



UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

**DEPARTAMENTO DE CONSTRUCCIONES ARQUITECTÓNICAS, INGENIERÍA DEL
TERRENO, MECÁNICA DE LOS MEDIOS CONTINUOS Y TEORÍA DE ESTRUCTURAS**

TESIS DOCTORAL

***Metodología para el desarrollo de entornos amigables
de diseño distribuido en ingeniería utilizando Internet***

***Aplicación al diseño de cajones flotantes de hormigón
armado para muelles y diques***

***Presentada por JOSÉ PEREDA LLAMAS para
optar al grado de doctor por la Universidad de Valladolid***

Dirigida por
ANTONIO FOCES MEDIAVILLA
Dr. Ingeniero Industrial

ESCUELA DE INGENIERÍAS INDUSTRIALES

DEPARTAMENTO DE CONSTRUCCIONES ARQUITECTÓNICAS, INGENIERÍA DEL
TERRENO, MECÁNICA DE LOS MEDIOS CONTINUOS Y TEORÍA DE ESTRUCTURAS

TESIS DOCTORAL

***Metodología para el desarrollo de entornos amigables
de diseño distribuido en ingeniería utilizando Internet***

***Aplicación al diseño de cajones flotantes de hormigón
armado para muelles y diques***

Presentada por **JOSÉ PEREDA LLAMAS**, Ingeniero Industrial por la
Escuela de Ingenierías Industriales de la Universidad de Valladolid

Dirigida por **ANTONIO FOCES MEDIAVILLA**, Profesor Titular en el
Departamento de Construcciones Arquitectónicas, Ingeniería del Terreno y
Mecánica de los Medios Continuos y Teoría de Estructuras, Universidad de
Valladolid

Esta Tesis Doctoral fue leída en la Escuela de Ingenierías Industriales de la
Universidad de Valladolid, el día _____, estando compuesto el
tribunal calificador por:

Presidente

Secretario

Vocal Primero

Vocal Segundo

Vocal Tercero

Obteniendo la calificación de _____ .

Valladolid, a _____ de _____ de 2013

*Anyone who has never made a mistake
has never tried anything new*

A. Einstein

A mis hijos, de ellos es el futuro

Agradecimientos

Quiero expresar mi más sincero agradecimiento a las personas que han hecho posible la realización y conclusión de esta Tesis Doctoral después de tanto tiempo.

Empezando por mis padres que fueron los que permitieron que iniciara esta andadura hace ya muchos años y siempre han estado ahí confiando en mí y en mis posibilidades de llegar a buen puerto.

Siguiendo por mi mujer y mis hijos, que han sabido entender el sacrificio que suponen las largas horas de dedicación que les he robado.

A mis suegros, en estos difíciles momentos, por haberme animado incondicionalmente a concluir la Tesis.

A Ricardo, por las horas interminables de trabajo compartido, la ilusión permanente y la esperanza de un tiempo mejor.

A mi director de Tesis, Antonio Foces, por sus consejos y aportaciones, por estar ahí siempre.

A José Antonio Garrido, allá donde esté.

A los profesores Eugenio Oñate y Benjamín Suárez, por haber posibilitado la colaboración entre CIMNE y el Departamento, haber creído en mis posibilidades, y haber iniciado el proyecto *Cajones* que hoy da sus frutos en este documento.

A mis compañeros de Departamento, Mariano, Antolín y Juan Carlos, especialmente, por los largos años de convivencia.

A Fernando Jordán, compañero de viaje en el proyecto *Cajones*, a Ángel Moreno y Martín Rey, que con la realización de sus PFCs contribuyeron a este gran proyecto.

A Enrique Escolano y Francisco Zárate, de CIMNE, por sus aportaciones para la integración del proyecto *Cajones* con GiD y CALSEF, respectivamente.

A todos, de verdad, muchas gracias.

Jose

Resumen

En las zonas portuarias se realizan continuas actuaciones de ampliación y mejora, siendo el dique la obra por antonomasia, bien sea para espigones de protección o para líneas de atraque, muelles, etc. Fundamentalmente estos son de escollera o diques verticales mediante cajones aligerados sobre banquetas. España es uno de los países con más experiencia en la utilización de cajones aligerados y uno de los más avanzados en el desarrollo de la tecnología necesaria para su fabricación.

El objetivo es diseñar elementos prefabricados de gran volumen y de relativamente poco peso en relación a éste, con los que cubrir grandes zonas de la obra portuaria. Particularmente, en lo que concierne al trazado de diques mediante alineamiento de cajones en zonas de gran calado, la definición inicial de la geometría del cajón es la fase en la que el proyectista debe tomar un mayor número de decisiones para armonizar un amplio listado de prescripciones. Entre otras, estas disposiciones hacen referencia al rango de dimensiones admisibles en cada caso, a la infraestructura de fabricación disponible (las características del encofrado inciden directamente en el grado de aligeramiento del fuste a través del diámetro de las celdas), a las condiciones de navegabilidad en transporte o de estabilidad en construcción.

En el cálculo de armaduras, que sigue al establecimiento de la geometría del cajón, deben considerarse un elevado número de solicitaciones cualitativamente distintas. Así, por ejemplo, además de las acciones de empuje y succión que provoca el oleaje, hay que estudiar las provenientes del programa de lastrado de celdas para controlar el brazo estabilizador en la fabricación del cajón y el resguardo necesario en los trabajos de encofrado. La anegación y el rellenado de celdas para hundir la estructura en la posición definitiva en condiciones de bajamar o de pleamar, el empuje del terreno en el lado tierra o la presencia de superestructuras cerrando el cajón superiormente, junto con las cargas de explotación derivadas del uso final del conjunto, son también algunas solicitaciones a atender en el cálculo.

En este documento se describe un modelo completo para el diseño y cálculo resistente de cajones y la definición de las armaduras necesarias. En paralelo, se ha desarrollado una aplicación informática para la realización de todas y cada una de las etapas involucradas, automatizando aquellos procesos que no requieran la intervención del usuario. El código está concebido para asistir al proyectista en el establecimiento de la geometría y de las hipótesis de carga a considerar, incluidas las correspondientes a las etapas de construcción, transporte y hundimiento, además de las de servicio.

Finalizada esta primera fase, el usuario remite el diseño vía Internet a un servidor remoto que se encarga del cálculo resistente, de la determinación de cuantías y de la homogeneización de la solución de armado. Se consigue así trasladar al servidor los altos requerimientos en cuanto a coste computacional asociados a la etapa de cálculo.

El cálculo en el servidor consiste básicamente en un análisis tensional tridimensional del cajón discretizado mediante elementos finitos hexaédricos cuadráticos de veinte nodos.

Para la generación de una malla mediante el programa GiD de CIMNE se ha desarrollado un algoritmo adaptado a la particular geometría del cajón convirtiendo mediante técnicas multibloque la geometría inicial en un conjunto de volúmenes en forma de hexaedros, con los cuales es inmediato realizar una malla estructurada, teniendo en cuenta las posibles variaciones en la tipología y forma de los aligeramientos de la sección, así como las exigencias de definición en las gargantas en las que se realizará con posterioridad el armado.

Tras determinar las tensiones con la aplicación CALSEF, también de CIMNE, se ha desarrollado un algoritmo de integración de las mismas en las gargantas para evaluar los esfuerzos y determinar así las cuantías necesarias, tanto en las armaduras horizontales que se disponen en forma de aros alrededor de las celdas, como en las barras verticales. El proceso finaliza con un programa de homogeneización de armado apropiado a la práctica constructiva.

Es importante indicar que las condiciones en cuanto a espesores de pared y recubrimientos hacen necesario confiar al hormigón la resistencia a cortante, de forma que, salvo la losa que cierra inferiormente a la estructura, no se disponen armaduras de cortante en este tipo de elementos. Nótese también que el cajón es una pieza tridimensional geoméricamente peculiar que no presenta ningún plano predominante de flexión por lo que no es de aplicación directa la teoría general de piezas lineales en cuanto al cálculo de armaduras. Por otro lado, el método alternativo de bielas y tirantes daría lugar a celosías equivalentes muy complicadas y de difícil interpretación y ejecución. Por ello, la técnica que se presenta es una alternativa a los modelos 2D mediante elementos finitos combinados con prácticas basadas en la teoría de silos procedimientos estos con los que se han calculado un buen número de cajones actualmente en servicio.

Un aspecto destacable a la hora de desarrollar sistemas de software es el diseño de su arquitectura, entendida ésta como la estructura de los componentes de un programa o sistema, sus interrelaciones y los principios o reglas que gobiernan su diseño y evolución en el tiempo. El nivel de abstracción y las posibilidades de reutilización de la misma o de algunos de sus componentes aumentan mediante su descripción a través de diagramas y vistas lógicas realizados mediante lenguajes de modelado como UML. Las ideas que sustentan esta filosofía de programación se plasman en la realización de frameworks o marcos de trabajo, incluyendo una serie de componentes que serán instanciados, extendidos y reutilizados en cada caso concreto. Un framework permite encapsular un diseño marco específico de un dominio, con la idea de que pueda ser aprovechado por una familia de aplicaciones. Para materializarlo se implementan diversos patrones de diseño, que son otro enfoque útil para construir una arquitectura marco documentada.

En el caso concreto de este trabajo, se ha optado por una arquitectura multi-plataforma basada en un esquema tipo Cliente/Servidor que incluye varias capas de software (*multi-tier*), cada una de ellas con responsabilidades específicas y bien delimitadas. Se ha conseguido que la parte del proceso relativa al cálculo, la de mayor coste computacional, se ejecute en el Servidor con posibilidad de reparto equilibrado de los trabajos en diferentes equipos. Como consecuencia, en el ordenador Cliente sólo son necesarios recursos mínimos.

Las conexiones y transmisiones de datos entre el cliente y el servidor se establecerán mediante llamadas del cliente al servidor a través de un contenedor de aplicaciones, encargado de gestionar una serie de programas residentes, que en última instancia son los que gestionan con el PC la realización del cálculo en local en el software específico (motor de cálculo), chequean el estado del mismo y tras la finalización devuelven los resultados cuando el usuario los solicite.

Por tanto, en este trabajo se desarrolla una metodología que permita la implementación de aplicaciones de diseño y análisis de elementos en ingeniería civil, distribuidas en varios ordenadores conectados entre sí a través de Internet.

Como aplicación concreta de la metodología a desarrollar, se adaptará al diseño y análisis de cajones multicelulares aligerados de hormigón armado.

Razón de ser

Esta aplicación concreta se plantea a partir del Proyecto I+D titulado '*Desarrollo de un entorno informático para diseño y análisis por el método de elementos finitos de cajones multicelulares de hormigón armado*' generado a partir del Convenio entre la Universidad de Valladolid, su Fundación General y el Centro Internacional de Métodos Numéricos en Ingeniería (CIMNE) iniciado en julio de 2001 y refrendado con la creación del Aula UVa-CIMNE a fecha 18 de abril de 2002.

Fruto del convenio y del trabajo realizado se han calculado diversos cajones para los Puertos de Barcelona y Bilbao, que hoy en día se encuentran fondeados y prestando servicio a plena satisfacción.

Índice general

Razón de ser	13
Capítulo I. Introducción y Objetivos	25
I.1.- Introducción	25
I.2.- Objetivos	26
I.3.- Organización del Documento	27
Capítulo II. Estado del Arte	29
II.1.- Introducción a los cajones portuarios	29
II.2.- Introducción a los frameworks	36
II.2.1.- Framework con arquitectura multiplataforma	39
II.2.2.- El nacimiento de la Web	40
II.2.3.- El contenido dinámico	43
II.2.4.- Los servlets	46
II.2.5.- Web Application Frameworks <i>WAF</i>	47
II.2.6.- WAFs en Java	51
II.2.7.- Aplicaciones de frameworks a la ingeniería civil	54
Capítulo III. Los Cajones Multicelulares	55
III.1.- La geometría de un cajón	55
III.1.1.- Descripción de la geometría de un cajón tipo	55
III.1.2.- La sección transversal	57
III.1.3.- Determinación de la eslora y la manga	59
III.1.4.- Las celdas ovoidales	62
III.1.5.- Grado de achaflanamiento	64
III.1.6.- Tipología de aligeramientos	65
III.1.7.- Posicionamiento y conteo de aligeramientos	67
III.1.8.- Área aligerada de la sección transversal	73
III.1.9.- Centro de gravedad de la sección transversal	74
III.1.10.- Momentos de Inercia de la sección transversal aligerada	76
III.1.11.- Determinación de las variables de la sección	77
III.1.12.- Volumen	79
III.1.13.- Estabilidad naval	84
III.2.- Geometría multi-bloque por tipologías	91
III.2.1.- Elección del tipo de elemento finito	91
III.2.2.- Gestión del mallado de cada celda	110
III.2.3.- Calidad y edición de la geometría multibloque	133
III.3.- Las acciones en el cajón	136
III.3.1.- Base de diseño	136
III.3.2.- Casos de carga considerados en el estudio de un cajón	139
III.3.3.- Aplicación de las condiciones de contorno en carga y desplazamientos a la geometría multibloque	157
III.3.4.- Modelado de las hipótesis	158
III.4.- Esquema General de Resolución mediante el MEF	163
III.4.1.- Introducción	163
III.4.2.- Pre-procesado en GiD	164
III.5.- Generación de esfuerzos en secciones críticas	202
III.5.1.- Generación de gargantas	203
III.5.2.- Esfuerzos horizontales en gargantas verticales	213

III.5.3.- Esfuerzos verticales en gargantas horizontales	217
III.5.4.- Validación de resultados	221
III.5.5.- Esfuerzos en la losa	223
III.6.- Armado de los elementos estructurales	224
III.6.1.- El armado según la EHE	224
III.6.2.- Planteamiento del problema	225
III.6.3.- Hipótesis de Cálculo.....	226
III.6.4.- Caracterización de la ley de deformaciones en el Estado Límite de Agotamiento Resistente	228
III.6.5.- Ecuaciones de campo en el Estado Límite de Agotamiento Resistente..	231
III.6.6.- Dimensionado en el Estado Límite de Agotamiento Resistente por flexión	237
III.6.7.- Dimensionado en el Estado Límite de Agotamiento Resistente por cortante	241
III.6.8.- Obtención de los resultados de armado.....	242
Capítulo IV. Un Framework para el cálculo de cajones.....	249
IV.1.- Descripción del framework para el cálculo de cajones.....	249
IV.1.1.- Análisis de los requisitos	252
IV.1.2.- Tecnología empleada	254
IV.1.3.- Arquitectura del Framework	254
IV.1.4.- Seguridad	256
IV.1.5.- Identificación de entidades.....	256
IV.1.6.- Servicios.....	260
Capítulo V. Cajones Realizados	277
V.1.- Introducción	277
V.2.- Puerto de Barcelona	277
V.2.1.- Abril 2004	277
V.2.2.- Marzo 2005	281
V.2.3.- Marzo 2005	284
V.2.4.- Junio 2008	288
V.2.5.- Octubre 2008	292
V.2.6.- Octubre 2008	296
V.2.7.- Marzo 2009	300
V.3.- Puerto de Bilbao	305
V.3.1.- Noviembre 2008	306
V.3.2.- Junio 2009	311
Capítulo VI. Conclusiones y Líneas Futuras	315
VI.1.- Aportaciones originales.....	315
VI.2.- Conclusiones	315
VI.3.- Líneas futuras.....	316
Bibliografía.....	319

Índice de figuras

Figura II-1. Perfil constituido por bloques artificiales de hormigón del dique del puerto de Barcelona a finales del XIX. De ‘El puerto de Barcelona’, de José Ayxelá [1].....	29
Figura II-2. Tipología de diques de abrigo: en talud (izquierda), verticales de bloques o monolíticos (centro) y verticales compuestos (derecha) [9].....	30
Figura II-3. Infraestructura para fabricar cajones [6]	31
Figura II-4. Prolongación del dique de Poniente en el puerto de Motril [3]	32
Figura II-5. Planta general del Proyecto de Muelle Adosado AZ-2 de la AP de Bilbao adosado al dique de Zierbana	32
Figura II-6. Obra de atraque de cajón flotante [11].....	33
Figura II-7. Ejemplos de tipologías estructurales según aligeramiento rectangular en el puerto de Cartagena (izquierda, [26]) o circular, en el puerto de Santander (derecha)..	33
Figura II-8. Diferentes leyes de presiones de oleaje sobre el paramento vertical del cajón,	34
Figura II-9. Mezcla de modos de fallo en un muelle [17].	35
Figura II-10. Desplazamiento de 16 de los 37 cajones del Muelle del Prat del Puerto de Barcelona por licuefacción del terreno [22], enero 2007. Se tuvieron que demoler 5 y se pudieron reflotar los 11 restantes.....	36
Figura II-11. Operaciones de demolición de cajones dañados [23]	36
Figura II-12. Contexto de aplicación de arquitectura multicapa	40
Figura II-13. Captura de pantalla del primer navegador, el WorldWideWeb, tomada en 1993 [40]	41
Figura II-14. Página web en Mosaic [43].....	42
Figura II-15. Historia de los navegadores web [44].....	43
Figura II-16. Esquema general del servidor de aplicaciones JOnAS [58].....	45
Figura II-17. Diagrama UML de clases del Java Servlet API [63]	47
Figura II-18. Arquitectura de tres capas [64]	48
Figura II-19. Comparativa de patrón síncrono tradicional con asíncrono en AJAX [65]	49
Figura II-20. Modelo básico del funcionamiento de los servicios Web [67]	50
Figura II-21. Estructura modular de Spring [70].....	52
Figura II-22. Arquitectura del <i>runtime</i> de Vaadin [73]	53
Figura III-1. Vista general en planta y en 3D de un cajón flotante	55
Figura III-2. Sección transversal con retícula superpuesta de hexágonos (rojo) y de triángulos equiláteros (verde).	58
Figura III-3. Cotas en la sección transversal en el Lado Tierra.....	58
Figura III-4. Ejemplo de secciones: a) con simetría, y b) sin simetría	59
Figura III-5. Sección maciza y cotas	60
Figura III-6. Definición geométrica de los topes laterales	60
Figura III-7. Cajón multicelular con doble tope en LT y LM	61
Figura III-8. Variables empleadas en la determinación del ovoide.....	63
Figura III-9. Trazado del ovoide en una esquina del Lado Tierra.....	64
Figura III-10. Ángulos definidos a partir del grado de achaflanamiento	64
Figura III-11. División de una sección transversal según tipologías.....	66
Figura III-12. Tipología de aligeramientos	67
Figura III-13. Diseño de la sección transversal según criterios adoptados	68
Figura III-14. Secciones con simetría transversal y filas y columnas pares o impares..	69

Figura III-15. Algoritmo de colocación de aligeramientos para definir la sección transversal por tipologías.....	71
Figura III-16. Centros de referencia de los aligeramientos de la sección.....	74
Figura III-17. Acopio de encofrados usados previos a colocación en pontona.....	78
Figura III-18. Losa de cierre inferior, con voladizos y cartabones.....	80
Figura III-19. Cajón con muro perimetral. Muelle del Prat Fase I-A, Puerto de Barcelona, 2005.....	81
Figura III-20. Superestructura en línea de cajones ya fondeada (izquierda) junto a nuevo cajón a fondear (derecha)	83
Figura III-21. Viga cantil integrada en el cajón. Muelle del Prat, Fase II, Puerto de Barcelona, 2009.....	84
Figura III-22. Condiciones de equilibrio de un cajón en flotación.....	85
Figura III-23. Estado de flotación del cajón tras una rotación	87
Figura III-24. Posición de equilibrio con escora transversal.....	89
Figura III-25. Pesos y empujes en la construcción de un cajón. Zonas de estabilidad...	91
Figura III-26. Sección del fuste en GiD®	92
Figura III-27. Malla no estructurada. Tamaño elemento 1, línea 0.5	93
Figura III-28. Malla no estructurada. Tamaño elemento 1, línea 0.1	94
Figura III-29. Malla no estructurada. Tamaño elemento 1, línea 0.5.....	95
Figura III-30. Malla no estructurada. Tamaño elemento 1, línea 0.1.....	95
Figura III-31. Malla generada por un método algebraico.....	96
Figura III-32. Malla generada mediante resolución de EDP de Poisson [96].....	97
Figura III-33. Técnica de barrido [100].....	97
Figura III-34. Método de división de triángulos en cuadriláteros	97
Figura III-35. Método de división de tetraedros en hexaedros.....	98
Figura III-36. Método de combinación de triángulos en cuadriláteros [104].....	98
Figura III-37. Técnica de <i>medial axis</i>	98
Figura III-38. Técnicas de <i>plastering</i> y <i>Whisker Weaving</i>	99
Figura III-39. Generación de malla multibloque tras descomposición inicial del dominio [115]	100
Figura III-40. Reglas usadas para manejar el contorno de una parte	101
Figura III-41. Topología de bloques global y malla final generada en dominio con 4 círculos	101
Figura III-42. Celda tipo <i>Circular</i> . 12 cuadriláteros	103
Figura III-43. Celda tipo <i>Circular-Mitad</i> . 6 cuadriláteros	103
Figura III-44. Celda tipo <i>Ovoide</i> . 34 cuadriláteros	104
Figura III-45. Detalle de generación de cuadriláteros de la celda <i>Ovoide</i>	104
Figura III-46. Celda tipo <i>Circular-Ovoide</i> . 16 cuadriláteros	105
Figura III-47. Celda <i>Ovoide-Esquina</i> . 45 cuadriláteros	105
Figura III-48. Celda <i>Circular con Chaflán Frontal</i> . (a) Forma general de la celda. (b) a (d) División en cuadriláteros según el grado de chaflán	106
Figura III-49. (a) Celda <i>Circular-Mitad con Chaflán Frontal</i> . (b) y (d) División en 15 cuadriláteros, (c) división en 17 cuadriláteros.....	107
Figura III-50. Celda <i>Circular-Ovoide con Chaflán Frontal</i> . División en 36 cuadriláteros, con ovoide corto de esquina.....	107
Figura III-51. Celda <i>Circular con Chaflán Lateral</i> . (a) Forma de la celda. (b) División en 34 cuadriláteros, sin chaflán (b) División en 32 cuadriláteros con chaflán.....	108
Figura III-52. Celdas de <i>Esquina</i> , casos según grado achaflanamiento paredes.....	109
Figura III-53. Celda <i>Tope</i> . División en cuadriláteros según celdas adyacentes.....	110
Figura III-54. Diagrama UML de la clase CPunto.....	111

Figura III-55. Diagrama UML de la clase CColPuntos	112
Figura III-56. Diagrama UML de la clase CLinea.....	112
Figura III-57. Diagrama UML de la clase CColLineas	113
Figura III-58. Diagrama UML de la clase CArco	114
Figura III-59. Diagrama UML de la clase CColArcos	115
Figura III-60. Diagrama UML de la clase CSuperficie.....	116
Figura III-61. Diagrama UML de la clase CColSuperficies	117
Figura III-62. Diagrama UML de la clase CVolumen.....	117
Figura III-63. Diagrama UML de la clase CColVolumenes.....	118
Figura III-64. Diagrama UML de la clase CeldaBean (indicados sólo los métodos get).....	119
Figura III-65. Diagrama UML de la interfaz CeldaFactory	119
Figura III-66. Diagrama UML de la clase abstracta CeldaBase	120
Figura III-67. Diagrama UML de la clase CeldaCircular.....	121
Figura III-68. Interfaz CeldasID con los tipos posibles de celdas.....	121
Figura III-69. Diagrama UML de la interfaz MultiMap.....	122
Figura III-70. Diagrama UML de la clase AbstractMultiMap.....	123
Figura III-71. Diagrama UML de la clase CMultiMapCeldas	124
Figura III-72. Diagrama UML de la clase CCubo	125
Figura III-73. Diagrama UML de la clase CMultiMapCubos	126
Figura III-74. Diagrama UML de la clase CCajon.....	127
Figura III-75. Resumen de generación de bloques hexaédricos por tipología de celda	128
Figura III-76. Resumen de generación de cubos según tipologías por rebanada	129
Figura III-77. Resumen de generación de cubos del cajón según rebanadas	129
Figura III-78. Sección del fuste generada con el algoritmo multibloque	130
Figura III-79. Malla estructurada. 1 elemento por lado.....	131
Figura III-80. Malla estructurada. 2 elementos por lado.....	131
Figura III-81. Malla estructurada. 3 elementos por lado. 2610 cuadriláteros y 3052 nodos.....	132
Figura III-82. Mallas finas estructurada (672 nodos) y no estructurada (669 nodos) en el entorno de una celda circular.....	132
Figura III-83. Distribución de elementos clasificados por su mínimo ángulo	133
Figura III-84. Visualización de superficies con pequeños ángulos mínimos.....	134
Figura III-85. Edición de puntos o superficies para corrección de ángulos mínimos ..	135
Figura III-86. Corrección de superficie, con restricción de movimiento circunferencial	136
Figura III-87. Calado del cajón y cargas durante el transporte	141
Figura III-88. Condiciones de equilibrio del cajón en flotación	142
Figura III-89. Diferencia entre área modelada y área real.....	142
Figura III-90. Fondeo del cajón y cargas según pleamar o bajamar.....	144
Figura III-91. Posibles situaciones de relleno parcial de celdas.....	146
Figura III-92. Situaciones de relleno parcial o completo en bajamar o pleamar.....	146
Figura III-93. Ejemplo de curvas de presión de oleaje en el intradós	147
Figura III-94. Descomposición de las presiones sobre paredes exteriores en la primera fase de relleno del trasdós.....	148
Figura III-95. Presiones sobre el cajón en la primera fase de relleno del trasdós en pleamar o bajamar	148
Figura III-96. Presiones sobre el cajón en la puesta en servicio, en pleamar o bajamar	149
Figura III-97. Situación del área de operación en un muelle.....	149

Figura III-98. Cargas de operación y mantenimiento.....	150
Figura III-99. Ejemplo de grúa de movilidad restringida.....	151
Figura III-100. Cargas debidas a trenes de movilidad restringida.....	151
Figura III-101. Cargas debidas a trenes de movilidad no restringida, 2 casos.....	151
Figura III-102. Elementos de bolardo y defensas sobre viga cantil	152
Figura III-103. Cargas del tiro de bolardo	152
Figura III-104. Impacto en las defensas durante el atraque de un buque	153
Figura III-105. Envolvente de tensiones máximas en la losa.....	156
Figura III-106. Presión máxima en las celdas circulares.....	156
Figura III-107. Diagrama UML de la clase CCBean.....	157
Figura III-108. Diagrama UML de la clase CSalidaCC	158
Figura III-109. Diagrama UML del paquete HipotesisTipo	158
Figura III-110. Diagrama UML de la interfaz HipotesisFactory	158
Figura III-111. Diagrama UML de la clase HipotesisBase.....	159
Figura III-112. Diagrama UML de la clase CMultiMapHipotesis.....	159
Figura III-113. Diagrama UML de la clase CHipRelleno.....	160
Figura III-114. Diagrama UML del paquete de Calculo.....	160
Figura III-115. Diagrama UML de la clase de CProcesaHipotesis	161
Figura III-116. Diagrama UML de la clase de CCombinaHipotesis	161
Figura III-117. Diagrama UML de la clase de CArmaHipotesis.....	162
Figura III-118. Geometría modelada: Losa y fuste de un cajón multicelular.....	169
Figura III-119. Detalle de la geometría en torno a una celda.....	170
Figura III-120. Vista de las divisiones del cajón en altura	171
Figura III-121. Diálogo de creación y asignación de materiales en GiD	172
Figura III-122. Diálogo para introducción de datos generales del problema en GiD ..	172
Figura III-123. Visualización de restricciones superficiales en GiD.....	173
Figura III-124. Diálogos de asignación de cargas puntuales o superficiales en ejes globales en GiD	173
Figura III-125. Visualización de presiones uniformes sobre superficies en GiD.....	174
Figura III-126. Visualización de superficies con presión hidrostática asignada en GiD	175
Figura III-127. Malla realizada en losa y fuste del cajón	178
Figura III-128. Vista en altura de la malla realizada	178
Figura III-129. Vista tridimensional de la malla	179
Figura III-130. De talle de la malla en torno a la celda circular.....	180
Figura III-131. Región plana dividida en elementos finitos [127]	181
Figura III-132. Ejemplo de curvas de contorno y mapa de vectores para un sistema 2x2	186
Figura III-133. Resolución por GC de un sistema 2x2.....	189
Figura III-134. Ligera mejora de las curvas elipsoides tras acondicionamiento diagonal	190
Figura III-135. Rutina principal en datos.f90	193
Figura III-136. Planteamiento general de obtención de esfuerzos en paredes verticales	202
Figura III-137. Planteamiento general de obtención de esfuerzos en paredes horizontales	203
Figura III-138. Diagrama UML de la clase CGargantaVertical	204
Figura III-139. Diagrama UML de la clase CColGargantasVerticales.....	204
Figura III-140. Orden de numeración de los puntos que definen las gargantas verticales	205

Figura III-141. Geometría de las gargantas horizontales interiores	205
Figura III-142. Garganta horizontal interna y puntos que la definen	206
Figura III-143. Diagrama UML de la clase CGargantaHorizontal	206
Figura III-144. Diagrama UML de la clase CColGargantasHorizontales	207
Figura III-145. Geometría de las gargantas horizontales de pared.....	207
Figura III-146. Diagrama UML de la clase CMuro	207
Figura III-147. Diagrama UML de la clase CColMuros	208
Figura III-148. Ejemplo de gargantas verticales en un cajón.....	211
Figura III-149. Ejemplo de gargantas horizontales en un cajón.....	212
Figura III-150. Equivalencia entre tensiones normales y esfuerzos.....	213
Figura III-151. Reparto de área para cada nodo en elementos cuadráticos.....	214
Figura III-152. Criterio de esfuerzos positivos	216
Figura III-153. Gráficas de esfuerzos axil y axil reducido en gargantas verticales.....	216
Figura III-154. Gráficas de momento flector y flector reducido en gargantas verticales	217
Figura III-155. Ejes locales en garganta interior	218
Figura III-156. Criterio de esfuerzos positivos	219
Figura III-157. Gráficas de esfuerzos axil y axil reducido en gargantas horizontales .	220
Figura III-158. Gráficas de momento flector y flector reducido en gargantas horizontales	221
Figura III-159. Modelo de viga para validación de algoritmos.....	221
Figura III-160. Tensiones normales en secciones de estudio	222
Figura III-161. Placa circular empotrada bajo carga uniforme.....	223
Figura III-162. Secciones de referencia para el cálculo de esfuerzos en el voladizo ...	224
Figura III-163. Variables principales de la sección.....	225
Figura III-164. Formas de trabajo de la sección y leyes lineales de deformación	226
Figura III-165. Ley de comportamiento del hormigón.....	227
Figura III-166. Ley de comportamiento del acero.....	228
Figura III-167. Dominios de deformación de la EHE	229
Figura III-168. Equilibrio de fuerzas y momentos en la sección.....	231
Figura III-169. Esquema de armado del domino 2.b.....	232
Figura III-170. Gráfica habitual de la ley de momentos flectores en paredes verticales	239
Figura III-171. Diagrama UML de la clase CArmadoHorizontal	242
Figura III-172. Diagrama UML de la clase CArmadoVertical	242
Figura III-173. Ejemplo de tabla resumen de armado horizontal por alturas y tipos de garganta	244
Figura III-174. Ejemplo de tabla resumen de armado horizontal por tramos unificados y tipos de garganta.....	244
Figura III-175. Ejemplo de tabla resumen de armado vertical por alturas y tipos de garganta	245
Figura III-176. Ejemplo de tabla resumen de armado vertical por tramos unificados y tipos de garganta.....	245
Figura III-177. Ejemplo de tabla resumen de armado longitudinal de losa y voladizos	246
Figura III-178. Ejemplo de esquema de armado en pared y celdas circulares achaflanadas del Lado Tierra.....	247
Figura III-179. Ejemplo de armado de solera y voladizos	248
Figura IV-1. Esquema global de la estructura de la aplicación.....	250
Figura IV-2. Casos de uso de la aplicación para el actor “Usuario”	253

Figura IV-3. Casos de uso de la aplicación para el actor “Administrador”	253
Figura IV-4. Gráfico de la estructura.....	256
Figura IV-5. Campos de la entidad CDatosRetorno	257
Figura IV-6. Campos de la entidad CDatosCalculo	258
Figura IV-7. Campos de la entidad CDatosStatus.....	258
Figura IV-8. Campos de la entidad CUserData	259
Figura IV-9. Función doPost.....	260
Figura IV-10. Servicio de descarga	261
Figura IV-11. Diagrama de clases del servicio de descarga.....	261
Figura IV-12. Diagrama de secuencia del servicio de descarga para la petición del tamaño del archivo	262
Figura IV-13. Diagrama de secuencia del servicio de descarga para la descarga real del archivo	263
Figura IV-14. Diagrama de actividad del servicio de descarga.....	264
Figura IV-15. Excepción en caso de error en la descarga	264
Figura IV-16. Comprobación de usuario válido.....	265
Figura IV-17. Obtención del tamaño del archivo	265
Figura IV-18. Descarga real del archivo	266
Figura IV-19. Diagrama de clases para el objeto DownloadBO	267
Figura IV-20. Cabeceras de los métodos de DownloadBO	267
Figura IV-21. Consulta SQL	268
Figura IV-22. Ejecución sincronizada de la consulta SQL	269
Figura IV-23. Registro de CicloEventos como servlet de inicio.....	269
Figura IV-24. Diagrama de herencia de la clase CicloEventos.....	270
Figura IV-25. Función init de inicio del sistema.....	271
Figura IV-26. Función start de inicio del sistema	271
Figura IV-27. Función stop de inicio del sistema.....	272
Figura IV-28. Función setDefaultValues de inicio del sistema.....	272
Figura IV-29. Ejecución sincronizada de la consulta SQL	273
Figura IV-30. Diagrama de estados durante el ciclo de eventos	274
Figura IV-31. Secuencia de procesamiento de estados	275
Figura V-1. Vista del Puerto de Barcelona y de los diques y muelles objeto de estudio	277
Figura V-2. Infografía del cajón.....	278
Figura V-3. Geometría modelada de media sección. (a) Losa, (b) Fuste.	279
Figura V-4. Malla de media sección. (a) Losa, (b) Fuste.	279
Figura V-5. Vista en 3D de la mitad simétrica del cajón.....	280
Figura V-6. Vista del desplazamiento resultante en isométrico	280
Figura V-7. Infografía del cajón.....	281
Figura V-8. Geometría modelada de media sección. (a) Losa, (b) Fuste.	282
Figura V-9. Vista en 3D de la mitad simétrica del cajón	282
Figura V-10. Malla de media sección. (a) Losa, (b) Fuste	282
Figura V-11. Vista del desplazamiento resultante en isométrico	283
Figura V-12. Esquema de armado horizontal del fuste	283
Figura V-13. Infografía del cajón.....	284
Figura V-14. Geometría modelada. (a) Losa, (b) Fuste.....	285
Figura V-15. Vista en 3D del cajón.....	285
Figura V-16. Malla utilizada. (a) Losa, (b) Fuste.....	286
Figura V-17. Esquema de armado de las Celdas Circulares Achaflanadas de Esquina en Pared Tierra	287

Figura V-18. Infografía del cajón.....	288
Figura V-19. Geometría modelada. (a) Losa, (b) Fuste.....	289
Figura V-20. Vista en 3D del cajón.....	289
Figura V-21. Malla utilizada. (a) Losa, (b) Fuste.....	290
Figura V-22. Vista del desplazamiento resultante en isométrico.....	290
Figura V-23. Esquema de armado de las Celdas Ovoidales en esquina Lado Mar.....	291
Figura V-24. Infografía del cajón.....	292
Figura V-25. Geometría modelada. (a) Losa, (b) Fuste.....	293
Figura V-26. Vista en 3D del cajón.....	293
Figura V-27. Malla utilizada. (a) Losa, (b) Fuste.....	294
Figura V-28. Vista del desplazamiento resultante en isométrico.....	294
Figura V-29. Esquema de armado de la Pared Lateral.....	295
Figura V-30. Infografía del cajón.....	296
Figura V-31. Geometría modelada. (a) Losa, (b) Fuste.....	297
Figura V-32. Vista en 3D del cajón.....	297
Figura V-33. Malla utilizada. (a) Losa, (b) Fuste.....	298
Figura V-34. Vista del desplazamiento resultante en isométrico.....	298
Figura V-35. Esquema de armado de la Pared Lado Mar.....	299
Figura V-36. Infografía del cajón.....	300
Figura V-37. Geometría modelada. (a) Losa, (b) Fuste.....	301
Figura V-38. Vista en 3D del cajón. Alterantiva 1.....	301
Figura V-39. Vista en 3D del cajón. Alterantiva 2.....	302
Figura V-40. Vista en 3D de la malla – alternativa 1.....	302
Figura V-41. Vista en 3D de la malla – alternativa 2.....	303
Figura V-42. Vista del desplazamiento resultante en isométrico – alternativa 1.....	303
Figura V-43. Vista del desplazamiento resultante en isométrico – alternativa 2.....	304
Figura V-44. Vista del muelle de cajones fondeados.....	304
Figura V-45. Vista actual del muelle en servicio.....	305
Figura V-46. Vista del Puerto de Bilbao y de los muelles objeto de estudio.....	305
Figura V-47. Infografía del cajón.....	306
Figura V-48. Geometría modelada. (a) Losa, (b) Fuste.....	307
Figura V-49. Vista en 3D del cajón.....	308
Figura V-50. Malla utilizada. (a) Losa, (b) Fuste.....	309
Figura V-51. Esquema de armado de las Celdas Ovoidales Esquina.....	310
Figura V-52. Vista de cajones fondeados del muelle AZ2.....	311
Figura V-53. Infografía del cajón.....	311
Figura V-54. Geometría modelada. (a) Losa, (b) Fuste.....	312
Figura V-55. Vista en 3D del cajón.....	313
Figura V-56. Malla utilizada. (a) Losa, (b) Fuste.....	313
Figura V-57. Vista del desplazamiento resultante en isométrico.....	314
Figura V-58. Esquema de armado de las Celdas Ovoidales Esquina.....	314

Índice de tablas

Tabla III-1. Número de celdas por tipología en la sección del cajón	72
Tabla III-2. Área aligerada de cada tipología de celdas	73
Tabla III-3. Centros de gravedad de los aligeramientos por tipología de celdas respecto de su punto de referencia	75
Tabla III-4. Momentos de inercia de los aligeramientos por tipología de celdas respecto de su centro de gravedad	76
Tabla III-5. Casos de carga más comunes a estudiar	140
Tabla III-6. Posibles combinaciones de sobrecargas y sus coeficientes de combinación	153
Tabla III-7. Coeficientes de mayoración de cargas	154
Tabla III-8. Ordenación de entidades en fichero de geometría en GiD	165
Tabla III-9. Ejemplo de fichero de geometría de un cajón	168
Tabla III-10. Orden de generación de mallado en línea de comandos	176
Tabla III-11. Fichero script para la obtención de datos de malla	176
Tabla III-12. Fichero con los datos de malla	177
Tabla III-13. Llamada genérica a la librería MatMan	191
Tabla III-14. Orden de generación de mallado en línea de comandos	193
Tabla III-15. Ejemplo de fichero *.sal para resolución en Calsef	194
Tabla III-16. Ejemplo de fichero *.sal tras resolución en Calsef	198
Tabla III-17. Ejemplo de fichero *.flavia.res tras resolución en Calsef	198
Tabla III-18. Fichero por lotes para generación de imágenes en post-proceso en GiD	199
Tabla III-19. Orden de generación de imágenes en línea de comandos	199
Tabla III-20. Resultado de tensiones normales en el cajón en una hipótesis de cálculo	200
Tabla III-21. Colección de imágenes de postproceso generadas automáticamente en GiD	201
Tabla III-22. Diagrama UML de la clase CSalidaGarg	208
Tabla III-23. Fichero con gargantas verticales	209
Tabla III-24. Fichero con gargantas horizontales	210
Tabla III-25. Fichero de esfuerzos horizontales en gargantas verticales	215
Tabla III-26. Fichero de esfuerzos verticales en gargantas horizontales	219
Tabla III-27. Comparativa de resultados según algoritmo propuesto	222
Tabla III-28. Límites de los dominios	229
Tabla III-29. Deformaciones y tensiones de armaduras y hormigón	235
Tabla III-30. Fuerzas y momentos de las armaduras y del hormigón para cada dominio	236
Tabla III-31. Fichero de armado horizontal de gargantas verticales	243
Tabla III-32. Fichero de armado vertical de gargantas horizontales	243

Capítulo I. Introducción y Objetivos

1.1.- Introducción

En este trabajo se describe un modelo completo para el diseño y cálculo resistente de cajones multicelulares portuarios, comprendiendo todas las etapas de diseño de geometría e imposición de condiciones de contorno, generación de geometría multibloque para mallado estructurado automático, resolución por elementos finitos, determinación de esfuerzos y definición de las armaduras necesarias.

En el Capítulo 3 comprende seis grandes apartados. En el primero se recoge de forma detallada y profusa toda la formulación explícita para la definición geométrica del cajón, así como para los cálculos de estabilidad naval.

El segundo se centra en la descripción completa del algoritmo desarrollado específicamente para convertir el cajón en una geometría multibloque, con las siguientes ventajas:

- La geometría resultante sea fácilmente mallable de forma estructurada por un mallador comercial.
- Se disponga de la información de la geometría de forma que sea automatizable la aplicación de condiciones de contorno para cada hipótesis de cálculo que se quiera resolver.
- Se dispongan suficientes nodos específicamente ubicados en las zonas críticas para el posterior armado.

En el tercer apartado se describen de forma genérica las situaciones más habituales en el cálculo de cajones para muelles o diques.

En el cuarto se presenta el esquema de resolución mediante elementos finitos de una hipótesis de cálculo, describiendo la forma en que se gestiona la información de la geometría y las condiciones de contorno para generar de forma automática un cajón en un pre-procesador comercial, realizar una malla estructurada y utilizar un *solver* específico para resolver un sistema lineal cercano al millón de ecuaciones.

Resuelto el sistema y obtenidas las tensiones en los nodos, en el apartado quinto se presenta un algoritmo de integración en las secciones críticas para la generación automática de leyes de esfuerzos que permitan realizar el armado horizontal y vertical del cajón, lo que se hará en el sexto apartado.

En paralelo, el Capítulo 4 introduce el framework o marco de trabajo realizado para implementar informáticamente la formulación presentada y a los algoritmos descritos.

En el Capítulo 5 se presentan los diversos trabajos profesionales realizados por el doctorando en colaboración con el Centro Internacional de Métodos Numéricos en Ingeniería de la Universidad Politécnica de Cataluña, el propio Director de la Tesis y las Autoridades Portuarias de Bilbao y Barcelona, como fruto de la investigación y trabajo realizado en la generación de la formulación, de los algoritmos descritos en el apartado 3, y de una herramienta informática desarrollada específicamente para satisfacer todas y cada una de las etapas del proceso.

La aplicación se ha desarrollado para la realización de todas y cada una de las etapas involucradas, automatizando aquellos procesos que no requieran la intervención del usuario. El código está concebido para asistir al proyectista en el establecimiento de la geometría y de las hipótesis de carga a considerar, incluidas las correspondientes a las etapas de construcción, transporte y hundimiento, además de las de servicio.

Finalizada esta primera fase, el usuario remite el diseño vía Internet a un servidor remoto que se encarga de manera completamente automatizada del cálculo resistente, de la determinación de cuantías y de la homogeneización de la solución de armado. Se consigue así trasladar al servidor los altos requerimientos en cuanto a coste computacional asociados a la etapa de cálculo y reducir al máximo las necesidades de un usuario cualificado para la realización de estas etapas.

No obstante, no es objetivo de esta tesis describir la aplicación informática, por lo que no se entrará a detallar su funcionamiento. Por ello, el Capítulo 4 presentará de forma genérica la descripción del marco de trabajo, atendiendo a la organización multicapa y a los servicios desarrollados, abstrayendo en la medida de lo posible las características de los mismos, con independencia de su aplicación a los cajones multicelulares.

1.2.- Objetivos

En el desarrollo de la tesis doctoral que ahora se presenta se han marcado los siguientes objetivos fundamentales, en relación a los cajones multicelulares portuarios.

1. Desarrollar un tratamiento singular y unificado mediante un enfoque tridimensional a través del análisis automatizado por elementos finitos para el diseño, cálculo y armado de cajones multicelulares portuarios.
2. Desarrollo de una formulación explícita de las ecuaciones de geometría, centros de gravedad e inercia de flotación para cualquier variable que determina la sección transversal del cajón y la estabilidad en flotación.
3. Desarrollar un algoritmo automático de creación de geometría multibloque para su inmediato mallado tridimensional estructurado con elementos hexaédricos.
4. Desarrollar un algoritmo de ubicación de secciones críticas, denominadas gargantas, en paralelo con la generación de la geometría multibloque, de manera que en el mallado se ubiquen suficientes de nodos en estas secciones.
5. Desarrollar un algoritmo de aplicación de condiciones de contorno en cargas en las distintas secciones del cajón
6. Desarrollar un algoritmo de integración de las tensiones obtenidas en los nodos ubicados en las gargantas para la obtención de los esfuerzos para el armado de estas secciones.

En cuanto al marco de trabajo desarrollado, los objetivos que se han perseguido son los siguientes:

7. Desarrollo de software único de integración de las etapas de pre-proceso, cálculo y post-proceso de un cajón multicelular, en versión standalone, o cliente-servidor.
8. Desarrollo de un entorno profesional de gestión mediante la implementación de servicios cliente/servidor: registro y comprobación de licencias, distribución de las hipótesis de cálculo y su correspondiente resolución por elementos finitos a diversos ordenadores para su cálculo y posterior armado tras integración de las tensiones obtenidas, la posibilidad de obtención de imágenes de los resultados generados, la descarga de ficheros, la comprobación periódica del estado del proceso y el aviso al usuario, etc.
9. Extensión del Framework cliente-servidor a otros entornos.

1.3.- Organización del Documento

La presente Tesis se organiza de la siguiente manera.

En el Capítulo 2 se realiza una amplia descripción del Estado del Arte actual. Por un lado, se presenta la situación relativa a los cajones como elemento constructivo de gran actualidad en el ámbito portuario y, por otro, se presentarán alguno de los frameworks de software más conocidos, y sus principales características y virtudes.

En el Capítulo 3 se presenta una completa descripción de las fases de diseño, cálculo por elementos finitos y armado de un cajón multicelular, atendiendo a las tres etapas básicas de los procesos de cálculo: pre-procesado, análisis y post-procesado.

En el Capítulo 4 se presenta el framework objeto de esta tesis doctoral, del cual la aplicación de cajones es el primer producto.

En el Capítulo 5 se presentan los proyectos realizados como producto final resultante de la armonización de las dos grandes líneas presentadas.

En el Capítulo 6 se presentan las conclusiones y las líneas futuras.

El documento termina con las referencias consultadas por el autor.

Capítulo II. Estado del Arte

II.1.- Introducción a los cajones portuarios

España es uno de los países con más experiencia en la utilización de cajones aligerados y uno de los más avanzados en el desarrollo de la tecnología necesaria para su fabricación, lo que se explica en parte por su extenso litoral (8000 km) y de gran calado (a partir de los 12 m), por su gran número de puertos y ubicación estratégica en las rutas internacionales de transporte.

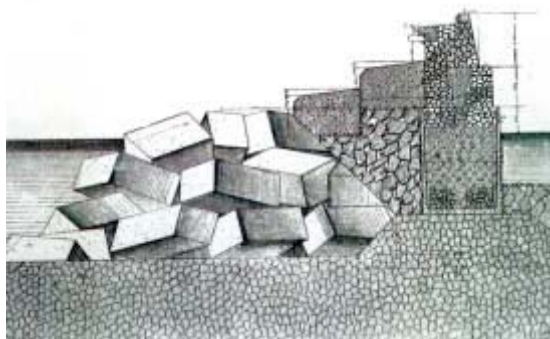


Figura II-1. Perfil constituido por bloques artificiales de hormigón del dique del puerto de Barcelona a finales del XIX. De 'El puerto de Barcelona', de José Ayxelá [1]

En paralelo con el continuo crecimiento y evolución de los puertos españoles, que pasan a ser entes autónomos, el transporte comercial marítimo gana peso al terrestre y aéreo, lo que obliga a una continua expansión de la zona portuaria, ganando terreno al mar. En estas actuaciones de ampliación y mejora el *dique* es la obra por antonomasia, bien sea para espigones de protección o abrigo o bien para líneas de atraque, muelles, etc.

Conforme a la clasificación de Takahashi [9], existen dos tipologías fundamentales de diques de abrigo: los *diques en talud*, o diques rompeolas, con el dique de escollera como principal referente (**Figura II-2.izquierda**), cuyo diseño original se basa en romper el oleaje, y *los diques verticales*, cuya misión principal es reflejar el oleaje. Éstos se pueden subdividir en monolíticos o de bloques (**Figura II-2.centro**) y en compuestos, con los diques de cajones (**Figura II-2.derecha**) como principal referencia.

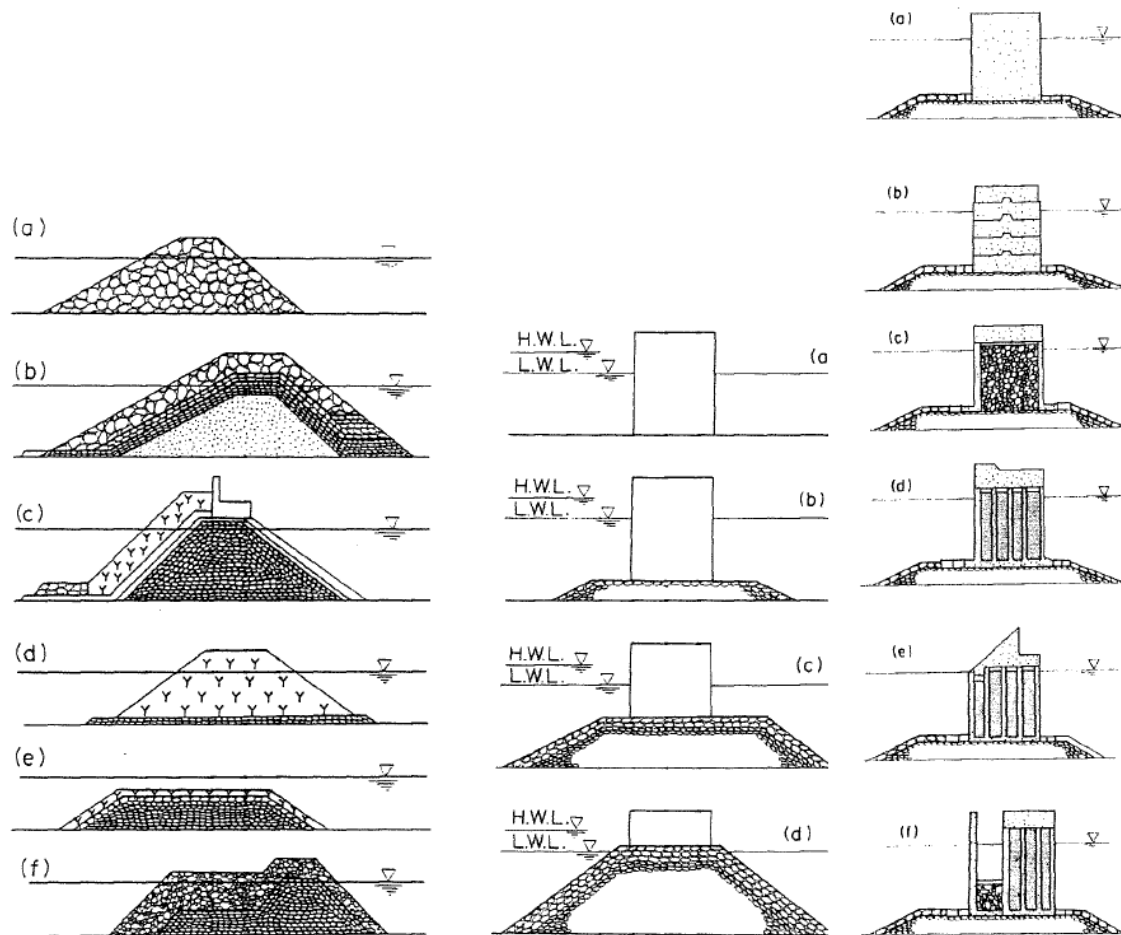


Figura II-2. Tipología de diques de abrigo: en talud (izquierda), verticales de bloques o monolíticos (centro) y verticales compuestos (derecha) [9]

Los cajones portuarios suponen una alternativa rápida y práctica a la construcción de los diques de escollera. Se diseñan elementos prefabricados de gran volumen y de relativamente poco peso en relación a éste y, a partir de una explanada mejorada, se llegan a cubrir distancias de uno a dos kilómetros de dique en cuestión de semanas en calados profundos. La producción de los cajones se regulariza y se consiguen plazos de fabricación de un cajón por semana, gracias a la prefabricación en diques flotantes o pontonas, **Figura II-3**.

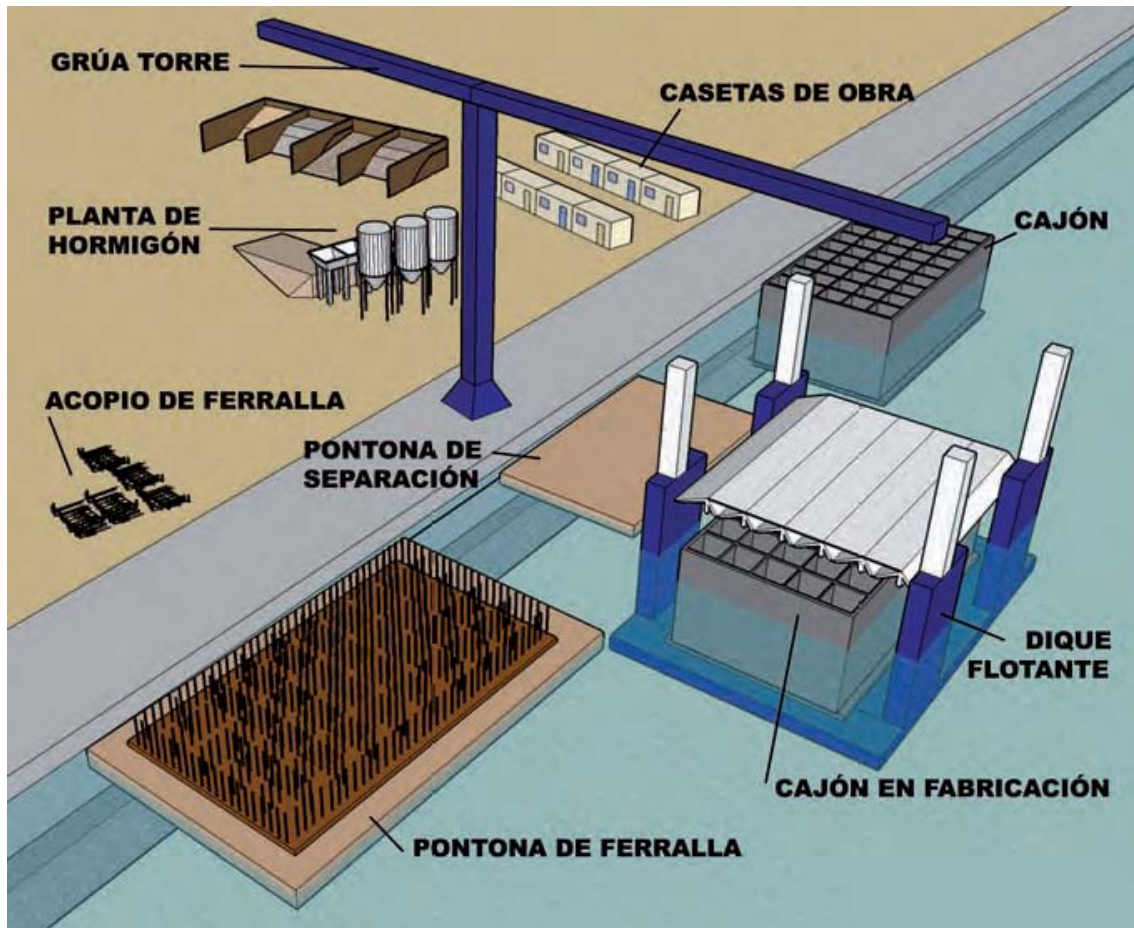


Figura II-3. Infraestructura para fabricar cajones [6]

Las primeras obras de cajones que se construyeron en España fueron el muelle de Levante del puerto de Huelva, en la ría del Odiel, que entró en servicio en 1932 con 8 metros de calado máximo; y el dique-muelle del Sagrado Corazón en Tarifa, que entró en servicio en 1945 con 10 m de calado máximo. En las décadas posteriores se extendió la técnica de fabricación de cajones a numerosas obras de atraque en los puertos de Pasajes, Avilés, Gijón, Cádiz, Cartagena [5].

Hoy en día, las 28 Autoridades Portuarias existentes en España [7] cuentan con alguna obra significativa realizada con cajones flotantes. De hecho, se tienen contabilizados más de 100 km de obras de cajones [8].



Figura II-4. Prolongación del dique de Poniente en el puerto de Motril [3]



Muchas de estas actuaciones son conducentes a mejorar las condiciones de atraque y amarre de buques (Figura II-5), por lo que en la redacción del proyecto se deben asegurar las condiciones adecuadas de permanencia en el puerto y de las operaciones de carga y descarga. En general se estudian por separado la cimentación, la estructura, la superestructura, el relleno y los elementos de uso y explotación (vigas carriles, defensas, puntos de amarre, etc.), Figura II-6.

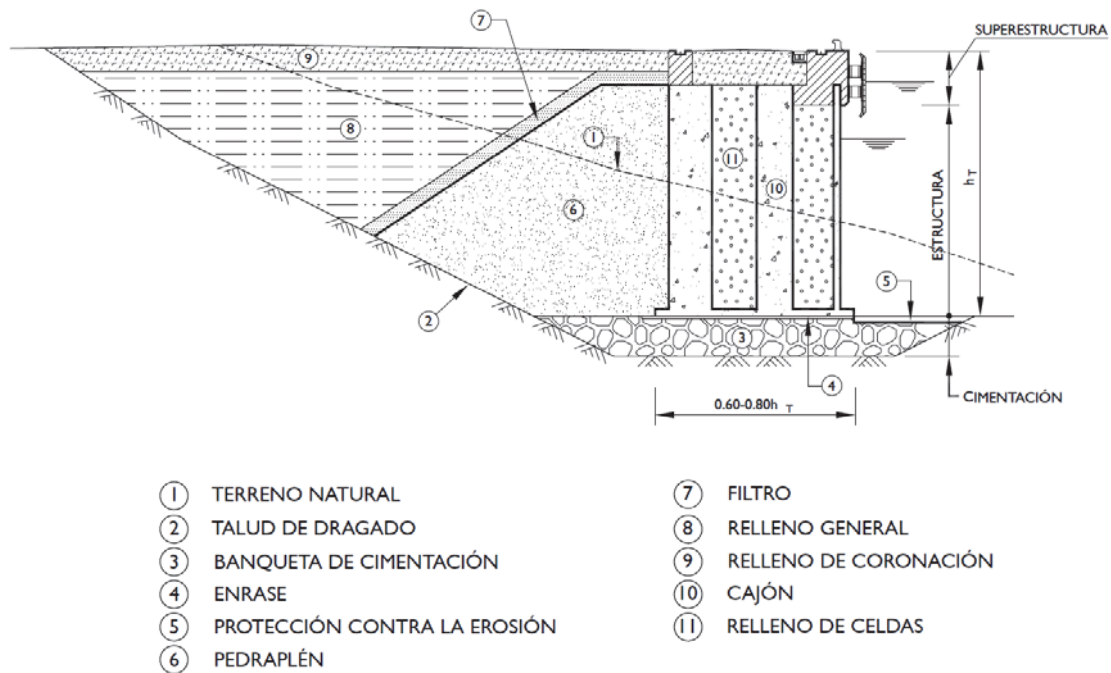


Figura II-6. Obra de atraque de cajón flotante [11]

España no es el único referente mundial. Países con gran extensión de litoral como Japón [12], [24] o Italia [25] han sido pioneros en el uso de cajones multicelulares en sus construcciones portuarias.

Dentro de los diques verticales construidos con cajones, existen varias tipologías estructurales, en relación al tipo de aligeramiento empleado en el cajón flotante, que puede ser de sección rectangular o circular, Figura II-7.



Figura II-7. Ejemplos de tipologías estructurales según aligeramiento rectangular en el puerto de Cartagena (izquierda, [26]) o circular, en el puerto de Santander (derecha)

Si bien el diseño de cajones más extendido es el de planta rectangular, existen otras tipologías en las que se introducen soluciones innovadoras, como cajones hincados, cajones circulares monocelulares con patio, cajones rectangulares con patio interior y cámaras delanteras perforadas, cajones flotantes anclados, etc. [8].

No obstante, en este trabajo se estudiará en profundidad exclusivamente el cajón de planta rectangular y de aligeramiento circular.

Acciones

Como se ha mencionado, los cajones son diques verticales que reflejan el oleaje incidente. Es por ello que además de las cargas permanentes y variables asociadas a la estructura y las presiones hidrostáticas debidas al nivel del mar, uno de los criterios de diseño más importante es la presión que dicho oleaje ejerce sobre la pared vertical del cajón. Aunque no forma parte de esta tesis el estudio del oleaje y la determinación de la ley de presiones, ya que será dato aportado por la Autoridad Portuaria para el análisis estructural del cajón, se presentan a continuación algunos de los modelos empleados históricamente [8], **¡Error! No se encuentra el origen de la referencia.**

Hiroi [13] en 1919 habla de una distribución homogénea de presiones a partir de la altura H de ola de diseño, con validez para olas rompientes en condiciones de aguas someras, **Figura II-8.a.** Sainflou [14] en 1928 introduce una fórmula aplicada a ondas estacionarias que se reflejan sin romperse en la pared vertical, dando lugar a una ley trocoidal en función de H y para paso de cresta y seno, válida para calados profundos, **Figura II-8.b.** Goda [15] desarrolla en 1974 una nueva metodología, proponiendo un diagrama trapezoidal que se adapte tanto a olas rompientes como estacionarias. En 1985 considera nuevos parámetros en su formulación [12].

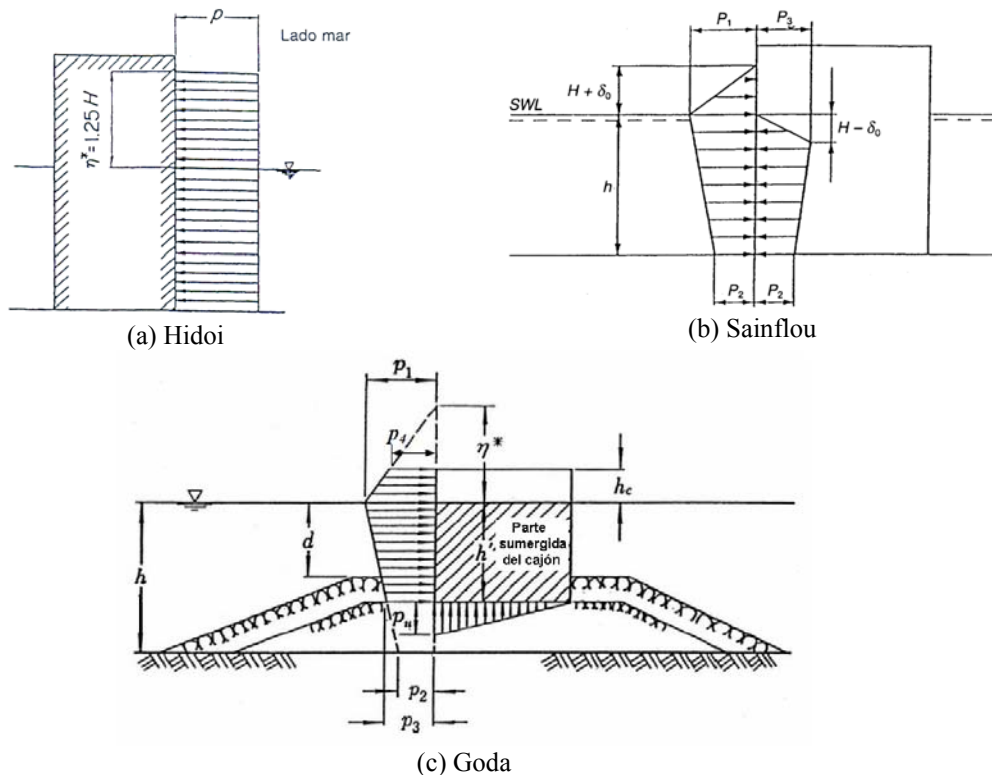


Figura II-8. Diferentes leyes de presiones de oleaje sobre el paramento vertical del cajón,

Cálculo

Tradicionalmente se han empleado técnicas simplificadas para la modelización de cajones flotantes mediante estados planos ([5], [2]), si bien es posible analizar el comportamiento resistente del cajón ante diversas sollicitaciones de forma tridimensional mediante el método de los elementos finitos con técnicas de análisis lineal e incluso con técnicas no lineales [16].

Existen diversos programas comerciales [17] para el cálculo de estabilidad por equilibrio límite y por métodos numéricos para estudiar los modos de fallo correspondientes a deslizamiento profundo, hundimiento y vuelco plástico, en estructuras lineales tales como muelles o diques, que son tratadas de manera bidimensional [18].

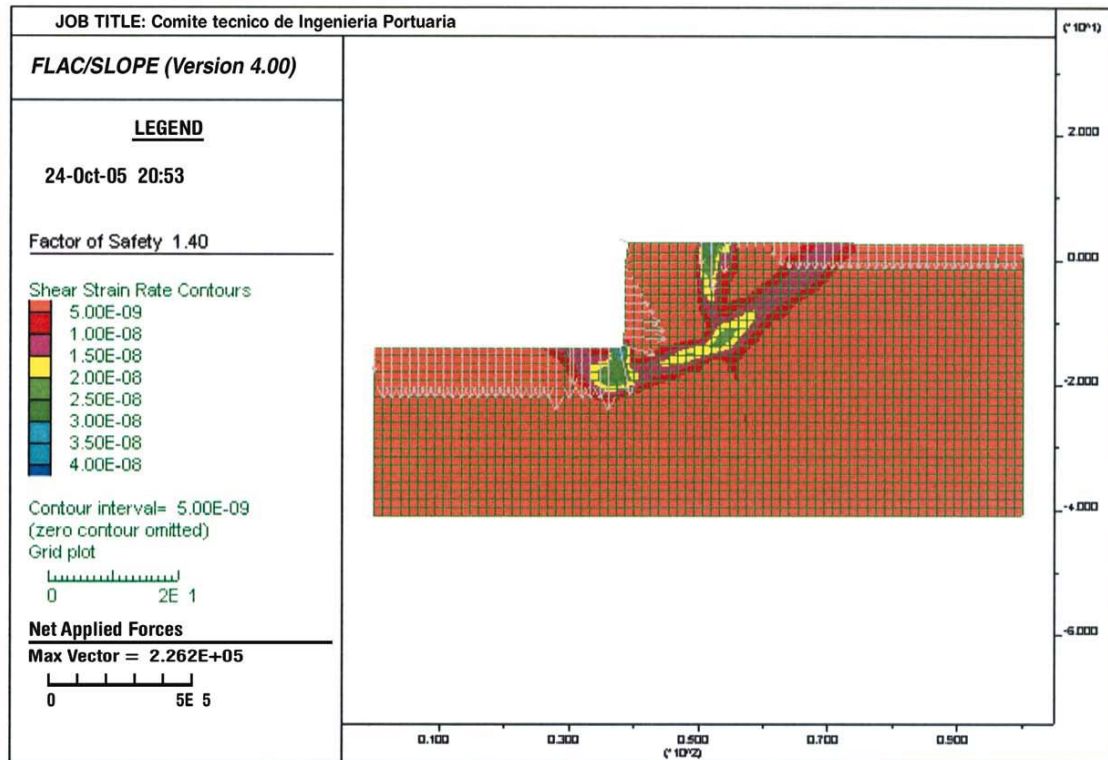


Figura II-9. Mezcla de modos de fallo en un muelle [17].

La cimentación de estas estructuras marinas se realiza en el lecho marino por lo que la determinación de las características portantes del terreno así como su complejo comportamiento estático y dinámico en las situaciones de carga a las que se ve sometido es otro problema esencial en el proyecto.

Los suelos existentes en el ámbito portuario suelen presentar resistencia a corto plazo muy baja y ser muy compresibles, además de presentar posibilidades de drenaje muy lentas, lo que conduce a una generación muy lenta de asentos, bajo carga constante. En [19] se analiza el comportamiento estático de diques verticales apoyados en suelos finos blandos sometidos a la acción del oleaje. En [20] se desarrolla un modelo teórico-numérico que permita analizar el comportamiento dinámico de la cimentación de un dique ante la acción cíclica del oleaje, en cajones fondeados en terrenos arcillosos.

Métodos como el de Método de los Elementos Finitos y Partículas se han empleado en problemas de interacción fluido-estructura de gran complejidad por la no linealidad geométrica y mecánica [21], [22].

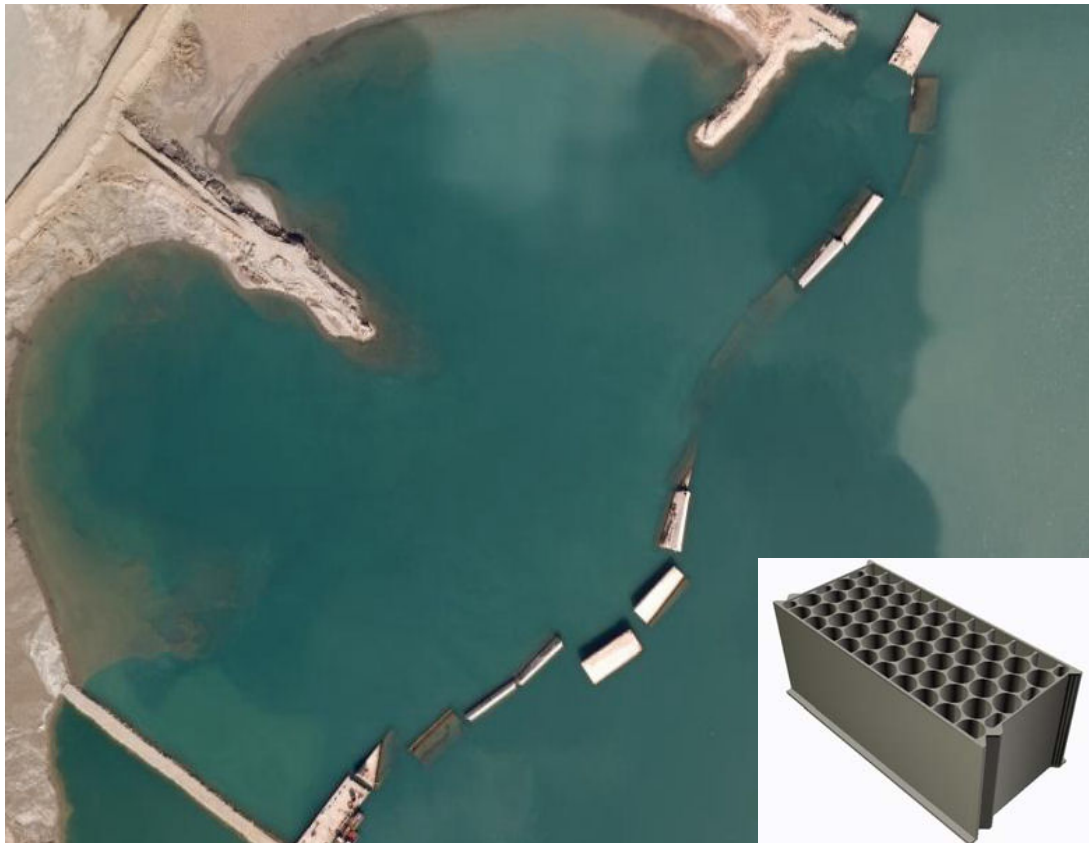


Figura II-10. Desplazamiento de 16 de los 37 cajones del Muelle del Prat del Puerto de Barcelona por licuefacción del terreno [22], enero 2007. Se tuvieron que demoler 5 y se pudieron reflotar los 11 restantes.
Barcelona, Muelle del Prat Fase I-A, 2005



Figura II-11. Operaciones de demolición de cajones dañados [23]

II.2.- Introducción a los frameworks

Según la definición del IEEE, software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a las operaciones de un sistema informático [27]. Un proceso de software es

el conjunto de actividades, métodos y prácticas empleadas en la producción y evolución del software [28].

La Ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de ingeniería al software [27]. Es la rama de la ingeniería que aplica los principios de la ciencia informática y las matemáticas para lograr soluciones económicas a los problemas de desarrollo de software.

Un proceso de ingeniería de software es un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo: la obtención de un producto de software de calidad [29]. Se define también como el conjunto de actividades de ingeniería de software necesarias para transformar los requisitos del usuario en software [28]. El proceso de desarrollo de software es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo [29].

A este proceso también se le llama ciclo de vida del software, una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software [30]. Comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios [31].

La ingeniería de dominio surge como una evolución de la reutilización sistemática del software basada en modelos donde las arquitecturas software juegan un papel ampliamente destacado. *La reusabilidad debe ser una característica inherente al software*. La reutilización de software que no ha sido desarrollado para su reutilización es técnicamente más difícil y costosa que la reutilización del software desarrollado para ese fin [32].

La ingeniería de dominio se puede definir como el proceso clave que se necesita para el diseño sistemático de una arquitectura y de un conjunto de elementos software reutilizables que pueden ser usados en la construcción de una familia de aplicaciones relacionadas o subsistemas.

Un aspecto crítico a la hora de desarrollar sistemas de software complejos es el diseño de su arquitectura, representada como un conjunto de elementos computacionales y de datos interconectados de una cierta manera [33].

La Arquitectura del Software ha surgido recientemente como un campo específico de estudio para los investigadores e ingenieros del software. Fenómenos como el creciente uso de la Web como medio para la comunicación y el comercio global demandan el desarrollo de nuevas tecnologías para la construcción de aplicaciones abiertas y distribuidas. Estas aplicaciones tienen una estructura compleja y dinámica, formada por la interconexión de una gran cantidad de componentes.

La importancia de realizar de forma explícita representaciones arquitectónicas de los sistemas software es doble. En primer lugar, estas representaciones elevan el nivel de abstracción, facilitando la comprensión de sistemas de software complejos. En segundo lugar, aumentan las posibilidades de reutilizar tanto las arquitecturas como los componentes software que aparecen en ellas.

Un ejemplo de nivel de abstracción de una arquitectura de software lo constituyen los *frameworks*. La palabra anglosajona *framework*, cuya traducción literal sería *marco de trabajo*, define, en términos generales, un conjunto estandarizado de *conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar*.

Los frameworks son *generadores de aplicaciones* que se relacionan directamente con un dominio específico, con una familia de problemas relacionados [34]. Es *un patrón arquitectónico que proporciona una plantilla extensible para aplicaciones dentro de un dominio* [35]. Un Marco de Trabajo o Framework es *un diseño reutilizable de todo o parte de un sistema, representado por un conjunto de clases abstractas y la forma en la que sus instancias interactúan* [36].

En concreto, en el campo del desarrollo de software, un framework o infraestructura digital, es una *estructura conceptual y tecnológica de soporte definido*, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

Representa una *arquitectura de software* que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

En los últimos años se ha extendido el uso de frameworks comerciales o de código libre en el campo del desarrollo de software, con el fin de que diseñadores y programadores se centren en los requisitos específicos del software en lugar de en el desarrollo tedioso y repetitivo de crear desde cero un sistema funcional y robusto en cada nuevo proyecto que se deba afrontar.

Los *patrones de diseño* son otro enfoque útil tanto para diseñar una arquitectura marco como para documentar el diseño realizado. Un patrón de diseño es una pequeña arquitectura orientada a solucionar problemas muy concretos dentro de la arquitectura general de un sistema. Existe una lista bastante extensa con todos los patrones que se han desarrollado hasta el momento, con una documentación que describe las condiciones bajo las cuales se recomienda su implantación para obtener resultados efectivos.

Los patrones de diseño aparecen tras la tesis de E. Gamma publicados en un libro [47] que recoge no sólo su definición, sino todo un catálogo de patrones aplicables a la programación básica orientada a objetos, y que cubren la mayor parte de los pequeños problemas recurrentes que se plantean en el desarrollo de cualquier programa.

Los patrones de diseño y los MTs son disciplinas complementarias en el sentido de que la arquitectura marco puede contener uno o varios patrones de diseño.

II.2.1.- Framework con arquitectura multiplataforma

Es habitual que los frameworks para aplicaciones de negocios estén diseñados mediante una arquitectura multi-plataforma de múltiples capas o *Multi-tier*, es decir, un esquema de modelado Cliente/Servidor donde hay varias capas de software con las responsabilidades bien delimitadas.

El Cliente, o capa *Front-end*, es *ligero*: en una aplicación diseñada con el modelo Cliente/Servidor la mayor parte del procesamiento se ejecuta en el Servidor y se requieren recursos mínimos en la computadora del Cliente. En el Servidor es posible repartir de manera equilibrada la carga entre varias máquinas que componen la capa de negocio, *Business-Tier*.

Así, la capa de negocios es la que se encarga de llevar a cabo el trabajo dentro del framework, mediante una arquitectura distribuida que permite la distribución de la carga entre diferentes máquinas. Los requerimientos similares que ocurren en gran cantidad son distribuidos entre diferentes servers, logrando gran eficiencia en la atención de cada uno. Los procesos que exigirían un equipo de grandes prestaciones para ser ejecutados en forma tradicional, pueden ser divididos y realizados en paralelo por varias máquinas más pequeñas, logrando una eficiencia muy elevada y una baja de costos.

Finalmente, la capa de datos o *Back-end*, es la encargada de dar persistencia a los datos sobre los cuales trabaja el sistema.

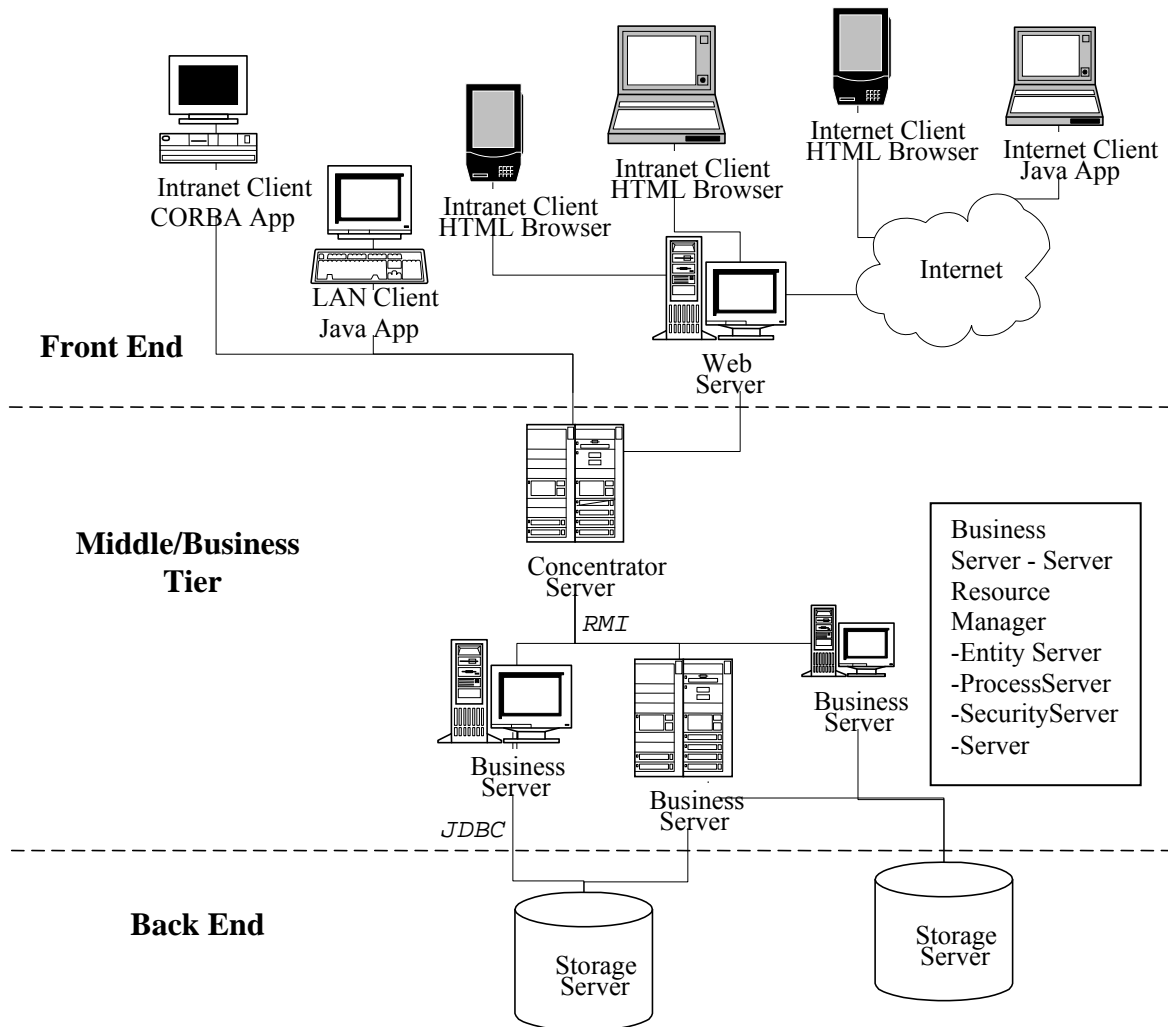


Figura II-12. Contexto de aplicación de arquitectura multicapa

II.2.2.- El nacimiento de la Web

El diseño de la World Wide Web, o simplemente la Web, que en 1989 realizó Tim Berners-Lee en la Organización Europea para la Investigación Nuclear CERN [38] no era dinámico, ya que el contenido HTML era almacenado y publicado desde servidores web. Inicialmente fue concebida como un medio de compartir la información entre los científicos de universidades e institutos de todo el mundo. Así, en 1990 se describía un proyecto de hipertexto llamado “WorldWideWeb” en el cual una web de documentos de hipertexto se podían visualizar en navegadores del cliente [39], accediendo a los servidores que albergan los datos a través de alguna red.

El HTML o Lenguaje de Marcado de Hipertexto, creado por Tim Berners-Lee, es el lenguaje de marcado para páginas web, basado en el etiquetado de SGML, Estándar de Lenguaje de Marcado Generalizado, ISO 8879:1986, y ENQUIRE, un programa de hipertexto que él mismo creó en 1980. El hipertexto es un texto mostrado en un dispositivo electrónico con un hipervínculo o enlace, que permite acceder directamente a textos relacionados. En 1991 en el documento HTML Tags describe los 18 elementos que definen el diseño inicial de HTML.

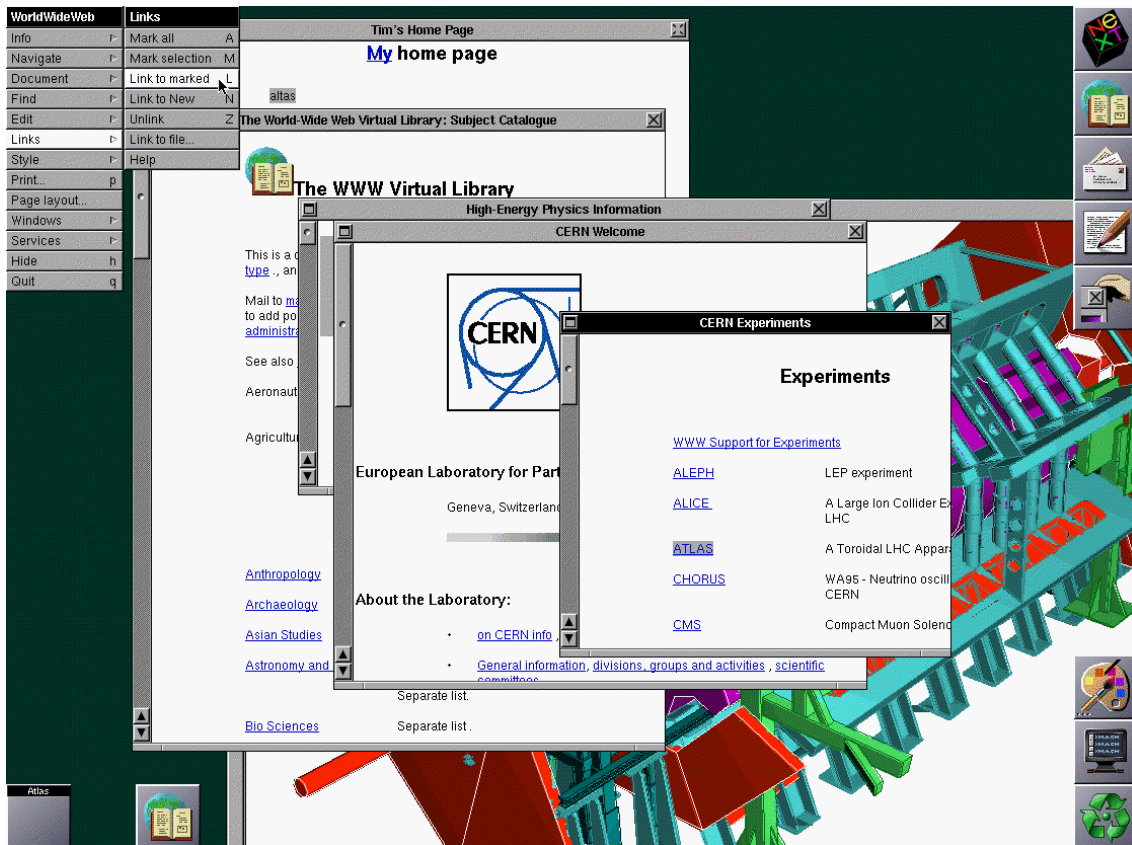


Figura II-13. Captura de pantalla del primer navegador, el WorldWideWeb, tomada en 1993 [40]

La primera página web fue <http://info.cern.ch/hypertext/WWW/TheProject.html>, el primer portal web y servidor web fue Info.cern.ch [<http://info.cern.ch/>]. Se desarrolló en un ordenador NeXT. Se creó el primer navegador web, llamado WorldWideWeb y más adelante Nexus, y servía a la vez de editor HTML. Corría sólo en máquinas NeXT. En 1991 surgieron los primeros servidores en Europa. En 1992 había 26 servidores en todo el mundo, 200 en 1993.

Para portarlo a otros entornos, como el X Windows, en CERN Tim Berners-Lee y Nicola Pellow crearon en 1991 el Line Mode Browser, que sólo mostraba texto. Sería reemplazado en 1992 por Lynx, el navegador en modo línea y texto plano, sin gráficos, que corría en cualquier ordenador, de la Universidad de Kansas, desarrollado en 1992 por estudiantes para distribuir información del campus [41].

En 1993 el National Center for Supercomputing Applications NCSA de la Universidad de Illinois, a través de Marc L. Andreessen y Eric J. Bina, lanzó la primera versión del navegador Mosaic [42][41], para sistemas X-Windows, y poco después PC y Macintosh. Fue el primer navegador web gráfico para estos entornos. Daría lugar al nacimiento de Netscape en 1994, co-fundado por el propio Marc L. Andreessen como Mosaic Communications Corporation, y en 1995 de Internet Explorer a través de Spyglass, compañía de Internet saliente de la Universidad de Illinois que desarrolló su propio código base de Mosaic.

2. Estado del Arte

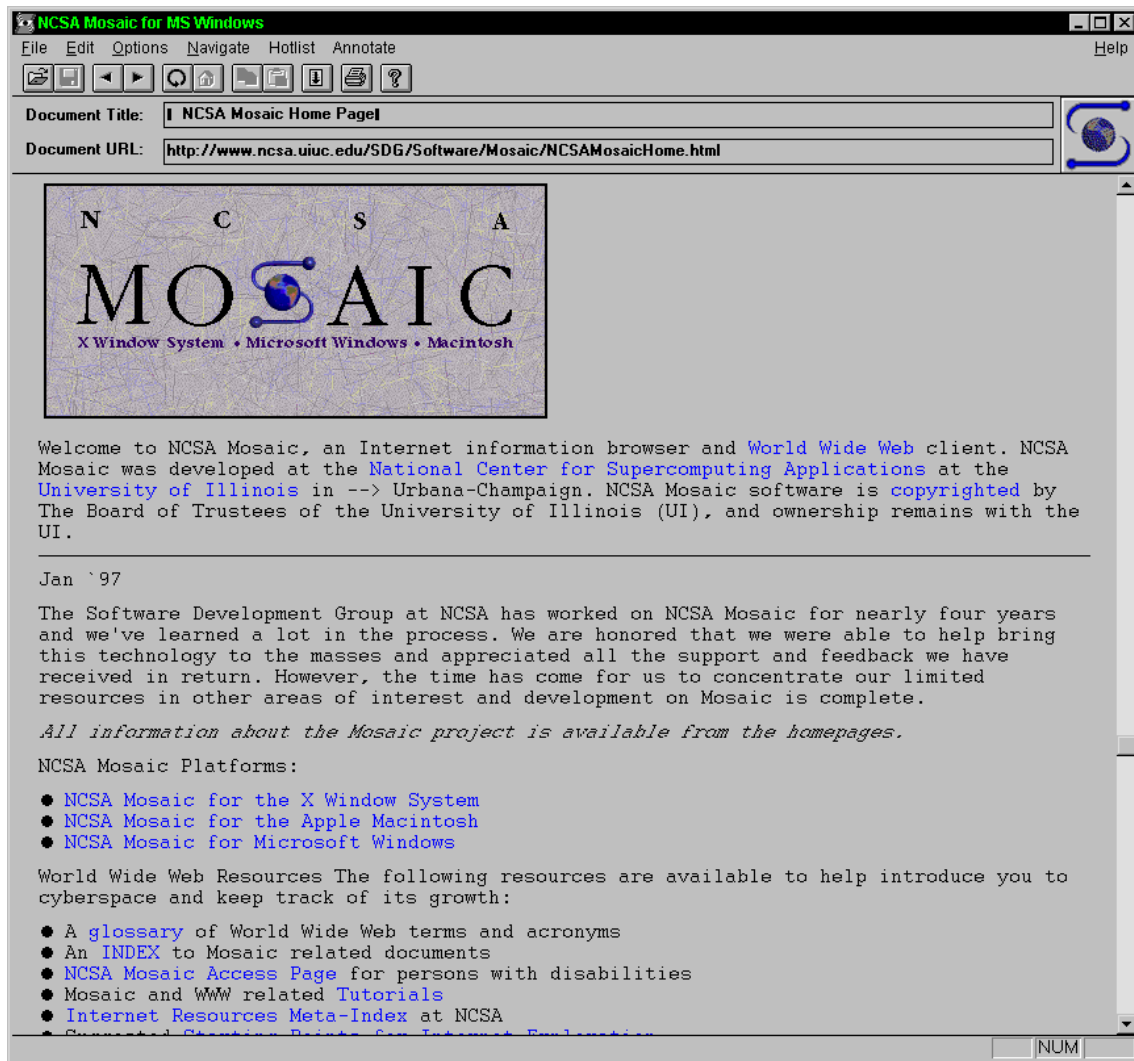


Figura II-14. Página web en Mosaic [43]

En 1994 se fundó el World Wide Web Consortium W3C para promocionar la interoperatividad entre tecnologías web.

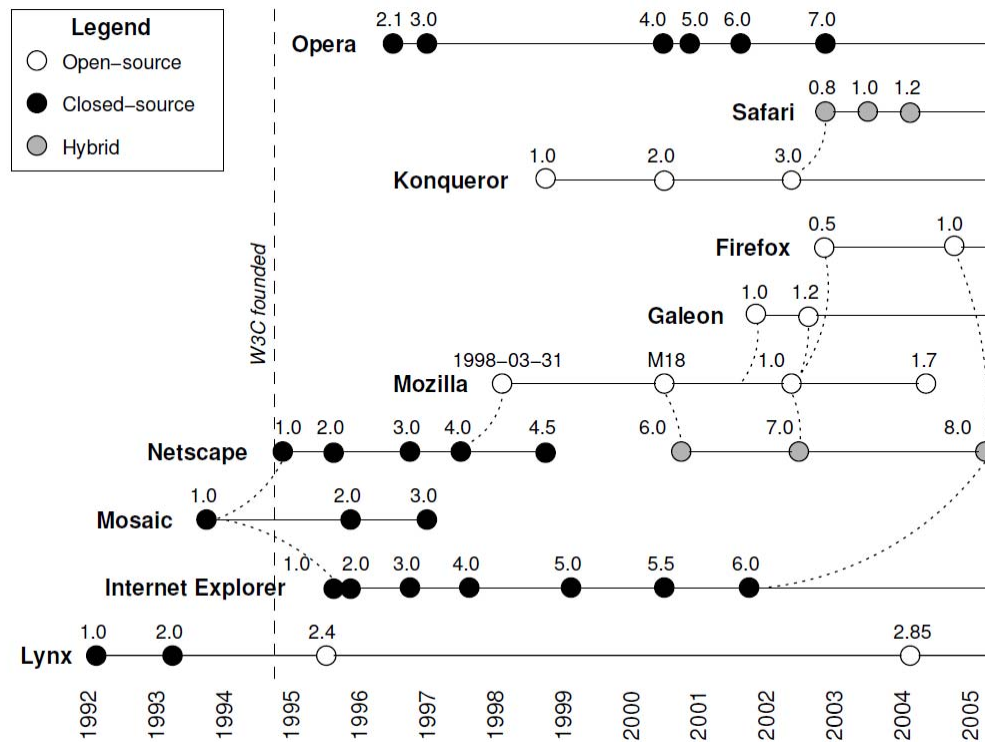


Figura II-15. Historia de los navegadores web [44]

II.2.3.- El contenido dinámico

Para añadir interacción con el usuario se introdujo desde 1993 el estándar CGI Common Gateway Interface [45] es una interfaz que permite la ejecución de scripts (interpretados, lenguaje Perl) o programas externos en servidores HTTP, devolviendo al usuario una página web con la respuesta de la ejecución.

Se crea un proceso separado en el servidor para cada petición por lo que se requiere mucho tiempo y memoria. Esto se intenta solventar mediante:

- La introducción de CGI compilado (C/C++)
- La introducción de FastCGI [46] manteniendo un proceso vivo en un servidor de aplicaciones, activado a través de un socket o una conexión TCP que aceptan la conexión desde el servidor web y procesan de forma concurrente diversas peticiones mediante técnicas de programación multi-hilo, devolviendo la respuesta al servidor web por la misma conexión. Se realiza una separación real entre ambos servidores y se puede usar cualquier lenguaje de programación que admita sockets.
- El uso de Simple Common Gateway Interface SCGI [47], 2006.
- El uso de intérpretes embebidos, con módulos de extensión como mod_php. PHP es un lenguaje de script que corre en un servidor web para crear contenido dinámico, y que hoy en día es usado en el 79% [48] de los portales web como lenguaje de programación del lado del servidor.

- Los servidores web desarrollan mecanismos de extensión que permiten la ejecución de software de terceros: módulos Apache, plugins de Netscape o de IIS ISAPI.

Netscape Server Application Programming Interface, desarrollado por Rob McCool para el servidor web de Netscape. El plugin NSAPI corre dentro del proceso del servidor, implementando Server Application Function SAF. Netscape, fundada en 1994, es la creadora del navegador web Netscape Navigator, cuya versión Netscape Communicator 4.0 se abrió como código libre en 1998 bajo el proyecto Mozilla justo antes de la compra de Netscape por AOL. Las siguientes versiones se basaron en Mozilla (6 desde 2000 y 7 desde 2002) y luego en Firefox (8, en 2005 y 9 en 2007, cuando es finalmente cerrado) [49]. Es creador de SSL, protocolo para comunicación online segura y de JavaScript, el lenguaje script más usado en páginas web desde el lado cliente, un 92% [48]. El servidor web de Netscape, lanzado en 1994 fue adquirido por Sun Microsystems como iPlanet, en 1998, dentro de la alianza Sun-AOL que duró hasta 2002. Luego pasó a propiedad de Sun, como Sun ONE y en 2003 como Sun Java System Web Server. En 2010 Oracle adquiere Sun e iPlanet se sigue distribuyendo como Oracle iPlanet Web Server [50]. En 2003 se crea también Sun Java System Application Server para el desarrollo de aplicaciones Java para servidores y servicios Web. En 2005 Sun lanza el proyecto GlassFish, un servidor de aplicaciones de código libre para la plataforma Java EE En 2009 se llamó Sun GlassFish Enterprise Server y finalmente en 2010 Oracle GlassFish Server [51].

Internet Server Application Programming Interface, es un API multicapas de Microsoft. No requiere procesos separados, al emplear múltiples hilos para sincronizar el trabajo. Se pueden generar dos tipos de aplicaciones, filtros (filtran las peticiones web hasta encontrar una que procesar) y extensiones (como ASP y ASP.NET, usado en un 20% de los portales web como lenguaje de programación del lado del servidor), ambas compiladas como DLL que se registran en el servidor web IIS [52]. Se escriben en C/C++.

Módulos Apache [53]. El servidor HTTP Apache fue desde 1996 la primera alternativa al servidor web de Netscape. Desarrollado para servidores Linux por la Apache Software Foundation (ASF), en diciembre de 2012 sirve el 64% de los portales web, mientras que el segundo, IIS, llega al 17% [48]. Desarrollado en C, soporta módulos compilados de autenticación, soporte de lenguajes de programación del lado servidor, soporte de SSL, TSL,... Apache Tomcat [54], como servidor web y contenedor de Java servlets y JavaServer Pages JSP, fue desarrollado por la ASF, con su versión inicial 3.0.x en 1999, a partir de la donación de código de Sun Java Web Server. Hoy en día la versión estable es la 7.0.x. En 2012 aparece Apache TomEE como Tomcat para Java EE.

- La Plataforma Java [55], con Java Empresarial Java EE o Java Standard SE emplea un contenedor de servlets para correr código Java que sirva contenido dinámico.

La Plataforma Java Enterprise Edition, o Java EE, antes conocida como Java 2 Platform Enterprise Edition J2EE hasta la versión de Java 1.4, es la versión de

Java Empresarial que permite el desarrollo y ejecución de software de aplicaciones Java, utilizando arquitecturas de N capas distribuidas con componentes de software modular, sobre un servidor de aplicaciones Java EE.

La especificación original J2EE fue desarrollada por la empresa Sun Microsystems. Comenzando con J2EE 1.3, la especificación fue desarrollada bajo el Java Community Process. JSR 58 especifica J2EE 1.3 y JSR 151 especifica J2EE 1.4. El SDK de J2EE 1.3 fue distribuido inicialmente como beta en abril de 2001. La beta del SDK de J2EE 1.4 fue distribuida por Sun en diciembre de 2002.

La especificación Java EE 5 fue desarrollada bajo el JSR 244 y el lanzamiento final se hizo el 11 de mayo de 2006. La versión actual tiene el nombre de Java EE 6 y fue lanzada en 2009. En 2013 se espera la versión Java EE 7.

Como principales servidores de aplicaciones, que incluyen todas las características de Java EE destacan WebLogic [56] de Oracle (antes BEA Systems) y WebSphere de IBM [57] como servidores comerciales, y JOnAS del consorcio ObjectWeb [58], JBoss AS de JBoss (división de Red Hat) [59], Geronimo de Apache [60] y el ya mencionado GlassFish de Oracle, como servidores libres.

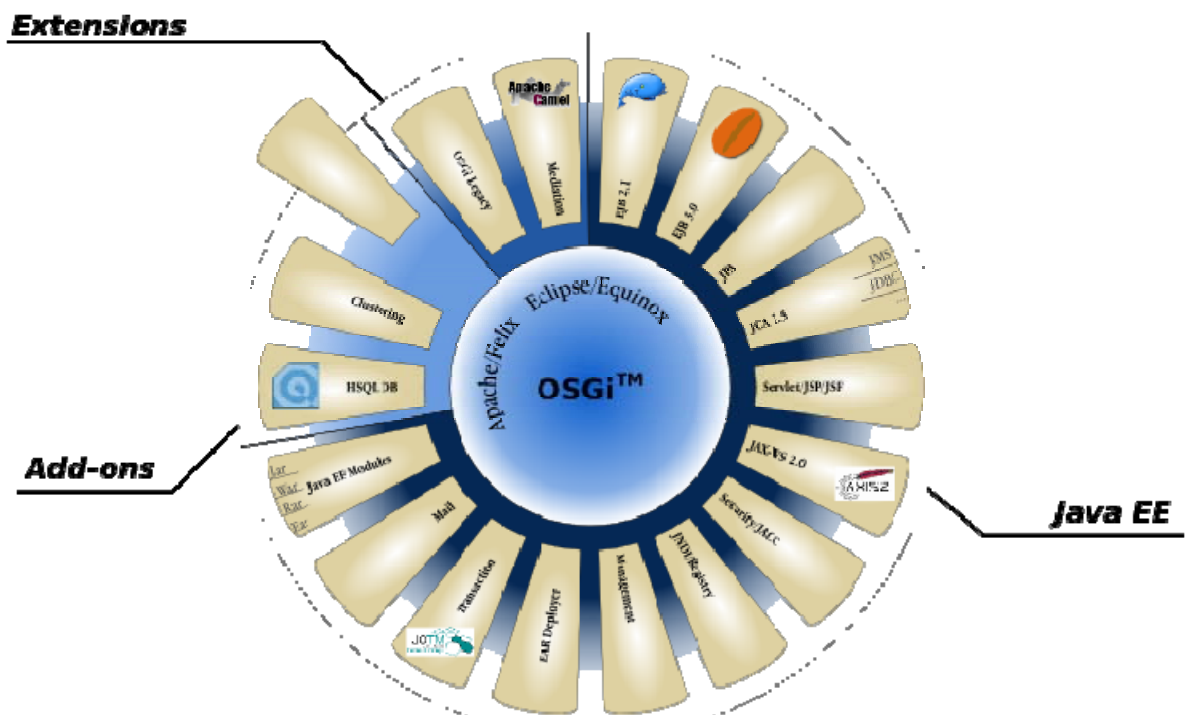


Figura II-16. Esquema general del servidor de aplicaciones JOnAS [58]

Algunos servidores de aplicaciones dejan gran parte de las características de Java EE (como EJB y JMS) y se centran más en los servlets y JSP: Tomcat de Apache y Jetty de la Eclipse Foundation [61].

II.2.4.- Los servlets

Los servlets son clases de Java EE empleadas para extender la capacidad de las aplicaciones de un servidor web, a modo de Java applets que corren en el servidor en lugar de en el cliente web. Son la contrapartida a tecnologías de contenido dinámico como PHP y ASP.NET. La primera especificación data de Sun Microsystems, 1997. Fueron concebidos por James Gosling, el padre de Java, en 1995, y fueron incluidos por primera vez en el producto Java Web Server. Su ventaja sobre CGI radica en la persistencia del servlet: una vez iniciado no se cierra, por lo que no es necesario iniciar procesos nuevos para nuevas peticiones.

Los servlets se definen en la Java Servlet API. A partir de la 2.3 las API se desarrollan bajo el Java Community Process JCP, mecanismo para desarrollar las especificaciones técnicas estándares para diversas tecnologías de Java, garantizando la estabilidad y compatibilidad entre plataformas. Las tecnologías y especificaciones propuestas para ser incorporadas a Java se definen en las Java Specification Requests JSR [62]. La JSR 53 define las especificaciones de Java Servlet y Java Server Pages desde 2002. La JSR 154, desde 2007, define la especificación Java Servlet 2.4. Desde 2010 la JSR 315 define la 3.0. La versión 3.1, JSR 340, está aprobada desde abril de 2013.

Un servlet es un componente Web basado en tecnología Java que manejado por un contenedor genera contenido dinámico. Es una clase independiente de la plataforma, compilado en bytecode¹ neutral que puede ser cargado dinámicamente en cualquier servidor Web que admita tecnología Java. Un contenedor o motor servlet es una extensión del servidor Web que permite funcionalidad servlet. Los servlets interactúan con los clientes Web vía modelos petición/respuesta implementados en el contenedor, que soporta, al menos, protocolo HTTP, o también HTTPS.

¹ Bytecode es el código intermedio entre el código fuente y el código máquina generado al compilar el código fuente. Es interpretado por la máquina virtual de Java JVM, por lo que se reduce la dependencia con el hardware y lo hace portable a diferentes arquitecturas y plataformas. Finalmente, en la máquina receptora un compilador *just-in-time* traduce el bytecode a código máquina para su ejecución.

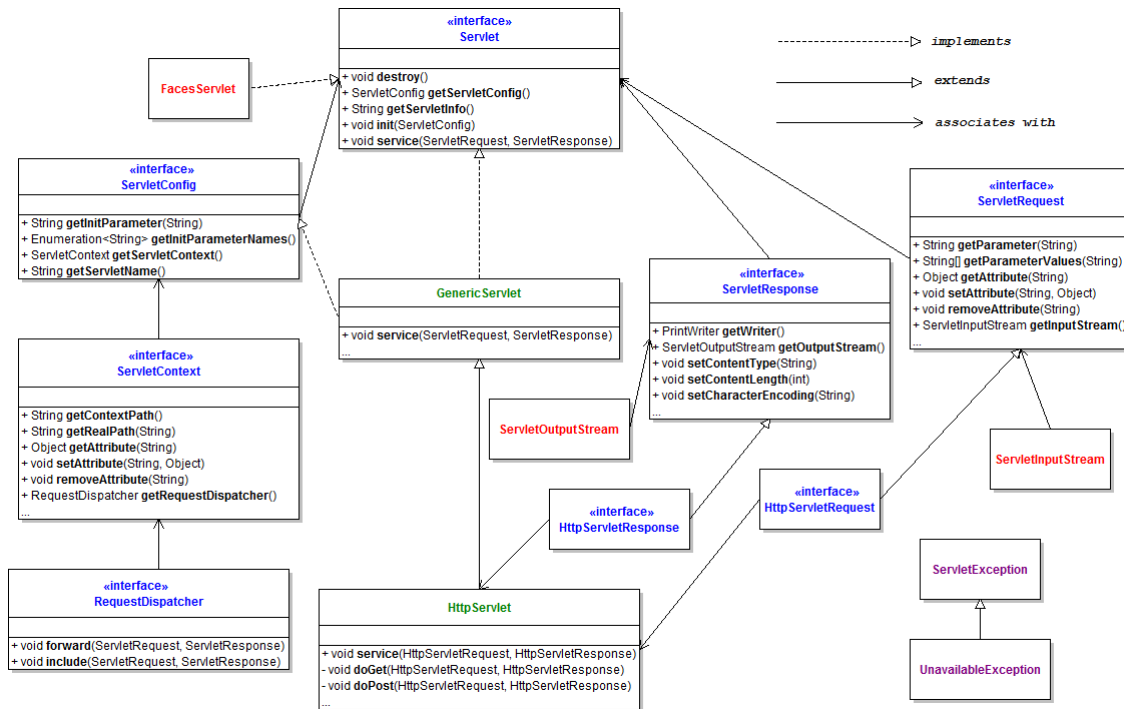


Figura II-17. Diagrama UML de clases del Java Servlet API [63]

Servlet es una interfaz implementada por `GenericServlet` o por `HttpServlet`, por lo que para implementar esta interfaz basta con escribir un servlet que extienda a alguna de estas. Lo habitual es que se extienda la segunda, `HttpServlet`.

El ciclo de vida de un servlet sigue la siguiente secuencia:

- El servlet se construye e inicializa en el método `init`. Se carga en memoria.
- Se procesa cualquier llamada de clientes dentro del método `service`.
- Se para el servlet y se destruye el método `destroy`, es limpiado por el *recolector de basura* GC y finaliza. Se descarga de memoria.

Además, la interfaz cuenta con dos métodos que proporcionan un objeto `ServletConfig` con los parámetros de inicialización del servlet y un objeto `String` con información del servlet.

II.2.5.- Web Application Frameworks WAF

Un WAF es un framework diseñado para dar soporte al desarrollo de sitios web dinámicos, aplicaciones web y servicios web. Incluye librerías para el acceso a bases de datos, plantillas, gestión de sesiones, tratando de promover la reutilización de código.

Los WAF pueden presentar distintos tipos de arquitectura:

- **MVC** La arquitectura principal está basada en el patrón modelo-vista-controlador MVC, que separa los datos del modelo con las reglas del negocio de la interfaz del usuario. Se modulariza el código, se puede reusar, y se pueden aplicar múltiples interfaces o vistas.

- **Push based.** La mayoría de los frameworks MVC siguen una arquitectura basada en push o en acciones: usan acciones para hacer el procesamiento requerido y luego envían los datos a la capa vista para mostrar los resultados. Ejemplos: Struts, Django, Ruby on Rails, Spring MVC...
- **Pull based.** Alternativamente los frameworks basados en pull o en componentes. El framework comienza con la capa vista y ésta descarga los resultados de los múltiples controladores según se requiera. Ejemplos: Tapestry, JBoss Seam, JSF, Wicket.
- **Tres capas.** Separación en tres capas físicas: cliente, que en aplicaciones web es un navegador web que corre HTML generado por la capa aplicación; aplicación con la lógica del negocio corriendo en un servidor, que se comunica por HTTP con la capa cliente; y base de datos generalmente relacional, conectada con la aplicación a través de una conexión JDBC.

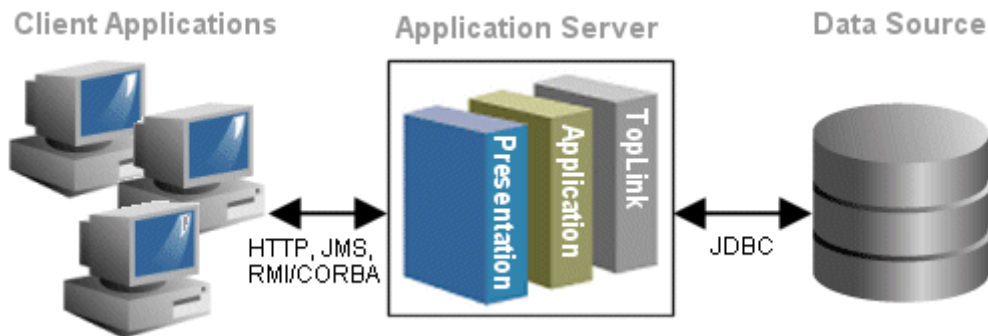


Figura II-18. Arquitectura de tres capas [64]

Características de los WAF

Entre las características más importantes de los frameworks WAF destacan:

- Sistema de plantillas, para generación de contenido HTML de forma dinámica, a partir del uso de etiquetas.
- Caché, almacenado de documentos web para reducir el ancho de banda usado y la carga del servidor.
- Seguridad, a través de sistemas de autenticación y autorización, para identificar al usuario de la aplicación y restringir el acceso a determinadas funcionalidades.
- Acceso a base de datos, permitiendo programar a alto nivel. Muchos frameworks de Java usan Hibernate como capa de persistencia, por lo que no es necesario definir explícitamente un esquema de la base de datos. Otros como Ruby on Rails definen propiedades del modelo basándose en el esquema de la base de datos.
- *Scaffolding*, método para construir aplicaciones basadas en bases de datos: a partir de una especificación de cómo debe ser usada la base de datos, el

compilador genera código que la aplicación usará para crear, leer, actualizar y eliminar los registros de la base de datos (CRUD create, read, update & delete).

- **Mapeado de direcciones URL**, para interpretar direcciones URL a partir de patrones, y describirlas a direcciones más sencillas de leer o indexar.
- **AJAX** (Asynchronous JavaScript and XML) en el navegador. Es una técnica para el desarrollo de aplicaciones web, haciendo que la página web responda más ágilmente, mediante el intercambio de pequeñas cantidades de datos con el servidor, de forma transparente al usuario, de modo que la página no tenga que recargarse completamente tras cada petición del usuario.

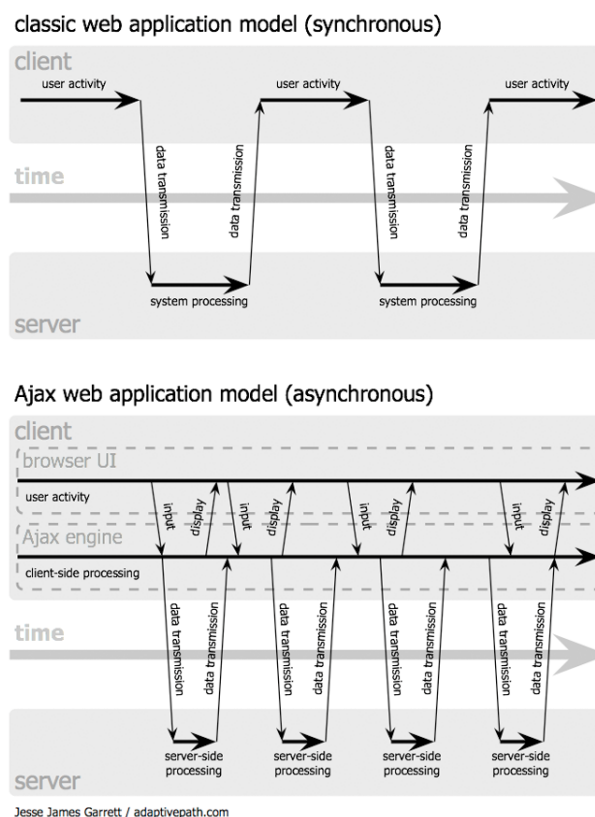


Figura II-19. Comparativa de patrón síncrono tradicional con asíncrono en AJAX [65]

Por la complejidad de programación en JavaScript, hay muchos frameworks de Ajax, del lado del navegador, en JavaScript: jQuery, MooTools o EXT JS. Incluso se generan soluciones Ajax a partir de lenguajes de más alto nivel (Java, Python o Ruby) mediante compilación a JavaScript, con compiladores como Morfik, Google Web Toolkit GWT. Ejemplos de Java: Apache Wicket, GWT, JBoss RichFaces, ICEFaces y PrimeFaces, Vaadin, GXT. También existe la opción contraria, desarrollo de JavaScript en el lado del servidor: ASP de Microsoft, Node.js creado en 2009.

- **Web Services**. Algunos frameworks permiten la creación o el uso de servicios web. Según el W3C [66], se trata de "...un sistema de software diseñado para soportar interacción interoperable máquina a máquina sobre una red. Este tiene una interface descrita en un formato procesable por una máquina (específicamente WSDL). Otros sistemas interactúan con el servicios web en

una manera prescrita por su descripción usando mensajes SOAP, típicamente enviados usando HTTP con una serialización XML en relación con otros estándares relacionados con la web”.

Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

El conjunto de servicios y protocolos para los servicios web es conocido comúnmente como “Web Services Protocol Stack” y básicamente son utilizados para definir, localizar, implementar y hacer que un servicio web interactúe con otro. Este conjunto está conformado esencialmente de cuatro subconjuntos [67]:

- **Servicio de transporte** de los mensajes entre aplicaciones. Los protocolos más usados son HTTP, FTP, SMTP, BEEP, JMS.
- **Mensajería XML**, codifica los mensajes en XML estándar, para que pueda ser interpretado por cualquier nodo de la red. REST Representational State Transfer, RPC, XML-RPC, XML, SOAP Simple Object Access Protocol, mensaje basado en XML entre un emisor y un receptor, que deben ser nodos SOAP.
- **Descripción del servicio**, el servicio web cuenta con una interfaz pública descrita por un formato WSDL Web Services Description Languages
- **Descubrimiento de Servicios**, UDDI Universal Description Discovery and Integration, marco para describir servicios, negocios e integrar servicios de negocios, permite comprobar qué servicios web están disponibles.

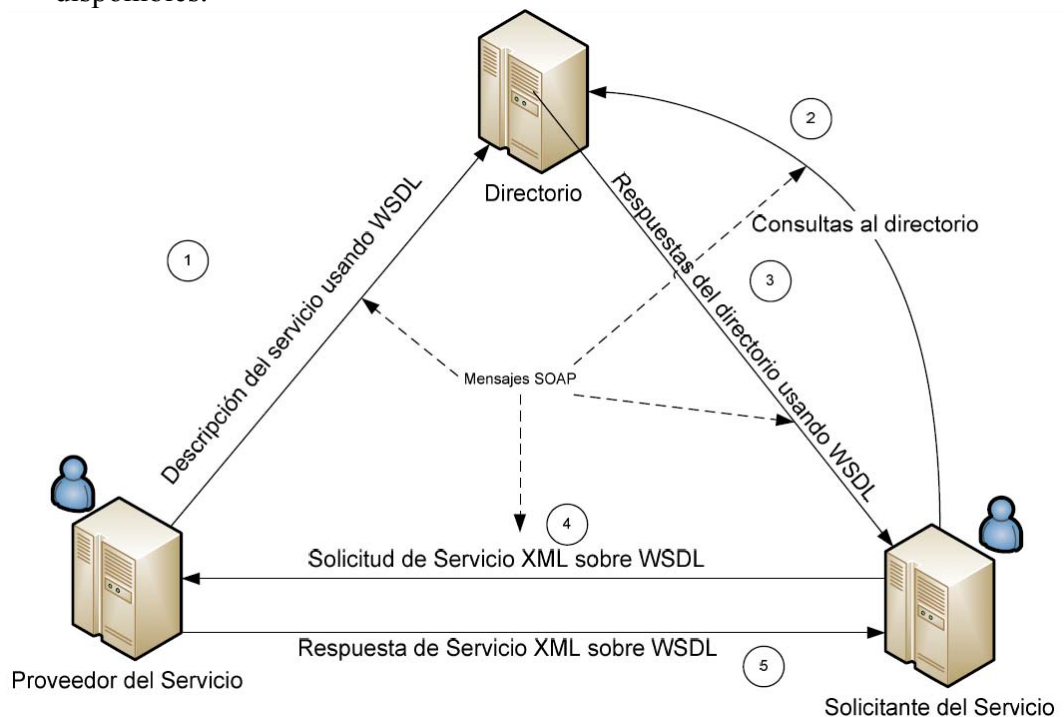


Figura II-20. Modelo básico del funcionamiento de los servicios Web [67]

II.2.6.- WAFs en Java

A continuación se detallan brevemente algunos de los frameworks de aplicaciones Web empleados actualmente, así como sus características principales.

Apache Struts [68]

Versión inicial Mayo 2000, versión estable 2.3.8, diciembre 2012. Es un Framework de código abierto para el desarrollo de aplicaciones web Java EE. A partir de la API Java Servlet adopta una arquitectura MVC.

Struts separa el modelo (lógica de la aplicación que interactúa con la base de datos) de la vista (páginas HTML presentadas al cliente) y del controlador (instancia que pasa información entre modelo y vista). Struts provee de un controlador propio, un servlet llamado `ActionServlet`, que captura las solicitudes HTTP y las pasa a componentes especiales llamados `Actions`, o funciones específicas de la aplicación Web. Para transferir los datos introducidos en los formularios de entrada hacia los componentes `Actions`, se utilizan los componentes `ActionForms`. En cuanto a la vista, se utilizan páginas JSP para generar las interfaces. Un `ActionForward` realiza la asociación entre un nombre lógico que devuelve un componente `Action` y la página JSP a la cual el `ActionServlet` debe transferir el control.

Spring [69]

Versión estable 3.2.1 enero 2013. Spring es un framework de aplicaciones Java EE, orientado a facilitar la tarea de crear una arquitectura coherente integrando lo mejor de los frameworks de una capa.

Spring Framework comprende diversos módulos que proveen un rango de servicios. Esencialmente es un contenedor de inversión de control que permite que la configuración de los componentes de aplicación y la administración del ciclo de vida de los objetos Java se lleve a cabo principalmente a través de la inyección de dependencias. Usando reflexión el contenedor es responsable de gestionar el ciclo de vida de los objetos Java.

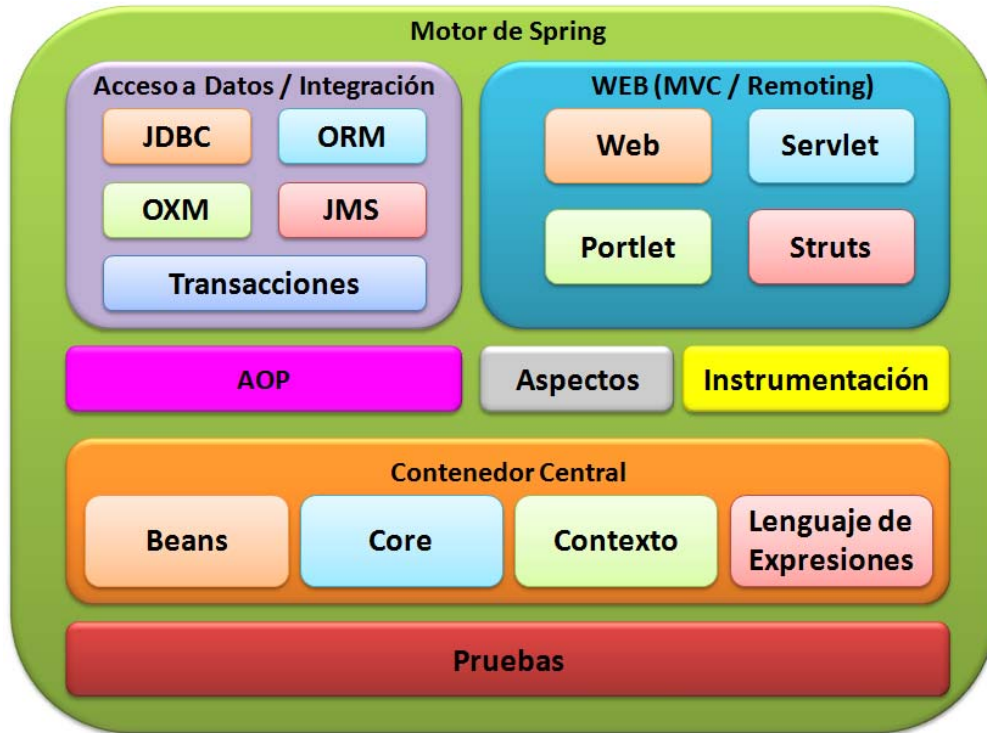


Figura II-21. Estructura modular de Spring [70]

Java Server Faces [71]

Versión estable 2.1.17, enero 2013. Es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE basadas en componentes mediante el uso de Java Server Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas (JSF 1.x) o mediante el uso de Facelets (JSF 2). También usa otras tecnologías como XUL (XML-based User-interface Language). Se suele usar en combinación con Ajax.

Con este framework se facilita de forma significativa la tarea de la construcción y mantenimiento de aplicaciones Web con UIs del lado del servidor. De manera sencilla se pueden conectar eventos generados en el cliente a código de la aplicación en el lado del servidor, mapear componentes UI a una página de datos del lado del servidor, construir un UI con componentes reutilizables y extensibles o grabar y restaurar el estado del UI más allá de la vida de las peticiones de servidor.

Algunas extensiones de JSF son JBoss RichFaces, ICEFaces, jQuery4jsf, PrimeFaces u OpenFaces.

Google Web Toolkit GWT [72]

Versión estable 2.5.0 octubre 2012. Framework creado por Google para crear y mantener complejas aplicaciones JavaScript y tecnología Ajax mediante la programación en Java y el uso de un compilador que lo traduce a HTML y JavaScript, compatible para varios navegadores. GWT contiene GWT Java-to-JavaScript Compiler, cuya misión es la de traducir el código desarrollado en Java al lenguaje JavaScript, cuando se usa GWT en modo producción.

Vaadin [73]

Framework Java de código abierto para construir aplicaciones de Internet enriquecidas (RIA). Versión estable 7.0 de febrero de 2013.

La programación se orienta al desarrollo de aplicaciones de escritorio, se crea el cliente en Java y luego se compila a JavaScript para poder desplegarlo en un navegador web, empleando el Google Web Toolkit GWT. Su arquitectura está orientada al servidor, por lo que casi toda la lógica corre en el servidor. Para la comunicación con el servidor se emplea tecnología AJAX en el navegador

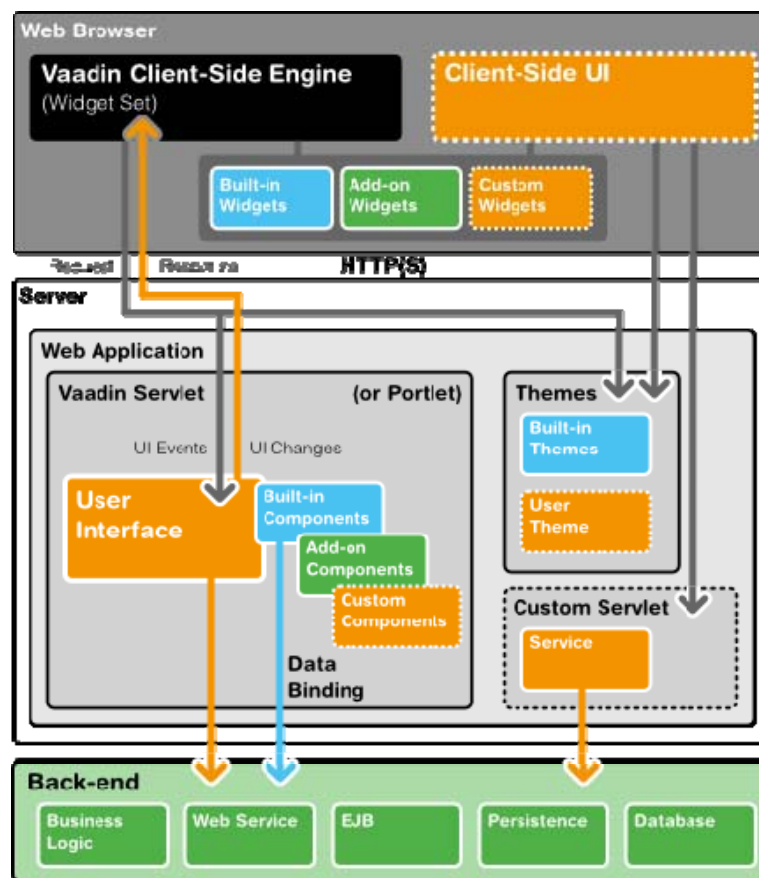


Figura II-22. Arquitectura del runtime de Vaadin [73]

Frameworks para aplicaciones de Internet enriquecidas RIA

Son Frameworks para el desarrollo de aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales. Se ejecutan en un navegador web por medio de plugins o mediante una máquina virtual. En lugar de un desarrollo web tradicional, se apoyan en un desarrollo cliente servidor.

Los frameworks para el desarrollo de RIA basados en plugins más importantes son Adobe Flash, JavaFX y Microsoft Silverlight. Los basados en HTML5 son Java applet, HTML5, GWT, Silverlight, Adobe Flex.

II.2.7.- Aplicaciones de frameworks a la ingeniería civil

La posibilidad de emplear sistemas distribuidos de cálculo gracias a Internet, disponiendo de clientes ligeros y servidores con la potencia de cálculo requerida para el tipo de problema a resolver no es una idea novedosa en sí misma, ya que diversos autores presentan esquemas de trabajo en esta línea.

Así, Chen y Lin [74] plantean un framework de elementos finitos a través de Internet, disponiendo en el cliente ligero de capacidades gráficas 3D e independencia de la plataforma. El cálculo se realiza de forma paralelizada.

Peng y Law [75] plantean igualmente un framework de elementos finitos basados en servicios web, a través de un navegador, empleando bases de datos en el servidor para almacenar información relativa al proyecto.

Eynard *et all* [76] realizan un entorno web colaborativo para diseño mecánico y análisis estructural.

Capítulo III. Los Cajones Multicelulares

III.1.- La geometría de un cajón

III.1.1.- Descripción de la geometría de un cajón tipo

Los cajones multicelulares flotantes están compuestos típicamente por dos elementos básicos: La losa y el fuste. El primero de ellos, también llamado solera, es un elemento plano y macizo de entre 0.5 y 1.5 metros de canto que cierra inferiormente la estructura, mientras que el segundo es básicamente un bloque paralelepípedo fuertemente aligerado mediante una serie de huecos verticales de forma circular dispuestos convenientemente para conseguir la mayor flotabilidad posible, manteniendo la resistencia y rigidez adecuadas. Las dimensiones habituales del fuste oscilan entre los 20 y 50 m de eslora (longitud), 15 y 25 m de manga (ancho) y 12 a 24 m de calado (altura).

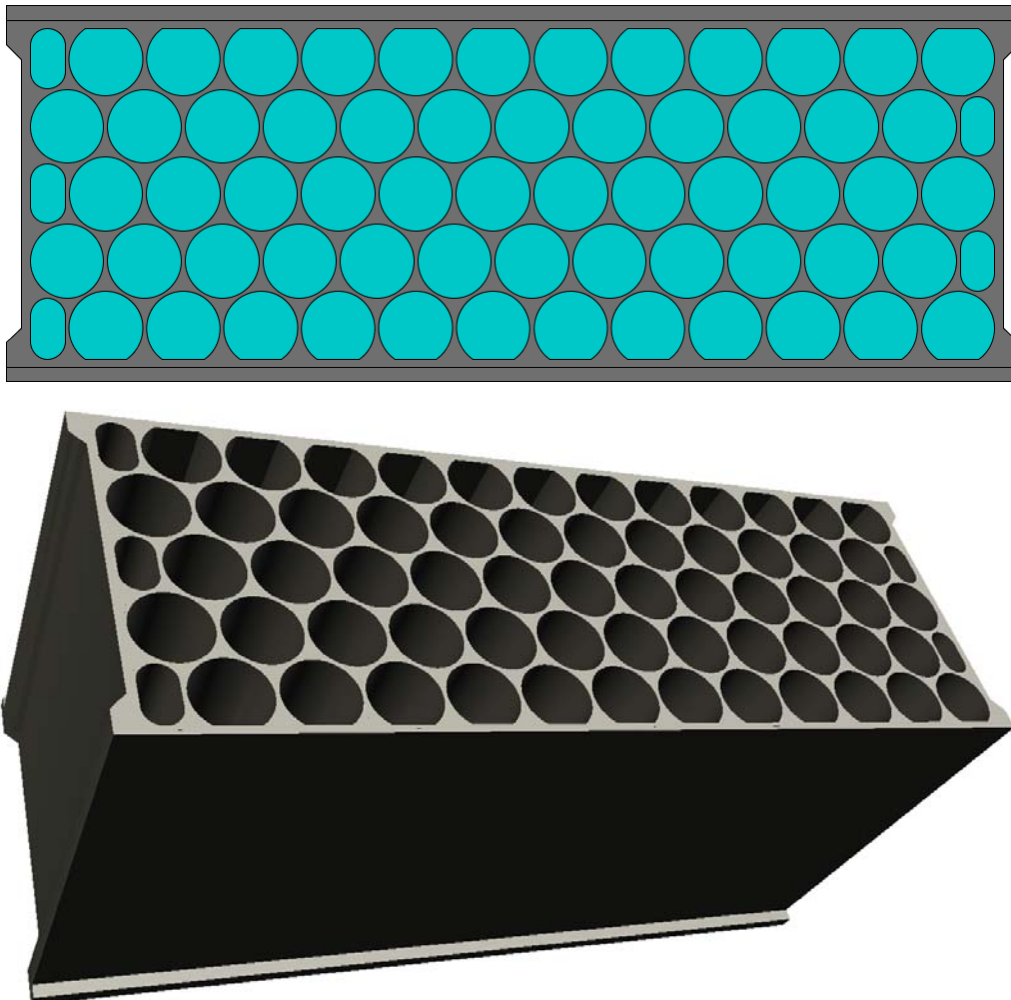


Figura III-1. Vista general en planta y en 3D de un cajón flotante
Puerto de Bilbao, Muelle AZ-2, 2008

En la **Figura III-1** se muestra una vista tridimensional de un cajón flotante típico y una vista en planta en la que se aprecia la sección transversal del fuste, caracterizada por la disposición de los huecos circulares *al tresbolillo*² para maximizar la superficie aligerada. En las proximidades de las paredes laterales, donde el corte brusco de estos huecos circulares plantearía problemas de ejecución y excesivas concentraciones de tensiones en ángulos entrantes, se emplean huecos *ovoidales*³ formados por un rectángulo y dos medios círculos. Así mismo, para lograr mantener unas dimensiones exteriores determinadas junto con unos espesores dados para las paredes frontales y laterales, puede ser preciso achaflanar los huecos circulares más próximos a dichas paredes.

Un cajón tipo se utiliza para construir un muelle o un dique de cajones y, por tanto, estará sometido en la fase de servicio a las acciones y sobrecargas de uso asociadas a la superestructura final de acuerdo con las indicaciones establecidas por el Autoridad Portuaria del puerto al que va destinado y las recomendaciones incluidas en la normativa [10] y [11].

Así, la estimación de las dimensiones exteriores vendrá condicionada por las condiciones resistentes exigibles a la estructura, por las condiciones de flotabilidad, estabilidad naval y condiciones marítimas, por el calado máximo admisible o por los condicionantes de explotación. Otros factores que pueden afectar a esta determinación son la capacidad del dique flotante o pontona donde se fabrica el cajón, o los posibles asientos que pueda tener el terreno.

Los cajones flotantes deberán proyectarse y construirse para que sean capaces de soportar todas las acciones que los puedan solicitar, con un nivel de seguridad aceptable, tanto durante la construcción como en servicio durante su vida útil, así como la agresividad del ambiente.

El método de los estados límite define las situaciones que de superarse indicarían que la estructura ha dejado de cumplir alguna de las funciones para las que fue proyectada. Los cajones se dimensionarán para no superar estados límite últimos de equilibrio, estabilidad en flotación, agotamiento por tensiones normales y por esfuerzos cortantes, así como el estado límite de servicio en fisuración.

Si bien existen técnicas simplificadas para la modelización de cajones flotantes mediante estados planos ([5], [2]), en el presente trabajo se pretende analizar el comportamiento resistente del cajón ante diversas sollicitaciones de forma tridimensional mediante el método de los elementos finitos, realizando el mallado de su volumen completo, con técnicas de análisis lineal. Técnicas no lineales se pueden encontrar en [16].

² De acuerdo con la definición de la RAE, *al tresbolillo* se refiere a la colocación de plantas en filas paralelas de modo que las de cada fila correspondan al medio de los huecos de la fila inmediata, de suerte que formen triángulos equiláteros. Esta acepción es aplicable a los huecos circulares del cajón por cuanto uniendo sus centros de 3 en 3 se forman triángulos equiláteros.

³ Estos huecos, si bien se denominan *ovoidales*, y así serán tratadas en lo que sigue, son *oblongos*, es decir rectángulos redondeados cuyo radio de acuerdo en las esquinas es igual a la mitad de su ancho. Coloquialmente se identifican más bien con óvalos.

Como la geometría es perfectamente conocida, para unas dimensiones dadas, cualquier programa generador de malla podría hacerlo rápidamente utilizando elementos tetraédricos. Sin embargo, dadas las peculiares características de la sección del fuste, la malla resultante no sería la más adecuada y contendría un elevado número de nodos y elementos, con el consiguiente aumento del coste computacional asociado al tratar de obtener resultados más precisos.

En cambio, la malla estructurada es más adecuada para el estudio del cajón, aunque su generación sea más costosa, ya que requiere un tratamiento previo de la geometría para facilitar al programa de pre-procesado el mallado con elementos estructurados.

En este capítulo se describirá en detalle este tratamiento de la geometría, partiendo del análisis de la sección transversal típica del fuste según las variables que la definen. Será básico disponer de un algoritmo que, dadas estas variables, sitúe con precisión la colocación de los aligeramientos en la sección.

III.1.2.- La sección transversal

En la **Figura III-2** se muestra la retícula de hexágonos que circunscriben a cada uno de los aligeramientos sobre la sección transversal de un cajón flotante típico. Se observa la disposición en panal de abejas que adoptan estos hexágonos, en los que cada aligeramiento ocupa una *celda*⁴. En lo que sigue, se empleará este nombre para referirse a cualquiera de estos aligeramientos. El empaquetamiento hexagonal de celdas es la forma más efectiva de agrupar tantas celdas como sea posible en un espacio limitado, empleando el mínimo material posible⁵.

En la **Figura III-3** se muestra ampliada una de las esquinas de la sección para acotar algunas de las variables más significativas del problema:

- D es el diámetro de las celdas circulares, R es el radio.
- e_{Cir} es el espesor entre celdas circulares adyacentes
- e_{Ov} es el espesor entre celdas circulares y ovoidales
- e_{Lat} es el espesor de las paredes laterales y e_{ChaLat} es el grado de achaflanamiento de las celdas circulares de los paramentos laterales

⁴ *Cel.la* o celda en latín, es el nombre específico de cada una de las casillas hexagonales de que está construido un panal de abejas.

⁵ Las abejas guardan la miel en celdillas individuales, formando un mosaico sin huecos ni salientes entre las celdillas, con objeto de aprovechar el espacio al máximo. De todos los polígonos regulares con el mismo perímetro, encierra más área aquél que tenga mayor número de lados. Si se añade la condición de no dejar huecos al rodearse de polígonos adyacentes, se podrían usar triángulos equiláteros, cuadrados y hexágonos regulares. Como estos últimos son los que más lados tienen, la figura geométrica más óptima es el hexágono. La reciente prueba matemática de esta afirmación se encuentra en [77], estableciendo que cualquier partición del plano en regiones de igual área tiene un perímetro al menos igual que el de la malla de hexágonos regulares.

$$L = \frac{D_a}{2 \cdot \cos(\pi/6)} = \frac{D + e_{Cir}}{\sqrt{3}} \quad (\text{III-1})$$

III.1.3.- Determinación de la eslora y la manga

La eslora y la manga son las dimensiones externas de la sección, sin tener en cuenta los topes. Es conveniente introducir el parámetro booleano b_{Sim} para contemplar la posibilidad de simetría respecto del eje transversal o no (Figura III-4):

$$b_{Sim} = \begin{cases} 1 & \text{si hay simetría transversal} \\ 0 & \text{si no hay simetría transversal} \end{cases} \quad (\text{III-2})$$

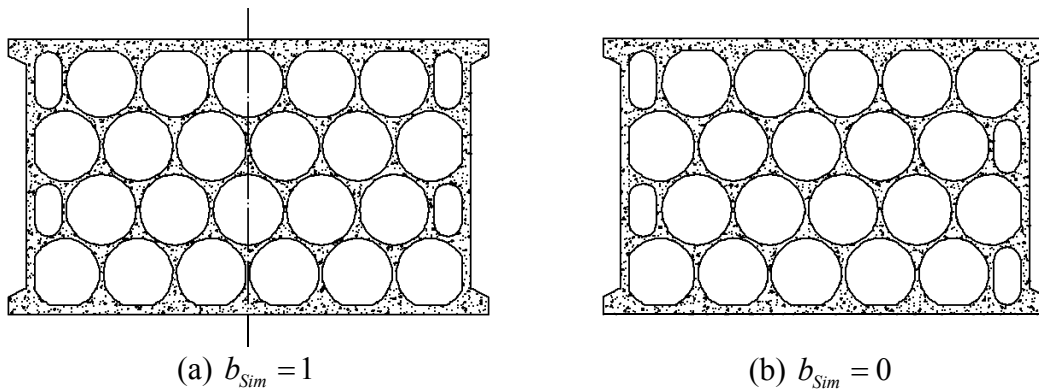


Figura III-4. Ejemplo de secciones: a) con simetría, y b) sin simetría

A partir de la Figura III-3, si N_{Col} es el número de celdas circulares a disponer en cada fila del cajón completo (medido en cualquier fila sin aligeramientos ovoidales en el caso de secciones con simetría transversal, o en cualquier fila en caso contrario), el valor de la eslora se determina a partir de la siguiente ecuación:

$$Eslora = (D + e_{Cir}) \cdot \left(N_{Col} + \frac{b_{Sim}}{2} \right) - e_{Cir} + 2 \cdot (e_{Lat} - e_{ChaLat}) \quad (\text{III-3})$$

Así mismo, el valor de la manga dependerá del número de filas dispuestas, N_{Fil} :

$$Manga = \frac{3 \cdot L}{2} \cdot (N_{Fil} - 1) + D + e_{LTierra} - e_{ChaFroLT} + e_{LMar} - e_{ChaFroLM} \quad (\text{III-4})$$

El valor obtenido por las expresiones (III-3) y (III-4) es independiente de cómo sean los aligeramientos ovoidales y de que el número de filas dispuesto sea par o impar. Por otro lado, y como norma general, $N_{Col} \geq 2$ y $N_{Fil} \geq 2$.

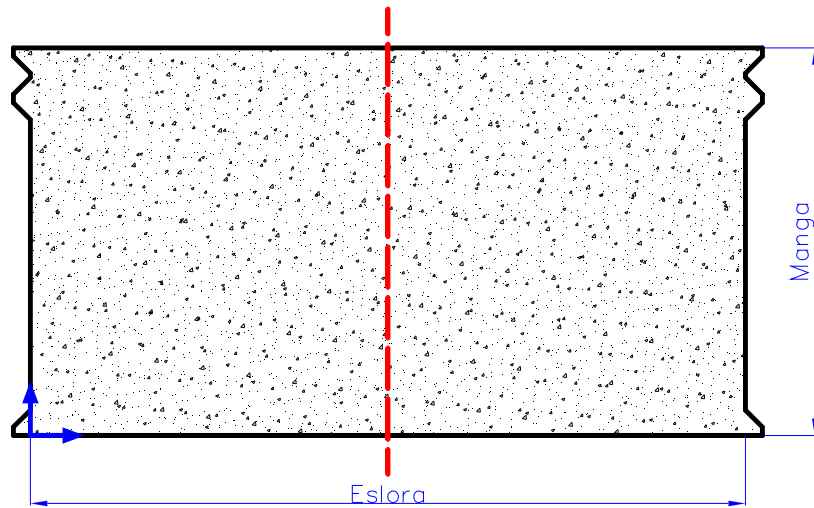


Figura III-5. Sección maciza y cotas

En el valor de la eslora no se ha considerado la longitud adicional que introducen los *topes* laterales de separación. Su misión principal es la de rigidizar las esquinas del cajón para evitar que durante la colocación del mismo se golpee la pared lateral de un cajón ya ubicado, que es más frágil, a la vez que sirven de guía durante esta operación.

Normalmente se dispone un tope en cada esquina, aunque en cajones de gran manga es habitual añadir un segundo tope en el Lado Mar e incluso en el Lado Tierra. La siguiente figura ilustra los parámetros geométricos que definen estos elementos.

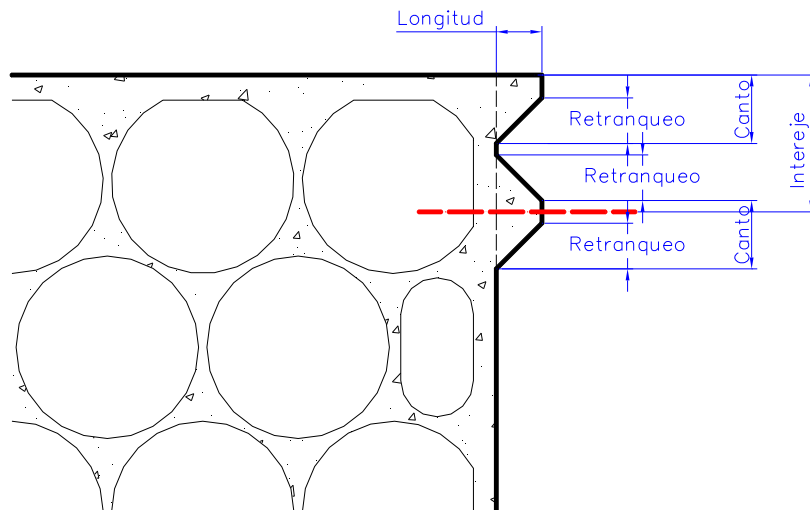


Figura III-6. Definición geométrica de los topes laterales



Figura III-7. Cajón multicelular con doble tope en LT y LM
Barcelona, Muelle del Prat Fase II, 2008.

