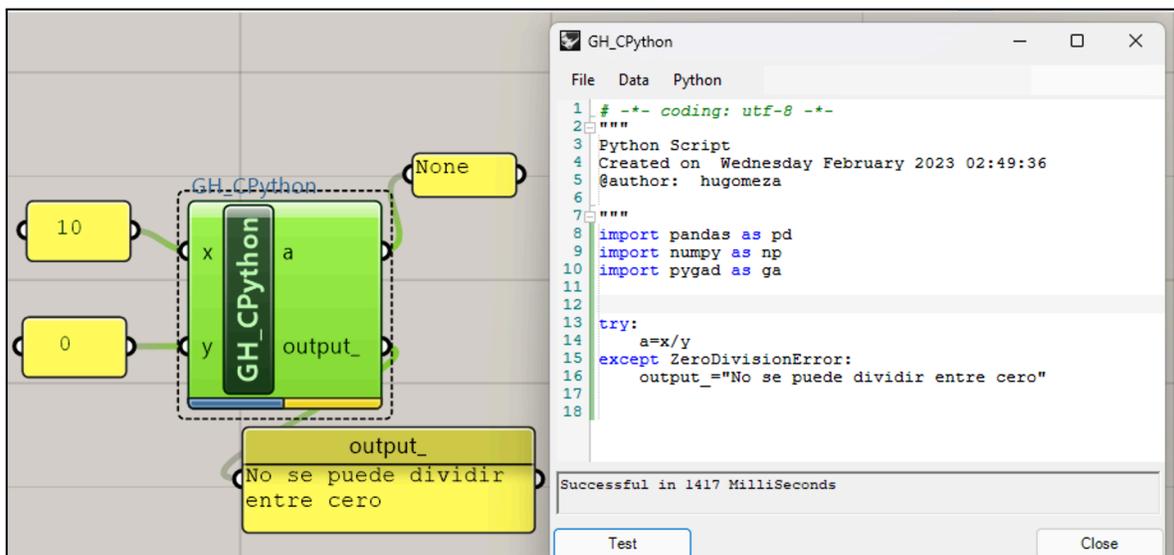


Figura 15. Código Python: utilizando Gh_CPython Plugin for Grasshopper.

4.4.2. Creación de filtros de la arquitectura

Tomando como base la arquitectura del programa, las hojas de cálculo de la empresa y las normas técnicas complementarias [21], diseñé los componentes de

manera individual, utilizando el plugin Gh_CPython [23] y programación orientada a objetos con lenguaje Python. Se replicó cada tarea en el proceso de diseño y cálculo de acero de refuerzo de muros pantalla.

Al salvaguardar la privacidad y propiedad intelectual de la empresa, no me es posible mostrar el código interno de todos los elementos de la arquitectura salvo algunas excepciones. Sin embargo, puedo mostrar captura de pantalla de todos los elementos que conforman la arquitectura del programa de manera general.

En la figura 16, se aprecia el primer elemento de la arquitectura. Este elemento se encarga de obtener la dirección del archivo de datos de elementos mecánicos (archivo de datos de entrada) y la dirección de la carpeta de salida donde el programa guardará la solución del muro pantalla. Esto con el objetivo de crear y organizar las carpetas necesarias en el directorio de salida para que se guarde la información de manera organizada.

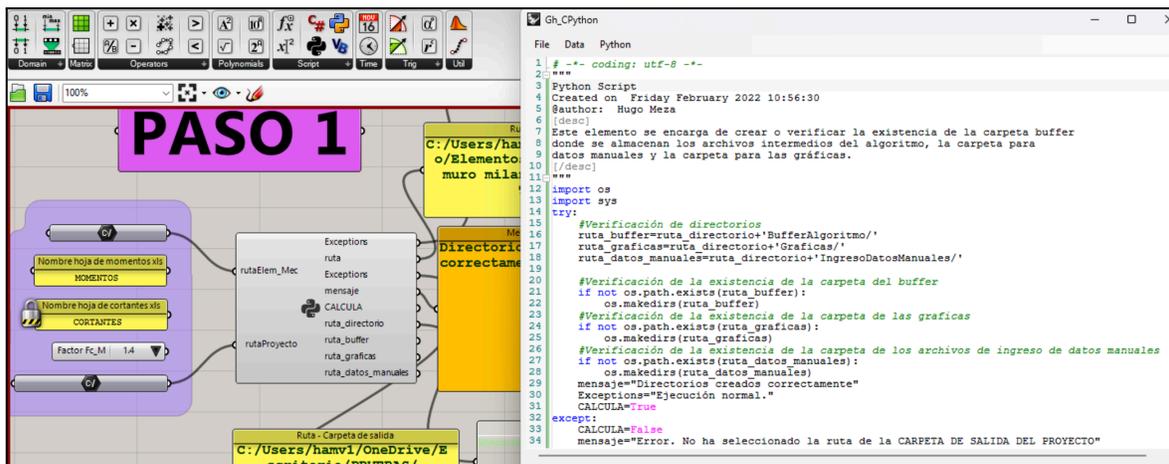


Figura 16. Construcción del primer filtro de la arquitectura de programa.

El resto de elementos de la arquitectura los construí de manera similar a lo mostrado en la figura 16, destacando el desarrollo del algoritmo genético encargado de optimizar el uso de acero y los algoritmos solucionadores de refuerzo adicional para muro pantalla, mismos que serán mencionados en el siguiente subcapítulo.

De manera general, en la figura 17 se muestra el programa terminado con todos los elementos de su arquitectura. En la figura puede observarse los filtros de la arquitectura que se encuentran destacados por recuadros de colores rojo, azul y morado incluyendo una breve descripción del elemento.

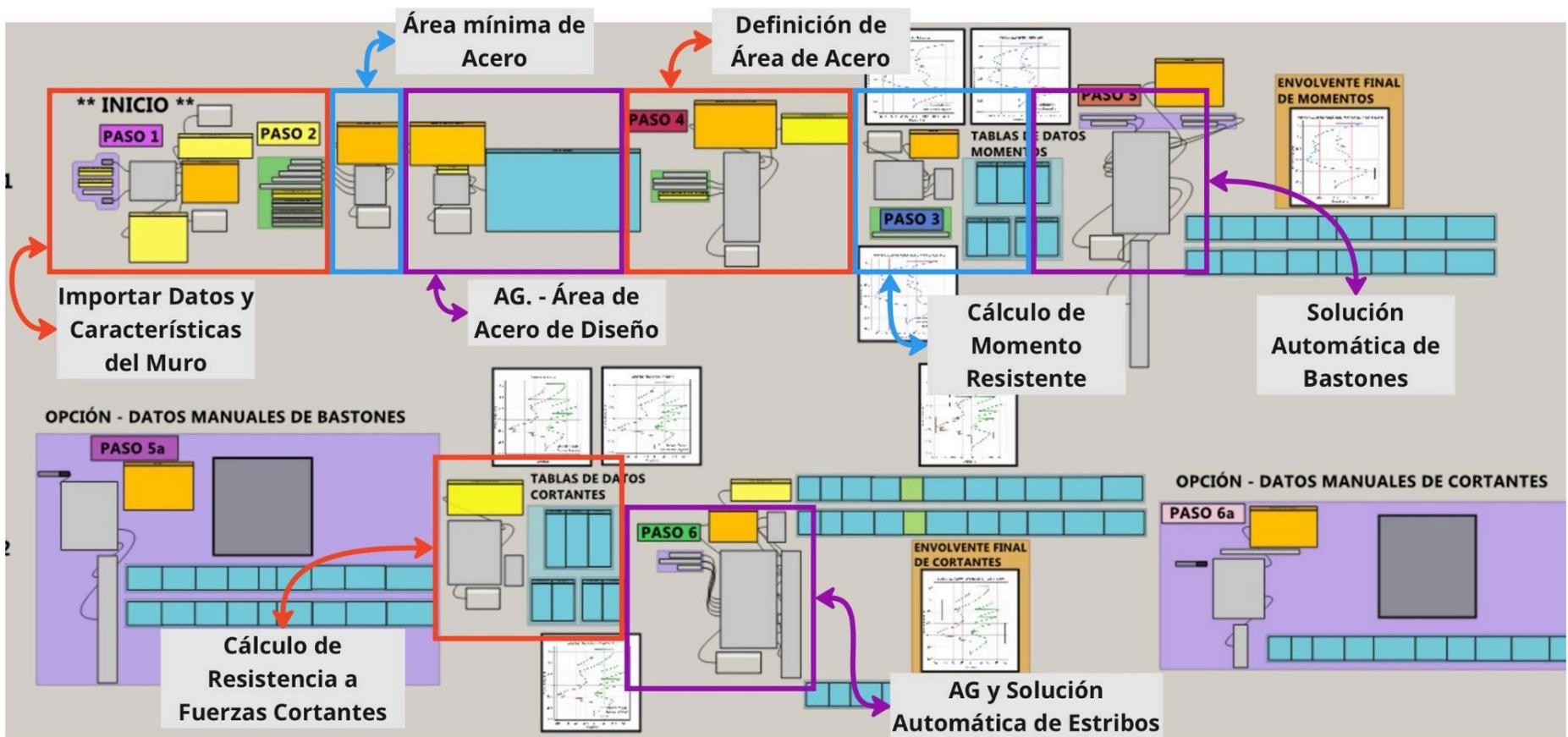


Figura 17. Programa de automatización - Componentes de la arquitectura.

En la figura 17, se logran apreciar los 8 filtros fundamentales (componentes en color gris) de la arquitectura del programa de automatización. Estos elementos son fundamentales para que el programa sea capaz de generar una solución automática para el acero de refuerzo de muros pantalla.

Con respecto a los dos elementos extras que se encuentran encerrados por un recuadro color morado, son elementos adicionales que diseñé como requerimiento adicional por parte del área de ingeniería, con el propósito de que el usuario pueda proporcionar una solución de manera manual para los apartados de refuerzo adicional de muro, bastones para momentos flexionantes y estribos para fuerzas cortantes.

4.4.3. Implementación de algoritmo genético para optimización

Uno de los principales objetivos del programa es optimizar el área de acero del refuerzo principal. Propuse implementar una técnica de algoritmos evolutivos, en específico el algoritmo genético, para la búsqueda de diámetro de varilla y separación entre varillas que nombramos como el acero principal del muro además de obtener el arreglo óptimo en cuanto a área de acero se refiere.

Existen diversas opciones para implementar algoritmos genéticos en el lenguaje de programación Python. En este trabajo, propuse utilizar la biblioteca de código abierto PyGAD gracias a la facilidad que ofrece al momento de construir los algoritmos y la alta personalización de las funciones fundamentales del algoritmo.

El primer paso que realicé fue la construcción de la función objetivo (fitness function). Esta función es llamada por la instancia del algoritmo genético para generar el parámetro "Fitness", que es el valor a optimizar. La función involucra el área de acero mínima y el área de acero de cada una de las soluciones posibles.

A lo largo del desarrollo realicé diferentes modelos del algoritmo, modificando las operaciones genéticas de cruce, selección de padres y mutación con el objetivo de encontrar la instancia que converja al área de acero mínima en el menor número de generaciones posibles. Obteniendo así un valor absoluto de Fitness lo más cercano a 1.0, que es lo que se considera como óptimo.

Cada individuo del algoritmo genético tiene 2 genomas, el diámetro de varilla (tipo de dato int) con un dominio de [5,12] octavos de pulgada (parámetro de identificación del tipo de varilla, comúnmente empleado en México) y la separación entre varillas (tipo de dato float) con un dominio de [12.5-25] cm.

Es importante mencionar que todos los modelos tienen fijos los siguientes hiper parámetros de la instancia de algoritmo genético de la biblioteca PyGAD:

- **num_generations = 30** - Corresponde al número de generación que realizará el algoritmo.
- **sol_per_pop = 20** - Valor que indica el número de soluciones que son consideradas dentro de la población.
- **num_genes = 2** - Número de genes en la solución. Dicho en otras palabras son los cromosomas que forman a la solución.
- **fitness_func = function_fitness** - Se indica el nombre de la función Fitness que el algoritmo va a utilizar para obtener el valor Fitness.
- **gene_type = [int, float]** - Parámetro que recibe el listado de los tipos de datos correspondientes a los genes.
- **gene_space = [[5,6,8,10,12], {'low':12.5, 'high':25}]** - Parámetro que permite especificar el dominio de valores que puede tomar cada gen del modelo.
- **on_generation = on_generation** - Parámetro para especificar el nombre de la función que será llamada al terminar cada generación del algoritmo. Esta función resulta ser útil para analizar la información y realizar diversas acciones a lo largo del ciclo de vida del algoritmo genético.
- **allow_duplicate_genes = False** - Parámetro para especificar que no existan duplicados en los valores de los genes.
- **mutation_percent_genes = 10** - Se especifica el porcentaje de genes que se van a mutar de toda la población. En este caso se va a mutar el 10% de la población.
- **mutation_num_genes = 2** - Parámetro para especificar el número de genes a mutar. El valor por defecto es ninguno, pero para este caso se especifica que se van a modificar tanto el diámetro de varilla como su separación para evitar caer en mínimos locales o llegar a una solución sesgada.

El exhaustivo proceso de configuración de los modelos de algoritmo genético se llevó a cabo en un tiempo aproximado de 3 semanas realizando pruebas con diferentes muros pantalla.

En este trabajo se muestran 4 modelos de algoritmo genético con los que se realizaron pruebas para un muro pantalla. Es importante mencionar que estos 4 modelos se seleccionaron por ser los que ofrecieron mejores resultados dentro de un conjunto de modelos previamente probados. El modelo elegido como modelo final para la herramienta de software, es el correspondiente al número 2 de la lista de modelos de este capítulo.

Las diferencias principales entre los 4 modelos a mostrar en esta sección, radican en las operaciones genéticas que utilizan, hablando de selección de padres, tipo de mutación y cruce genético. Cada modelo es diferente entre sí al utilizar

combinaciones distintas de estas operaciones genéticas, destacando que las operaciones utilizadas son las que se encuentran implementadas en la biblioteca PyGAD.

En las figuras 19, 21, 23 y 25 se muestran los rendimientos de los 4 modelos de algoritmo genético en donde todos realizan 30 generaciones para encontrar la solución de acero de refuerzo principal.

Se trata de una prueba por modelo, para un muro con las siguientes características: espesor de 60 cm, ancho de 600 cm y profundidad de 27 metros. Las pruebas tuvieron como objetivo encontrar el modelo que converja a la solución óptima en la menor cantidad de generaciones, obteniendo un Fitness absoluto lo más cercano a 1.0 que se traduce en la optimización del uso de acero, teniendo como consecuencia un área de acero muy cercana a la mínima necesaria para el muro.

En la figura 18, se muestra la configuración de la primera instancia del modelo de algoritmo genético, el cual tiene como operación de cruce “uniform”, selección de padres “rank” y tipo de mutación “random”.

```
ga_instance = pygad.GA(num_generations=30,
                       sol_per_pop=20,
                       num_genes=2,
                       num_parents_mating=2,
                       fitness_func=funcion_fitness,
                       gene_type=[int, float],
                       gene_space=[[5,6,8,10,12], {'low':12.5, 'high':25}],
                       on_generation=on_generation,
                       allow_duplicate_genes=False,
                       crossover_type="uniform",
                       parent_selection_type="rank",
                       mutation_type="random",
                       mutation_percent_genes=2,
                       mutation_num_genes=2)
```

Figura 18. Modelo 1 de algoritmo genético - acero principal de muro pantalla.

Al ejecutar el primer modelo, el algoritmo obtiene su convergencia a partir de la generación 25. Arroja como solución óptima un diámetro de varilla del 10 a una separación 12.587 cm y el valor de la razón del área de acero que propone el algoritmo genético entre el área mínima de acero que requiere el muro es igual a -1.0076.

Las gráficas de Fitness vs generaciones indican qué tanto evoluciona el algoritmo al encontrar el mejor resultado al paso de sus generaciones o iteraciones. En este

caso, cuando el valor de Fitness es igual o muy próximo a ser -1.0 se considera la mejor solución.

En la figura 19 se muestra la gráfica de rendimiento del Fitness en cada generación para el algoritmo configurado en la figura 18.

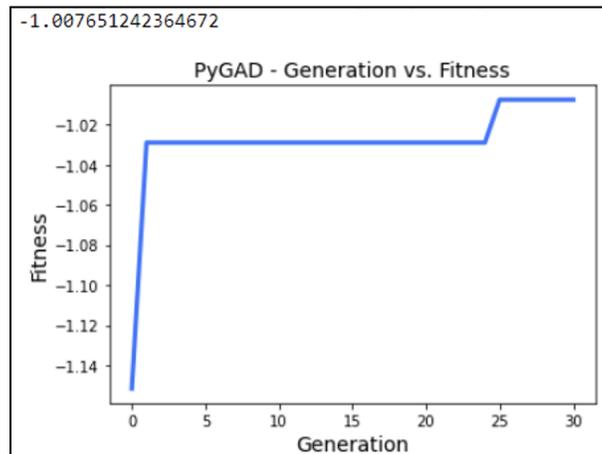


Figura 19. Modelo 1: mejor solución y rendimiento del Fitness.

El segundo modelo de algoritmo genético, figura 20, se utiliza la operación de cruce “uniform”, selección de padres “sus - stochastic universal selection” y mutación “random”.

```
ga_instance = pygad.GA(num_generations=30,
                        sol_per_pop=20,
                        num_genes=2,
                        num_parents_mating=2,
                        fitness_func=funcion_fitness,
                        gene_type=[int, float],
                        gene_space=[[5,6,8,10,12], {'low':12.5, 'high':25}],
                        on_generation=on_generation,
                        allow_duplicate_genes=False,
                        crossover_type="uniform",
                        parent_selection_type="sus",
                        mutation_type="random",
                        mutation_percent_genes=2,
                        mutation_num_genes=2)
```

Figura 20. Modelo 2 de algoritmo genético para acero principal de muro pantalla.

El segundo modelo, bajo las mismas condiciones de muro pantalla, tiene un rendimiento que se aprecia en la figura 21. Este modelo a diferencia del primero, llega a una solución muy similar pero en menor cantidad de generaciones. A partir de la 5ta generación, el algoritmo encuentra la solución final. Como solución óptima se tiene diámetro de varilla del 10 a una separación 12.5656 cm y el valor de la razón del área de acero que propone el algoritmo genético entre el área

mínima de acero que requiere el muro es igual a -1.007651. En la figura 18 se muestra la gráfica de rendimiento del fitness en cada generación.

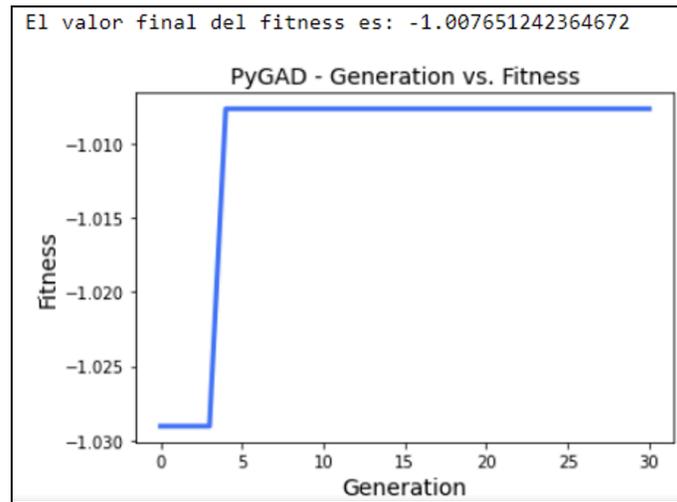


Figura 21. Modelo 2: mejor solución y rendimiento del Fitness por generación.

Tercer modelo de algoritmo genético, figura 22, se utiliza la operación de cruce “uniform”, selección de padres “sss - steady state selection” y mutación “swap”.

```
ga_instance = pygad.GA(num_generations=30,  
                       sol_per_pop=20,  
                       num_genes=2,  
                       num_parents_mating=2,  
                       fitness_func=funcion_fitness,  
                       gene_type=[int, float],  
                       gene_space=[[5,6,8,10,12], {'low':12.5, 'high':25}],  
                       on_generation=on_generation,  
                       allow_duplicate_genes=False,  
                       crossover_type="uniform",  
                       parent_selection_type="sss",  
                       mutation_type="swap",  
                       mutation_percent_genes=2,  
                       mutation_num_genes=2)
```

Figura 22. Modelo 3 de algoritmo genético para acero principal de muro pantalla.

El tercer modelo converge rápidamente a su solución a partir de la segunda generación. Se obtiene como solución un diámetro de varilla del 10, a una separación de 12.80 cm y con una razón del área de acero de la solución propuesta por el modelo entre el área mínima de acero que requiere el muro igual -1.029. En la figura 23 se muestra la gráfica de rendimiento del fitness en cada generación.

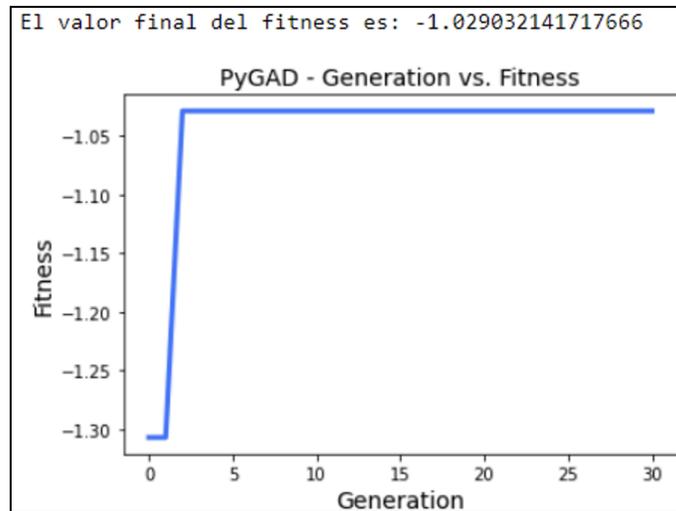


Figura 23. Modelo 3: mejor solución y rendimiento del Fitness.

Finalmente, el cuarto modelo de algoritmo genético, figura 24, se utiliza la operación de cruce “uniform”, selección de padres “sus - stochastic universal selection” y mutación “inversion”.

```
ga_instance = pygad.GA(num_generations=30,
    sol_per_pop=20,
    num_genes=2,
    num_parents_mating=2,
    fitness_func=funcion_fitness,
    gene_type=[int, float],
    gene_space=[[5,6,8,10,12], {'low':12.5, 'high':25}],
    on_generation=on_generation,
    allow_duplicate_genes=False,
    crossover_type="uniform",
    parent_selection_type="sus",
    mutation_type="inversion",
    mutation_percent_genes=2,
    mutation_num_genes=2)
```

Figura 24. Modelo 4 de algoritmo genético para acero principal de muro pantalla.

Este último modelo ofrece su mejor solución a partir de la primera generación del algoritmo genético. Tal como se aprecia en la figura 25, la mejor solución es con diámetro de varilla del 8 con una separación entre varillas de 20.83 cm. Para esta solución, la razón del área de acero de la solución propuesta por el modelo entre el área mínima de acero que requiere el muro es igual a -1.067. En la figura 26 se muestra la gráfica de rendimiento del fitness en cada generación.

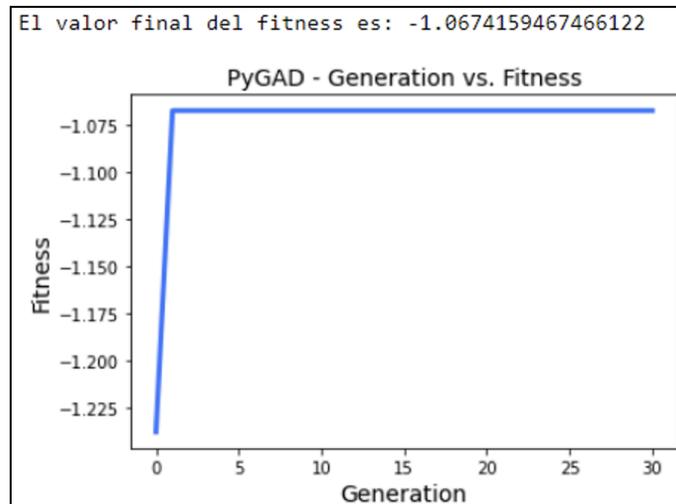


Figura 25. Modelo 4: mejor solución y rendimiento del Fitness por generación.

Después de una serie de pruebas y análisis de los modelos, el algoritmo genético número 2 demostró ser el que mejor optimiza el área de acero encontrando la convergencia de su solución en el menor número de generaciones con un buen rendimiento del fitness.

4.4.4. Construcción de algoritmos solucionadores de refuerzos

Para comprender mejor el funcionamiento de los algoritmos descritos en este apartado, es necesario explicar el significado de las gráficas y los datos que conforman el escenario que el algoritmo resuelve.

El muro pantalla está sometido principalmente a 3 tipos de fuerzas:

- Momentos flexionantes: Estas fuerzas intentan flexionar y deformar al muro.
- Fuerzas cortantes: Intentan provocar fallas súbitas cortando el muro de manera transversal.
- Fuerzas normales: Son aquellas dadas por el peso del muro y las cargas que se apliquen de manera longitudinal al muro, estas últimas no se toman en cuenta para este trabajo.

Dicho lo anterior, los 2 tipos de gráficas a explicar son las gráficas de momentos flexionantes y gráficas de fuerzas cortantes.

Es importante recordar que el muro pantalla es una estructura de contención que se utiliza para estabilizar excavaciones profundas en el subsuelo, como se aprecia en la figura 26.



Figura 26. Ejemplo de funcionamiento de muro pantalla. [25]

Los datos requeridos para conformar las gráficas se denominan “elementos mecánicos”. Estos se obtienen al realizar un análisis del comportamiento del muro en las condiciones en las que será construido. Se toma en cuenta el tipo de suelo, profundidad, fuerzas de empuje y otros criterios definidos en la ingeniería civil. Estos datos dan vida a las gráficas que representan las fuerzas a las que estará sometido el muro a lo largo de la profundidad de la excavación.

En la figura 27 se aprecia una aproximación sencilla del surgimiento de las gráficas, donde se muestra un muro pantalla en servicio y la representación gráfica de las fuerzas actuantes en el.

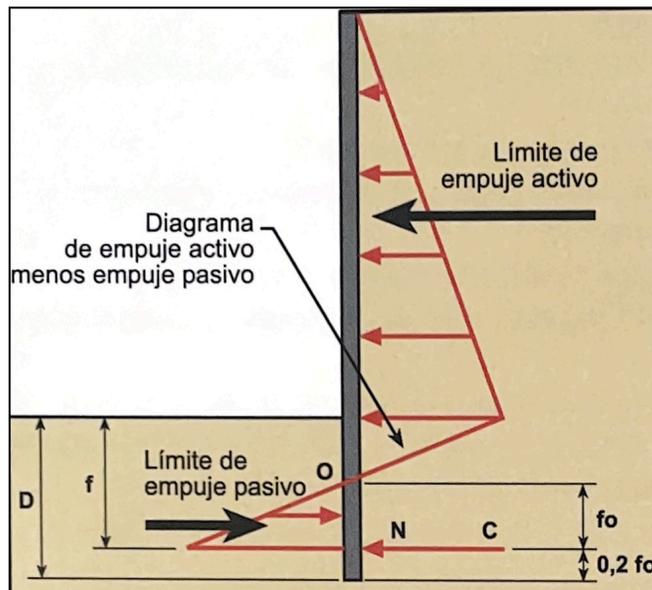


Figura 27. Representación de muro pantalla en servicio [2].