Por conveniencia, se definen los siguientes parámetros que especifican si hay o no doble tope en Lado Mar y en Lado Tierra:

$$b_{Tope2LM} = \begin{cases} 1 & \text{si hay segundo tope en LM} \\ 0 & \text{si no hay segundo tope en LM} \end{cases} \quad (\text{III-7})$$

$$b_{Tope2LT} = \begin{cases} 1 & \text{si hay segundo tope en LT} \\ 0 & \text{si no hay segundo tope en LT} \end{cases} \quad (\text{III-8})$$

Determinados los valores de la eslora y manga y los tope a colocar, la sección transversal maciza de la losa del cajón, sin contabilizar los voladizos que puede presentar en Lado Tierra y Lado Mar viene dada por la ecuación (III-9), mientras que la posición del centro de gravedad de la sección maciza, respecto de la posición de referencia ubicada en la esquina inferior izquierda (sin contar el tope), **Figura III-5**, se indica en la ecuación (III-10). Por conveniencia, se ha definido el canto medio del tope como $CM_{Tope} = C_{Tope} - R_{Tope}/2$.

$$A_{Maciza}^{Fuste} = Eslora \cdot Manga + 2 \cdot L_{Tope} \cdot (2 \cdot CM_{Tope} + C_{Tope} \cdot (b_{Tope2LM} + b_{Tope2LT})) \quad (\text{III-9})$$

$$CG_{Maciza}^{Fuste} = \left\{ \frac{Eslora}{2}, \frac{1}{A_{Maciza}^{Fuste}} \cdot \left((Eslora \cdot Manga + 4 \cdot L_{Tope} \cdot CM_{Tope}) \cdot \frac{Manga}{2} + 2 \cdot L_{Tope} \cdot C_{Tope} \cdot (b_{Tope2LM} \cdot (Manga - I_{Tope}) + b_{Tope2LT} \cdot I_{Tope}) \right) \right\} \quad (\text{III-10})$$

A efectos de determinar la estabilidad naval del cajón es necesario igualmente conocer la inercia de la sección transversal maciza del fuste. Tomando como referencia la esquina exterior del tope de Lado Tierra y la parte central exterior del tope intermedio:

$$\begin{aligned}
 CG_{Exterior}^{Tope} &= \left\{ \frac{L_{Tope}}{2} \cdot \left(1 + \frac{R_{Tope}}{6 \cdot CM_{Tope}} \right), \frac{R_{Tope}^2}{24 \cdot CM_{Tope}} + \frac{CM_{Tope}}{2} \right\} \\
 CG_{Intermedio}^{Tope} &= \left\{ \frac{L_{Tope}}{2} \cdot \left(1 + \frac{R_{Tope}}{3 \cdot C_{Tope}} \right), 0 \right\} \\
 I_{Exterior}^{Tope} &= \left\{ \frac{L_{Tope}^3}{36 \cdot CM_{Tope}} \cdot (2 \cdot CM_{Tope}^2 + 2 \cdot CM_{Tope} \cdot C_{Tope} - C_{Tope}^2), \right. \\
 &\quad \left. \frac{L_{Tope}}{36 \cdot CM_{Tope}} \cdot (8 \cdot CM_{Tope}^4 - 8 \cdot CM_{Tope}^3 \cdot C_{Tope} + 4 \cdot CM_{Tope} \cdot C_{Tope}^3 - C_{Tope}^4) \right\} \\
 I_{Intermedio}^{Tope} &= \left\{ \frac{L_{Tope}^3}{12} \cdot \left(C_{Tope} - \frac{R_{Tope}^2}{3 \cdot C_{Tope}} \right), \frac{L_{Tope} \cdot C_{Tope}}{12} \cdot (C_{Tope}^2 + R_{Tope}^2) \right\}
 \end{aligned} \tag{III-11}$$

De tal modo que la inercia de la sección maciza del fuste es:

$$\begin{aligned}
 I_{Y,Macizo}^{Fuste} &= \frac{1}{12} \cdot Eslora^3 \cdot Manga + 4 \cdot I_{Y,Exterior}^{Tope} \\
 &\quad + 4 \cdot L_{Tope} \cdot CM_{Tope} \cdot \left(\frac{Eslora}{2} + L_{Tope} - CG_{X,Exterior}^{Tope} \right)^2 \\
 &\quad + 2 \cdot (b_{Tope2LT} + b_{Tope2LM}) \cdot \left(I_{Y,Intermedio}^{Tope} + L_{Tope} \cdot C_{Tope} \cdot \left(\frac{Eslora}{2} + L_{Tope} - CG_{X,Intermedio}^{Tope} \right)^2 \right) \\
 I_{X,Macizo}^{Fuste} &= \frac{1}{12} \cdot Eslora \cdot Manga^3 + Eslora \cdot Manga \cdot \left(\frac{Manga}{2} - CG_{Y,Macizo}^{Fuste} \right)^2 \\
 &\quad + 4 \cdot I_{X,Exterior}^{Tope} + 2 \cdot L_{Tope} \cdot CM_{Tope} \cdot (CG_{Y,Macizo}^{Fuste} - CG_{Y,Exterior}^{Tope})^2 \\
 &\quad + 2 \cdot L_{Tope} \cdot CM_{Tope} \cdot (Manga - CG_{Y,Exterior}^{Tope} - CG_{Y,Macizo}^{Fuste})^2 + \\
 &\quad + 2 \cdot b_{Tope2LT} \cdot \left(I_{X,Intermedio}^{Tope} + L_{Tope} \cdot C_{Tope} \cdot (CG_{Y,Macizo}^{Fuste} - I_{Tope})^2 \right) \\
 &\quad + 2 \cdot b_{Tope2LM} \cdot \left(I_{X,Intermedio}^{Tope} + L_{Tope} \cdot C_{Tope} \cdot (Manga - I_{Tope} - CG_{Y,Macizo}^{Fuste})^2 \right)
 \end{aligned} \tag{III-12}$$

III.1.4.- Las celdas ovoidales

Las celdas *ovoidales* permiten aligerar las zonas próximas a las paredes laterales que no son abarcadas por celdas circulares completas. No sería adecuado diseñar celdas circulares *mitad*, por los problemas de ejecución de los ángulos agudos generados y de las posibles concentraciones de tensiones en éstos.

En su lugar, se colocan estas celdas que quedan definidas por el radio de los círculos r_{Ov} y por la altura del rectángulo h_{Ov} .

Según se muestra en la Figura III-8, el radio del ovoide es función del espesor entre gargantas (circular-circular y circular-ovoidal), el diámetro de las celdas circulares, y la magnitud del chaflán lateral:

$$r_{Ov} = \frac{D + e_{Cir}}{4} - \frac{e_{Ov} + e_{ChaLat}}{2} \quad \text{(III-13)}$$

La altura del rectángulo del ovoide se determinará a partir del espesor de garganta existente con las celdas adyacentes. La distancia entre el centro del semicírculo superior del ovoide y el centro de la celda circular superior adyacente es $d = r_{Ov} + e_{Ov} + D/2$. Su componente en la dirección longitudinal es $d_l = r_{Ov} + e_{Ov} - e_{Cir}/2$, por lo que la componente transversal será $d_t = \sqrt{d^2 - d_l^2}$. Dado que la distancia vertical entre centros de celdas circulares es $3 \cdot L/2$, la altura buscada será el doble de la diferencia entre este valor y d_t :

$$h_{Ov} = 3 \cdot L - 2 \cdot d_t \quad \text{(III-14)}$$

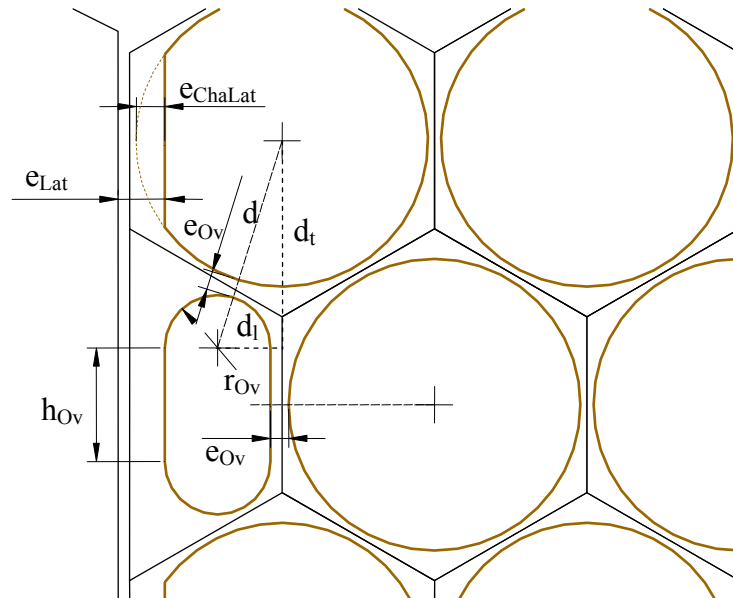


Figura III-8. Variables empleadas en la determinación del ovoide

En el caso particular de las celdas ovoidales de esquina (**Figura III-9**), se asume que el círculo exterior se ajusta al espesor del muro del Lado Tierra o Lado Mar según corresponda. El radio del ovoide será el mismo que en el caso anterior, ecuación (III-13), mientras que la altura es, en función de cada paramento:

$$h_{OvEsqLT} = \frac{h_{Ov}}{2} + R - e_{ChaFroLT} - r_{Ov} \quad \text{(III-15)}$$

$$h_{OvEsqLM} = \frac{h_{Ov}}{2} + R - e_{ChaFroLM} - r_{Ov} \quad \text{(III-16)}$$

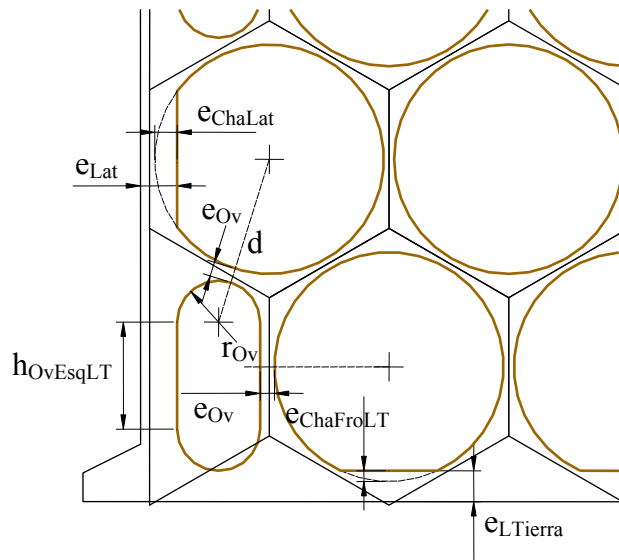


Figura III-9. Trazado del ovoide en una esquina del Lado Tierra

III.1.5.- Grado de achaflanamiento

A continuación se definirán los ángulos con los que definir el grado de achaflanamiento en las celdas frontales (del Lado Tierra y del Lado Mar) y laterales.

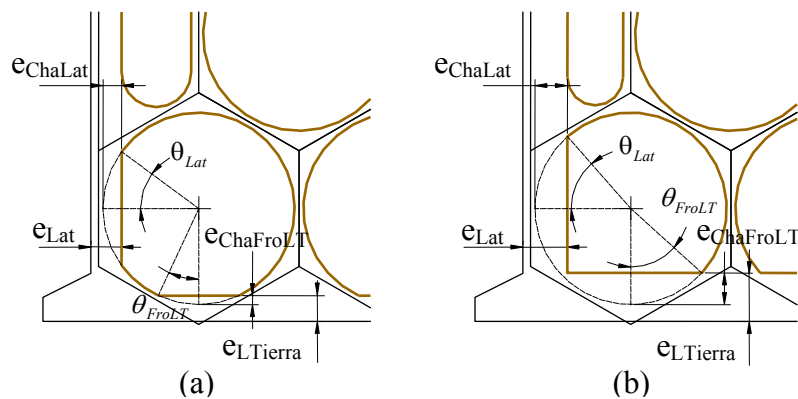


Figura III-10. Ángulos definidos a partir del grado de achaflanamiento

De acuerdo con la **Figura III-10**, se considerarán los siguientes ángulos:

$$\theta_{Lat} = \arccos\left(1 - \frac{e_{ChaLat}}{R}\right) \quad \text{(III-17)}$$

$$\theta_{FroLT} = \arccos\left(1 - \frac{e_{ChaFroLT}}{R}\right) \quad \text{(III-18)}$$

$$\theta_{FroLM} = \arccos\left(1 - \frac{e_{ChaFroLM}}{R}\right) \quad \text{(III-19)}$$

Como norma general, estos ángulos no deberán ser superiores a 60°, ya que no es conveniente que los chaflanes superen la mitad del radio de la celda circular, por limitaciones constructivas y para evitar concentraciones de tensiones en los ángulos

entrantes y porque se reduce de forma notable el efecto arco precisamente en las zonas más críticas, las expuestas directamente al oleaje del mar.

En función del grado de achaflanamiento, las celdas de tipo *Circular de Esquina* pueden presentar dos geometrías distintas. Si los chaflanes son relativamente pequeños, entre el chaflán vertical y el horizontal la celda presenta un arco que los une (**Figura III-10.a**). En cambio, este arco no existe cuando el grado de achaflanamiento es elevado, ya que ambos chaflanes (el vertical y el horizontal) se cortan dentro de la celda, formando un ángulo recto (**Figura III-10.b**).

La presencia o no del arco, determinante para conocer la forma de la celda de esquina y, por tanto, su área, depende del valor de los ángulos anteriormente definidos. El parámetro b_{ArcEsq} valdrá 1 si hay arco y 0 en caso contrario, según se trate del Lado Tierra o Mar:

$$b_{ArcEsqLT} = \begin{cases} 1 & \text{si } \theta_{Lat} + \theta_{FroLT} \leq \pi/2 \\ 0 & \text{si } \theta_{Lat} + \theta_{FroLT} > \pi/2 \end{cases} \quad \text{(III-20)}$$

$$b_{ArcEsqLM} = \begin{cases} 1 & \text{si } \theta_{Lat} + \theta_{FroLM} \leq \pi/2 \\ 0 & \text{si } \theta_{Lat} + \theta_{FroLM} > \pi/2 \end{cases} \quad \text{(III-21)}$$

III.1.6.- Tipología de aligeramientos

Conocida la geometría de la sección transversal, se procede a la subdivisión de la misma en diferentes *regiones poligonales* [136] en el entorno de cada aligeramiento, salvo en los topes. Para ello, se empleará el mismo tipo de región en todos aquellos aligeramientos que, independientemente de su posición, reúnan las mismas condiciones topológicas. De este modo, se tendrá un número reducido de tipologías y el tratamiento posterior será idéntico para cada una de las regiones pertenecientes a una tipología.

La **Figura III-11** muestra las regiones obtenidas en media sección de un cajón simétrico dado, empleando en este caso 9 tipologías diferentes.

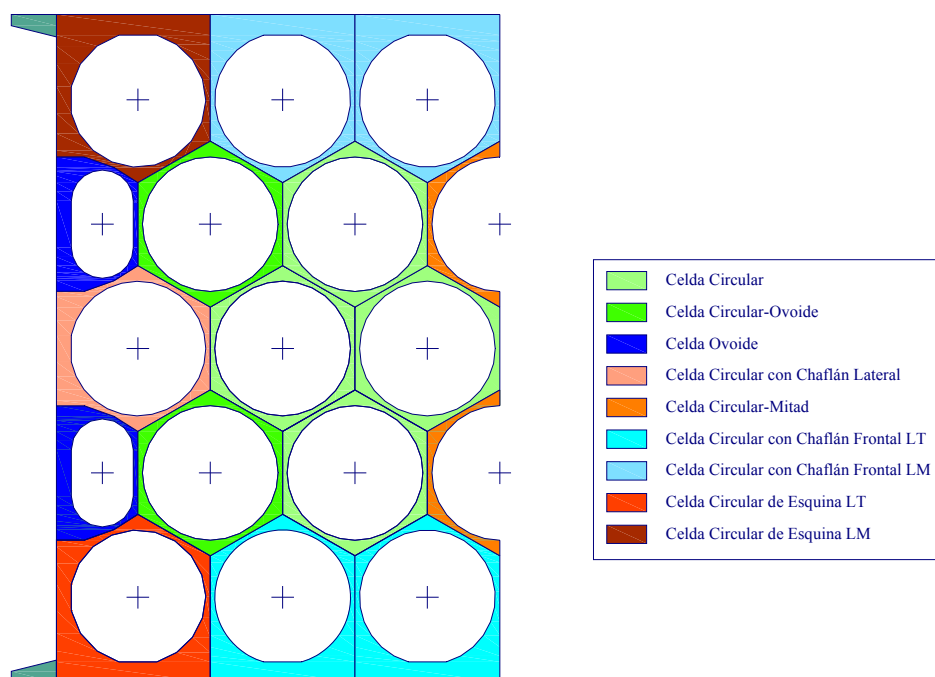


Figura III-11. División de una sección transversal según tipologías

Aunque únicamente se emplean aligeramientos circulares y ovoidales, las posibles modificaciones por achaflanamiento de los mismos y el tipo de celdas adyacentes dan lugar a un máximo de 18 tipologías. En la **Figura III-12** se recogen 12. Las 6 no representadas se corresponden con el Lado Mar.

Las celdas *circulares* (Figura III-12.a entre aligeramientos circulares y Figura III-12.c entre huecos adyacentes circular y ovoidal) son los aligeramientos más comunes en la sección. Las celdas *circular-mitad* (Figura III-12.b y Figura III-12.d) surgen al aplicar la condición de simetría sobre el plano transversal central a una celda circular, en el caso de que sea posible ($b_{Sim} = 1$) y se analice el cajón únicamente media sección.

Las celdas *ovoidales* (Figura III-12.e) y celdas *ovoidales de esquina* (Figura III-12.f; LT y LM) son los aligeramientos posibles con forma de ovoide.

Las celdas *circulares con chaflán frontal* (Figura III-12.g y Figura III-12.i, para el caso de celdas adyacentes de tipo circular y ovoidal, respectivamente; LT y LM) son los aligeramientos circulares de las filas primera y última, las más próximas a las paredes exteriores del Lado Tierra y del Lado Mar, sin contar aquéllas que, a su vez, están en contacto con las paredes laterales. Pueden presentar cierto grado de achaflanamiento en la zona próxima al muro. Las celdas *circulares con chaflán frontal-mitad* (Figura III-12.h y Figura III-12.j para el caso de aligeramientos adyacentes de tipo circular y ovoidal; LT y LM) surgen al aplicar la condición de simetría sobre el plano transversal central, en los casos ya comentados.

Análogamente, las celdas tipo *circular con chaflán lateral* (Figura III-12.k) son los aligeramientos circulares en contacto con las paredes laterales, descontando las que a su vez se sitúan en las paredes de Lado Mar y Tierra. Pueden presentar cierto grado de achaflanamiento en la zona próxima al muro.

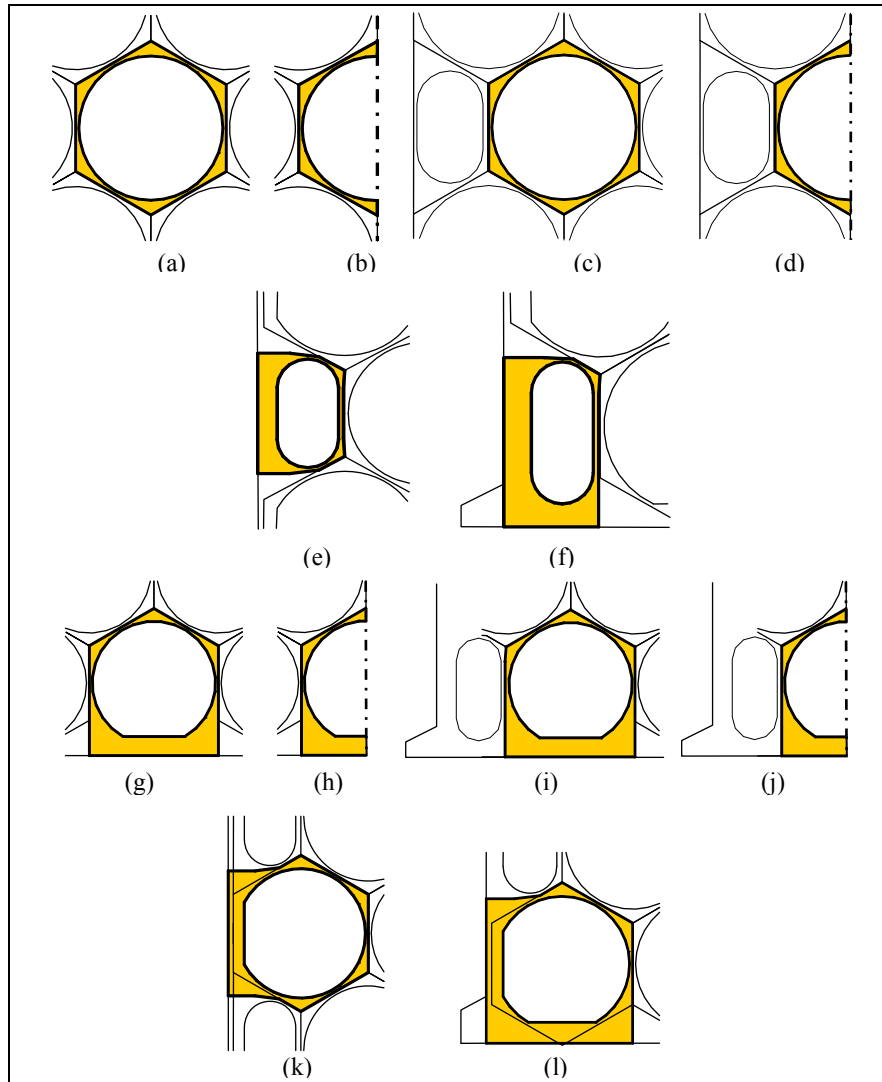


Figura III-12. Tipología de aligeramientos

Por último, las celdas tipo *circular de esquina* (Figura III-12.l; LT y LM) se corresponden con los huecos circulares en contacto tanto con las paredes laterales como con las frontales. Pueden presentar cierto grado de achaflanamiento en cualquiera de las zonas próximas a los muros.

III.1.7.- Posicionamiento y conteo de aligeramientos

Una vez que se conocen el número de filas N_{Fil} y el de columnas N_{Col} , y se ha seleccionado diseñar una solución simétrica ($b_{Sim} = 1$) o no simétrica ($b_{Sim} = 0$) el siguiente algoritmo realiza el posicionamiento de los aligeramientos, especificando la tipología que corresponde en cada caso.

Si bien no existe un criterio universal al respecto de cómo se debe comenzar la colocación de celdas en los paramentos exteriores, en la solución que se propone se adoptan los siguientes criterios:

3. Los Cajones Multicelulares

- En secciones simétricas de número par de filas, la primera y la última celda del Lado Tierra deben ser *Ovoide de Esquina* y, como consecuencia, la primera y la última del Lado Mar serán *Circular de Esquina*, Figura III-13.a.
- En secciones simétricas de número impar de filas, no habrá celdas *Ovoide de Esquina*, Figura III-13.b.

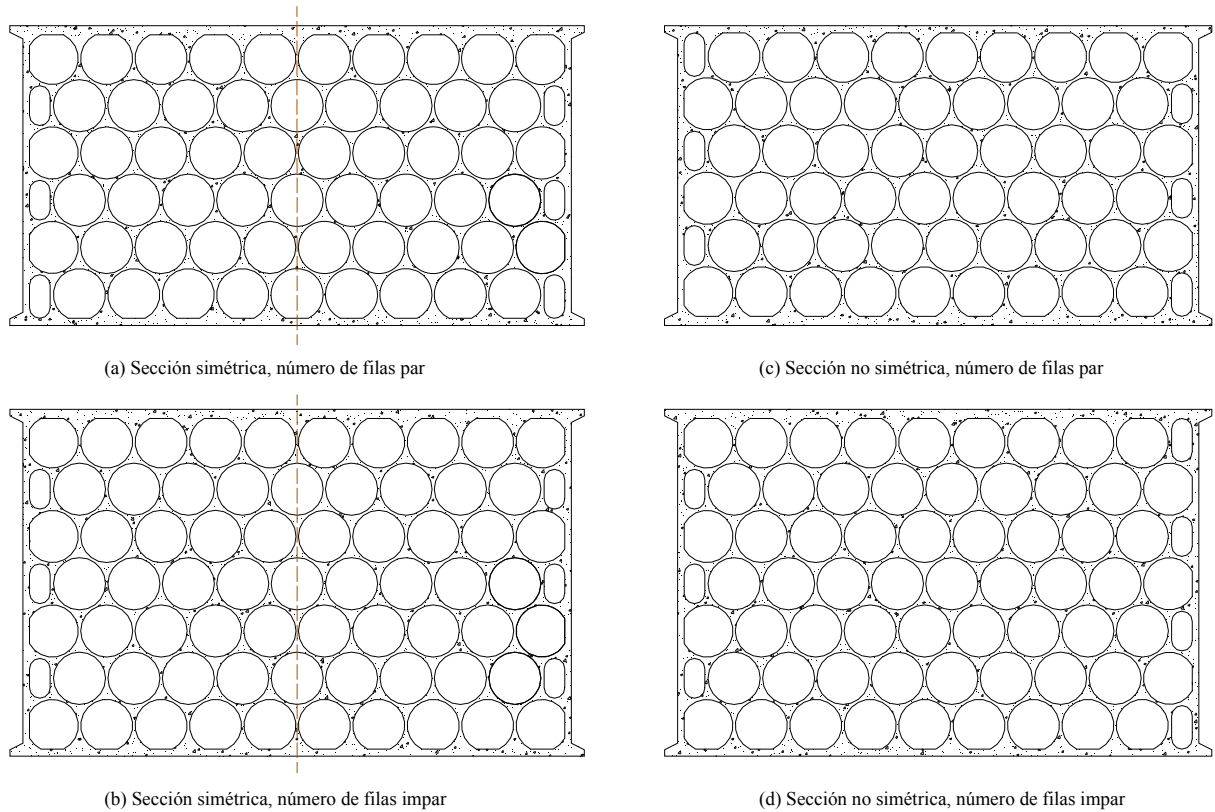


Figura III-13. Diseño de la sección transversal según criterios adoptados

- En secciones no simétricas de número par de filas, la primera celda del Lado Tierra y la última del Lado Mar deben ser *Circular de Esquina* y, como consecuencia, la última del Lado Tierra y la primera del Lado Mar serán *Ovoide de Esquina*, Figura III-13.c.
- En secciones no simétricas de número impar de filas, la primera celda del Lado Tierra y la primera del Lado Mar deben ser *Circular de Esquina* y, como consecuencia, la última del Lado Tierra y la última del Lado Mar serán *Ovoide de Esquina*, Figura III-13.d.

Para el algoritmo de posicionamiento de los aligeramientos mostrado en la **Figura III-15** y para el posterior conteo del número de celdas de cada tipología es conveniente definir las variables booleanas siguientes:

$$b_{Fil} = \begin{cases} 1 & \text{si } N_{Fil} \text{ es par} \\ 0 & \text{si } N_{Fil} \text{ es impar} \end{cases} \quad \text{(III-22)}$$

$$b_{Col} = \begin{cases} 1 & \text{si } N_{Col} \text{ es par} \\ 0 & \text{si } N_{Col} \text{ es impar} \end{cases} \quad \text{(III-23)}$$

$$b_{C2} = \begin{cases} 1 & \text{si } N_{Col} = 2 \\ 0 & \text{si } N_{Col} > 2 \end{cases} \quad \text{(III-24)}$$

Las variables negadas (denotadas como $\overline{b_{Fil}}$, $\overline{b_{Col}}$ y $\overline{b_{C2}}$) toman los valores opuestos. En la Figura III-14 se muestran cuatro combinaciones de filas y columnas pares e impares para un corte simétrico de la sección de cajones de diferentes tamaños.

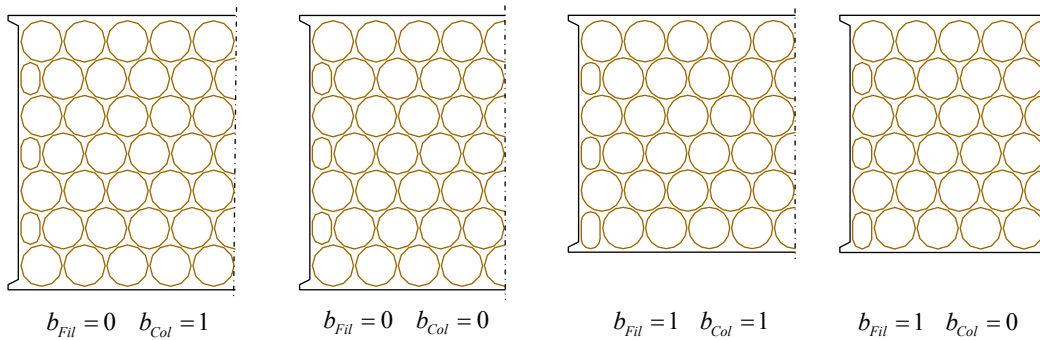


Figura III-14. Secciones con simetría transversal y filas y columnas pares o impares

En la **Tabla III-1** se indican las expresiones para obtener el número de celdas de cada una de las tipologías definidas anteriormente considerando que la sección transversal del fuste de un cajón típico no presenta simetría transversal (segunda columna), la presenta pero se estudia completa (tercera columna) y la presenta y únicamente se estudia una mitad de la misma (cuarta columna). La última fila de la tabla indica el número total de aligeramientos que presenta la sección.

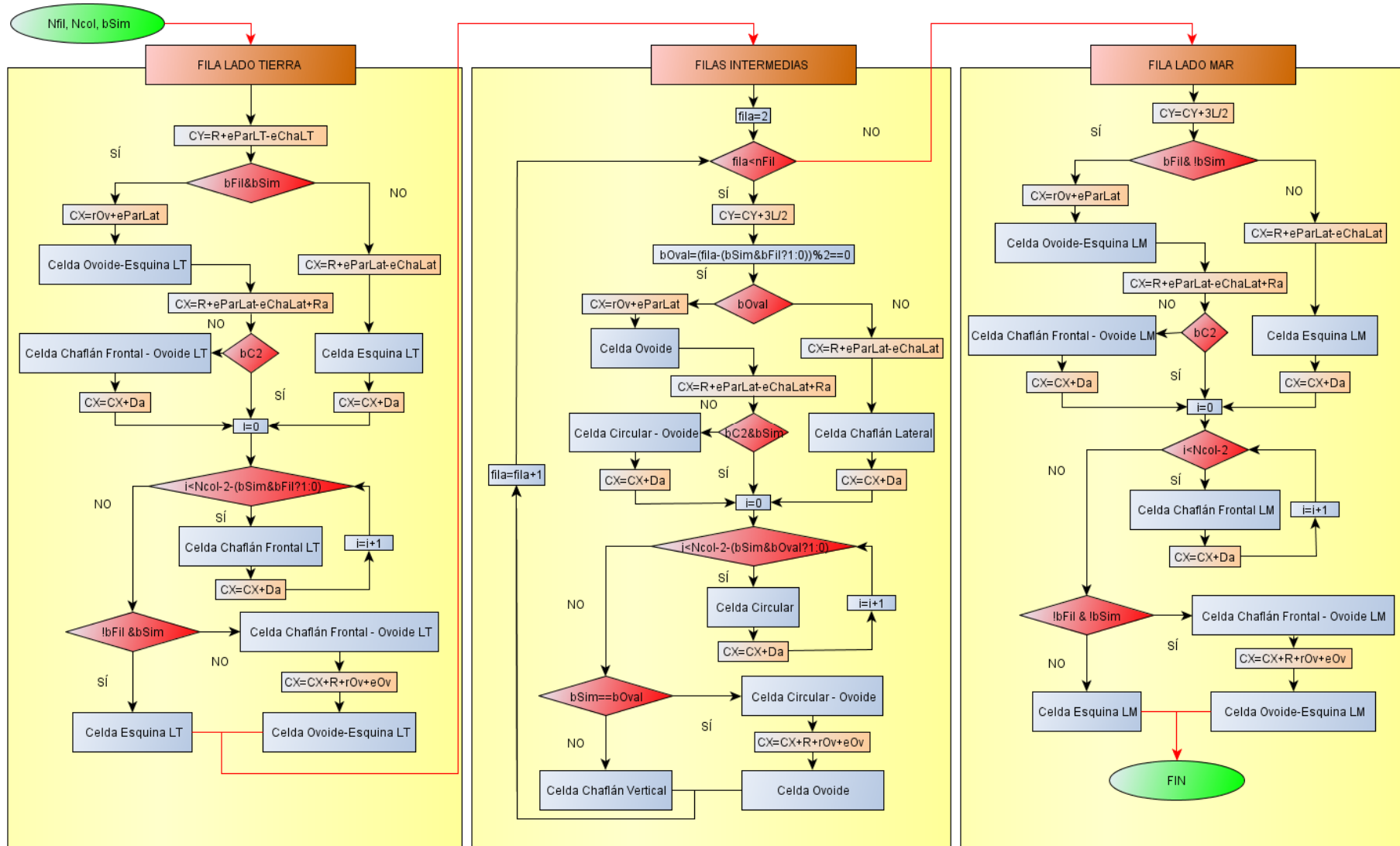


Figura III-15. Algoritmo de colocación de aligeramientos para definir la sección transversal por tipologías

3. Los Cajones Multicelulares

Tipo de Celda	Número de Celdas		
	Sección No Simétrica	Sección Simétrica	
		Completa	Media Sección
<i>Circular</i>	$(N_{Col} - 2) \cdot (N_{Fil} - 2)$	$\overline{b}_{C2} \cdot (N_{Col} \cdot (N_{Fil} - 2) - (5 \cdot N_{Fil} - 9 - b_{Fil}) / 2)$	$\overline{b}_{C2} \cdot (N_{Col} \cdot (N_{Fil} - 2) - 3 \cdot N_{Fil} + 6 - b_{Col} \cdot \overline{b}_{Fil}) / 2$
<i>Circular Mitad</i>	-	-	$\overline{b}_{C2} \cdot (N_{Fil} - 3 + b_{Fil} + 2 \cdot b_{Col} \cdot \overline{b}_{Fil}) / 2$
<i>Circular-Ovoide</i>	$N_{Fil} - 2$	$(1 + \overline{b}_{C2}) \cdot (N_{Fil} - 1 - b_{Fil}) / 2$	$\overline{b}_{C2} \cdot (N_{Fil} - 1 - b_{Fil}) / 2$
<i>Circular-Ovoide Mitad</i>	-	-	$b_{C2} \cdot (N_{Fil} - 1 - b_{Fil}) / 2$
<i>Ovoide</i>	$N_{Fil} - 2$	$N_{Fil} - 1 - b_{Fil}$	$(N_{Fil} - 1 - b_{Fil}) / 2$
<i>Ovoide-Esquina [LT//LM]</i>	$2 [1//1]$	$2 \cdot b_{Fil} [2 \cdot b_{Fil} // 0]$	$b_{Fil} [b_{Fil} // 0]$
<i>Circular con Chaflán Frontal [LT//LM]</i>	$2 \cdot (N_{Col} - 2)$ $[N_{Col} - 2 // N_{Col} - 2]$	$\overline{b}_{C2} \cdot (2 \cdot (N_{Col} - 2) - b_{Fil})$ $[\overline{b}_{C2} \cdot (N_{Col} - 2 - b_{Fil}) // N_{Col} - 2]$	$\overline{b}_{C2} \cdot (N_{Col} - 3 + b_{Col} \cdot \overline{b}_{Fil})$ $\overline{b}_{C2} \cdot [(N_{Col} - 3 + b_{Col}) / 2 - b_{Col} \cdot b_{Fil} // (N_{Col} - 3 - b_{Col}) / 2]$
<i>Circular con Chaflán Frontal Mitad [LT//LM]</i>	-	-	$\overline{b}_{C2} \cdot (b_{Col} \cdot (2 - b_{Fil}) + b_{Col} \cdot b_{Fil})$ $\overline{b}_{C2} \cdot [(b_{Col} \cdot b_{Fil} + \overline{b}_{Col} \cdot \overline{b}_{Fil}) // \overline{b}_{Col}]$
<i>Circular-Ovoide con Chaflán Frontal [LT//LM]</i>	$2 [1//1]$	$b_{Fil} \cdot (2 - b_{C2}) [b_{Fil} \cdot (2 - b_{C2}) // 0]$	$b_{Fil} \cdot \overline{b}_{C2} [b_{Fil} \cdot \overline{b}_{C2} // 0]$
<i>Circular-Ovoide con Chaflán Frontal Mitad</i>	-	-	$b_{Fil} \cdot b_{C2} [b_{Fil} \cdot b_{C2} // 0]$
<i>Circular con Chaflán Lateral</i>	$N_{Fil} - 2$	$N_{Fil} - 3 + b_{Fil}$	$(N_{Fil} - 3 + b_{Fil}) / 2$
<i>Circular de Esquina [LT//LM]</i>	$2 [1//1]$	$4 - 2 \cdot b_{Fil} [2 - 2 \cdot b_{Fil} // 2]$	$2 - b_{Fil} [\overline{b}_{Fil} // 1]$
Número Total de Celdas	$N_{Fil} \cdot (N_{Col} + 1)$	$\overline{b}_{C2} \cdot N_{Fil} \cdot (N_{Col} - 2) + (5 \cdot N_{Fil} - \overline{b}_{Fil}) / 2$	$\overline{b}_{C2} \cdot (N_{Fil} \cdot (N_{Col} - 2) + \overline{b}_{Col} \cdot \overline{b}_{Fil}) / 2 +$ $+ (3 \cdot N_{Fil} - \overline{b}_{Fil}) / 2$

Tabla III-1. Número de celdas por tipología en la sección del cajón

III.1.8.- Área aligerada de la sección transversal

Conocida la forma geométrica que tiene cada tipología de celdas, el área aligerada de la sección transversal puede ser calculada en función de las variables de entrada del problema concreto.

En la **Tabla III-2** se indica el área que le corresponde a cada tipo de aligeramiento.

Tipo de Celda	Área Aligerada
- Circular - Circular-Ovoide	$\pi \cdot R^2$
- Circular Mitad - Circular-Ovoide Mitad	$\pi \cdot \frac{R^2}{2}$
- Ovoide	$\pi \cdot r_{Ov}^2 + 2 \cdot r_{Ov} \cdot h_{Ov}$
- Ovoide-Esquina ¹	$\pi \cdot r_{Ov}^2 + 2 \cdot r_{Ov} \cdot h_{OvEsq}$
- Circular con Chaflán Frontal ¹ - Circular-Ovoide con Chaflán Frontal ¹	$R^2 \cdot \left(\pi - \theta_{Fro} + \frac{1}{2} \cdot \sin(2 \cdot \theta_{Fro}) \right)$
- Circular con Chaflán Frontal Mitad ¹ - Circular-Ovoide con Chaflán Frontal Mitad ¹	$\frac{R^2}{2} \cdot \left(\pi - \theta_{Fro} + \frac{1}{2} \cdot \sin(2 \cdot \theta_{Fro}) \right)$
- Circular con Chaflán Lateral	$R^2 \cdot \left(\pi - \theta_{Lat} + \frac{1}{2} \cdot \sin(2 \cdot \theta_{Lat}) \right)$
- Circular de Esquina ¹	$\frac{R^2}{4} \cdot \left((3 + b_{ArcEsq}) \cdot \pi + 4 \cdot \overline{b_{ArcEsq}} \cdot \cos(\theta_{Fro}) \cdot \cos(\theta_{Lat}) + (1 + b_{ArcEsq}) \cdot (\sin(2 \cdot \theta_{Fro}) + \sin(2 \cdot \theta_{Lat}) - 2 \cdot (\theta_{Fro} + \theta_{Lat})) \right)$

¹Los parámetros referidos en las celdas frontales serán los relativos al Lado Tierra o al Lado Mar, según corresponda

Tabla III-2. Área aligerada de cada tipología de celdas

Por último, el área total aligerada de la sección transversal se obtendrá sumando las áreas de cada tipología obtenidas al multiplicar el número de celdas (**Tabla III-1**) por el área de éstas según su tipo (**Tabla III-2**). En el caso de considerar únicamente la mitad de la sección, el área aligerada será el doble del valor obtenido.

$$A_{Aligerada} = \sum_{i=1}^{N_{Celdas}} A_{Celda,i} \quad \text{(III-25)}$$

Con este valor, y a partir de (III-9) se determina el grado de aligeramiento de la sección:

$$\text{Grado de Aligeramiento (\%)} = \frac{A_{\text{Aligerada}}}{A_{\text{maciza}}} \cdot 100 \quad \text{(III-26)}$$

Habitualmente, el grado de aligeramiento suele estar próximo al 75%.

III.1.9.- Centro de gravedad de la sección transversal

En la **Figura III-15**, además del tipo de celda a colocar, se han indicado las coordenadas $\{C_X, C_Y\}$ de cada aligeramiento, coordenadas que hacen referencia a los vértices de la malla de triángulos equiláteros de la **Figura III-2** sobre los que se dibujan las celdas circulares. En el caso de las celdas ovoidales, se refieren al centro del rectángulo de altura h_{Ov} . La **Figura III-16** ilustra estos puntos de referencia.

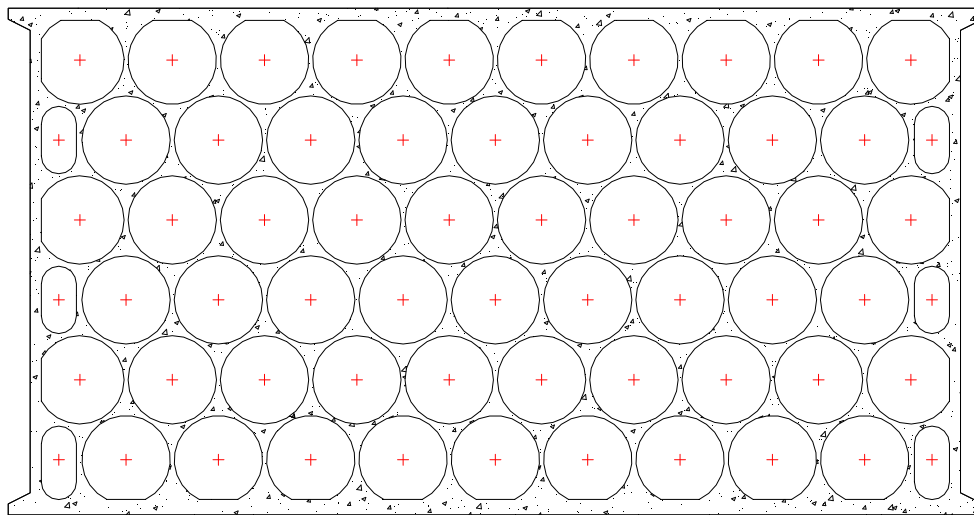


Figura III-16. Centros de referencia de los aligeramientos de la sección

El centro de gravedad del aligeramiento coincide con estos puntos de referencia en las celdas circulares y ovoidales, no así en el resto por las modificaciones realizadas en las proximidades de las paredes.

La **Tabla III-3** recoge los valores de las coordenadas del centro de gravedad de cada celda respecto de su centro de referencia.

Tipo de Celda	Coordenadas $\{CG_X, CG_Y\}$ del Centro de Gravedad
- Circular y Circular-Ovoide	$\{0,0\}$
- Circular Mitad y Circular-Ovoide Mitad	$\{0,0\}$
- Ovoide	$\{0,0\}$
- Ovoide-Esquina ¹	$\left\{0, \pm \left(r_{Ov} + \frac{h_{OvEsq}}{2} - R + e_{ChaFro} \right) \right\}$
- Circular con Chaflán Frontal ¹ , Circular-Ovoide con Chaflán Frontal ¹ , Circular con Chaflán Frontal Mitad ¹ y Circular-Ovoide con Chaflán Frontal Mitad ¹	$\left\{0, \pm \frac{2}{3 \cdot A_{ChaFro}} \cdot (R \cdot \sin(\theta_{Fro}))^3 \right\}$
- Circular con Chaflán Lateral ²	$\left\{ \pm \frac{2}{3 \cdot A_{ChaLat}} \cdot (R \cdot \sin(\theta_{Lat}))^3, 0 \right\}$
- Circular de Esquina ^{1,2}	$\left. \begin{aligned} & \frac{R^3}{6 \cdot A_{Esquina}} \cdot \left\{ \pm \left(\frac{4 \cdot b_{ArcEsq} \cdot \sin(\theta_{Lat})^3 +}{b_{ArcEsq}} \cdot (2 \cdot \sin(\theta_{Fro}) - \cos(\theta_{Lat})) \cdot (\cos(\theta_{Lat}) + \sin(\theta_{Fro}))^2 \right) \right\} \\ & , \pm \left(\frac{4 \cdot b_{ArcEsq} \cdot \sin(\theta_{Fro})^3 +}{b_{ArcEsq}} \cdot (2 \cdot \sin(\theta_{Lat}) - \cos(\theta_{Fro})) \cdot (\cos(\theta_{Fro}) + \sin(\theta_{Lat}))^2 \right) \end{aligned} \right\}$

¹Los parámetros referidos en las celdas frontales serán los relativos al Lado Tierra o al Lado Mar, según corresponda. El signo de la coordenada Y es positivo en el Lado Tierra, negativo en el Lado Mar.
²El signo de la coordenada X es positivo en la pared lateral izquierda, negativo en la pared lateral derecha.

Tabla III-3. Centros de gravedad de los aligeramientos por tipología de celdas respecto de su punto de referencia

A partir del algoritmo de posicionamiento de la **Figura III-15**, en el que se establece de manera ordenada y de una en una cada tipo de celda a ubicar, empezando por la esquina izquierda del Lado Tierra, junto con su coordenada $\{C_X, C_Y\}$ de referencia, es posible calcular el centro de gravedad de la sección transversal empleando la **Tabla III-2** para conocer el área de la celda y la **Tabla III-3** para corregir su centro de gravedad:

$$A_{Sección}^{Fuste} = A_{Maciza}^{Fuste} - A_{Aligerada}^{Fuste} \tag{III-27}$$

$$CG_{Sección}^{Fuste} = \frac{1}{A_{Sección}^{Fuste}} \left(A_{Maciza}^{Fuste} \cdot \{CG_{X,maciza}^{Fuste}, CG_{Y,maciza}^{Fuste}\} - \sum_{i=1}^{N_{Celdas}} A_{Celda,i} \cdot \{C_{X,i} + CG_{X,i}, C_{Y,i} + CG_{Y,i}\} \right) \tag{III-28}$$

III.1.10.- Momentos de Inercia de la sección transversal aligerada

Para el cálculo de la estabilidad naval es preciso conocer la inercia de la sección transversal aligerada (en el caso de que las celdas estén comunicadas) o de cada una de las celdas (si éstas son estancas y no están comunicadas).

La **Tabla III-3** recoge los valores de inercia de cada tipo de celda respecto de su centro de gravedad.

Tipo de Celda	Momentos de inercia $\{I_Y, I_X\}$
Circular y Circular-Ovoide	$\left\{ \pi \cdot \frac{R^4}{4}, \pi \cdot \frac{R^4}{4} \right\}$
Ovoide $h = h_{Ov}$ Ovoide-Esquina ¹ $h = h_{OvEsqLT}, h_{OvEsqLM}$	$\left\{ \pi \cdot \frac{r_{Ov}^4}{4} + \frac{2}{3} \cdot r_{Ov}^3 \cdot h, \right.$ $\left. \pi \cdot \frac{r_{Ov}^4}{4} + \pi \cdot r_{Ov}^2 \cdot h \cdot \left(\frac{4 \cdot r_{Ov}}{3 \cdot \pi} + \frac{h}{4} \right) + \frac{1}{6} \cdot r_{Ov} \cdot h^3 \right\}$
Circular con Chaflán Frontal ¹ , Circular-Ovoide con Chaflán Frontal ¹ $\theta_{Fro} = \theta_{FroLT}, \theta_{FroLM}$	$I_y = \frac{R^4}{4} \cdot \left(\pi - \theta_{Fro} + \frac{2}{3} \cdot \sin(2 \cdot \theta_{Fro}) - \frac{1}{12} \cdot \sin(4 \cdot \theta_{Fro}) \right) - A_{Fro} \cdot CG_{X,Fro}^2$ $I_x = \frac{R^4}{4} \cdot \left(\pi - \theta_{Fro} + \frac{1}{4} \cdot \sin(4 \cdot \theta_{Fro}) \right) - A_{Fro} \cdot CG_{Y,Fro}^2$
Circular con Chaflán Lateral $\theta_{Lat}, \theta_{Fro} = 0$	$I_y = \frac{R^4}{4} \cdot \left(\pi - \theta_{Lat} + \frac{1}{4} \cdot \sin(4 \cdot \theta_{Lat}) \right) - A_{Lat} \cdot CG_{X,Lat}^2$ $I_x = \frac{R^4}{4} \cdot \left(\pi - \theta_{Lat} + \frac{2}{3} \cdot \sin(2 \cdot \theta_{Lat}) - \frac{1}{12} \cdot \sin(4 \cdot \theta_{Lat}) \right) - A_{Lat} \cdot CG_{Y,Lat}^2$
Circular de Esquina ¹ $b_{ArcEsq} = 1, \theta_{Lat} >$ $\theta_{Fro} = \theta_{FroLT}, \theta_{FroLM}$	$I_y = \frac{R^4}{4} \cdot \left(\pi - \theta_{Fro} - \theta_{Lat} + \frac{1}{4} \cdot \sin(4 \cdot \theta_{Lat}) + \frac{2}{3} \cdot \sin(2 \cdot \theta_{Fro}) - \frac{1}{12} \cdot \sin(4 \cdot \theta_{Fro}) \right)$ $- A_{Esq} \cdot CG_{X,Esq}^2$ $I_x = \frac{R^4}{4} \cdot \left(\pi - \theta_{Fro} - \theta_{Lat} + \frac{1}{4} \cdot \sin(4 \cdot \theta_{Fro}) + \frac{2}{3} \cdot \sin(2 \cdot \theta_{Lat}) - \frac{1}{12} \cdot \sin(4 \cdot \theta_{Lat}) \right)$ $- A_{Esq} \cdot CG_{Y,Esq}^2$
Circular de Esquina ¹ $b_{ArcEsq} = 0, \theta_{Lat} >$ $\theta_{Fro} = \theta_{FroLT}, \theta_{FroLM}$	$I_y = \frac{R^4}{96} \cdot \left(18 \cdot \pi - 12 \cdot (\theta_{Fro} + \theta_{Lat}) + 3 \cdot \sin(4 \cdot \theta_{Lat}) + 6 \cdot \sin(2 \cdot \theta_{Fro}) \right)$ $+ 8 \cdot \cos(\theta_{Fro}) \cdot \left(4 \cdot \cos(\theta_{Lat})^3 + \sin(\theta_{Fro})^3 \right) - A_{Esq} \cdot CG_{X,Esq}^2$ $I_x = \frac{R^4}{96} \cdot \left(18 \cdot \pi - 12 \cdot (\theta_{Fro} + \theta_{Lat}) + 3 \cdot \sin(4 \cdot \theta_{Fro}) + 8 \cdot \sin(2 \cdot \theta_{Lat}) - \sin(4 \cdot \theta_{Lat}) \right)$ $+ 32 \cdot \cos(\theta_{Fro})^3 \cdot \cos(\theta_{Lat}) - A_{Esq} \cdot CG_{Y,Esq}^2$

¹Los parámetros referidos a las celdas frontales serán los relativos al Lado Tierra o al Lado Mar, según corresponda.

Tabla III-4. Momentos de inercia de los aligeramientos por tipología de celdas respecto de su centro de gravedad

Por tanto, en el caso de que las celdas sean estancas y no haya comunicación entre las mismas, la inercia del conjunto de celdas, tomada desde el centro de gravedad de cada una será:

$$I_{Celdas\ Estancas}^{Lastre} = \sum_{i=1}^{N_{Celdas}} \{I_{X,i}, I_{Y,i}\} \quad \text{(III-29)}$$

Mientras que en el caso de que haya comunicación, la inercia del conjunto de celdas será:

$$I_{Celdas\ Comunicadas}^{Lastre} = \sum_{i=1}^{N_{Celdas}} \left\{ I_{X,i} + A_i \cdot (CG_{Y,i} - CG_{Y,Maciza}^{Fuste})^2, I_{Y,i} + A_i \cdot (CG_{X,i} - CG_{X,Maciza}^{Fuste})^2 \right\} \quad \text{(III-30)}$$

III.1.11.- Determinación de las variables de la sección

De todas las variables involucradas en la determinación completa de la sección transversal, algunas vienen impuestas en el Proyecto de la Autoridad Portuaria, otras vienen fijadas por condicionantes constructivos o basadas en la experiencia, y el resto deben ser determinadas, tomando valores discretos dentro de un rango admisible o cualquier valor resultante.

La manga viene determinada por las condiciones resistentes exigibles a la estructura (muelle o dique) en que se van a utilizar los cajones, o puede fijarse por condiciones de estabilidad naval, calado máximo admisible, o condicionantes de explotación o continuidad con líneas ya existentes.

La eslora depende entre otros factores de la capacidad del dique flotante o instalación donde se vayan a fabricar los cajones, de los condicionantes marítimos para su remolque y fondeo (corrientes, oleaje, viento, etc.) y de los condicionantes impuestos por la posibilidad de asientos diferenciales del cimiento.

Para la resolución completa de la geometría de la sección se dispone únicamente de las dos ecuaciones de las dimensiones exteriores de eslora (III-3) y manga (III-4). En función de las incógnitas, las posibles situaciones que se pueden presentar se dividen en dos tipos:

- Dimensiones exteriores desconocidas con todas de las dimensiones interiores conocidas o alguna dentro de un rango posible de valores.
- Dimensiones exteriores conocidas o dentro de un rango posible de valores, con alguna de las dimensiones interiores desconocidas o dentro de un rango posible de valores.

Eslora y Manga son desconocidas

En este caso habrá una solución para cualquier rango de valores de los pares siguientes: (N_{Fil}, N_{Col}) , (D, e_{Cir}) , una vez se hayan fijado los valores de los espesores

3. Los Cajones Multicelulares

de las paredes. En este caso, no es preciso achaflanar las celdas exteriores, y se considerarán nulos, salvo que tengan un valor prefijado.

Así, iterando con todos los posibles valores de estas cuatro variables, y fijando la condición de simetría o no, se tendrán las siguientes soluciones:

$$\left\{ \begin{array}{l} Eslora_i = (D_k + e_{Cir,l}) \cdot \left(N_{Col,m} + \frac{\overline{b_{sim,n}}}{2} \right) - e_{Cir,l} + 2 \cdot (e_{Lat} - e_{ChaLat}) \\ Manga_j = \frac{3 \cdot L_o}{2} \cdot (N_{Fil,p} - 1) + D_k + e_{LTierra} - e_{ChaFroLT} + e_{LMar} - e_{ChaFroLM} \end{array} \right. \quad \text{(III-31)}$$

$k = 1 \dots n_D, l = 1 \dots n_{e_{Cir}}, m = 1 \dots n_{N_{Col}}, n = 1, 2, o = 1 \dots n_D \cdot n_{e_{Cir}}, p = 1 \dots n_{N_{Fil}}$
 $i = 1 \dots 2 \cdot n_D \cdot n_{e_{Cir}} \cdot n_{N_{Col}}, j = 1 \dots n_D \cdot n_{e_{Cir}} \cdot n_{N_{Fil}}$
 $n_{Soluciones} = 2 \cdot n_D \cdot n_{e_{Cir}} \cdot n_{N_{Col}} \cdot n_{N_{Fil}}$

Dimensiones interiores desconocidas

Además de conocer las dimensiones exteriores (fijas o dentro de un rango de valores conocido), habitualmente alguna de las variables internas de la sección será conocida. Por ejemplo, el diámetro de las celdas suele venir marcado por el del encofrado que se emplea durante la construcción del cajón, y que es reutilizado sin variación en sucesivos proyectos, **Figura III-17**, y está comprendido entre 2.5 y 3.80 m, el espesor de las gargantas circulares suele estar prefijado entre 15 y 30 cm, el espesor de las paredes exteriores toma valores entre 0.2 y 0.60 m [5].



Figura III-17. Acopio de encofrados usados previos a colocación en pontona

En el caso de que el número de celdas sea indeterminado, con los valores límites anteriores, y dado que filas y columnas sólo tomarán valores enteros, se pueden estimar unas cotas mínimas y máximas para el rango de valores posibles del número de celdas:

$$\begin{aligned}
 rg_{N_{Col}} &= \left[\max\left(2, Ent\left(\frac{Eslora}{5.1}\right)\right), Ent\left(\frac{Eslora}{2.7}\right) + 1 \right] \\
 rg_{N_{Fil}} &= \left[\max\left(2, Ent\left(\frac{Manga}{5}\right)\right), Ent\left(\frac{Manga}{2.3}\right) + 1 \right]
 \end{aligned}
 \tag{III-32}$$

Por otro lado, los valores de diámetro de celdas y espesor de gargantas, en caso de que no sean conocidos, se fijan con una precisión máxima de 1 cm, dentro de los límites anteriormente especificados. Igualmente, las paredes exteriores tomarán valores dentro de los límites indicados, con precisión máxima de 1 cm.

Por tanto, para un rango de valores $(Eslora, Manga)$, fijado el grado de achaflanamiento máximo admisible en las paredes laterales y/o frontales, el problema se plantea de manera iterativa buscando soluciones para valores discretos de (N_{Fil}, N_{Col}) , (D, e_{Cir}) y (e_{Lat}, e_{Fro}) dentro de sus rangos concretos, si están definidos, o de los anteriormente indicados en caso contrario, calculando el ajuste de achaflanamiento requerido dentro de los valores máximos fijados. Por ello, no habrá solución en todos los casos.

$$\begin{cases}
 \Delta_{Lat} = \left[(D_k + e_{Cir,l}) \cdot \left(N_{Col,m} + \frac{b_{sim,n}}{2} \right) - e_{Cir,l} + 2 \cdot e_{Lat,o} \right] - Eslora_i \\
 \Delta_{Fro} = \left[\frac{\sqrt{3} \cdot (D_k + e_{Cir,l})}{2} \cdot (N_{Fil,p} - 1) + D_k + e_{LTierra,q} + e_{LMar,r} \right] - Manga_j
 \end{cases}
 \tag{III-33}$$

$$si \ 0 \leq \Delta_{Lat} \leq 2 \cdot e_{ChaLat,max} \quad y \quad 0 \leq \Delta_{Fro} \leq e_{ChaFroLM,max} + e_{ChaFroLT,max} \Rightarrow$$

$$n_{Soluciones} = n_{Soluciones} + 1$$

$$i = 1 \dots n_{Eslora}, \quad j = 1 \dots n_{Manga}, \quad k = 1 \dots n_D, \quad l = 1 \dots n_{e_{Cir}}, \quad m = 1 \dots n_{N_{Col}}, \quad n = 1, 2$$

$$o = 1 \dots n_{e_{Lat}}, \quad p = 1 \dots n_{N_{Fil}}, \quad q = 1 \dots n_{e_{LTierra}}, \quad r = 1 \dots n_{e_{LMar}}$$

III.1.12.- Volumen

Hasta el momento se ha definido la sección horizontal del fuste. Su extrusión en vertical permite obtener el cuerpo principal del cajón. El resto de elementos que completan el cajón son la losa, con posibilidad de voladizos y cartabones, el muro perimetral y la superestructura o viga cantil.

III.1.12.1.- Losa

La losa o solera es el elemento inferior de cierre. Elemento plano, macizo y de planta rectangular, presenta cantos de 0.4 a 1 m.

Habitualmente se añaden zapatas o voladizos en los lados Tierra y Mar, a fin de ganar más estabilidad y superficie de apoyo. Tienen el mismo canto que la losa y un vuelo entre 0.5 y 1.5 m.

Respecto al origen de coordenadas de la **Figura III-5**, el centro de gravedad de la losa viene dado por la ecuación (III-35).

$$A_{Losa} = A_{Maciza}^{Fuste} + (Vuelo_{Tierra} + Vuelo_{Mar}) \cdot (Eslora + 2 \cdot L_{Tope}) \quad (III-34)$$

$$CG_{Losa} = \left\{ \frac{Eslora}{2}, \frac{1}{A_{Losa}} \left(A_{Maciza}^{Fuste} \cdot CG_{Y,Maciza}^{Fuste} + (Eslora + 2 \cdot L_{Tope}) \cdot \left(Vuelo_{Mar} \cdot \left(Manga + \frac{Vuelo_{Mar}}{2} \right) - \frac{(Vuelo_{Tierra})^2}{2} \right) \right) \right\} \quad (III-35)$$

En algunos casos es posible suavizar la transición losa-fuste en los voladizos con la introducción de cartabones. Se definen por el triángulo rectángulo de base paralela al voladizo y de altura paralela a la pared del Lado Tierra y Lado Mar.

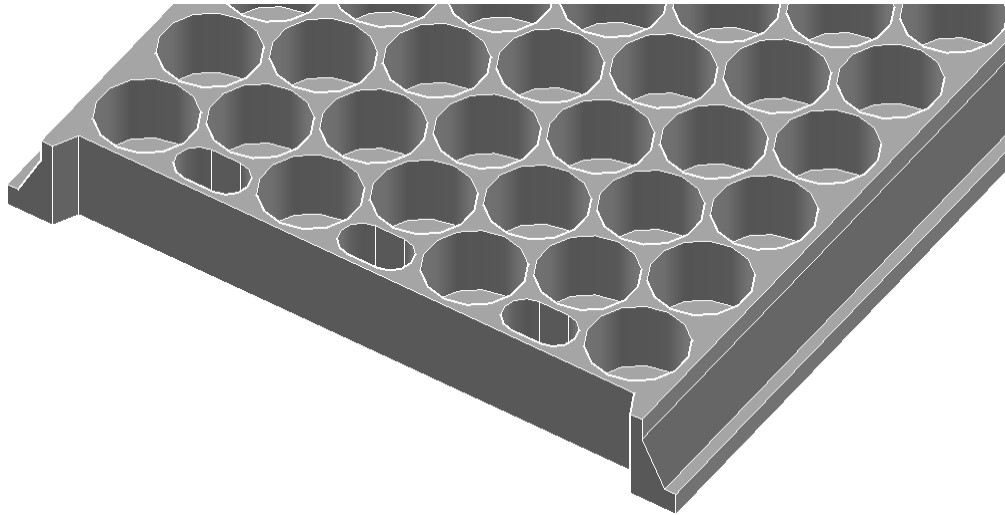


Figura III-18. Losa de cierre inferior, con voladizos y cartabones

El volumen de cada cartabón viene dado por la siguiente expresión:

$$V_{Cartabon} = \frac{1}{2} \cdot B_{Cartabon} \cdot H_{Cartabon} \cdot (Eslora + 2 \cdot L_{Tope}) \quad (III-36)$$

III.1.12.2.- Muro Perimetral

En algunas ocasiones, en la coronación del fuste se realiza una prolongación de las paredes exteriores del cajón, dando lugar a un cierre o muro perimetral de una altura de hasta 0.5 m.

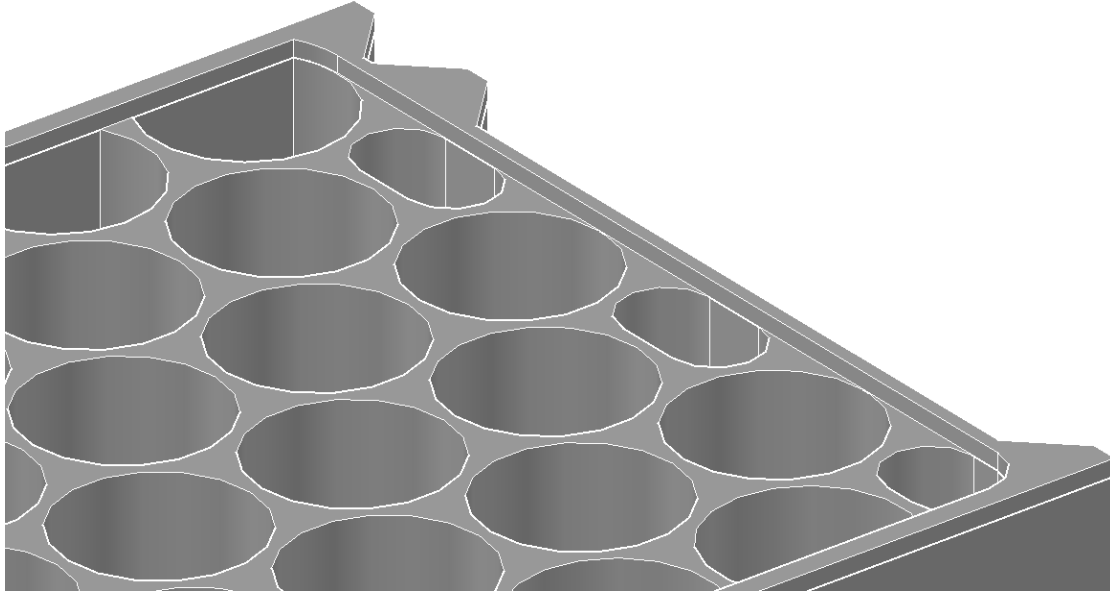


Figura III-19. Cajón con muro perimetral. Muelle del Prat Fase I-A, Puerto de Barcelona, 2005

Respecto al origen de coordenadas de la **Figura III-5**, el área y las coordenadas del centro de gravedad del murete vienen dadas por las siguientes ecuaciones:

$$\begin{aligned}
 A_{Esq}^1[\phi_{Fro}] &:= \frac{b_{ArcEsq} \cdot R^2}{4} \cdot (4 \cos(\phi_{Fro}) \cos(\phi_{Lat}) - \sin(2 \cdot \phi_{Fro}) - \sin(2 \cdot \phi_{Lat}) - \pi + 2 \cdot (\phi_{Fro} + \phi_{Lat})) \\
 A_{Esq,LT} &= (1 + (1 - 2 \cdot b_{Fil}) \cdot b_{Sim}) \cdot A_{Esq}^1(\phi_{LT}) \quad A_{Esq,LM} = (1 + b_{Sim}) \cdot A_{Esq}^1(\phi_{LM}) \\
 A_{OvEsq}^1 &:= \frac{r_{Ov}^2}{2} \cdot (4 - \pi) \quad A_{OvEsq} = (1 - (1 - b_{Fil}) \cdot b_{Sim}) \cdot A_{OvEsq}^1 \\
 A_{Murete} &= A_{Maciza}^{Fuste} - (Manga - e_{LMar} - e_{LTierra}) \cdot (Eslora - 2 \cdot e_{Lat}) + A_{Esq,LT} + A_{Esq,LM} + A_{OvEsq} \\
 CG_{X,Esq}^1[\phi_{Fro}] &:= \left(\frac{1}{2} \cdot \cos(\phi_{Fro})^2 \cdot \cos(\phi_{Lat}) - \frac{1}{3} \cdot \cos(\phi_{Fro})^2 \cdot \sin(\phi_{Fro}) - \right. \\
 &\quad \left. - \frac{1}{6} \cdot \cos(\phi_{Lat}) \cdot \sin(\phi_{Lat})^2 - \frac{1}{3} \cdot (\cos(\phi_{Lat}) - \sin(\phi_{Fro})) \right) \cdot \frac{R^3}{A_{Esq}^1(\phi_{Fro})} \\
 CG_{X,Esq,LT} &= e_{LT} + R \cdot \cos(\phi_{LT}) - CG_{Y,Esq}^1(\phi_{LT}) \\
 CG_{X,Esq,LM} &= Manga - e_{LM} - R \cdot \cos(\phi_{LM}) + CG_{Y,Esq}^1(\phi_{LM}) \\
 CG_{X,OvEsq}^1 &:= \frac{r_{Ov}^3}{6 \cdot A_{OvEsq}^1} \quad CG_{Y,OvEsq} = (r_{Ov} - CG_{Y,OvEsq}^1) \cdot b_{Sim} + \\
 &\quad + \frac{1}{2} \cdot (Manga - e_{LM}) \cdot (1 - b_{Sim}) + \frac{1}{2} \cdot e_{LT} \cdot (1 - b_{Sim}) \\
 CG_{X,Murete} &= \left(A_{Maciza}^{Fuste} \cdot CG_{X,Maciza}^{Fuste} - \right. \\
 &\quad - (Manga - e_{LMar} - e_{LTierra}) \cdot (Eslora - 2 \cdot e_{Lat}) \cdot \frac{Eslora}{2} + \\
 &\quad \left. + A_{Esq,LT} \cdot CG_{X,Esq,LT} + A_{Esq,LM} \cdot CG_{X,Esq,LM} + A_{OvEsq} \cdot CG_{X,OvEsq} \right) \cdot \frac{1}{A_{Murete}}
 \end{aligned} \tag{III-37}$$

$$\begin{aligned}
 CG_{X,OvEsq}^1 &:= \frac{r_{Ov}^3}{6 \cdot A_{OvEsq}^1} \quad CG_{Y,OvEsq} = (r_{Ov} - CG_{Y,OvEsq}^1) \cdot b_{Sim} + \\
 &\quad + \frac{1}{2} \cdot (Manga - e_{LM}) \cdot (1 - b_{Sim}) + \frac{1}{2} \cdot e_{LT} \cdot (1 - b_{Sim}) \\
 CG_{X,Murete} &= \left(A_{Maciza}^{Fuste} \cdot CG_{X,Maciza}^{Fuste} - \right. \\
 &\quad - (Manga - e_{LMar} - e_{LTierra}) \cdot (Eslora - 2 \cdot e_{Lat}) \cdot \frac{Eslora}{2} + \\
 &\quad \left. + A_{Esq,LT} \cdot CG_{X,Esq,LT} + A_{Esq,LM} \cdot CG_{X,Esq,LM} + A_{OvEsq} \cdot CG_{X,OvEsq} \right) \cdot \frac{1}{A_{Murete}}
 \end{aligned} \tag{III-38}$$

$$\begin{aligned}
 CG_{Y,Esq}^1[\phi_{Fro}] &:= \left(\frac{1}{2} \cdot \cos(\phi_{Fro})^2 \cdot \cos(\phi_{Lat}) - \frac{1}{3} \cdot \cos(\phi_{Fro})^2 \cdot \sin(\phi_{Fro}) - \right. \\
 &\quad \left. - \frac{1}{6} \cdot \cos(\phi_{Lat}) \cdot \sin(\phi_{Lat})^2 - \frac{1}{3} \cdot (\cos(\phi_{Lat}) - \sin(\phi_{Fro})) \right) \cdot \frac{R^3}{A_{Esq}^1(\phi_{Fro})} \\
 CG_{Y,Esq,LT} &= e_{LT} + R \cdot \cos(\phi_{LT}) - CG_{Y,Esq}^1(\phi_{LT}) \\
 CG_{Y,Esq,LM} &= Manga - e_{LM} - R \cdot \cos(\phi_{LM}) + CG_{Y,Esq}^1(\phi_{LM}) \\
 CG_{Y,OvEsq}^1 &:= \frac{r_{Ov}^3}{6 \cdot A_{OvEsq}^1} \quad CG_{Y,OvEsq} = (r_{Ov} - CG_{Y,OvEsq}^1) \cdot b_{Sim} + \\
 &\quad + \frac{1}{2} \cdot (Manga - e_{LM}) \cdot (1 - b_{Sim}) + \frac{1}{2} \cdot e_{LT} \cdot (1 - b_{Sim}) \\
 CG_{Y,Murete} &= \left(A_{Maciza}^{Fuste} \cdot CG_{Y,Maciza}^{Fuste} - \right. \\
 &\quad - (Manga - e_{LMar} - e_{LTierra}) \cdot (Eslora - 2 \cdot e_{Lat}) \cdot \frac{(Manga - e_{LMar} + e_{LTierra})}{2} + \\
 &\quad \left. + A_{Esq,LT} \cdot CG_{Y,Esq,LT} + A_{Esq,LM} \cdot CG_{Y,Esq,LM} + A_{OvEsq} \cdot CG_{Y,OvEsq} \right) \cdot \frac{1}{A_{Murete}}
 \end{aligned} \tag{III-39}$$

III.1.12.3.- Estructura final

El cajón queda definido por los elementos que finalmente se añadan al fuste y losa (voladizos, cartabones, muro perimetral).

El peso y el centro de gravedad serán datos de interés para el posterior estudio de estabilidad naval que garantice el proceso constructivo y posterior transporte del cajón.

Las siguientes ecuaciones permiten el cálculo de estos valores, ubicando los ejes de coordenadas según la **Figura III-5**, en la cota $Z = 0$ en la base apoyada del cajón.

$$\begin{aligned}
 V_{Cajon} &= A_{Losa} \cdot H_{Losa} + A_{Seccion}^{Fuste} \cdot H_{Fuste} + A_{Murete} \cdot H_{Murete} + 2 \cdot V_{Cartabon} \\
 P_{Cajon} &= \rho_{Hormigon} \cdot g \cdot V_{Cajon} \\
 CG_{Cajon} &= \frac{1}{V_{Cajon}} \cdot \left(A_{Losa} \cdot H_{Losa} \cdot \left\{ CG_{X,Losa}, CG_{Y,Losa}, \frac{H_{Losa}}{2} \right\} + \right. \\
 &\quad + A_{Seccion}^{Fuste} \cdot H_{Fuste} \cdot \left\{ CG_{X,Seccion}^{Fuste}, CG_{Y,Seccion}^{Fuste}, H_{Losa} + \frac{H_{Fuste}}{2} \right\} + \\
 &\quad + A_{Murete} \cdot H_{Murete} \cdot \left\{ CG_{X,Murete}, CG_{Y,Murete}, H_{Losa} + H_{Fuste} + \frac{H_{Murete}}{2} \right\} + \\
 &\quad \left. + 2 \cdot V_{Cartabon} \cdot \left\{ \frac{Eslora}{2}, \frac{Manga}{2}, H_{Losa} + \frac{H_{Cartabon}}{3} \right\} \right)
 \end{aligned} \tag{III-41}$$

Nótese que en las fórmulas anteriores se puede sustituir la altura total del fuste por la altura construida y así tener el volumen, el peso y la posición del centro de gravedad del cajón en todas y cada una de las etapas constructivas.

III.1.12.4.- Superestructura

Habitualmente el cajón se cierra superiormente con una losa de entre 1 y 2 m de canto, denominada superestructura.



Figura III-20. Superestructura en línea de cajones ya fondeada (izquierda) junto a nuevo cajón a fondear (derecha)

En otras ocasiones se emplean vigas cantil. Viga cantil o vigas de atado de cajones flotantes son vigas de hormigón armado, construidas *in situ*, que cierran el extremo superior del muelle en el Lado Mar, otorgando monolitismo al muelle, al hacer trabajar solidariamente a los distintos elementos a medida que se van atando los cajones entre sí. Además, sirven para aminorar el efecto de los asentamientos diferenciales entre cajones, para el cierre de la explanada y la correcta alineación del muelle [78].

A nivel de servicio, permiten el anclaje de bolardos, defensas, escalas, etc, y sirven de guía y soporte de los carriles de la grúa, lo cual determinará una serie de hipótesis de carga a considerar en el estudio resistente del cajón.

Desde el punto de vista de cálculo, es habitual considerar cajón y viga cantil como entes diferenciados, por lo que no se suele tratar dicha viga como parte del cajón. Si bien, es posible que en circunstancias especiales se deba integrar dicha viga con el cajón y, en las fases de servicio, realizar su cálculo resistente y, consecuentemente, la determinación de su armado.

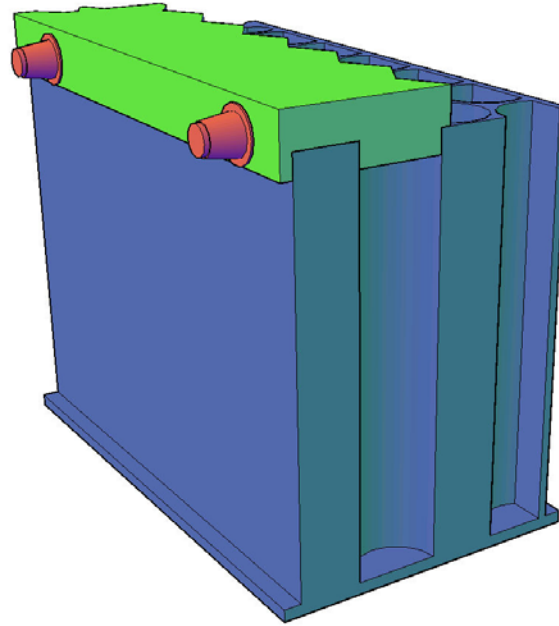


Figura III-21. Viga cantil integrada en el cajón. Muelle del Prat, Fase II, Puerto de Barcelona, 2009

III.1.13.- Estabilidad naval

La seguridad del proceso constructivo de un cajón se define fundamentalmente por criterios de estabilidad del cajón y del conjunto del parque de prefabricación flotante. El objetivo del dimensionado del cajón es buscar un equilibrio entre una suficiente flotación y una estabilidad mínima, para poder transportarlo flotando a su posición definitiva.

III.1.13.1.- Estabilidad inicial

Partiendo de una situación inicial en la que el cajón se encuentra flotando en el agua, es preciso determinar su posición, en condiciones de estabilidad inicial, en equilibrio y sin escoras, y considerando además la práctica habitual de adición de lastre para aumento de la estabilidad naval, consistente en el relleno de todas las celdas con agua de mar hasta una cierta altura. En este caso, el peso total del cajón se incrementa según el peso de dicha altura de agua. Igualmente, el centro de gravedad del conjunto se ve modificado, ecuación (III-42).

$$\begin{aligned}
 P_{Total} &= P_{Cajon} + \rho_{Agua} \cdot g \cdot A_{Aligerada}^{Fuste} \cdot H_{Agua} \\
 CG_{Calado}^{Cajon} &= \frac{1}{P_{Total}} \cdot \left(\rho_{Hormigon} \cdot g \cdot V_{Cajon} \cdot CG_{Cajon} + \right. \\
 &\quad \left. + \rho_{Agua} \cdot g \cdot A_{Aligerada}^{Fuste} \cdot H_{Agua} \cdot \left\{ CG_{X,Aligerada}^{Fuste}, CG_{Y,Aligerada}^{Fuste}, H_{Losa} + \frac{H_{Agua}}{2} \right\} \right)
 \end{aligned}
 \tag{III-42}$$

Conforme al principio de Arquímedes, el empuje del mar sobre el cajón es igual al peso del volumen de fluido desalojado por dicho cajón. Dicha fuerza se aplica en el centro de dicho volumen, al que se denomina *centro de carena*.

$$\begin{aligned}
 V_{Carena} &= A_{Losa} \cdot H_{Losa} + 2 \cdot V_{Cartabon} + A_{Maciza}^{Fuste} \cdot H_{Calado}^{Fuste} \\
 Empuje &= \rho_{Agua} \cdot g \cdot V_{Carena} \\
 CG_{Carena} &= \frac{1}{V_{Carena}} \cdot \left(A_{Losa} \cdot H_{Losa} \cdot \left\{ CG_{X,Losa}, CG_{Y,Losa}, \frac{H_{Losa}}{2} \right\} + \right. \\
 &+ A_{Maciza}^{Fuste} \cdot H_{Calado}^{Fuste} \cdot \left\{ CG_{X,Maciza}^{Fuste}, CG_{Y,Maciza}^{Fuste}, H_{Losa} + \frac{H_{Calado}^{Fuste}}{2} \right\} + \\
 &\left. + 2 \cdot V_{Cartabon} \cdot \left\{ \frac{Eslora}{2}, \frac{Manga}{2}, H_{Losa} + \frac{H_{Cartabon}}{3} \right\} \right)
 \end{aligned} \tag{III-43}$$

Por tanto, en la posición de equilibrio, el peso total (cajón más lastre de agua), aplicado en su centro de gravedad G , estará contrarrestado por el empuje, aplicado en el centro de carena C , **Figura III-22**.

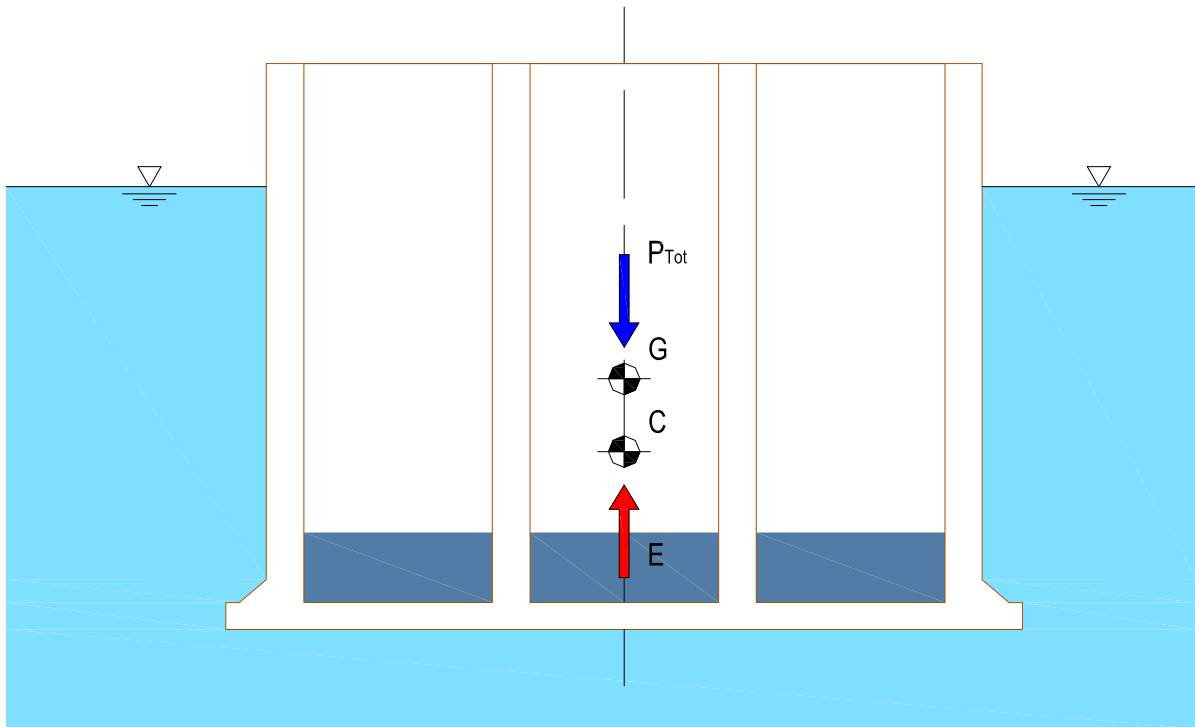


Figura III-22. Condiciones de equilibrio de un cajón en flotación

De esta condición de equilibrio se puede determinar la altura de calado del cajón:

$$H_{Calado}^{Fuste} = \frac{1}{A_{Maciza}^{Fuste}} \cdot \left(\frac{\rho_{Hormigon}}{\rho_{Agua}} \cdot V_{Cajon} + A_{Aligerada}^{Fuste} \cdot H_{Agua} - A_{Losa} \cdot H_{Losa} - 2 \cdot V_{Cartabon} \right) \tag{III-44}$$

Y por tanto, la altura libre emergida del cajón, denominada francobordo o resguardo, es:

$$H_{Francobordo} = H_{Fuste} + H_{Murete} - H_{Calado}^{Fuste} \tag{III-45}$$

El resguardo dependerá, por tanto, del peso propio del cajón, del lastre de las celdas y de posibles sobrecargas superiores de uso (maquinaria, operarios, etc.).

III.1.13.2.- Estabilidad final

Dada una determinada posición de equilibrio en flotación, se define el metacentro como el centro instantáneo de rotación entre dos flotaciones próximas.

La posición del metacentro, desde la base del cajón, se determina mediante la *Fórmula de Bouguer* [79]:

$$H_{Metacentro} = \frac{I_{Flotacion}}{V_{Carena}} + CG_{Z,Carena} \quad \text{(III-46)}$$

donde $I_{Flotacion}$ es el momento de inercia del área en flotación, es decir, de la sección horizontal del volumen en flotación respecto al eje de giro de la flotación, ecuación (III-12). De los dos ejes principales de giro (según eslora o manga) es más crítico el eje débil según la eslora: el cajón puede volcar en sentido transversal del ancho (*roulis* o balanceo, según el eje longitudinal).

La presencia del lastre de agua disminuye la inercia de flotación por efecto de la superficie libre. Si las celdas no están comunicadas entre sí, el momento de inercia de las celdas lastradas se considera respecto del centro de gravedad de cada una de dichas celdas, ecuación (III-29), por lo que la reducción no es significativa (en términos medios inferior al 5%), y en general se puede simplificar cancelando esta disminución con la aportación de los topes.

$$I_{Flotacion} = I_{X,Maciza}^{Fuste} - I_{X,CeldasEs\ tan\ cas}^{Lastre} \approx \frac{1}{12} \cdot Eslora \cdot Manga^3 \quad \text{(III-47)}$$

Por el contrario, en caso de realizarse la comunicación de las celdas (inercia según la ecuación (III-30) calculada en el centro de gravedad del cajón), ésta deberá hacerse cuando la altura de lastre sea suficiente (a partir de 2 m en términos medios), de modo que la elevada disminución de la inercia de flotación se vea compensada por el aumento en la altura del centro de carena, y no se ponga en riesgo la estabilidad del cajón. En todo caso, es conveniente dejar aisladas las celdas de esquina, para poder corregir posibles escoras.

$$I_{Flotacion} = I_{X,Maciza}^{Fuste} - I_{X,CeldasEs\ tan\ cas}^{Lastre} - I_{X,CeldasComunicadas}^{Lastre} \quad \text{(III-48)}$$

La altura metacéntrica GM se define como la distancia entre el centro de gravedad y el metacentro. Se denomina también brazo estabilizador. Debe ser positiva, ya que en caso contrario el cajón sería inestable y se daría la vuelta, buscando una configuración más estable. Cuanto mayor sea este valor, mayor será la estabilidad contra el volcado. Valores excesivos reducen de forma notable los períodos de balance o rólido, de manera indeseable.

$$Brazo_{Estabilizador} = \overline{GM} = H_{Metacentro} - CG_{Z, Cajon} \quad (III-49)$$

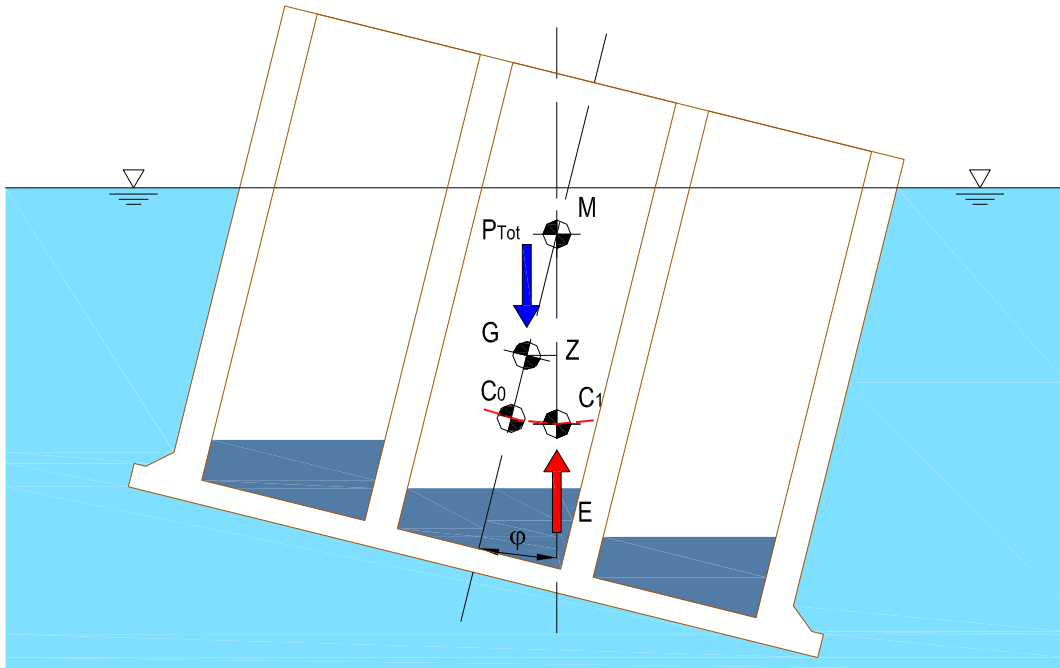


Figura III-23. Estado de flotación del cajón tras una rotación

Si por acción de un momento externo el cajón es llevado a la posición indicada en la **Figura III-23**, adoptando un ángulo de escora respecto de la vertical φ y luego dejado libre, el peso se mantiene invariable y aplicado en G . El empuje también se mantiene constante (el volumen desalojado sigue siendo el mismo) pero el centro de carena se desplaza a la posición C_1 . Se genera así una nueva recta de acción y un momento que tiende a hacer regresar al cajón a la posición inicial, denominado momento *adrizante*.

De la figura, el brazo adrizante se obtiene a partir de la altura metacéntrica y del ángulo girado en la rotación.

$$Brazo_{Adrizante} = \overline{GZ} = Brazo_{Estabilizador} \cdot \sin(\varphi) \quad (III-50)$$

Por tanto, la estabilidad de una flotación estará definida por el sentido del par que aparezca al apartar el cuerpo de esa posición de equilibrio: será estabilizador (adrizante) si se ejerce en sentido contrario al giro impuesto.

III.1.13.3.- Condiciones de estabilidad naval

En general se establecen dos condiciones para garantizar, durante la construcción y en el posterior transporte, la estabilidad naval del cajón en flotación.

- El cajón debe flotar con un suficiente resguardo. Se considera que un resguardo de 1 m da las suficientes garantías de seguridad. Si es posible, para aumentar el resguardo lo que debe hacerse es disminuir el lastre.

- El cajón debe ser lo suficientemente estable para no volcar *espontáneamente* (por equilibrio inestable). Ello se logra fijando un brazo estabilizador mínimo de 0.3 a 1 m. Al contrario que en el caso anterior, el brazo estabilizador aumenta al aumentar el lastre (siempre que el cajón siga permaneciendo a flote).

Por tanto, con las condiciones de brazo estabilizador y francobordo mínimos se puede determinar la altura de agua de lastre uniforme requerida para la flotación dentro de las condiciones de estabilidad especificadas.

$$\begin{aligned} \text{Brazo}_{\text{Estabilizador}} &\geq \text{Brazo}_{\text{Estabilizador, M\u00ednimo}} \\ H_{\text{Francobordo}} &\geq H_{\text{Francobordo, M\u00ednimo}} \end{aligned} \quad \text{(III-51)}$$

III.1.13.4.- Adici\u00f3n de lastre de adrizado

En aquellos casos en los que la secci\u00f3n transversal del caj\u00f3n no presente doble simetr\u00eda, cuando sea puesto a flote, sin lastre o con un lastre id\u00e9ntico en todas las celdas, \u00e9ste presentar\u00e1 una peque\u00f1a escora, que se puede determinar a partir de la **Figura III-24**, considerando que la posici\u00f3n de equilibrio se alcanza cuando el caj\u00f3n gira seg\u00fan un par desestabilizador.

$$\begin{aligned} \text{Escora}_{\text{Longitudinal}} &= \text{Eslora} \cdot \sin(\varphi_0) = \text{Eslora} \cdot \sin\left(\arctan\left(\frac{CG_{X,\text{Calado}}^{\text{Fuste}} - CG_{X,\text{Carena}}}{CG_{Z,\text{Calado}}^{\text{Fuste}} - CG_{Z,\text{Carena}}}\right)\right) \\ \text{Escora}_{\text{Transversal}} &= \text{Manga} \cdot \sin(\varphi_0) = \text{Manga} \cdot \sin\left(\arctan\left(\frac{CG_{Y,\text{Calado}}^{\text{Fuste}} - CG_{Y,\text{Carena}}}{CG_{Z,\text{Calado}}^{\text{Fuste}} - CG_{Z,\text{Carena}}}\right)\right) \end{aligned} \quad \text{(III-52)}$$

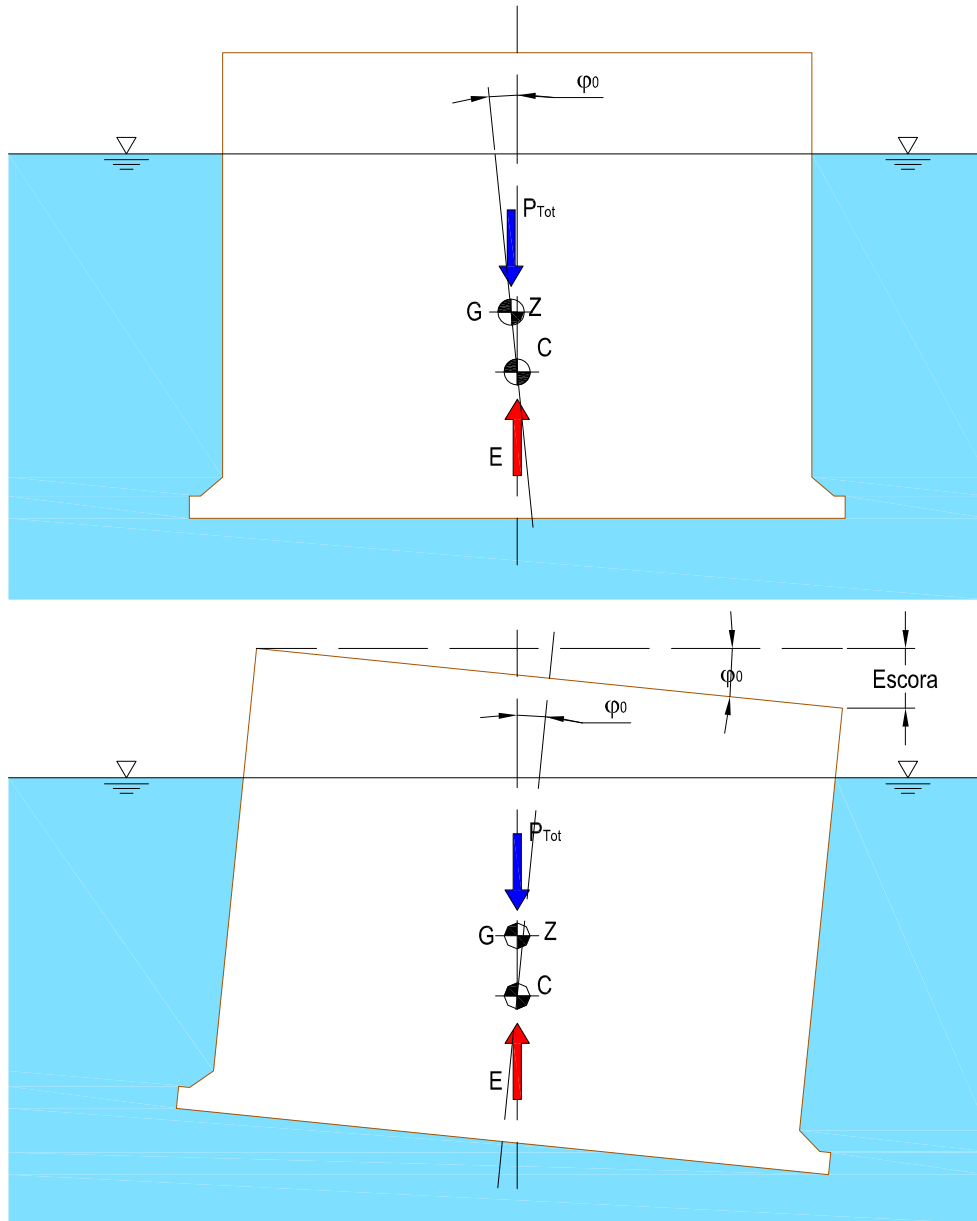


Figura III-24. Posición de equilibrio con escora transversal

Es práctica habitual lastrar adicionalmente alguna de las celdas circulares para evitar dicha escora, tratando de desplazar el centro de gravedad del cajón, para que pase por la misma línea de acción que el centro de carena. Se considerará que las celdas no están comunicadas entre sí.

Determinadas a priori las celdas que llevarán este lastre de adrizado, por ejemplo, todas las de la fila del Lado Tierra cuando la falta de simetría es en la dirección transversal, el problema se resuelve imponiendo que la escora sea nula y que el brazo estabilizador sea igual al mínimo exigido, obteniendo la altura de agua uniforme en todas las celdas y la adicional a añadir en las celdas seleccionadas.

$$\left\{ \begin{array}{l} CG_{Y,Calado}^{Fuste} - CG_{Y,Carena} = 0 \\ Brazo_{Estabilizador} = Brazo_{Estabilizador,Minimo} \end{array} \right. \Rightarrow (H_{Agua}, H_{AguaAdicional}) \quad (III-53)$$

$$H_{Francobordo} \geq H_{Francobordo,Minimo}$$

Finalmente se comprobará que se cumple la condición de resguardo mínimo.

III.1.13.5.- Fase de construcción

Durante la fase de construcción del cajón en la pontona, a partir de encofrados deslizantes, toda vez que se ha fabricado la losa y el primer metro de fuste, la aplicación de las ecuaciones anteriores permite determinar las condiciones de estabilidad naval del cajón así como el empuje necesario que debe aportar la pontona en las fases iniciales en las que el cajón está muy poco aligerado y no sería posible su flotación.

El proceso se puede dividir en tres etapas, conforme aumenta la altura de fuste construida (véase la **Figura III-25**):

1. El conjunto pesa demasiado y se hundiría: requiere empuje adicional de la pontona. Se fija el resguardo en 1 m. No se añade lastre uniforme de agua, ya que elevaría el peso del conjunto aún más, aunque sí se tiene en cuenta el lastre adicional en determinadas celdas para evitar la escora. La altura construida de fuste a partir de la cual ya no se requiere empuje de la pontona marca el final de esta etapa.

$$\left\{ \begin{array}{l} CG_{Y,Calado}^{FusteConstruido} - CG_{Y,Carena} = 0 \\ Empuje_{Pontona} = P_{Total} - Empuje_{Mar} \geq 0 \Rightarrow (H_{AguaAdicional}, Empuje_{Pontona}) \\ H_{Francobordo} = H_{Francobordo,Minimo} \end{array} \right. \quad \text{(III-54)}$$

2. El conjunto flota por sí mismo, de manera estable: no requiere empuje adicional ni lastre uniforme de agua. No es necesario añadir lastre uniforme de agua, aunque sí se tiene en cuenta el lastre adicional en determinadas celdas para evitar la escora. La altura construida de fuste a partir de la cual se requiere lastre uniforme de agua para mantener el brazo estabilizador mínimo marca el final de esta segunda etapa.

$$\left\{ \begin{array}{l} CG_{Y,Calado}^{FusteConstruido} - CG_{Y,Carena} = 0 \\ Brazo_{Estabilizador} \geq Brazo_{Estabilizador,Minimo} \Rightarrow (H_{AguaAdicional}) \\ H_{Francobordo} \geq H_{Francobordo,Minimo} \end{array} \right. \quad \text{(III-55)}$$

3. El conjunto flota por sí mismo, pero es inestable y puede volcar, por lo que requiere peso adicional del lastre uniforme de agua. Se fija el brazo estabilizador en 0.5 m. Igualmente, se tiene en cuenta el lastre adicional en determinadas celdas para evitar la escora.

$$\left\{ \begin{array}{l} CG_{Y,Calado}^{FusteConstruido} - CG_{Y,Carena} = 0 \\ Brazo_{Estabilizador} = Brazo_{Estabilizador,Minimo} \Rightarrow (H_{Agua}, H_{AguaAdicional}) \\ H_{Francobordo} \geq H_{Francobordo,Minimo} \end{array} \right. \quad \text{(III-56)}$$

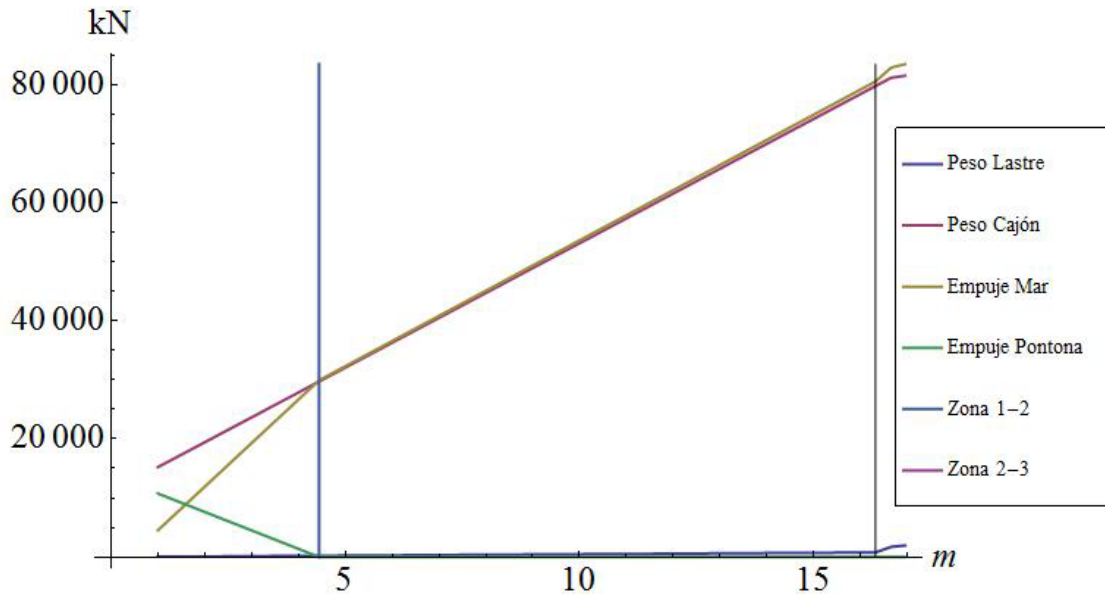


Figura III-25. Pesos y empujes en la construcción de un cajón. Zonas de estabilidad

III.2.- Geometría multi-bloque por tipologías

Conocidas las variables de diseño de la sección (eslora y manga, diámetro de las celdas circulares, número de filas y columnas, espesores de las paredes y grado de achaflanamiento), tal y como se ha establecido en el apartado anterior, es inmediato conocer el número de aligeramientos por tipología y la posición de cada uno de ellos, y generar un modelo geométrico adecuado que reproduzca fielmente la sección transversal del cajón.

Dado que el problema real del cajón sometido a ciertas condiciones de contorno no se puede resolver analíticamente, es preciso emplear algún método que discretice el dominio físico en un número finito de subdominios, en los cuales puedan obtenerse soluciones numéricas de las ecuaciones diferenciales que gobiernan el problema.

Este proceso de generación de un malla de elementos finitos puede hacerse fácilmente en un pre-procesador determinado, si bien el tipo de elementos a usar y las características de la malla dependerán de los algoritmos de mallado que incorpore y de cómo se adapten los elementos empleados a la geometría concreta de la pieza.

Por tanto, habrá que elegir el tipo de malla, que puede ser estructurada, no estructurada o híbrida, y el tipo de elemento, que en dos dimensiones pueden ser triángulos o cuadriláteros y en tres tetraedros o hexaedros, fundamentalmente.

III.2.1.- Elección del tipo de elemento finito

En la **Figura III-26** se muestra la mitad de la sección del fuste de un cajón simétrico transversalmente de 5 filas y 6 columnas de aligeramientos. Se han calculado

las celdas, sus formas geométricas y sus coordenadas y han sido introducidas en el pre-procesador GiD[®][81] en forma de puntos, líneas y arcos.

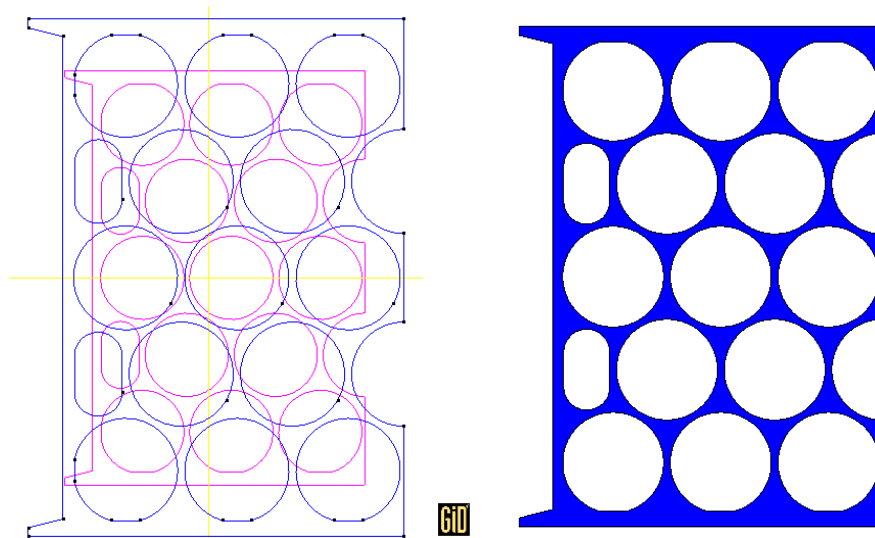


Figura III-26. Sección del fuste en GiD[®]

La superficie final se ha generado a partir de operaciones booleanas de adición y sustracción de superficies NURBS⁶ creadas por el propio pre-procesador. Esta superficie ya puede ser mallada, prescindiendo en este momento de otras consideraciones acerca de las condiciones de contorno en carga y desplazamientos.

A continuación se discutirá la elección del mallado adecuado a la geometría específica de la sección transversal del cajón. Como indica Ribó [83], esta elección es uno de los cuellos de botella en la aplicación práctica del método de los elementos finitos.

III.2.1.1.- Mallado no estructurado

Inicialmente, el tipo de mallado conveniente para una geometría compleja es el *mallado no estructurado*, ya que se adapta sin problemas a cualquier tipo de dominio sin necesidad de manipulaciones previas de la geometría original. En general, se emplean elementos triangulares de superficie o tetraédricos de volumen en el caso del mallado tridimensional del fuste.

Por el contrario, los algoritmos de generación de este tipo de mallas que incorporan los pre-procesadores son los más complejos. Pueden ser algunos de los siguientes, o combinaciones de los mismos:

- *Triangulación de Delaunay* [85],[86]. Consiste en la generación de puntos sobre el dominio que son conectados para formar una triangulación que maximice, por ejemplo, el mínimo de los ángulos de cada par de triángulos adyacentes. El círculo que pasa por los tres vértices de cada triángulo no puede contener otros puntos de la triangulación.

⁶ NURBS: Acrónimo inglés de Non-Uniform Rational B-Spline, modelo matemático empleado en computación gráfica para la generación y representación de curvas y superficies [82].

- *Método frontal o del frente de avance* [87]. Consiste en la modificación del dominio original por la generación sucesiva de triángulos a partir de su borde.
- *Subdivisiones sucesivas*. Se basa en la utilización de una cuadrícula que se superpone al dominio. Las casillas exteriores se descartan, mientras que el resto se puede subdividir sucesivamente de acuerdo al grado de discretización deseado en cada región del dominio. Finalmente se dividen las celdas en triángulos. Esta técnica es conocida como método de árboles cuaternarios o *quadtrees* (árboles octales u *octree* en tres dimensiones), [88].

En la **Figura III-27** se muestra la sección transversal mallada de forma no estructurada mediante triángulos, observándose que la aproximación de la malla a la geometría real no es adecuada cuando se emplea un número reducido de elementos.

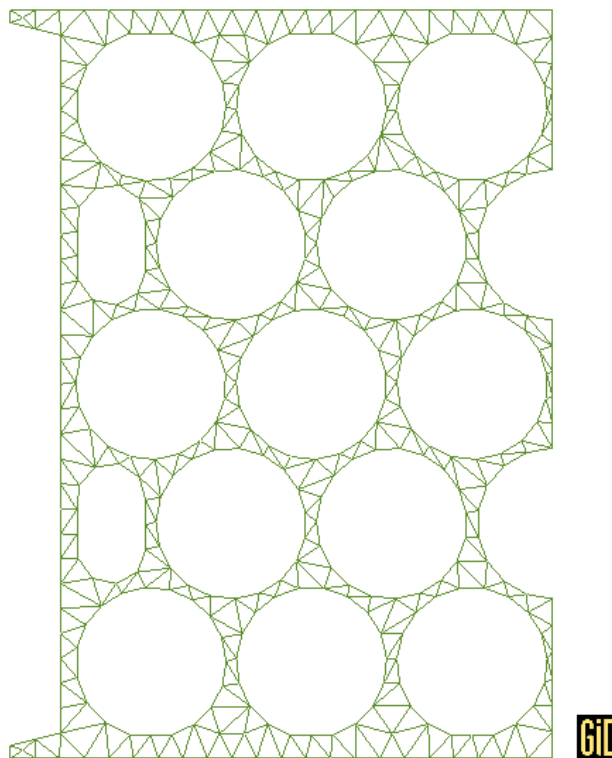


Figura III-27. Malla no estructurada. Tamaño elemento 1, línea 0.5
(430 elementos y 394 nodos)

Como es de esperar, la geometría se aproximará cada vez mejor al ir refinando la malla (**Figura III-28**), aunque el coste computacional irá creciendo.

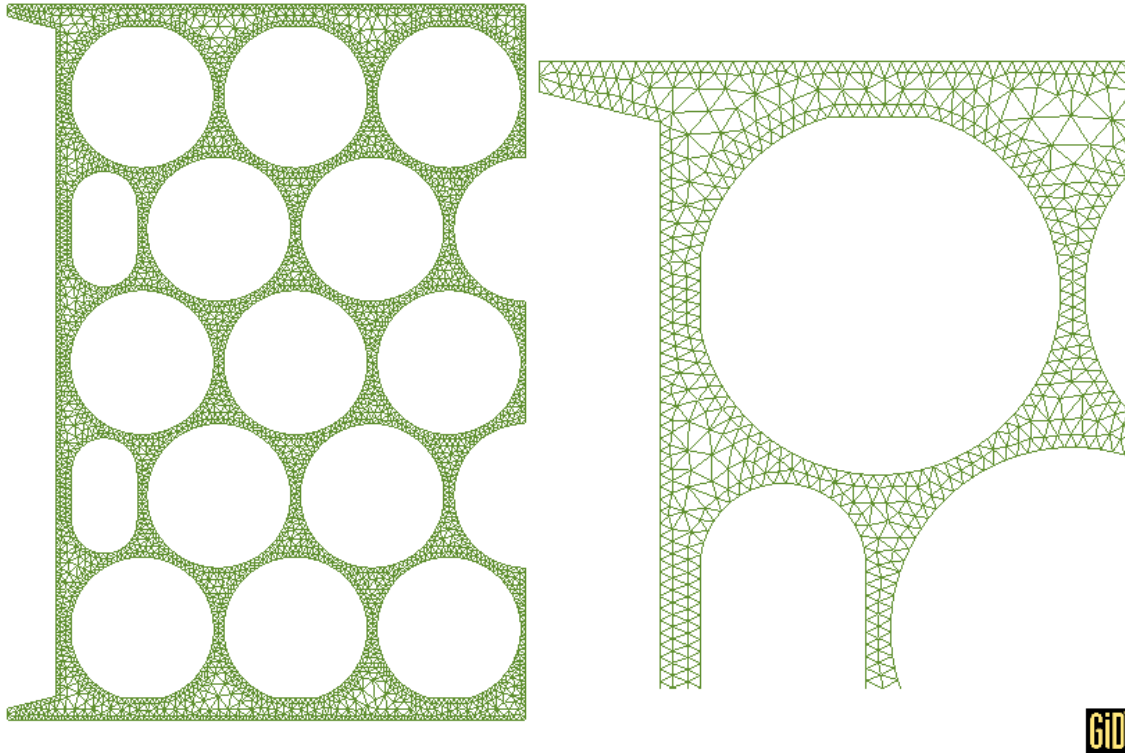


Figura III-28. Malla no estructurada. Tamaño elemento 1, línea 0.1
(6058 elementos y 3970 nodos)

A la vista de esta última malla obtenida, el mallado no estructurado se podría considerar adecuado para reproducir fielmente la geometría de la sección y captar con cierta precisión el estado tensional en las gargantas. Sin embargo, teniendo en cuenta que la sección será extruída en altura para dar lugar al fuste, y que los elementos a usar serán tetraedros en vez de triángulos, la fuerte desproporción existente entre las dimensiones de altura frente a las de espesor de paredes y gargantas (del orden de 100 a 1) obligará a colocar una gran cantidad de elementos, para que no resulten con una relación de aspecto deficiente.

Así, extruyendo tan sólo una altura de 2 metros, se pasa de los 430 elementos triangulares y 394 nodos para una malla de tamaño 0.5 en las líneas (**Figura III-27**) a 4002 tetraedros y 1619 nodos (**Figura III-29**) y de los 6058 elementos y 3970 nodos para un tamaño de 0.1 (**Figura III-28**) a 66025 tetraedros y 21502 nodos (**Figura III-30**).

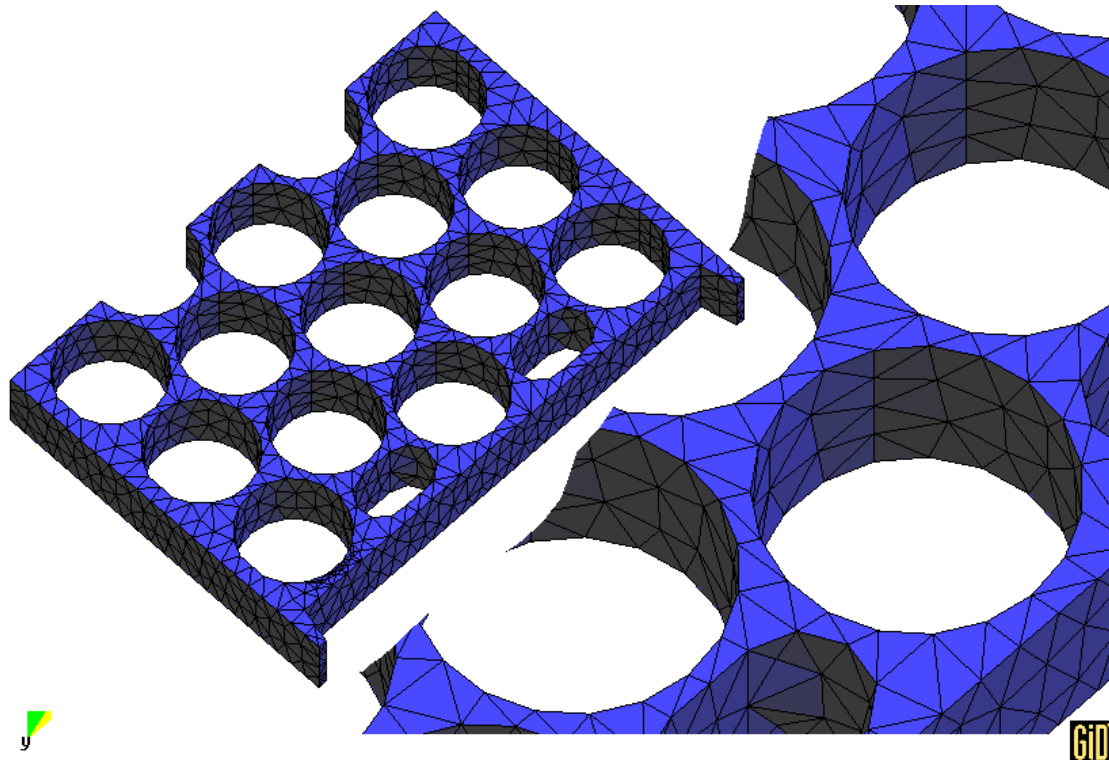


Figura III-29. Malla no estructurada. Tamaño elemento 1, línea 0.5.
(4002 tetraedros y 1619 nodos)

Es claro que, para un fuste de entre 15 y 20 m de altura, el número final de elementos y nodos será muy elevado, si se emplea una malla lo más fina posible que capte con cierta precisión los contornos circulares de las celdas y las zonas críticas entre dichas celdas (*gargantas*) en las cuales se van a estudiar las tensiones.

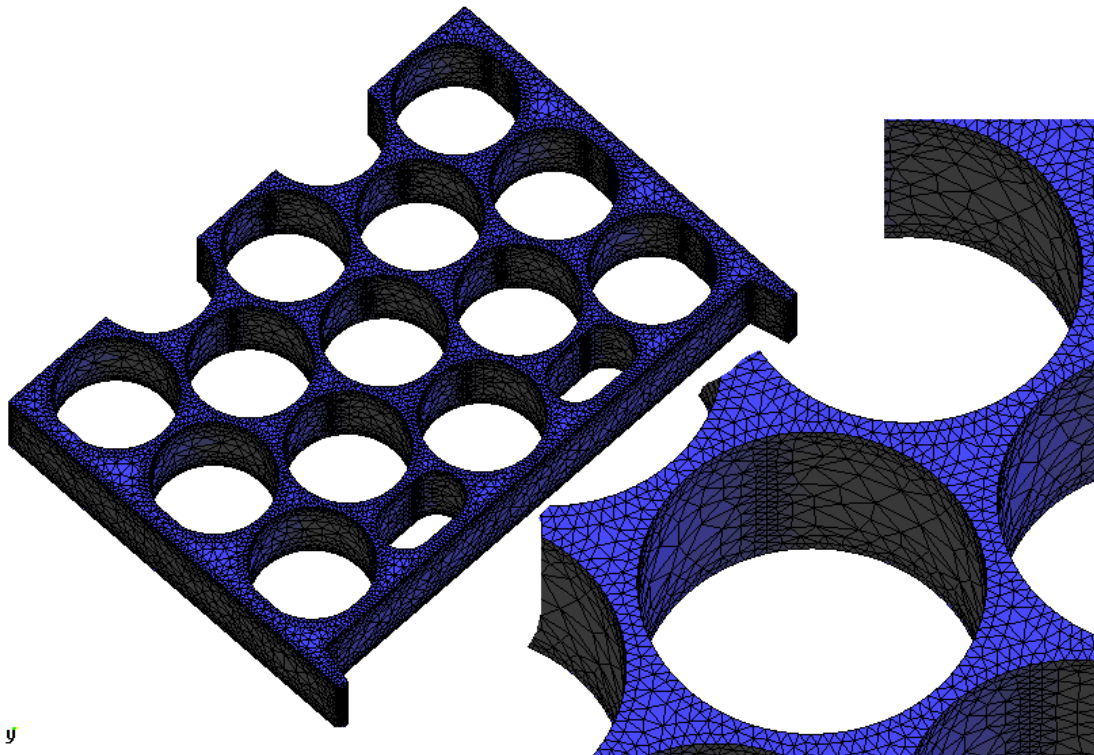


Figura III-30. Malla no estructurada. Tamaño elemento 1, línea 0.1.
(66025 tetraedros y 21502 nodos)

III.2.1.2.- Mallado estructurado

A pesar de la compleja geometría del cajón, la utilización de cuadriláteros para mallado de superficies o hexaedros para volúmenes presenta más ventajas que el uso de triángulos/tetraedros. Con éstos, será necesario un número menor de nodos y elementos, se colocarán en posiciones de interés para el análisis, se comportará mejor en el análisis estructural [89] y, para el mallado del fuste, se podrán emplear elementos hexaédricos con grandes alturas y bases estrechas, sin que ello afecte en exceso a su factor de forma.

El uso de cuadriláteros/hexaedros habitualmente está relacionado con el *mallado estructurado*. Una malla es estructurada si todos sus nodos interiores tienen el mismo número de elementos que los contienen. Es decir, que el número de elementos que contienen a cada nodo como una de sus conectividades es constante para todos los nodos del interior del dominio.

La característica de una malla de este tipo es la regularidad. Los algoritmos de generación de malla estructurada posicionan los nodos de manera ordenada en el interior de las superficies o volúmenes. En dominios rectangulares es simple construir una malla. El problema aparece cuando se requiere construir mallas estructuradas en dominios irregulares: *De alguna manera, la regularidad y la repetición de la malla resultante debe estar ya reflejada en la composición de la geometría original* [84].

Para dominios simplemente conexos, es decir, sin agujeros, e irregulares, existen diferentes métodos para la construcción de mallas. En estos casos el dominio computacional es un rectángulo en dos dimensiones o un paralelepípedo en tres dimensiones. Algunos de los métodos que actualmente se usan son [90]:

- Generación algebraica de mallas, **Figura III-31**, mediante *mapping*, donde el dominio original se transforma en un dominio canónico sobre el que se genera una retícula y se realiza la transformación inversa para ubicar los nodos en el dominio real, o mediante *interpolación*, donde los puntos interiores de la malla se encuentran interpolando las fronteras del dominio físico y la malla de nodos internos se genera a partir de los nodos del contorno [91], [92], [93], [94].

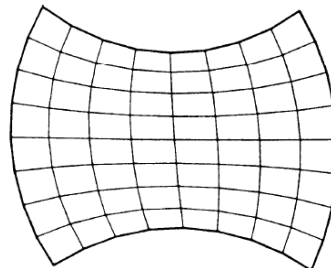


Figura III-31. Malla generada por un método algebraico

- Métodos donde los puntos interiores se encuentran resolviendo una ecuación diferencial parcial (EDP), empleando como condición de contorno la propia geometría del contorno del dominio, y ecuaciones de Laplace o Poisson [95], [96].

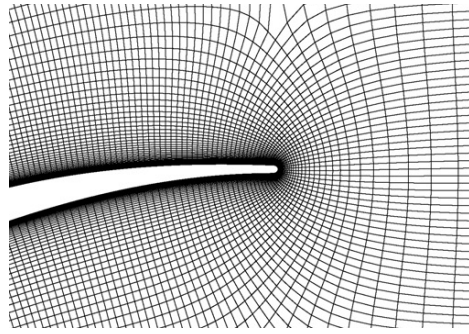


Figura III-32. Malla generada mediante resolución de EDP de Poisson [96]

- Generación variacional de mallas, donde algún funcional es maximizado o minimizado [97].
- Superposición y deformación, donde la malla es generada por superposición de una retícula cuadrada sobre el dominio del problema, y posterior eliminación de los elementos que no intersectan a dicho dominio y adaptación de los que cortan parcialmente el contorno por deformación de cuadrados o por uso de una retícula jerárquica *quadtree/octree* [98], [99].
- Crecimiento estructurado, extrusión o *sweeping* [100], [101], a partir de una malla de cuadriláteros sobre la superficie de partida se genera una malla de hexaedros al emplear como referencia una curva que une esta con la superficie destino, siendo ambas de igual topología, **Figura III-33**.

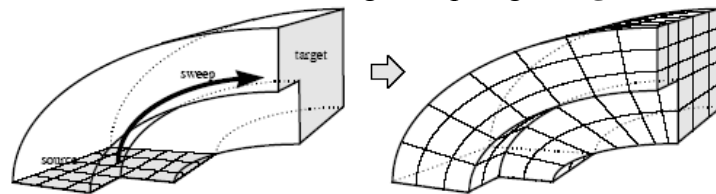


Figura III-33. Técnica de barrido [100]

Para dominios más complejos y con posible presencia de agujeros existen otras técnicas de mallado con cuadriláteros/hexaedros que dan lugar a mallas que son, en general, no estructuradas. Suelen clasificarse en métodos directos o indirectos [102].

En los métodos indirectos se realiza un mallado previo mediante triángulos/tetraedros que luego es convertido, empleando diferentes técnicas, en cuadriláteros/hexaedros.

- Un método indirecto consiste en dividir cada triángulo en tres cuadriláteros, [103]. Es rápido pero introduce un alto número de nodos irregulares y proporciona elementos de baja calidad, **Figura III-34**. Igualmente, los tetraedros pueden dividirse en cuatro hexaedros, **Figura III-35**.

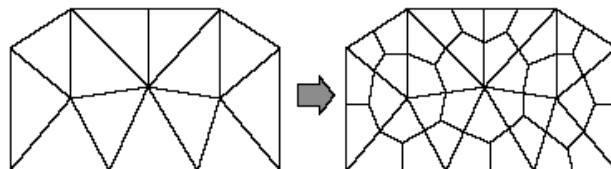


Figura III-34. Método de división de triángulos en cuadriláteros

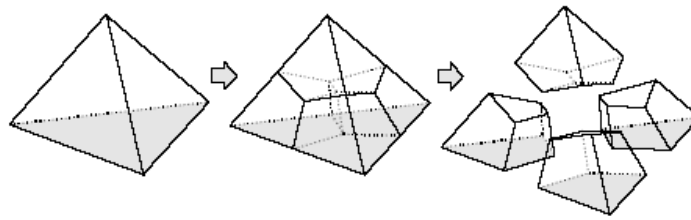


Figura III-35. Método de división de tetraedros en hexaedros

- El método opuesto consiste en combinar triángulos adyacentes en cuadriláteros, **Figura III-36**. La unión de tetraedros para dar lugar a hexaedros es compleja, pues requiere la búsqueda de combinaciones de 5 o más tetraedros.

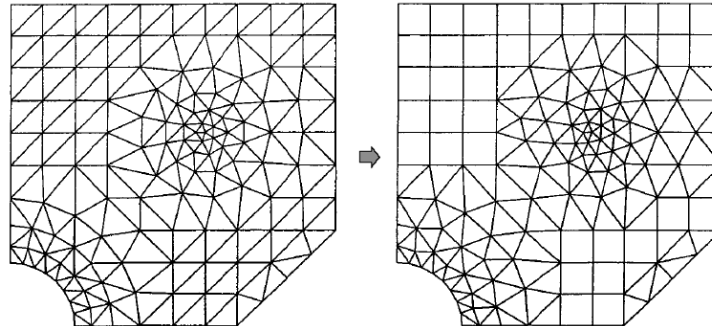


Figura III-36. Método de combinación de triángulos en cuadriláteros [104]

En los métodos directos, por otro lado, se colocan los cuadriláteros/hexaedros directamente sobre la superficie/volumen, siguiendo diferentes técnicas.

- Mallado por descomposición del dominio en regiones más sencillas, con varias técnicas:
 - Método de árboles cuaternarios o *quadtree*, los cuadriláteros se insertan en las ramas, ajustando los nodos para mantener la conformidad en las fronteras [105].
 - Subdivisión del dominio en formas poligonales más simples, sobre las cuales se pueden ajustar diversas plantillas de cuadriláteros [106].
 - Empleando la transformación del eje central o *medial axis*, **Figura III-37**, se divide el dominio en subregiones topológicamente simples mediante una serie de líneas generadas desde los centros de los círculos máximos inscritos en el dominio y, posteriormente, se descomponen estas regiones en cuadriláteros [107].

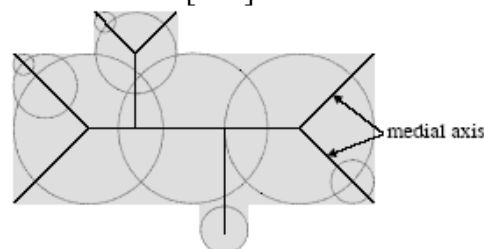
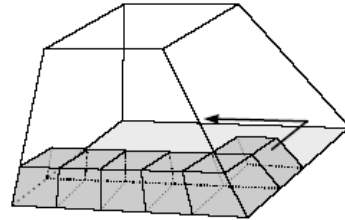
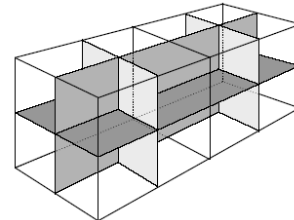


Figura III-37. Técnica de *medial axis*

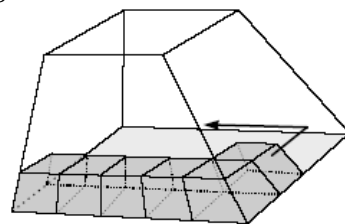
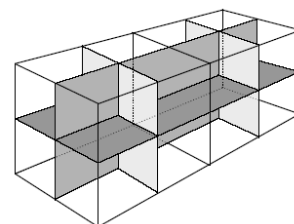
- Para el caso tridimensional se emplea la superficie central o *medial surface*, colocando esferas en el volumen, que luego son malladas con hexaedros [108].
- Técnicas de avance frontal o *paving* [109], mediante la cual se construye progresivamente la malla insertando filas de cuadriláteros desde el

contorno del objeto hacia el interior, poniendo especial cuidado en formar una malla válida en las zonas de solape de estas hileras. Se requiere que el número de caras del contorno sea par [110].

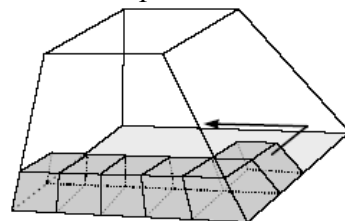
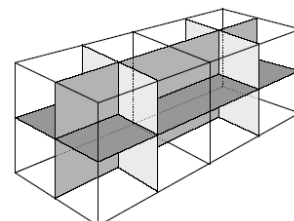
- La técnica de *plastering* [111], en tres dimensiones. En un volumen cerrado, que previamente ha sido definido a partir de caras con forma de cuadrilátero se forma una malla de hexaedros por adición de elementos desde dichas caras hacia el interior,

(a) *Plastering*(b) *Whisker Weaving*

- **Figura III-38.a**

(a) *Plastering*(b) *Whisker Weaving*Figura III-38. Técnicas de *plastering* y *Whisker Weaving*

- La técnica de *Whisker Weaving* [112] es otro método basado en un algoritmo de avance frontal para la generación de mallas de hexaedros mediante el uso del dual de dicha malla: un conjunto de superficies que se intersectan y biseccionan a los hexaedros en cada dirección. Este dual se denomina *spacial twist continuum (STC)*,

(a) *Plastering*(b) *Whisker Weaving*

- **Figura III-38.b.**
- Dados los problemas de conectividad de la malla que aparecen al emplear algoritmos de avance frontal, se ha desarrollado una aproximación recursiva que inserta capas de hexaedros separando el dominio completo en dos subdominios [113].
- Técnicas basadas en rejillas o *grid* [114], consistentes en la generación de una cuadrícula o rejilla tridimensional que se ajuste al dominio a mallar. La zona del contorno se malla mediante algoritmos de refinamiento.

Técnica Multibloque

Las técnicas anteriormente descritas para la discretización de dominios dados pueden generar resultados no satisfactorios para geometrías complejas. La técnica *multi-*

bloque surge como solución en estos casos, **Figura III-39**. Requiere que se realice previamente una subdivisión de la geometría en múltiples bloques o super-elementos con una topología más sencilla, para que un pre-procesador automático pueda a continuación generar la malla de cuadriláteros/hexaedros en función de la densidad de malla deseada, con alguna de las técnicas descritas, tanto con mallas estructuradas como con mallas no estructuradas, o incluso con ambas (malla híbrida).

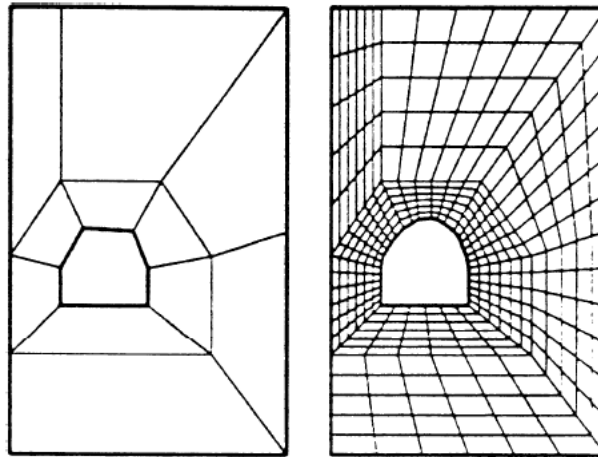


Figura III-39. Generación de malla multibloque tras descomposición inicial del dominio [115]

Se requiere, en todo caso, una estructura general para ordenar los bloques, conservar la información de la conectividad interna, y asegurar la correcta unión de las mallas, para lo cual es preceptivo que las interfaces comunes presenten una única discretización.

Para automatizar la tarea de generar los bloques en [116] se propone extraer una serie de reglas heurísticas basadas en la experiencia, **Figura III-40**, mientras que en [117] se propone la técnica de *divide y vencerás* para dividir el dominio global recursivamente en sub-dominios, hasta que todos los sub-dominios permitan una generación automática de malla estructurada, empleando igualmente reglas heurísticas, **Figura III-41**.

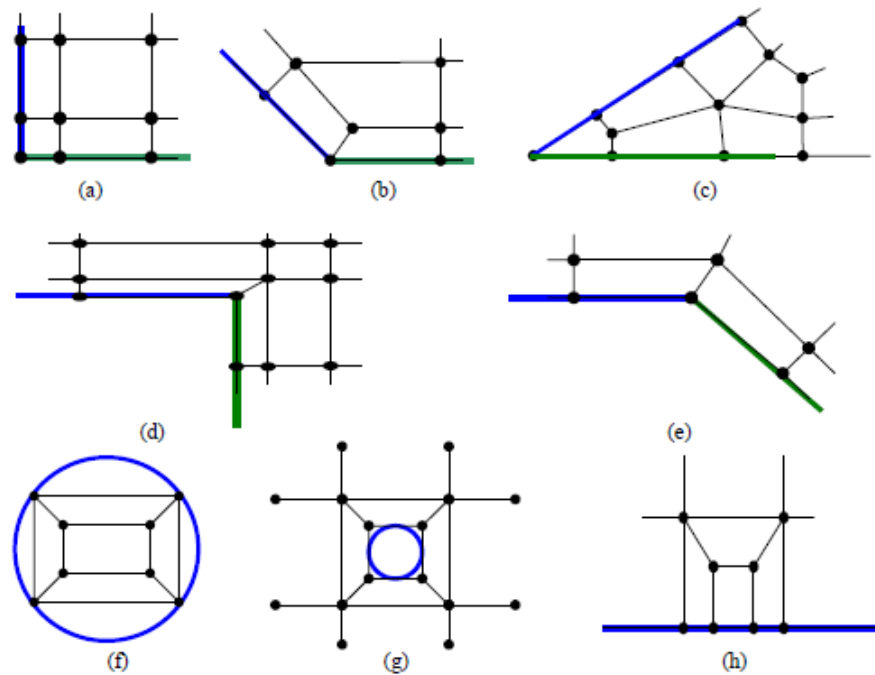


Figura III-40. Reglas usadas para manejar el contorno de una parte

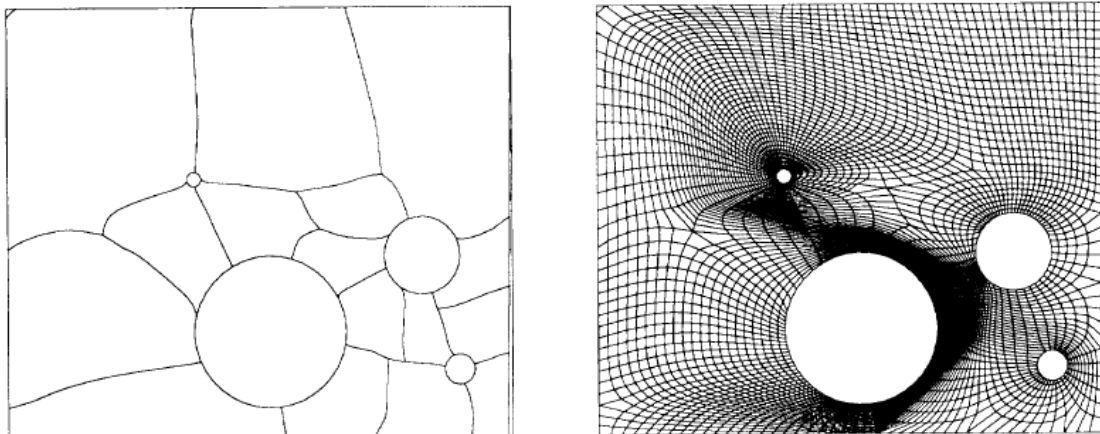


Figura III-41. Topología de bloques global y malla final generada en dominio con 4 círculos

III.2.1.3.- Mallado estructurado de un cajón multicelular

El método que se propone en la presente Tesis para generar, de forma automática y sin intervención del usuario, una malla estructurada en la sección transversal de un cajón multicelular consiste en adaptar la propuesta de la técnica multi-bloque a la geometría específica de dicha sección, caracterizada por la disposición de celdas al tresbolillo.

Este método se realiza en dos etapas: en primer lugar se generan bloques no estructurados, aprovechando la peculiar disposición de las celdas aligeradas en la sección. En segundo lugar, sobre cada uno de estos bloques se realizan nuevas divisiones, esta vez en bloques estructurados, que serán idénticas para todas las celdas que compartan la misma tipología.

A continuación se describen en detalle estos dos pasos.

III.2.1.3.1.- Generación de bloques no estructurados en torno a aligeramientos

La primera etapa consiste en generar automáticamente un primer nivel de bloques no estructurados a partir del modelo subyacente de hexaedros que circunscriben a los aligeramientos circulares u ovoidales de la **Figura III-2**. Según sea el aligeramiento o su ubicación se pueden identificar diversas tipologías que se repiten a lo largo de la sección, **Figura III-11** y **Figura III-12**.

Las reglas que definen este algoritmo son sencillas:

1. Cada aligeramiento define un único bloque, asociado al hexágono en que está inscrito.
2. Si el hexágono está en la proximidad de una pared (lado mar, tierra o lateral), se adaptará de manera que forme un polígono cerrado que incluya una porción de dicha pared, de tal modo que no quede ninguna zona de éstas sin asignar a hexágonos.
3. Cada tope de cada esquina (incluido el doble tope si existe), aunque no tenga aligeramiento, es también un bloque.
4. Cada uno de los bloques llevará asignada una de las tipologías existentes, así como un número de orden.

III.2.1.3.2.- Generación de bloques estructurados por tipologías

A continuación, en una segunda etapa, se aplicará para cada una de estas tipologías un nuevo algoritmo de generación de bloques, esta vez estructurados, sobre los que finalmente el pre-procesador genere elementos cuadriláteros en la sección y hexaédricos en el volumen.

En esta etapa cobran relevancia las peculiaridades de la sección: espesores de las paredes, de las gargantas, grado de achaflanamiento de las celdas exteriores, etc.

Otra característica fundamental a la hora de generar bloques cuadriláteros en la sección es la de colocar siempre una arista paralela a la garganta en la que posteriormente se realizará el cálculo de esfuerzos para el armado. Como se ha mencionado previamente, esta es una de las razones que justifican el esfuerzo de generación de una malla estructurada.

Igualmente, se tendrá en cuenta que en la transición losa – fuste es preciso mantener la misma distribución de bloques.

El algoritmo de generación de cuadriláteros debe calcular para cada cuadrilátero la posición de sus vértices, las líneas rectas o curvas que los unen y la superficie formada.

De esta manera la división en bloques estructurados a realizar por tipologías es la siguiente.

Celdas tipo *Circular*

La división se realiza a partir de los 6 vértices del hexágono que circunscribe la celda y los 6 puntos medios de sus aristas y de los 12 puntos sobre la celda circular que se obtienen al unir radialmente los puntos exteriores con el centro del hexágono. Se obtienen 12 cuadriláteros idénticos, siendo uno de sus lados curvos⁷, el correspondiente al arco de circunferencia de ángulo central 30° , **Figura III-42**.

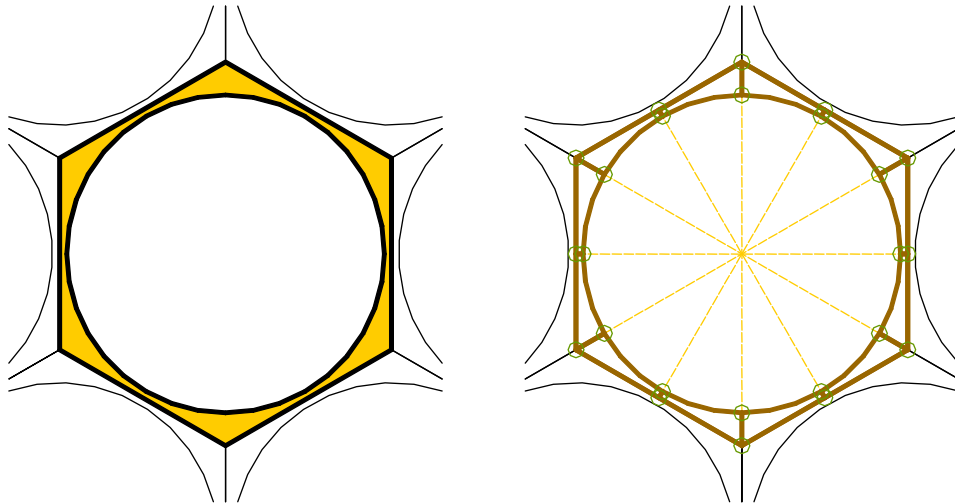


Figura III-42. Celda tipo *Circular*. 12 cuadriláteros

Celdas tipo *Circular-Mitad*

Tras aplicar la simetría vertical en la sección, resulta esta celda, en la que se pueden obtener 6 cuadriláteros, **Figura III-43**.

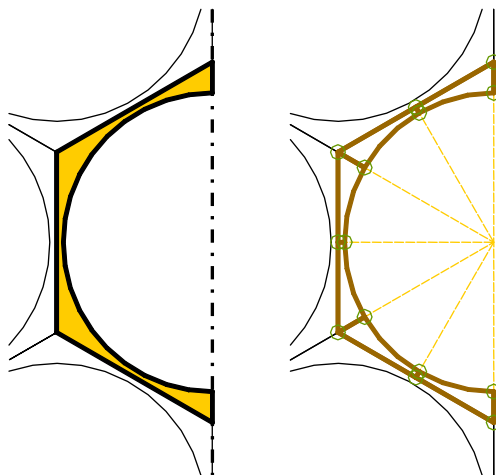


Figura III-43. Celda tipo *Circular-Mitad*. 6 cuadriláteros

⁷ Si bien, formalmente, el término cuadrilátero hace referencia a un polígono de 4 lados rectos, en la formulación de elementos finitos es habitual usar esta denominación aún cuando tenga algún lado curvo [118] en términos de cuadrilátero isoparamétrico curvo. En lo que sigue se empleará indistintamente el término “cuadrilátero” tanto si tiene todos sus lados rectos como si tiene alguno curvo.

Celdas tipo Ovoide

La celda no se corresponde exactamente con medio hexágono, ya que las zonas próximas a las celdas circulares superior e inferior se agruparán con éstas. En la **Figura III-44** se muestra cómo es la celda (izquierda) y qué división en cuadriláteros se realiza (derecha). El algoritmo de creación de estas divisiones debe tener presente los rangos de variación de las variables de entrada, para que no se produzcan zonas defectuosas. También tiene en cuenta que puede haber un tope con el que compartir nodos, por lo que cualquiera de los puntos de la pared lateral (salvo el primero y el último) podrá desplazarse sobre la misma convenientemente.