En la figura 28 se muestra una gráfica de momentos flexionantes actuantes ($t \cdot m$) versus profundidad (m), que se obtiene de graficar los datos de elementos mecánicos de momentos actuantes en el muro. En ella se aprecia una explicación de cada elemento presente en la gráfica.

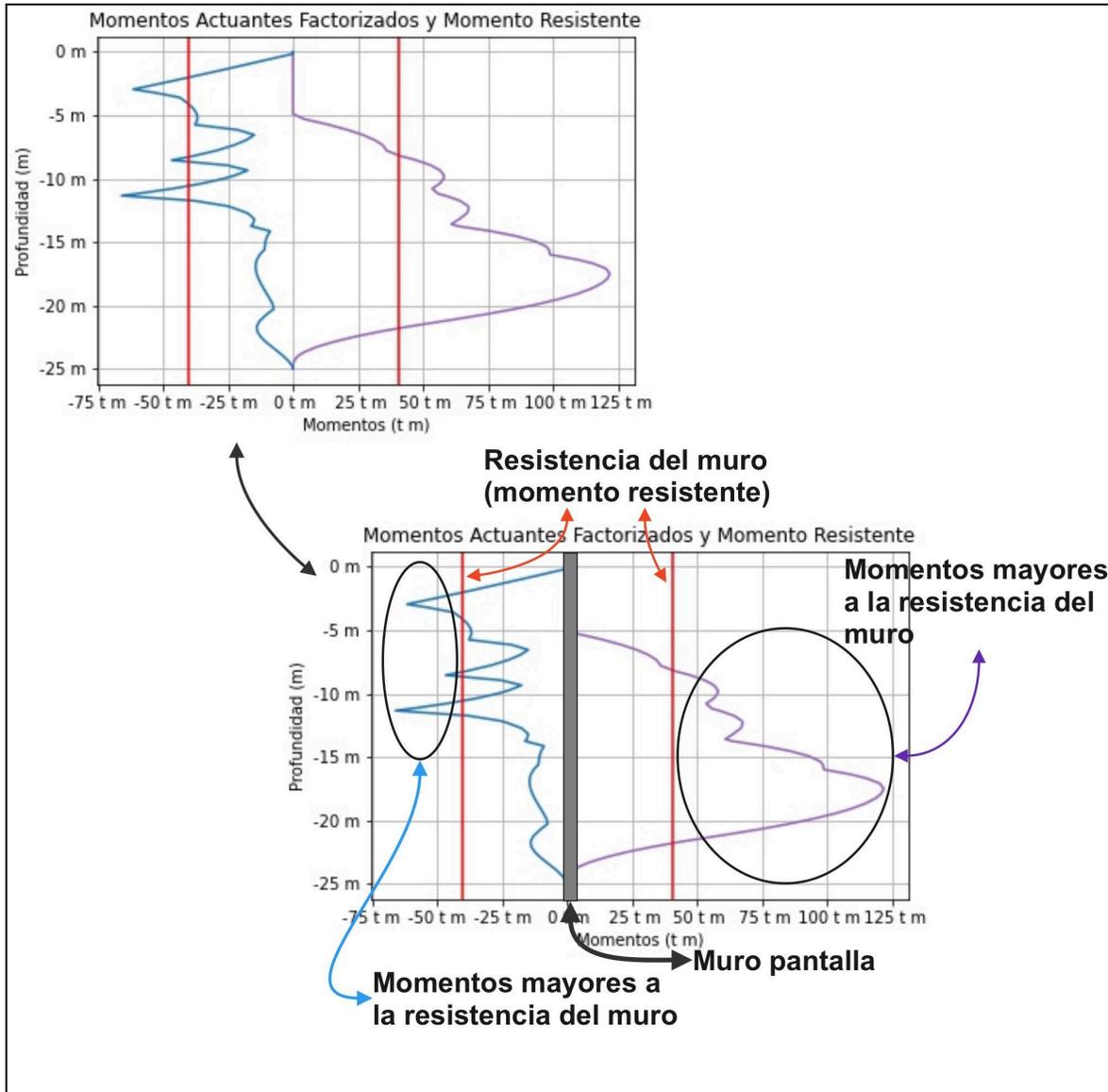


Figura 28. Gráfica de momentos flexionantes actuantes.

Las zonas dentro de las elipses señalan los momentos flexionantes capaces de provocar agrietamientos o inclusive la ruptura del muro. Para solucionar este problema, se debe reforzar el muro en las zonas donde los momentos rebasan la resistencia a la deformación del muro (momento resistente). Esto implica buscar y calcular el arreglo de acero de refuerzo conformado por bastones y colocarlo en las zonas donde sea requerido, protegiendo así al muro de deformaciones y fallas.

Finalmente, en la figura 29 se observa un ejemplo de gráfica de fuerzas cortantes (t) versus profundidad (m), generada a partir de los datos de elementos mecánicos de fuerzas cortantes actuantes en el muro.

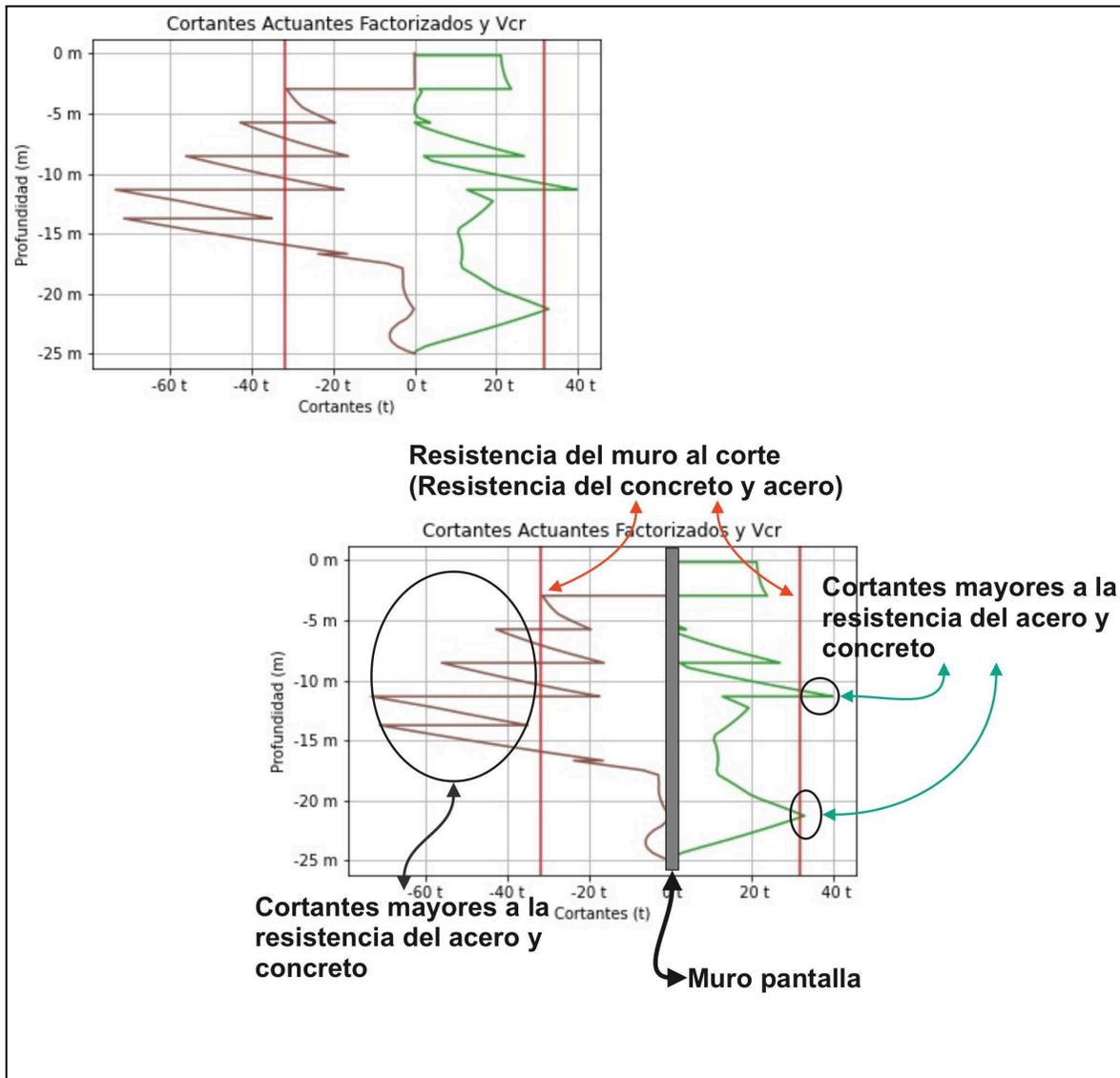


Figura 29. Gráfica de fuerzas cortantes actuantes.

En este caso las elipses indican las profundidades donde las fuerzas cortantes rebasan la resistencia del concreto y del acero que conforman al muro. En esas zonas, el muro fallaría súbitamente de manera transversal lo que provocaría un corte en esa sección.

En respuesta al problema se realiza la búsqueda y cálculo del acero de refuerzo denominado “estribo”. Esto consiste en buscar la resistencia obtenida de combinar el diámetro de varilla y la separación entre estas que sumados a la resistencia del

concreto y acero del muro, eviten el corte transversal causado por las fuerzas cortantes actuantes.

Para resolver los casos donde se requiere de bastones para cubrir momentos flexionantes y estribos para cubrir fuerzas cortantes, desarrollé dos algoritmos que se encargan de analizar los datos de elementos mecánicos para encontrar las zonas del muro donde se requieren refuerzos y posteriormente solucionar el problema calculando y colocando refuerzo adicional.

El algoritmo para la colocación de bastones consiste en la identificación de los elementos mecánicos que rebasan la magnitud del momento resistente en cada zona del muro para el caso de momentos flexionantes.

En las figuras 30 y 31 se ejemplifica el proceso que realiza el algoritmo para la solución de colocación de bastones de manera automática.

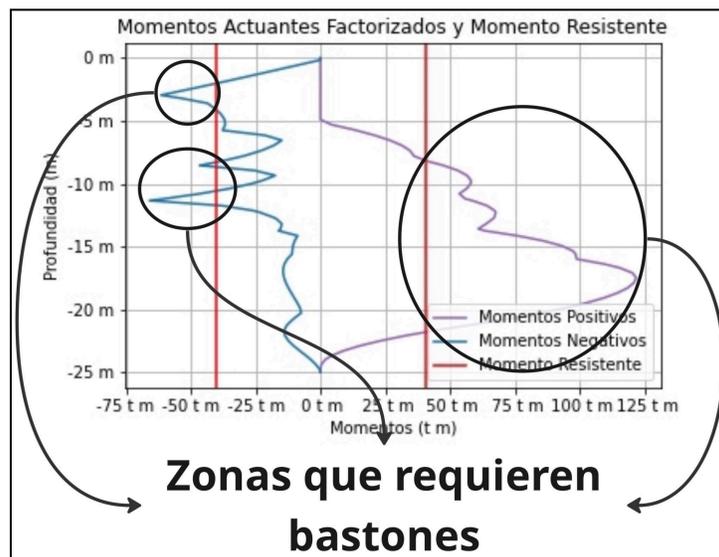


Figura 30. Reconocimiento de zonas que requieren bastones como refuerzo adicional para momentos flexionantes.

Una vez identificados los grupos de elementos mecánicos que rebasan el momento resistente, se realiza una búsqueda iterativa de diámetro de varilla, conservando la separación del acero de refuerzo principal dada por el algoritmo genético para optimización de acero. De modo que la resistencia del bastón en conjunto con la resistencia a flexión del acero de refuerzo principal, sean capaces de resistir los momentos flexionantes en servicio.

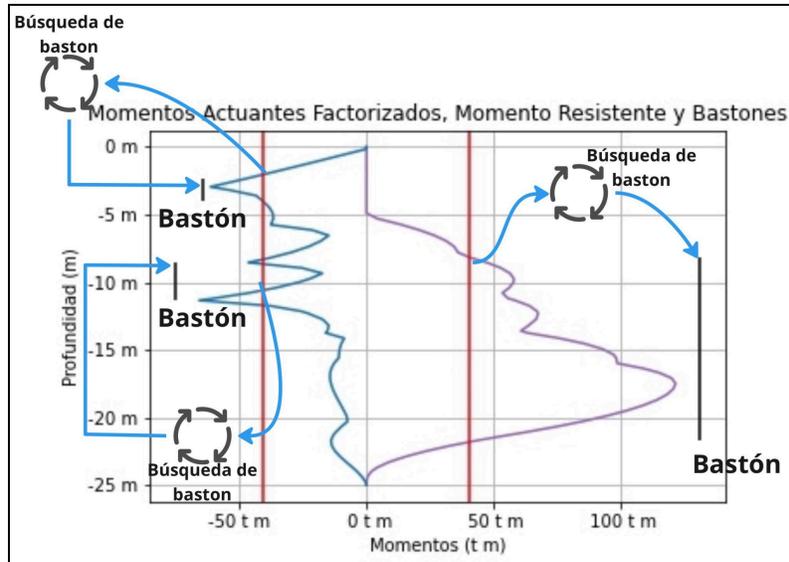


Figura 31. Búsqueda iterativa y colocación de bastones.

Por otra parte, el algoritmo para búsqueda y colocación de estribos se trata de una combinación del algoritmo de búsqueda de bastones y una versión modificada del algoritmo genético para optimización de acero. En este caso, se realiza una búsqueda de diámetro y separación óptima entre estribos haciendo uso de un algoritmo genético configurado para esta tarea y que, en combinación con la resistencia del muro al corte, sea posible cubrir las fuerzas cortantes excedentes.

Posteriormente, diseñé un algoritmo que realiza el análisis de distribución de los estribos obtenidos por el primer algoritmo de colocación de estribos, haciendo un arreglo en espejo y eliminando traslapes entre estribos de ambos lados de la gráfica, manteniendo los estribos de mayor resistencia en caso de intersección. Ver figuras 32 (algoritmo de colocación temporal de estribos) y 33 (algoritmo de distribución final de estribos).

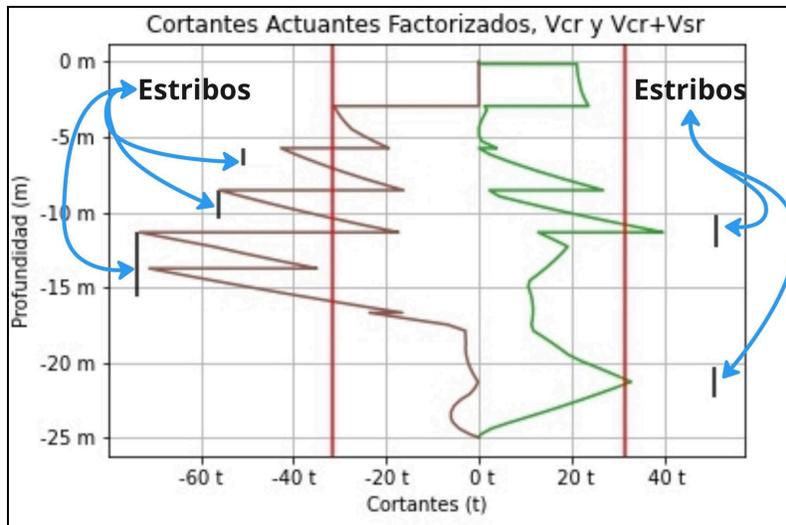


Figura 32. Búsqueda a través de AG y colocación temporal de estribos.

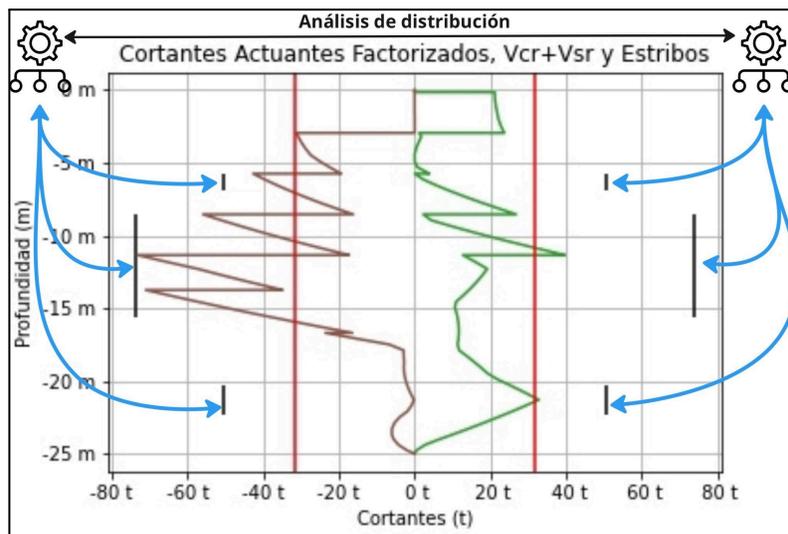


Figura 33. Análisis de distribución y colocación final de estribos.

Estos algoritmos trabajan bajo un conjunto de criterios definidos y acordados con el área de ingeniería. Su objetivo es automatizar la colocación y distribución de refuerzos además de permitir al usuario ajustar la separación entre las zonas que requieren refuerzos para hacer variar la cantidad de estribos que el algoritmo puede llegar a colocar.

4.5. Pruebas

En este capítulo, se revisarán distintas pruebas que se realizaron con el programa para verificar el correcto funcionamiento de los 4 puntos importantes en el proceso de cálculo y diseño de muros pantalla. Probando diferentes datos de elementos

mecánicos y muros con características diferentes, se mostrarán capturas de pantalla de la solución para cada caso de muro.

Las pruebas se llevaron a cabo utilizando información de proyectos que fueron desarrollados por “la constructora” en años anteriores. El objetivo de las pruebas fue monitorear el comportamiento de la herramienta a la hora de proponer soluciones para el acero de refuerzo de los muros pantalla.

Durante las pruebas se esperaba obtener resultados similares a lo que el equipo de ingeniería determinó para cada proyecto, y sobre todo, obtener mejores resultados en cuanto a optimización y la solución de los bastones para momentos flexionantes y estribos para fuerzas cortantes.

En esta etapa del desarrollo se realizaron múltiples pruebas que también permitieron identificar errores en las soluciones automáticas proporcionadas por la herramienta, corregir los mismos y mejorar los modelos de algoritmo genético.

Las pruebas también ayudaron a monitorear el comportamiento de los algoritmos genéticos y de los algoritmos solucionadores de búsqueda y colocación de bastones para momentos flexionantes, búsqueda y colocación de estribos para fuerzas cortantes y la distribución automática de estribos.

En este capítulo se ejemplifica el tipo de pruebas que se realizaron durante la etapa de pruebas. Los datos de salida que arroja la herramienta de software, se compararon con la solución que el área de ingeniería de “la constructora” determinó en su momento y adicionalmente el equipo verificó los cálculos mediante el método manual.

4.5.1. Prueba 1

4.5.1.1. Área de acero de diseño - Prueba 1

Una de las principales funciones del programa es proponer soluciones optimizadas para acero de refuerzo de muros pantalla. El objetivo de esta prueba es realizar diversas corridas del programa para muros con distintas características. Permitiendo obtener diversas soluciones para mirar el comportamiento del programa al desempeñar esta tarea.

La primera corrida es para un muro pantalla con las características mostradas en la figura 34.

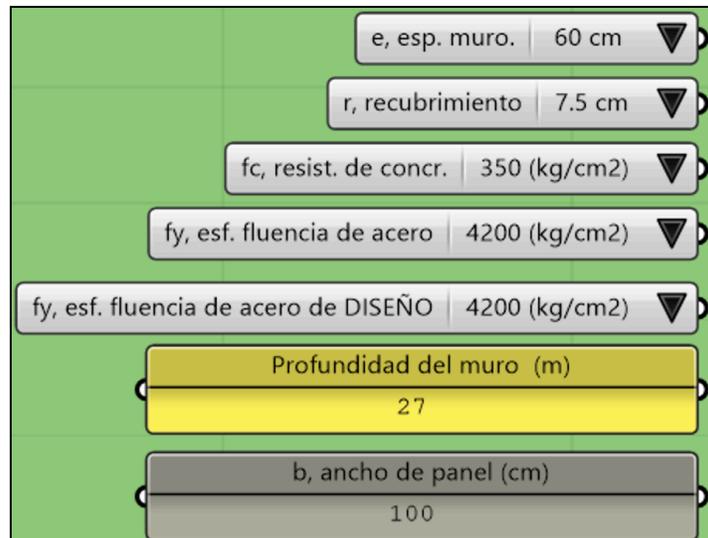


Figura 34. Propiedades de muro pantalla para prueba 1.

Las características destacables de este muro pantalla son la profundidad de 27 m, el espesor de muro que es igual a 60 cm y el ancho del panel de 100 cm.

Tras ejecutar el programa, ver figura 35, el programa proporcionó el área mínima de acero requerida para este muro en particular y la solución óptima que el programa, por medio del algoritmo genético, puede ofrecer.

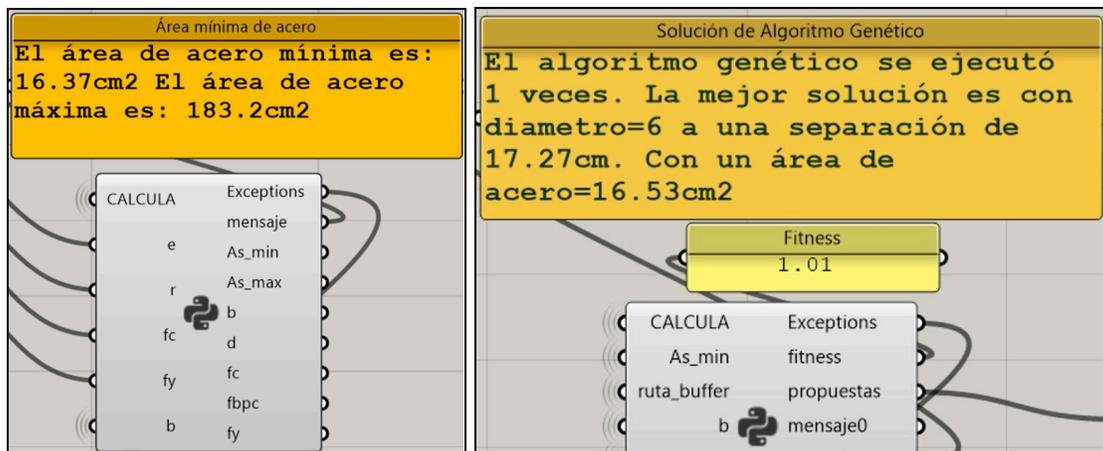


Figura 35. Prueba 1: resultado de área de acero.

De acuerdo con el resultado del programa, el área mínima de acero es de 16.37 cm^2 . La mejor solución automática propuesta es utilizar varilla de diámetro no. 6 a una separación de 17.27 cm. Esta solución representa un área de acero de 16.53 cm^2 , dando como resultado una relación (Fitness) entre el área de acero de diseño y el área mínima de acero igual a 1.01. En la figura 36 se aprecia el rendimiento del Fitness en cada generación del algoritmo genético.

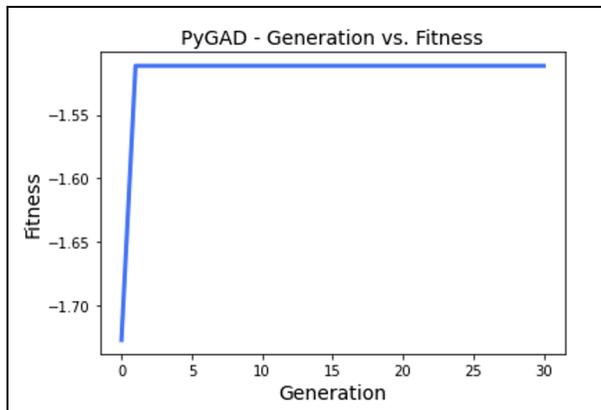


Figura 36. Prueba 1: Gráfica de rendimiento del parámetro Fitness.

4.5.1.2. Solución automática de bastones - Prueba 1

Retomando el muro de la prueba 1, profundidad de 30 m, el espesor de muro igual a 60 cm y ancho del panel de 100 cm, el programa soluciona automáticamente la propuesta de acero adicional para el caso de momentos flexionantes. Ver figura 37.

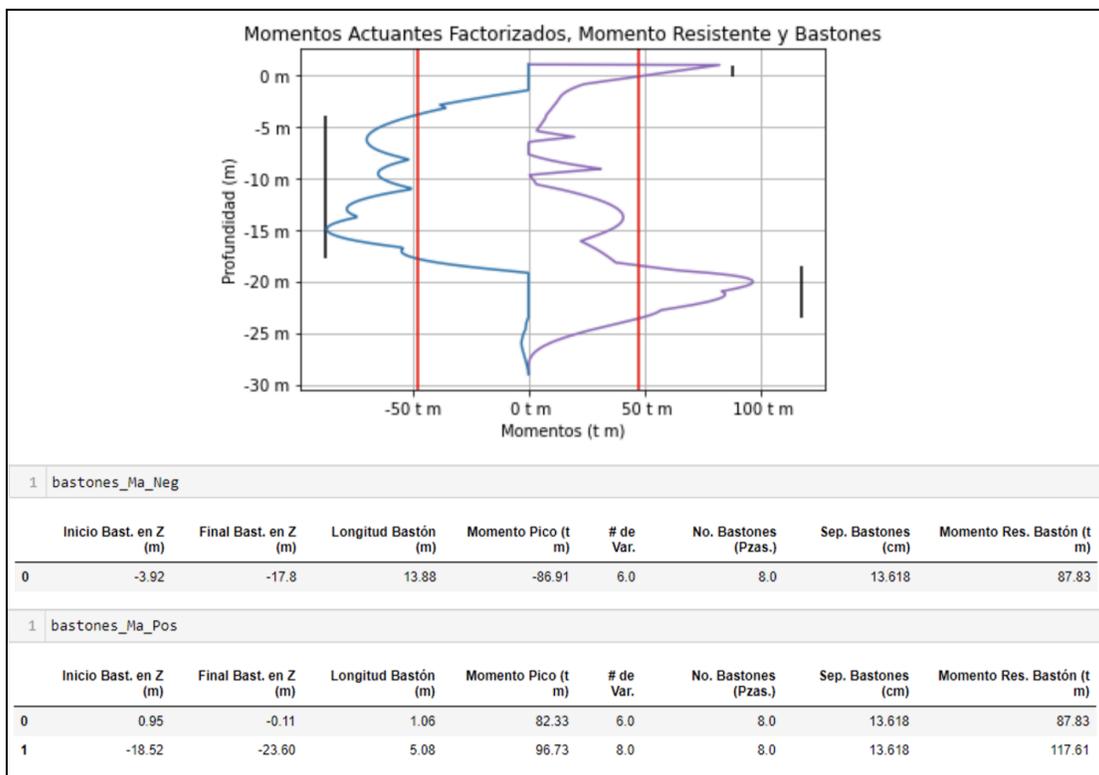


Figura 37. Prueba 1: Solución automática de bastones.

4.5.1.3. Solución automática de estribos - Prueba 1

Los estribos como refuerzo para fuerzas cortantes, para la prueba 1 del muro con profundidad de 27 m, el espesor de muro igual a 60 cm y ancho del panel de 100 cm.

El programa, con ayuda del algoritmo genético modificado para cortantes, encontró la solución mostrada en la figura 38. Este realiza una búsqueda de estribos para cada segmento de elementos mecánicos que rebasan la resistencia del acero y concreto del muro.

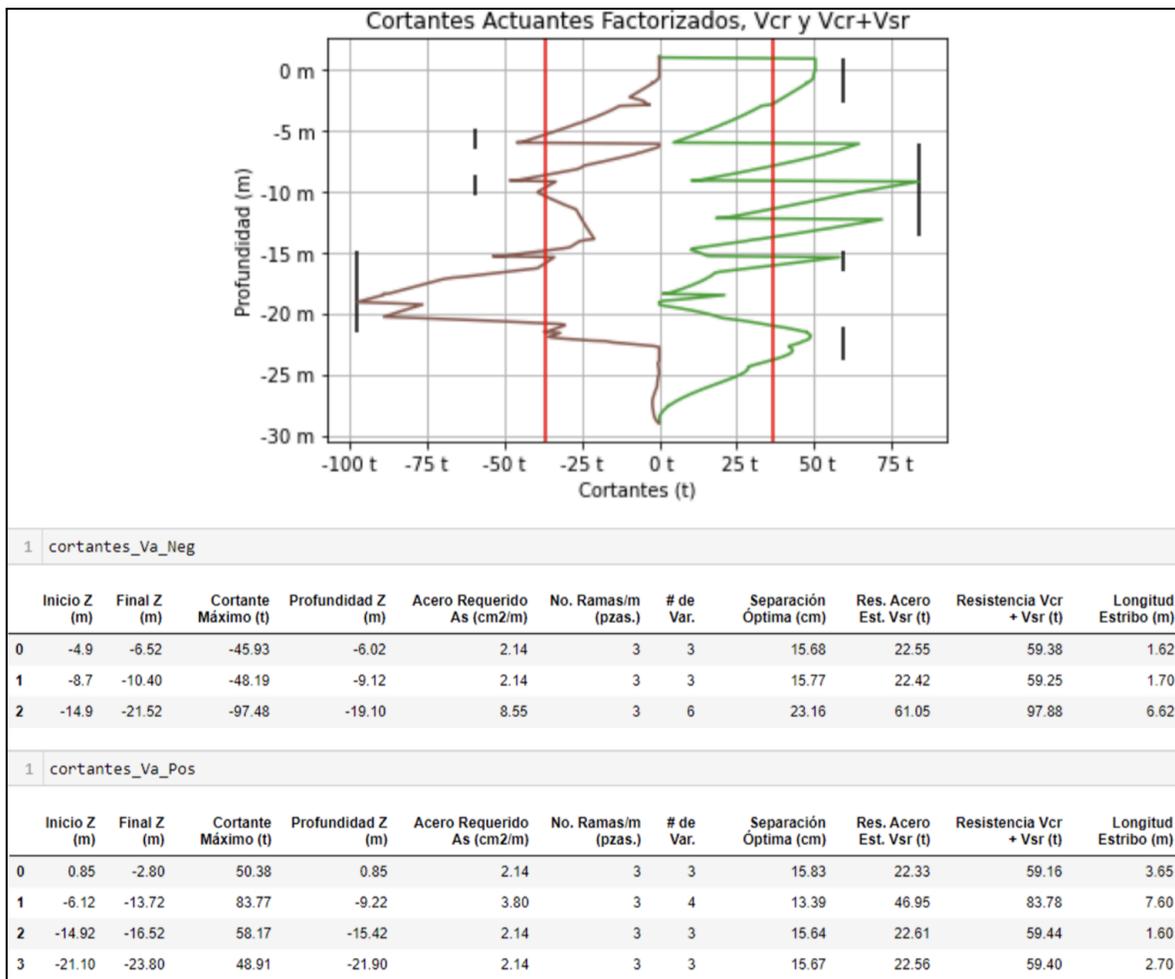


Figura 38. Prueba 1: Búsqueda de estribos mediante AG.

Durante la primera fase de la solución de cortantes, el algoritmo se encarga de reconocer las zonas donde se requieren estribos adicionales, ya que por defecto el muro debe llevar un arreglo base incluso por el propio proceso constructivo. Una vez reconocidas estas zonas, mediante el modelo de algoritmo genético para el cálculo de resistencia por cortante, el programa realiza una propuesta optimizada para cada estribo necesario.

Lo anterior puede ser verificado con ayuda de las figuras 39 y 40. En ellas se aprecia las gráficas correspondientes a cada estribo calculado por el algoritmo.

Para cada estribo necesario, el algoritmo genético realiza 50 generaciones por estribo para encontrar el diámetro de varilla y separación que soportan los elementos mecánicos en servicio.

Para esta prueba, se necesitan 7 estribos y cada uno fue calculado con el algoritmo genético obteniendo el diámetro de varilla y la separación ideal para cubrir los elementos mecánicos excedentes evitando la sobreestimación en las resistencias de los estribos.

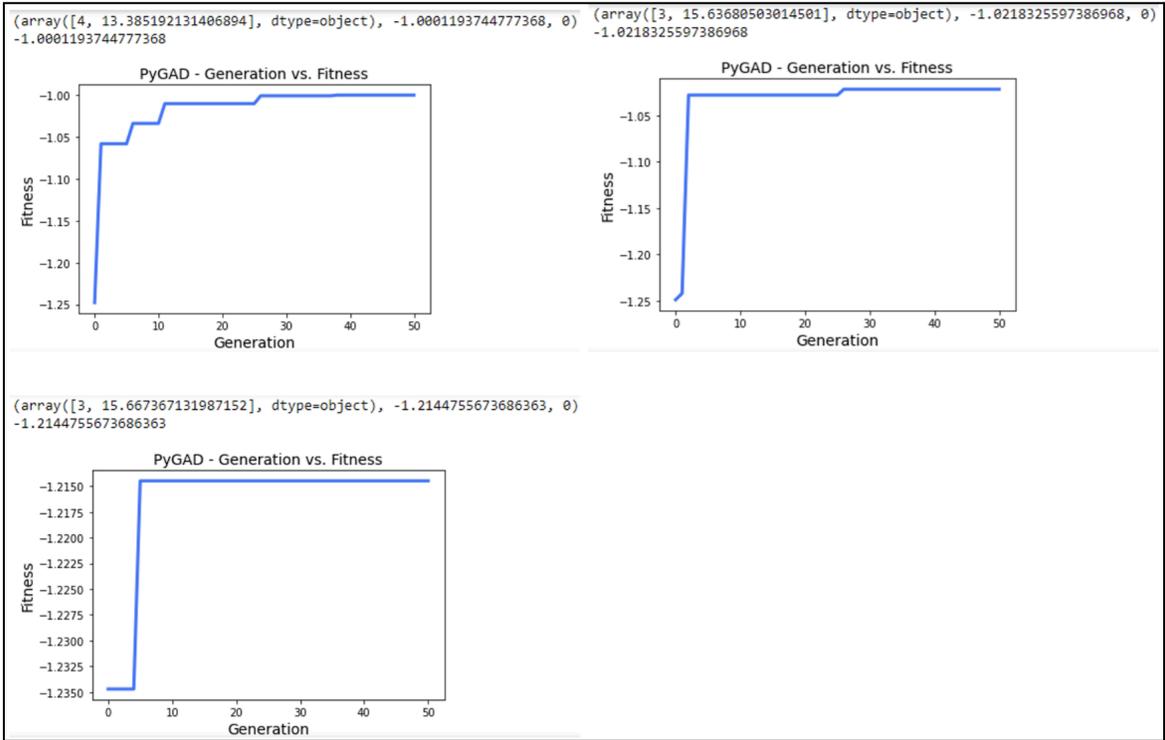


Figura 39. Prueba 1: Rendimiento del Fitness - Estribos lado negativo.

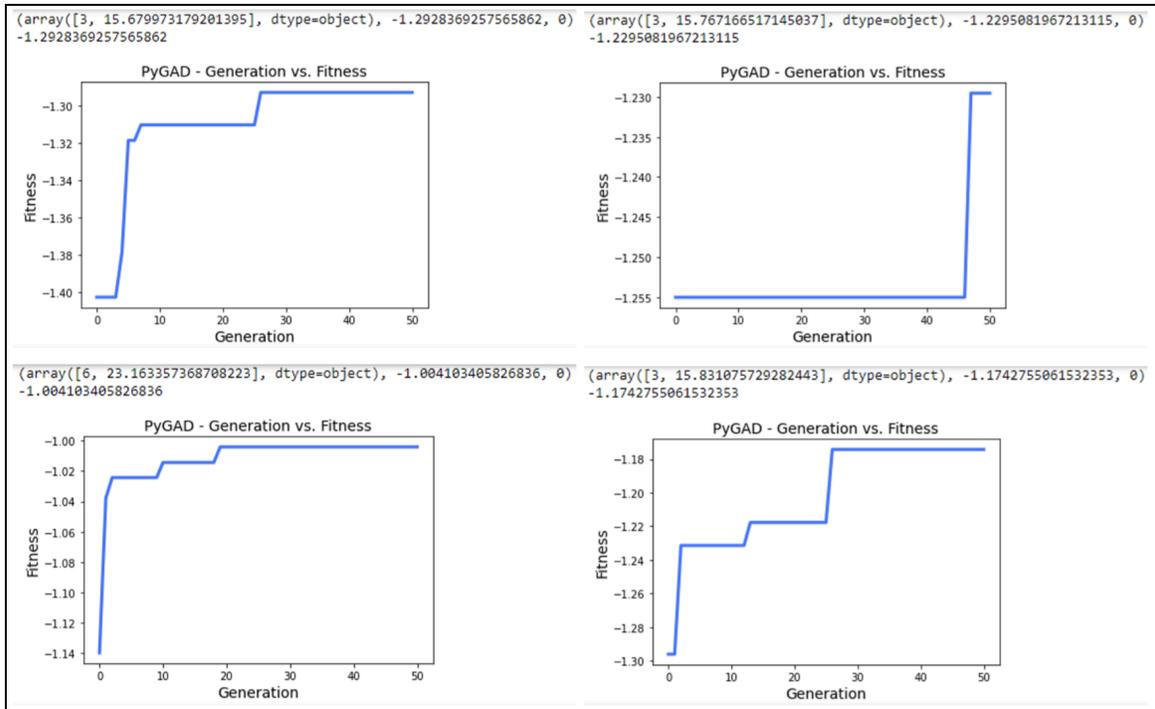


Figura 40. Prueba 1: Rendimiento del Fitness - Estribos lado positivo.

Posteriormente, se ejecuta el algoritmo de análisis de distribución de estribos para finalizar la colocación de estribos en el diseño final. Como parte de la solución se proporciona una gráfica y una tabla con las características de cada uno de los estribos pertenecientes a la misma. Ver figura 41.

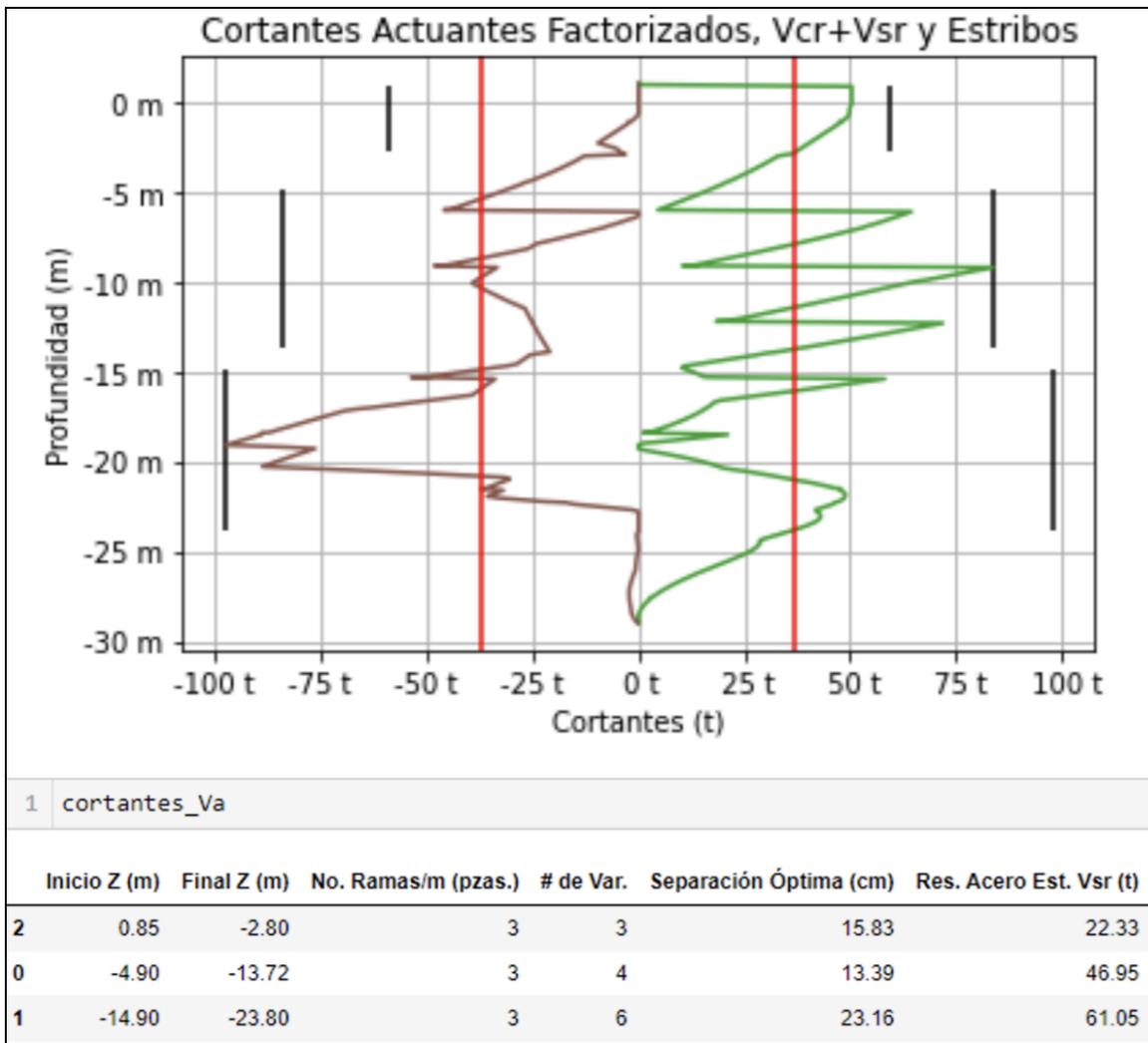


Figura 41. Prueba 1: Análisis distribución y solución final de estribos.

4.5.2. Prueba 2

4.5.2.1. Área de acero de diseño - Prueba 2

Para el caso de prueba 2, con un muro Milán que tiene propiedades diferentes (ver figura 42) se genera otra ejecución del programa.



Figura 42. Propiedades de muro pantalla para prueba 2.

Las características más importantes de este muro pantalla son la profundidad de 15 m, el espesor de muro que es igual a 60 cm y el ancho del panel, que es de 100 cm.

El programa entrega la solución, mostrada en la figura 43, para el área mínima de acero.

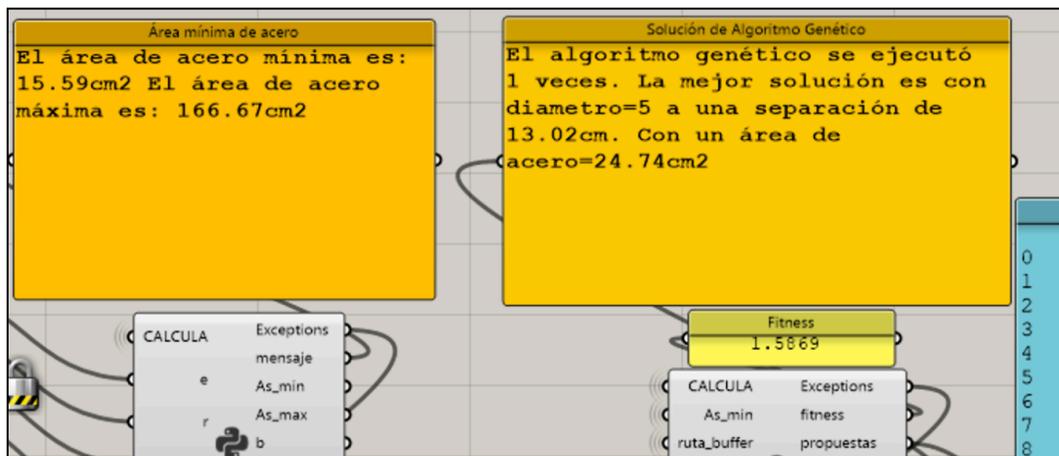


Figura 43. Prueba 2: resultado de área de acero.

De acuerdo con el resultado del programa, el área mínima de acero es de 15.59 cm^2 . La mejor solución automática propuesta es utilizar varilla de diámetro no. 5 a una separación de 13.02 cm. Esta solución representa un área de acero de 24.74 cm^2 , dando como resultado un Fitness igual a 1.5869.

En la figura 44 se puede observar el rendimiento del Fitness durante la ejecución del algoritmo genético.

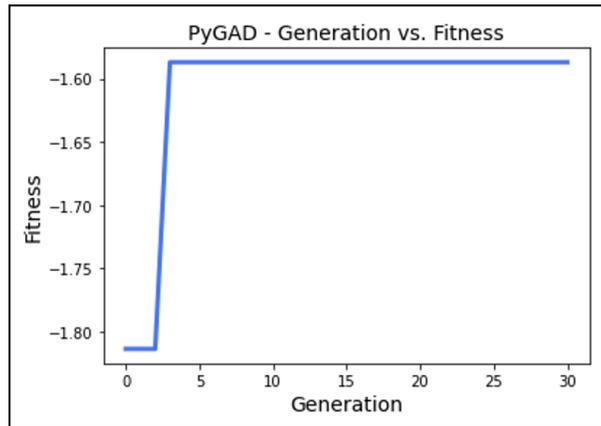


Figura 44. Prueba 2: Gráfica de rendimiento del parámetro Fitness.

4.5.2.2. Solución automática de bastones - Prueba 2

El muro con profundidad de 45 m, espesor de muro igual a 50 cm y ancho del panel de 100 cm, correspondiente a la prueba 2, el programa propone la solución para momentos flexionantes que exceden al momento resistente. A continuación, en la figura 45, se muestra la solución obtenida por el programa.

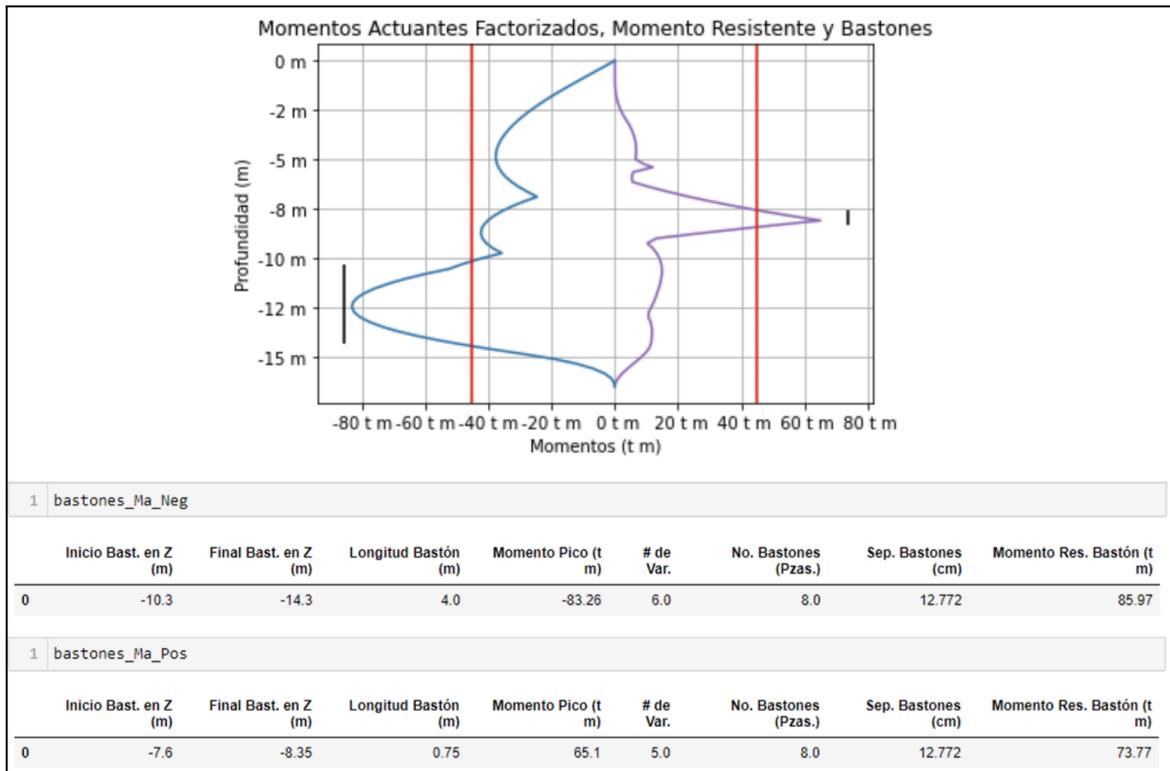


Figura 45. Prueba 2: solución automática de bastones.

4.5.2.3. Solución automática de estribos - Prueba 2

Para la prueba 2 del muro con profundidad de 45 m, espesor de muro igual a 50 cm y ancho del panel de 100 cm, el algoritmo analiza los datos de elementos mecánicos para proponer una solución de acero de refuerzo adicional.

El programa realiza una búsqueda de estribos para cada segmento de elementos mecánicos que rebasan la resistencia del acero y concreto del muro (línea vertical en color rojo).

La primera fase del algoritmo se puede apreciar en la figura 46. Donde se hace el reconocimiento de las zonas que requieren estribos y el cálculo de cada estribo.

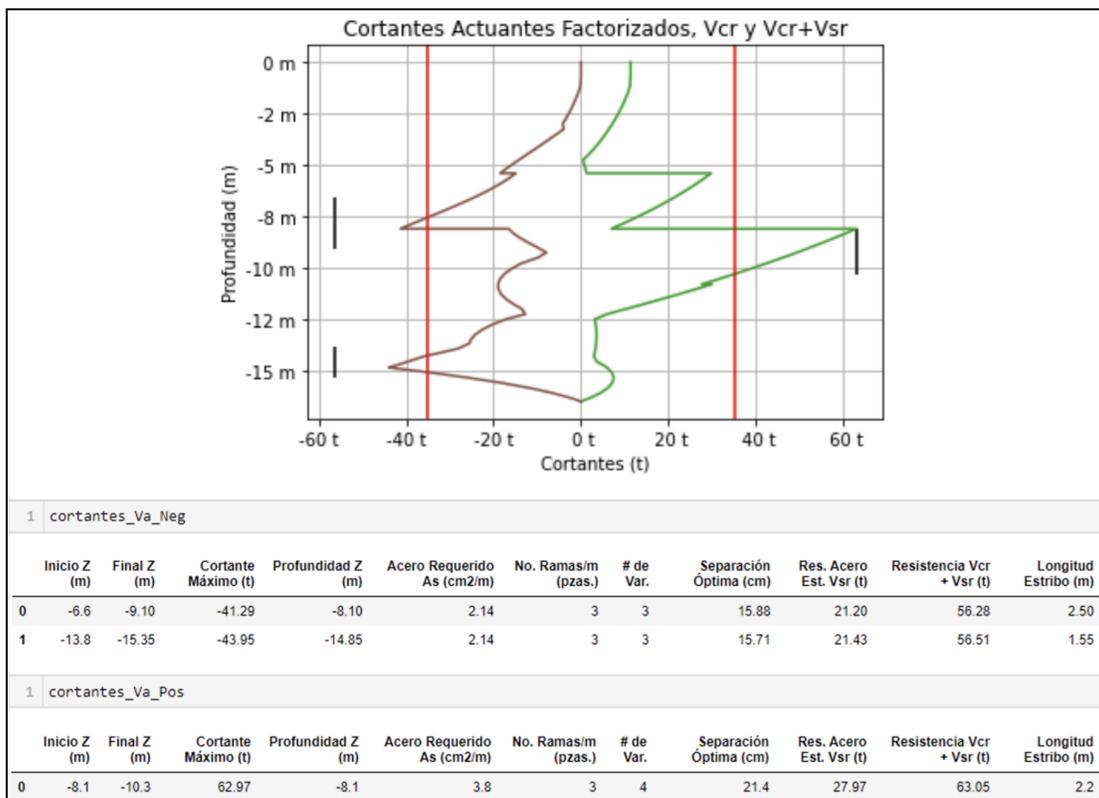


Figura 46. Prueba 2: Búsqueda de estribos mediante AG.

En esta prueba, la gráfica indica que son 3 zonas las que requieren estribo de refuerzo. El programa lo detecta y gracias al algoritmo genético se obtiene el diámetro de varilla y separación para cada estribo.

En la figura 47 y 48, se muestra la gráfica de rendimiento del parámetro Fitness correspondiente a cada estribo colocado. Para cada caso se logra apreciar el diámetro y separación asignada a cada estribo, así como el valor final, en negativo, de su Fitness.

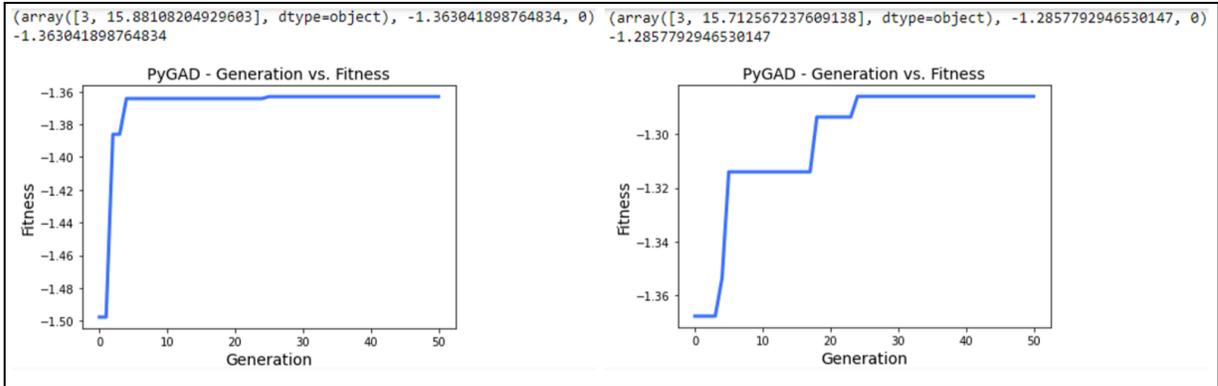


Figura 47. Prueba 2: Rendimiento del Fitness - Estribos lado negativo.