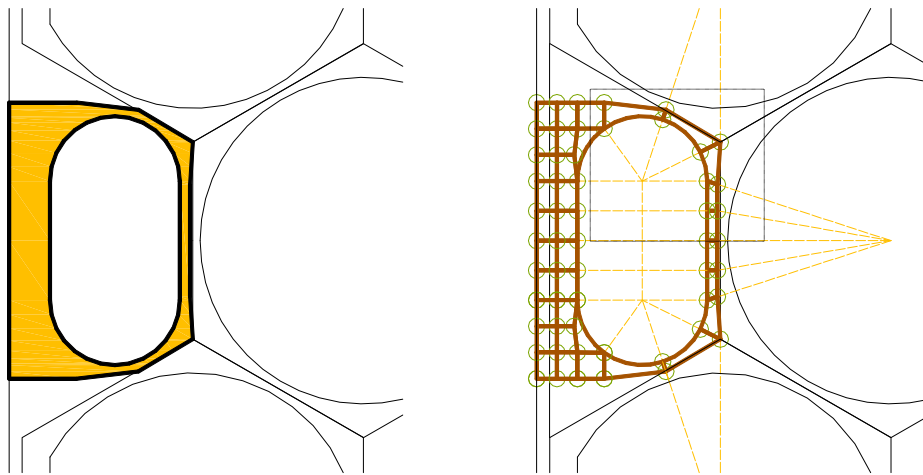


Figura III-44. Celda tipo Ovoide. 34 cuadriláteros

Así, siempre que sea posible se toman los puntos medios de las rectas o arcos que unen distintos puntos relevantes. Como se muestra en la **Figura III-45**, las zonas entre el ovoide y la celda circular adyacente y entre el ovoide y las celdas circular chaflán lateral no se sitúan sobre el lado del hexágono frontera entre ambas, sino que se colocan a la mitad de la distancia entre ambas, para evitar que valores dispares de espesores ($e_{Ov} \gg e_{Cir}$ o $e_{Ov} \ll e_{Cir}$) den lugar a cuadriláteros mal formados.

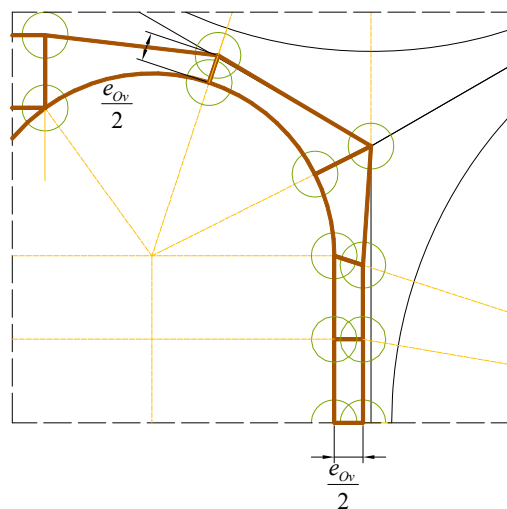


Figura III-45. Detalle de generación de cuadriláteros de la celda *Ovoide*

Celdas tipo *Circular-Ovoide*

La división se realiza del mismo modo que en la *Circular*, añadiéndose 8 vértices más en la zona de contacto con el ovoide, **Figura III-46**. Se unen con rectas desde el centro de la celda circular a los diversos puntos del perímetro de la celda ovooidal más próximos, que no se ubican sobre la recta del hexágono sino sobre la que pasa a la mitad de distancia entre celdas (**Figura III-45**).

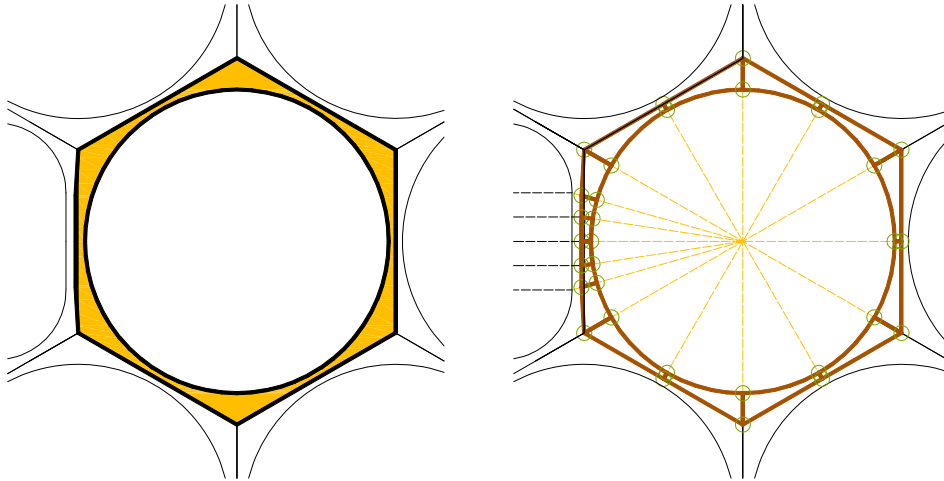


Figura III-46. Celda tipo *Circular-Ovoide*. 16 cuadriláteros

Celdas *Ovoide-Esquina*

Al igual que en la celda ovoide, las zonas entre el ovoide y la celda circular con chaflán frontal adyacente se sitúan a la mitad de la distancia entre ambas, **Figura III-45**. Nótese que la posición de los puntos situados en la pared lateral es variable, para permitir la unión con los topes dispuestos, cuyo tamaño es desconocido a priori. La división del rectángulo se hace sobre su centro de gravedad, y no sobre el centro de referencia **Figura III-47**.

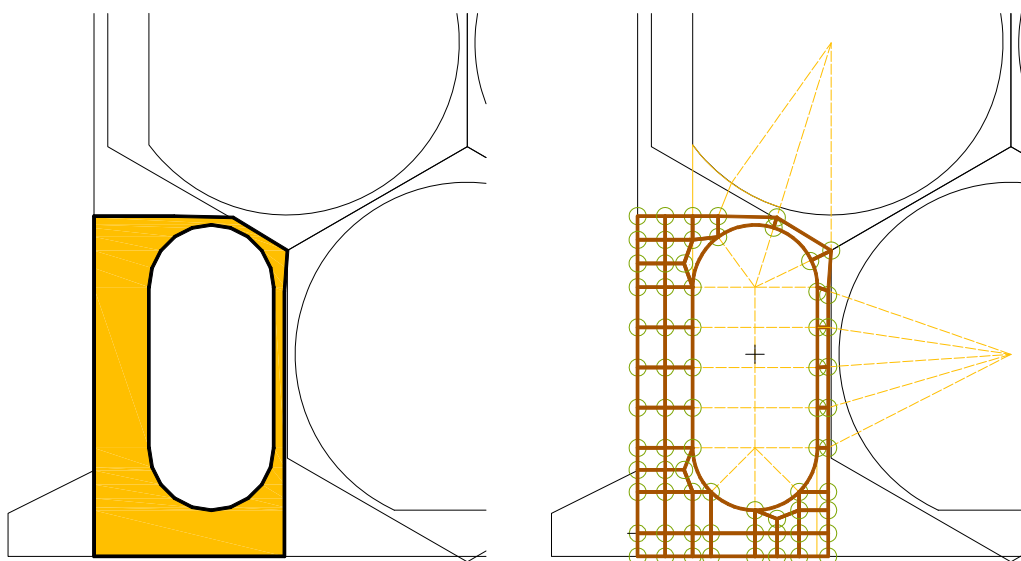


Figura III-47. Celda *Ovoide-Esquina*. 45 cuadriláteros

Celdas Circular con Chaflán Frontal

En función del grado de achaflanamiento hay que distinguir tres posibles casos. En todos ellos se ha mantenido la simetría vertical y el mismo tipo de reparto que en las celdas circulares, apoyándose en el hexágono que circunscribe a la celda achaflanada. Los parámetros referidos en las celdas frontales serán los relativos al Lado Tierra o al Lado Mar, según corresponda.

- $e_{ChaFro} = 0$, es decir, no hay chaflán, **Figura III-48.b**, 30 cuadriláteros.
- $0 < e_{ChaFro} < r - \frac{L}{2}$, **Figura III-48.c**, 34 cuadriláteros.
- $r - \frac{L}{2} \leq e_{ChaFro}$, **Figura III-48.d**, 30 cuadriláteros.

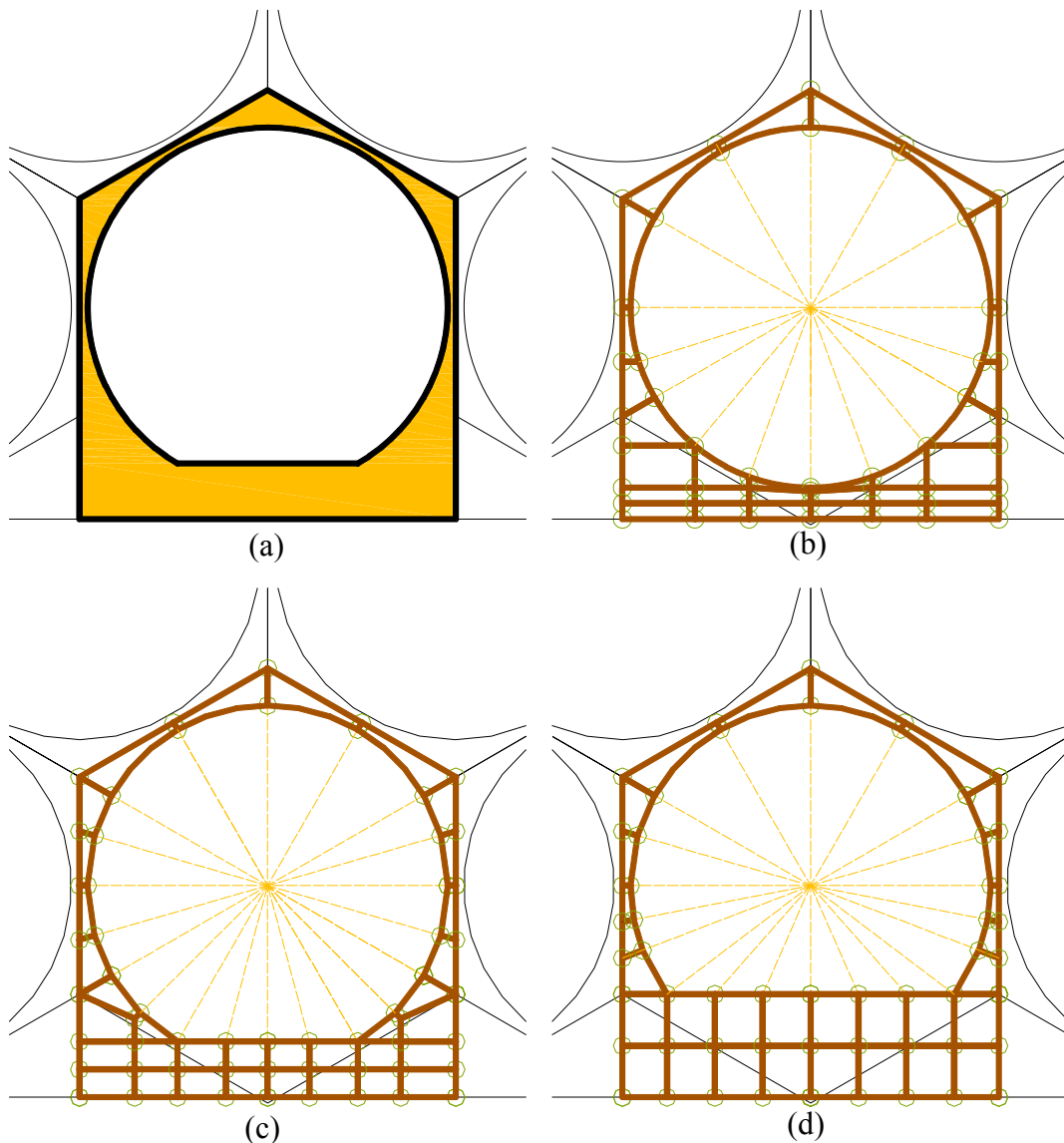


Figura III-48. Celda Circular con Chaflán Frontal. (a) Forma general de la celda. (b) a (d) División en cuadriláteros según el grado de chaflán

Celdas *Circular-Mitad con Chaflán Frontal*

Se repiten los mismos casos que en la celda completa, y la sección es la misma que la mostrada en la **Figura III-48**, pero aplicando la simetría vertical, **Figura III-49**.

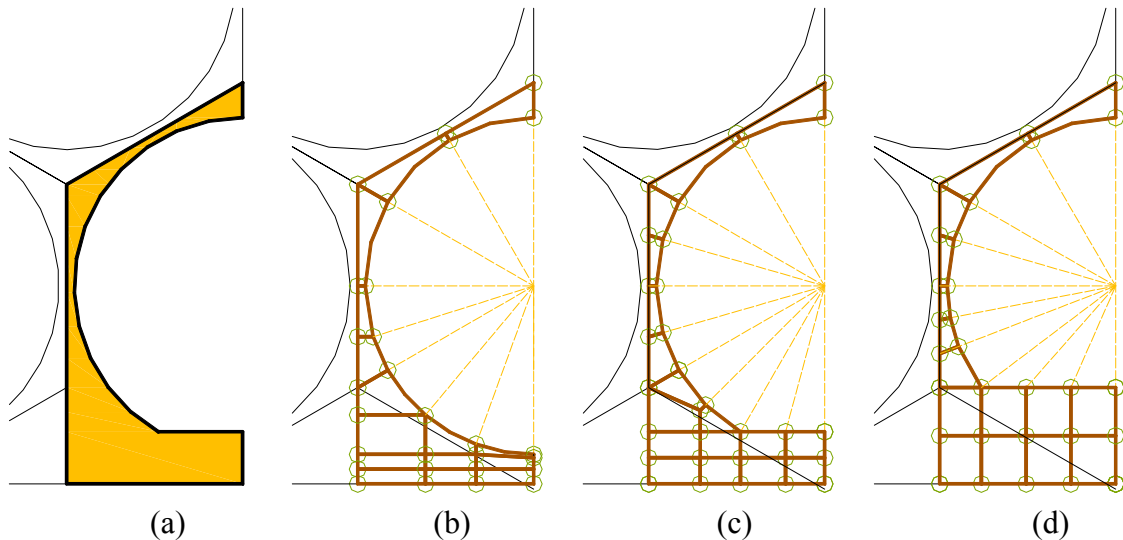


Figura III-49. (a) Celda *Circular-Mitad con Chaflán Frontal*. (b) y (d) División en 15 cuadriláteros, (c) división en 17 cuadriláteros.

Celdas *Circular-Ovoide con Chaflán Frontal*

Se repiten los mismos tres casos que en la celda *Circular con Chaflán Frontal* y la sección es la misma que la mostrada en la **Figura III-48**, con la salvedad de que para unir adecuadamente el lado en contacto con el ovoide hace falta introducir dos cuadriláteros más. Además, se desplaza el lado común a la mitad de la distancia entre ambas celdas. En la **Figura III-50** se ilustra el resultado de combinar el caso (b) de chaflán frontal con un ovoide de esquina, **Figura III-47**.

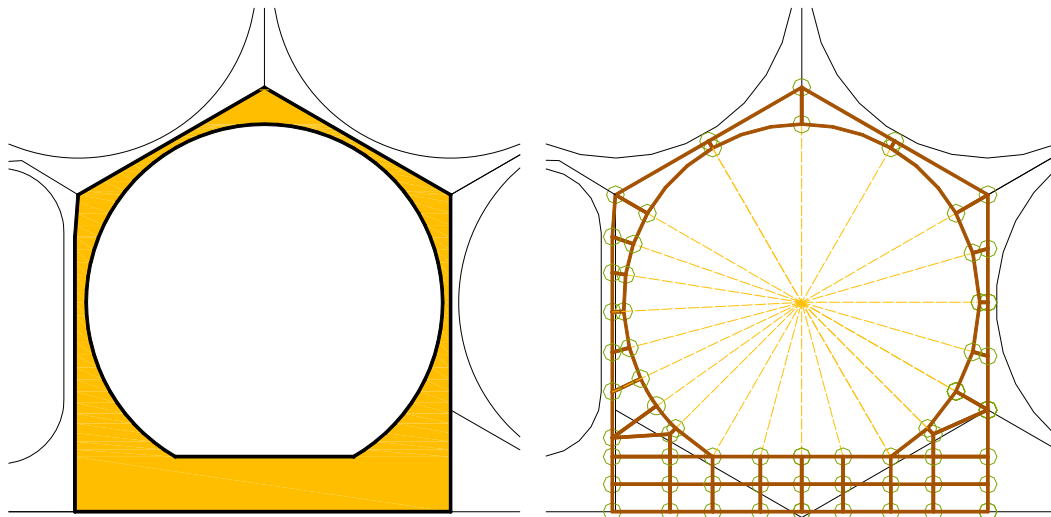


Figura III-50. Celda *Circular-Ovoide con Chaflán Frontal*. División en 36 cuadriláteros, con ovoide corto de esquina

Celdas tipo *Circular con Chaflán Lateral*

Se presentan dos casos posibles, en función del grado de achaflanamiento lateral: Si éste es 0, **Figura III-51.b**, o si es mayor que 0, **Figura III-51.c**. En ambos casos, la zona adyacente con celdas ovoides se desplaza a la mitad del espesor (véase **Figura III-45**). La posición de los puntos situados en la pared lateral es variable, para permitir la unión con los topes dispuestos.

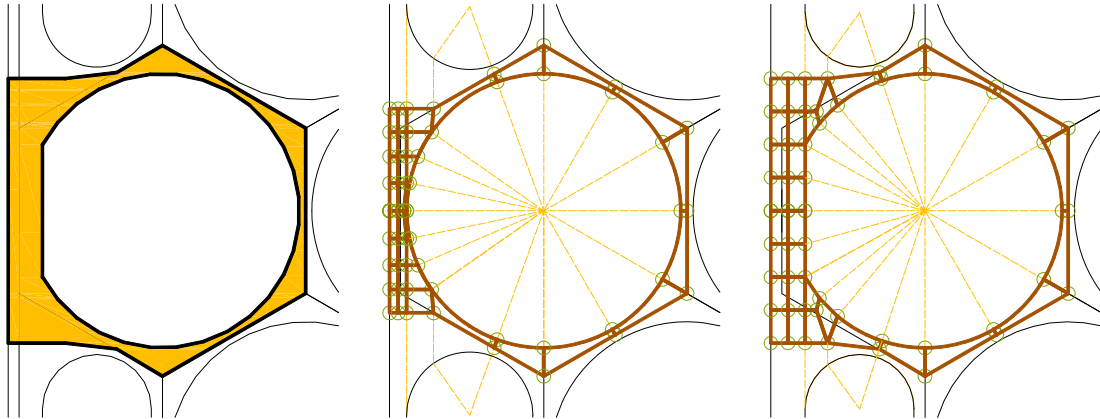


Figura III-51. Celda *Circular con Chaflán Lateral*. (a) Forma de la celda. (b) División en 34 cuadriláteros, sin chaflán (c) División en 32 cuadriláteros con chaflán

Celdas tipo *Circular de Esquina*

Teniendo presente las posibles variaciones del grado de achaflanamiento frontal y lateral, y en función de que el corte de los chaflanes en la celda origine un arco (**Figura III-10.a**) o un ángulo recto (**Figura III-10.b**), se presentan una serie de casos básicos que deben ser tratados por separado.

Los cuadrantes 1 y 4 de la celda se corresponden, en función del chaflán frontal, con los 3 casos de celdas *Circulares con Chaflán Frontal* de la **Figura III-48**. El cuadrante 2 es similar al de la celda con Chaflán Lateral, **Figura III-51**. El tercer cuadrante dependerá claramente de la existencia de arco o ángulo recto, según se indicaba en las ecuaciones (III-20) y (III-21). Los puntos de la pared lateral se pueden adaptar a los puntos del tope.

- Caso 1, $e_{ChaFro} = 0$. Dos posibles situaciones, **Figura III-52**, con 51 cuadriláteros en ambas.
- Caso 2, $0 < e_{ChaFro} < r - \frac{L}{2}$. Tres posibles situaciones, **Figura III-52**, con 51 cuadriláteros en la primera y en la tercera, y 46 en la segunda.
- Caso 3, $r - \frac{L}{2} \leq e_{ChaFro}$. Tres posibles situaciones, **Figura III-52**, con 44 cuadriláteros en los tres casos.

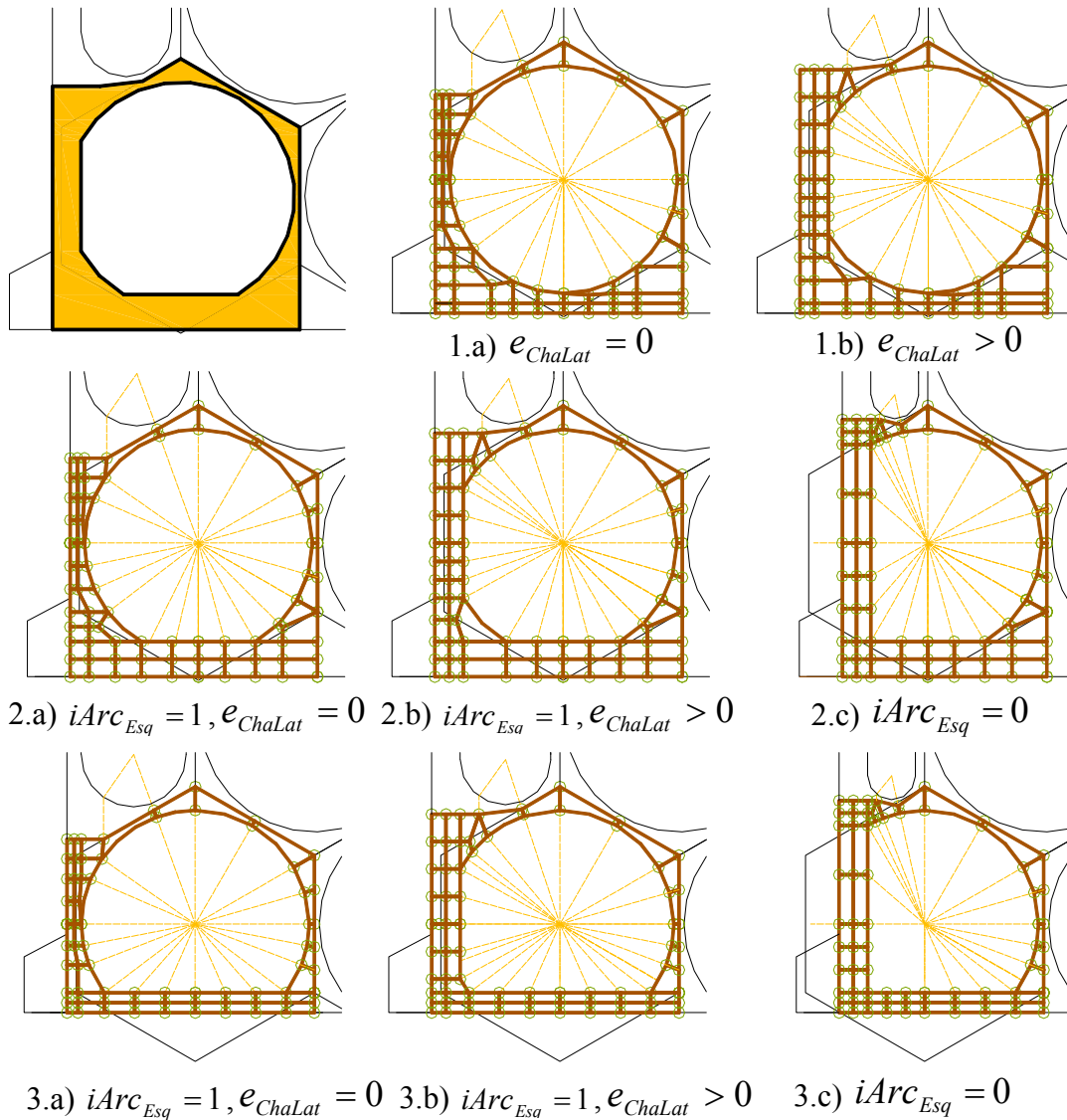


Figura III-52. Celdas de *Esquina*, casos según grado achaflanamiento paredes

Celdas tipo *Tope*

Aunque no son celdas aligeradas, se deben tener presentes en la sección, según sus dimensiones y se disponga o no doble tope en Lado Tierra y/o Lado Mar. Dado que pueden afectar a una o varias celdas contiguas y que éstas tienen configuraciones diversas según el grado de achaflanamiento y el espesor de las paredes, el tope deberá adoptar la misma discretización que la celda adyacente en cuanto al número de nodos a colocar. Será necesario ajustar el nodo final del tope en la celda vecina.

La **Figura III-53** muestra un ejemplo de discretización del tope a partir de las celdas adyacentes de esquina y ovoidal, en las que únicamente los nodos más cercanos a los extremos de los topes se han movido.

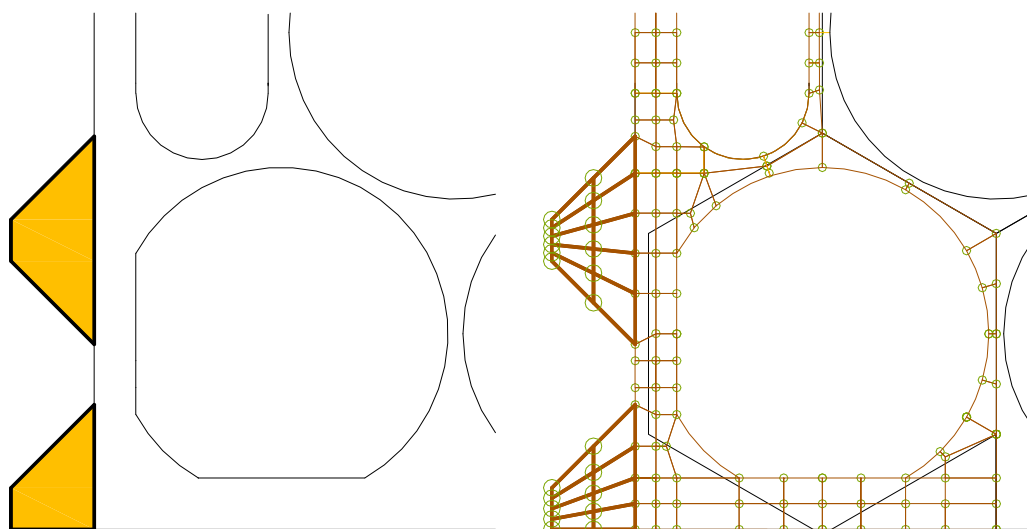


Figura III-53. Celda *Tópe*. División en cuadriláteros según celdas adyacentes

III.2.2.- Gestión del mallado de cada celda

Como se ha mencionado, el algoritmo de generación de cuadriláteros debe calcular la posición de sus vértices, las líneas rectas o curvas que los unen, la superficie formada y el volumen extruido para formar un hexaedro, para cada todos y cada uno de los cuadriláteros que definen la sección.

Es preciso almacenar toda esta información para cada celda, de modo que se disponga de la misma cuando se vaya a generar el fichero de la geometría completa del cajón a partir de todos y cada uno de los hexaedros generados.

Dada la estructura jerárquica que presenta la información que describe el cajón, el algoritmo desarrollado hace uso de la filosofía de la programación orientada a objeto, en la que un objeto es un contenedor de las propiedades y las variables propias así como de las interrelaciones entre el objeto y los demás.

A continuación se describen diversos paquetes⁸ y clases con los que se realizará la programación óptima de esta gestión.

III.2.2.1.- Entidades básicas

Atendiendo a las posibles entidades geométricas a desarrollar, se detallan las clases incluidas en el paquete *Entidades*, que servirán de soporte para la generación y almacenado de toda la información relativa. Los datos guardados en estas clases tienen relación con la forma de tratar GiD[®] las curvas, las superficies Coons⁹ [120] y los volúmenes, y cómo debe ser exportada la geometría a su formato nativo (fichero **.geo*) [122].

⁸ Un paquete en Java, o *package*, es un contenedor de clases que permite agrupar elementos de cierta afinidad.

⁹ Una superficie Coons, en honor a Steven. A. Coons, es la obtenida al interpolar cuatro curvas, que se cortan dos a dos en cuatro vértices, en un plano o en el espacio [121]

Clase CPunto

Es la clase que contiene como variables la información de numeración local, coordenadas espaciales y número de líneas que lo conectan para un punto dado.

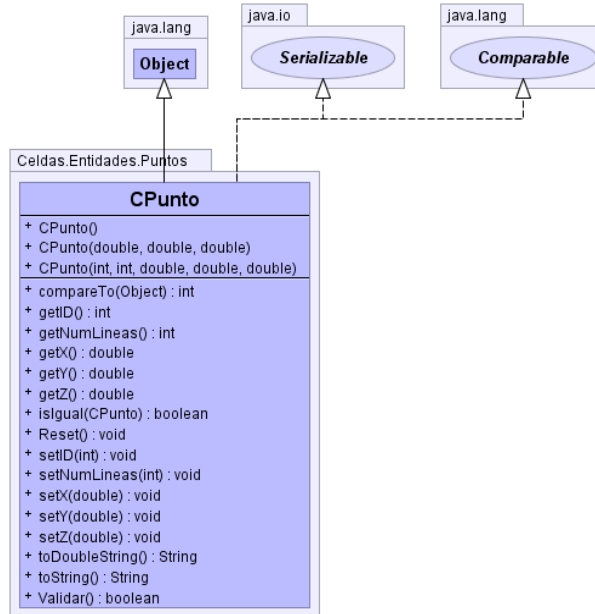


Figura III-54. Diagrama UML de la clase CPunto

Clase CColPuntos

La clase CColPuntos agrupa todos los puntos que pertenecen a la misma celda básica. Se trata, por tanto, de una colección¹⁰ de objetos CPunto. Contiene los métodos para ir añadiendo puntos a la colección a medida que se van generando, para localizar un punto dado, para modificar algún parámetro de dicho punto (como el número de líneas conectadas a medida que éstas se van creando, o sus coordenadas si es necesario).

¹⁰ Esta colección fue generada usando la clase Vector y Java 1.4. No existían las colecciones genéricas, que serían introducidas en Java 1.5. Hoy en día Vector se considera heredado o anticuado (*legacy*) y debería ser reemplazado por una colección sincronizada [119]:

```
m_vPuntos = Collections.synchronizedList(new ArrayList<CPunto>());
```



Figura III-55. Diagrama UML de la clase CColPuntos

Clase CLinea

La clase CLinea, contiene además de la numeración y número de superficies que la conectan, las clases CPunto inicial y CPunto final que definen una línea. Incluye además un parámetro que identifica si la línea está posicionada sobre una garganta.

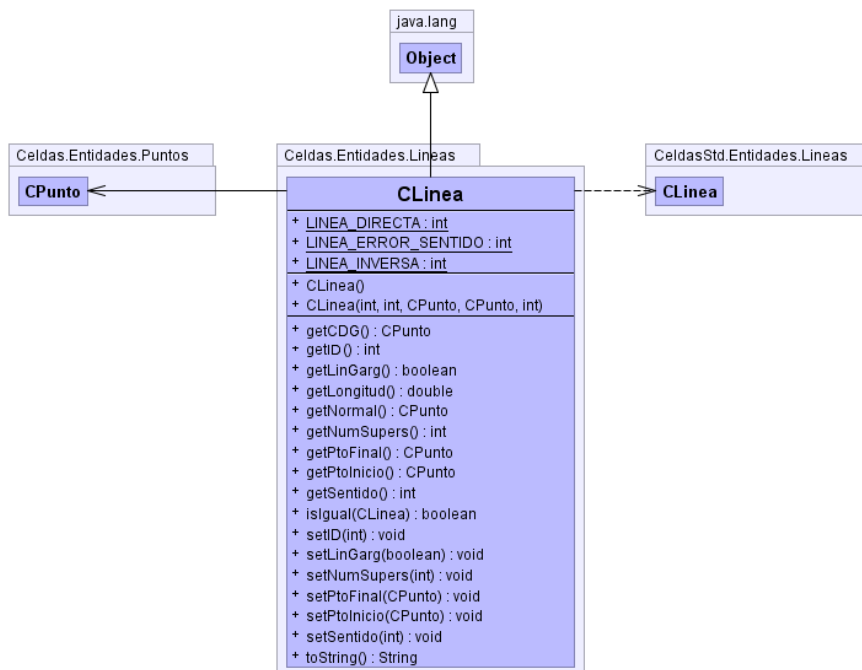


Figura III-56. Diagrama UML de la clase CLinea

Clase CColLineas

La clase CColLineas agrupa todas las líneas que pertenecen a la misma celda básica. Se trata, por tanto, de una colección de objetos CLinea. Contiene los métodos para ir añadiendo líneas a la colección a medida que se van generando, para localizar una línea dado, para modificar algún parámetro de dicha línea (como el número de superficies conectadas a medida que éstas se van creando).

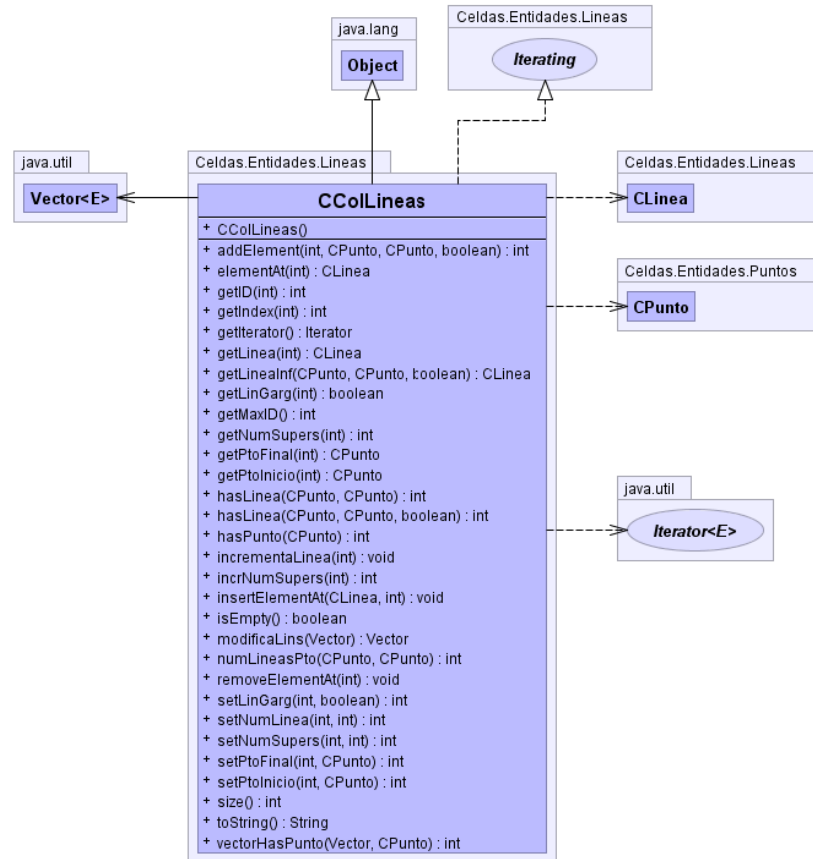


Figura III-57. Diagrama UML de la clase CColLineas

Clase CArco

De manera similar a la clase CLinea, en esta clase se guardan los puntos inicial y final del arco, además del que define el centro del mismo, su radio y ángulos inicial y final asociados. También se guardan la dirección normal del punto medio del arco y el sentido de recorrido.



Figura III-58. Diagrama UML de la clase CArco

Clase CColArcos

La clase CColArcos agrupa todos los arcos que pertenecen a la misma celda básica. Se trata, por tanto, de una colección de objetos CArco. Contiene los métodos para ir añadiendo arcos a la colección a medida que se van generando, para localizar un arco dado, para modificar algún parámetro de dicho arco (como el número de superficies conectadas a medida que éstas se van creando).



Figura III-59. Diagrama UML de la clase CColArcos

Clase CSuperficie

Esta clase contiene los 4 elementos que definen el cuadrilátero de la sección transversal: bien cuatro rectas o bien tres rectas y un arco. Se guardan además el sentido de recorrido de cada una de las cuatro líneas, los puntos correspondientes al centro de gravedad de esta superficie, calculado de forma automática para cualquier condición, y de la celda completa, el área de la superficie, y su dirección normal.

Igualmente, se pueden guardar los datos de superficies verticales: bien cuatro rectas o bien dos rectas y dos arcos, y la posibilidad de que esté situada sobre una garganta.

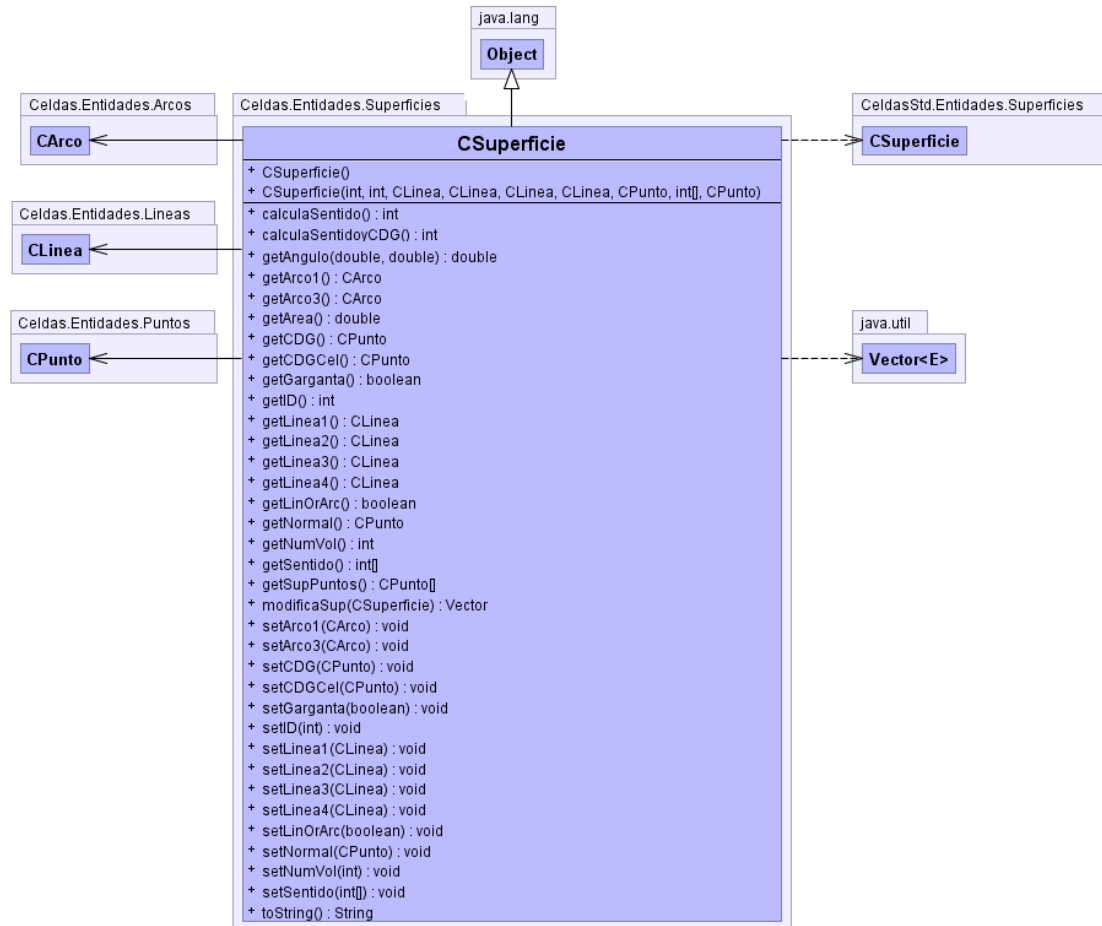


Figura III-60. Diagrama UML de la clase CSuperficie

Clase CColSuperficies

La clase CColSuperficies agrupa todas las superficies que pertenecen a la misma celda básica. Se trata, por tanto, de una colección de objetos CSuperficie. Contiene los métodos para ir añadiendo superficies a la colección a medida que se van generando, para localizar una superficie dada, para modificar algún parámetro de la misma (como el número de volúmenes conectados a medida que éstos se van creando).



Figura III-61. Diagrama UML de la clase CColSuperficies

Clase CVolumen

Contiene las seis superficies que conforman el volumen hexaédrico generado: la generada en la celda básica, o inferior, la superior idéntica pero desplazada según la altura de extrusión, y las cuatro superficies verticales definidas sobre las 4 líneas perimetrales. Igualmente contiene el sentido de cada una de estas superficies y el centro de gravedad del volumen.

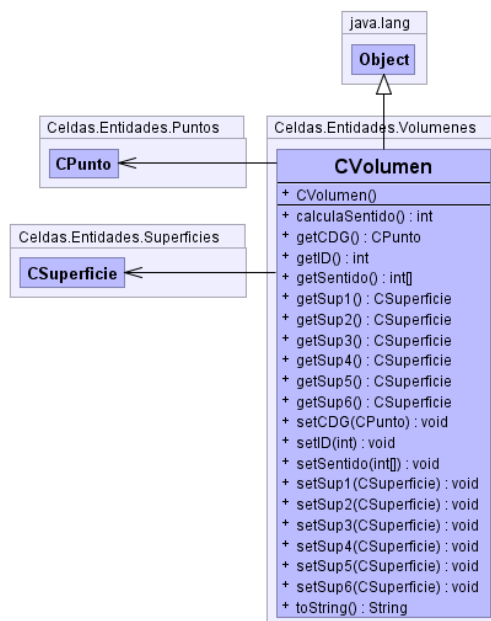


Figura III-62. Diagrama UML de la clase CVolumen

Clase CCo1Volúmenes

Por último, la clase CCo1Volúmenes agrupa todos los volúmenes que se han extruido verticalmente a partir de las superficies definidas en una celda básica. Se trata, por tanto, de una colección de objetos CVolumen. Contiene los métodos para ir añadiendo volúmenes a la colección a medida que se van generando.

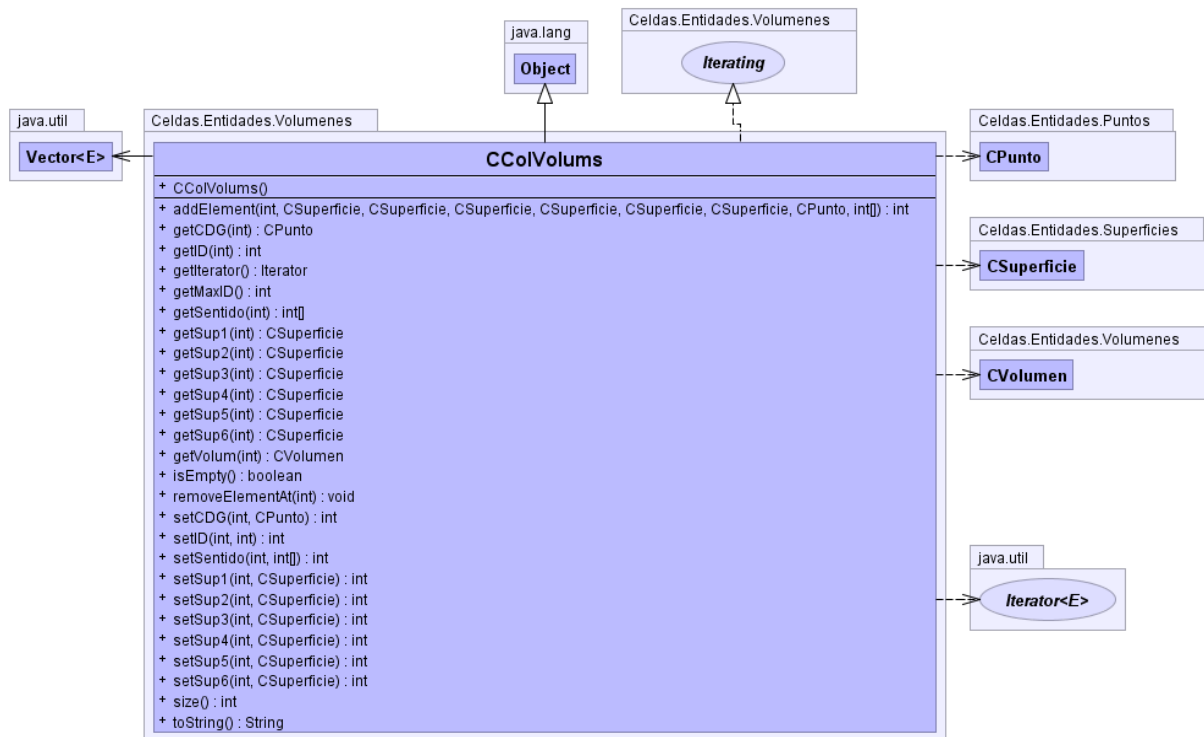


Figura III-63. Diagrama UML de la clase CCo1Volúmenes

III.2.2.2.- Generación del contenido de las celdas

Por otro lado, se crea, dentro del paquete Celdas, la clase CeldaBean¹¹ como entidad genérica que contiene todas las listas simples de objetos que definen una celda: puntos, líneas, arcos y superficies, así como sus métodos *get* y *set* para acceder o modificar estas propiedades.

¹¹ Como su nombre indica, es un JavaBean [123], o componente de software reutilizable y con soporte de serialización de Java, que encapsula muchos objetos en uno solo, a modo de vaina o *bean* según la acepción inglesa.

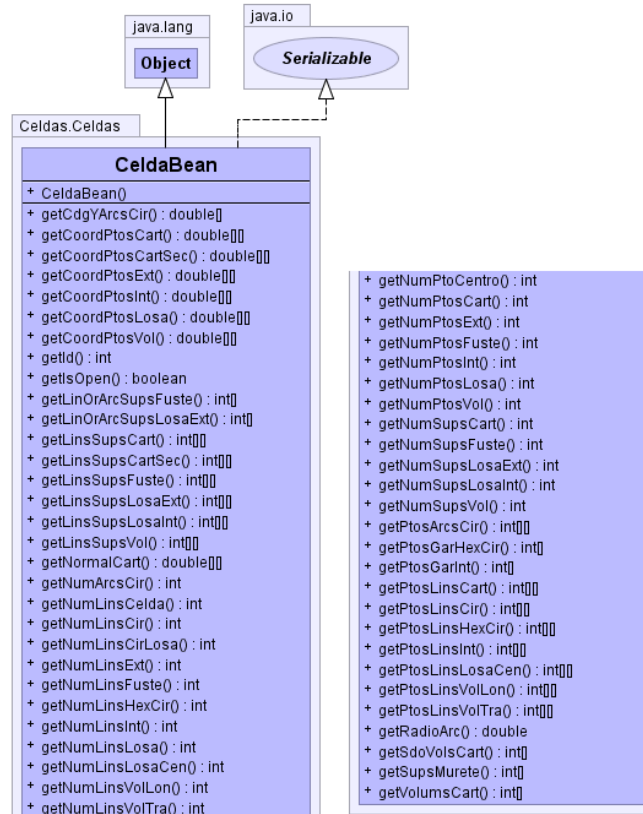


Figura III-64. Diagrama UML de la clase CeldaBean (indicados sólo los métodos get)

Interfaz¹² CeldaFactory

Se define esta interfaz con los métodos comunes que van a tener todas las diferentes tipologías de celdas. La clase CDatos contiene las variables básicas del cajón que han sido descritas en el apartado III.1.11.-.

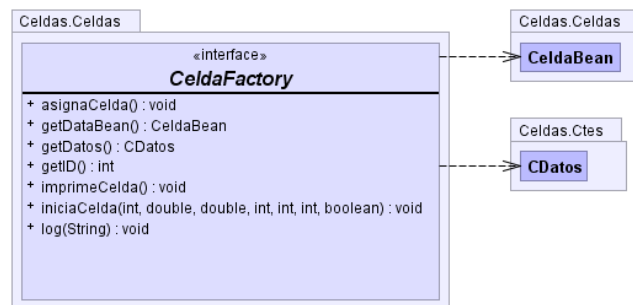


Figura III-65. Diagrama UML de la interfaz CeldaFactory

El método principal es asignaCelda(). Para cada tipología de aligeramientos, se encargará de generar todos los datos específicos de la celda (puntos, líneas, arcos y superficies), a partir de lo indicado en III.2.1.3.- y guardarlos en un objeto CeldaBean.

¹² Una interfaz de Java es un tipo abstracto que contiene únicamente métodos abstractos no implementados que, a modo de contrato, obligan a ser declarados y descritos en las clases que implementan esta interfaz. De este modo, todas estas clases pueden ser agrupadas y referenciar el acceso a estos métodos por su interfaz en lugar de por su tipo propio.

Clase abstracta *CeldaBase*

De forma genérica, una celda tendrá un objeto *CDatos*, con los datos generales del cajón, un objeto *CeldaBean*, que encapsulará las listas específicas de datos de la celda (puntos, líneas, arcos y superficies), y un objeto *CCelda*, instancia de la clase interna que encapsula las colecciones (*CColPuntos*, *CColLíneas*, *CColArcos* y *CColSuperficies*).

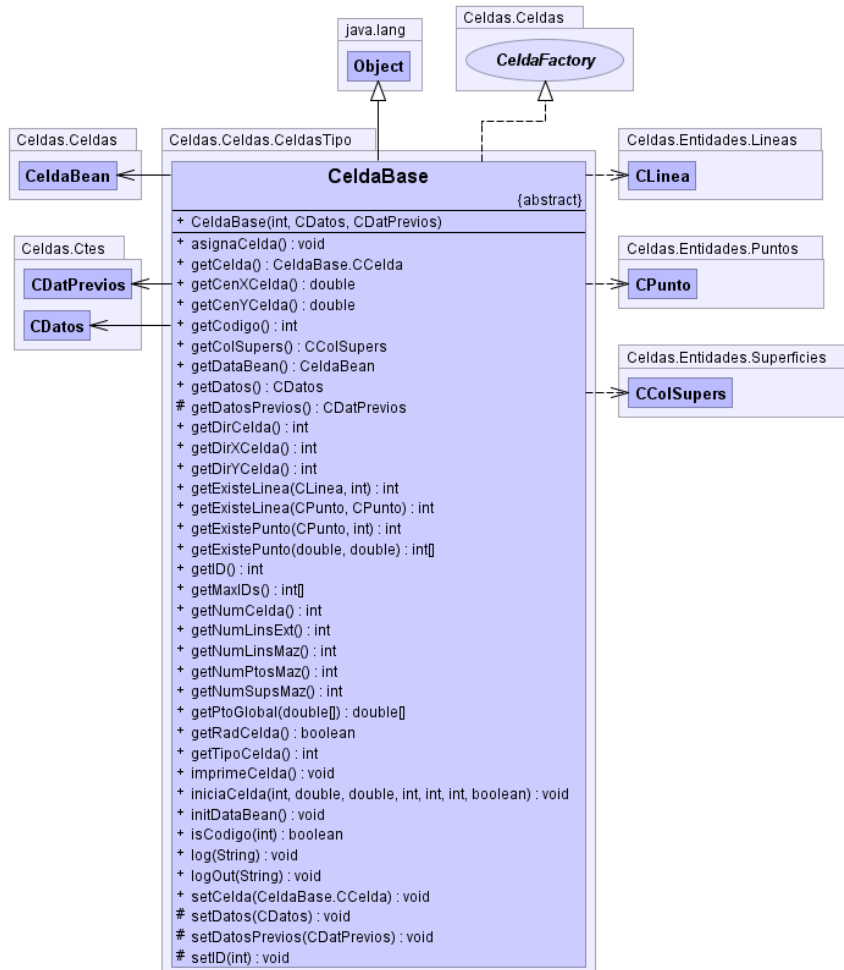


Figura III-66. Diagrama UML de la clase abstracta *CeldaBase*

Si bien implementa *CeldaFactory*, el método *asignaCelda()* será desarrollado en cada una de las clases hijas que hereden esta clase abstracta, en función de su tipología.

Clases de las diferentes tipologías

Para cada una de las tipologías se define una clase que hereda de *CeldaBase* e implementa la interfaz *CeldaFactory*. Por ejemplo, la celda de tipología circular sigue el diagrama mostrado en la **Figura III-67**. El resto de tipologías de la **Figura III-68** siguen un diagrama idéntico.

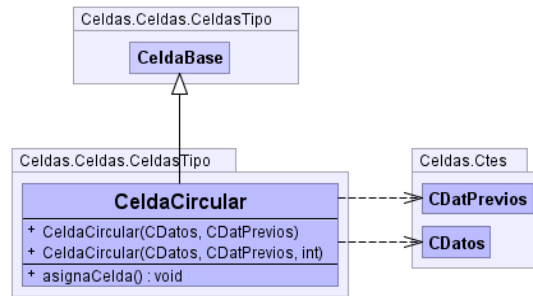


Figura III-67. Diagrama UML de la clase CeldaCircular

En su implementación del método `asignaCelda()`, cada clase contiene detalladamente las ecuaciones que permiten el cálculo de las coordenadas de los puntos de los cuadriláteros especificados en III.2.1.3.-. A partir de estos puntos, se listan las líneas y arcos, y con estos se definen las superficies. Todos estos datos, finalmente se guardan en el objeto `CeldaBean`.

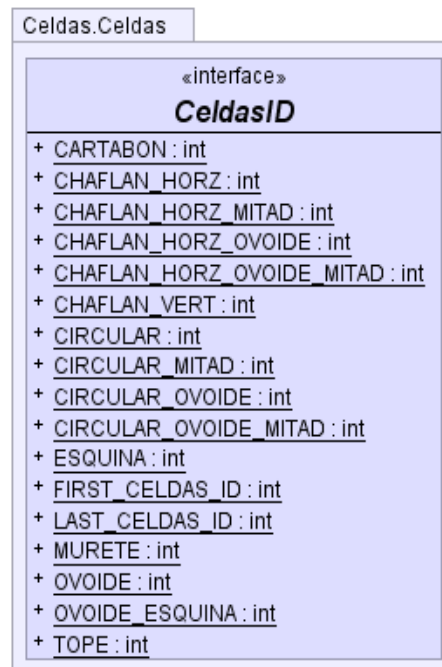


Figura III-68. Interfaz CeldasID con los tipos posibles de celdas

III.2.2.3.- Contenedores

Una vez que se ha generado la información completa de cada celda, pero de manera aislada, el siguiente paso consiste en generar la información relativa a la sección transversal completa, integrando todas y cada una de las celdas. Por un lado, la forma en la que se han discretizado individualmente cada una de ellas garantiza que, al unir las, las aristas comunes compartan dicha discretización, pero por otro, será preciso establecer una numeración única, de modo que no haya elementos repetidos.

Se empleará el algoritmo de la **Figura III-15** y será necesario disponer de un objeto que permita almacenar la información parcial de cada una de las celdas. Se crea, por ello, un contenedor de objetos tipo `CeldaFactory`.

Interfaz `MultiMap`

Es una adaptación de la colección `Map` del Java Collections Framework [124], en la que se define un mapa que contiene una colección de valores para cada clave o *key*. De este modo, si la clave es el tipo de celda (por ejemplo, Celda tipo *Circular*), bajo esta clave se podrán almacenar todas las celdas circulares de la sección.

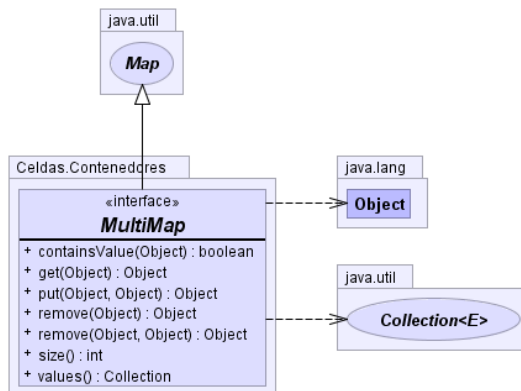
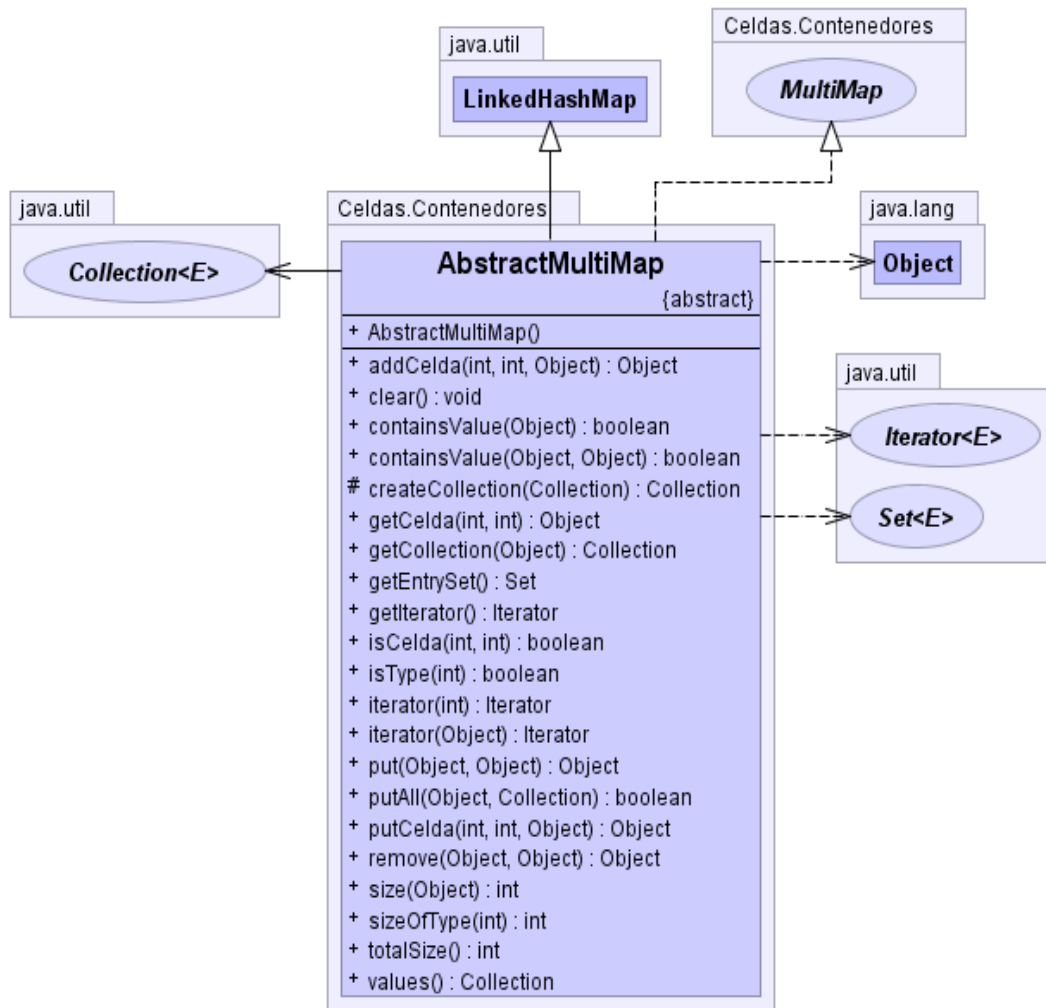


Figura III-69. Diagrama UML de la interfaz `MultiMap`

Clase abstracta `AbstractMultiMap`

Los métodos genéricos definidos en la interfaz `MultiMap` se implementan en esta clase abstracta. Así, los métodos `addCelda()` y `getCelda()` permiten añadir o extraer una celda completa de la colección, en función de la clave especificada, según los tipos recogidos en la **Figura III-68**.

Figura III-70. Diagrama UML de la clase *AbstractMultiMap*

Clase *CMultiMapCeldas*

Esta clase, que hereda de *AbstractMultiMap* y por tanto implementa *MultiMap*, es el contenedor de todas las celdas que resultan de la aplicación del algoritmo de la **Figura III-15**, donde se determina el tipo y el centro de referencia de todas y cada una de las celdas de las distintas filas de aligeramientos.

Tras el dimensionamiento inicial del mapa y la asignación de las celdas de la sección transversal, ordenadas por tipología y dentro de cada tipología, por orden de aparición empezando por la primera celda del Lado Tierra hasta la última del Lado Mar, procede calcular el contenido de dichas celdas.

Así, recorriendo el mapa, se extrae cada celda, se llama al método *asignaCelda()*, con el que se calculan, de manera específica a la tipología que representa, los datos que definen la geometría, y se almacenan en su *CeldaBean* en la propia celda. Finalmente, la celda modificada se vuelve a guardar en la misma posición del mapa.

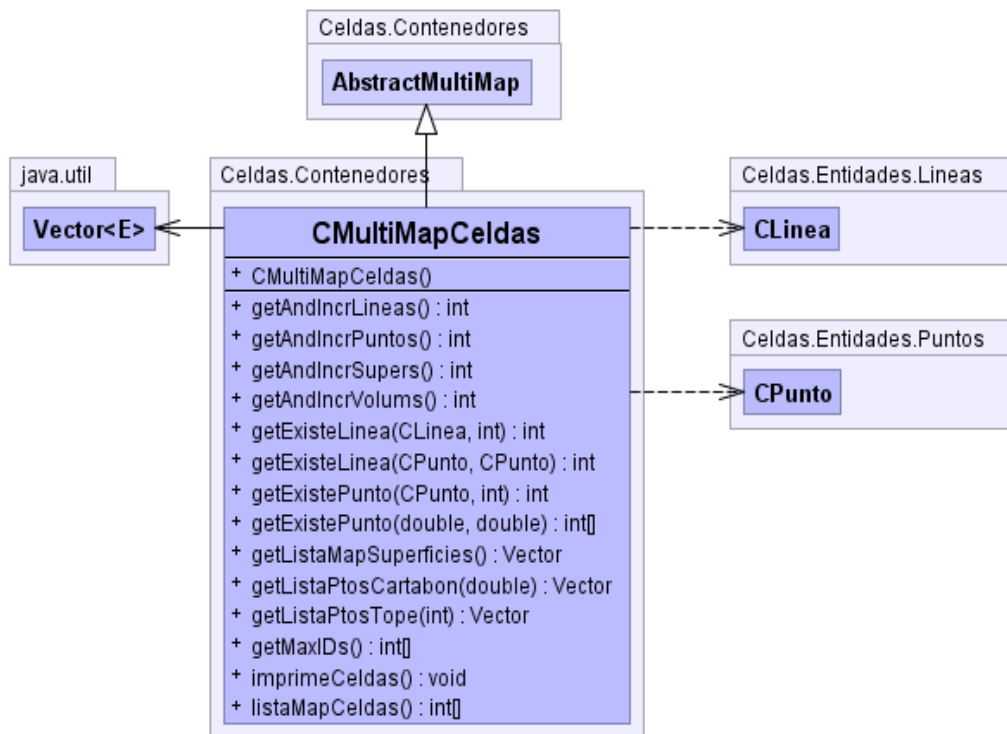


Figura III-71. Diagrama UML de la clase CMultiMapCeldas

En el caso particular de que la celda sea de tipo *Tope*, se recorre de nuevo el mapa en busca de los puntos de la pared lateral ya guardados en las celdas de tipo *circular de esquina*, *ovoide*, *ovoide de esquina*, o *circular con chaflán lateral*, ya que serán necesarios para realizar el mallado estructurado del tope. Adicionalmente, se ajustan los extremos de los topes, modificando convenientemente las coordenadas de los puntos en dichas celdas.

En una siguiente iteración se vuelve a recorrer el mapa, para cada celda se extraen los datos de *CeldaBean* y se van creando las colecciones de puntos, líneas, arcos y superficies. En esta ocasión, es imprescindible realizar una asignación única de etiquetas a todas las entidades generadas en el proceso, para que no existan elementos repetidos. Por este motivo, cada vez que se crea una nueva entidad en el perímetro de la celda, se debe buscar en las entidades equivalentes de celdas ya procesadas (puntos y líneas), y tomar las etiquetas de estas, en caso de que existan. Se guardan en *CCelda*, y se guarda la celda modificada de nuevo en el mapa.

Clase CCubo

Por analogía con la clase *CCelda*, la clase *CCubo* contiene las colecciones de puntos para la altura $z0$ y para la altura $z1$, las colecciones de líneas horizontales para la altura $z0$ y para la altura $z1$ y verticales uniendo puntos de una y otra altura, las colecciones de arcos para la altura $z0$ y para la altura $z1$, las de superficies horizontales para la altura $z0$ y para la altura $z1$, y las verticales uniendo líneas de una y otra altura, y finalmente las colecciones de volúmenes.



Figura III-72. Diagrama UML de la clase CCubo

Clase **CMultiMapCubos**

Esta clase, que hereda de **AbstractMultiMap** y por tanto implementa **MultiMap**, es el contenedor de todas las celdas que resultan de la aplicación del algoritmo de la **Figura III-15**, y que serán extruidas para formar volúmenes hexaédricos a partir de los cuadriláteros definidos, entre una altura z_0 y una altura z_1 , una vez que se ha calculado el mapa de celdas **CMultiMapCeldas**.



Figura III-73. Diagrama UML de la clase **CMultiMapCubos**

Así, recorriendo de nuevo el mapa de celdas **CMultiMapCeldas** se extrae cada celda, se genera el cubo **CCubo**, y se guarda en un nuevo mapa **CMultiMapCubos**, específico para la rebanada de altura z_0 a z_1 .

En este proceso, de nuevo es importante no repetir entidades, por lo que en todo momento hay que ir comprobando que la generación de las nuevas adopte las etiquetas

de las ya existentes tanto en los cubos adyacentes como en los de mapas inmediatamente inferiores.

Clase *singleton* CCajon

Todos los mapas de cubos que se generen para las diferentes alturas deben ser almacenados en un nuevo contenedor, la clase CCajon. Se define usando el patrón singleton de manera que su información pueda ser accesible desde cualquier otra parte del programa.

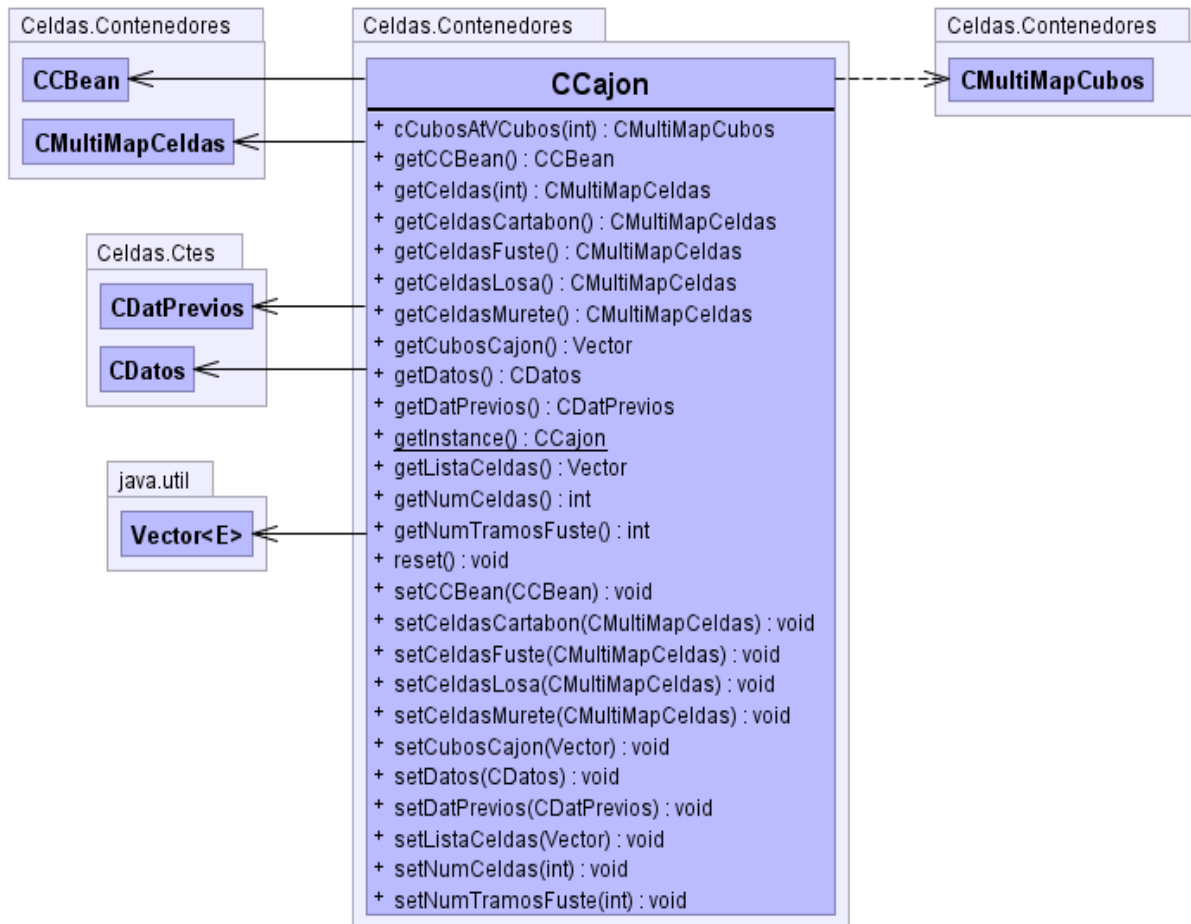


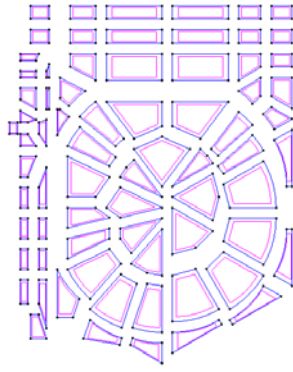
Figura III-74. Diagrama UML de la clase CCajon

Como se verá más adelante, la clase CCBean recogerá las condiciones de contorno en desplazamientos y cargas sobre el cajón.

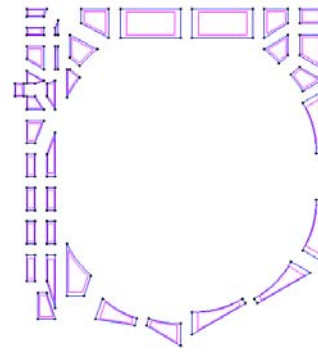
Básicamente en este contenedor se guardan los mapas de celdas del fuste y de la losa, así como el del murete si existe, y en un vector se almacenan todos los mapas de cubos de todas y cada una de las rebanadas definidas.

III.2.2.4.- Resumen de la generación de la geometría

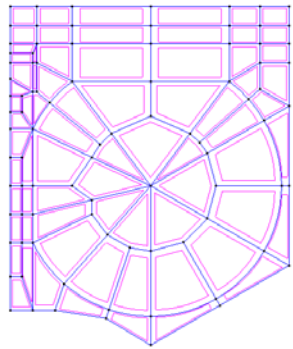
A modo de resumen, las siguientes figuras recogen los distintos pasos seguidos en la generación de una geometría multibloque en bloques de hexaedros.