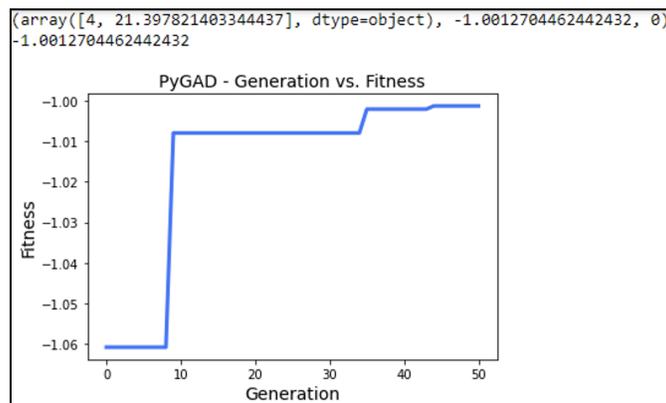


Figura 48. Prueba 2: Rendimiento del Fitness - Estribos lado positivo.

Finalmente, el programa realiza el análisis de distribución para colocar de manera correcta los refuerzos calculados. Este análisis es importante porque define los estribos que se irán al diseño final del muro.

En esta prueba se puede observar la modificación que realiza el algoritmo. Reemplaza los dos refuerzos de la parte media de la gráfica corrigiendo la intersección entre ellos y colocándolos en ambas caras de la gráfica (parte positiva y parte negativa). Esta corrección consiste en el cambio de longitud del estribo manteniendo el estribo de mayor resistencia.

En la figura 49 se observa la solución final para la prueba 2, quedando una solución de dos estribos. En la tabla que se encuentra debajo de la gráfica, se aprecian las características de cada estribo. Información necesaria para el diseño del muro pantalla.

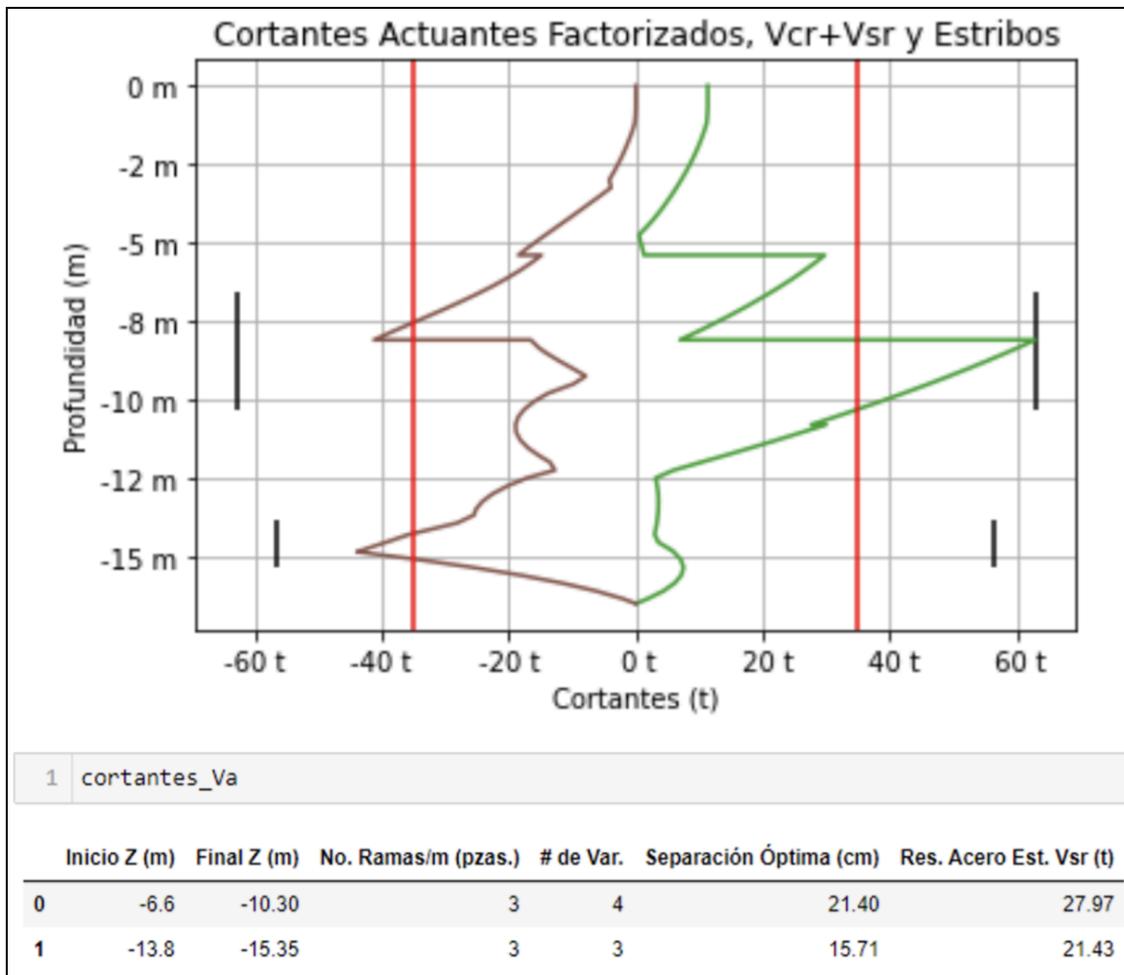


Figura 49. Prueba 2: Análisis y solución final de estribos.

4.6. Liberación a producción

Después del periodo de pruebas y validación de los requerimientos funcionales, se agendó una reunión formal con los usuarios que utilizarán la aplicación, en este caso el área de ingeniería de “la constructora”. Al mismo tiempo me encargué de enviar comunicados a los usuarios finales que realizan el proceso de cálculo y diseño de muros pantalla, para que estuvieran enterados de los nuevos cambios y así mismo, brindarles capacitación. En este punto es dónde se dió por concluida la construcción del proyecto.

Posterior a la liberación, el programa comenzó a ser utilizado inmediatamente en nuevos proyectos de “la constructora”. La transición del método manual al uso de la herramienta de software se dió de manera fluida y sin problema alguno.

El primer proyecto donde la constructora decidió implementar el uso de las herramientas de software fue en un hospital que en su momento se estaba construyendo en Tijuana, Baja California. En dicho proyecto se utilizó el programa 1 para obtener la mejor distribución de paneles de muro, el programa 2, que es el descrito en este trabajo, para resolver el acero de refuerzo de cada tipo de panel arrojado por el programa 1 y finalmente, el programa 3 para generar el modelo en 3D a partir de la información de salida de los programas 1 y 2.

En esta primera implementación, “la constructora” obtuvo muy buenos resultados durante el desarrollo del proyecto del hospital de Tijuana. Se obtuvieron los planos de taller de los muros pantalla a partir de las herramientas de software, reduciendo considerablemente los tiempos de trabajo en el área de ingeniería y el área de dibujo.

4.7. Documentación y mantenimiento

A lo largo del desarrollo redacté un documento con las características principales del proyecto y la descripción de las actividades hechas en cada etapa. Incluí la explicación de los algoritmos construidos en cada etapa y la estructura de la aplicación con el objetivo de facilitar su comprensión en caso de tener nuevos miembros en el equipo y actualizaciones a corto o largo plazo.

De manera adicional elaboré un manual de usuario que describe el funcionamiento de la aplicación, incluyendo errores comunes de operación, errores de funcionamiento y sus posibles soluciones. Dicho manual también sirve para la capacitación de los usuarios finales.

Para brindar el soporte, se estableció un periodo de 6 meses de mantenimiento correctivo al programa en donde el área de ingeniería me comunicó las incidencias con el programa o algún detalle de cálculo al momento de recibir las soluciones automáticas por parte de la herramienta de software.

5. Resultados

5.1. Comparación del tiempo de solución entre la herramienta de software y el método manual

El tiempo necesario para realizar el proceso de cálculo y diseño del acero de refuerzo de muro pantalla, utilizando el programa, está en un promedio de 1 - 1:30 horas. Este tiempo se obtuvo tras consultar con los usuarios el tiempo que invierten para calcular el acero de refuerzo de un muro pantalla desde la preparación de los archivos de datos de elementos mecánicos, preparación de la

información en el formato especificado para el programa y la obtención de un diseño final.

Los usuarios lograron ahorrar tiempo con el uso del programa:

- El método manual toma alrededor de medio día a un día laboral (**7 horas promedio de trabajo**) para tener listo el diseño de acero de refuerzo del muro pantalla.
- Mientras que al ingeniero le toma alrededor de **20 minutos a 1 hora y 30 minutos** tener un diseño utilizando el programa de automatización.

Se habla de un ahorro en tiempo promedio de 82% contemplando el caso donde el ingeniero realice modificaciones a la solución entregada por la herramienta de software por diversos factores, por ejemplo la indisponibilidad de algún diámetro de varilla o cambios en las condiciones constructivas, esto en un caso extraordinario.

A continuación, se presenta una tabla comparativa de tiempos estimados entre el proceso manual y el programa de automatización para el cálculo y diseño de acero de refuerzo de muros pantalla. La información presentada corresponde a un análisis comparativo que tiene como finalidad ejemplificar los cambios con la propuesta de mejora a este procedimiento. Consultar tabla 5.

Proceso manual	Tiempo	Proceso con el programa	Tiempo
El ingeniero de proyecto toma los datos de elementos mecánicos obtenidos con software especializado y los traslada a una hoja de cálculo excel para hacer un análisis de la información.	1 hora	Preparar el archivo de entrada con los datos de elementos mecánicos e ingresar al programa, seleccionar la ruta del archivo de datos y la ruta de la carpeta de salida.	5 min.
Se elige, de manera manual e iterativa, un diámetro de varilla y separación entre varillas para verificar que el área de acero propuesta, cumpla con lo establecido en el reglamento de construcción, (As.min.).	1 hora	El programa automáticamente realiza la propuesta de diámetro de varilla y separación entre varillas para el acero de refuerzo principal. El usuario tiene la opción de ingresar su propia solución de manera manual y que el programa lo calcule y valide.	0.5 - 2 seg.

Cálculo del momento resistente y graficación de momentos actuantes en servicio. Esto se realiza de forma manual para cada elevación y en cada cara del muro para definir el refuerzo donde se tengan elementos mecánicos máximos.	1 hora	A partir de la solución encontrada por el programa, se calcula y grafica automáticamente los momentos.	1-2.5 seg.
El ingeniero, de manera manual e iterativa, define la colocación de acero adicional para momentos en las zonas de mayor demanda, proponiendo nuevamente diámetro y separación de varillas hasta cubrir los momentos máximos a los que será sometido el muro.	1.5 horas	En caso de ser necesario, el programa automáticamente realiza la colocación de bastones como refuerzo adicional al detectar elementos mecánicos que rebasen la resistencia a flexión del muro. El programa define el diámetro del bastón.	0.5-1.5 seg.
Cálculo del cortante resistente del muro y graficación de fuerzas cortantes actuantes en servicio.	1 hora	Graficación automática de datos y cálculo del cortante resistente del muro pantalla.	1-2 seg.
De manera manual e iterativa, se define la colocación de acero adicional para cortantes en las zonas de mayor demanda, proponiendo diámetro y separación de entre varillas hasta cubrir las fuerzas cortantes máximas a las que será sometido el muro. Además, realizar el análisis de distribución de estribos de manera manual.	1.5 horas	Búsqueda automatizada de diámetro y separación de estribos para las zonas del muro que requieran refuerzo por cortantes. Posteriormente, el programa realiza el análisis de distribución de estribos para entregar la solución final.	1.5-3 seg.
-	-	Tiempo de afinación de la solución. En este tiempo el ingeniero decide si realizar cambios o conservar la solución obtenida de manera automática.	20 min.- 1 hora 30 min.

Tiempo total invertido con el método tradicional para obtener el diseño de un tipo de muro pantalla:	7 horas	Tiempo total invertido utilizando el programa de automatización del cálculo y diseño de muros pantalla para obtener el diseño de un tipo de muro pantalla:	20.5 min. ⁵
--	---------	--	------------------------

Tabla 5. Comparación de tiempos entre el método manual y el programa de automatización.

5.2. Comparación de resultados entre la herramienta de software y el método manual

En esta sección voy a mostrar una comparativa detallada de los resultados obtenidos con la herramienta de software y los obtenidos con el método manual por parte del área de ingeniería de “la constructora”, mismo que consiste en realizar los cálculos a través de una hoja de excel.

El ejercicio a desarrollar en este apartado utilizara el mismo muro pantalla para ambas metodologías, tratándose de un muro con las siguientes características:

- Espesor de muro: 60 *cm*.
- Recubrimiento: 7.5 *cm*.
- Factor de resistencia del concreto (f_c): 350 kg/cm^2 .
- Fluencia del acero (f_y): 4200 kg/cm^2 .
- Ancho del muro: 100 *cm*.
- Factor de resistencia a flexión (Fr_M): 0.9.
- Factor de resistencia a cortante (Fr_V): 0.75
- Factor de carga (FC): 1.4.

La entrada de datos se observa de la siguiente forma en la herramienta de software, ver figura 50:

⁵ 20.5 min. en caso de que se utilice la solución entregada por la herramienta de software y hasta 1 hora 30 min. en caso de que se realicen modificaciones para ajustar la solución a casos extraordinarios.

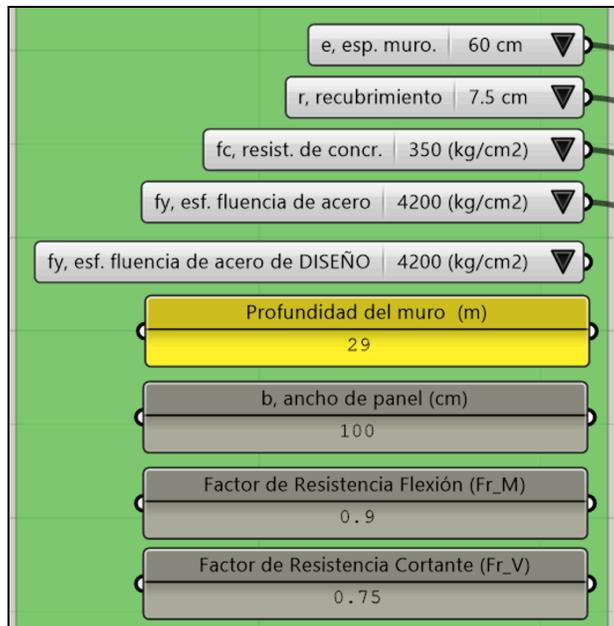


Figura 50. Comparación de resultados: datos de entrada en herramienta de software.

Es importante mencionar que la herramienta de software necesita como parámetros de entrada el archivo de excel donde se encuentran los datos de elementos mecánicos del muro, especificando el nombre de la hoja para datos de momentos flexionantes y la hora para datos de fuerzas cortantes, así como la ruta de la carpeta de salida donde el programará almacenará la solución para el muro. Lo descrito anteriormente se puede observar en la figura 51.

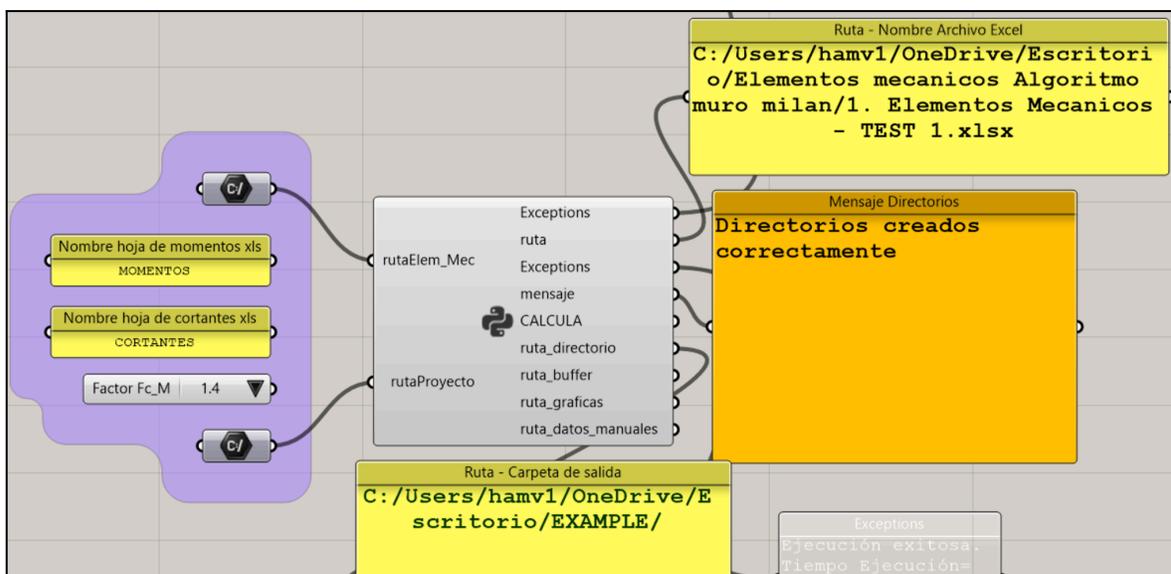


Figura 51. Comparación de resultados: especificación de rutas en herramienta de software.

Hablando del método manual, la hoja de excel que utilizaba el área de ingeniería de “la constructora” requería que se cargaran los datos de elementos mecánicos de momentos flexionantes, fuerzas cortantes y características del muro en el mismo archivo excel que contiene las fórmulas y reglas necesarios para llevar a cabo el proceso de cálculo y diseño del acero de refuerzo de muros pantalla.

En la figura 52 se observan los datos de entrada en la hoja de excel para el método manual:

Diseño a Flexion - Una sola cara de refuerzo.			
ACERO DE REFUERZO			
Barra	Diametro	Area	Peso
No	(cm)	(cm ²)	(Kg/m)
2	0.64	0.32	0.249
3	0.95	0.71	0.559
4	1.27	1.27	0.994
5	1.59	1.98	1.554
6	1.91	2.85	2.237
7	2.22	3.88	3.045
8	2.54	5.07	3.978
10	3.18	7.92	6.215
11	3.49	9.58	7.520
12	3.81	11.40	8.950
	f'c=	5500	Psi
	f'c=	387	kg/cm2
	f'c=	400	kg/cm2
	FC=	1.4	
Momentos flexionantes y cortantes en servicio		Momentos flexionantes y cortantes últimos	
M _s	V _s	M _u	V _u
(t-m/m)	(t/m)	(t-m/m)	(t/m)
8.45	86.93	11.8	121.7
Diseño a Flexion			
Espesor Muro	e	cm	60
Recubrimiento de acero principal - inferior.	r	cm	7.5
Recubrimiento de acero principal - superior.	d'		52.5
Ancho efectivo seccion de analisis.	b	cm	100
Resistencia compresion concreto.	f'c	Kg/cm ²	350
Esfuerzo de fluencia del acero de refuerzo.	f'y	Kg/cm ²	4200
Esfuerzo de fluencia del acero de refuerzo * 0,6.	f'y	Kg/cm ²	3780
Factor de reduccion de resistencia concreto.	φ		0.85
	f'c		297.5
Factor de resistencia.	FR		0.9

Figura 52. Comparación de resultados: datos de entrada en hoja de excel del método manual.

5.2.1. Áreas de acero: mínima y máxima

Los primeros dos resultados a comparar, dentro de todo el proceso del cálculo y diseño del acero de refuerzo de muros pantalla, son el área de acero mínima y máxima. En ambos casos se espera obtener resultados iguales, ya que ambas metodologías utilizan las mismas fórmulas y procedimientos para obtener estos valores.

En la figura 53 se puede observar el resultado que proporciona la herramienta de software:

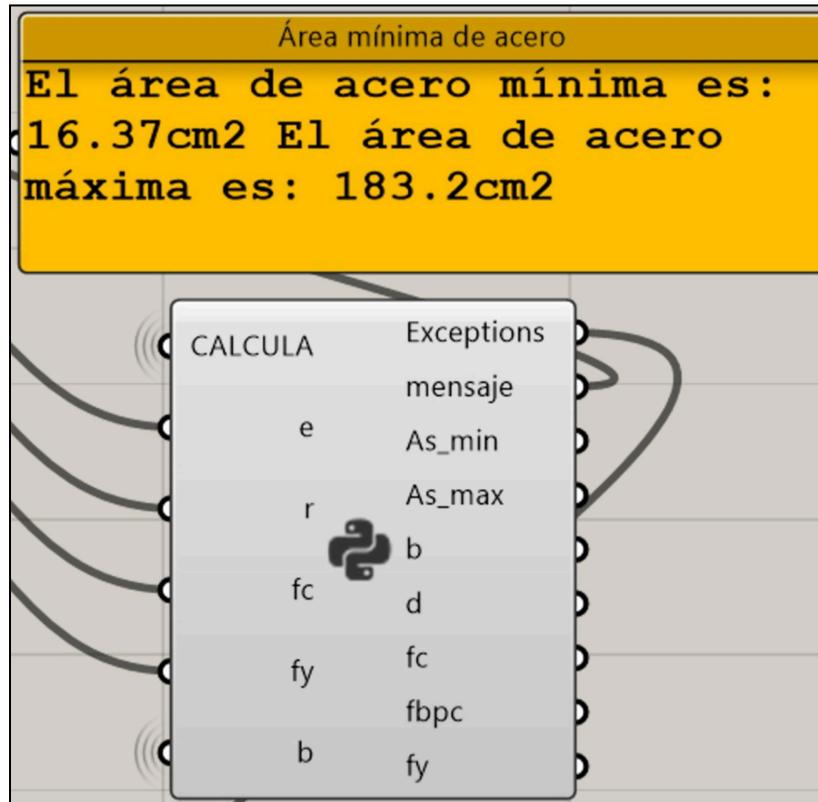


Figura 53. Comparación de resultados: área mínima y máxima de acero proporcionadas por la herramienta de software.

De lo anterior se obtiene que:

- Área de acero mínima: 16.37 cm^2
- Área de acero máxima: 183.2 cm^2

En el método manual, se obtienen los resultados mostrados en la figura 54.

Asmin	cm^2/m	16.37	
		0.84	
Asmax	cm^2/m	183.2	OK

Figura 54. Comparación de resultados: área mínima y máxima de acero proporcionadas por método manual.

De la figura 54 se observan los siguientes resultados:

- Área de acero mínima: 16.37 cm^2
- Área de acero máxima: 183.2 cm^2

Se puede apreciar que ambas metodologías obtienen los mismos valores para el área de acero mínima y máxima.

5.2.2. Propuesta de solución para acero de refuerzo principal

En este paso existe una diferencia importante entre la herramienta de software y el método manual.

En el método manual el ingeniero de proyecto realiza una selección de diámetro de varilla (pudiendo elegir los valores 5, 6, 8, 10 y 12) y separación entre varillas (pudiendo elegir valores entre 12.5cm y 25cm), mismos que ingresa en su hoja de excel para conocer el área de acero de su solución. Su objetivo es encontrar aquella solución cuya área de acero sea lo más próxima al área de acero mínima. El ingeniero hace varias pruebas hasta encontrar la solución que bajo su criterio considere óptima.

Con la herramienta de software, el programa propone la mejor solución que encuentra a través del algoritmo genético que tiene programado. Este busca la combinación, de diámetro de varilla y separación entre varillas, que represente un área de acero lo más cercana al área de acero mínima que se calcula en el paso anterior.

El programa tiene un parámetro llamado Fitness, este valor es una razón del área de acero de la solución y el área de acero mínima para el muro.

Es importante recordar que los valores esperados para el Fitness de propuestas válidas debe cumplir la regla: $1.0 \leq \textit{Fitness}$ y que mientras más aproximado a 1.0 sea el Fitness de la propuesta, más óptima se considera.

En la figura 55 se observa la mejor solución de acero de refuerzo principal que se encuentra con la herramienta de software.

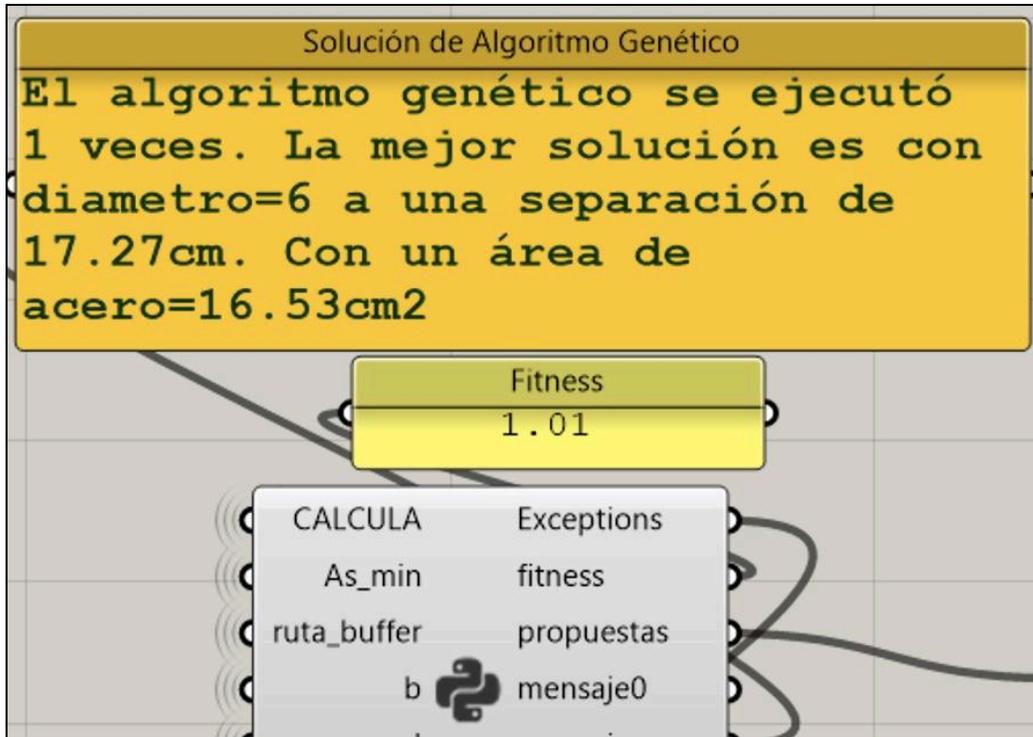


Figura 55. Comparación de resultados: mejor solución de acero de refuerzo principal encontrado por herramienta de software.

De acuerdo con la figura 54, el programa obtiene los siguientes valores para el refuerzo principal del muro pantalla:

- Diámetro de varilla en octavos de pulgada: 6
- Separación entre varillas: 17.27 *cm*
- Área de acero de la solución: 16.53 *cm*²

Tomando en cuenta que el área de acero mínima, obtenida en el punto anterior, es igual a 16.37 *cm*², se obtiene que el Fitness de la solución es:

$$Fitness = \frac{As}{As_{min}} = \frac{16.53cm^2}{16.37cm^2} = 1.0097 \approx 1.01$$

Adicionalmente, el programa proporciona un listado de posibles soluciones para el acero de refuerzo principal, proporcionando el diámetro de varilla, separación entre varillas, área de acero de la solución, cantidad de piezas de varillas y el momento resistente teórico de dicha solución, tal como se muestra en la figura 56.