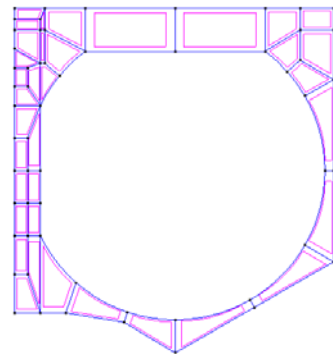
CSuperficie supLosa



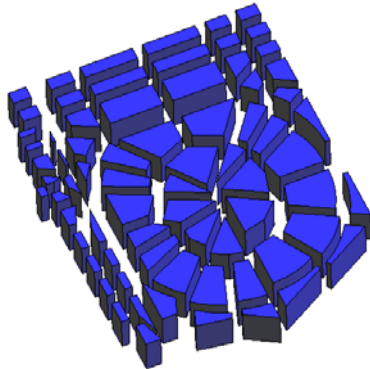
CSuperficie supFuste



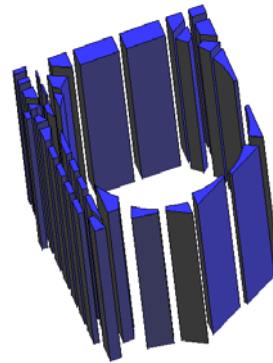
CeldaFactory celdaLosa



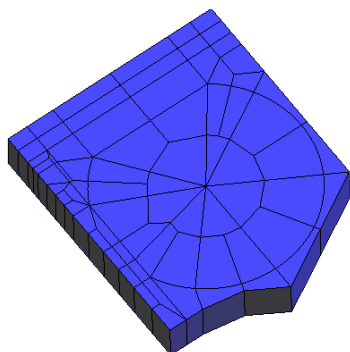
CeldaFactory celdaFuste



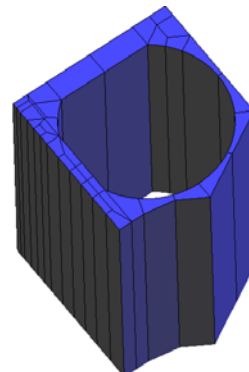
CVolumen volLosa



CVolumen volFuste

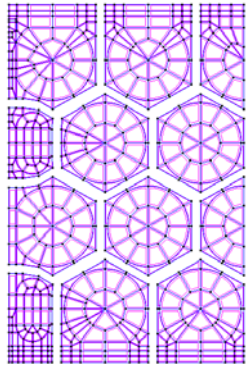


CCubo cuboLosa

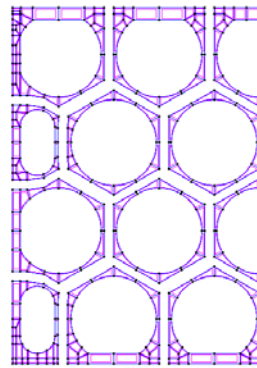


CCubo cuboFuste

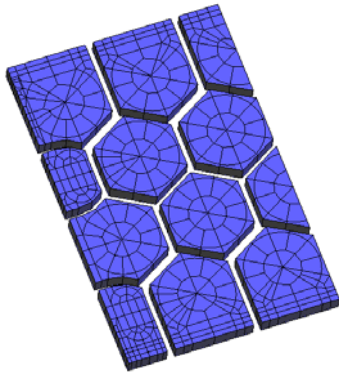
Figura III-75. Resumen de generación de bloques hexaédricos por tipología de celda



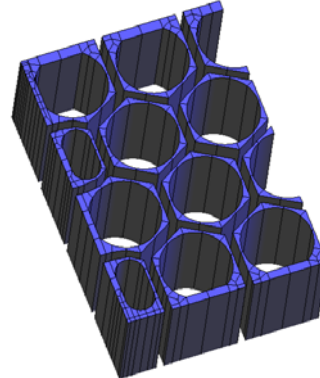
CMultiMapCeldas mapLosa->
CeldaFactory celda



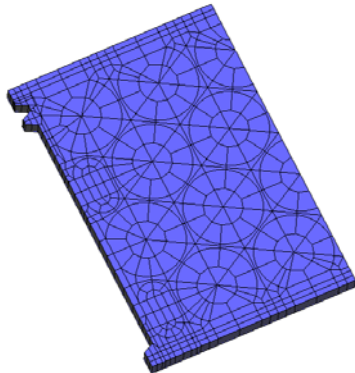
CMultiMapCeldas mapFuste->
CeldaFactory celda



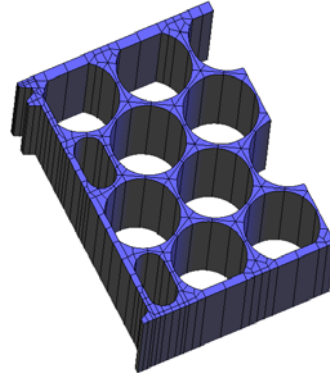
CCubo cuboLosa



CCubo cuboFuste

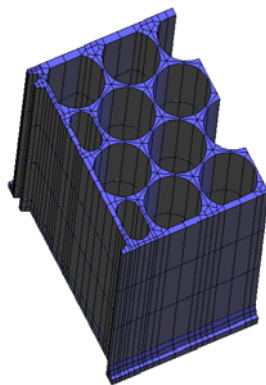


(CMultiMapCubos)vCubosFuste.elementAt(0)



(CMultiMap)vCubosFuste.elementAt(i)

Figura III-76. Resumen de generación de cubos según tipologías por rebanada



Vector vCubosFuste

Figura III-77. Resumen de generación de cubos del cajón según rebanadas

```
CMultiMapCeldas celdasLosa;
CMultiMapCeldas celdasFuste;
Vector vCubosFuste;
```

```
int iNumTramosFuste;
int iNumCeldas;
Vector vListaCeldas;
```

```
CDatos datos;
CDatPrevios dp;
CCBean ccBean;
```

III.2.2.5.- Mallado estructurado resultante

Mediante la aplicación del algoritmo multibloque descrito se genera la geometría *estructurada* de un cajón multicelular, es decir, compuesta exclusivamente por cuerpos hexaédricos. La **Figura III-78** ilustra un corte del fuste y un detalle de una garganta entre dos celdas adyacentes. En esta geometría es sencillo realizar una malla de elementos finitos estructurada.

A diferencia del mallado no estructurado (**Figura III-27**), se conseguirá una malla de mayor calidad, con un número menor de elementos, mucho más controlados, bien posicionados para el post-procesado y, en definitiva, con un coste computacional más reducido.

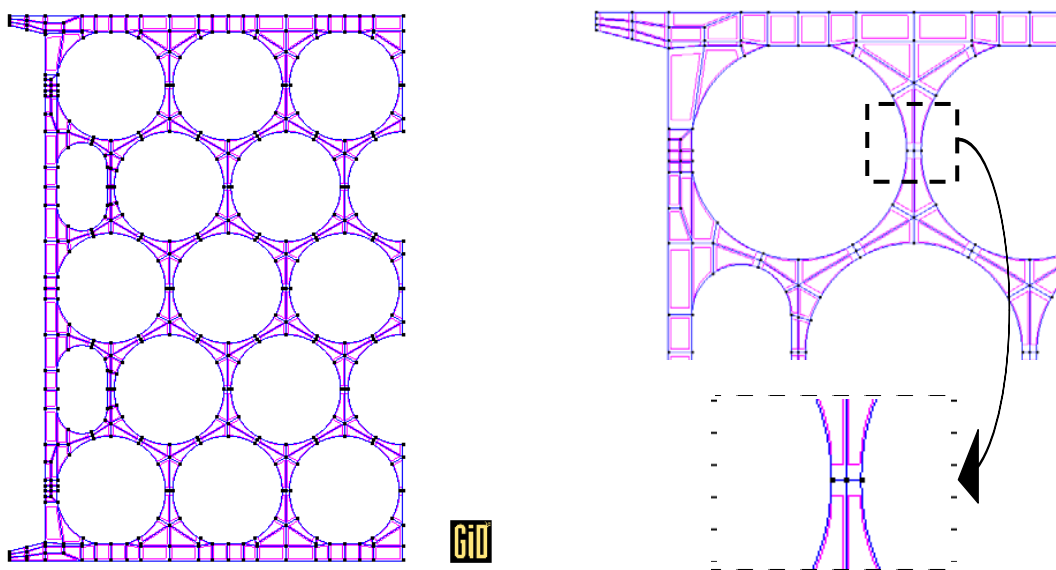


Figura III-78. Sección del fuste generada con el algoritmo multibloque

Así, de cara al armado de las paredes internas del cajón, es preciso conocer el estado tensional en las gargantas, o zonas donde el espesor de la pared es mínimo. Para ello, ha situado alguna arista de estos cuadriláteros sobre las gargantas. De este modo, en el proceso de mallado se generará un número adecuado de nodos en las gargantas para el tratamiento posterior de las tensiones en las mismas.

Se puede observar que la división realizada ya constituye de por sí sola una malla base (la más grosera posible, una malla de tan sólo un elemento por lado, **Figura III-79**) cuya bondad podría ser adecuada para un estudio tensional.

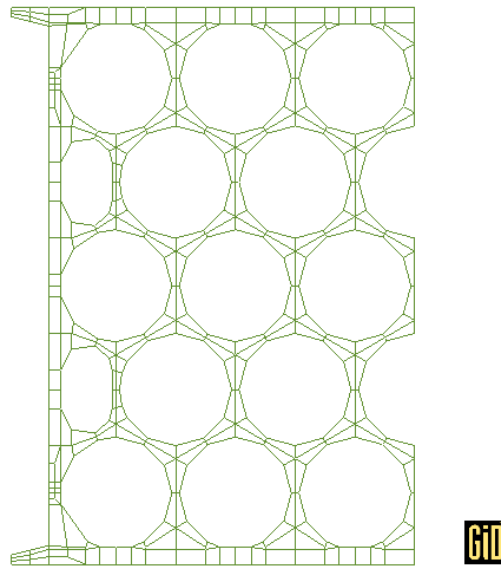


Figura III-79. Malla estructurada. 1 elemento por lado.
290 cuadriláteros y 428 nodos

Esta geometría de partida puede mallarse de forma más fina, dividiendo cada una de las superficies en n_{ELado}^2 elementos, donde n_{ELado} es el número de elementos que comparten un mismo lado. De esta manera, los resultados mejoran notablemente empleando tan solo 2 ó 3 elementos por lado (**Figura III-80** y **Figura III-81**, respectivamente), sobre todo en la aproximación de las gargantas.

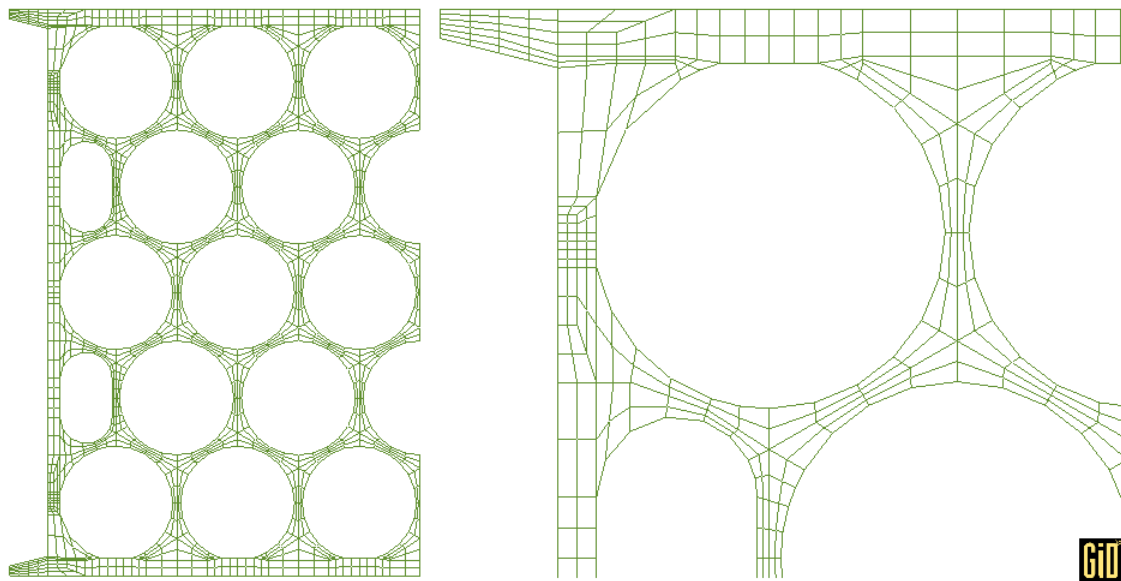


Figura III-80. Malla estructurada. 2 elementos por lado.
1160 cuadriláteros y 1450 nodos

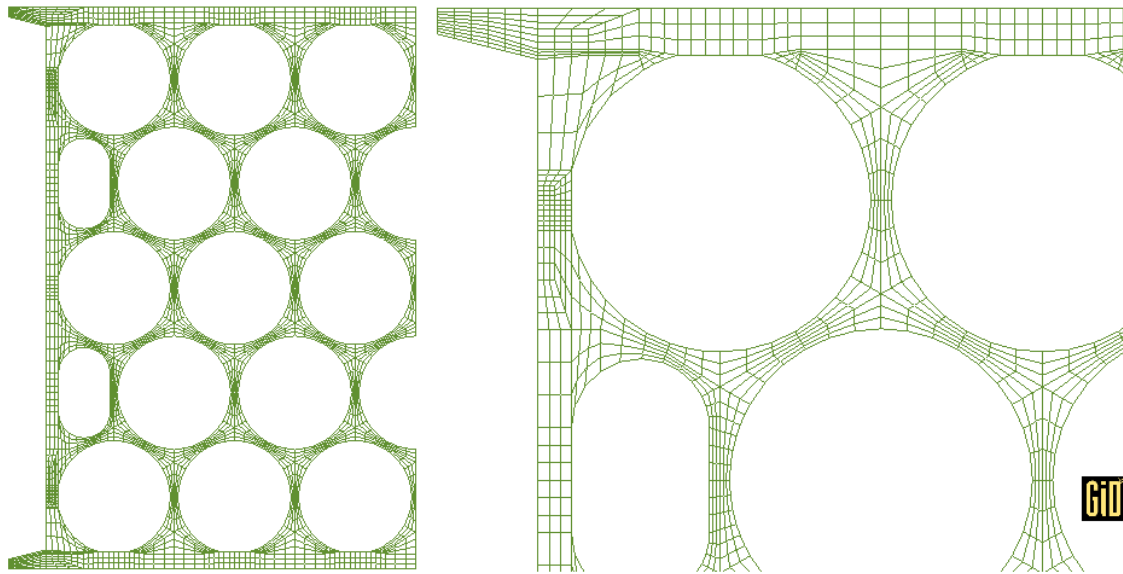


Figura III-81. Malla estructurada. 3 elementos por lado. 2610 cuadriláteros y 3052 nodos

En la **Figura III-82** se compara en el entorno de una celda el mallado realizado de forma estructurada y no estructurada. Se observa que el primero aproxima muy bien las zonas del contorno de la celda y las gargantas, precisamente donde es más importante un buen mallado, mientras que el segundo, en cambio, es más regular en las zonas opuestas, donde la información carece de tanta importancia.

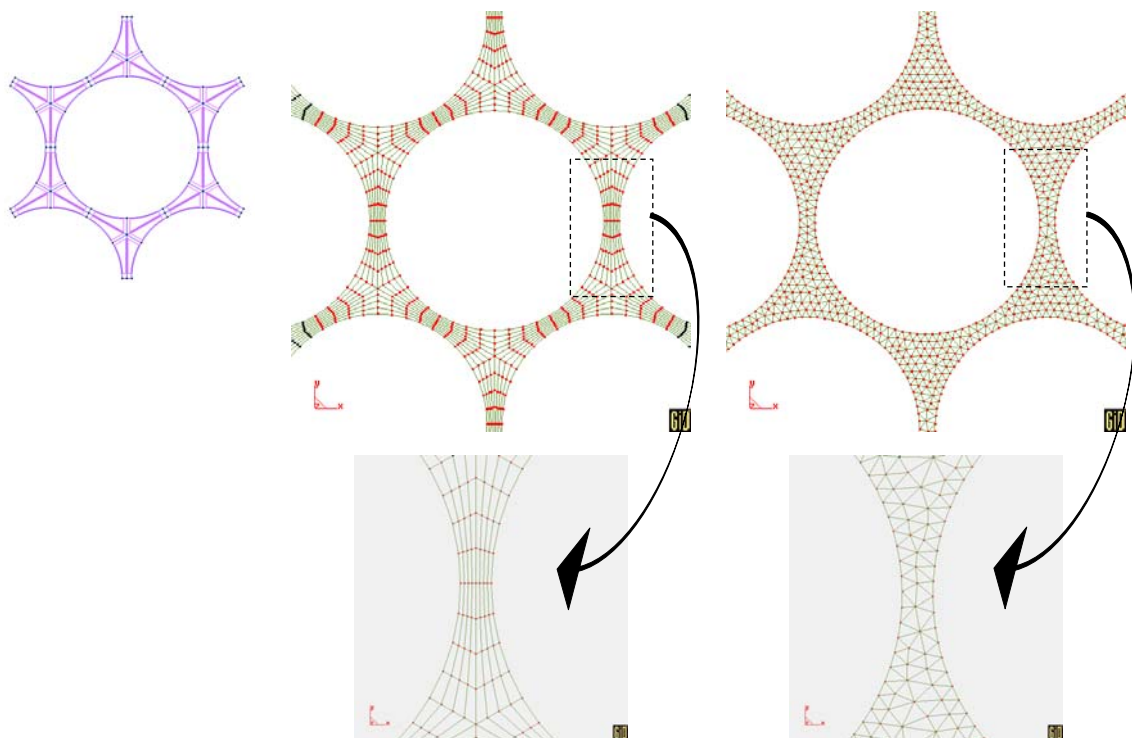


Figura III-82. Mallas finas estructurada (672 nodos) y no estructurada (669 nodos) en el entorno de una celda circular

III.2.3.- Calidad y edición de la geometría multibloque

El algoritmo automático detallado con el que se genera la geometría multibloque, que posteriormente será mallada con elementos hexaédricos, puede generar, en alguna ocasión, elementos con mala relación de aspecto que provoquen una matriz mal condicionada.

Una manera de comprobar si aparecen tales elementos es realizar una medición del mínimo ángulo formado entre dos lados consecutivos del total de los elementos generados.

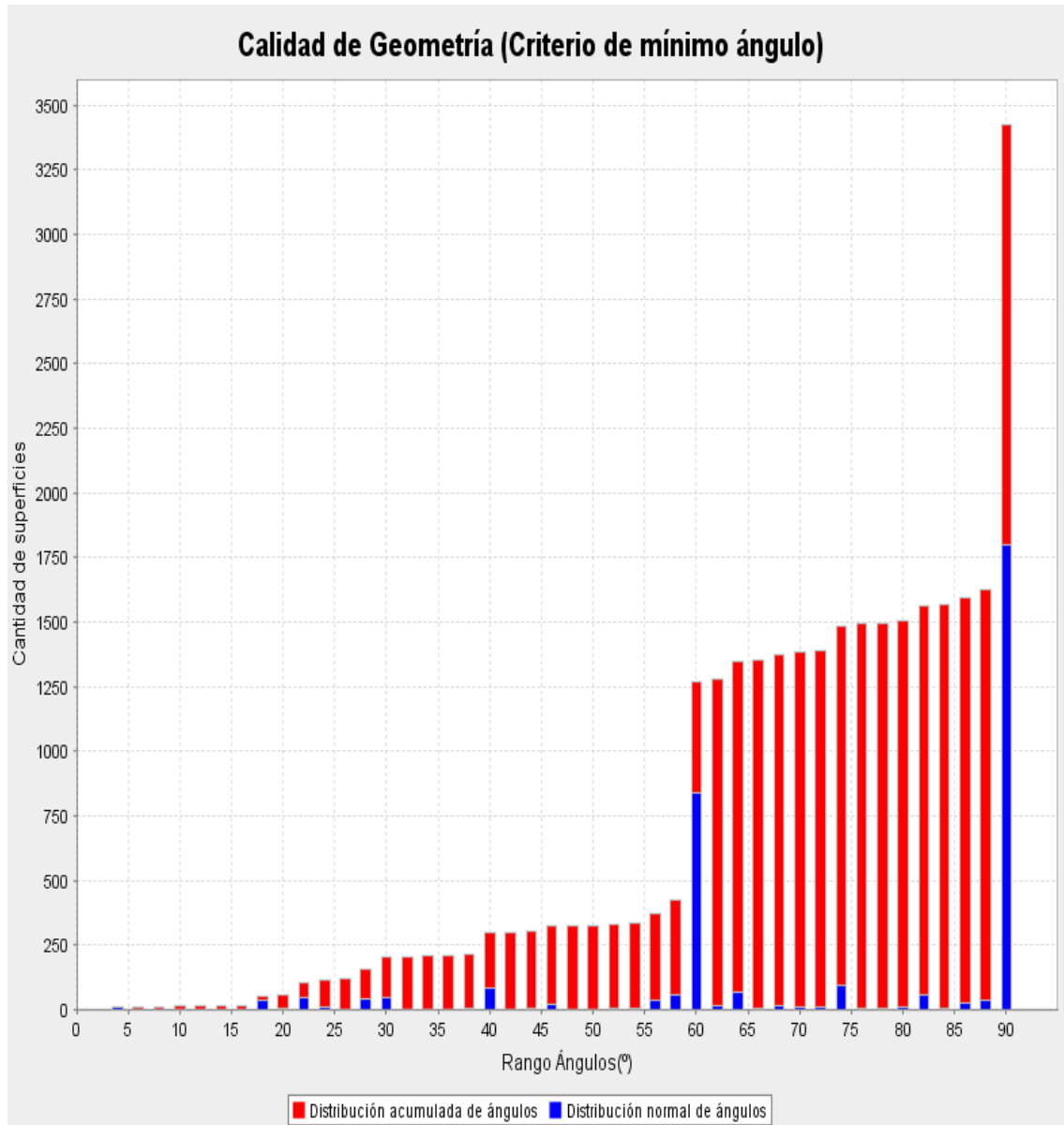


Figura III-83. Distribución de elementos clasificados por su mínimo ángulo

Se puede establecer un ángulo mínimo por debajo del cual sería necesario refinar la geometría generada, o se podría realizar una edición manual de dicha geometría, permitiendo la modificación oportuna de ciertos puntos o superficies por parte un usuario.

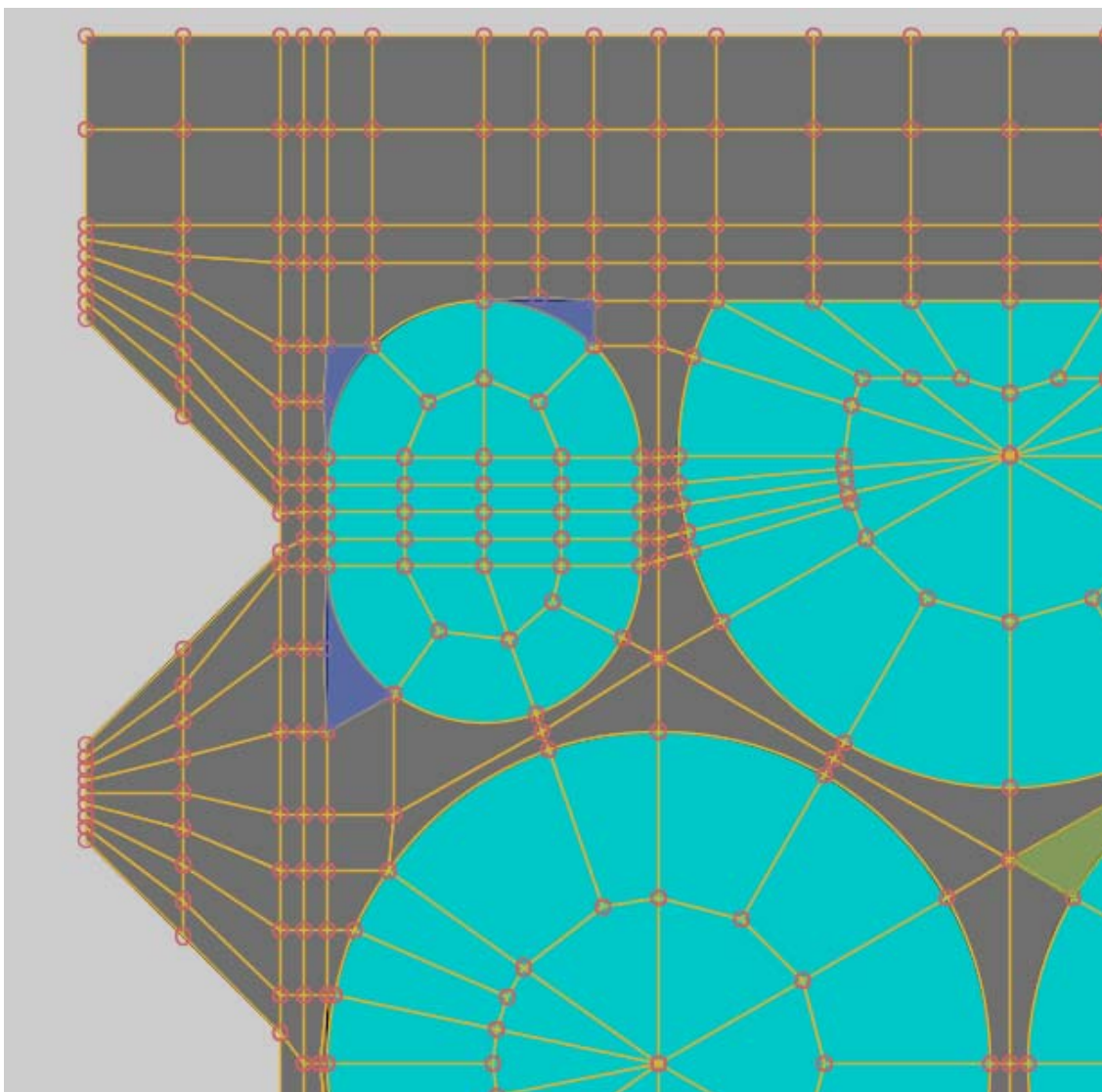


Figura III-84. Visualización de superficies con pequeños ángulos mínimos.

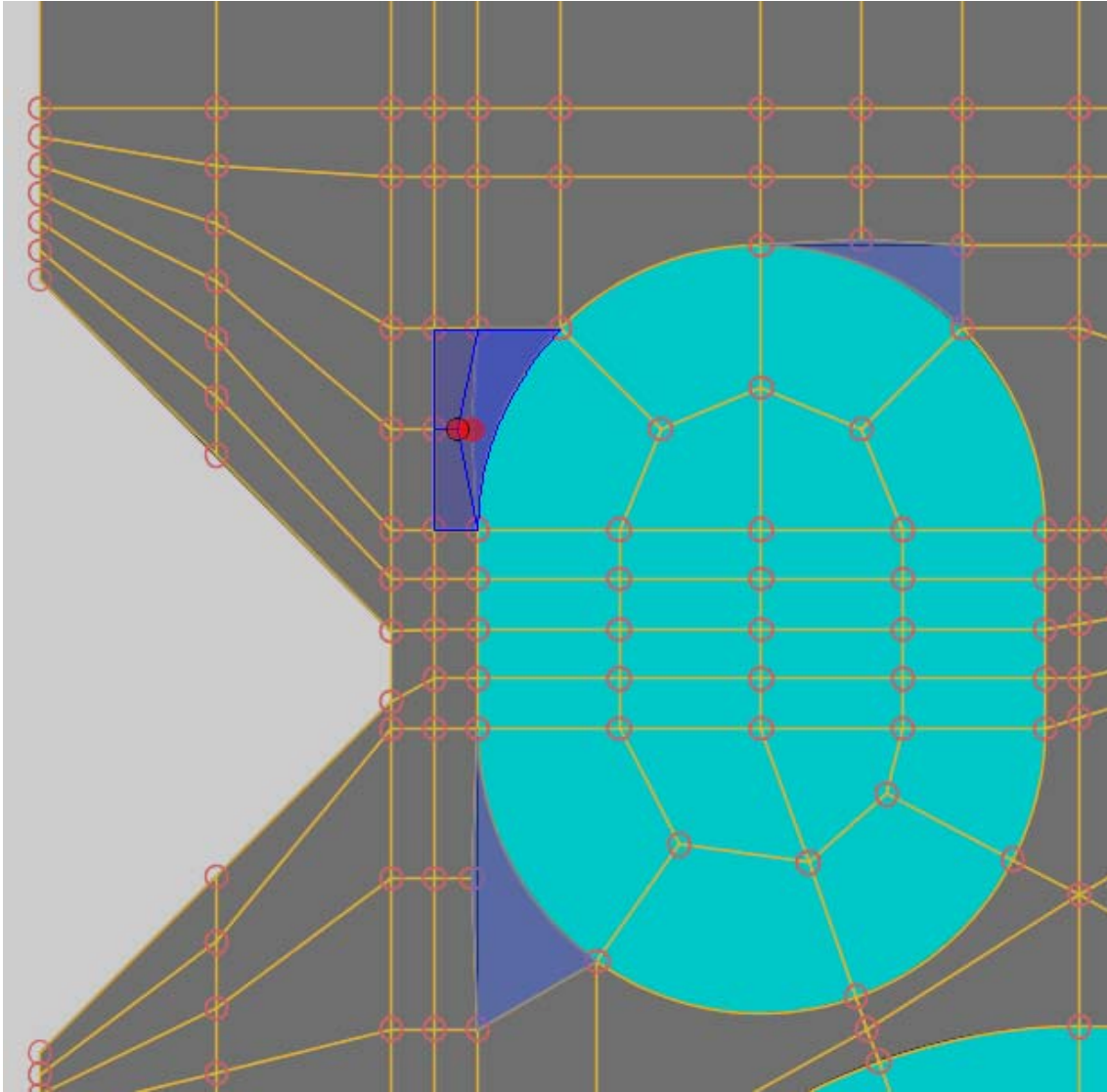


Figura III-85. Edición de puntos o superficies para corrección de ángulos mínimos

Las correcciones a realizar deberán hacerse con la precaución de mantener ciertas condiciones de contorno (puntos de los bordes externos del cajón o del perímetro de las celdas circulares, ubicación de gargantas, etc.).

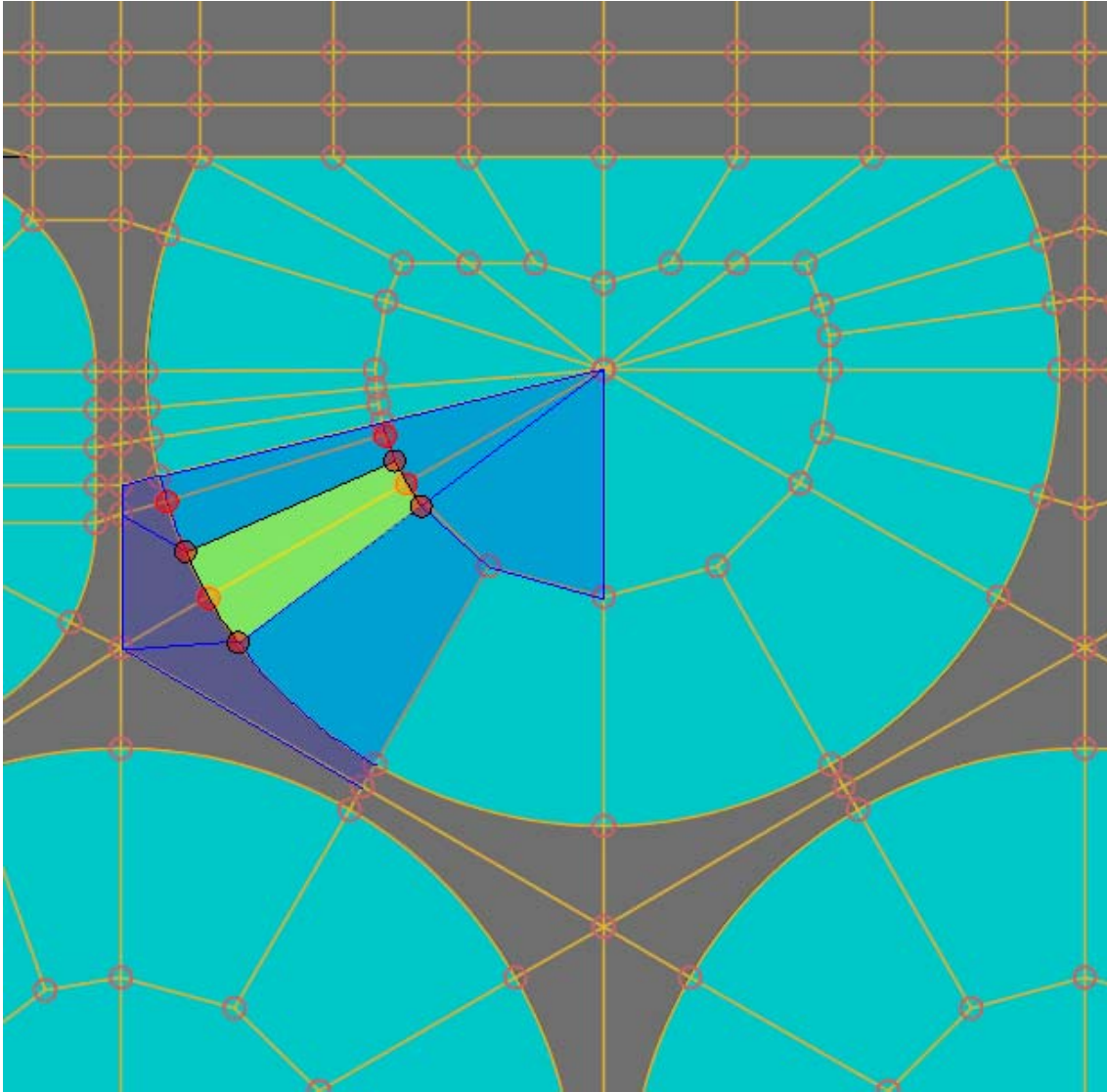


Figura III-86. Corrección de superficie, con restricción de movimiento circunferencial

III.3.- Las acciones en el cajón

III.3.1.- Base de diseño

La verificación de que un muelle de atraque o de un dique de abrigo en su conjunto, sus tramos y elementos alcanzan los niveles de fiabilidad, aptitud al servicio y operatividad exigidos se realiza mediante el procedimiento general de cálculo del Método de los Estados Límite. Se distinguen los estados límite últimos ELU, en relación al comportamiento resistente, los estados límite de servicio ELS, de aptitud al servicio y los estados límite operativos ELO, de uso y explotación, a los que se considera que pueda estar sometida la obra durante la fase de proyecto analizada.

La ROM_2.0-11 [11] define los siguientes modos de fallo asociados a cada uno de los estados límite definidos.

Estados límite últimos (ELU)

Los modos de fallo son aquéllos que producen la ruina de la obra o de una parte de la misma por rotura o colapso estructural. Pueden clasificarse en los siguientes grupos:

- EQU: Pérdida de equilibrio estático. La obra o una parte de la misma pierde sus condiciones de estabilidad. Es el caso del vuelco de un muelle.
- STR: Estructurales o de inestabilidad interna. La obra o parte de la misma alcanza su capacidad resistente o se produce en la misma una deformación local o global o cambios en la geometría de la obra excesivos que puede llevar al agotamiento estructural. Un ejemplo de estos modos de fallo puede ser el agotamiento de la pared exterior de un cajón ante esfuerzos de tracción y flexión.
- GEO: Geotécnicos o de inestabilidad externa. Son aquellos fallos debidos a la rotura o deformación del terreno sobre el que se asienta la obra, excesiva para la seguridad estructural. Ejemplos de estos modos de fallo puede ser el hundimiento o el deslizamiento profundo de una obra de gravedad.
- UPL: Fallos debidos al exceso de la presión de agua. Son aquellos fallos producidos por levantamientos o hundimientos provocados por excesos de presión hidrostática o hidrodinámica en los que la resistencia del terreno y de la estructura desempeñan un papel secundario. Un ejemplo de estos modos de fallo puede ser el hundimiento de obras de atraque o amarre flotantes en fase de servicio o de construcción por inundación.
- HYD: De inestabilidad hidráulica. Fallos ocasionados por la existencia de gradientes hidráulicos en el terreno o en rellenos, por las fuerzas de arrastre generadas por éstos, así como por los movimientos de las aguas libres exteriores. Ejemplos de estos modos de fallo pueden ser las erosiones externas o socavaciones en el pie del intradós o en los taludes de protección de las obras de atraque debidas a la acción de corrientes naturales o generadas por las hélices y otros equipos de propulsión y maniobra de los buques, o el oleaje.

Estados Límite de Servicio (ELS)

Los modos de fallo asociados son todos aquellos estados que producen la pérdida total o parcial de funcionalidad de la obra o de parte de ella, de forma reversible o irreversible, debido a un fallo estructural, de tipo formal o estético, ambiental o por condicionante legal, reduciendo o condicionando el uso y explotación de la obra. En obras de atraque y amarre, los principales modos de fallo asociados que deben considerarse adscritos a los siguientes grupos:

- DUR: Durabilidad. Son aquellos fallos debidos a la pérdida de durabilidad de la obra, bien por pérdida de las características de los materiales frente a los agentes del medio físico, del terreno, de la construcción o de uso y explotación, bien por pérdida de funcionalidad de la obra. Un ejemplo de estos modos de fallo es la fisuración del hormigón o la corrosión de una tablestaca metálica.
- REP: Reparabilidad. Son aquellos fallos asociados al máximo nivel de daños de la obra que permiten la utilización de procedimientos previstos y

planificados de mantenimiento y reparación. Un ejemplo de estos fallos es la iniciación de averías en el manto de protección en un talud de escollera en un muelle de pilotes.

- VIB: Vibraciones excesivas. Son aquellos fallos que producen la pérdida de funcionalidad de la obra por causa de la amplitud o frecuencia de vibraciones sobre la misma. Un ejemplo de estos modos pueden ser los daños en elementos e instalaciones por vibraciones inducidas directamente por la acción del oleaje, por ondas largas o por las acciones de amarre.
- DEX: Deformaciones excesivas. Son aquellos modos de fallo que producen la pérdida o limitan la normal explotación de la obra de atraque por causa de deformaciones, desplazamientos o asientos excesivos debido a causas estructurales, geotécnicas o hidráulicas. Un ejemplo de estos modos de fallo puede ser una flecha superior a las tolerancias para el uso y explotación de los equipos de manipulación.
- EST: Estéticos. Son aquellos modos de fallo que afectan al cumplimiento de aspectos formales requeridos para la obra. Como ejemplos de estos modos de fallo pueden citarse las pérdidas de alineación del cantil o de verticalidad de un muelle.

Estados Límite de Parada Operativa (ELO)

Los modos de parada asociados son aquellos estados en los que se reduce o suspende temporalmente la explotación de la instalación de atraque por causas ajenas a la obra o sus instalaciones, sin que haya daño estructural ni formal en ellas o en algunos de sus elementos. Generalmente estos estados están asociados a la excedencia de factores climáticos o a condicionantes legales o de seguridad y, por tanto, no serán considerados dentro de las situaciones de cálculo.

Método de cálculo

Los modos de fallo o parada se definen, ordenan, correlacionan y secuencian por medio del establecimiento de los *árboles de fallo o parada*, correspondientes a cada una de las series de estados límite posibles.

Dada la complejidad de los árboles de fallo, es admisible del lado de la seguridad simplificar el proceso, considerando únicamente un *diagrama de fallo o parada*, es decir, un conjunto completo de modos de fallo que no se producen simultáneamente, en el que todos los mecanismos de fallo o parada quedan descritos, organizándolos en serie y considerando que son mutuamente excluyentes.

Acciones a considerar

Cargas permanentes (G)

- Peso propio del cajón P_0 . Se calculará a partir de las dimensiones geométricas definidas en planos para la losa y el fuste. Como peso específico del hormigón armado se adopta un valor entre 24 y 25 kN/m³. En servicio se podrá considerar adicionalmente el peso propio de la superestructura.
- Empuje del mar P_a . Empuje ascensional del agua equivalente al peso del agua desplazada por el cajón, considerado como presión uniforme sobre la base de la estructura, tomando un peso específico del agua de 10.1 kN/m³.


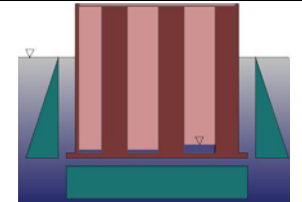
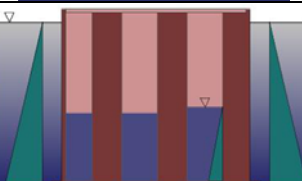
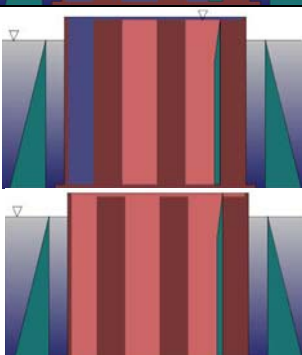
- Carga muerta: peso de la superestructura y tierras colocadas sobre el cajón, salvo en servicio que se considera agrupada en P_0 .
- Empuje hidrostático del agua E_a , lineal con la profundidad. Adicionalmente se considerará el peso del lastre de agua en las celdas del cajón, P_l , y el peso del agua sobre los voladizos de la losa, P_{lz} .

$$e_a = \gamma_a \cdot z \quad \text{(III-57)}$$

- Peso de las tierras P_t que gravitan en el trasdós sobre el voladizo de la losa, estando sumergidas hasta el nivel freático.
- Acciones verticales del material ensilado P_r , debidas a las tierras del relleno de las celdas, en las que se produce el denominado efecto silo [125], por el cuál se reducen las compresiones verticales como consecuencia del rozamiento con las paredes.

III.3.2.- Casos de carga considerados en el estudio de un cajón

A modo de resumen, la siguiente tabla recoge los casos que generalmente se consideran en el estudio del cajón.

<p>1. Construcción</p>	
<p>2. Botadura del cajón y transporte</p>	
<p>3. Hundimiento. Anegación de celdas Estricto/Completo Bajamar/Pleamar</p>	
<p>4. Relleno de celdas Completo/Parcial (desde Lado Tierra/Lado Mar/Lateral/Libre) Bajamar/Pleamar</p>	

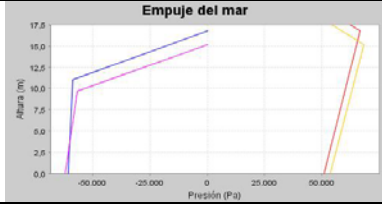
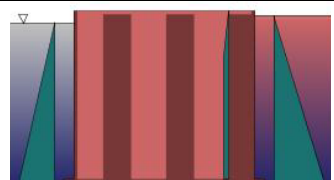

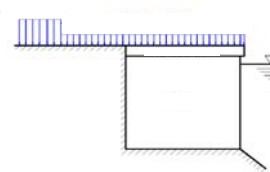
<p>5. Empuje del oleaje antes del trasdosado Impacto/Retirada del oleaje en Intradós/Trasdós Bajamar/Pleamar</p>	
<p>6. Relleno del trasdós Bajamar/Pleamar Con oleaje en intradós Relleno parcial</p>	
<p>7. Puesta en Servicio del Cajón Bajamar/Pleamar Relleno completo</p>	
<p>8. Sobrecargas de explotación Cargas de operación Tren de cargas restringido Tren de cargas no restringido Tiro de Bolardo/Defensas</p>	

Tabla III-5. Casos de carga más comunes a estudiar

En el estudio se sigue la secuencia: construcción-transporte-hundimiento-relleno y puesta en servicio. Se incluyen las posibles variaciones a tener en cuenta en cada hipótesis y las gráficas que las ilustran (corresponden a pantallas del asistente del proyecto en preprocesado).

Las condiciones de contorno sobre la losa en hipótesis de servicio se deberían introducir aplicando la teoría de contacto y modelando la interacción en la dirección normal a la losa mediante muelles cuyas rigideces dependen del coeficiente de balasto. Pero el coste computacional sería muy elevado debido a la no linealidad que implica el desconocer a priori las zonas de adhesión y de deslizamiento, problema este que ha de dilucidarse mediante un trabajoso procedimiento iterativo.

Por ello se realizan simplificaciones notables en la forma de sustentación del cajón, de modo que según la condición de cargas se apoya una zona determinada de la losa, y la otra se libera, aceptando las inconsistencias producidas en la frontera de ambas zonas.

Si bien se puede hacer un cálculo lineal sin iteraciones, los resultados del método de elementos finitos en la losa no permitirán realizar el armado de la misma, y se deberá recurrir a métodos tradicionales para ello, como se verá más adelante.

A continuación se detalla, para cada una de las hipótesis de cálculo a resolver, las cargas y su determinación.

III.3.2.1.- La hipótesis de transporte

Estimación de cargas

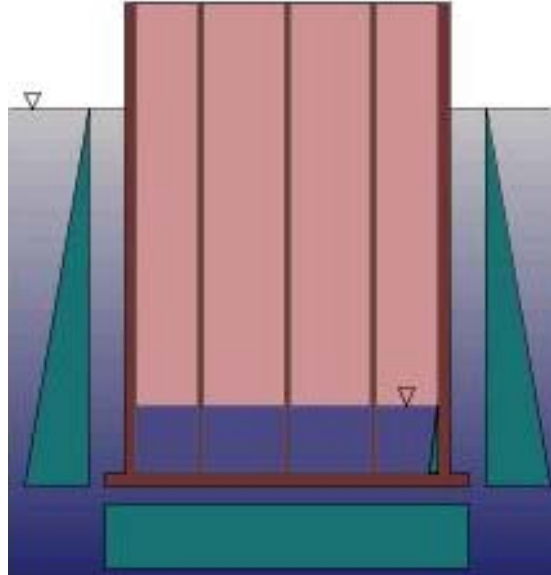


Figura III-87. Calado del cajón y cargas durante el transporte

Una vez que el cajón se ha construido, se bota y, tal y como se muestra en la **Figura III-87**, se considera una situación de estabilidad en flotación. Las ecuaciones del apartado III.1.13.- permiten determinar el calado H_{Calado}^{Fuste} y la altura de lastre uniforme H_{Agua} y adicional requerido $H_{AguaAdicional}$. Con estos datos y despreciando la carga de tiro del remolcador se determina la presión hidrostática sobre las paredes exteriores según la altura de calado y sobre las celdas interiores según la altura de lastre. Así mismo, el empuje es calculado según la ecuación (III-43).

Condiciones de contorno

En el caso de la hipótesis de transporte el cajón no está apoyado en ningún punto, pero es necesario establecer las condiciones de contorno derivadas de su situación de equilibrio en flotación. Para ello se supondrá que el cajón se apoya en un medio elástico, cuya rigidez es preciso determinar.

En esta situación de equilibrio:

$$V_{Hor}^{Real} \cdot \gamma_{Hor} = V_{Sumergido}^{Real} \cdot \gamma_{agua} \quad \text{(III-58)}$$

Si se ejerce una determinada fuerza ficticia sobre el cajón, éste se hunde a partir de la posición de equilibrio una cantidad Δx :

$$V_{Hor}^{Real} \cdot \gamma_{Hor} + F = V_{Sumergido}^{Real} \cdot \gamma_{agua} + \Delta x \cdot A_{Fuste}^{Bruta} \cdot \gamma_{agua} \quad \text{(III-59)}$$

por tanto, la fuerza que hay que realizar, si se cumple la ecuación de equilibrio, es justamente igual al empuje del fluido adicional desalojado:

$$F = \Delta x \cdot A_{Fuste}^{Bruta} \cdot \gamma_{agua} \quad (III-60)$$

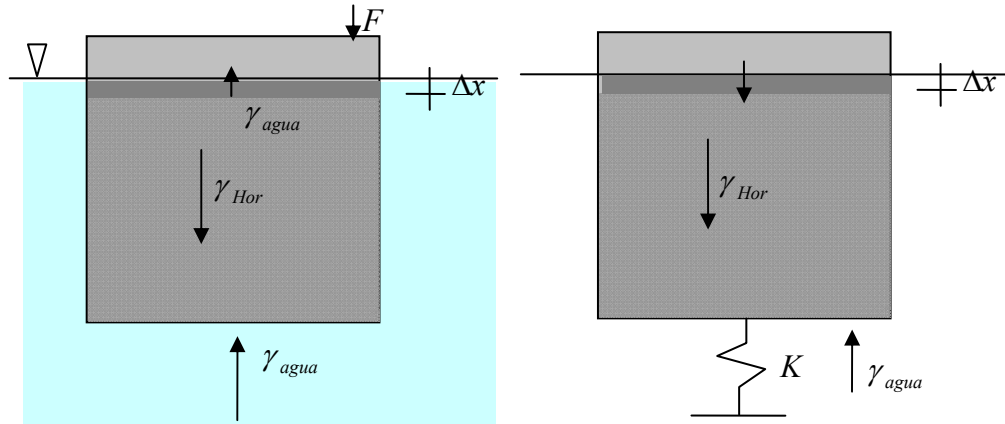


Figura III-88. Condiciones de equilibrio del cajón en flotación

Sustituyendo el sistema original por uno en el que el mar es reemplazado por un muelle:

$$V_{Hor}^{Real} \cdot \gamma_{Hor} + \Delta x \cdot A_{Fuste}^{Bruta} \cdot \gamma_{agua} = V_{Sumergido}^{Real} \cdot \gamma_{agua} + K \cdot \Delta x \quad (III-61)$$

En estas condiciones ideales de equilibrio, la constante de rigidez del muelle será

$$K = \frac{V_{Sumergido}^{Real} \cdot \gamma_{agua} - V_{Hor}^{Real} \cdot \gamma_{Hor}}{\Delta x} + A_{Fuste}^{Real} \cdot \gamma_{agua} \Rightarrow K = A_{Fuste}^{Bruta} \cdot \gamma_{agua} \quad (III-62)$$

No obstante, existe un pequeño error derivado de la simulación del cuerpo mediante una malla de elementos finitos al no reproduce con exactitud la geometría real.

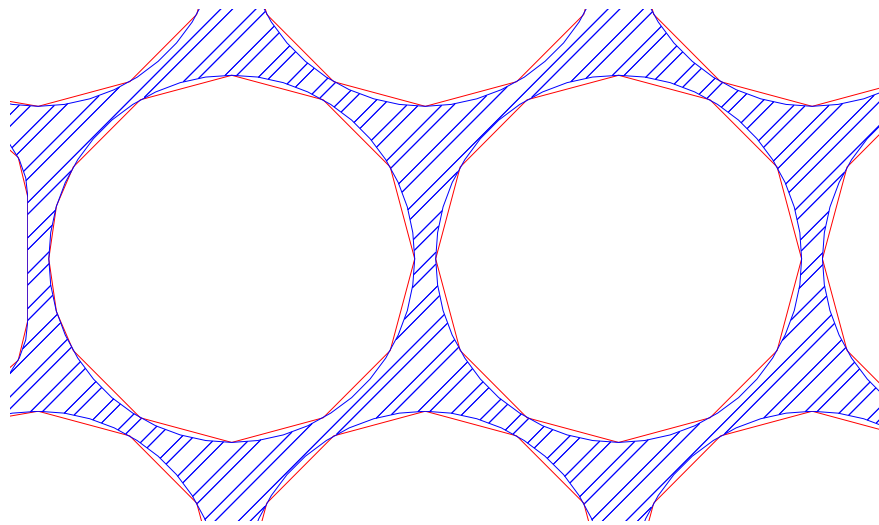


Figura III-89. Diferencia entre área modelada y área real

En la modelización, los arcos del interior de las celdas se toman como rectas y la malla realizada (en rojo) tiene más área que la sección real (en azul): por tanto tiene más volumen y más peso que el real.

En la ecuación de equilibrio anterior, ahora el volumen de hormigón se determina a partir de la malla realizada, mientras que el volumen sumergido ha sido calculado previamente a partir del volumen real:

$$V_{Hor}^{Mallado} = V_{Hor}^{Real} + \Delta V_{Hor}$$

$$\Delta V_{Hor} = \Delta A_{Fuste}^{Neta} \cdot H_{Fuste} = (A_{Fuste}^{Mallado} - A_{Fuste}^{Real}) \cdot H_{Fuste} \quad \text{(III-63)}$$

$$V_{Hor}^{Mallado} \cdot \gamma_{Hor} + \Delta x \cdot A_{Fuste}^{Bruta} \cdot \gamma_{agua} = V_{Sumergido}^{Real} \cdot \gamma_{agua} + K \cdot \Delta x$$

Por tanto, por el error inevitable en la discretización (que será menor cuanto más fina sea ésta), aparece una fuerza adicional $\Delta V_{Hor} \cdot \gamma_{Hor}$.

Se puede realizar una modificación del peso específico del hormigón para que esta fuerza se cancele, y el estado tensional en el cajón se deba a las cargas reales que se producen en el equilibrio en flotación:

$$V_{Hor}^{Mallado} \cdot \gamma_{Hor}^* = V_{Hor}^{Real} \cdot \gamma_{Hor}$$

$$\gamma_{Hor}^* = \frac{V_{Hor}^{Real}}{V_{Hor}^{Mallado}} \cdot \gamma_{Hor} = \frac{V_{Hor}^{Mallado} - \Delta V_{Hor}}{V_{Hor}^{Mallado}} \cdot \gamma_{Hor} = \gamma_{Hor} \cdot \left(1 - \frac{\Delta V_{Hor}}{V_{Hor}^{Mallado}} \right) = \quad \text{(III-64)}$$

$$= \gamma_{Hor} \cdot \left(1 - \frac{H_{Fuste} \cdot (A_{Fuste}^{Mallado} - A_{Fuste}^{Real})}{H_{Losa} \cdot A_{Losa} + H_{Fuste} \cdot A_{Fuste}^{Mallado}} \right)$$

III.3.2.2.- Hundimiento del cajón con agua. Fondeo estricto en pleamar y bajamar

Una vez situado el cajón en la posición definitiva se procede al hundimiento del mismo por anegación con agua de mar de las celdas interiores. Este proceso se realiza comunicando interiormente entre sí las distintas celdas de una misma hilera longitudinal de forma que en el interior del cajón se reproduce el fenómeno de los vasos comunicantes.

La cota de hundimiento la marca la altura de calado del puerto según la escollera realizada y se expresa en función del Nivel Medio del mar, en condiciones normales de operación. Sobre esta altura hay una cota de pleamar (positiva) $H_{pleamar}$ y una de bajamar (negativa) $H_{bajamar}$ sobre dicho Nivel Medio.

Aunque el fondeo se realiza anegando completamente las celdas con agua, a efectos de estimar la situación más crítica en cuanto a cargas en el interior del cajón, por la mayor diferencia de presiones entre el exterior y el interior, se contemplará únicamente la

situación de fondeo estricto, aquella que emplea la cantidad mínima imprescindible para que el cajón se hunda.

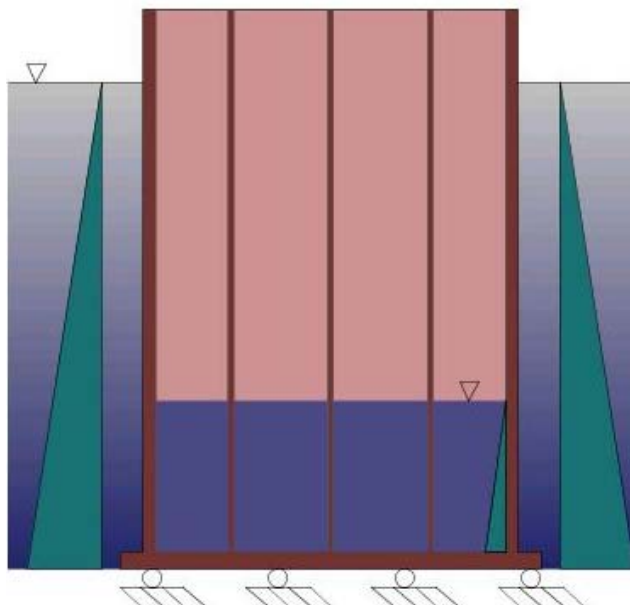


Figura III-90. Fondeo del cajón y cargas según pleamar o bajamar

La determinación de la altura de lastre mínimo que hunde el cajón se obtiene de la ecuación (III-45) imponiendo que la altura de francobordo sea nula.

III.3.2.3.- Relleno de las celdas

Una vez posicionado el cajón en el lugar adecuado se procede a rellenar las celdas con material granular, completamente hasta la coronación del fuste.

La acción que el relleno ejerce sobre los tabiques exteriores e interiores se supone dadas las dimensiones de las celdas bajo el efecto de ensilado.

La presión que ejerce el relleno de las celdas sobre las paredes del cajón se determina siguiendo las recomendaciones de la R.O.M. 0.2.90 del Ministerio de Obras Públicas, Transportes y Medio Ambiente, Abril 1990, sobre las Acciones en el Diseño de Obras Marítimas y Portuarias.

Las mencionadas recomendaciones en sus capítulos sobre diseño de cajones multicelulares proponen, de forma simplificada y del lado de la seguridad, considerar una variación lineal de las presiones horizontales y verticales en una altura z_0 permaneciendo constante en el resto de la altura (páginas 125-126), de modo que las leyes en función de la cota z , tomada en la cota de superficie libre del agua o del relleno y hacia abajo, son:

$$\begin{cases} P(z)_{h,pv} = \lambda \cdot (q + \gamma \cdot z) \\ P(z)_{v,pv} = \lambda \cdot (q + \gamma \cdot z) \cdot \tan \delta \\ P(z)_{v,ph} = q + \gamma \cdot z \end{cases} \quad 0 \leq z \leq z_0 \quad \text{(III-65)}$$

$$\begin{cases} P(z)_{h,pv} = \lambda \cdot (q + \gamma \cdot z_0) \\ P(z)_{v,pv} = \lambda \cdot (q + \gamma \cdot z_0) \cdot \tan \delta & z \geq z_0 \\ P(z)_{v,ph} = q + \gamma \cdot z_0 \end{cases}$$

donde λ es el coeficiente de empuje, γ el peso específico aparente del relleno y z_0 se define como una profundidad crítica que viene dada por:

$$z_0 = \frac{A}{u \cdot \lambda \cdot \tan \delta} \quad (\text{III-66})$$

siendo:

- A área de la celda en contacto con el relleno (**Tabla III-2**)
- u perímetro de la celda en contacto con el relleno
- λ coeficiente de empuje que toma un valor entre 0.5-1
- δ ángulo de rozamiento suelo estructura. Este ángulo se puede tomar igual a $0.75 \cdot \Phi$ siendo Φ el ángulo de rozamiento interno del relleno

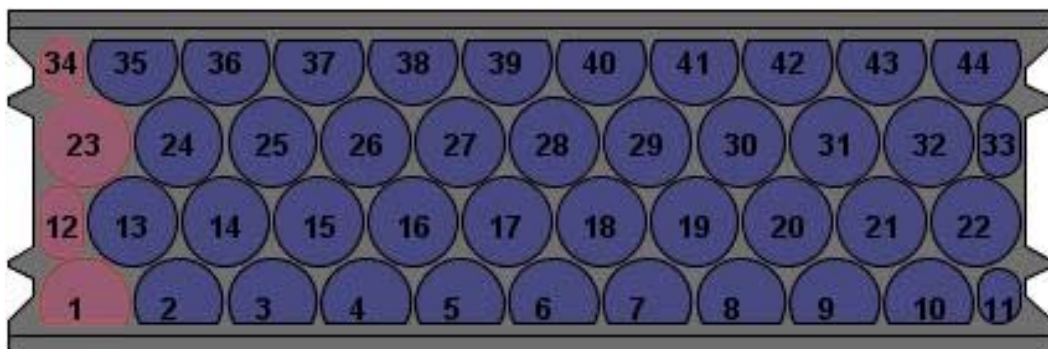
Para λ se toma en este estudio un valor de 0.5 tanto para determinar la profundidad crítica z_0 como para determinar el empuje del relleno granular durante el llenado.

Los subíndices pv y ph hacen referencia a la actuación de la presión horizontal (h) o vertical (v) sobre un plano vertical y sobre un plano horizontal, respectivamente.

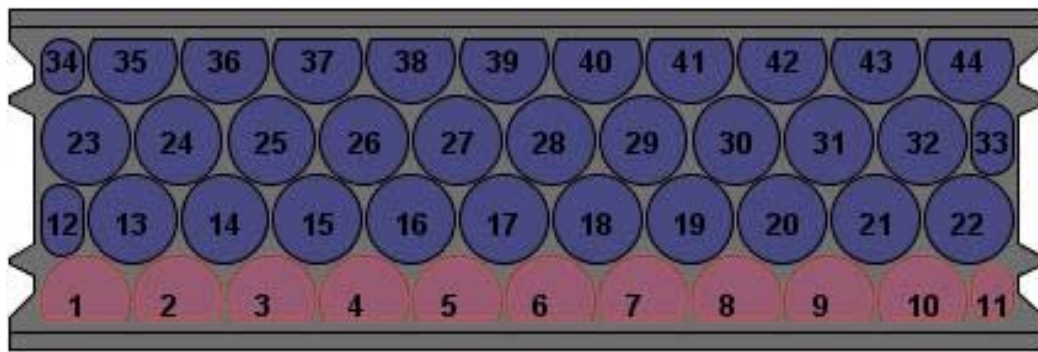
A estas presiones debidas al relleno se añadirá la hidrostática del agua.

Los valores anteriores valen tanto para la situación de pleamar como para la de bajamar.

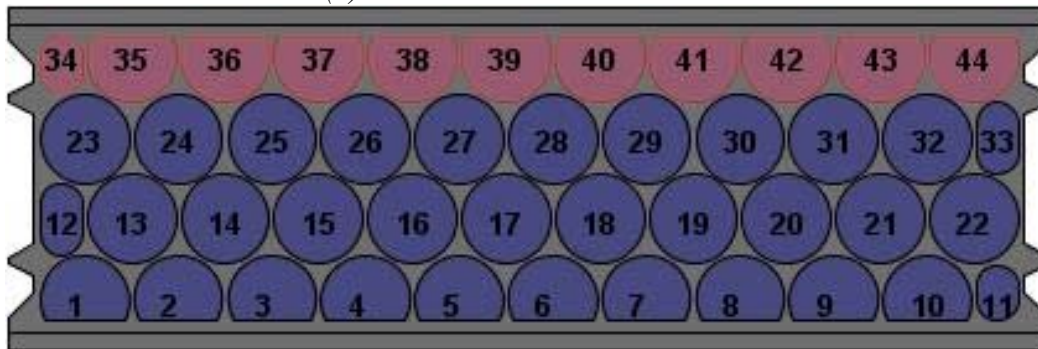
Se considerará inicialmente el estudio de la carga de relleno parcial del cajón, suponiendo que se comienza a rellenar la primera línea de celdas desde el lado Lateral, desde el Lado Tierra, desde el Lado Mar, o bien de forma libre desde el centro del cajón y finalmente se considerará el caso de relleno completo.



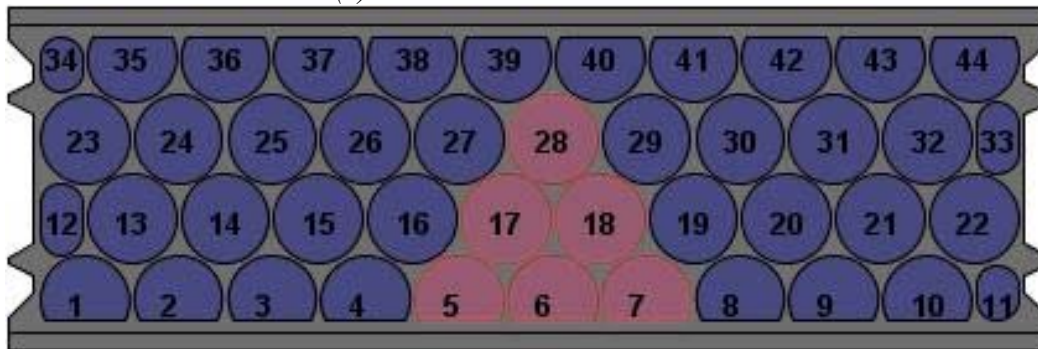
(a) Relleno desde el Lateral



(b) Relleno desde el Lado Tierra



(c) Relleno desde el Lado Mar



(d) Relleno parcial libre

Figura III-91. Posibles situaciones de relleno parcial de celdas

El relleno se realiza en *condiciones críticas* de operación, por lo que las cotas de pleamar se añade 0.5 m y a la de bajamar se resta 0.5 m.

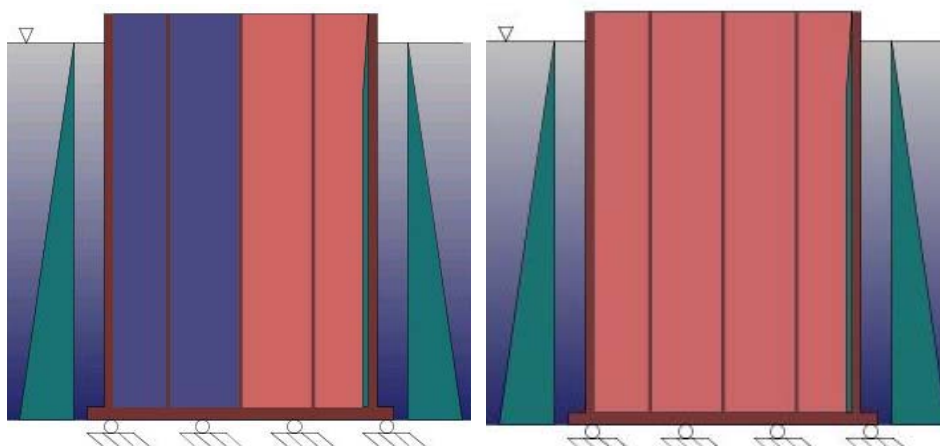


Figura III-92. Situaciones de relleno parcial o completo en bajamar o pleamar

III.3.2.4.- Empuje del oleaje en pleamar y bajamar

En el caso de que se considere la acción del oleaje sobre la estructura del cajón serán datos aportados las presiones que se ejercen en bajamar o en pleamar, conforme al modelo de oleaje empleado, **Figura II-8**.

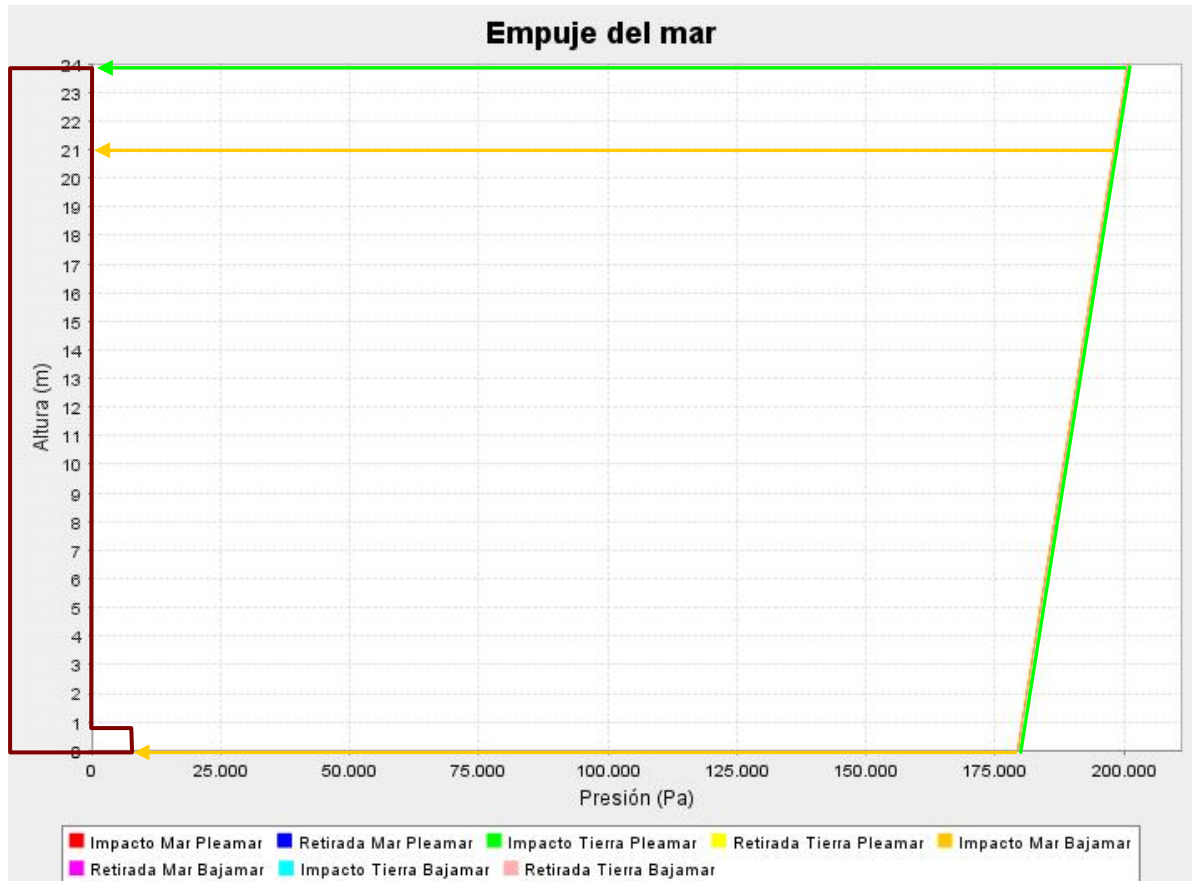


Figura III-93. Ejemplo de curvas de presión de oleaje en el intradós

Adicionalmente, se considera que sobre el interior del cajón actúa la presión hidrostática del agua y las cargas del relleno sobre las celdas interiores, mientras que en el exterior, actúa la presión hidrostática sobre la pared del Lado Tierra y en los laterales. No se considera en este caso sobrecarga de uso.

III.3.2.5.- Relleno parcial del trasdós (muelles)

En el caso de cajones para la construcción de muelles, será un dato aportado el tipo de material de relleno a emplear. Según la ROM, el terreno quedará definido por su densidad γ^* , porcentaje de huecos n , ángulo de rozamiento interno Φ , coeficiente de empuje λ , pesos específicos saturados $\gamma_{sat} = g \cdot (\gamma^* + n \cdot \rho_{Agua})$ y aparente $\gamma = g \cdot (\gamma^* - (1 - n) \cdot \rho_{Agua})$, y empuje activo de Rankine $K_a = \tan^2(45^\circ - \Phi/2)$.

3. Los Cajones Multicelulares

En la situación de trasdosado los niveles de pleamar y bajamar son diferentes a un lado y otro del cajón, según sea el nivel freático en el trasdós.

La **Figura III-94** recoge las diferentes cargas en el intradós y en el trasdós debidas al empuje horizontal del agua y del terreno.

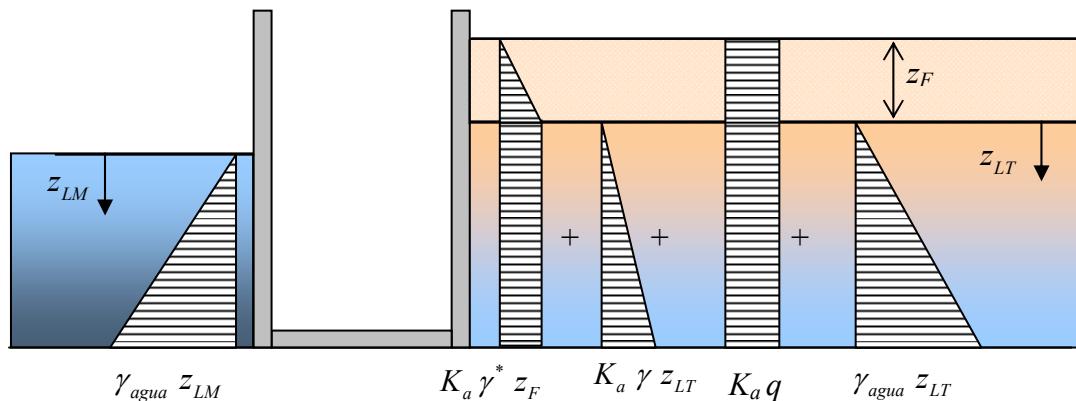


Figura III-94. Descomposición de las presiones sobre paredes exteriores en la primera fase de relleno del trasdós

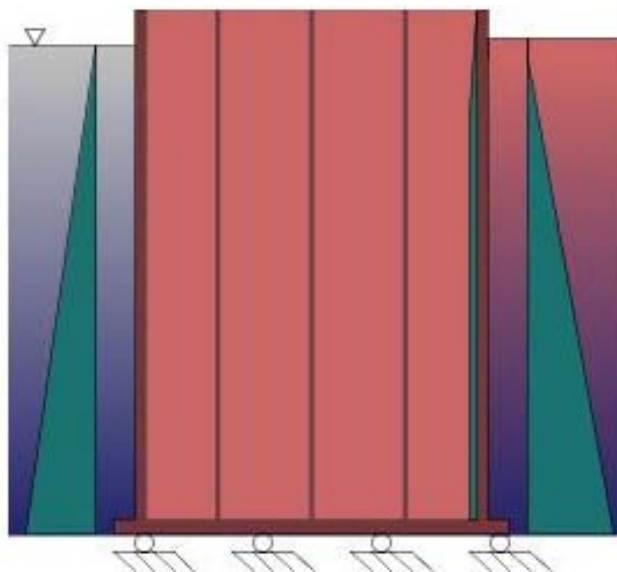


Figura III-95. Presiones sobre el cajón en la primera fase de relleno del trasdós en pleamar o bajamar

III.3.2.6.- Puesta en Servicio del Cajón

Para la puesta en servicio del cajón se procede a la colocación de la superestructura. Como ya se indicó en el apartado III.1.12.4.- suele ser una losa maciza de cierto canto y vuelo sobre el Lado Mar. En el caso de un muelle, la cuna del trasdós se rellena hasta la cota del muelle.

Para el análisis del cajón en servicio, se tiene en cuenta que sobre el terreno no actúa ninguna sobrecarga. Tampoco se considera la acción del oleaje en el intradós. Estas cargas se considerarán posteriormente como sobrecargas de uso. Sí que se tendrá en cuenta la sobrecarga que ejerce la superestructura sobre el cajón.

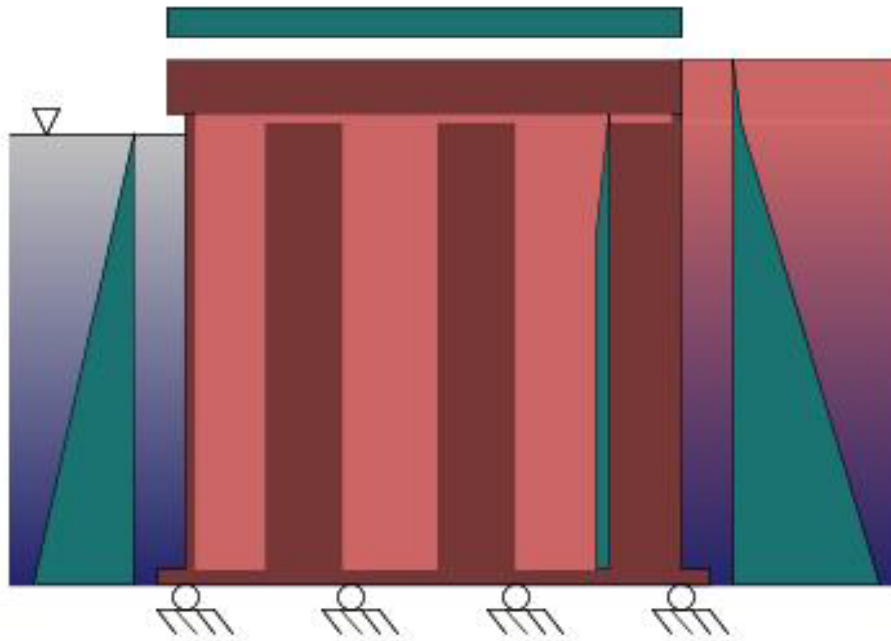


Figura III-96. Presiones sobre el cajón en la puesta en servicio, en pleamar o bajamar

III.3.2.7.- Sobrecargas de explotación

Las cargas de estacionamiento y almacenamiento se deben fundamentalmente al peso de los materiales, suministros o mercancías, almacenados bien en instalaciones específicas como silos o depósitos o bien en instalaciones auxiliares para su transporte y manipulación como contenedores, semirremolques,...

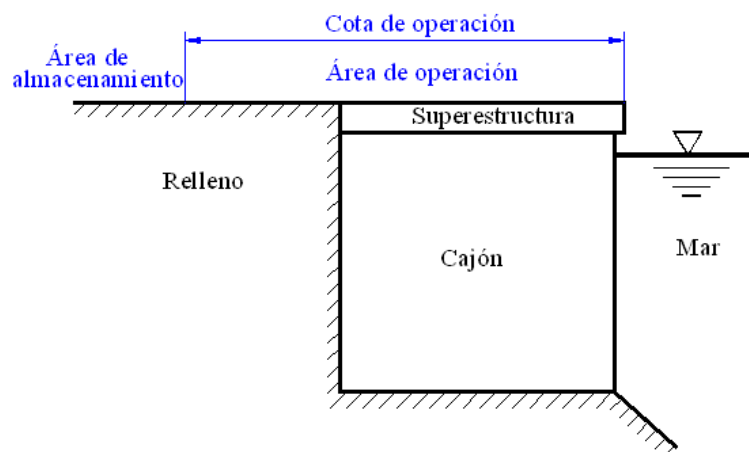


Figura III-97. Situación del área de operación en un muelle

Sobrecargas de operación

El *área de operación* indicado en la Figura III-97 es la zona destinada a la transferencia y manipulación de mercancías, sin acumulación duradera, mientras que el *área de almacenamiento* es la destinada a estancias prolongadas.

3. Los Cajones Multicelulares

La Tabla 3.4.2.3.1.3 de la ROM 0.2-90 indica unos valores mínimos para las cargas en las diferentes áreas en función del tipo de uso del muelle: general, comercial, deportivo, etc. Además, siguiendo la recomendación de la ROM, se debe considerar el caso de carga de operación mínima de 10000 N/m^2 sobre el cajón

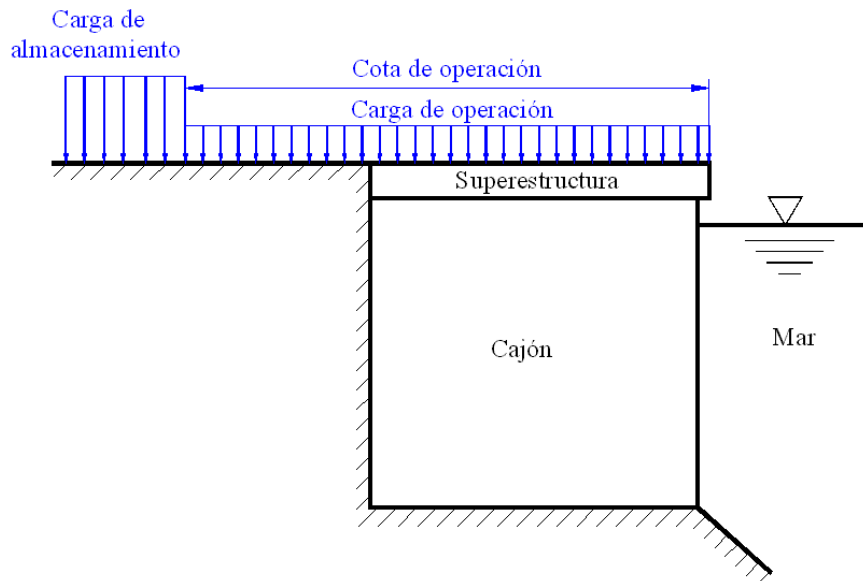


Figura III-98. Cargas de operación y mantenimiento

La sobrecarga en la estructura se traslada al cajón como carga distribuida en la superficie neta del fuste, mientras que la sobrecarga del terreno se traslada al cajón como carga horizontal $P_{ah} = K_a \cdot q$ sobre la pared del trasdós.

Sobrecargas de instalaciones

Las sobrecargas de instalaciones de manipulación de mercancías son cargas de naturaleza variable transmitidas a la estructura resistente por los sistemas y equipos de manipulación de mercancías.

Se distinguen los sistemas discontinuos o de *movilidad restringida* (equipos fijos, sobre carriles o sobre neumáticos) de los continuos, en los que alguno de los carriles recae sobre el cajón, a cierta distancia d del cantil, **Figura III-100**, o de *movilidad no restringida* (tuberías, cintas transportadoras,...), **Figura III-101**.



Figura III-99. Ejemplo de grúa de movilidad restringida

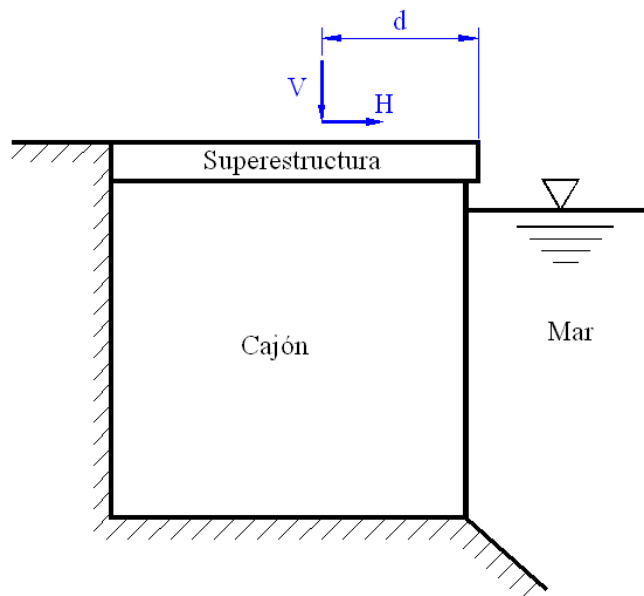


Figura III-100. Cargas debidas a trenes de movilidad restringida

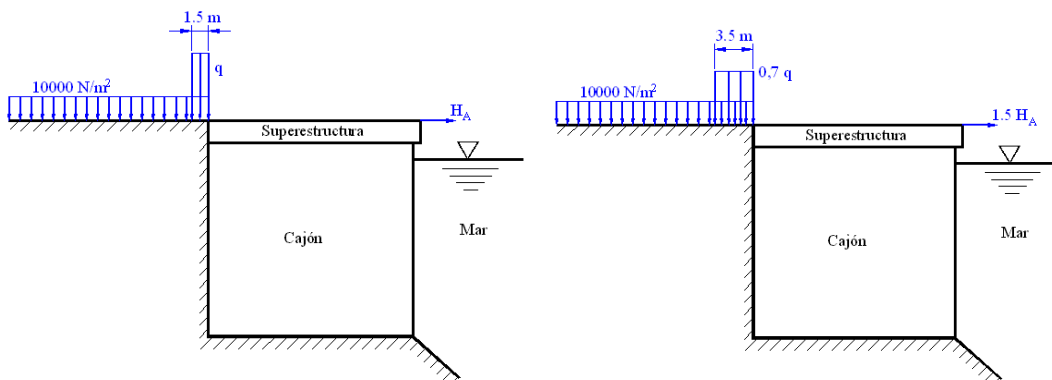


Figura III-101. Cargas debidas a trenes de movilidad no restringida, 2 casos

En estos casos, la carga vertical se aplica sobre el cajón como carga horizontal trapecial, siendo constante $P_{ah} = K_a \cdot q$ desde la cota del muelle hasta una profundidad $b = d_q \cdot \tan\left(45^\circ + \frac{\Phi}{2}\right)$ y disminuyendo a 0 a una cota $1.5 \cdot b$. Dado que la carga prescrita de 10000 N/m^2 sobre el terreno se trata en la carga de operación mínima, se descontará este valor de la carga vertical indicada.

Por otro lado, la carga horizontal se distribuye sobre la pared del Lado Mar.

Sobrecargas de atraque

En la viga cantil ubicada sobre el cajón se pueden disponer, separados determinada distancia entre sí, bolardos para el amarre de los buques que atraquen en el muelle y defensas para absorber el impacto que puedan generar en dicho atraque.

Las cargas que se generan en dichos elementos de atraque son incompatibles entre sí.

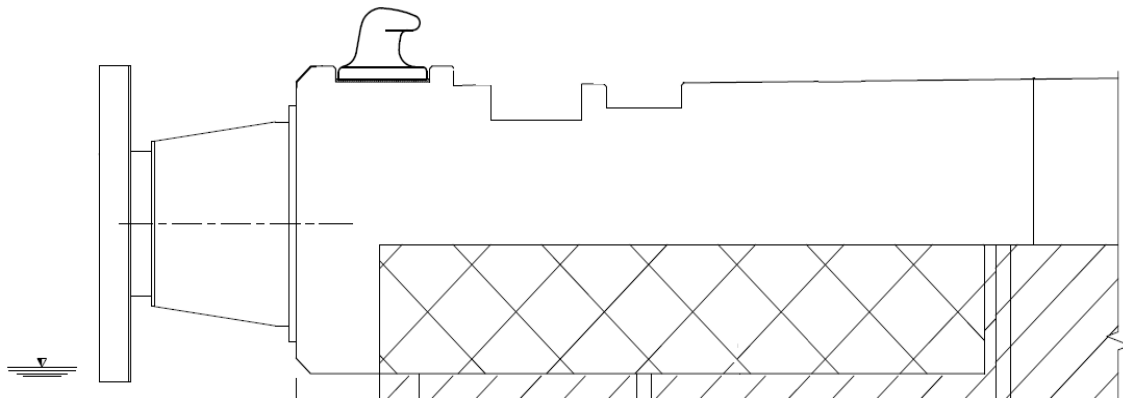


Figura III-102. Elementos de bolardo y defensas sobre viga cantil

Tiro de bolardo

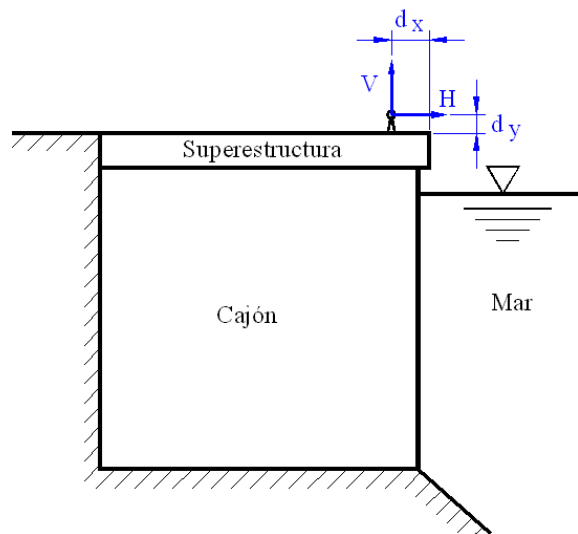


Figura III-103. Cargas del tiro de bolardo

En este caso, la carga horizontal se distribuye proporcionalmente sobre la pared del Lado Mar, mientras que la carga vertical al ser de tracción no se considera.

Defensas

Las defensas de última generación combinan una gran capacidad de absorción de energía con una baja fuerza de reacción.

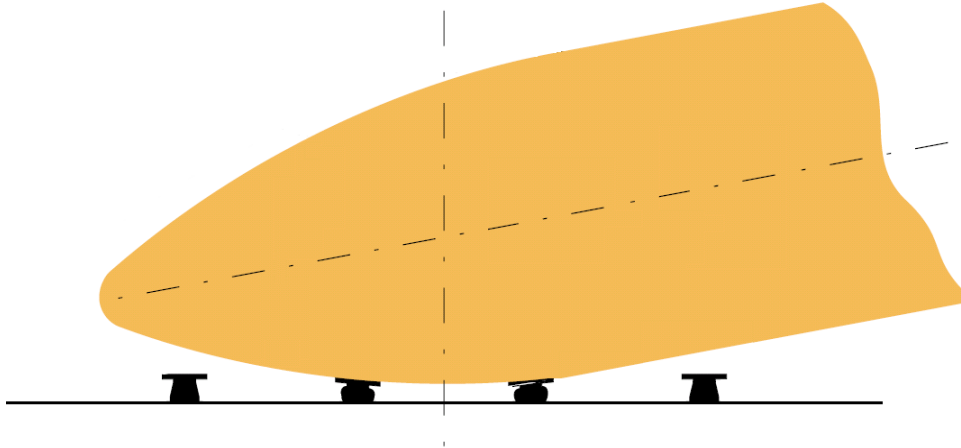


Figura III-104. Impacto en las defensas durante el atraque de un buque

El tipo de defensa y la separación entre defensas, así como las cargas máximas que reciben dichas defensas son datos aportados. Se pueden plantear varias situaciones de cálculo: impacto en dos defensas, en tres, etc.

Combinación de sobrecargas

Conforme a la Tabla 3.4.2.3.2.4 de la ROM 0.2-90 relativa a la *compatibilidad* de sobrecargas de estacionamiento con las sobrecargas de equipos e instalaciones de manipulación de mercancías, se indica en la **Tabla III-6** las posibles combinaciones entre las cargas con sus respectivos coeficientes de combinación.

Combinaciones CASO #1 CASO #2	Cargas de Operación		Movilidad Restringida	Movilidad No Restringida				Atraque
	Uso	Mínima	Tren A1	Tren B1(a)	Tren B1(b)	Tren B2(a)	Tren B2(b)	#1 Bolardo #2 Defensas
Combinación 1	1.0	-	0.7	-	-	-	-	0.7
Combinación 2	0.7	-	1.0	-	-	-	-	0.7
Combinación 3	0.7	-	0.7	-	-	-	-	1.0
Combinación 4	-	1.0	-	1.0	-	-	-	0.7
Combinación 5	-	0.7	-	0.7	-	-	-	1.0
Combinación 6	-	1.0	-	-	1.0	-	-	0.7
Combinación 7	-	0.7	-	-	0.7	-	-	1.0
Combinación 8	-	1.0	-	-	-	1.0	-	0.7
Combinación 9	-	0.7	-	-	-	0.7	-	1.0
Combinación 10	-	1.0	-	-	-	-	1.0	0.7
Combinación 11	-	0.7	-	-	-	-	0.7	1.0

Tabla III-6. Posibles combinaciones de sobrecargas y sus coeficientes de combinación

3. Los Cajones Multicelulares

Como la carga de bolardo no es compatible con la carga en la defensa, se plantean dos situaciones: cargas de operación y movilidad combinadas con el tiro de bolardo (caso #1) o con las defensas (caso #2). Esto se puede hacer extensivo a otras situaciones posibles.

Como se muestra en la tabla, resultan 11 combinaciones por caso, que deben ser superpuestas a la situación de servicio. Se superpondrán las combinaciones con la situación de servicio en pleamar y en bajamar. Esto supone resolver 22 hipótesis diferentes por caso, 44 en total, con un alto coste en tiempo de cálculo.

Sin embargo, si se resuelven por separado los casos básicos de sobrecargas de explotación sin ninguna otra carga más (peso propio, relleno de celdas, empujes de terreno, de oleaje, cargas hidrostáticas,...) y sin combinar entre sí, se dispondrán los resultados canónicos en desplazamientos y tensiones necesarios para añadir linealmente a las dos hipótesis de servicio, haciendo uso de la tabla de coeficientes de combinación y de los correspondientes coeficientes de seguridad.

III.3.2.8.- Coeficientes de seguridad

Las cargas anteriormente comentadas se han indicado en valores característicos, por lo que para su estudio en estados límite últimos es necesario definir los coeficientes de mayoración correspondientes, según la naturaleza permanente o variable de las mismas.

A falta de datos, se asume un nivel de control de ejecución *normal*. La tabla siguiente recoge los coeficientes empleados.

	Peso Propio		Agua		Sobrecarga		Relleno		Empuje Terreno		Empuje Oleaje	
	Fav	Des	Fav	Des	Fav	Des	Fav	Des	Fav	Des	Fav	Des
Transporte	1.0	1.5	1.0	1.5	0.0	1.6	-	-	-	-	-	-
Fondeo	1.0	1.5	1.0	1.5	0.0	1.6	-	-	-	-	-	-
Relleno Parcial	1.0	1.5	1.0	1.5	0.0	1.6	1.0	1.5	-	-	-	-
Relleno Completo	1.0	1.5	1.0	1.5	0.0	1.6	1.0	1.5	-	-	-	-
Empuje mar	1.0	1.5	1.0	1.5	0.0	1.6	1.0	1.5	-	-	1.0	1.6
Empuje terreno	1.0	1.5	1.0	1.5	0.0	1.6	1.0	1.5	1.0	1.6	1.0	1.6
Cajón en Servicio	1.0	1.5	1.0	1.5	0.0	1.6	1.0	1.5	1.0	1.6	1.0	1.6
Cargas de Explotación	-	-	-	-	0.0	1.6	-	-	1.0	1.6	-	-

Tabla III-7. Coeficientes de mayoración de cargas

III.3.2.9.- Acciones sobre la losa

Acciones estabilizadoras y volcadoras

Para la obtención de las acciones del terreno sobre la losa de cierre del cajón se determinan previamente el esfuerzo axial y el momento flector que actúan sobre la misma como consecuencia de las cargas verticales y horizontales derivadas de los estados de carga mencionados en los apartados anteriores, en función del carácter estabilizador o desestabilizador que puedan presentar.

Así, para cada una de las cargas consideradas se determina la fuerza generada, con su sentido y su punto de aplicación, y el momento flector respecto del extremo de la losa en Lado Mar. A continuación se agrupan por un las cargas estabilizadoras (peso superestructura, peso cajón vacío, peso relleno, peso mar sobre voladizo Lado Mar, peso terreno sobre voladizo Lado Tierra, sobrecargas verticales, empuje terreno vertical, empuje hidrostático vertical) y por otro las volcadoras (subpresión, tiro de bolardo, defensas, empuje terreno horizontal, empuje hidrostático horizontal, sobrecargas horizontales).

Con los valores obtenidos, se pueden calcular los coeficientes de seguridad al deslizamiento y al vuelco, conforme a las siguientes fórmulas:

$$U_{\text{Deslizamiento}} = \frac{\left(F_{\text{Verticales}}^{\text{Estabilizadoras}} - F_{\text{Verticales}}^{\text{Desestabilizadoras}} \right) \cdot \mu + F_{\text{Horizontales}}^{\text{Estabilizadoras}}}{F_{\text{Horizontales}}^{\text{Desestabilizadoras}}} \quad \text{(III-67)}$$

$$U_{\text{Vuelco}} = \frac{M^{\text{Estabilizadores}}}{M^{\text{Desestabilizadores}}}$$

Para un rozamiento habitual de 0.6, se calculan los coeficientes de seguridad para cada una de las hipótesis y se comprueba que el mínimo valor supera la unidad en ambos casos.

Resultantes sobre la losa

Como se ha indicado, para cada una de las cargas consideradas en cada una de las hipótesis de cálculo, se determina la fuerza generada, con su sentido y su punto de aplicación, y el momento flector respecto del extremo de la losa en Lado Mar. La suma de todas las cargas da lugar al axil y al momento flector que actúan sobre la losa, valores con los cuales se pueden obtener la ley trapezoidal de tensiones normales para cada hipótesis de cálculo.

Para el cálculo en la losa se tomarán los valores máximos de la tensión en cada uno de los extremos, aunque no se produzcan bajo la misma hipótesis de forma simultánea, **Figura III-105**.

A partir de estas tensiones máximas, la presión uniforme media que actúa sobre la losa se obtendrá al considerar el centro de la celda más próxima al borde del cajón más la presión hidrostática adicional diferencial entre el interior y el exterior del cajón, ambos mayorados con el coeficiente de seguridad correspondiente, menos el peso de la losa y el empuje vertical del terreno, **Figura III-106**.

$$P_{\max} = 1.5 \cdot \left(\sigma_A - \frac{\sigma_A - \sigma_B}{Manga} \cdot (V_{LM} + e_{LM} + R) \right) + 1.5 \cdot \rho_{\text{Agua}} \cdot g \cdot H_{\text{Losa}} - 1.0 \cdot \rho_{\text{Hormigón}} \cdot g \cdot H_{\text{Losa}} - 1.0 \cdot P_{v,ph} \quad (\text{III-68})$$

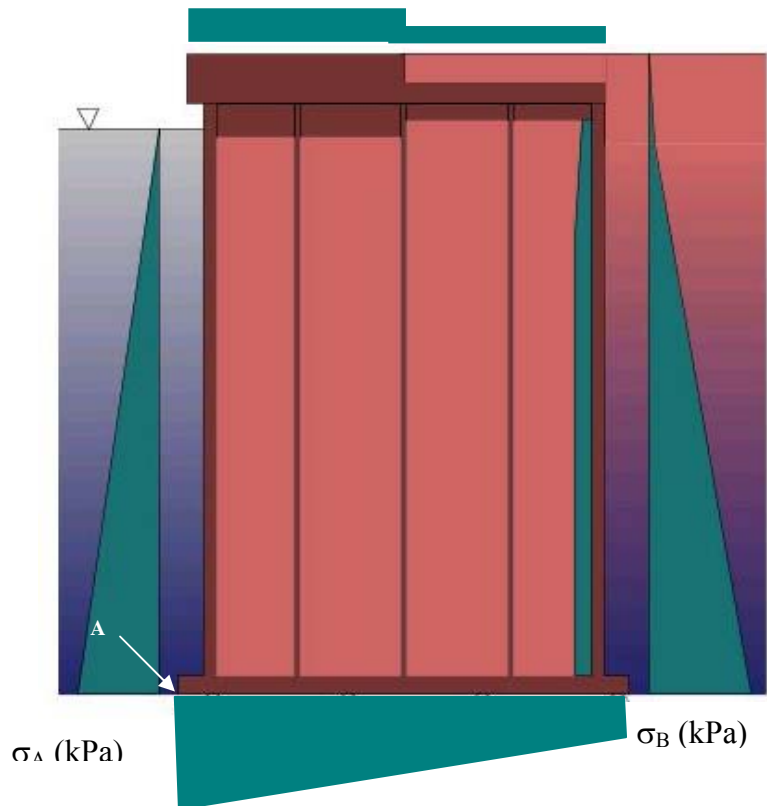


Figura III-105. Envoltorio de tensiones máximas en la losa

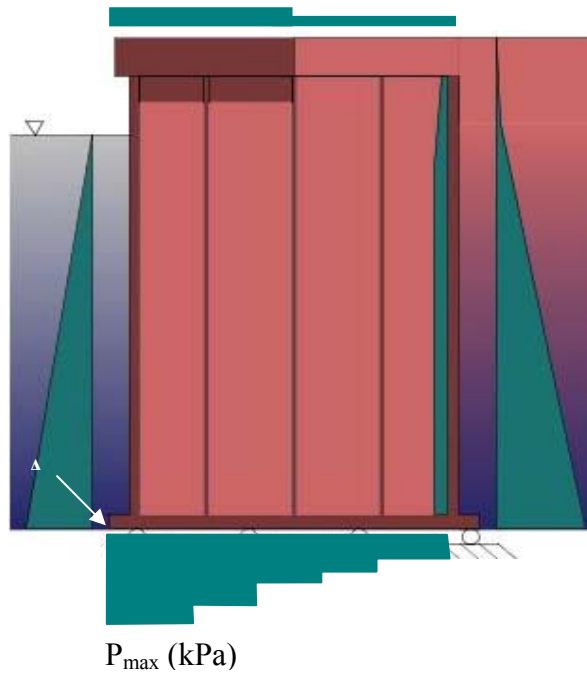


Figura III-106. Presión máxima en las celdas circulares

III.3.3.- Aplicación de las condiciones de contorno en carga y desplazamientos a la geometría multibloque

Una de las principales ventajas del algoritmo presentado en esta tesis para la generación de la geometría multibloque consiste en disponer de manera precisa de la ubicación de todas y cada una de las celdas y sus superficies, sobre las cuales serán aplicadas las condiciones de contorno en carga y desplazamientos.

Así, en el cálculo de cada hipótesis, determinadas las cargas a imponer, se dispone de un algoritmo que recorre uno a uno los cubos de cada una de las rebanadas en las que se ha dividido la geometría del cajón y, para cada una de las superficies que lo componen, comprueba si dicha superficie se ve afectada o no.

La clase CCBean



Figura III-107. Diagrama UML de la clase CCBean

La clase CCBean contiene todas las selecciones de puntos, líneas y superficies sobre las que se aplica alguna carga o desplazamiento restringido y que se escribirán en el fichero preciso para su aplicación a la malla generada en GiD (como se verá en el apartado III.4.2.2.-).

La clase CSalidaCC

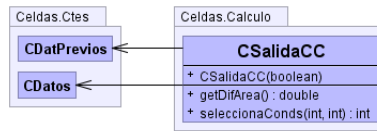


Figura III-108. Diagrama UML de la clase CSalidaCC

En esta clase se implementa el algoritmo que recorre uno a uno los cubos de la geometría y en función de la hipótesis de cálculo seleccionada rellena las colecciones de vectores con los puntos, líneas y superficies que se almacenan en CCBean.

III.3.4.- Modelado de las hipótesis

Para la resolución por elementos finitos del problema elástico lineal de un cajón multicelular sometido a una hipótesis tipo de carga, se define un paquete básico de clases que permiten resolver cualquiera de las hipótesis de carga indicadas en el apartado anterior.

El paquete HipotesisTipo

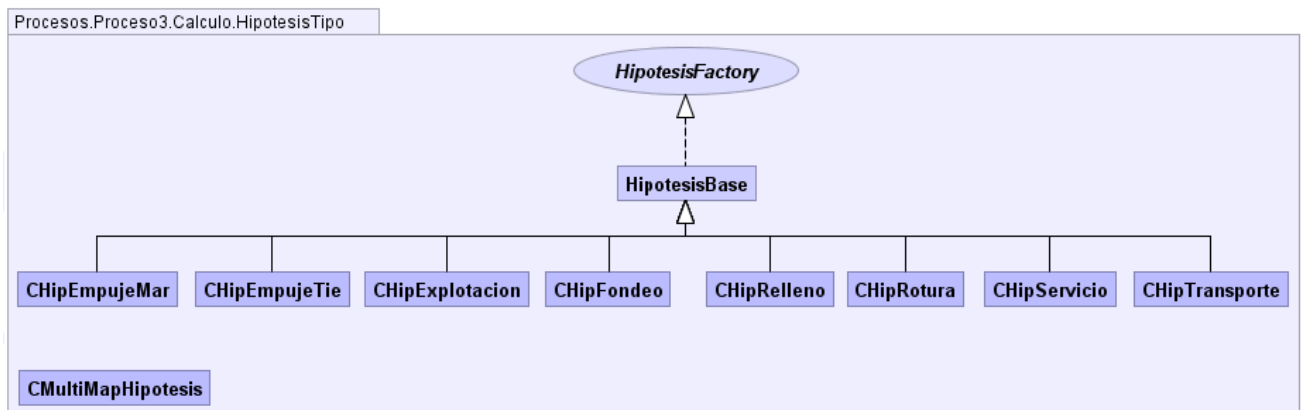


Figura III-109. Diagrama UML del paquete HipotesisTipo

Interfaz HipotesisFactory

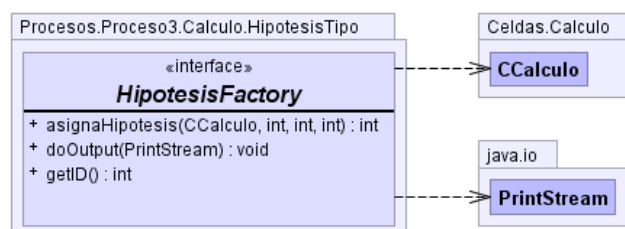


Figura III-110. Diagrama UML de la interfaz HipotesisFactory

Las clase abstracta *HipotesisBase*

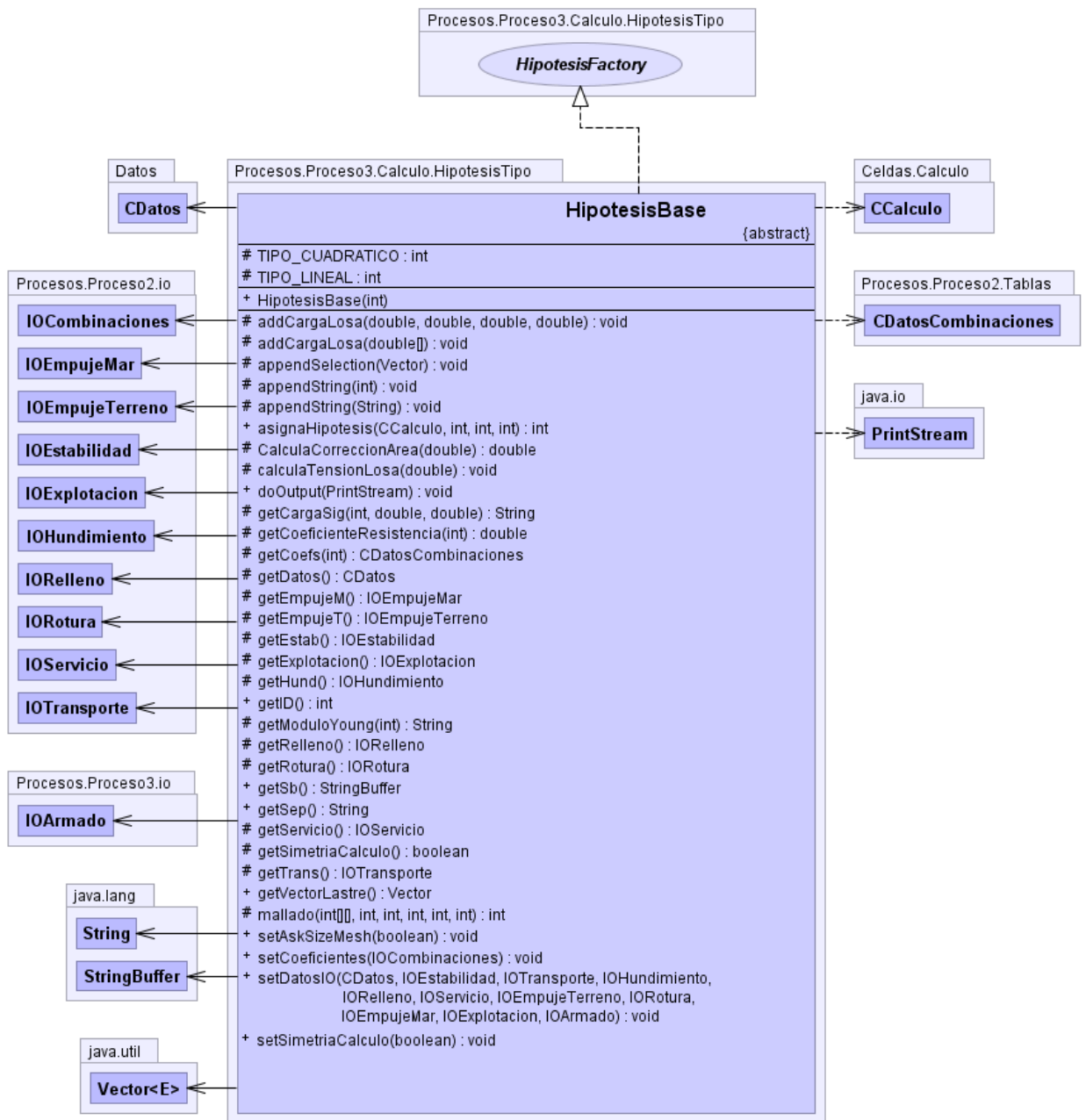


Figura III-111. Diagrama UML de la clase HipotesisBase

El contenedor *CMultiMapHipotesis*

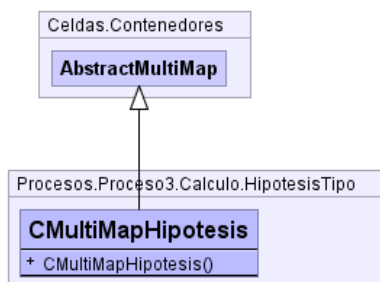


Figura III-112. Diagrama UML de la clase CMultiMapHipotesis

Las hipótesis son todas idénticas. Un ejemplo lo constituye la hipótesis de relleno.

Clase CHipRelleno

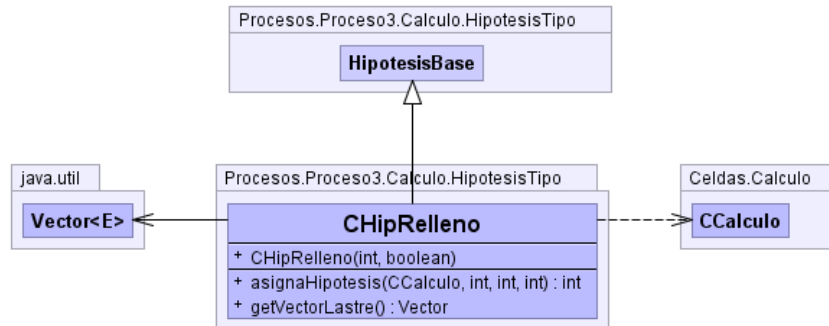


Figura III-113. Diagrama UML de la clase CHipRelleno

Paquete de Calculo

A partir de este paquete básico, se definen tres clases funcionales que permiten la resolución de hipótesis básicas por separado, combinación de hipótesis o armado de hipótesis.

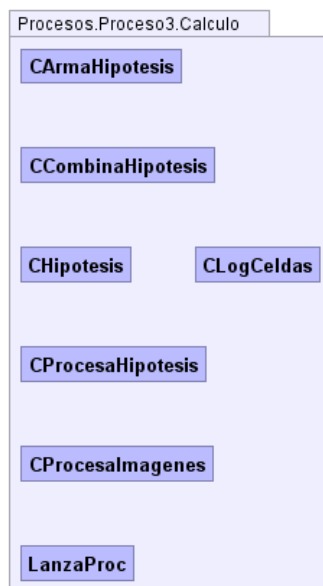


Figura III-114. Diagrama UML del paquete de Calculo

La clase CProcesaHipotesis

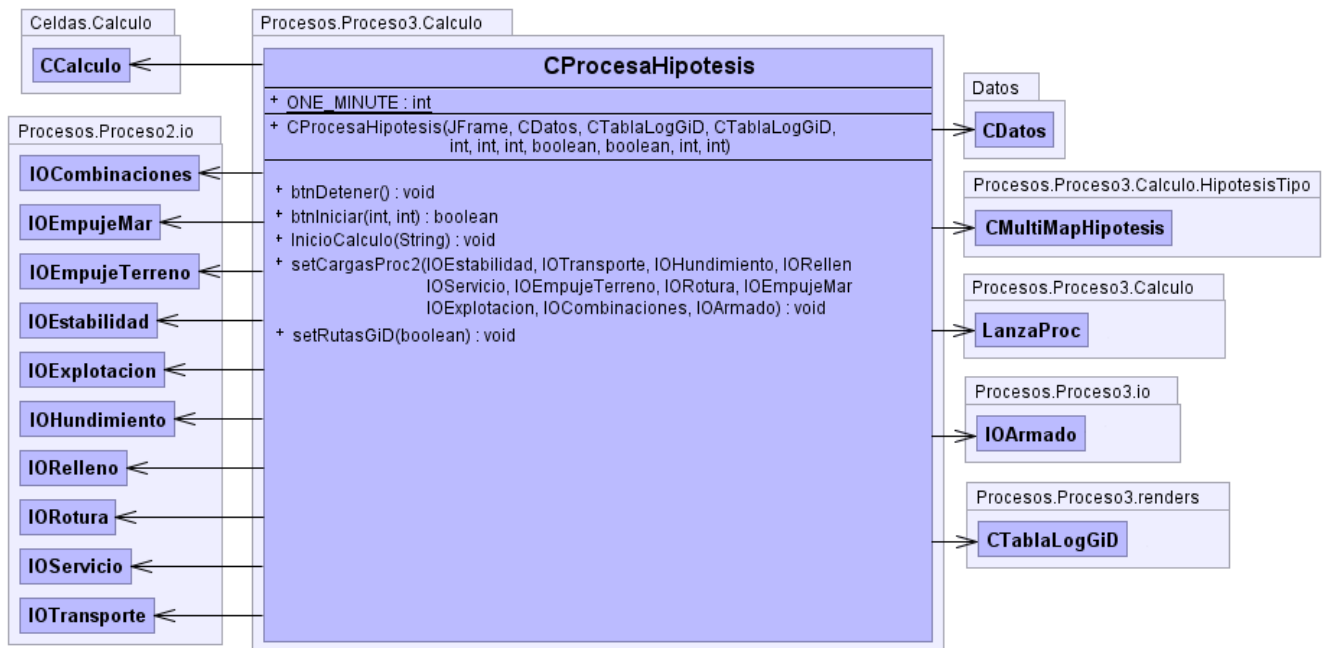


Figura III-115. Diagrama UML de la clase de CProcesaHipotesis

La clase CCombinaHipotesis

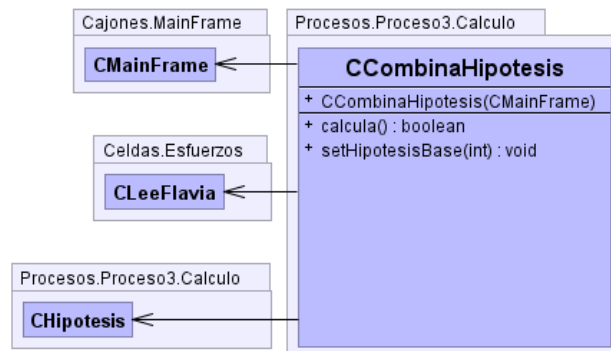


Figura III-116. Diagrama UML de la clase de CCombinaHipotesis

La clase CArmaHipotesis

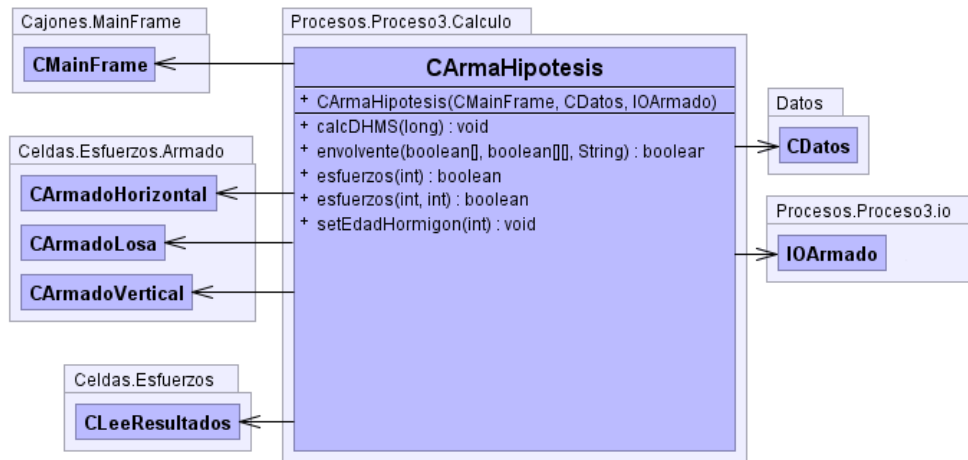


Figura III-117. Diagrama UML de la clase de CArmaHipotesis

III.4.- Esquema General de Resolución mediante el MEF

III.4.1.- Introducción

Como es bien sabido, la resolución mediante elementos finitos se estructura en tres grandes etapas: Pre-procesado, cálculo o análisis y post-procesado.

Pre-procesado

La etapa de pre-procesado comprende toda la parte de generación del modelo. En primer lugar, se genera la geometría y, a continuación, se procede a la discretización o mallado del conjunto, al que se le asignarán las propiedades del material y del elemento, las cargas y condiciones de contorno.

En concreto, en el caso de cajones multicelulares en el apartado III.2.- se ha presentado un algoritmo que se encarga de generar de forma automática la geometría del cajón con la particularidad de que las entidades creadas son, de facto, volúmenes hexaédricos colocados convenientemente.

La geometría creada se pasa al pre-procesador para su mallado. Concretamente se emplea el software GiD[®], desarrollado por el Centro Internacional de Métodos Numéricos en Ingeniería CIMNE [81].

Como la geometría está completamente definida por volúmenes hexaédricos, la generación de malla estructurada con elementos de tipo hexaedro de 20 nodos es inmediata.

Para enlazar GiD con el software de análisis, se introduce en el pre-procesador el concepto de *Problem Type* o tipo de problema [84], información que hay que proporcionar al sistema para que éste pueda operar con un programa de análisis determinado.

Análisis

Esta etapa se realizará en el *solver*. En esta fase se ensamblan las matrices de rigidez de los elementos y se resuelven los sistemas bien por métodos directos o iterativos.

Para los distintos estados de carga que se han detallado en apartados anteriores se obtiene una solución de desplazamientos y tensiones. Concretamente se emplea el software CALSEF[®] [126], desarrollado por el Centro Internacional de Métodos Numéricos en Ingeniería CIMNE.

Como *Problem Type* en GiD, se empleará *Calsef2001_Solidos_3D*, desarrollado por los mismos autores como interfaz entre pre-procesador y software de análisis.

Post-procesado

En esta fase ya se dispondrá de los resultados del cálculo y de lo que trata es de proveer de una serie de herramientas que permitan la visualización, generalmente mediante mapas de colores, animaciones, iso-superficies, ficheros de datos, etc. para la comparación y comprobación de los resultados obtenidos de la etapa anterior, con lo cual quedaría finalizado el proceso.

Se obtienen unos resultados que en esencia son de dos tipos:

- a) *Resultados gráficos*: Proporcionan información sobre el valor de las tensiones principales en puntos y/o en secciones estratégicas del interior de la sección con un mayor detalle para las de tracción que en definitiva son las que fundamentalmente intervienen en el posterior dimensionamiento de la armadura necesaria para resistirlos.
- b) *Resultados numéricos*: Proporcionan para cada caso de carga considerado información completa de:
 - Características generales del problema.
 - Definición de la malla de Elementos Finitos.
 - Propiedades del material considerado.
 - Datos de las cargas consideradas.
 - Desplazamientos obtenidos en los nodos.
 - Valores en los nodos de cada elemento de la malla de las tres tensiones asociadas al estado bidimensional y de las dos tensiones principales obtenidas a partir de ellas.
 - Esfuerzos obtenidos por integración de las tensiones en las distintas secciones de diseño consideradas como más críticas durante la fase de dimensionamiento.

III.4.2.- Pre-procesado en GiD

Como se ha mencionado, el algoritmo de generación de la geometría multibloque contiene la información precisa para construir el cajón mediante bloques hexaédricos. Para obtener una malla de elementos hexaédricos a partir de esta geometría se empleará GiD, por lo que es necesario enviar tanto la geometría como las condiciones del problema (material, condiciones de contorno) en un formato de intercambio que sea admitido por GiD.

A continuación se describe el formato de los ficheros de intercambio de geometría (*.geo) y de condiciones del problema (*.bat) empleados para generar de manera automática, sin intervención del usuario, la generación de la malla de elementos finitos previa a la resolución del problema.

III.4.2.1.- Formato de ficheros de geometría

La estructura del fichero de geometría en formato ASCII [122] es la siguiente:

Cabecera
ProblemType IsQuadratic
Debe Repararse
Una entrada para cada Capa
Entidad nula <fin de capa>
Una entrada para cada dato de mallado
Entidad nula <fin de datos de mallado>
Una entrada para cada punto
Una entrada para cada curva
Una entrada para cada superficie
Una entrada para cada volumen
Entidad nula <fin de entidades>
Hay cambios
Último tamaño de mallado
Una entrada para cada acotación
Entidad nula < fin de acotaciones>
Versión de GiD
Variables de mallado

Tabla III-8. Ordenación de entidades en fichero de geometría en GiD

Como *Cabecera* se indica, para identificar el tipo de fichero, RAMSAN-ASCII-gid-v2.22

Como *Problem Type* o tipo de problema se indica UNKOWN, en caso de no cargar ningún tipo concreto o CALSEF2001_Solidos_3D, y en *IsQuadratic* se señala si se emplean elementos lineales (0) o cuadráticos (1).

En *Debe repararse* se indica con un 0 que no hace falta usar la función Repair al acabar la lectura.

A continuación se detalla para cada una de las capas en las que se divide la geometría (una por cada rebanada en altura en las que se divide el cajón): número, nombre de la capa, 0/1 si está inhabilitada o no, 0/1 si está inactiva o no, y tres parámetros RGB_R RGB_G RGB_B con valores de 0 a 255

Como código de fin de capas se indica un 0.

En el caso de indicarse datos de malla, se reflejarían primero los datos de malla no estructurada: código -1, número de identidad, número de entidades que la apuntan, tipo de elemento [0 ninguno, 1 lineal, 2 triangular, 3 cuadrilátero, 4 tetraedro, 5 hexaedro, 6 prisma, 7 sólo puntos], 0/1 si debe mallarse, tamaño del lado del elemento. A continuación se indicarían los datos de la malla estructurada, con código -2 y los datos de la malla estructurada con pesos, con código -3.

Como código de fin de datos de mallado se indica un 0.

Para cada una de las capas:

Se detallan los puntos: código 1, número de punto, etiqueta (0), selección (0), número de entidades (líneas o arcos) a las que está subordinado, condiciones (0),

material (0), número de capa a la que pertenece, mesdata(0) y en línea aparte, las coordenadas X, Y, Z.

Se detallan las líneas rectas: código 2, número de línea, etiqueta (0), selección (0), número de entidades (superficies) a las que está subordinada, condiciones (0), material (0), número de capa a la que pertenece, mesdata(0) y en línea aparte: número de punto inicial, número de punto final.

Se detallan los arcos: código 3, número de arco (sigue numeración correlativa con líneas), etiqueta (0), selección (0), número de entidades (superficies) a las que está subordinada, condiciones (0), material (0), número de capa a la que pertenece, mesdata(0) y en línea aparte: número de punto inicial, número de punto final, coordenadas X e Y del centro, radio, ángulo inicial y ángulo final. En líneas aparte la matriz de transformación 4x4 identidad, siendo el elemento 4x3 la coordenada Z del arco.

En el caso de que halla polilíneas, se reflejaría código 4. Las curvas NURB tienen código 11. Las superficies planas tienen código 5.

Se detallan las superficies COON de 4 lados: código 6, número de superficie, etiqueta (0), selección (0), número de entidades (volúmenes) a las que está subordinada, condiciones (0), material (0), número de capa a la que pertenece, mesdata(0). En línea aparte, número de líneas de contorno: 4. En línea aparte números identificativos de las 4 curvas. En línea aparte sentido del tramo 0/1 de cada una de las 4 curvas. En línea aparte coordenadas X, Y, Z del centro de la superficie. Finalmente, en línea aparte, las 3 componentes de la dirección del vector normal al plano de la superficie.

En caso de indicar superficies NURB, se emplea el código 14.

Por último, para los volúmenes: código 9, número de volumen, etiqueta (0), selección (0), número de entidades a las que está subordinado (0), condiciones (0), material (0), número de capa a la que pertenece, mesdata(0). En línea aparte, número de superficies de contorno: 6. En línea aparte números identificativos de las 6 curvas. En línea aparte sentido del tramo 0/1 de cada una de las 6 superficies. Finalmente, en línea aparte coordenadas X, Y, Z del centro del volumen.

Concluidas todas las capas, se indica 0 como fin de entidades.

Se añade 1/0 si hay o no cambios.

Se indica el último tamaño de mallado.

Se indican las acotaciones, si existen.

Se indica 0 como fin de acotaciones.

Por último, se indican las variables de mallado: *MeshUntilEnd* 0/1, *AllowAutomaticStructured* 0/1, *MaintainOldMesh* 0/1, *NoMeshFrozenLayers* 0/1, *AutomaticCorrectSizes* 0/1, *HighQualitySmoothing* 0/1, *SizeTransitionsFactor*,

SurfaceMesher 0/1, *VolumeMesher* 0/1, *ForceMeshEntities* 0/1. Y en línea aparte, la ruta y el nombre del fichero con la malla de fondo, si existe.

El siguiente extracto de un fichero de geometría resume el formato que se emplea para la lectura en GiD de la geometría multibloque de un cajón, con comentarios indicativos en la columna derecha.

<pre>RAMSAN-ASCII-gid-v2.24 CALSEF2001_Solidos_3D 1 0 1 Cajon_Losa 0 1 0 0 255 2 Cajon_Fuste_Tramo1 0 1 0 0 245 3 Cajon_Fuste_Tramo2 0 1 0 0 235 4 Cajon_Fuste_Tramo3 0 1 0 0 225 ... 17 Cajon_Murete 0 1 0 0 95 0</pre>	<p><i>Cabecera</i> <i>ProblemType - Cuadrático</i></p> <p><i>Capas 1 a 17</i></p> <p><i><Fin de capas></i></p>
<pre>-2 1 26836 5 0 0 -2 2 94316 3 0 0 -2 3 71068 1 0 1 -2 4 29610 1 0 2 -2 5 6632 1 0 1 0</pre>	<p><i>Mallado estructurado de hexaedros (volúmenes), cuadriláteros (superficies) y líneas (rectas y arcos)</i></p> <p><i><Fin de datos de malla></i></p>
<pre>1 1 0 0 7 0 0 1 0 3.860000 2.288599 0.000000 1 2 0 0 5 0 0 1 0 2.932500 2.824091 0.000000 ... 1 39831 0 0 3 0 0 17 0 40.278000 15.250000 17.500000</pre>	<p><i>Punto 1</i></p> <p><i>Punto 2</i></p> <p><i>Punto 39831</i></p>
<pre>2 1 0 0 3 0 0 1 3 1 2 2 2 0 0 3 0 0 1 3 2 3 ... 2 107310 0 0 2 0 0 17 5 38903 39831</pre>	<p><i>Línea 1</i></p> <p><i>Línea 2</i></p> <p><i>Línea 107310</i></p>
<pre>3 189 0 0 3 0 0 1 3 51 52 2.005 1.217614 1.755 0.523599 1.047198 1.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 1.000000 ... 3 102249 0 0 2 0 0 16 3 38667 38656 31.685 14.0694 1.755 0.0 0.5236 1.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000 0.000000 0.000000 17.200001 1.000000</pre>	<p><i>Arco 189</i></p> <p><i>Arco 102249</i></p>

3. Los Cajones Multicelulares

<pre> 6 1 0 0 1 1 0 1 2 4 1 76 189 75 0 0 1 1 3.405138 2.431756 0.000000 0.000000 0.000000 1.000000 6 2 0 0 1 1 0 1 2 4 2 77 190 76 0 0 1 1 2.356408 3.037240 0.000000 0.000000 0.000000 1.000000 ... 6 94316 0 0 1 1 0 17 2 4 102701 107310 106382 107309 0 0 1 1 40.278000 15.291667 17.350000 -1.000000 0.000000 0.000000 </pre>	<p><i>Superficie 1</i></p> <p><i>Superficie 2</i></p> <p><i>Superficie 94316</i></p>
<pre> 9 1 0 0 0 0 2 1 1 6 1 6857 6932 7045 6931 3429 0 1 1 0 0 1 3.405138 2.431756 0.275000 9 2 0 0 0 0 2 1 1 6 2 6858 6933 7046 6932 3430 0 1 1 0 0 1 2.356408 3.037240 0.275000 ... 9 26836 0 0 0 0 2 17 1 6 88414 94303 94310 94304 94316 92750 0 0 1 1 0 1 39.979140 15.042617 17.350000 0 </pre>	<p><i>Volumen 1</i></p> <p><i>Volumen 2</i></p> <p><i>Volumen 26836</i></p> <p><i><Fin entidades></i></p>
<pre> 0 0.000000 0 7.2 1 0 0 1 1 0 0.600000 0 0 0 </pre>	<p><i>No hay cambio</i></p> <p><i>Último tamaño mallado</i></p> <p><i><Fin acotaciones></i></p> <p><i>Versión de GiD</i></p> <p><i>Datos de malla</i></p>

Tabla III-9. Ejemplo de fichero de geometría de un cajón

Nótese que los datos de geometría inicialmente no incluyen datos de malla. No obstante, una vez generada la malla en GiD, se puede exportar el proyecto como ASCII y el fichero *.geo contendrá además de la geometría los datos de la malla generada (tal y como se ha indicado en azul en la **Tabla III-9**).

La siguiente figura ilustra un ejemplo de la geometría multibloque del cajón generada y leída en GiD. En concreto se han empleado:

- 39831 puntos
- 107310 líneas y arcos
- 94316 superficies
- 26836 volúmenes

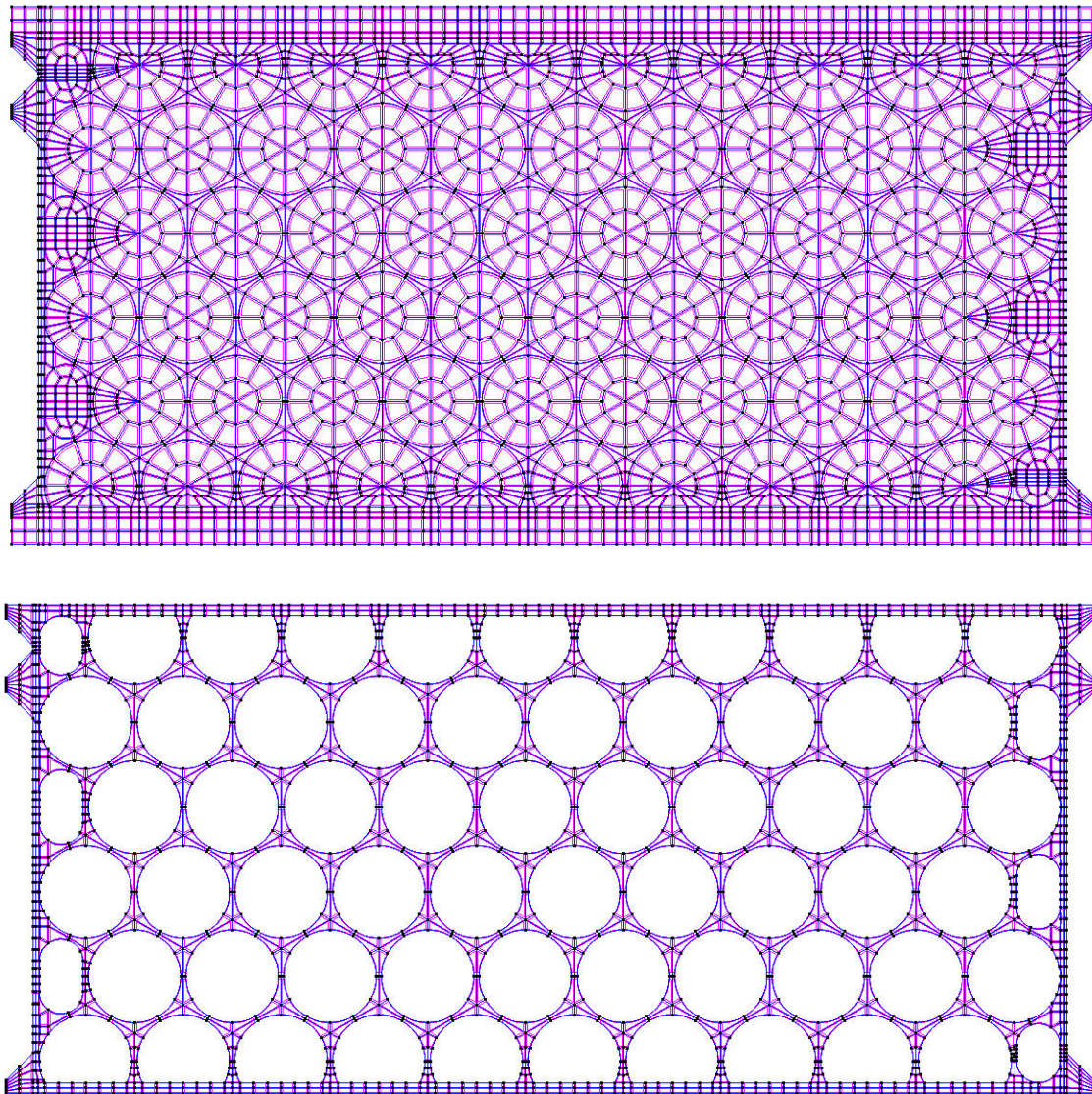


Figura III-118. Geometría modelada: Losa y fuste de un cajón multicelular

La siguiente figura muestra ampliada la geometría del entorno de una celda circular, en la que se puede apreciar que cada superficie es un cuadrilátero. Al extruir en altura se genera un volumen hexaédrico de 6 superficies. También se puede observar que los puntos han sido colocados estratégicamente en las zonas de mayor interés en el posterior análisis de las tensiones: en las *gargantas*, o zonas más críticas de la pieza, que son las de menor espesor de hormigón (en este caso es de 20 cm). Se garantiza así que en el mallado se sitúe un significativo número de nodos en dichas posiciones y, por tanto, se disponga de suficientes resultados tensionales para obtener los esfuerzos con los que realizar el armado.

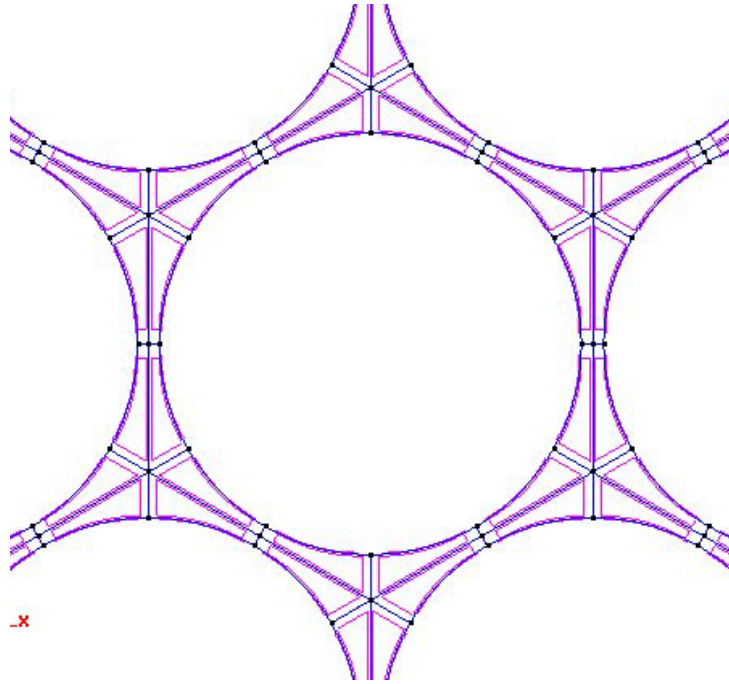


Figura III-119. Detalle de la geometría en torno a una celda

Por otro lado, tal y como muestra la **Figura III-120**, en la dirección vertical se han efectuado 16 divisiones del fuste, con el objeto de no tener elementos de gran longitud frente a las otras dos dimensiones y para disponer del máximo número de nodos posibles en altura y conocer con más precisión la variación de los esfuerzos con los que realizar el armado.

En la zona próxima a la losa, la altura de los volúmenes del fuste es bastante menor (concretamente 55 cm), con el fin de minimizar la desviación en los resultados en términos de esfuerzos que se obtienen al considerar en la zona en contacto con la losa, las tensiones generadas en esta. La última división, correspondiente al muro perimetral, tiene una altura de 30 cm. El resto de volúmenes del fuste tiene una altura de $(16.65-0.55)/14=1.15$ m. Para la losa se considera un único volumen en altura, de 0.55 m por tanto.

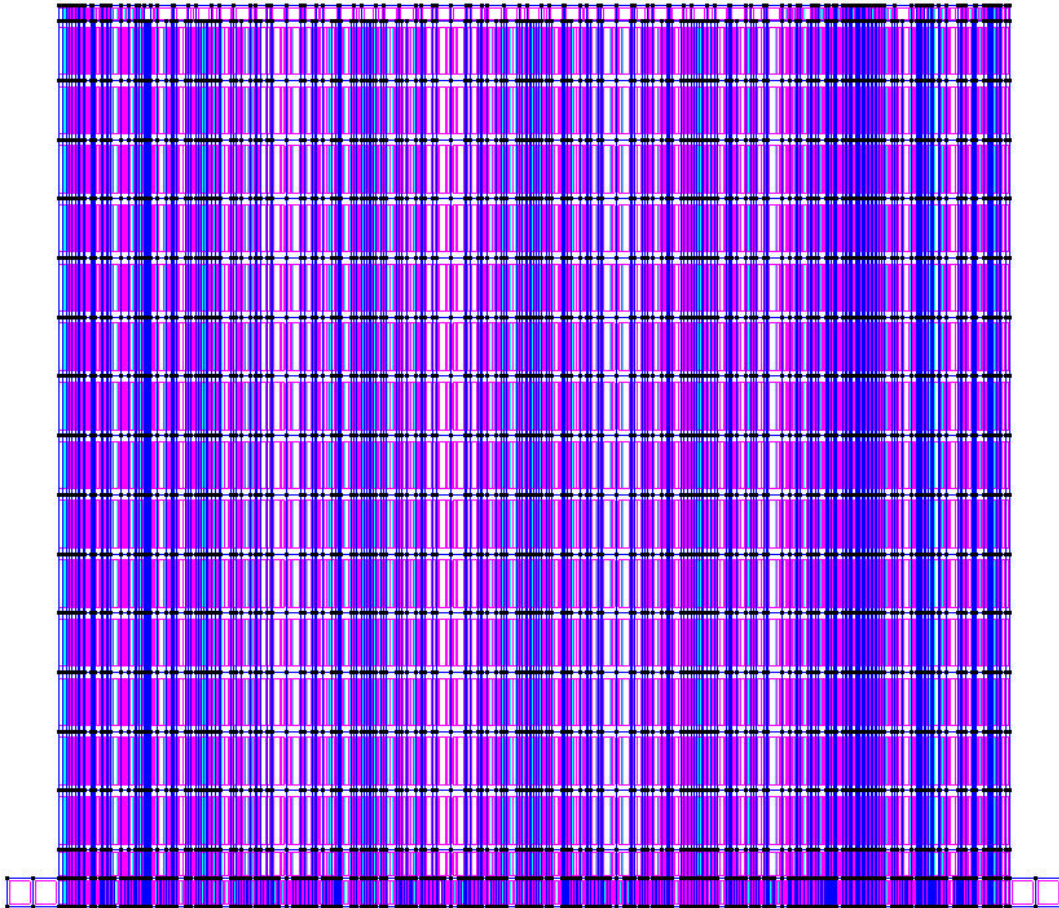


Figura III-120. Vista de las divisiones del cajón en altura

III.4.2.2.- Datos del problema, condiciones de contorno y de mallado

Conocida la geometría, es necesario completar las propiedades del material del cajón así como definir las condiciones de contorno en desplazamientos y en cargas que se van a aplicar en la resolución del problema. También se indicará la forma de mallar el cajón.

Para ello se emplea un fichero por lotes, *.bat, con la estructura que se describe a continuación. Se trata de definir todos los parámetros a través de los diferentes diálogos en GiD que definen el problema, pero de modo automático mediante scripts las acciones realizadas para rellenar los campos solicitados en cada uno de ellos.

ProblemType

En primer lugar se especifica el nombre de un fichero de registro en el que se guardarán todos los mensajes que se generan durante la ejecución. Se indican dos parámetros relativos a la corrección de la geometría al importar el fichero y se especifica el problemtype.

```
****OUTPUTFILENAME <ruta proyecto>\proyecto.gid\proyectoErr.txt
escape escape escape escape utilities variables AutomaticCorrectSizes 0 escape escape
escape escape escape escape utilities variables AutoCollapseAfterImport 0 escape escape
escape escape escape escape data defaults ProblemType yes CALSEF2001 Solidos 3D escape
```

Material

A continuación se define un nuevo material, indicando nombre asignado y los valores del módulo de elasticidad, coeficiente de Poisson y peso propio (convenientemente afectado de coeficiente de seguridad y del factor de corrección de la geometría, ecuación (III-64)). Se indica la selección de volúmenes que se ven afectados por este material

```
escape escape escape escape data materials newmaterial Hormigon_(N,m,rad)
Hormigon_(N,m,rad) yes 30545573721.45 0.2 33108.75
escape escape escape escape data mat assign Hormigon_(N,m,rad) volumes
selection
1:26836
```

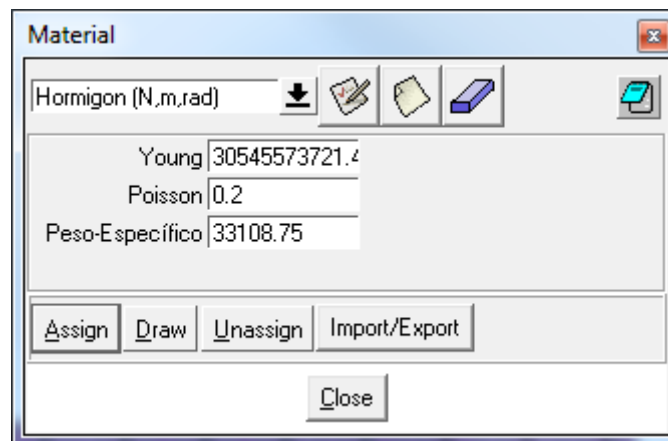


Figura III-121. Diálogo de creación y asignación de materiales en GiD

Datos del problema

Se define un título para el problema, indicando si se desea tener una salida de texto además de en binario, considerar el peso propio en el problema y los tres grados de libertad asociados a cada nodo.

```
escape escape escape escape data problemdata Cajon Multicelular Si Si 1.0 1 1 1
```

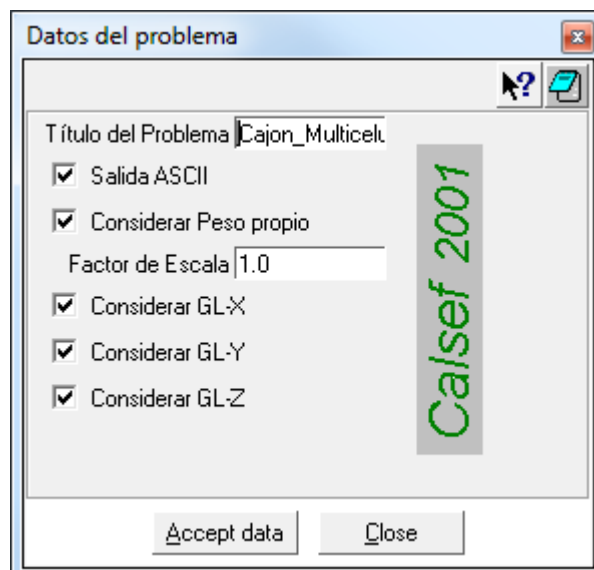


Figura III-122. Diálogo para introducción de datos generales del problema en GiD

Condiciones de contorno en desplazamientos

A continuación se definen las condiciones de contorno en desplazamiento relativas a cada hipótesis de carga, asignando restricciones puntuales, lineales o superficiales, en ejes globales, en las direcciones X, Y, Z (1 restringido, 0 libre) con valores prescritos (0 por defecto). A continuación se indica la selección de entidades afectadas (puntos, líneas o superficies).

```
escape escape escape escape data cond assign Restricciones_Superf. change -GLOBAL- 0
0.0 0 0.0 1 0.0
selection
1:3428
```

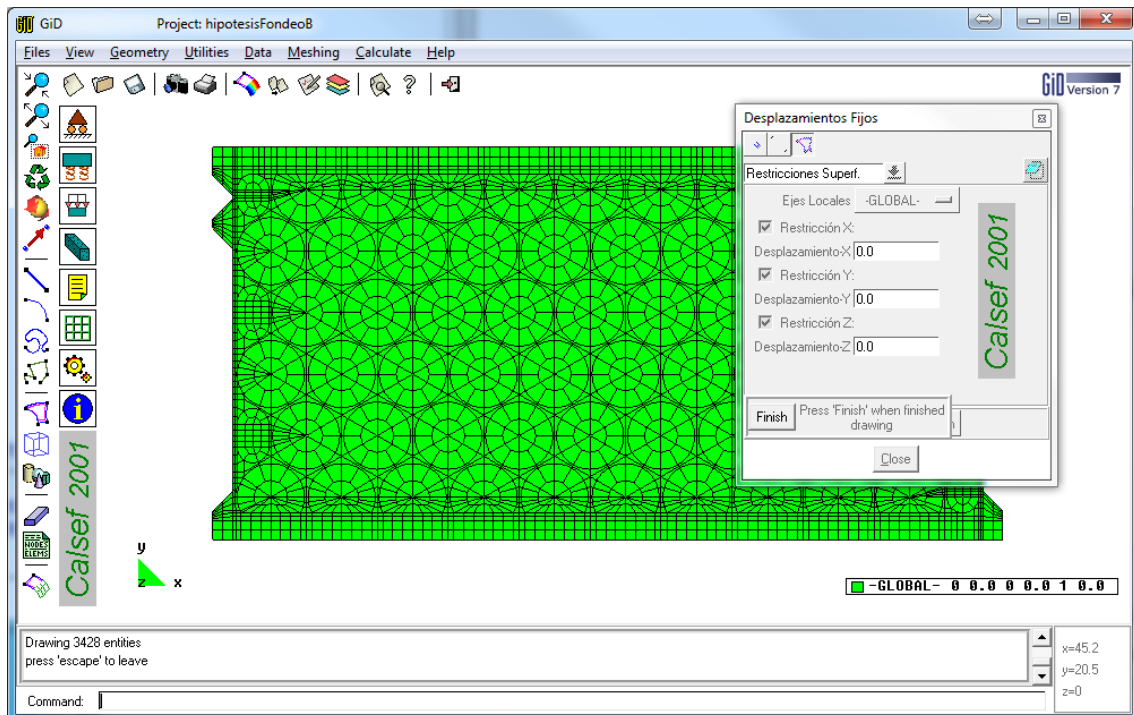


Figura III-123. Visualización de restricciones superficiales en GiD

Condiciones de contorno en carga

A continuación se definen las condiciones de contorno en carga relativas a cada hipótesis de carga, asignando restricciones puntuales o superficiales en ejes globales, en las direcciones X, Y, Z.