Tabla de Soluciones							
	Num_var	Separación(cm)	A_S(cm2)	Cantidad	Var(Pzas.)	Mr Teorico	(ton-m)
0	6	17.14	16.53	5.8		32.07	
1	6	17.14	16.53	5.8		32.07	
2	6	16.28	17.39	6.1		33.70	
3	6	14.81	19.38	6.8		37.46	
4	6	13.27	21.38	7.5		41.21	
5	8	22.60	22.30	4.4		42.93	
6	6	12.73	22.52	7.9		43.34	
7	8	17.46	28.88	5.7		55.09	
8	8	16.68	30.40	6.0		57.86	
9	8	15.85	31.92	6.3		60.63	
10	10	23.90	33.25	4.2		63.03	
11	8	13.53	37.50	7.4		70.67	
12	10	20.65	38.00	4.8		71.56	
13	10	19.06	41.17	5.2		77.18	
14	10	17.77	44.34	5.6		82.75	
15	10	16.02	49.09	6.2		90.99	
16	12	22.81	50.16	4.4		92.83	
17	12	22.23	51.30	4.5		94.78	
18	12	21.83	52.44	4.6		96.73	
19	12	19.85	57.00	5.0		104.45	
20	12	15.89	71.83	6.3		128.78	

Figura 56. Comparación de resultados: tabla de posibles soluciones encontradas por herramienta de software.

Por el contrario, el método manual depende de un proceso de prueba y error que ejemplificaré a continuación: El ingeniero de proyecto prueba diferentes valores para diámetro de varilla y separación entre varillas. La hoja de excel realiza los cálculos y le indica al ingeniero si su solución cumple con el área de acero mínima.

En la figura 57 se aprecia un ejercicio donde el ingeniero de proyecto ingresó el diámetro de varilla número 6 con una separación de 20 cm.

Tipo de barra. (Ø).	-----		6	
Separación del acero de refuerzo.	S	cm	20.0	
Diámetro de barra.	Dia	cm	1.905	
Area de barra.	A	cm ²	2.85	
Area de acero de refuerzo.	As	cm ² /m	14.3	FALTA ACERO
Area de acero de refuerzo minima.	Asmin	cm ² /m	16.37	
Fator β1			0.84	
Area de acero de refuerzo maxima.	Asmax	cm ² /m	183.2	OK

Figura 57. Comparación de resultados: propuesta 1 de solución para acero de refuerzo principal con el método manual.

Se observa que la hoja de excel presenta la leyenda “FALTA ACERO” indicando que la propuesta no cumple con el área de acero mínima para el muro. La propuesta tiene un área de acero de 14.3 cm^2 y comparado contra el área de acero mínima igual a 16.37 cm^2 la propuesta no cumple.

Calculando el Fitness de esta solución se obtiene que:

$$Fitness = \frac{As}{As_{min}} = \frac{14.3cm^2}{16.37cm^2} = 0.873 \approx 0.88$$

El Fitness está indicando que la propuesta no es una solución válida dado que no cumple con la regla $1.0 \leq Fitness$ ya que el área de acero es menor al área de acero mínima.

En otro ejercicio donde el ingeniero de proyecto propone una solución distinta, utilizando un diámetro de varilla número 6 y una separación de 13 cm, la hoja de excel entrega los resultados mostrados en la figura 58.

Tipo de barra. (Ø).	-----		6	
Separacion del acero de refuerzo.	S	cm	13.0	
Diametro de barra.	Dia	cm	1.905	
Area de barra.	A	cm ²	2.85	
Area de acero de refuerzo.	As	cm ² /m	21.9	OK
Area de acero de refuerzo minima.	Asmin	cm ² /m	16.37	
Fator β1			0.84	
Area de acero de refuerzo maxima.	Asmax	cm ² /m	183.2	OK

Figura 58. Comparación de resultados: propuesta 2 de solución para acero de refuerzo principal con el método manual.

Analizando los resultados de la hoja de excel se observa que la solución propuesta cumple con el área de acero mínima y el área de acero máxima, indicando que se tiene un área de acero igual 21.9 cm^2 .

Calculando el Fitness de esta solución obtenemos que:

$$Fitness = \frac{As}{As_{min}} = \frac{21.19cm^2}{16.37cm^2} = 1.294 \approx 1.3$$

Si bien la propuesta cumple con el área de acero mínima y máxima, el valor $Fitness=1.3$ indica que el área de acero de la propuesta es 1.3 veces el área de acero mínima, lo que representa una solución sobreestimada. Se puede mejorar el resultado haciendo que el área de acero se aproxime al área de acero mínima, modificando el diámetro de varilla o la separación entre varillas.

Después de realizar varios ejercicios, el ingeniero de proyecto llegó a la solución mostrada en la figura 59.

Tipo de barra. (Ø).	-----		6	
Separacion del acero de refuerzo.	S	cm	17.0	
Diametro de barra.	Dia	cm	1.905	
Area de barra.	A	cm ²	2.85	
Area de acero de refuerzo.	As	cm ² /m	16.8	OK
Area de acero de refuerzo minima.	Asmin	cm ² /m	16.37	
Fator β1			0.84	
Area de acero de refuerzo maxima.	Asmax	cm ² /m	183.2	OK

Figura 59. Comparación de resultados: propuesta 3 de solución para acero de refuerzo principal con el método manual.

De la propuesta anterior se observa se obtiene un área de acero igual a 16.8 cm^2 y calculado el Fitness de esta propuesta se obtiene lo siguiente:

$$Fitness = \frac{As}{As_{min}} = \frac{16.8 \text{ cm}^2}{16.37 \text{ cm}^2} = 1.0262 \approx 1.03$$

Finalmente, en una comparación con la propuesta de solución para el acero de refuerzo principal del muro pantalla obtenida con la herramienta de software se obtuvieron los siguientes valores:

- Diámetro de varilla en octavos de pulgada: 6
- Separación entre varillas: 17.27 cm
- Área de acero de la solución: 16.53 cm^2
- Fitness: 1.01

Y después de varios ejercicios, la última solución obtenida con el método manual se obtuvieron los siguientes resultados:

- Diámetro de varilla en octavos de pulgada: 6
- Separación entre varillas: 17 cm
- Área de acero de la solución: 16.8 cm^2
- Fitness: 1.03

Cotejando los valores Fitness de ambas soluciones se puede observar que el Fitness de la propuesta de la herramienta de software (1.01) es más eficiente y cercano a 1.0 en comparación con el fitness de la propuesta del ingeniero de proyecto (1.03).

$$1.0 < 1.01 < 1.03$$

En conclusión, ambas soluciones son adecuadas y se encuentran dentro de lo que el área de ingeniería considera óptimo. No obstante, es importante recalcar que la herramienta de software requiere de nula intervención por parte del ingeniero de proyecto para buscar y proponer una solución optimizada para el acero de

refuerzo principal. El programa tiende a entregar soluciones optimizadas en pocos segundos con la posibilidad de ejecutar el programa N veces para obtener diferentes resultados con tendencias a la mejor solución a diferencia del método manual que requiere de un ingeniero experimentado y varias horas de trabajo.

5.2.3. Cálculo de momento resistente

El siguiente valor a comparar entre ambas metodologías es el momento resistente. El momento resistente es una magnitud que indica la resistencia que tiene el muro frente a momentos flexionantes antes de deformarse y romperse. Es la suma de la resistencia del acero de refuerzo principal, mismo que proviene de la propuesta del apartado 5.2.2, y la resistencia del concreto. Si se presentan fuerzas que excedan el momento resistente, el muro comenzará a agrietarse, y en el peor de los casos, a romperse.

La herramienta de software utiliza la misma metodología que usa el método manual para calcular el momento resistente del muro, a partir de sus características y la solución de acero de refuerzo principal. Procedimiento que se puede encontrar explicado detalladamente en las Normas técnicas complementarias para diseño y construcción de estructuras de concreto. (referencia [22] página 411).

En la figura 60 se observa el resultado que calcula la herramienta de software en función de la propuesta de acero de refuerzo que encuentra el algoritmo genético.

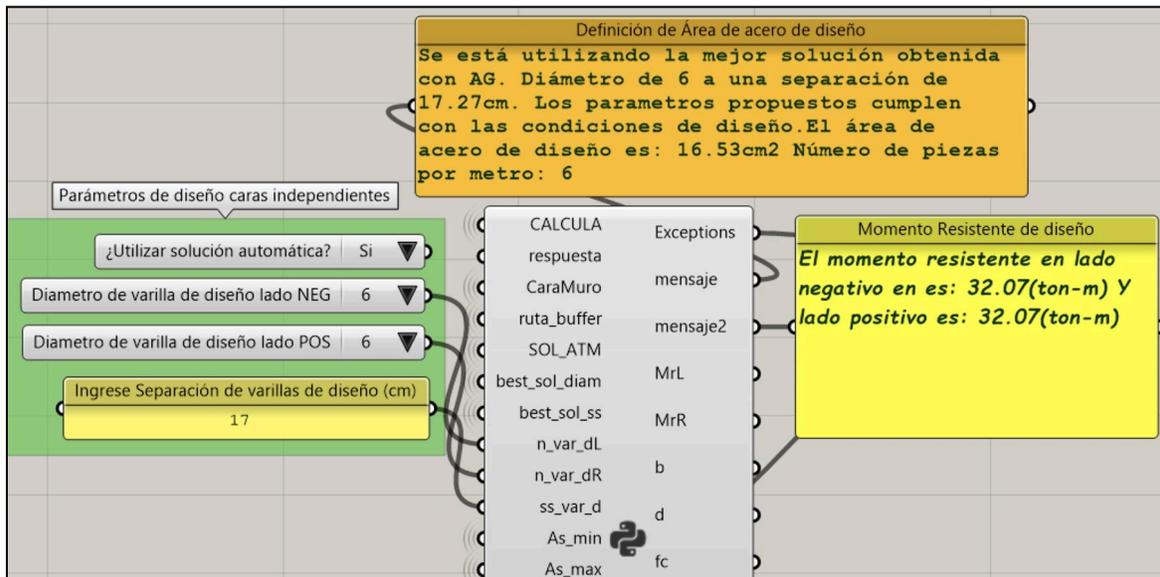


Figura 60. Comparación de resultados: cálculo de momento resistente por herramienta de software.

Es importante resaltar que en la figura anterior se aprecian 3 elementos relevantes. Comenzando por el lado izquierdo, se muestra un Input para este apartado del programa. El colapsable con la pregunta “¿Utilizar solución automática?” permite elegir entre la solución automática propuesta por el algoritmo genético o una solución propuesta por el ingeniero de proyecto, ya sea tomando como referencia la tabla de propuestas válidas entregadas por el algoritmo genético (figura 56) o alguna otra que pudiese proponer el usuario. La solución “manual” es evaluada por el programa, valida que cumpla con todas las condiciones de diseño necesarias para el muro, de lo contrario notifica con un mensaje de error.

El segundo elemento, el recuadro de color naranja en la parte superior, indica la solución que se está utilizando para el cálculo del momento resistente. En este caso indica que se está obteniendo la mejor propuesta de acero de refuerzo principal encontrada por el algoritmo genético. Adicionalmente proporciona la información del diámetro de varilla, la separación entre varillas, el área de acero de la solución en cuestión y el número de piezas de varillas de la solución.

El tercer elemento, el cuadro de texto color amarillo del lado derecho, indica el resultado del momento resistente para ambas caras del muro. Recordando que se puede optar por diámetros de varilla diferentes para ambas caras del muro.

El resultado obtenido con la herramienta para el momento resistente es:

$$M_r = 32.07 \text{ tm}$$

Por otro lado, mediante el método manual la hoja de excel proporciona el cálculo del momento resistente en función de la propuesta final de acero de refuerzo principal a la que llegó el ingeniero de proyecto.

En la figura 61 se observa el resultado del momento resistente obtenido mediante el método manual.

Tipo de barra. (Ø).	-----		6	
Separacion del acero de refuerzo.	S	cm	17.0	
Diametro de barra.	Dia	cm	1.905	
Area de barra.	A	cm ²	2.85	
Area de acero de refuerzo.	As	cm ² /m	16.8	OK
Area de acero de refuerzo minima.	Asmin	cm ² /m	16.37	
Fator β1			0.84	
Area de acero de refuerzo maxima.	Asmax	cm ² /m	183.2	OK
Alto efectivo seccion de analisis.	d	cm	52.5	
Procentaje de acero de refuerzo.	ρ	%	0.0031935	
Variable.	w - q		0.045085	
MOMENTO NOMINAL		Mn	Kg cm	3252220
			Ton m	32.52

Figura 61. Comparación de resultados: cálculo de momento resistente con el método manual.

De acuerdo con lo anterior, se puede apreciar la propuesta de acero de refuerzo principal que como consecuencia produce un momento resistente, valor encerrado con el rectángulo rojo, igual a:

$$M_r = 32.52 \text{ tm}$$

Analizando el resultado obtenido con la herramienta de software, $M_r = 32.07 \text{ tm}$, y el resultado obtenido mediante el método manual $M_r = 32.52 \text{ tm}$, se observa una pequeña diferencia entre los valores. Esto se debe a que el valor de momento resistente depende de la solución de acero de refuerzo principal. Si bien ambas metodologías utilizan propuestas de acero de refuerzo principal muy similares, no son las mismas, por esa razón los valores del momento resistente difieren en unos cuantos decimales.

Como comprobación final, con la herramienta de software se ingresó la solución de acero de refuerzo principal que el ingeniero de proyecto determinó como la mejor en el método manual, tratándose de un diámetro de varilla de 6 a una separación de 17cm.

En la figura 62 se observa el resultado que se obtiene con el programa.

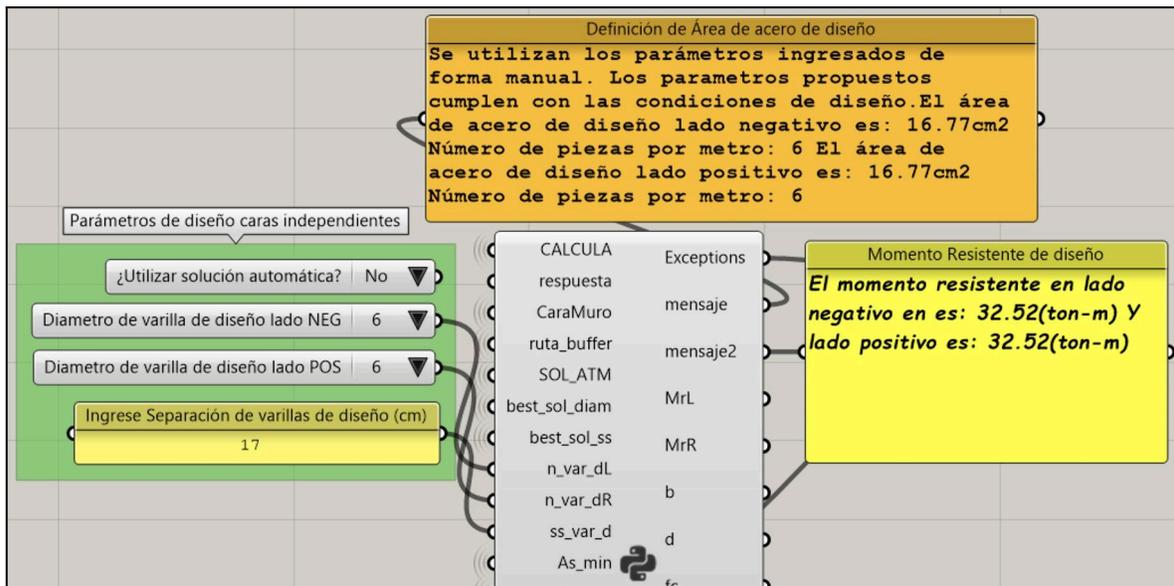


Figura 62. Comparación de resultados: cálculo de momento resistente por herramienta de software (misma propuesta de acero de refuerzo principal que el método manual).

En el recuadro color amarillo se observa que el momento resistente $M_r = 32.52 \text{ tm}$ es igual al momento resistente obtenido con el método manual.

5.2.4. Búsqueda de bastones como acero de refuerzo adicional para momentos flexionantes.

Diferentes condiciones, tipo de suelo, geometría del terreno, entre otras, provocan que los muros pantalla estén sometidos a empujes y momentos. La presencia de estas fuerzas pueden sobrepasar la resistencia del muro a la flexión considerando como límite al momento resistente. La resistencia del muro a la flexión varía en función de su solución de acero de refuerzo principal y la resistencia del concreto.

Como ya se ha hablado en capítulos anteriores, parte de la solución del diseño de muros pantalla es el análisis de los elementos mecánicos para determinar si se vuelve necesario colocar acero de refuerzo adicional, en este caso para momentos en servicio. Este refuerzo adicional se conoce como bastones.

Esta solución consiste en graficar los datos de elementos mecánicos y el momento resistente para visualizar en qué nivel o profundidad del muro se requiere colocar bastones para que el muro soporte los momentos en servicio.

Para el muro que se analiza en este capítulo, la herramienta de software proporciona una serie de 3 gráficas de elementos mecánicos correspondientes a los momentos en servicio, tal como se muestra en la figura 63.

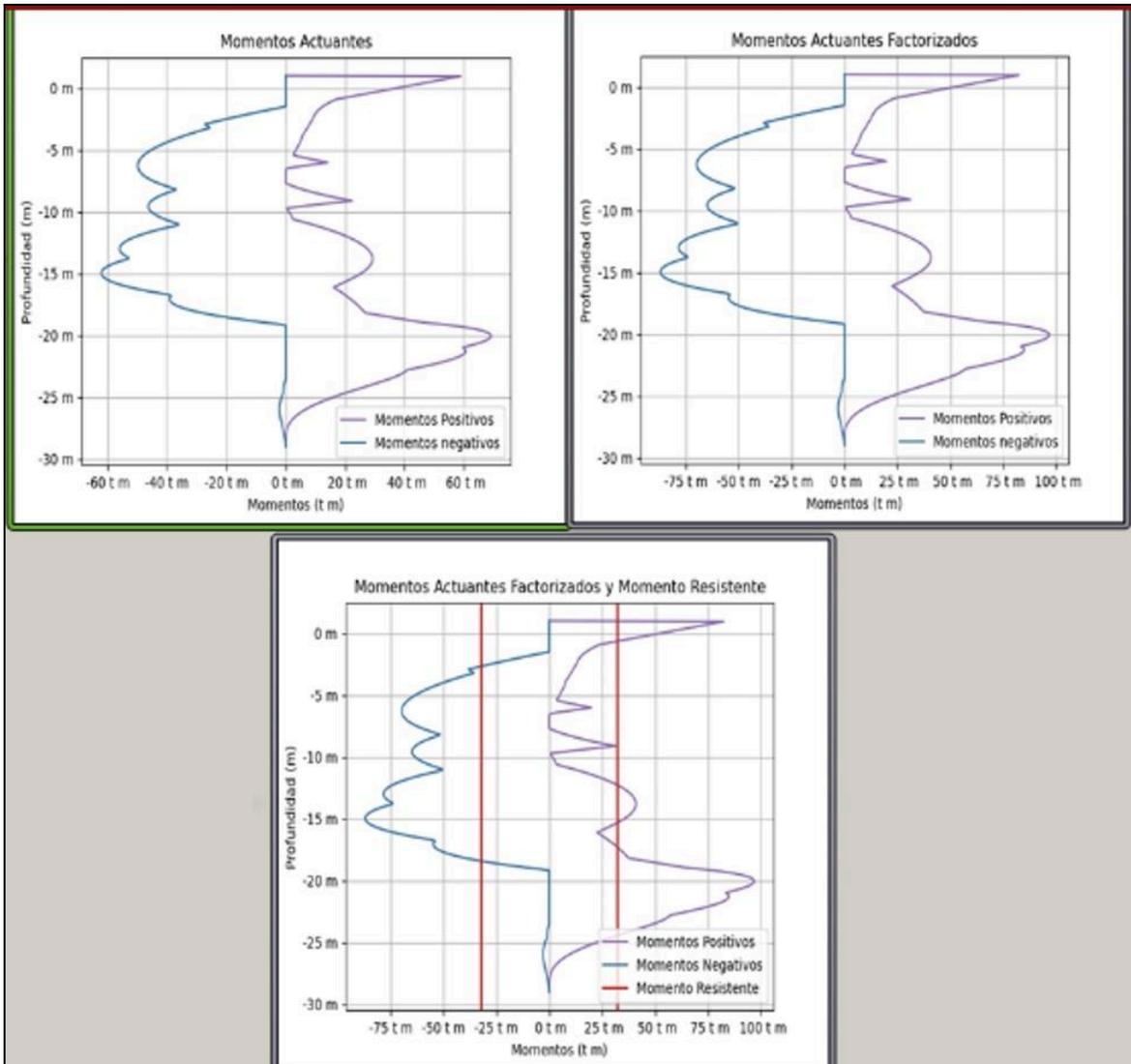


Figura 63. Comparación de resultados: gráficas de elementos mecánicos para momentos en servicio generadas por herramienta de software.

La primera gráfica, esquina superior izquierda, representa los datos de momentos en servicio tal cual son proporcionados a la herramienta de software.

La segunda gráfica, esquina superior derecha, corresponde a la gráfica de los momentos en servicios factorizados. Esto significa que de acuerdo con las normas técnicas complementarias para diseño y construcción de cimentaciones y normas técnicas complementarias para estructuras de concreto (Referencias [11] y [12]), se debe diseñar el muro multiplicando, por un factor llamado F_c , los datos de los momentos y fuerzas en servicio como medida de seguridad.

Finalmente, la tercera gráfica contiene los datos de momentos en servicio factorizados y el momento resistente en ambas caras del muro (lado negativo y

positivo de la gráfica). Sobre esta última se realiza el análisis para refuerzo de acero adicional para momentos flexionantes.

Respecto al método manual, el ingeniero de proyecto debe generar sus gráficas de momentos en servicio a través de excel, dando como resultado el mismo tipo de gráfica.

Posteriormente, la herramienta de software, a través del algoritmo que implementé, proporciona la solución final del acero de refuerzo adicional para cubrir los momentos que sobrepasan el momento resistente del muro. En las figuras 64 se observa lo que sería la solución completa, con colocación de bastones en las regiones del muro donde se requieren.

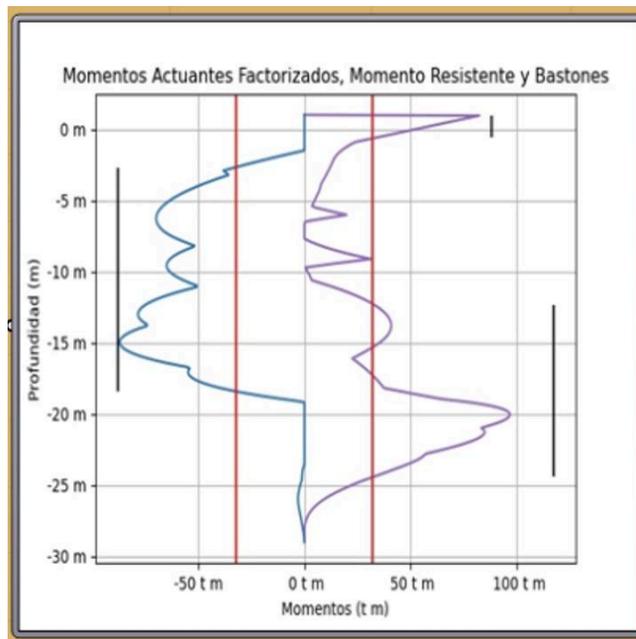


Tabla Resumen Bastones Momentos Negativos																	
0	Inicio Bast. en Z (m)	0	Final Bast. en Z (m)	0	Longitud Bastón (m)	0	Momento Pico (t m)	0	# de Var.	0	No. Bastones (Pzas.)	0	Sep. Bastones (cm)	0	Momento Res. Bastón (t m)	0	Mr Ref. Prin. + Mr Bastón (t m)
1	-2.7	1	-18.42	1	15.72	1	-86.91	1	8	1	6	1	17.27	1	55.93	1	88.0

Tabla Resumen Bastones - Momentos Positivos																	
0	Inicio Bast. en Z (m)	0	Final Bast. en Z (m)	0	Longitud Bastón (m)	0	Momento Pico (t m)	0	# de Var.	0	No. Bastones (Pzas.)	0	Sep. Bastones (cm)	0	Momento Res. Bastón (t m)	0	Mr Ref. Prin. + Mr Bastón (t m)
1	0.95	1	-0.6	1	1.55	1	82.33	1	8	1	6	1	17.27	1	55.93	1	88.0
2	-12.32	2	-24.4	2	12.08	2	96.73	2	10	2	6	2	17.27	2	85.37	2	117.44

Figura 64. Comparación de resultados: solución y colocación de bastones realizado por la herramienta de software.

Se puede observar en la gráfica anterior que la herramienta de software coloca los bastones exactamente donde hay presencia de momentos ya sea negativos o positivos, que excedan el momento resistente del muro.

La dos tablas de color azul proporcionan información importante sobre los bastones que la herramienta de software calculó y colocó en la gráfica.

- Las columnas con los nombres de *“Inicio Bast. en Z (m)”* y *“Final Bast. en Z (m)”* indican desde qué profundidad comienza el bastón (Inicio Bast. en Z (m)) y hasta donde termina el bastón (Final Bast. en Z (m)), recordando que 0 metros es el inicio del muro al nivel de la superficie del suelo y comienza a aumentar en profundidad hacia el subsuelo.
- *“Longitud Bastón (m)”* indican la longitud total del bastón que la herramienta de software colocó, esto es una diferencia absoluta entre *“Final Bast. en Z (m)”* y *“Inicio Bast. en Z (m)”*.
- *“Momento Pico (t m)”* indican el momento flexionante más grande que existe dentro de la zona cubierta por el bastón. Tomando como referencia el primer y único registro de la tabla de resumen de bastones para momentos negativos (primera tabla color azul de la figura 64), indica que entre la profundidad de 2.7m hasta 18.42m, el momento flexionante más grande presente en ese intervalo de profundidad es de 86.91 (t m).
- *“# de Var.”* proporcionan el diámetro de varilla que la herramienta de software asignó al bastón.
- *“No. Bastones (Pzas.)”* indican la cantidad de unidades de bastones que se requieren para cubrir el momento pico de ese intervalo de momentos flexionantes que rebasan el momento resistente del muro.
- *“Sep. Bastones (cm)”* muestran la separación que debe haber entre la cantidad de unidades de bastones obtenidas en el cálculo. En el caso de momentos flexionantes, el refuerzo de acero adicional (bastones) debe mantener la misma separación que la propuesta solución de acero de refuerzo principal, por lo que la herramienta se enfoca en buscar únicamente el diámetro de varilla que cumpla con las condiciones de diseño, sin necesidad de utilizar algoritmo genético.
- *“Momento Res. Bastón (t m)”* indican la resistencia a la flexión del arreglo de bastón por sí solos. Esto es dejando fuera la resistencia del muro a la flexión (momento resistente).
- *“Mr Ref. Prin. + Mr Bastón (t m)”* proporcionan la suma de la resistencia de los bastones más el momento resistente del muro, que como resultado, debe ser estrictamente mayor al momento pico que cubre el arreglo de bastones.

Por otro lado, el método manual requiere que el ingeniero de proyecto genere las gráficas de momentos flexionantes. El ingeniero de proyecto debe importar los datos de elementos mecánicos a su hoja de excel para poder generar una gráfica similar a la que proporciona la herramienta de software descrita en este trabajo.

Por simplicidad, el ingeniero de proyecto se encarga de generar la gráfica de momentos flexionantes factorizados con el momento resistente del muro pantalla. Esta gráfica sería la equivalente a la tercera gráfica con líneas rojas verticales que se observa en la figura 63.

Para fines prácticos, se utilizó la gráfica generada por la herramienta de software, misma que contiene los datos de elementos mecánicos factorizados y el momento resistente del muro pantalla. El ingeniero de proyecto que me apoyó con esta comparación de resultados, colocó su solución para este muro pantalla, misma que se muestra en la figura 65.

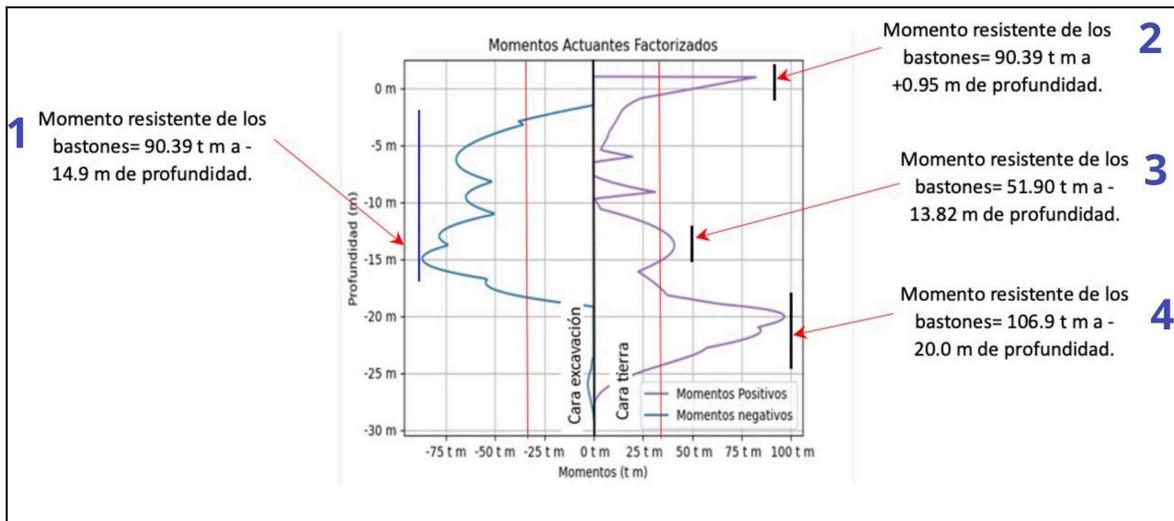


Figura 65. Comparación de resultados: solución final de bastones con el método manual.

A continuación, se detallan los cálculos que realizó el ingeniero de proyecto a través de su hoja de excel, recordando que para cada bastón, el ingeniero tuvo que realizar distintos ejercicios hasta encontrar el diámetro que cumpla con las condiciones de diseño, evitando colocar bastones con resistencia más allá de lo necesario.

- Arreglo de bastones número 1: El ingeniero identificó que el momento máximo a cubrir por este arreglo de bastones es de 86.87 t m. Después de haber calculado el momento resistente del arreglo de bastones con diferentes diámetros, el ingeniero de proyecto propone la solución que se observa en la figura 66.

Momento adicional Bastones.				
Numero de barras de refuerzo. (Pzas)	No		6	
Tipo de barra. (\emptyset).	-----		8	
Separacion del acero de refuerzo.	S	cm	17.00	
Diametro de barra.	Día	cm	2.54	
Area de barra.	A	cm ²	5.07	
Area de acero de refuerzo.	As	cm ² /m	30.4024	
Procentaje de acero de refuerzo.	p	%	0.0057909	
Variable.	w		0.0817545	
MOMENTO NOMINAL	Mn	Kg cm	5786739	
		Ton m	57.87	
MOMENTO NOMINAL= As Ref. long + Bastones=		Mn	Ton m	90.39

Figura 66. Comparación de resultados: solución final del arreglo de bastones 1 calculado con el método manual.

Explicando la información mostrada en la figura 66, el rectángulo en color rojo señala la propuesta de solución para el arreglo de bastones número 1 (ver figura 65) el cual consiste en 6 bastones con diámetro de varilla del número 8 a una separación de 17 cm.

Lo anterior da como resultado las resistencias señaladas por el rectángulo de color azul; Lo que se lee como momento nominal corresponde únicamente al momento resistente del arreglo de bastones. Dicha magnitud se compara con la columna “*Momento Res. Bastón (t m)*” de la tabla de soluciones que proporciona la herramienta de software.

La resistencia del arreglo de bastones de esta solución es igual a 57.87 t m y tomando en cuenta el momento resistente del muro pantalla, la resistencia total es igual a 90.39 t m. Esta magnitud es equivalente al dato proporcionado por la herramienta de software en la columna “*Mr Ref. Prin. + Mr Bastón (t m)*” de la tabla de soluciones de arreglo de bastones.

- Arreglo de bastones número 2: Para este caso, el ingeniero de proyecto identificó el momento máximo de 82.32 t m. Realizó una comparación con el arreglo de bastones 1, es un momento de magnitud similar y el ingeniero optó por utilizar la misma solución de arreglo de bastones (ver figura 66).

Para este caso particular, el ingeniero de proyecto ahorró tiempo al no tener que buscar y calcular otra solución para este arreglo de bastones.

- Arreglo de bastones número 3: Este arreglo busca cubrir un momento máximo de 40.88 t m. Después de realizar un búsqueda iterativa, el ingeniero de proyecto propone la solución mostrada en la figura 67.

Momento adicional Bastones.					
Numero de barras de refuerzo. (Pzas)	No		5		
Tipo de barra. (Ø).	-----		5		
Separacion del acero de refuerzo.	S	cm	17.00		
Diametro de barra.	Dia	cm	1.59		
Area de barra.	A	cm ²	1.98		
Area de acero de refuerzo.	As	cm ² /m	9.8966		
Procentaje de acero de refuerzo.	p	%	0.0018851		
Variable.	w		0.0266128		
MOMENTO NOMINAL	Mn	Kg cm	1937853		
		Ton m	19.38		
MOMENTO NOMINAL= As Ref. long + Bastones=		Mn	Ton m	51.90	OK

Figura 67. Comparación de resultados: solución final del arreglo de bastones 3 calculado con el método manual.

La solución está dada por un arreglo de bastones de diámetro 5, separación de 17cm y 5 piezas. Esta solución representa un momento resistente de 19.38 t m y sumada con el momento resistente del muro pantalla, se obtiene una resistencia a la flexión total de 51.90 t m.

- Arreglo de bastones número 4: Este arreglo de bastones debe cubrir un momento en servicio de 106.9 t m. Para esto, el ingeniero de proyecto planteó la solución mostrada en la figura 68.

Momento adicional Bastones.					
Numero de barras de refuerzo. (Pzas)	No		5		
Tipo de barra. (Ø).	-----		10		
Separacion del acero de refuerzo.	S	cm	17.00		
Diametro de barra.	Dia	cm	3.18		
Area de barra.	A	cm ²	7.92		
Area de acero de refuerzo.	As	cm ² /m	39.5865		
Procentaje de acero de refuerzo.	p	%	0.0075403		
Variable.	w		0.1064512		
MOMENTO NOMINAL	Mn	Kg cm	7437808		
		Ton m	74.38		
MOMENTO NOMINAL= As Ref. long + Bastones=		Mn	Ton m	106.90	OK

Figura 68. Comparación de resultados: solución final del arreglo de bastones 4 calculado con el método manual.

En este último arreglo de bastones, el ingeniero de proyecto propone utilizar 5 bastones de diámetro 10 a una separación de 17 cm. Dando como resultado un momento resistente para el arreglo de acero de 74.38 t m, que sumada con el momento resistente del muro pantalla se obtiene una resistencia a la flexión total de 106.90 t m.

En la figura 69 se muestra un resumen de la solución de acero de refuerzo adicional para momentos flexionantes obtenida con el método manual.

ACERO DE REFUERZO ADICIONAL - BASTONES				
CARA EXCAVACIÓN				
Profundidad, (m)	#piezas	Diámetro del bastón	Separación (cm)	M _R (t m)
-14.9	6	8	17	90.39

ACERO DE REFUERZO ADICIONAL - BASTONES				
CARA TIERRA				
Profundidad, (m)	#piezas	Diámetro del bastón	Separación (cm)	M _R (t m)
+0.95	6	8	17	90.39
-13.82	5	5	17	51.90
-20.00	5	10	17	106.90

Figura 69. Comparación de resultados: resumen de arreglos de bastones obtenidos con el método manual.

Con el objetivo de comparar los resultados, en la figura 70 anexo el resumen de la solución de acero de refuerzo adicional para momentos flexionantes obtenida con la herramienta de software.

Tabla Resumen Bastones Momentos Negativos																	
0	Inicio Bast. en Z (m)	0	Final Bast. en Z (m)	0	Longitud Bastón (m)	0	Momento Pico (t m)	0	# de Var.	0	No. Bastones (Pzas.)	0	Sep. Bastones (cm)	0	Momento Res. Bastón (t m)	0	Mr Ref. Prin. + Mr Bastón (t m)
1	-2.7	1	-18.42	1	15.72	1	-86.91	1	8	1	6	1	17.27	1	55.93	1	88.0

Tabla Resumen Bastones - Momentos Positivos																	
0	Inicio Bast. en Z (m)	0	Final Bast. en Z (m)	0	Longitud Bastón (m)	0	Momento Pico (t m)	0	# de Var.	0	No. Bastones (Pzas.)	0	Sep. Bastones (cm)	0	Momento Res. Bastón (t m)	0	Mr Ref. Prin. + Mr Bastón (t m)
1	0.95	1	-0.6	1	1.55	1	82.33	1	8	1	6	1	17.27	1	55.93	1	88.0
2	-12.32	2	-24.4	2	12.08	2	96.73	2	10	2	6	2	17.27	2	85.37	2	117.44

Figura 70. Comparación de resultados: tablas resumen de solución para arreglos de bastones obtenidos por la herramienta de software.

Las diferencias entre los resultados se deben a que la herramienta de software realiza los cálculos sin redondear los valores intermedios que se utilizan para el

cálculo de áreas de acero y momentos resistentes. A diferencia del método manual que tiende a redondear a números enteros los valores intermedios en el cálculo de áreas de acero y momentos resistentes.

La herramienta de software proporciona información adicional de las soluciones respecto al método manual, misma que se vuelve de gran utilidad para el ingeniero de proyecto y permite entender de mejor manera la solución en cada caso.

Si bien se observan soluciones similares entre ambas metodologías, es importante recalcar que la herramienta de software trabaja a partir de un algoritmo que construí para automatizar esta tarea con el objetivo de identificar las zonas de muro que requieren acero de refuerzo adicional y posteriormente tomar en cuenta todas las condiciones de diseño necesarias para generar la propuesta de acero de refuerzo adicional para momentos flexionantes. Remarcando que este proceso se ejecuta en cuestión de segundos.

Comparado con el método manual donde se requiere de mucha intervención e inversión de tiempo por parte del ingeniero de proyecto. Se vuelve evidente que el método manual es más complejo, requiere mucho criterio para ingresar datos a la hoja de excel y generar soluciones; criterios y condiciones que se encuentran implícitos en la lógica y programación de la herramienta de software.

5.2.5. Cálculo de la resistencia al corte: VCR

En este apartado se calcula el parámetro conocido como VCR, tratándose de la magnitud que representa la resistencia al corte del acero de refuerzo principal sumada con la resistencia del concreto. Es importante mencionar que tanto la herramienta de software como el método manual utilizan la misma metodología de cálculo de esta magnitud, misma que se puede encontrar en las Normas técnicas complementarias para diseño y construcción de estructuras de concreto. (referencia [22] página 417).

Dicho esto, se espera obtener exactamente el mismo resultado en ambos casos. En la figura 71 se observa el resultado obtenido con la herramienta de software mientras que en la figura 72 se observa el resultado que obtuvo el ingeniero de proyecto con el método manual.

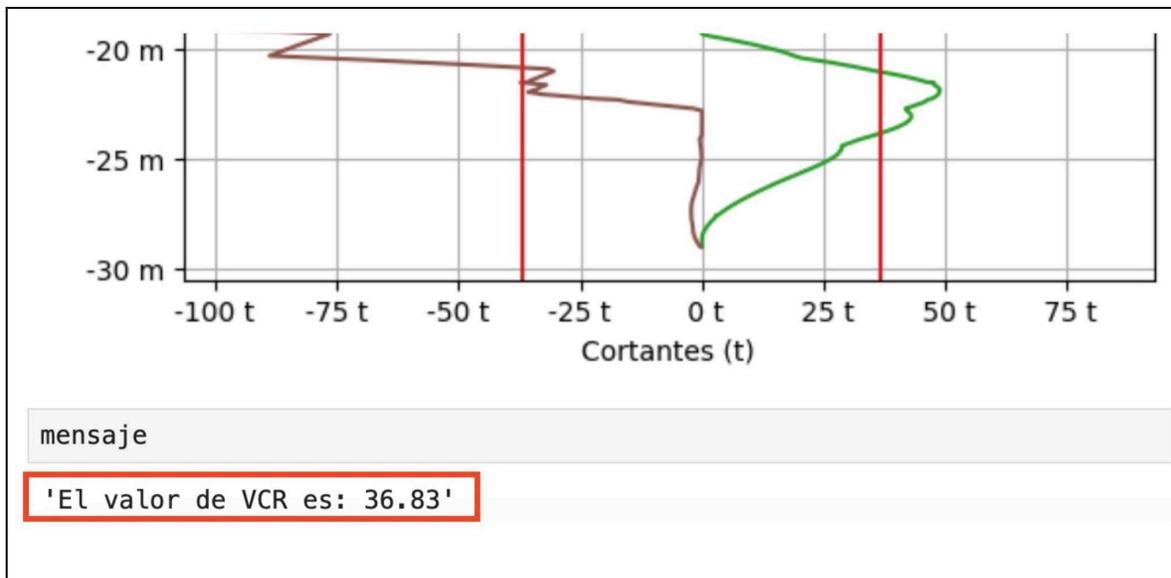


Figura 71. Comparación de resultados: VCR calculado por la herramienta de software.

Diseño a cortante			
Carga Cortante servicio	V	Ton	34.42
Carga cortante Ultima	Vu	Ton	48.19
Factor de resistencia.	FR		0.75
Procentaje de acero de refuerzo.	ρ		0.0032
Resistencia al corte concreto (elemento plano)	Vcr	Ton	36.83

Figura 72. Comparación de resultados: VCR calculado con el método manual.

Como se observa en las figuras anteriores, los valores para VCR son los mismos en ambas metodologías.

5.2.6. Búsqueda de estribos como acero de refuerzo adicional para fuerzas cortantes.

Este apartado corresponde a la última etapa que engloba el cálculo y diseño del acuerdo de refuerzo de muros pantalla. Comenzaré hablando sobre la solución que entrega la herramienta de software, descrita a lo largo de este documento.

La herramienta de software realiza un proceso de generación de gráficas de los datos de elementos mecánicos, correspondientes a las fuerzas cortantes en servicio en las condiciones donde se encontrará el muro pantalla.

El programa, de la misma forma que lo hace en el apartado de bastones para momentos flexionantes, genera 3 gráficas. Estas gráficas se pueden observar en la figura 73 y corresponden al ejercicio del muro pantalla analizado a lo largo del subcapítulo 5.2 de este documento.

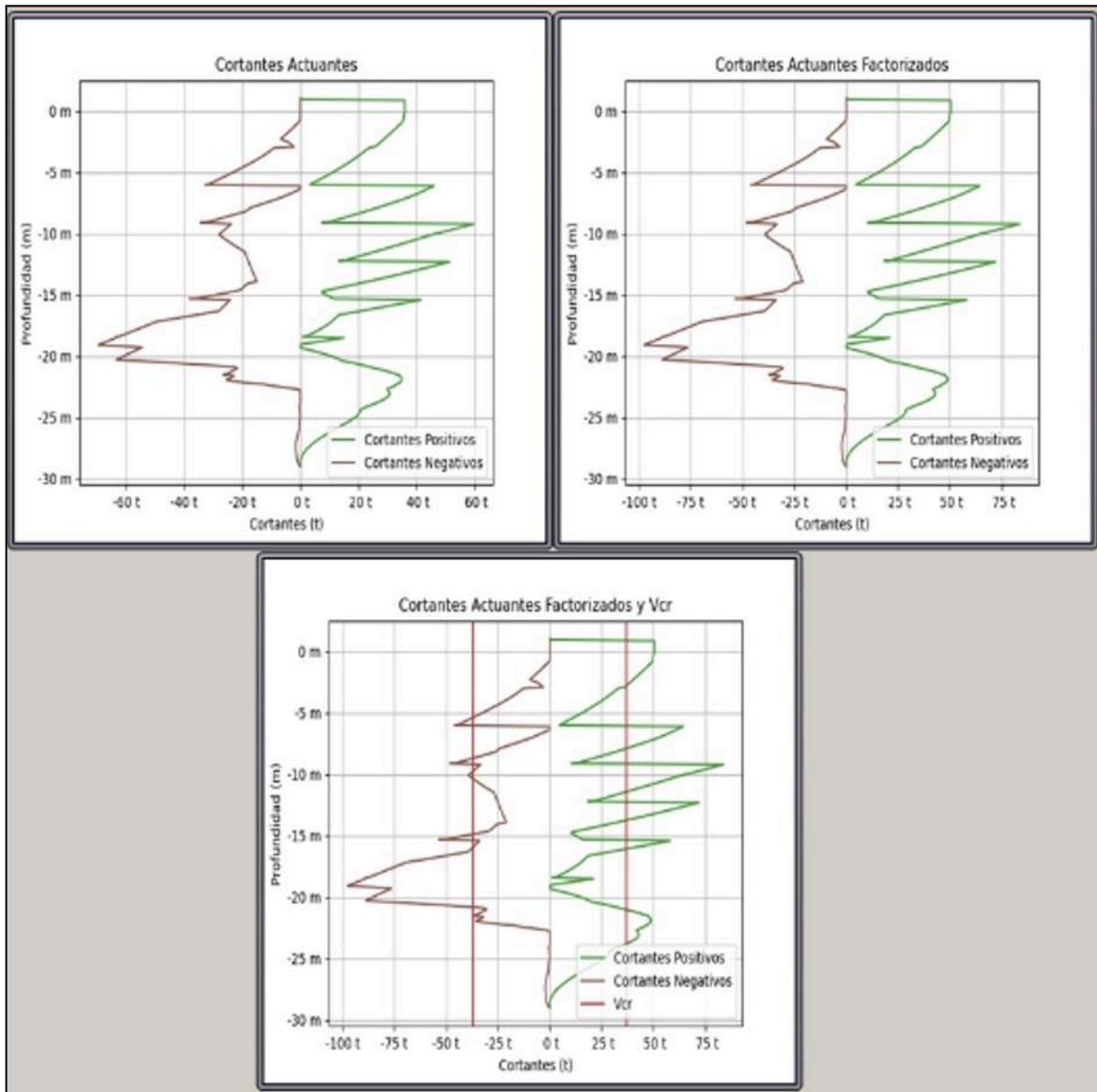


Figura 73. Comparación de resultados: gráficas de fuerzas cortantes generadas por la herramienta de software.

De lo anterior, la primera gráfica, esquina superior izquierda, es la mera representación de los datos de elementos mecánicos correspondientes a las fuerzas cortantes tal cual son proporcionados a la herramienta de software.

La segunda gráfica, esquina superior derecha, representa los datos de los elementos mecánicos correspondientes a fuerzas cortantes, multiplicados por un factor de seguridad denominado “factor de resistencia a cortante (Fr)”.

La tercera gráfica corresponde a los datos de fuerzas cortantes factorizados y adicionalmente se representa con dos líneas rojas verticales el VCR.

Por otra parte, con el método manual el ingeniero de proyecto necesita realizar las gráficas de manera manual a través de su hoja de excel, ordenando los datos, generando una copia de los datos pero multiplicándolos por el factor Fr y construir la gráfica.

Para fines prácticos, el ingeniero de proyecto utilizó la gráfica que contiene los datos de fuerzas cortantes y el VCR, generada por la herramienta de software.

El proceso para calcular y colocar estribos, como acero de refuerzo adicional para fuerzas cortantes, en ambas metodologías sigue el mismo flujo.

En primera instancia se identifican las zonas donde las fuerzas cortantes superan el VCR del muro. Posteriormente estas zonas se agrupan tomando en cuenta distintas condiciones de diseño, se proponen soluciones de arreglos de estribos para cada grupo de regiones de muro donde las fuerzas cortantes superan el VCR del muro y finalmente se realiza un acomodo del arreglo de estribos para que en ambas caras del muro “lado positivo y lado negativo de la gráfica” se visualice un arreglo en espejo de modo que si se doblara la gráfica justo a la mitad de manera vertical, los arreglos de bastones coincidan.

Comenzando con la herramienta de software, el programa realiza de manera automática todo lo descrito en el párrafo anterior. Utiliza un algoritmo genético para la búsqueda de arreglos de estribos donde las variables son el diámetro de varilla de los estribos y la separación entre estos.

El programa agrupa las zonas donde el VCR es rebasado por las fuerzas cortantes y realiza el cálculo de la mejor propuesta para cada zona del muro. En la figura 74 se muestra la gráfica con el arreglo temporal de bastones y se señala el valor del Fitness de cada solución.

El Fitness que utiliza el algoritmo genético está dado por la razón de la resistencia al corte del arreglo de estribos sumada con el VCR del muro entre el cortante máximo dentro del rango de profundidad donde los estribos trabajaran.

$$Fitness = \frac{VCR_{muro} + V_{sr_{estribos}}}{Cortante_{max}} \approx 1.0$$

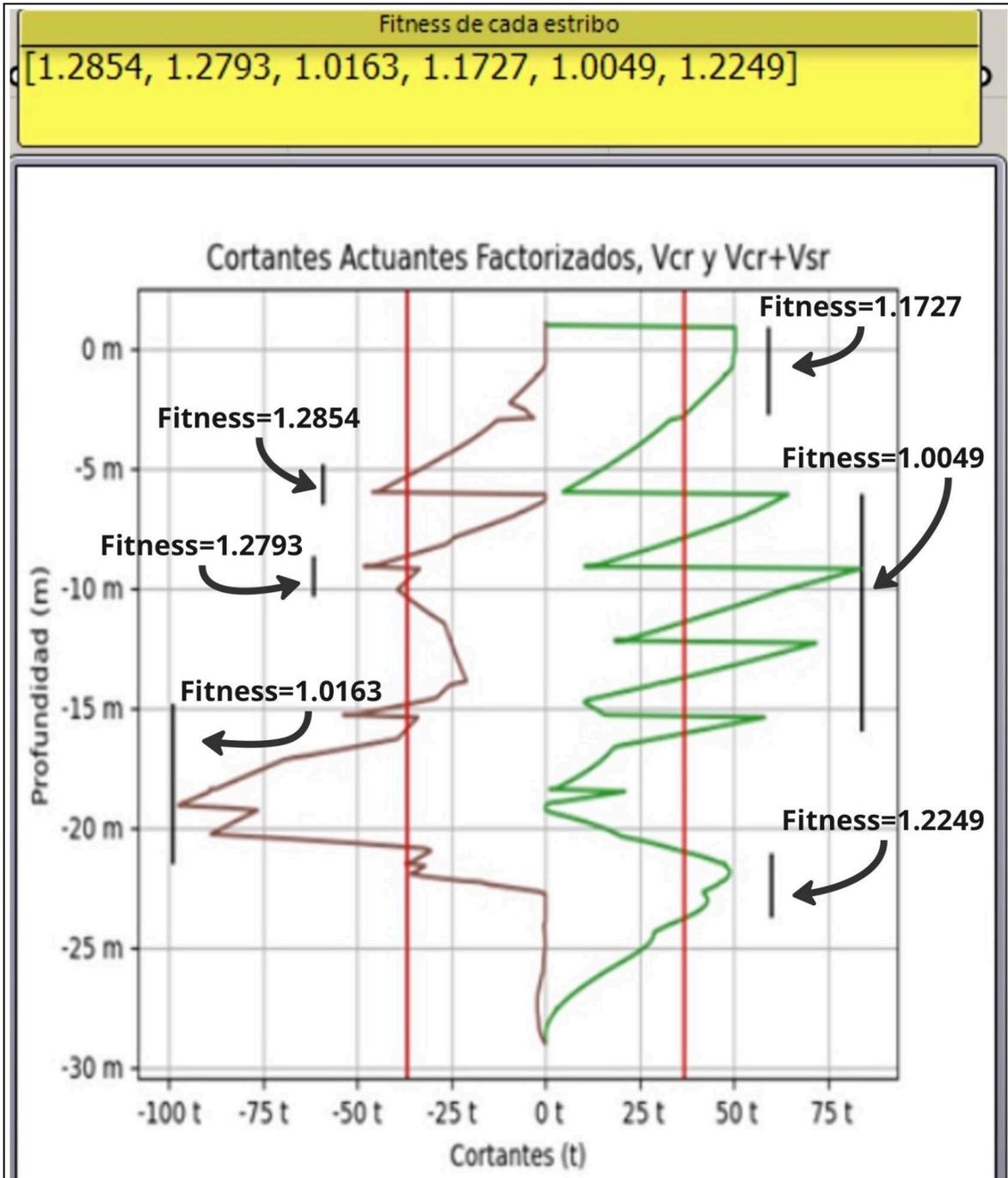


Figura 74. Comparación de resultados: propuesta preliminar de estribos calculados por la herramienta de software.

Como se puede observar en la figura anterior, el programa calcula un total de 6 arreglos de estribos en ambos lados de la gráfica. Los resultados van

acompañados de dos tablas que el programa llena con información de gran utilidad para los ingenieros de proyecto.

Las dos tablas proporcionan la misma información pero de cada lado de la gráfica. En otras palabras, existe una tabla para listar los resultados de los arreglos de estribos encontrados por el programa en el lado negativo de la gráfica y otra tabla para mostrar los resultados del lado positivo. Ver figura 75.

Tabla de Cortantes Negativos											
0	Inicio Z (m)	Final Z (m)	Cortante Máximo (t)	Profundidad Z (m)	A_S min (cm2)	Acero Requerido As (cm2/m)	No. Ramas/m (pzas.)	# de Var.	Separación Óptima (cm)	Res. Acero Est. Vsr (t)	Resistencia Vcr + Vsr (t)
1	-4.9	-6.52	-45.93	-6.02	1.77	1.78	2.5	3	13.26	22.21	59.04
2	-8.7	-10.4	-48.19	-9.12	2.82	3.17	2.5	4	21.11	24.81	61.65
3	-14.9	-21.52	-97.48	-19.1	2.53	7.13	2.5	6	18.93	62.24	99.07

Tabla de Cortantes Positivos											
0	Inicio Z (m)	Final Z (m)	Cortante Máximo (t)	Profundidad Z (m)	A_S min (cm2)	Acero Requerido As (cm2/m)	No. Ramas/m (pzas.)	# de Var.	Separación Óptima (cm)	Res. Acero Est. Vsr (t)	Resistencia Vcr + Vsr (t)
1	0.85	-2.8	50.38	0.85	3.15	3.17	2.5	4	23.54	22.25	59.08
2	-6.12	-16.02	83.77	-9.22	3.33	7.13	2.5	6	24.89	47.34	84.18
3	-21.1	-23.8	48.91	-21.9	1.71	1.78	2.5	3	12.76	23.08	59.91

Figura 75. Comparación de resultados: tabla de estribos calculados por la herramienta de software.

A continuación menciono las columnas que contienen las tablas de resultados de este primer análisis de fuerzas cortantes:

- Las columnas con los nombres de “*Inicio Z (m)*” y “*Final Z (m)*” indican desde qué profundidad comienza el estribo (*Inicio Z (m)*) y hasta dónde termina el estribo (*Final Z (m)*), recordando que 0 metros es el inicio del muro al nivel de la superficie del suelo y comienza a aumentar en profundidad hacia el subsuelo.
- “*Cortante Máximo (t)*” indican la magnitud del cortante más grande dentro del intervalo de “*Inicio Z (m)*” y “*Final Z (m)*”.
- “*Profundidad Z (m)*” proporcionan la ubicación del cortante máximo indicando la profundidad en metros a la que se encuentra “*Cortante Máximo (t)*”.
- “*A_S min (cm2)*” señaladas en color amarillo, indican el área de acero mínimo requerido para cubrir los elementos mecánicos excedentes. La solución de estribos debe tener un área de acero igual o mayor a “*A_S min (cm2)*”.
- “*Acero Requerido As (cm2)*” proporcionan el área de acero resultante de la solución de estribos.
- “*No. Ramas/m (pzas.)*” indican el valor de una constante que está relacionada con la cantidad de jaulas que encierran al arreglo de estribos a lo largo del muro. Se trata de una constante definida por el ingeniero de

proyecto y está basado en temas constructivos. Dicho valor se le puede cambiar y especificar al programa.

- “# de Var.” indican el diámetro en octavos de pulgada de la varilla correspondiente a la solución del arreglo de estribos.
- “Separación Óptima (cm)” indican la separación que debe existir entre los estribos para la solución encontrada por el algoritmo genético.
- “Res. Acero Est. $V_{sr}(t)$ ” proporcionan la resistencia del arreglo de estribos al corte. Esta magnitud indica la resistencia máxima de los estribos ante la presencia de fuerzas cortantes antes de que los estribos fallen.
- “Resistencia $V_{cr} + V_{sr}(t)$ ” indican la resistencia del muro al corte (VCR) sumado con la resistencia del arreglo de estribos al corte. En total la suma de ambas magnitudes deberían ser igual o mayor al “Cortante Máximo (t)”.

Respecto a la solución con el método manual, el ingeniero de proyecto propone una solución similar a la obtenida con la herramienta de software. La solución está dada por 6 arreglos de estribos para ambos lados de la gráfica, mismos que se pueden observar junto con sus Fitness en la figura 76.

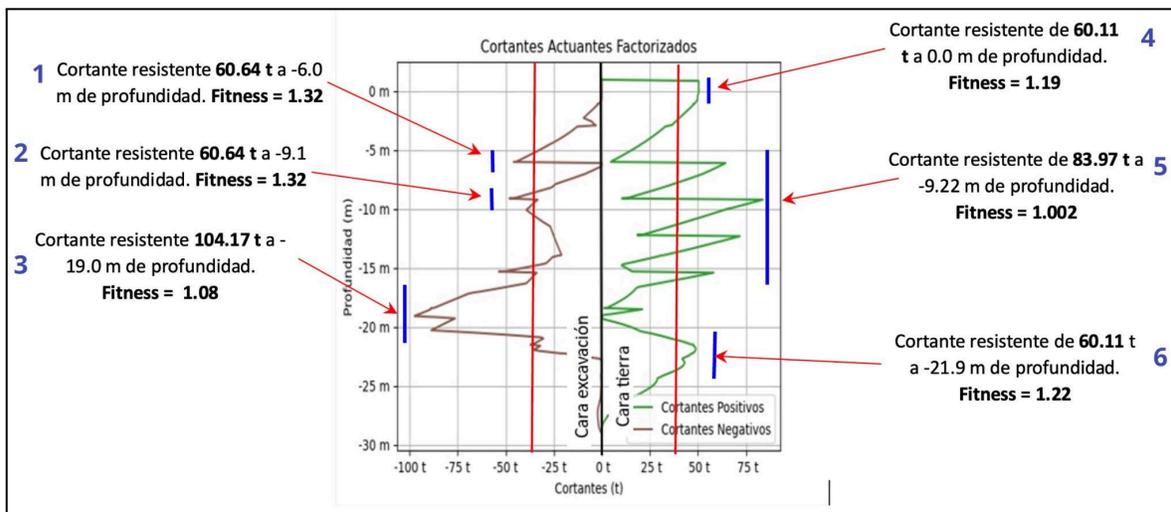


Figura 76. Comparación de resultados: propuesta preliminar de estribos calculados mediante el método manual.

Para cada uno de los 6 arreglos de estribos, el ingeniero tuvo que calcular reiteradas veces en cada solución para encontrar la que mejor considere en función del cortante máximo del intervalo. A continuación se muestran los cálculos finales para cada arreglo de estribos.

- Arreglo de estribos número 1: El ingeniero identificó que el cortante máximo a cubrir por este arreglo de estribos es de 45.93 t. Después de haber calculado la resistencia al corte de cada arreglo de estribos con diferentes

- Arreglo de estribos número 3: Este arreglo busca cubrir un cortante máximo de 96.30 t. Después de realizar un búsqueda iterativa, el ingeniero de proyecto propone la solución mostrada en la figura 78.

Requiere colocar Estribos				
Numero de barras de refuerzo. (Ramas)	No		2.5	
Tipo de barra. (Ø).	-----		6	
Separacion del acero de refuerzo.	S	cm	17.50	
Diametro de barra.	Dia	cm	1.91	
Area de barra.	A	cm ²	2.85	
Area de acero de refuerzo.	As	cm ² /m	7.13	
Area de acero minima	Av_min	cm ²	2.34	OK, CUMPLE CON EL Avmin
Resistencia al corte acero.	Vsr	Ton	67.34	
Resistencia concreto + estribos	Vst	Ton	104.17	OK
Criterio			Cumple	

Figura 78. Comparación de resultados: solución final del arreglo de estribos 3 calculado con el método manual.

La solución está dada por un arreglo de estribos de diámetro 6 y separación de 17.50 cm. Esta solución representa un Vsr de 67.34 t y sumada con el VCR del muro pantalla, se obtiene una resistencia total al corte de 104.17 t.

Esta solución tiene un Fitness=1.08 y comparada con la solución propuesta por la herramienta de software, Fitness=1.0163, se observan resultados muy similares con la diferencia que la solución de la herramienta de software es más cercana a 1.0 por ende, es la solución más optimizada.

- Arreglo de estribos número 4: Este arreglo de estribos debe cubrir un cortante máximo en servicio de 50.38 t. Para esto, el ingeniero de proyecto planteó la solución mostrada en la figura 79.

Requiere colocar Estribos				
Numero de barras de refuerzo. (Ramas)	No		2.5	
Tipo de barra. (Ø).	-----		4	
Separacion del acero de refuerzo.	S	cm	22.50	
Diametro de barra.	Dia	cm	1.27	
Area de barra.	A	cm ²	1.27	
Area de acero de refuerzo.	As	cm ² /m	3.17	
Area de acero minima	Av_min	cm ²	3.01	OK, CUMPLE CON EL Avmin
Resistencia al corte acero.	Vsr	Ton	23.28	
Resistencia concreto + estribos	Vst	Ton	60.11	OK
Criterio			Cumple	

Figura 79. Comparación de resultados: solución final del arreglo de estribos 4 calculado con el método manual.

La solución está dada por un arreglo de estribos de diámetro 4 y separación de 22.50 cm. Esta solución representa un Vsr de 23.28 t y sumada con el VCR del muro pantalla, se obtiene una resistencia total al corte de 60.11 t.

Esta solución tiene un $Fitness=1.19$ y comparada con la solución propuesta por la herramienta de software, $Fitness=1.17$, se observan resultados muy similares con la diferencia que la solución de la herramienta de software es más cercana a 1.0 por ende es la solución más optimizada.

- Arreglo de estribos número 5: Este arreglo de estribos debe cubrir un cortante máximo en servicio de 83.77 t. Para esto, el ingeniero de proyecto planteó la solución mostrada en la figura 80.

Requiere colocar Estribos				
Numero de barras de refuerzo. (Ramas)	No		2.5	
Tipo de barra. (\emptyset).	-----		6	
Separación del acero de refuerzo.	S	cm	25.00	
Diametro de barra.	Dia	cm	1.91	
Area de barra.	A	cm ²	2.85	
Area de acero de refuerzo.	As	cm ² /m	7.13	
Area de acero minima	Av_min	cm ²	3.34	OK, CUMPLE CON EL Avmin
Resistencia al corte acero.	Vsr	Ton	47.14	
Resistencia concreto + estribos	Vst	Ton	83.97	OK
Criterio			Cumple	

Figura 80. Comparación de resultados: solución final del arreglo de estribos 5 calculado con el método manual.

La solución está dada por un arreglo de estribos de diámetro 6 y separación de 25 cm. Esta solución representa un Vsr de 47.14 t y sumada con el VCR del muro pantalla, se obtiene una resistencia total al corte de 83.97 t.

Esta solución tiene un $Fitness=1.002$ y comparada con la solución propuesta por la herramienta de software, $Fitness=1.0049$, se observan resultados muy similares pero en este caso la solución obtenida con el método manual es más cercana a 1.0 por lo que se considera solución optimizada.

- Arreglo de estribos número 6: Este arreglo de estribos debe cubrir un cortante máximo en servicio de 48.91 t. Para esto, el ingeniero de proyecto planteó la solución mostrada en la figura 81.

Requiere colocar Estribos				
Numero de barras de refuerzo. (Ramas)	No		2.5	
Tipo de barra. (\emptyset).	-----		4	
Separación del acero de refuerzo.	S	cm	22.50	
Diametro de barra.	Dia	cm	1.27	
Area de barra.	A	cm ²	1.27	
Area de acero de refuerzo.	As	cm ² /m	3.17	
Area de acero minima	Av_min	cm ²	3.01	OK, CUMPLE CON EL Avmin
Resistencia al corte acero.	Vsr	Ton	23.28	
Resistencia concreto + estribos	Vst	Ton	60.11	OK
Criterio			Cumple	

Figura 81. Comparación de resultados: solución final del arreglo de estribos 6 calculado con el método manual.

En este último arreglo de estribos, el ingeniero de proyecto propone utilizar diámetro 4 a una separación de 22.50 cm. Dando como resultado un V_{sr} para el arreglo de acero de 60.11 t, que sumada con el VCR del muro pantalla se obtiene una resistencia total al corte de 48.91 t.

Esta solución tiene un $Fitness=1.22$ y comparada con la solución propuesta por la herramienta de software, $Fitness=1.2249$, se observan resultados prácticamente iguales en cuanto al rendimiento del $Fitness$, sin embargo a analizar detalladamente, se tratan de propuestas totalmente diferentes en cuanto a diámetro de varilla y separación entre estribos respecta. Mientras que la herramienta de software propone una solución con número de varilla 3 a una separación de 12.76 cm generando una resistencia total al corte de 59.91 t (ver figura 79), el método manual propone utilizar varilla del número 5 a una separación de 25 cm generando una resistencia total al corte de 60.11 t (ver figura 81).

En esta comparación se puede apreciar el excelente rendimiento que tiene la herramienta de software al proponer soluciones optimizadas en cuestión de segundos, en ocasiones ofreciendo soluciones similares o mejores a las que obtendría un ingeniero de proyecto con experiencia utilizando el método manual.

Para finalizar con el apartado del cálculo de acero de refuerzo adicional para fuerzas cortantes, el último paso consiste en un ordenamiento del arreglo de estribos de modo persistan los de mayor resistencia al momento de doblar la gráfica a la mitad de manera vertical. Con este proceso, se eliminan los arreglos de estribos de menor resistencia.

En primera instancia mostraré el resultado que la herramienta de software obtiene al acomodar automáticamente el arreglo de estribos respetando una serie de condiciones de diseño. En este punto entra en juego el algoritmo que diseñé para acomodar los arreglos de estribos, después de haberlos obtenido a través del algoritmo genético. Ver figura 82.

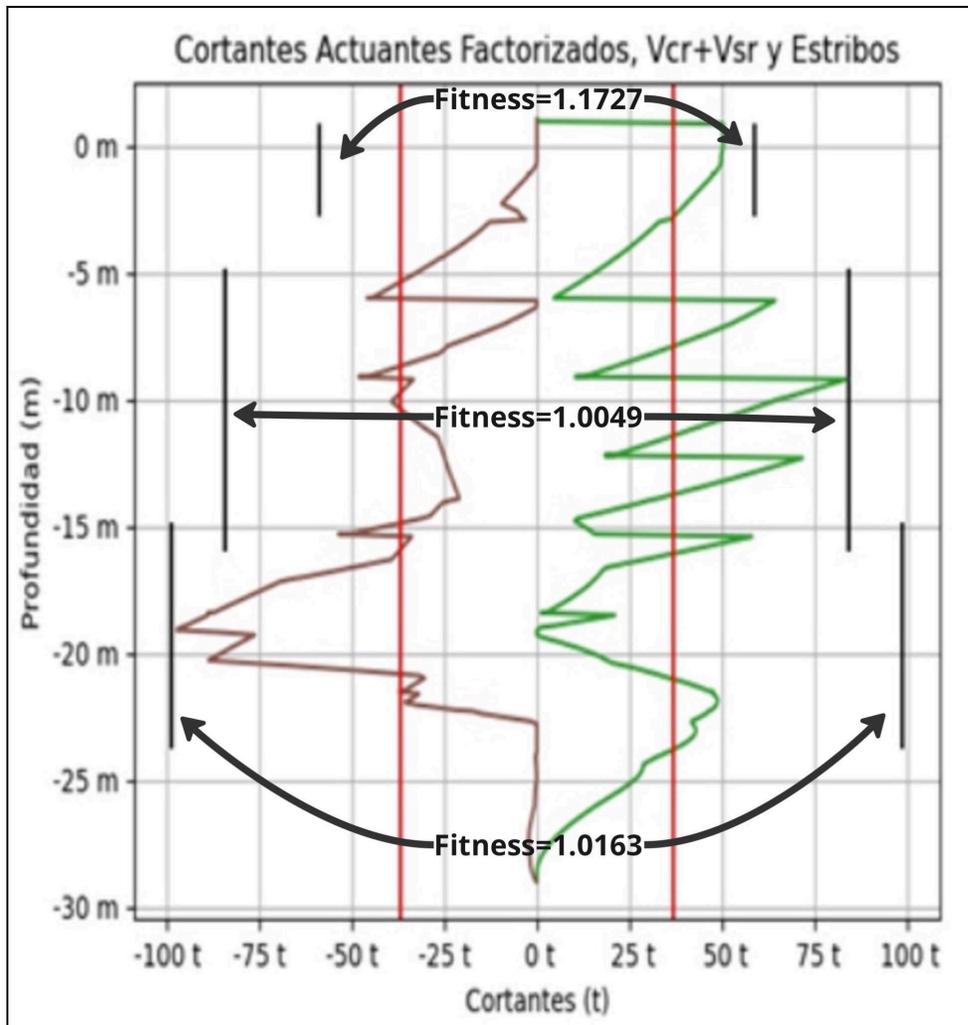


Figura 82. Comparación de resultados: propuesta final de estribos calculados y ordenados por la herramienta de software.

Se puede observar que el algoritmo de ordenamiento reemplazó los arreglos de estribos preliminares del lado negativo por los del lado positivo conservando posición y magnitud.

La tabla mostrada en la figura 83, muestra la información del arreglo de estribos final, dando un total de 3 arreglos de estribos. Esto se debe a que se trata de los mismos arreglos de estribos en ambos lados de la gráfica pero acomodados en “espejo”. En la tabla se puede observar la información detallada de cada uno de los arreglos de estribos, como la posición en profundidad, diámetro de varilla, separación total al corte entre otros valores de gran utilidad.

Tabla Resumen de Cortantes																			
0	Inicio Z (m)	0	Final Z (m)	0	Cortante Máximo (t)	0	Profundidad Z (m)	0	Acero Requerido As (cm ² /m)	0	No. Ramas/m (pzas.)	0	# de Var.	0	Separación Óptima (cm)	0	Res. Acero Est. Vsr (t)	0	Resistencia Vcr + Vsr (t)
1	0.85	1	-2.8	1	50.38	1	0.85	1	3.17	1	2.5	1	4	1	23.54	1	22.25	1	59.08
2	-4.9	2	-16.02	2	83.77	2	-9.22	2	7.13	2	2.5	2	6	2	24.89	2	47.34	2	84.18
3	-14.9	3	-23.8	3	97.48	3	-19.1	3	7.13	3	2.5	3	6	3	18.93	3	62.24	3	99.07

Figura 83. Comparación de resultados: tabla final de estribos calculados y acomodados por la herramienta de software.

Por otra parte, en lo que respecta a la solución con el método manual, la solución depende en gran medida del criterio y forma de pensar del ingeniero de proyecto. En la figura 84 se puede observar la solución de ordenamiento que propone el ingeniero de proyecto.

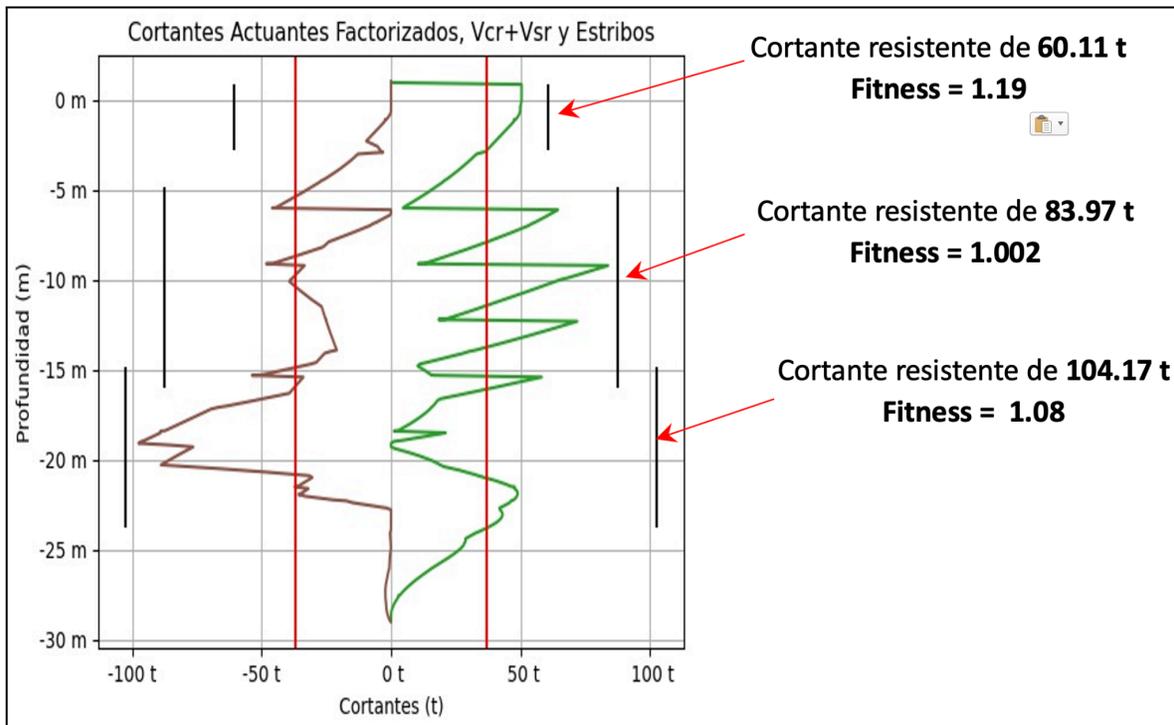


Figura 84. Comparación de resultados: propuesta final de estribos calculados mediante el método manual y acomodados por el ingeniero de proyecto.

De forma similar a lo visto en la herramienta de software, el ingeniero de proyecto generó una tabla en excel con los valores fundamentales de los 3 arreglos de estribos finales, mismos que se pueden apreciar en la figura 85.

ACERO DE REFUERZO ADICIONAL - ESTRIBOS				
CARA EXCAVACIÓN Y CARA TIERRA				
Profundidad, (m)	#ramas	Diámetro del estribo	Separación vertical (cm)	V_R (t)
0.0	2.5	4	22.5	60.11
-9.22	2.5	6	25.0	83.97
-19.00	2.5	6	17.5	104.17

Figura 85. Comparación de resultados: tabla final de estribos calculados mediante el método manual y acomodados por el ingeniero de proyecto.

Asimismo, se puede observar que las soluciones son muy similares entre lo que entrega la herramienta de software y lo que genera el ingeniero de proyecto con el método manual. No obstante, la herramienta de software ofrece una gran ventaja que es el ahorro de tiempo para resolver este apartado, asegurando que las soluciones son de las mejores y, muchas veces, consideradas óptimas gracias al uso del algoritmo genético.

5.3. Mejora en la experiencia de usuario

Los usuarios del programa reportaron una mejora importante respecto a la experiencia de usuario al momento de realizar los cálculos y el diseño de muro pantalla. Se mejoró la interacción con los datos gracias al entorno gráfico de Grasshopper y la claridad visual que ofrece.

El cálculo es más directo y requiere de poca intervención por parte del usuario. Una vez ingresada la ruta del archivo de datos de elementos mecánicos y el directorio de salida, el programa se encarga de realizar los cálculos y verificaciones que antes eran realizados por el ingeniero a través de sus hojas de cálculo. Todos los cálculos, verificaciones y condiciones de diseño se encuentran de manera interna en la lógica y programación del software, así que el ingeniero verifica visualmente que los resultados sean correctos gracias a las gráficas y tablas mostradas en el programa.

Es importante mencionar que el programa contiene algunos parámetros que el usuario puede modificar con el objetivo de alterar la solución, no obstante la mayoría del proceso se realiza de forma automática sin la necesidad de manipular estos parámetros a menos que el usuario lo desee.

5.4. Cumplimiento de los criterios mínimos de aceptación del programa

Tras haber realizado el despliegue de la aplicación, se verificó al 100% el cumplimiento de los requerimientos mínimos del programa con el área de ingeniería y el “equipo BIM”, ver tabla 6.

Nº	Requerimientos mínimos para la automatización de cálculo y diseño de muros pantalla
1	Uso del software Rhinoceros 3D y Grasshopper.
2	Solución automática para acero principal de muros pantalla.
3	Optimización del uso de acero para muros pantalla.
4	Análisis y solución automática de acero de refuerzo adicional.
5	Formato estándar de datos de salida para comunicarlo con el software de dibujo en Tekla.

Tabla 6. Comparación de tiempos según la experiencia de los usuarios..

Se consideraron los requerimientos indispensables para el funcionamiento del programa. Sin embargo, durante el desarrollo del software también se tomaron en cuenta otros requerimientos importantes.

La posibilidad de intervenir de manera manual en la solución del acero de refuerzo del muro pantalla, permitiendo al usuario realizar modificaciones o sugerencias de manera manual.

El programa es amigable con los usuarios, permitiendo una interacción sencilla para modificar diámetros y separaciones del acero de refuerzo del muro. Incluso permite al usuario agregar su propia solución de bastones y estribos. Todo esto a través de archivos Excel que son generados automáticamente por el programa, donde el usuario debe llenar o modificar la información en caso de requerir cambios en la solución e indicarle al programa que tome en cuenta los archivos modificados para actualizar la solución final del programa.

5.5. Solución final entregada por el programa

Una vez ejecutado el programa y que el ingeniero de proyecto esté de acuerdo con la solución, el programa genera como salida una serie de archivos que son almacenados en la carpeta de salida, la cual es seleccionada al iniciar la ejecución del programa.

Los archivos de salida consisten en imágenes en formato JPG de todas las gráficas que son mostradas en el flujo de ejecución del programa. En la figura 86 se muestra una captura de pantalla de la carpeta de imágenes de las gráficas de una de las soluciones. Estas gráficas ayudan al usuario a visualizar de mejor manera la solución en cada etapa, de este modo el usuario puede verificar que el programa lleva la solución por buen camino.

Las imágenes de las gráficas son muy útiles para los ingenieros ya que es común que usen este tipo de gráficas para generar reportes, boletines y diferente tipo de documentación valiosa para “la constructora”.

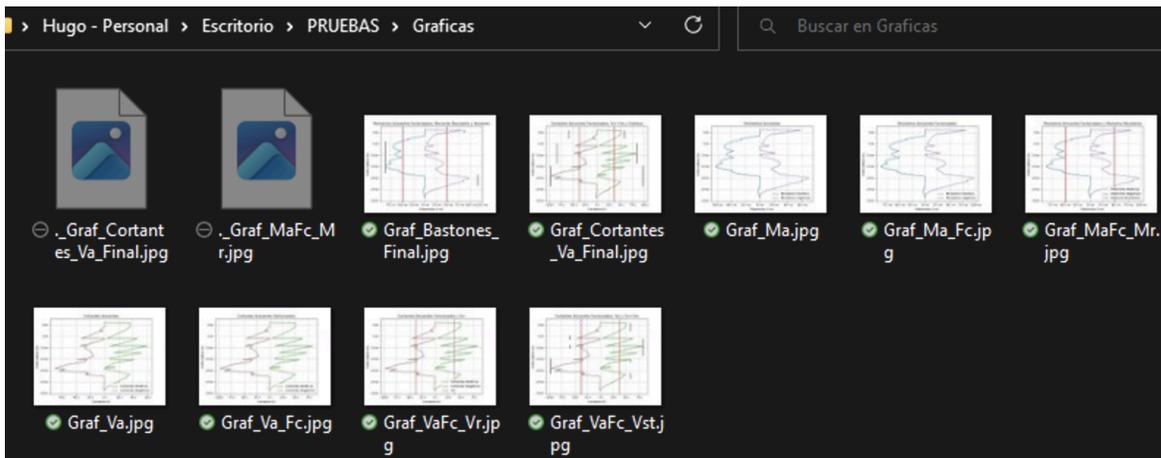


Figura 86. Imágenes de las gráficas presentes en el programa.

El archivo más importante y que corresponde a la solución del acero de refuerzo de muro pantalla es un archivo Excel. Dicho documento contiene 4 hojas que almacenan la información del refuerzo principal, bastones lado positivo, bastones lado negativo y estribos para fuerzas cortantes. Ver figuras 87 a 90.

	C	D	E	F	G	H	I	J	K	
1	h)	Recubrimiento (cm)	fc (kg/cm ²)	fy (kg/cm ²)	#. Var Cara Tierra	Mr Cara Tierra (t m)	#. Var Cara Excavación	Mr Cara Excavación (t m)	No. Var./m (Pzas.)	Separación Ref. Prin. (cm)
2	29	7,5	350	4200	5	47,46	5	47,46	8	13.
3										
4										
5										
6										
7										
8										

Figura 87. Archivo de salida: Hoja con información del refuerzo principal.

	A	B	C	D	E	F	G	H
1	Inicio Bast. en Z (m)	Final Bast. en Z (m)	Longitud Bastón (m)	# de Var.	No. Bastones (Pzas.)	Sep. Bastones (cm)	Momento Res. Bastón (t m)	Mr Ref. Prin. + Mr Bastón (t m)
2	0,95	-0,11	1,06	6	8	13,89	39,6	87,06
3	-18,52	-23,6	5,08	8	8	13,89	68,84	116,31
4								
5								
6								
7								
8								

Figura 88. Archivo de salida: Hoja con información de bastones cara tierra.

	A	B	C	D	E	F	G	H
1	Inicio Bast. en Z (m)	Final Bast. en Z (m)	Longitud Bastón (m)	# de Var.	No. Bastones (Pzas.)	Sep. Bastones (cm)	Momento Res. Bastón (t m)	Mr Ref. Prin. + Mr Bastón (t m)
2	-3,92	-17,8	13,88	6	8	13,89	39,6	87,06
3								
4								
5								
6								
7								
8								

Figura 89. Archivo de salida: Hoja con información de bastones cara excavación.

	A	B	C	D	E	F	G	H	I	J
1	Inicio Z (m)	Final Z (m)	Cortante Máximo (t)	Profundidad Z (m)	Acero Requerido As (cm2/m)	No. Ramas/m (pzas.)	# de Var.	Separación Óptima (cm)	Res. Acero Est. Vsr (t)	Resi
2	0,85	-2,8	50,38	0,85	3,17	2,5	4	23,66	22,14	
3	-4,9	-13,72	83,77	-9,22	4,95	2,5	5	17,41	47,01	
4	-14,9	-23,8	97,48	-19,1	7,13	2,5	6	19,31	61,01	
5										
6										
7										
8										

Figura 90. Archivo de salida: Hoja con información de estribos para cortantes.

El archivo de salida se encuentra listo para ser llamado por los programas desarrollados por el equipo BIM ya que se encuentra en el formato acordado por los miembros del equipo y del área de ingeniería.

5.6. Producto final de la integración del programa con las herramientas desarrolladas por el “equipo BIM”

En este apartado se mostrará el resultado obtenido de la integración de los programas desarrollados en conjunto con mi “equipo BIM”: programa 1 para la mejor distribución de paneles, programa 2 (el explicado en este informe) para el

cálculo y diseño del acero de refuerzo del muro y programa 3 para la representación del muro en 3D en el software Tekla Structures.

Como primer paso para la obtención de planos de taller y representación en 3D de muros pantalla, es necesario definir los tipos y cantidades de paneles de muro pantalla. Esto se logra ejecutando el programa 1, encargado de encontrar la mejor distribución de paneles de muro pantalla, su objetivo es encontrar la combinación de paneles que contenga la menor cantidad de tipos de panel que encierren el perímetro del terreno del proyecto. Es importante recordar que se considera tipo de panel a una sección de muro Milán de una longitud específica, habiendo posibles longitudes desde 6 a 7.2 metros.

Tras ejecutar el programa 1, ver figura 91, se obtiene la mejor distribución de paneles para un predio de geometría cuadrada. Esta solución se exporta a un archivo DWG (Archivo de dibujo) y queda lista para que el dibujante la lleve al software Tekla.

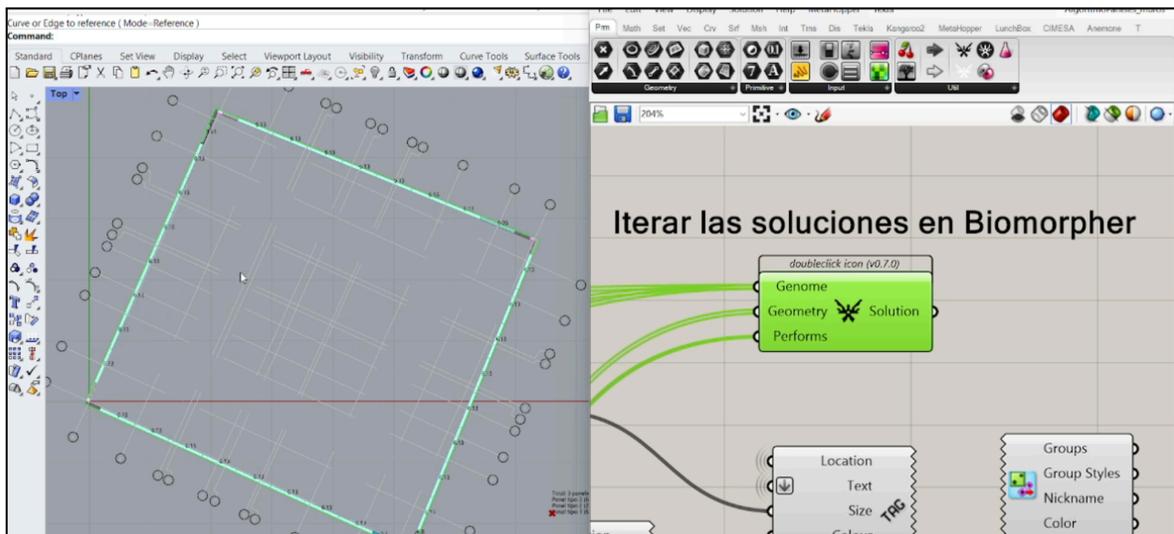


Figura 91. Programa 1: Mejor distribución de paneles de muro pantalla.

En el siguiente paso, se vuelve necesario generar el cálculo y diseño de acero de refuerzo para cada tipo de panel de muro pantalla. Esto es, ejecutar el programa 2 para cada tipo de panel. Tras ejecutar el programa, se obtiene el archivo de datos con la información de toda la solución. Ver figura 92.

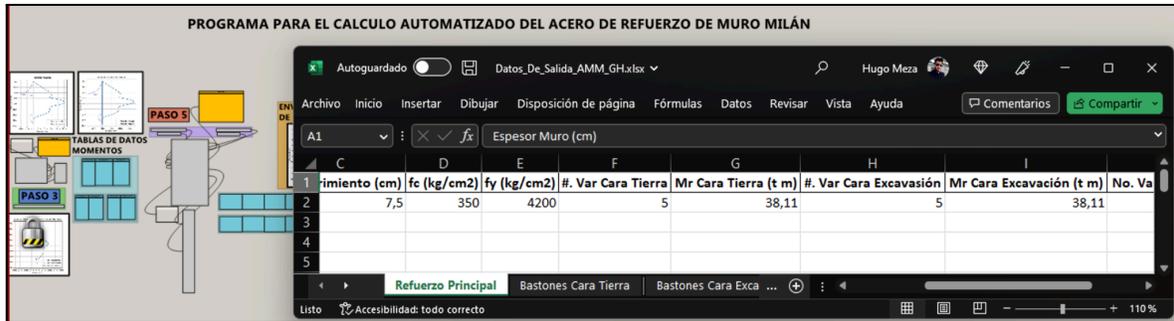


Figura 92. Obtención de la solución para acero de refuerzo de muro pantalla con programa 2.

Posterior a la ejecución de los programas 1 y 2, el área de dibujo puede concluir con la construcción del modelo en 3D del muro pantalla, así como la generación de planos de taller.

El dibujante se encarga de ejecutar el programa 3, cuyo objetivo es importar la información de cálculo y diseño del acero de refuerzo del muro pantalla, obtenida con el programa 2, para generar el modelo en 3D directamente en el software Tekla Structures.

Después de ejecutar el programa 3, se obtiene el modelo en 3D del muro pantalla. En las figuras 93 y 94 se muestran diferentes vistas del modelo 3D del muro pantalla utilizado para este ejemplo.

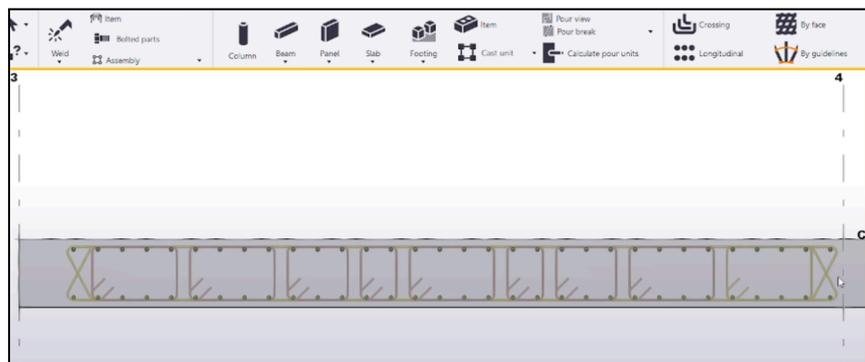


Figura 93. Programa 3: Representación en 3D del muro Pantalla calculado y diseñado con el programa 1 - Vista aérea.

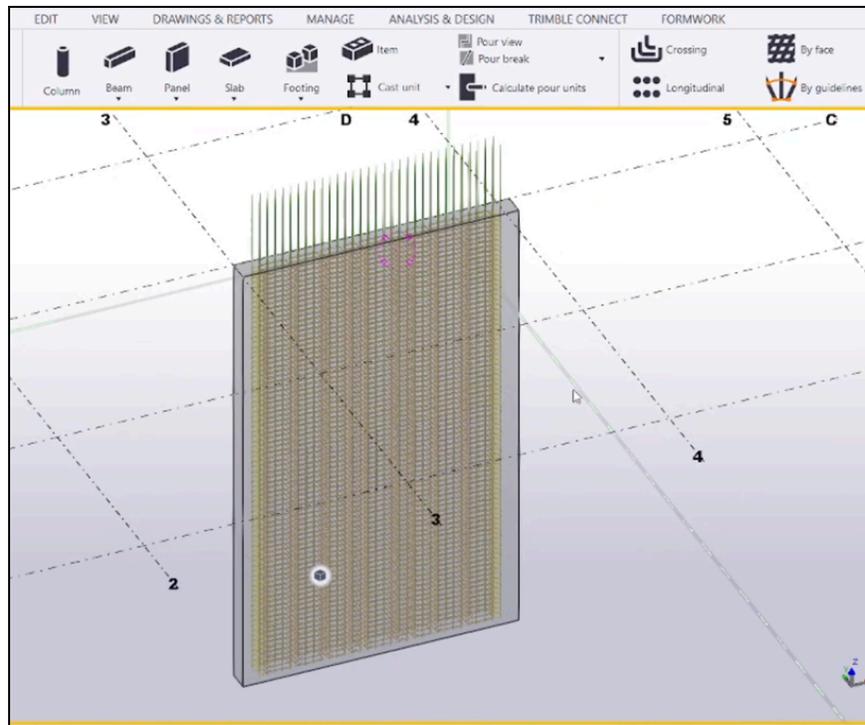


Figura 94. Programa 3: Representación en 3D del muro Pantalla calculado y diseñado con el programa 1 - Vista lateral.

Con el modelo en 3D generado en el software Tekla Structures, el dibujante ya puede obtener el plano de taller del muro pantalla para que sea enviado a obra y se construya sin la necesidad de dibujarlo. A través del software Tekla Structures el dibujante puede exportar el plano de taller del muro pantalla.

El proceso de generar el plano de taller es tan simple como dar una serie de clics para que el software, con toda la información proporcionada de cálculo y diseño, genere el plano de taller incluyendo las características del muro y entregando el plano en el formato que maneja “la constructora”. Ver figura 95.

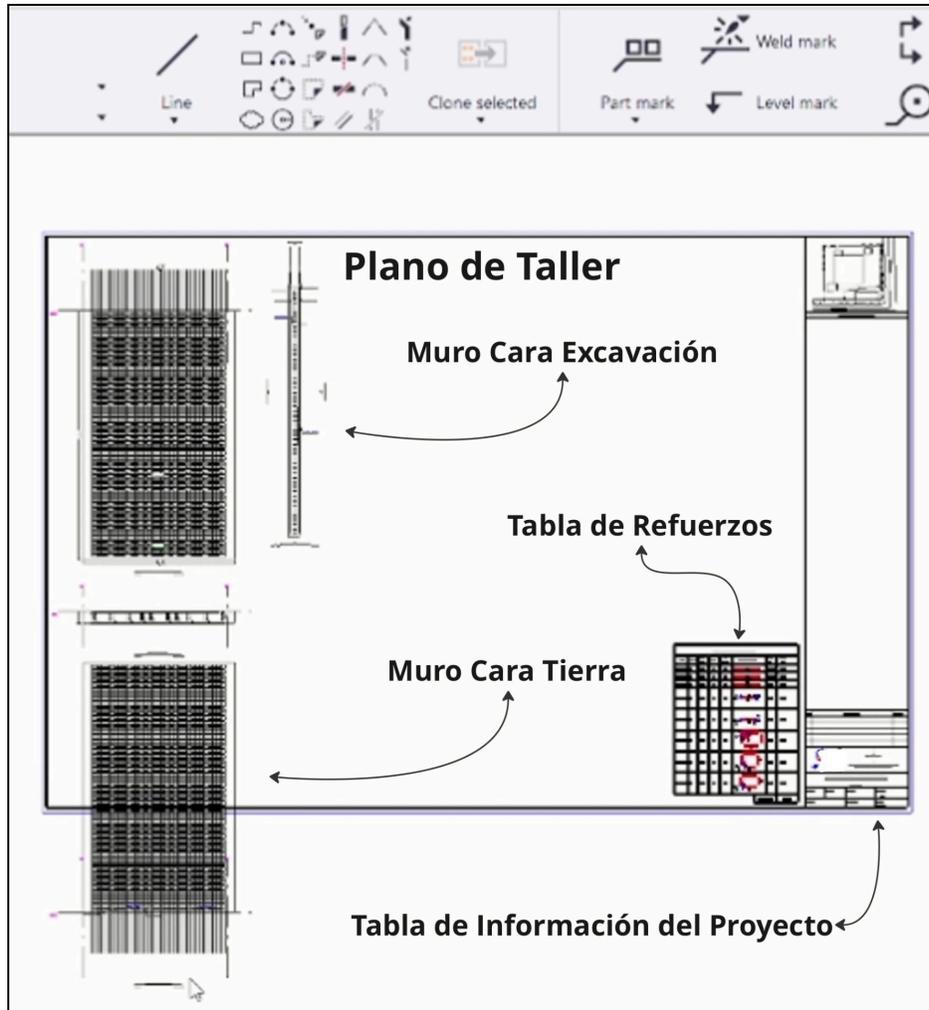


Figura 95. Plano de taller del muro pantalla generado en Tekla Structures con la información del programa 1, 2 y 3. Calidad de imagen reducida.

6. Participación profesional

Durante la culminación de mis créditos de la licenciatura en la Facultad de Ingeniería de la UNAM, en mi trabajo participé en diferentes proyectos y actividades relacionadas con el desarrollo de software. Cada proyecto representó grandes retos debido al constante cambio que existe en la tecnología y las necesidades de las personas que interactúan con ella.

Aunque no contaba con experiencia utilizando la herramienta Rhinoceros Grasshopper, logré adaptarme rápidamente gracias a las bases técnicas adquiridas a lo largo de mi formación como ingeniero en la Facultad de Ingeniería.

El “equipo BIM”, del cual formé parte, está encargado de implementar programas de automatización para el área de ingeniería. Como primer objetivo se tuvo la automatización del proceso de diseño de muros pantalla para la obtención de su representación en 3D y planos de taller, tarea que fue dividida en tres programas mismos que fueron asignados a mi “equipo BIM” conformado por 3 células individuales.

A lo largo del desarrollo del proyecto general, fui responsable de desarrollar en su totalidad la herramienta de cálculo y diseño de acero de refuerzo de muro pantalla (“Programa 2” del proyecto general de “la constructora”). Proponiendo soluciones destinadas a optimizar la solución del acero principal de los muros; implementando la metodología de cálculo y diseño, construyendo algoritmos, que hacen uso de algoritmos genéticos, encargados de buscar y colocar automáticamente refuerzos de acero en las zonas donde los muros requieren ser reforzados y finalmente construyendo el algoritmo encargado de distribuir automáticamente los estribos en el caso de las fuerzas cortantes, y que exista simetría en los refuerzos haciendo un arreglo en espejo con los estribos.

Para desarrollar cada una de las funcionalidades del sistema, utilicé recursos como el diseño de algoritmos, pensamiento analítico, algoritmos de búsqueda y ordenamiento, programación orientada a objetos, software CAD y la ingeniería de software para la parte de administración de proyectos de desarrollo de software. Por otra parte, fuí autodidacta para aprender la teoría de los algoritmos genéticos, su funcionamiento, construcción, implementación y casos de uso, así como estudiar y entender el proceso de diseño y construcción de muros pantalla de la ingeniería civil.

Adicionalmente, también colaboré con el “equipo BIM” en el desarrollo de las otras dos herramientas complementarias que hacen posible la representación del muro en 3D y la obtención de su respectivo plano de taller. Tuve participación con el área de ingeniería de “la constructora” desde el levantamiento de requerimientos y definición de alcance del proyecto general, hasta el lanzamiento a producción de los 3 programas.

Participé en el desarrollo del “Programa 1” del proyecto general de “la constructora”. Dicho programa consiste en la distribución de paneles de muro. En específico, participé en el manejo de la información de salida del programa, implementando un algoritmo de filtrado para selección de las soluciones válidas de los resultados arrojados por el programa. El objetivo del algoritmo era validar que las soluciones cumplieran con un grupo de condiciones y no existieran soluciones repetidas.

En el "Programa 3", sistema encargado de tomar la información del "Programa 2" para representar el muro en 3D en el software Tekla Structures y generar el plano de taller, mi colaboración consistió en el desarrollo de un plugin para Grasshopper, construido con Python, encargado de importar datos de hojas de excel o archivos csv, sirviendo como sustituto a un plugin llamado GhExcel⁶ que se estaba utilizando por el "equipo BIM" para realizar esta función, pero presentó problemas de funcionamiento en algunas computadoras de la empresa.

Realicé el acompañamiento con el área de ingeniería durante el primer proyecto donde se utilizaron las 3 herramientas de software por primera vez en un proyecto nuevo de "la constructora". Asistí a los usuarios del área de ingeniería con dudas y preguntas acerca del programa de automatización de cálculo y diseño del acero de refuerzo de muros pantalla. El área de ingeniería obtuvo resultados satisfactorios con la primera experiencia en producción utilizando las herramientas de software.

Este fue mi primer gran proyecto en el campo laboral de la ingeniería en computación, tratándose de mi primer trabajo, con el cual estuve seriamente comprometido para obtener los mejores resultados y cumplir con los objetivos de "la constructora".

7. Conclusiones

Con el programa 2, mismo que desarrollé y describí a lo largo de este documento, logré automatizar el proceso de cálculo y diseño del acero de refuerzo para muros pantalla, reduciendo significativamente el tiempo invertido en esta tarea y garantizando una buena utilización de la materia prima a través de la optimización del diseño que se obtiene gracias a la implementación del algoritmo genético en este programa.

El desarrollo de software actualmente sigue siendo un campo en constante crecimiento y representa un gran desafío debido a la complejidad en estimaciones de tiempo, gestión y costos. No obstante, gracias a las buenas prácticas que surgen de la ingeniería de software, es posible detectar áreas de oportunidad para que los proyectos tengan éxito y generen los resultados esperados.

Tal es el caso de este proyecto de desarrollo de software, el cual fui responsable de desarrollar y gestionar gracias a la necesidad de "la constructora" de digitalizar y automatizar sus procesos. Utilizando el modelo en cascada del desarrollo de

⁶ GhExcel [24], es un plugin para Grasshopper de Rhinoceros para intercambiar datos con Excel. Permite la lectura estática, dinámica y la escritura de archivos Excel.

software, se comprueba que es una buena manera de gestionar proyectos de desarrollo de software para empresas que buscan la automatización de sus procesos permitiendo diversos cambios durante el desarrollo y garantizando la calidad del producto de software.

Como resultado de implementar esta herramienta de software, “la constructora” obtuvo mejoras en aspectos muy importantes que mencionaré a continuación.

- Una reducción importante de los tiempos de diseño para la realización de planos de muros pantalla derivado de la automatización del proceso de diseño, con el cual es posible generar varios diseños de diferentes tipos de panel de muro pantalla en tiempos muy cortos, comparado con el método tradicional. Se habla de poder generar diseños de 4 a 6 tipos de paneles en un mismo día. Cosa que con el método manual no es posible.
- Se logró una buena optimización de recursos en materia prima (acero) gracias a la implementación de algoritmos genéticos, que ayudan a optimizar los diseños de acero de refuerzo, garantizando el uso de acero necesario para cada muro. Esta ventaja se ve reflejada directamente en un aspecto económico, sin comprometer la fiabilidad y seguridad de los diseños gracias a las validaciones y condiciones de diseño incluidas la lógica del programa de automatización.
- Con este programa se mitiga un inconveniente que el área de ingeniería presentaba con relación al cálculo y diseño de muros pantalla: la susceptibilidad del error humano; lo anterior debido a que el proceso manual implica manipulación de datos y toma de decisiones basadas en la experiencia. Los resultados podrían verse afectados por algún error de cálculo o incluso la generación de diseños con un alto margen de optimización. Ahora, con la herramienta de software se requiere mínima intervención por parte del usuario, disminuyendo enormemente la posibilidad de cometer errores en los diseños de muro pantalla.

Con la automatización a través de la ingeniería en computación aplicada y las tecnologías derivadas de la inteligencia artificial, logré generar software de calidad, que toma en cuenta las diferentes condiciones de diseño así como normativas, replicando y optimizando el proceso que siguen los ingenieros de proyecto de “la constructora” para el cálculo y diseño de muros pantalla, asegurando el cumplimiento de normas, optimización y fiabilidad de los diseños generados por el programa.

Con este trabajo quiero destacar que la ingeniería en computación puede aplicarse en cualquier otra rama de la ingeniería, pudiendo generar software para muchas áreas del conocimiento. Mi experiencia como ingeniero en computación

con un equipo multidisciplinario fue muy enriquecedora, obtuve conocimientos del área de ingeniería civil donde reforcé mi aprendizaje matemático y analítico. Aprendí a utilizar el software CAD Rhinoceros mismo que puede amplificar sus capacidades gracias a la buena integración que tiene con lenguajes de programación de alto nivel.

El aprendizaje que me proporcionó la Facultad de Ingeniería a través de asignaturas de ciencias básicas como mecánica, álgebra, probabilidad, cálculo y geometría me ayudaron a tener la capacidad de plantear problemas desde un enfoque matemático. Asignaturas de ingeniería aplicada como estructuras de datos y algoritmos, programación orientada a objetos, ingeniería de software, administración de proyectos de software, inteligencia artificial y minería de datos, por mencionar las más representativas, me permitieron adaptarme fácilmente a las condiciones del proyecto, agilizando mi adopción de las tecnologías de la metodología BIM para dar una solución robusta a los requerimientos del programa que desarrollé.

8. Referencias

- [1] A. Gavito, ¿Qué hace la ingeniería civil?, México: Universidad Autónoma de Guadalajara, 2022, [En línea], Disponible en: <https://www.uag.mx/es/mediaHub/que-hace-un-ingeniero-civil>
- [2] Guía Técnica, 2da edición. Soletanche Bachy, Francia, 2012, 125pp.
- [3] I. Sommerville, Ingeniería de Software, Novena edición, México: PEARSON, 2011, 773.
- [4] S. Mohanarajah. Software Development Methodologies: A Perspective Of Different Strategies Linked To A Similar Goal. International Conference on Information Technology and Multimedia (ICIMU). Putrajaya, Malaysia, noviembre 2014. IEEE Xplore. Disponible en: 978-1-4799-5423-0/14/\$31.00 ©2014 IEEE
- [5] R. S. Pressman, Ingeniería del Software un enfoque práctico, Séptima edición, México: Mc Graw Hill, 2010, 777.
- [6] B. García y D. Arias, Al-Juarismi, puente matemático entre civilizaciones, OpenMind BBVA, marzo 2019. [En línea] Disponible en: <https://www.bbvaopenmind.com/ciencia/matematicas/al-juarismi-puente-matematico-entre-civilizaciones/>

- [7] D. Gutiérrez, A. Tapia y A. Rodríguez, Algoritmos Genéticos con Python: Un enfoque práctico para resolver problemas de ingeniería, Primera edición, España: MARCOMBO, 2020, 451.
- [8] F. Sancho. Imagen del funcionamiento de los algoritmos genéticos. [En línea] Disponible en: <http://www.cs.us.es/~fsancho/images/2016-05/genetico.png>
- [9] IPN ESCOM. Fundamentos de algoritmos genéticos. Temática unidad 1. [En línea] Disponible en: <https://www.escom.ipn.mx/docs/oferta/matDidacticoISC2009/AGntcs/apuntesAlgsGeneticos.pdf>
- [10] A. Fawzy Gad, PyGAD - Python Genetic Algorithm!, 2020 [En línea] Disponible en: <https://pygad.readthedocs.io/en/latest/>
- [11] AUTODESK, DISEÑO ASISTIDO POR COMPUTADORA-SOFTWARE DE DISEÑO CAD, 2022. [En línea] Disponible en: <https://www.autodesk.mx/solutions/cad-software>
- [12] SIEMENS, Diseño asistido por ordenador (CAD), 2022. [En línea] Disponible en: <https://www.plm.automation.siemens.com/global/es/our-story/glossary/computer-ai-ded-design-cad/12507>
- [13] R. McNeel & Associates, Rhinoceros Información general, 2023 [En línea] Disponible en: <https://www.rhino3d.com/features/>
- [14] AUTODESK, DISEÑO Y CONSTRUCCIÓN CON BIM - MODELADO DE INFORMACIÓN PARA LA CONSTRUCCIÓN, 2022. [En línea] Disponible en: <https://www.autodesk.mx/solutions/bim>
- [15] Trimble. Tekla Structures, 2022. [En línea] Disponible en: <https://www.tekla.com/la/productos/tekla-structures>
- [16] MCAD, Programación visual y sus aplicaciones en BIM, Colombia, noviembre 2022. [En línea] Disponible en: <https://mcad.co/programacion-visual-y-sus-aplicaciones-en-bim/>
- [17] R. McNeel & Associates, Grasshopper - Novedades de Rhino 6l, 2023 [En línea] Disponible en: <https://www.rhino3d.com/es/6/new/grasshopper/>
- [18] S. Davidson. Rhino 7 for Windows and Mac includes Grasshopper. 2023. [En línea] Disponible en: <https://www.grasshopper3d.com/>

- [19] Python Software Foundation, Referencia del lenguaje Python. enero 2023. [En línea] Disponible en: <https://docs.python.org/es/3/reference/index.html>
- [20] W. Bel, Algoritmos y estructuras de datos en Python: un enfoque ágil y estructurado, Primera edición, Argentina: Editorial UADER, 2020, 252.
- [21] Gaceta oficial de la ciudad de México. Normas técnicas complementarias para diseño y construcción de cimentaciones. 1. 15 de diciembre de 2017.
- [22] Gaceta oficial de la ciudad de México. Normas técnicas complementarias para diseño y construcción de estructuras de concreto. 5. 15 de diciembre de 2017.
- [23] M. A. Rahman, Gh_CPython: CPython plugin for Grasshopper. 2017. [En línea] Disponible en: https://github.com/MahmoudAbdelRahman/GH_CPython
- [24] Xiaoming. GHEXCEL - INTERFACE WITH EXCEL. Food4Rhino. [En línea] Disponible en: <https://www.food4rhino.com/en/app/ghexcel-interface-excel>
- [25] Pilotech Levante. ¿Qué son los muros pantalla? [En línea] Disponible en: <https://www.pilotechlevante.com/que-son-los-muros-de-pantalla>
- [26] J. Cervantes et al. Introducción a la programación orientada a objetos, primera edición, México: UAM Unidad Cuajimalpa, 2016, 200.
- [27] IBM, Programación orientada a objetos, 2017. [En línea] Disponible en: <https://www.ibm.com/docs/es/spss-modeler/saas?topic=language-object-oriented-programming>