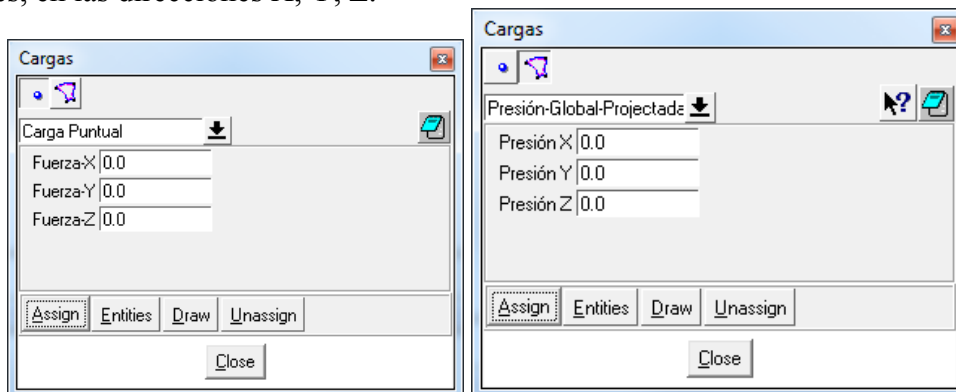


Figura III-124. Diálogos de asignación de cargas puntuales o superficiales en ejes globales en GiD

3. Los Cajones Multicelulares

Igualmente se pueden definir presiones uniforme sobre las paredes horizontales (Z) o verticales (X, Y) del cajón, sin más que identificar la dirección y el valor de la carga (con su coeficiente de seguridad) y la selección de superficies afectadas.

```
escape escape escape escape data cond assign Presión-Uniforme change GLOBAL 0.0 0.0 -  
88864.41474 0.0 0.0 0.0 0.0 0.0 0.0  
selection  
3473 3474 3475 3476 3477 3478 ... 3500 3501 3502  
escape
```

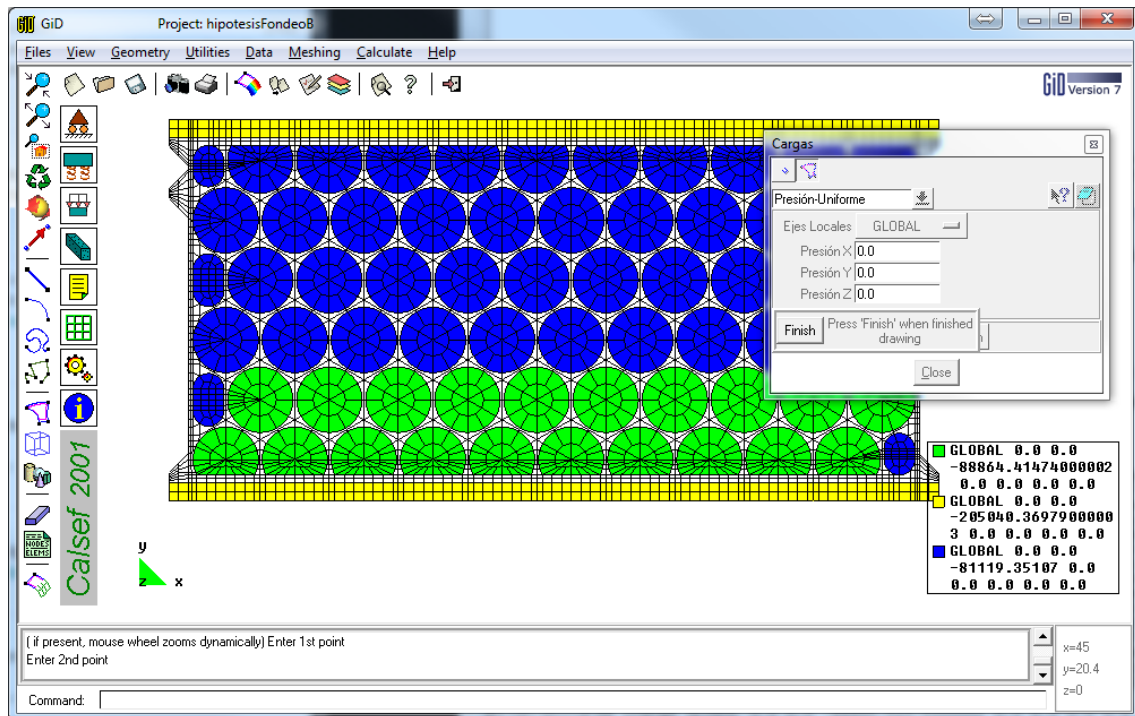


Figura III-125. Visualización de presiones uniformes sobre superficies en GiD

Del mismo modo, se pueden definir presiones hidrostáticas sobre las paredes verticales del cajón, sin más que identificar la cota Z desde el fondo del cajón que delimita la superficie libre del agua y la presión de la misma (con su coeficiente de seguridad).

```
escape escape escape escape data cond assign Presión-Hidrostática change 7.09  
13587.8310  
selection  
15485 15486 15487 15488 15489 ... 15504 20701  
escape
```

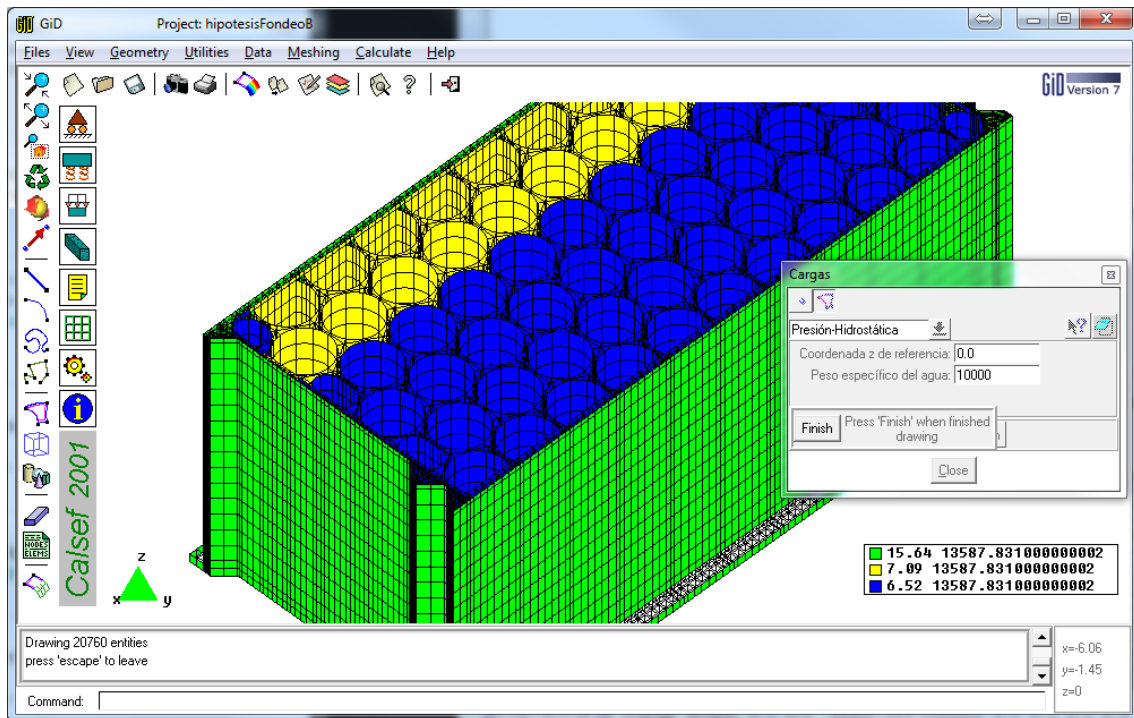


Figura III-126. Visualización de superficies con presión hidrostática asignada en GiD

Estos comandos se pueden repetir tantas veces como sea necesario, hasta haber asignado todas las cargas existentes.

Mallado

Finalmente, se especificará en el fichero la forma de mallar los volúmenes del cajón. Para lo cual se seleccionan todos y se indica que se realice un mallado estructurado, habitualmente con elementos cuadráticos (*Quadratic*) de 20 nodos o en su lugar con elementos lineales (*Normal*) de 8 nodos.

```
escape escape escape escape Meshing quadratic Quadratic escape
escape escape escape escape Meshing Structured Volumes
selection
1:26836
escape
```

A continuación, se indica el número de divisiones que se desean por cada lado en planta (1/2/3) y se seleccionan todas las líneas y posteriormente se deseleccionan las líneas verticales:

```
1
selection
1:107310
14033:17621
...
106383:107310
escape
```

Por último, se indica el número de divisiones que se desean en altura por cada línea vertical (1/2/3) del fuste y se seleccionan todas las líneas verticales. Igualmente con las líneas verticales de losa y murete, en las que se realiza una división menos que en las rebanadas del fuste:

3. Los Cajones Multicelulares

```
2
selection
27133:29247
...
102702:104816
escape
1
selection
14033:17621
...
106383:107310
escape
```

Tras la selección de entidades, se indica el comando de generar malla y visualizarla cuando finalice la generación.

```
escape escape escape escape Meshing Generate
escape escape escape escape Meshing MeshView
escape
```

Final

Para finalizar, se indica que se guarde el fichero de cálculo *.cal, se guarda el problema y se sale de GiD.

```
escape escape escape escape utilities variables DefaultFileNameInCalcFile 1 escape
escape escape escape escape files WriteCalcFile
escape escape escape escape utilities variables DefaultFileNameInCalcFile 0 escape
escape escape escape escape files save
escape escape escape escape quit
```

III.4.2.3.- Generación de mallado estructurado

Una vez que se dispone del fichero de geometría y del fichero con las condiciones de contorno, la forma de abrir GiD y generar el mallado de la geometría multibloque del cajón es a través de la siguiente orden en línea de comandos:

```
"<ruta GiD>\gid.exe" -b "<ruta proyecto>\proyecto.gid\proyecto.bat" -n
"<ruta proyecto>\proyecto.gid"
```

Tabla III-10. Orden de generación de mallado en línea de comandos

La opción `-b` hace referencia al uso de un fichero de lotes, y la opción `-n` indica que GiD no sea mostrado durante el proceso.

Para la obtención de los datos del número de nodos y elementos generados tras el mallado, justo antes de iniciar el proceso de cálculo se emplea el siguiente fichero '*mall.bas*' ubicado en la carpeta del problemtype `Calsef2001_Solidos_3D.gid`:

```
$-----
Número de elementos y nodos de la malla
*nelem
*npoin
$-----
```

Tabla III-11. Fichero script para la obtención de datos de malla

Tras la ejecución de dicho script se genera un fichero *.dat con la información solicitada.

<pre> \$----- Número de elementos y nodos de la malla 48088 258133 \$----- </pre>
--

Tabla III-12. Fichero con los datos de malla

Igualmente, se generan los siguientes ficheros: *.cnd, *.dat, *.cal, *.lin, *.msh, *Err.txt.

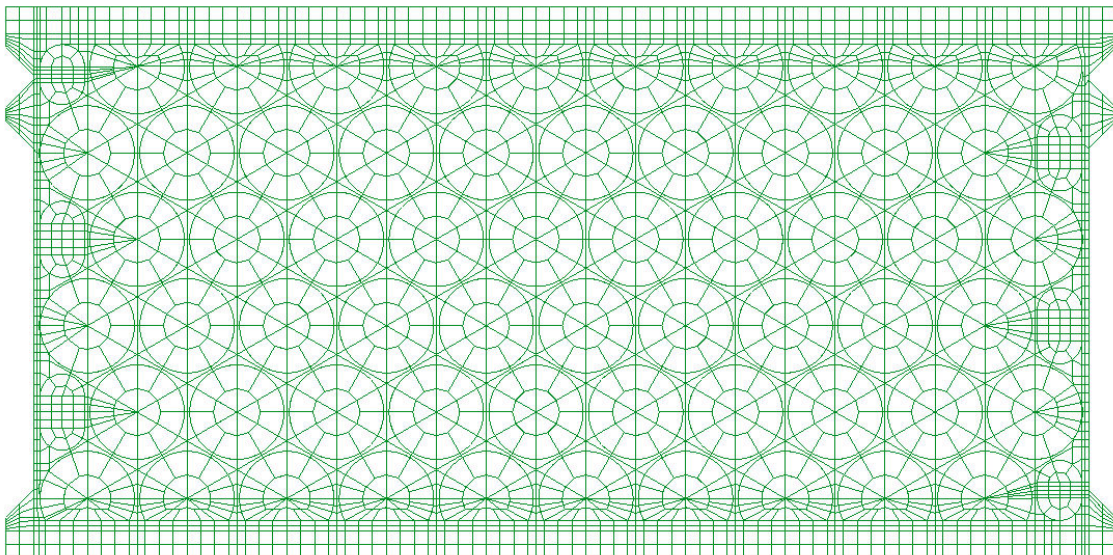
La **Figura III-127** ilustra la malla generada en GiD para el cajón del ejemplo. En concreto, en la losa, el muro perimetral y la primera división geométrica en altura del fuste, cada elemento está constituido por un volumen hexaédrico. En los 14 niveles restantes del fuste, cada volumen da lugar a dos elementos hexaédricos por división de su altura en dos partes iguales.

Se han obtenido las siguientes entidades:

- 48088 elementos hexaédricos
- 258133 nodos de tres grados de libertad, por lo que se deben calcular un total de 774399 grados de libertad. El número de ecuaciones será ligeramente menor, aproximadamente 750000, tras imponer las condiciones de contorno en cada hipótesis de carga

En la **Figura III-128** se reproduce una vista en planta de la malla obtenida, tanto para la parte de la losa, como para la parte del fuste, y en la **Figura III-129** se muestra la división de elementos en altura. En el fuste, salvo la primera altura de 55 cm y la última de 30 cm, los elementos tienen 57.5 cm de altura.

La **Figura III-129** muestra una vista isométrica de la malla del cajón.



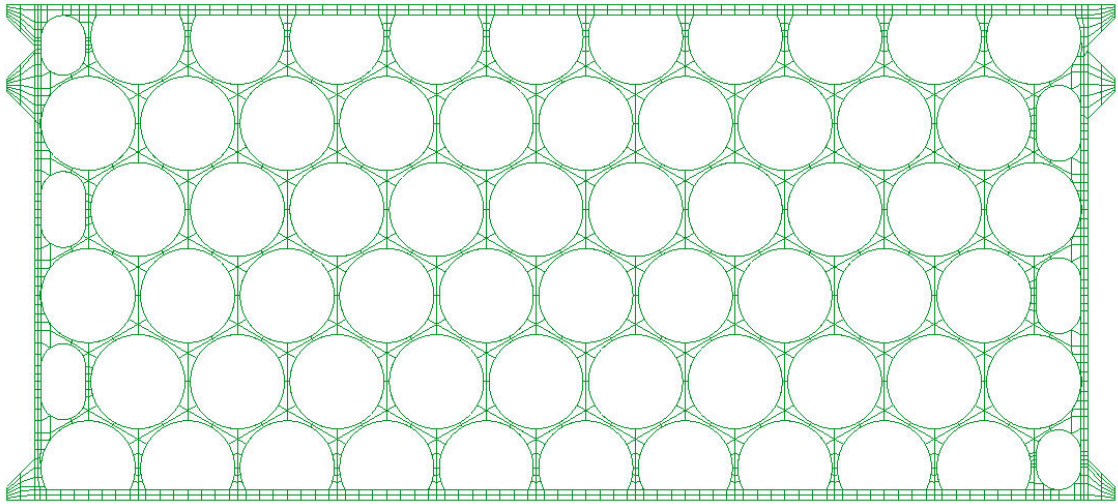


Figura III-127. Malla realizada en losa y fuste del cajón

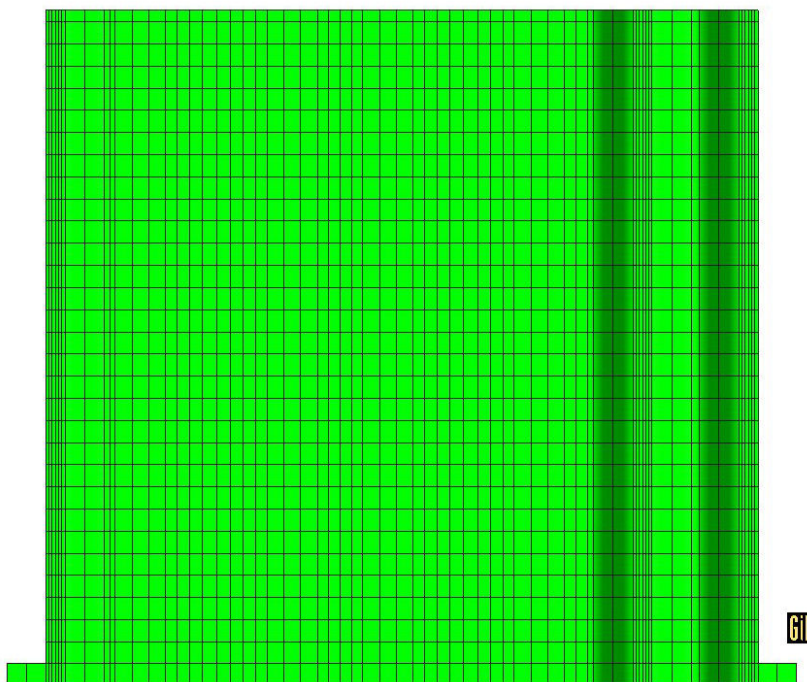


Figura III-128. Vista en altura de la malla realizada

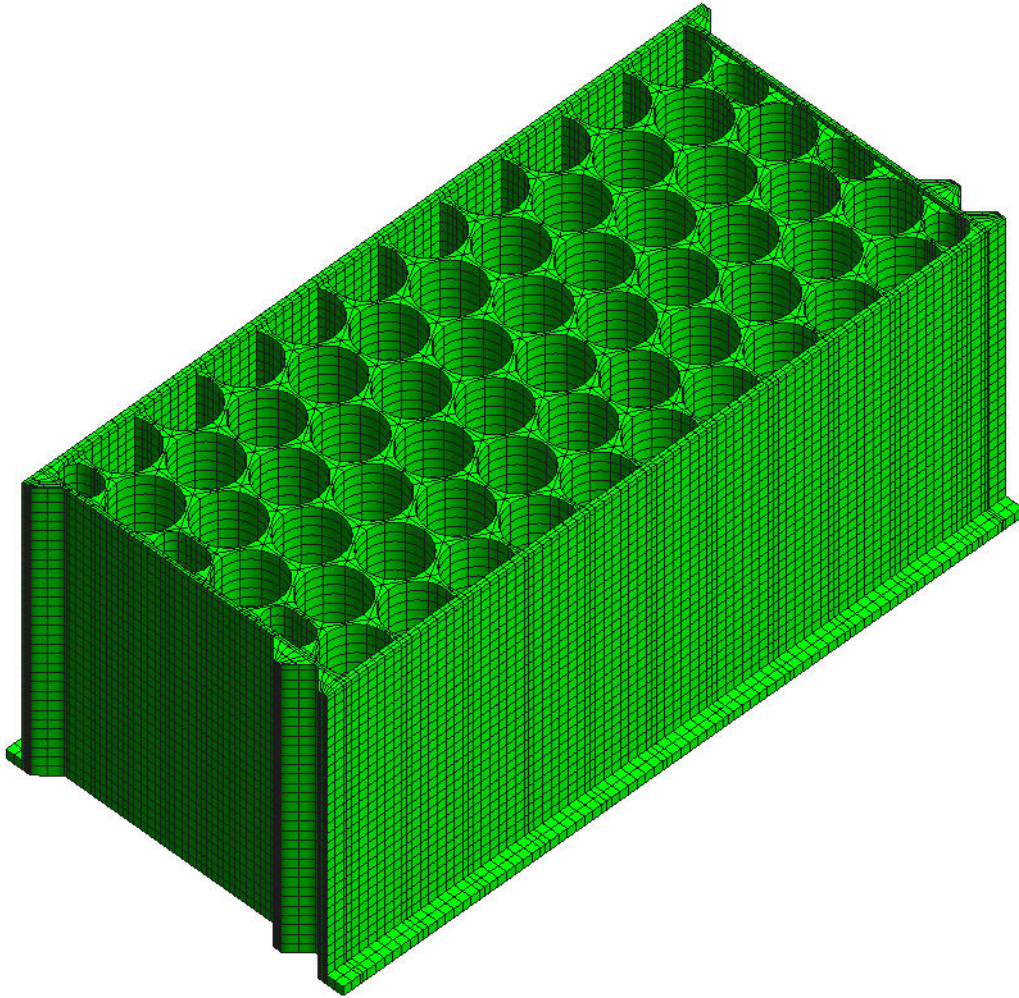


Figura III-129. Vista tridimensional de la malla

En la **Figura III-129** **Figura III-130** se muestra en detalle la colocación de los nodos en el entorno de la celda circular y de las gargantas, sobre los que se sitúan 5 nodos.

De no haber efectuado el tratamiento previo de la geometría para obtener volúmenes hexaédricos, habría que haber realizado una malla no estructurada con elementos tetraédricos. Ello hubiera supuesto un número muy elevado de nodos muy irregularmente dispuestos, lo que dificultaría notablemente la generación posterior de los esfuerzos para el armado.

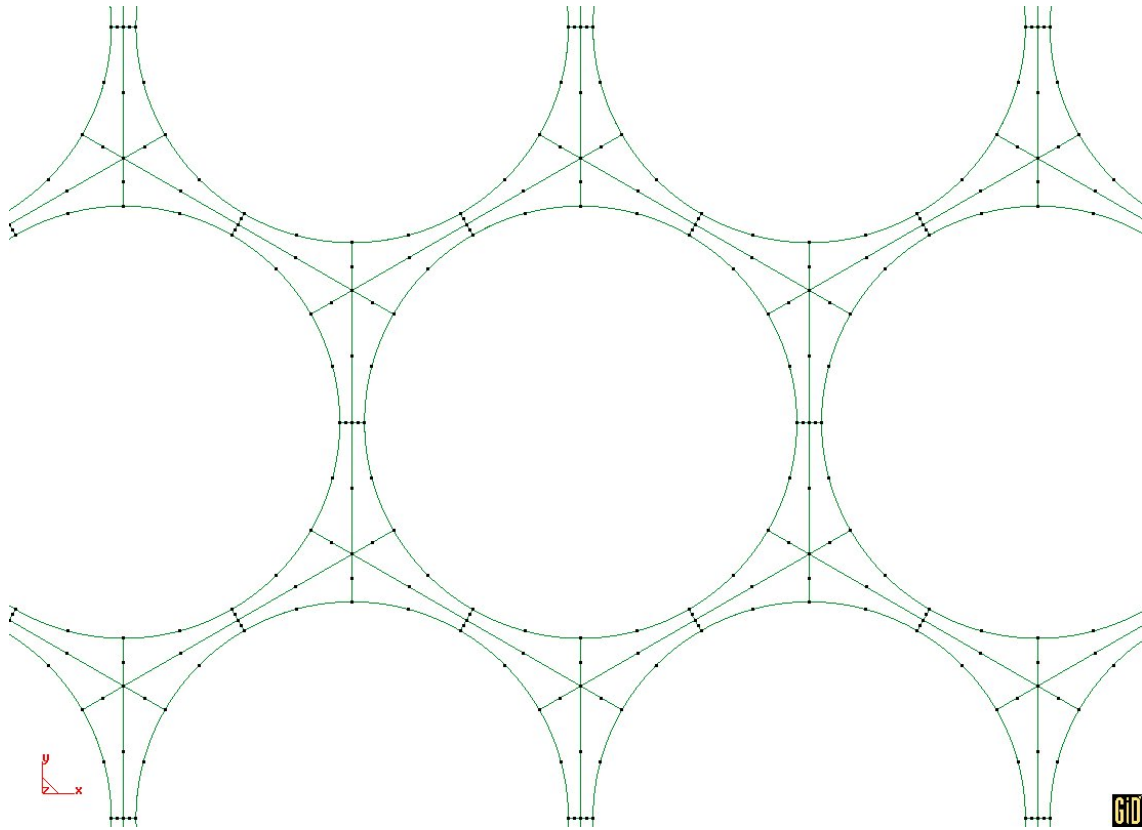


Figura III-130. De talle de la malla en torno a la celda circular

III.4.2.4.- Cálculo por elementos finitos

Calsef es un programa para calcular estructuras de sólidos por el método de elementos finitos, en un amplio rango de problemas: deformación y tensión plana, sólidos con simetrías de revolución, láminas, láminas de revolución, placas y sólidos tridimensionales.

En concreto, se empleará la versión Calsef2001 desarrollada expresamente para la resolución de cajones multicelulares incorporando ciertas especificidades propias de esta tipología estructural, como la presencia de cargas hidrostáticas. Dichas incorporaciones fueron validadas convenientemente.

Formulación General del MEF

La **Figura III-131** ilustra la división de una región en elementos finitos, en este caso elementos triangulares de 3 nodos.

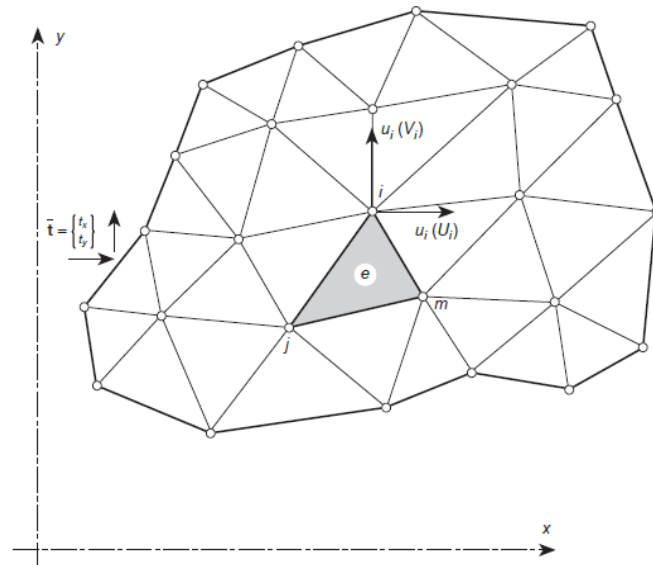


Figura III-131. Región plana dividida en elementos finitos [127]

En un elemento cualquiera de esta región e , definido por sus nodos i, j, m, \dots y por su contorno de líneas rectas, los desplazamientos \underline{u} de cualquier punto del elemento se aproximan mediante un vector $\hat{\underline{u}}$, siendo las componentes de \underline{N} funciones de posición dadas y \underline{a}^e un vector formado por los desplazamientos nodales del elemento considerado:

$$\underline{u} \approx \hat{\underline{u}} = \sum_k \underline{N}_k \cdot \underline{a}_k^e = \begin{bmatrix} \underline{N}_i & \underline{N}_j & \dots \end{bmatrix} \cdot \begin{Bmatrix} \underline{a}_i \\ \underline{a}_j \\ \vdots \\ \vdots \end{Bmatrix}^e = \underline{N} \cdot \underline{a}^e \quad \text{(III-69)}$$

Las llamadas funciones de forma N_i, N_j, N_m han de escogerse de manera que al sustituir en la ecuación (III-69) las coordenadas de los nodos se obtengan los correspondientes desplazamientos nodales: $\underline{N}_i(x_i, y_i, z_i) = \underline{I}$ y $\underline{N}_i(x_j, y_j, z_j) = \underline{0} \forall j \neq i$.

A partir de los desplazamientos para todos los puntos del elemento, pueden determinarse las deformaciones en cualquier punto. En forma vectorial, y utilizando un operador lineal apropiado \underline{S} que recoja las derivadas parciales de los desplazamientos, y la ecuación (III-69):

$$\begin{aligned} \underline{\varepsilon} &= \underline{S} \cdot \underline{u} \approx \underline{S} \cdot \underline{N} \cdot \underline{a} = \underline{B} \cdot \underline{a} \\ \underline{B} &= \underline{S} \cdot \underline{N} \end{aligned} \quad \text{(III-70)}$$

Nótese que si las funciones de forma siguen una expresión lineal, las deformaciones serán constantes en todo elemento.

Admitiendo un comportamiento elástico lineal del tipo más general, la relación entre tensiones y deformaciones será lineal. En general, el material puede estar sujeto a deformaciones iniciales $\underline{\varepsilon}_0$ y a un sistema conocido de tensiones residuales $\underline{\sigma}_0$. A partir de la matriz de elasticidad $\underline{\underline{D}}$ que contiene las propiedades del material apropiadas las tensiones serán:

$$\underline{\sigma} = \underline{\underline{D}} \cdot (\underline{\varepsilon} - \underline{\varepsilon}_0) + \underline{\sigma}_0 \quad \text{(III-71)}$$

Las fuerzas que actúan en los nodos \underline{q}^e , estáticamente equivalentes a las tensiones en el contorno y a las fuerzas distribuidas que actúan sobre el elemento \underline{b} , son:

$$\underline{q}^e = \begin{Bmatrix} \underline{q}_i^e \\ \underline{q}_j^e \\ \vdots \end{Bmatrix} \quad \text{(III-72)}$$

Para establecer la equivalencia estática entre las fuerzas nodales y las tensiones actuantes en el contorno y las fuerzas distribuidas se impone un desplazamiento virtual a los nodos $\delta \underline{a}^e$. Considerando las ecuaciones (III-69) y (III-70), los desplazamientos y deformaciones generados dentro del elemento por este desplazamiento son:

$$\begin{aligned} \delta \underline{u} &= \underline{\underline{N}} \cdot \delta \underline{a}^e \\ \delta \underline{\varepsilon} &= \underline{\underline{B}} \cdot \delta \underline{a}^e \end{aligned} \quad \text{(III-73)}$$

Igualando el trabajo efectuado por las fuerzas nodales, suma de los productos de las componentes de cada una de las fuerzas por sus correspondientes desplazamientos, con el trabajo interno efectuado por las tensiones y fuerzas distribuidas durante dicho desplazamiento, en el volumen del elemento:

$$(\delta \underline{a}^e)^T \cdot \underline{q}^e = \int_{V^e} ((\delta \underline{\varepsilon}^e)^T \cdot \underline{\sigma} - (\delta \underline{u})^T \cdot \underline{b}) \cdot dV = (\delta \underline{a}^e)^T \cdot \left(\int_{V^e} \underline{\underline{B}}^T \cdot \underline{\sigma} \cdot dV - \int_{V^e} \underline{\underline{N}}^T \cdot \underline{b} \cdot dV \right) \quad \text{(III-74)}$$

Como la relación es válida para cualquier desplazamiento virtual, se obtiene la siguiente expresión, válida con carácter general cualesquiera que sean las relaciones entre tensiones y deformaciones:

$$\underline{q}^e = \int_{V^e} \underline{\underline{B}}^T \cdot \underline{\sigma} \cdot dV - \int_{V^e} \underline{\underline{N}}^T \cdot \underline{b} \cdot dV \quad \text{(III-75)}$$

En el caso de la ley lineal de tensiones (III-71), la ecuación anterior se puede expresar como:

$$\underline{q}^e = \underline{\underline{K}}^e \cdot \underline{a}^e + \underline{f}^e \quad \text{(III-76)}$$

siendo la matriz de rigidez del elemento:

$$\underline{\underline{K}}^e = \int_{V^e} \underline{\underline{B}}^T \cdot \underline{\underline{D}} \cdot \underline{\underline{B}} \cdot dV \quad \text{(III-77)}$$

y el vector de fuerzas del elemento, debido a las fuerzas distribuidas, a fuerzas exteriores $\underline{\underline{t}}$ aplicadas en el contorno, a las deformaciones iniciales y a las tensiones iniciales, es:

$$\underline{\underline{f}}^e = -\int_{V^e} \underline{\underline{N}}^T \cdot \underline{\underline{b}} \cdot dV - \int_{A^e} \underline{\underline{N}}^T \cdot \underline{\underline{t}} \cdot dA - \int_{V^e} \underline{\underline{B}}^T \cdot \underline{\underline{D}} \cdot \underline{\underline{\epsilon}}_0 \cdot dV + \int_{V^e} \underline{\underline{B}}^T \cdot \underline{\underline{\sigma}}_0 \cdot dV \quad \text{(III-78)}$$

En general, podrá haber fuerzas externas concentradas actuando en los nodos y habrá de considerar el vector:

$$\underline{\underline{r}} = \begin{Bmatrix} r_1 \\ r_2 \\ \vdots \end{Bmatrix} \quad \text{(III-79)}$$

Imponiendo el equilibrio de fuerzas nodales en la malla, en cada nodo i debe verificarse que la suma de todas las fuerzas nodales de equilibrio de los elementos concurrentes al nodo sean igual al valor de las fuerzas exteriores sobre el nodo.

$$\underline{\underline{r}}_i = \sum_{e=1}^m \underline{\underline{q}}_i^e = \sum_{j=1}^n \left(\sum_{e=1}^m \underline{\underline{K}}_{ij}^e \right) \cdot \underline{\underline{a}}_j + \sum_{e=1}^m \underline{\underline{f}}_i^e \quad \text{(III-80)}$$

Por tanto, considerando todos los elementos:

$$\begin{aligned} \underline{\underline{K}}_{ij} &= \sum_{e=1}^m \underline{\underline{K}}_{ij}^e & \underline{\underline{f}}_i &= \sum_{e=1}^m \underline{\underline{f}}_i^e \\ \underline{\underline{K}}_{ij} \cdot \underline{\underline{a}} &= \underline{\underline{r}}_i - \underline{\underline{f}}_i \end{aligned} \quad \text{(III-81)}$$

Tras el ensamblaje se llega a la ecuación matricial global de la estructura, que relaciona la matriz de rigidez global con el vector de desplazamientos nodales y el de fuerzas nodales equivalentes:

$$\underline{\underline{K}} \cdot \underline{\underline{a}} = \underline{\underline{r}} - \underline{\underline{f}} \quad \text{(III-82)}$$

En general, el término libre de fuerzas exteriores menos fuerzas del elemento se suele agrupar por conveniencia en un único vector y escribir simplemente:

$$\underline{\underline{K}} \cdot \underline{\underline{a}} = \underline{\underline{f}} \quad \text{(III-83)}$$

La matriz de rigidez es singular, por lo que para resolver el sistema es preciso imponer condiciones de contorno, con valores prefijados en determinados

desplazamientos y fuerzas. Además, la matriz de rigidez es simétrica y definida positiva y, en general, se puede considerar una matriz dispersa para un número suficiente de elementos.

Métodos de resolución

Tras la imposición de las condiciones de contorno, se reordena el sistema para obtener en primera instancia los desplazamientos nodales desconocidos a_d a partir de las fuerzas prescritas f_p y los desplazamientos prescritos a_p :

$$\begin{bmatrix} K_{pd} & K_{pp} \\ K_{dd} & K_{dp} \end{bmatrix} \cdot \begin{bmatrix} a_d \\ a_p \end{bmatrix} = \begin{bmatrix} f_p \\ f_d \end{bmatrix} \Rightarrow a_d = K_{pd}^{-1} \cdot (f_p - K_{pp} \cdot a_p) \quad \text{(III-84)}$$

En este proceso es necesario invertir la submatriz resultante K_{pd} para lo cual se emplean métodos de resolución de sistemas lineales de ecuaciones directos o iterativos.

Tradicionalmente se utiliza el método directo de Gauss (pivotamiento y eliminación de filas hasta un sistema triangular) por su robustez, pero también existen otros métodos de factorización LU, y si teniendo en cuenta la simetría y estructura en banda de la matriz del sistema, métodos simplificados de descomposición de Crout $A = L \cdot D \cdot L'$, o de Cholesky $A = L \cdot L'$.

Estos métodos son precisos y exactos (si no existiera error de redondeo), pero lentos y costosos a medida que aumenta el tamaño de la matriz a invertir. El número de operaciones aritméticas realizadas al aplicar el algoritmo de eliminación de Gauss a un sistema $n \times n$ es $O(n^3)$.

Por ello, para grandes sistemas de ecuaciones (decenas o centenares de miles de ecuaciones), en los que la matriz es dispersa, se emplean métodos iterativos, que requieren de una solución inicial que se va refinando en una serie de iteraciones a medida que converge a una solución con un cierto margen de tolerancia.

En general los métodos iterativos más eficaces están basados en los subespacios de Krylov [129]. Para una matriz cuadrada $\underline{\underline{A}}$ y un vector $\underline{\underline{v}}$, es un subespacio vectorial de orden r generado por las sucesivas potencias de la matriz por el vector:

$$\kappa_r(\underline{\underline{A}}, \underline{\underline{v}}) = \langle \underline{\underline{v}}, \underline{\underline{A}} \cdot \underline{\underline{v}}, \dots, \underline{\underline{A}}^{r-1} \cdot \underline{\underline{v}} \rangle \quad \text{(III-85)}$$

Los métodos iterativos minimizan una cierta norma del vector residuo sobre un subespacio de Krylov generado por la matriz del sistema, con una convergencia suave y sin grandes fluctuaciones, a la vez de mantener un bajo coste computacional por iteración y bajos requisitos de almacenaje.

En sistemas de ecuaciones lineales con matriz de coeficientes simétrica y definida positiva, el algoritmo de Gradiente Conjugado GC alcanza, salvo errores de redondeo, la solución exacta en un número de iteraciones igual a la dimensión del sistema. Este método, al igual que otros como el método de Arnoldi FOM, el de

Mínimo Residuo Generalizado GMRES y el de Lanczos, pertenece a la familia de los métodos de ortogonalización al basarse en un proceso de proyección ortogonal sobre un subespacio de Krylov de dimensión menor que la del sistema.

Otras familias de métodos son las de biortogonalización (biortogonalización de Lanczos, Doble Gradiente Conjugado Bi-CG, *Conjugate Gradient Squared* CGS, *Biconjugate Gradient Stabilized* Bi-CGSTAB, Cuasi-mínimo Residuo QMR, *Transpose-Free* QMR, QMRCGSTAB) y los basados en la Ecuación Normal (Gradiente Conjugado para la Ecuación Normal CGN, *Least-Square* QR) [128].

El método del Gradiente Conjugado

A partir de la función cuadrática siguiente:

$$f(\underline{x}) = \frac{1}{2} \cdot \underline{x}^t \cdot \underline{A} \cdot \underline{x} - \underline{x}^t \cdot \underline{b} \quad (\text{III-86})$$

se demuestra que si la matriz \underline{A} es simétrica y definida positiva entonces la función se minimiza por la solución del sistema de ecuaciones: $\nabla f(\underline{x}) = 0 \rightarrow \underline{x}/\underline{A} \cdot \underline{x} = \underline{b}$, donde:

$$\nabla f(\underline{x}) = \begin{bmatrix} \frac{\partial f(\underline{x})}{\partial x_1} \\ \frac{\partial f(\underline{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\underline{x})}{\partial x_n} \end{bmatrix} \quad (\text{III-87})$$

El gradiente de una función cuadrática es un campo de vectores que, para un punto dado \underline{x} , apunta en la dirección del mayor incremento de $f(\underline{x})$ y es ortogonal a las curvas elipsoides obtenidas cortando el paraboloides $f(\underline{x})$ a valores constantes. En la **Figura III-132** se ilustran, para un caso sencillo [131] de un sistema de ecuaciones 2x2:

$$\begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ -8 \end{bmatrix}$$

cuya solución es $\underline{x} = \{2, -2\}^t$, tanto las curvas de contorno para valores crecientes de f como el campo de vectores del gradiente de f , siendo estos nulos precisamente en el punto correspondiente a la solución del sistema.

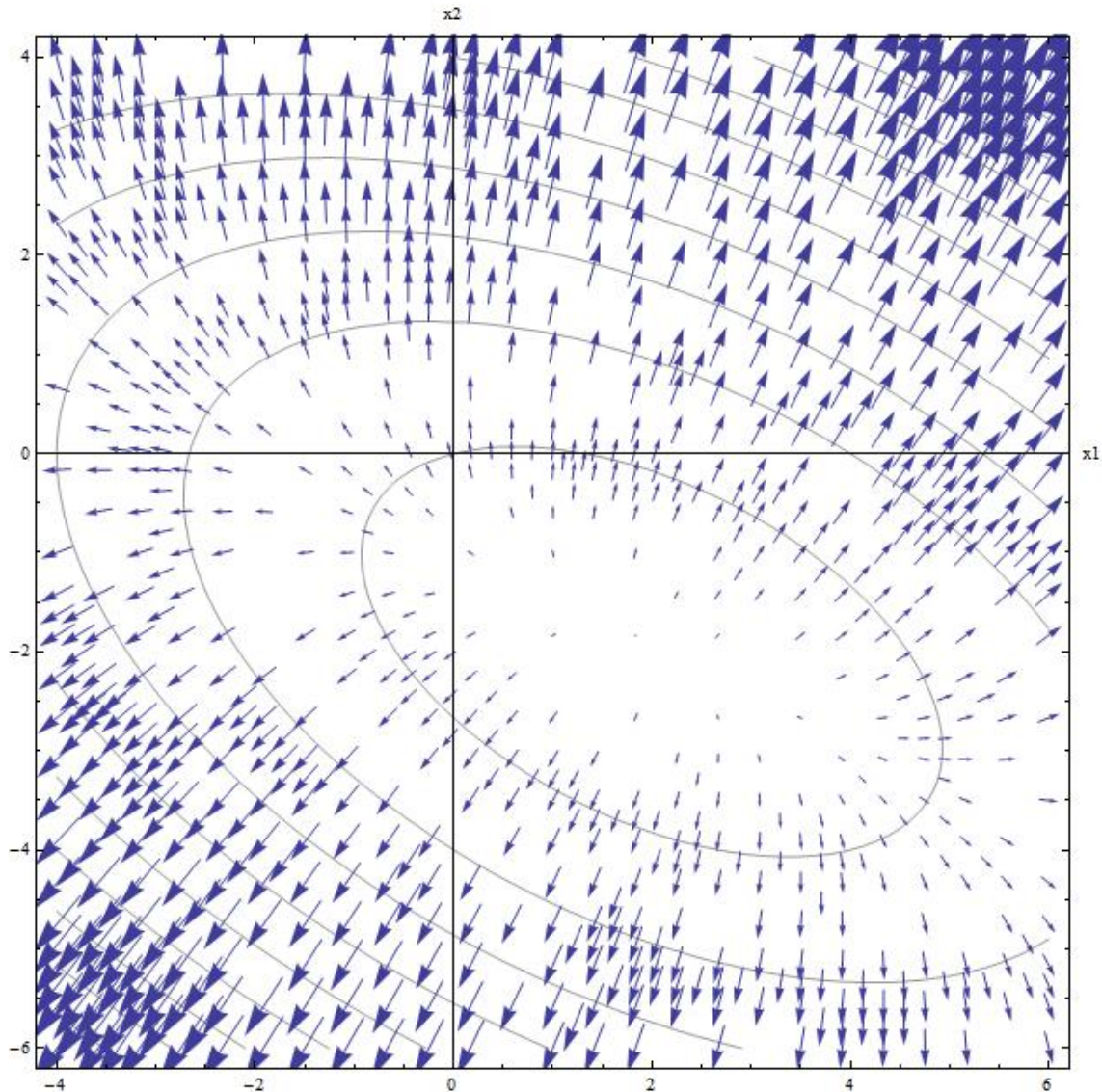


Figura III-132. Ejemplo de curvas de contorno y mapa de vectores para un sistema 2x2

Por tanto, haciendo el gradiente nulo se obtiene el sistema de ecuaciones que se desea resolver, lo que indica que si \underline{A} es simétrica y definida positiva, la solución \underline{x} del sistema $\underline{A} \cdot \underline{x} = \underline{b}$ es precisamente el mínimo de $f(\underline{x})$. Esto permite obtener la solución del sistema encontrando \underline{x} que haga mínimo $f(\underline{x})$.

Los métodos basados en algoritmos de descenso se basan en elegir una solución inicial \underline{x}_0 e ir bajando sucesivamente hasta el fondo del paraboloides, tomando en cada etapa la dirección en la cual $f(\underline{x})$ decrezca lo más rápido posible, la opuesta a la del gradiente en ese punto: $-\nabla f(\underline{x}_0)$. A esta dirección se le denomina residual.

$$\underline{r}_0 = -\nabla f(\underline{x}_0) = \underline{b} - \underline{A} \cdot \underline{x}_0 \quad \text{(III-88)}$$

El error en la aproximación será un vector que indique la distancia a la solución real:

$$\begin{aligned}\underline{e}_0 &= \underline{x}_0 - \underline{x} \\ \underline{r}_0 &= -\underline{A} \cdot \underline{e}_0\end{aligned}\tag{III-89}$$

Elegida la dirección hay que determinar el tamaño del paso a realizar. El punto siguiente será: $\underline{x}_1 = \underline{x}_0 + \alpha_0 \cdot \underline{r}_0$, por lo que se calculará el valor de α que minimice $f(\underline{x}_1)$:

$$\frac{df(\underline{x}_1)}{d\alpha} = 0 \Rightarrow \nabla f(\underline{x}_1)^t \cdot \underline{r}_0 = 0 \Rightarrow \nabla f(\underline{x}_1) \perp \underline{r}_0 \Rightarrow \underline{r}_1 \perp \underline{r}_0\tag{III-90}$$

Imponiendo la condición de ortogonalidad entre dos residuos consecutivos, el valor de α se obtiene a partir de la siguiente expresión:

$$\alpha_0 = \frac{\underline{r}_0^t \cdot \underline{r}_0}{\underline{r}_0^t \cdot \underline{A} \cdot \underline{r}_0}\tag{III-91}$$

Por tanto, el método de descenso, de lenta convergencia, consiste en:

$$\begin{aligned}\underline{r}_0 &= \underline{b} - \underline{A} \cdot \underline{x}_0 \\ i &= 1, \dots, n \\ \left\{ \begin{aligned} \alpha_i &= \frac{\underline{r}_i^t \cdot \underline{r}_i}{\underline{r}_i^t \cdot \underline{A} \cdot \underline{r}_i} \\ \underline{x}_{i+1} &= \underline{x}_i + \alpha_i \cdot \underline{r}_i \\ \underline{r}_{i+1} &= \underline{r}_i - \alpha_i \cdot \underline{A} \cdot \underline{r}_i \end{aligned} \right.\end{aligned}\tag{III-92}$$

Alternativamente, se toma un conjunto ortogonal de n direcciones de búsqueda, linealmente independientes entre sí, de tal manera que con cada dirección se dé exactamente un paso de la longitud precisa para alcanzar la solución \underline{x} . De este modo, después de n pasos se habrá alcanzado la solución. Así, cada paso será: $\underline{x}_{i+1} = \underline{x}_i + \alpha_i \cdot \underline{d}_i$. La condición de ortogonalidad de los vectores de búsqueda se realiza respecto de la matriz del sistema:

$$\underline{d}_{i+1}^t \cdot \underline{A} \cdot \underline{d}_i = 0\tag{III-93}$$

de modo que estas direcciones son linealmente independientes y A-ortogonales o conjugadas.

El subespacio i-dimensional dado por las direcciones de búsqueda es un subespacio de Krylov, al ser creado repetitivamente al aplicar una matriz a un vector: $\Delta_i = \langle \underline{d}_0, \underline{d}_1, \dots, \underline{d}_{i-1} \rangle = \langle \underline{d}_0, \underline{A} \cdot \underline{d}_0, \dots, \underline{A}^{i-1} \cdot \underline{d}_0 \rangle$. En cada iteración la dirección es tomada de este subespacio de manera que se minimiza el error $\|\underline{e}_i\|_A = \left(\underline{e}_i^t \cdot \underline{A} \cdot \underline{e}_i \right)^{1/2}$.

De este modo, el valor de α_i se obtiene ahora a partir de la ortogonalidad del vector error con la dirección elegida, de modo que no se vuelva a recorrer dicha dirección, llegando a la siguiente expresión:

$$\underline{e}_{i+1} \cdot \underline{d}_i = 0 \Rightarrow \alpha_i = \frac{\underline{d}_i^t \cdot \underline{r}_i}{\underline{d}_i^t \cdot \underline{A} \cdot \underline{d}_i} \quad (\text{III-94})$$

Y para el nuevo residuo:

$$\underline{r}_{i+1} = \underline{r}_i - \alpha_i \cdot \underline{A} \cdot \underline{d}_i \quad (\text{III-95})$$

Para encontrar el subespacio de direcciones A-ortogonales se recurre al proceso de ortogonalización de Gram-Schmidt a partir de un conjunto de vectores linealmente independientes, que son los residuos, de modo que:

$$\underline{d}_i = \underline{r}_i + \sum_{k=0}^{i-1} \beta_{ik} \cdot \underline{d}_k \rightarrow \beta_{ik} = 0 \quad \forall k \neq i-1 \rightarrow$$

$$\beta_i = \frac{\underline{r}_i^t \cdot \underline{r}_i}{\underline{r}_{i-1}^t \cdot \underline{r}_{i-1}} \quad (\text{III-96})$$

Como se ve, no es necesario guardar las direcciones de búsqueda anteriores, más allá de una iteración, por lo que el método del gradiente conjugado reduce notablemente el número de operaciones a realizar y el espacio de almacenamiento requerido.

Por tanto, el método del gradiente conjugado consiste en:

$$\begin{aligned} \underline{r}_0 &= \underline{b} - \underline{A} \cdot \underline{x}_0 \\ \underline{d}_0 &= \underline{r}_0 \\ i &= 1, \dots, n \\ \left\{ \begin{aligned} \alpha_i &= \frac{\underline{r}_i^t \cdot \underline{r}_i}{\underline{d}_i^t \cdot \underline{A} \cdot \underline{d}_i} \\ \underline{x}_{i+1} &= \underline{x}_i + \alpha_i \cdot \underline{d}_i \\ \underline{r}_{i+1} &= \underline{r}_i - \alpha_i \cdot \underline{A} \cdot \underline{d}_i \\ \beta_{i+1} &= \frac{\underline{r}_{i+1}^t \cdot \underline{r}_{i+1}}{\underline{r}_i^t \cdot \underline{r}_i} \\ \underline{d}_{i+1} &= \underline{r}_{i+1} + \beta_{i+1} \cdot \underline{d}_i \end{aligned} \right. \quad (\text{III-97}) \end{aligned}$$

La **Figura III-133** ilustra las dos iteraciones realizadas para alcanzar la solución del ejemplo anteriormente indicado, así como las direcciones de búsqueda empleadas.

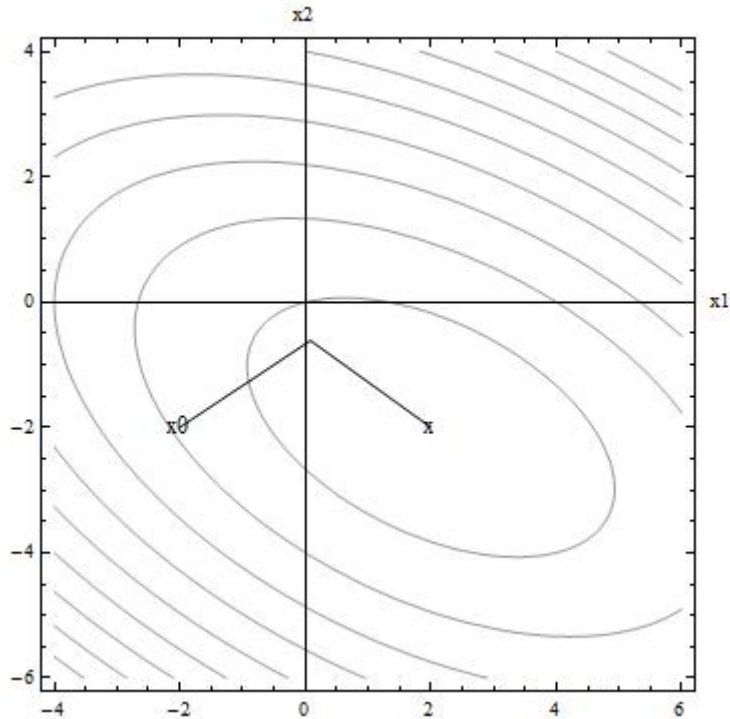


Figura III-133. Resolución por GC de un sistema 2x2

Aunque está garantizada la convergencia en n iteraciones (dimensiones del sistema), en grandes sistemas de ecuaciones no es viable realizar tal número de etapas, por lo que se introduce un criterio de parada a partir de una tolerancia en el residuo o un máximo número de iteraciones:

$$\|r_{i+1}\| < \varepsilon \cdot \|r_0\| \quad (\text{III-98})$$

Precondicionamiento

La convergencia de los métodos basados en los subespacios de Krylov mejora con el uso de las técnicas de precondicionamiento. El sistema original se cambia por otro de idéntica solución con mejor número de condicionamiento de la matriz. Si la matriz de precondicionamiento es $\underline{\underline{M}}^{-1}$, el precondicionamiento por la izquierda es:

$$\underline{\underline{A}} \cdot \underline{x} = \underline{b} \rightarrow \underline{\underline{M}}^{-1} \cdot \underline{\underline{A}} \cdot \underline{x} = \underline{\underline{M}}^{-1} \cdot \underline{b} \quad (\text{III-99})$$

Idealmente, $\underline{\underline{M}} = \underline{\underline{A}}$, ya que sería un sistema con el número de condición igual a la unidad. Por tanto la matriz de precondicionamiento debe ser similar a la del sistema, y fácilmente inversible. Los precondicionadores de Jacobi, Diagonal, SSOR e ILUT son los más empleados habitualmente.

El método de escalado diagonal de la matriz [130] se emplea para modificar el sistema original previamente al precondicionamiento y conseguir mejorar el número de condición de la matriz de coeficientes.

$$\begin{aligned}
 \underline{\underline{D}} &= \text{diag}(A) & D_{ii}^{-1/2} &= \frac{1}{\sqrt{|A_{ii}|}} \\
 \underline{\underline{\hat{A}}} &= \underline{\underline{D}}^{-1/2} \cdot \underline{\underline{A}} \cdot \underline{\underline{D}}^{-1/2} & \underline{\underline{\hat{x}}} &= \underline{\underline{D}}^{1/2} \cdot \underline{\underline{x}} & \underline{\underline{\hat{f}}} &= \underline{\underline{D}}^{-1/2} \cdot \underline{\underline{f}} \\
 & & \underline{\underline{\hat{A}}} \cdot \underline{\underline{\hat{x}}} &= \underline{\underline{\hat{b}}}
 \end{aligned}
 \tag{III-10}$$

Nótese que se toma como matriz $\underline{\underline{D}}$ la matriz diagonal de la propia matriz de coeficientes $\underline{\underline{A}}$, que es definida positiva (y todos sus valores propios son positivos, en consecuencia, por lo que $A_{ii} > 0$).

Continuando con el ejemplo anterior, la aplicación de este preconditionador diagonal mejoraría ligeramente las curvas de los elipsoides, haciéndolas más circulares. El número de condición baja de 3.5 a 2.8. La mejora se hace más notable cuanto más grande es n y cuanto peor sea el número de condición de partida.

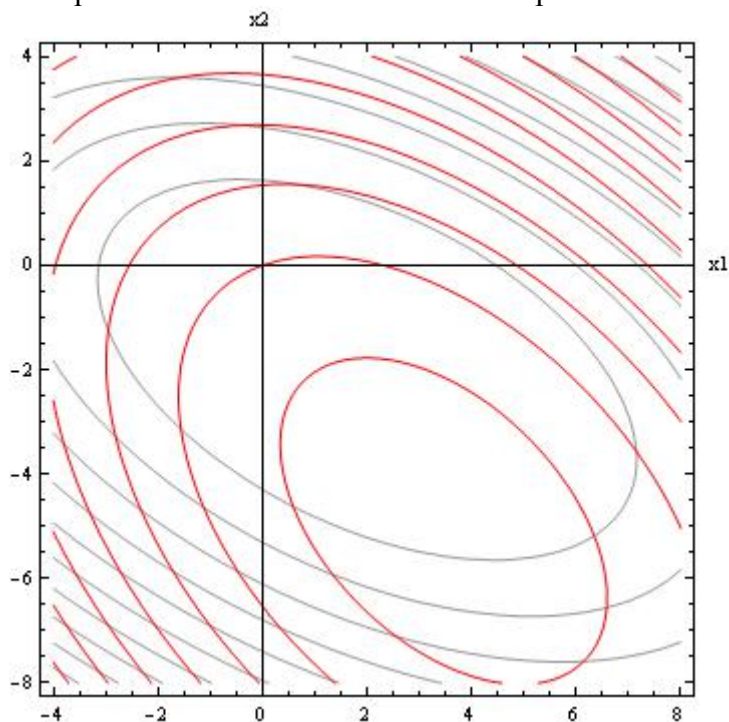


Figura III-134. Ligeramente mejoradas las curvas elipsoides tras el preconditionamiento diagonal

Por ello es necesario aplicar un segundo preconditionador a este sistema modificado, y una vez se resuelva el sistema con el método iterativo elegido, finalmente se deberá deshacer el escalado diagonal realizado.

Librería MatMan [132]

MatMan es una librería de subrutinas programadas en Fortran 77 para la resolución de los sistemas lineales de ecuaciones que aparecen en cualquier análisis por elementos finitos, desarrolladas por el Profesor Ramón Codina en CIMNE.

Incluye una amplia colección de métodos de resolución, tanto directos como iterativos. Entre los métodos directos está implementada la descomposición LDU, con o sin pivoteo. Entre los iterativos, se distinguen dos tipos principales, según que el

almacenamiento sea elemento a elemento (métodos de Gradiente Conjugado CG, y el método de Mínimo Residuo Generalizado GMRES) o sea disperso (diversos métodos, basados en la librería SPARSKIT [134] como Gradiente Conjugado CG, CGNR, Bi-CG,...).

La forma de uso de la librería es la llamada con una serie de parámetros y un número de tarea *itask*:

```
call MatMan(lnods,ifix,rhsid,unkno,ndofn,nnode,nnodt, &
&          nnodw,nelem,npoin,ntotv,nsist,lures,kfact, &
&          kimpo,iprob,intel,inte2,real1,itask)
```

Tabla III-13. Llamada genérica a la librería MatMan

El valor de ITASK puede ser:

- 1, para reenumerar las ecuaciones
- 2, para reservar la memoria de la matriz del sistema
- 3, para construir la matriz
- 4, para resolver el sistema lineal
- 5, para liberar la memoria empleada.

Los parámetros empleados son matrices, vectores, números enteros o reales:

- NNODT: Máximo número de nodos por elemento, incluyendo los nodos secundarios.
- NDOFN: Número de grados de libertad de los nodos principales.
- NNODE: Máximo número de nodos principales por elemento.
- NNODW: Número máximo de nodos activos por elemento.
- NPOIN: Número de nodos en la malla
- NTOTV: Número total de grados de libertad: NDOFN x NPOIN
- NELEM: Número de elementos.
- NSIST: Número de sistemas a resolver con la misma matriz y distinto vector.
- KFACT: Es un código para definir la factorización de la matriz. Si el sistema a resolver es $A \cdot x = f$, y x_p es el conjunto de grados de libertad prescritos, x_0 los desconocidos, A_p es la submatriz que multiplica a los primeros y A_0 la que multiplica a los segundos. Tras imponer condiciones de contorno, el sistema quedaría como $A_0 \cdot x_0 = f_0 - A_p \cdot x_p$. Con KFACT=0 se evalúa $A_p \cdot x_p$ pero no se factoriza la matriz A . Con KFACT=1 se evalúa $A_p \cdot x_p$ y se factoriza la matriz A . Con KFACT=2 se usan resultados previos de $A_p \cdot x_p$ y no se factoriza la matriz A .
- KIMPO: Código para prescribir las condiciones fijadas. Si es 0 el vector f viene ya modificado como entrada, si es 1 el vector se modifica internamente, restándole el término $A_p \cdot x_p$.
- IPROB: Número de problema
- IFFIX: Vector de condiciones fijadas, de dimensión NTOTV.
- RHSID: Vector f , de dimensiones NTOTV x NSIST.
- UNKNO: Vector de variables desconocidas, de dimensiones NTOTV x NSIST.

- LNODS: Matriz de conexiones nodales, de dimensiones NNODT x NELEM
- LURES: fichero de salida
- REAL1: Guarda en las tareas 1 y 4 CPUTI, el tiempo de cálculo. En la 3 guarda VALUE. En solvers directos son los valores a insertar en la matriz global, de dimensión INTE1. En iterativos, son los elementos de la matriz almacenados, de dimensión INTE1 x INTE1.
- INTE1: En la tarea 3 es NVALU para solver directo, número de valores a ensamblar, o NVALO en solvers iterativos, dimensión de la matriz a ensamblar. En la tarea 4, es ISTEP, el contador de pasos de tiempo, ó 0 si no se indica en el fichero de salida.
- INTE2: En la tarea 3 es IDEST para solver directo, destino de los valores a ensamblar de dimensión 2 x INTE1, o IELEM en solvers iterativos, elemento de la matriz almacenada en VALUE. En la tarea 4, es IITER, el contador de iteraciones, ó 0 si no se indica en el fichero de salida.

Además, es necesario seleccionar los parámetros que se van a emplear en el método de resolución:

- LENGP=8, longitud de punteros.
- KMSIP(IPROB) indica el modo de almacenamiento de la matriz y la estrategia de preconditionamiento. En métodos directos 0 es en banda, 1 es skyline, 2 es disperso. En métodos iterativos con almacenamiento elemento a elemento, -1 es sin preconditionamiento y -2 es con preconditionamiento de la diagonal. En métodos iterativos con almacenamiento disperso, 0 es sin preconditionamiento, 1 con preconditionamiento por la izquierda, 2 por la derecha, 3 por izquierda y derecha de la factorización LU incompleta, ILUT [135].
- KSYMM(IPROB) indica si la matriz es simétrica o no.
- KRENU(IPROB) indica si se renumera o no, en métodos directos.
- NMULT(IPROB) multiplicador de la dimensión de la matriz dispersa, siendo 3 el valor recomendado.
- KITES(IPROB) contiene un número identificativos del tipo de método a utilizar.
- ITMAX(IPROB) contiene el máximo número de iteraciones en métodos iterativos.
- TOLER(IPROB) contiene la tolerancia en métodos iterativos.
- KKRYL(IPROB) contiene el subespacio de Krylov en el caso de usar el método GMRES.
- THRES(IPROB) contiene el umbral para los preconditionadores ILUT.
- LFILL(IPROB) contiene el parámetro de relleno para los preconditionadores ILUT.
- WOSOL(IPROB) contiene el título del problema
- CONDE(IPROB) indica si hay condensación o no de grados de libertad de nodos internos.

Secuencia de cálculo

Una vez que la malla ha sido generada en GiD, se procede a llamar al Calsef para resolver la hipótesis de cálculo planteada.

```
"<ruta Calsef>\CALSEF2001.exe" "<ruta proyecto>\proyecto.gid\proyecto.cal"
```

Tabla III-14. Orden de generación de mallado en línea de comandos

La llamada al programa de análisis se ejecuta en una serie de etapas que se describen a continuación.

1. Lectura/datos.f90

A través del Problem Type, Calsef convertirá la información generada en GiD a un formato que pueda procesar. Para ello usa el script Calsef2001_Solidos_3D.bas y, a partir del fichero *.cal escrito al acabar el mallado en GiD, genera el fichero *.sal que contiene toda la información del problema previa a su resolución.

En la rutina datos.f90 se realiza el procesado del script.

```
DO WHILE (.NOT.done)
  CALL listen('datos')           ! Lectura línea a línea
  comand = words(1)             !buscando comandos

  SELECT CASE (comand)
  CASE ('CONTRO')               ! Lectura de
    CALL contol                 ! parametros
    contfl = .true.             ! de control
  CASE ('GEOMET')              ! Lectura de las
    CALL rdcoor                 ! coordenadas
    geomfl = .true.             ! nodales
  CASE ('SET ')                 ! Lectura de los
    CALL rdset(nsets)           ! set de elementos
  CASE ('MATERI')              ! Lectura de los
    IF (first) &                ! materiales
&    CALL ini_mate_typ(head,tail)
    first = .false.
    CALL rdmat
  CASE ('BOUNDA')              ! Lectura de las
    CALL rdbou                  ! condiciones de contorno
  CASE ('LOAD ')               ! Lectura de las
    CALL rdloa(npnod)           ! condiciones de carga
  CASE ('END ')
    IF (exists('DATA ')) &
&    done = .true.
  END SELECT
ENDDO
```

Figura III-135. Rutina principal en datos.f90

La siguiente tabla muestra un ejemplo de fichero *.sal generado.

```
Leyendo los datos del problema...

Indicador de escritura ..... IWRIT = 1

  D A T O S   G E N E R A L E S
=====

Indicador de escritura ..... 1
0: SIN escritura 1: CON escritura
```

3. Los Cajones Multicelulares

```

Titulo del Problema =Cajon Multicelular

Maximo numero de g.l. por nodo ..... 3

Escala de las coordenadas ..... SCALE = 1.0000

DEFINICION G E O M E T R I C A

      Nodo      Indice      X      Y      Z      Alfa      Beta      Gamma
      1          1      40.27800 -1.00000 0.00000
      2          2      40.27800 -0.75000 0.00000
      3          3      40.02200 -1.00000 0.00000
...
258132      258132      -1.02300 18.45800 17.50000
258133      258133      -1.02300 18.50000 17.50000

SET NAME 6 CHAR ONLY ..... NAME = SET1
ELEMENT TYPE ..... TYPE = STHE20
MATERIAL NUMBER ..... NUMBER = 1
SET DE ELEMENTOS STHE20 *****

Nombre del set ..... SET1
Numero de elementos en el set .. 48088
Numero del material en el set .. 1
Numero de nodos del elemento ... 20

Elem Num      CONECTIVIDADES

      1          197040      194879      193025      193003      193000      194856      197015
197023      196913      192016      191982      196894      196794      194086      190866      190845
190837      194053      196763      196769
      2          201660      199368      197040      197023      197015      199346      201645
201652      201774      196913      196894      201752      201897      199434      196794      196769
196763      199405      201871      201880

      48088      107083      108404      109823      108917      108039      106622      105268
106147      107726      110080      108262      105955      108433      109350      110289      109397
108503      107551      106650      107541

Numero del material ..... NUMBER = 1
Modulo de Young ..... YOUNG = 0.30546E+11
Modulo de Poisson ..... POISSO = 0.20000
Peso Especifico ..... SPECIF = 33109.

N O D O S R E S T R I N G I D O S Y R E S T R I C C I O N E S

      Nodo      Tipo de g.l.      Codigo      Valores Prescritos
      1          DISP-X      DISP-Y      DISP-Z      0 0 1      0.00000      0.00000      0.00000
      2          DISP-X      DISP-Y      DISP-Z      0 0 1      0.00000      0.00000      0.00000
      3          DISP-X      DISP-Y      DISP-Z      0 0 1      0.00000      0.00000      0.00000

      255258      DISP-X      DISP-Y      DISP-Z      0 0 1      0.00000      0.00000      0.00000

C A R G A R E P A R T I D A S O B R E L A S C A R A S

      Lado      CARGA

      193025      195180      197196      198121      198998      198034      197040      194879
0.00000      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000
0.00000      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000
0.00000      0.00000      0.00000      0.00000      0.00000      0.00000      *****
*****
      197040      198034      198998      200171      201268      201436      201660      199368
0.00000      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000
0.00000      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000
0.00000      0.00000      0.00000      0.00000      0.00000      0.00000      *****
*****

...
C A R G A H I D R O S T A T I C A S O B R E L A S C A R A S

      Lado      CARGA

      214921      214899      214894      214994      215118      215132      215153      215028
*****
*****
      214679      214644      214637      214762      214894      214899      214921      214799
*****
*****

...
P E S O P R O P I O

A C T I V A D O

```

Tabla III-15. Ejemplo de fichero *.sal para resolución en Calsef

2. Renumeración/renume.f90

Durante la lectura de los nodos de la malla en la fase anterior se determina la necesidad de realizar renumeración de los nodos en el caso de que las etiquetas de los nodos no estén numeradas correlativamente.

Aunque no es lo habitual, se utilizaría un algoritmo de renumeración por partición [133].

3. Solver/calgral.f90

Se inicializa el vector de grados de libertad por nodo.

4. Solver/ini_matman.f90

Se inicializa la librería MatMan. En primer lugar se inicializan la matriz de conexiones nodales LNODES, y los vectores IFFIX, RHSID y UNKNO.

Se fijan como parámetro NSIST=1, KFACT=1 al emplear un método iterativo, KIMPO=1, IPROB=1.

En la rutina matmanGCPD, con el número total de grados de libertad activos NTOTV, se seleccionan los parámetros que se van a emplear en el método de resolución iterativo:

- LENGP=8, longitud de punteros.
- KMSIP(1) = 1 con preconditionamiento
- KSYMM(1) = 1, matriz simétrica
- KRENU(1) = 1.
- NMULT(1) = 3.
- KITES(1) = **31 Gradiente Conjugado, método de almacenamiento disperso.**
- ITMAX(1) NTOTV.
- TOLER(1) = 10e-06.
- KKRYL(1) = 50, aunque no se usa.
- THRES(1) = 0.
- LFILL(1) = 0.
- CONDE(1) = falso

5. Control/eltask.f90

Se calcula y ensambla la matriz de rigidez, conforme al tipo de elemento seleccionado: rigima_STHE08.f90 o rigima_STHE20.f90.

Se emplea la matriz constitutiva $\underline{\underline{D}}$ para un material isótropo:

$$D = \frac{E}{(1+\nu) \cdot (1-2 \cdot \nu)} \cdot \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ & 1-\nu & \nu & 0 & 0 & 0 \\ & & 1-\nu & 0 & 0 & 0 \\ & & & \frac{1-2 \cdot \nu}{2} & 0 & 0 \\ & \text{Sím} & & & \frac{1-2 \cdot \nu}{2} & 0 \\ & & & & & \frac{1-2 \cdot \nu}{2} \end{bmatrix}$$

Se emplean 3 puntos de Gauss para la integración numérica en el volumen del elemento. De esta manera cualquier integral a realizar se puede aproximar de la siguiente regla de cuadratura, exacta para polinomios de grado 5 o inferior:

$$\int_{-1}^1 f(x) dx \approx \sum_{i=1}^{n_p} \omega_i \cdot f(x_i) \tag{III-102}$$

$$n_p = 3 \Rightarrow \begin{cases} x_1 = -\sqrt{3/5} & x_2 = 0 & x_3 = \sqrt{3/5} \\ \omega_1 = 5/9 & \omega_2 = 8/9 & \omega_3 = 5/9 \end{cases}$$

Se definen las funciones de forma para los 8 ó 20 nodos en coordenadas normalizadas, de modo que para un elemento paralelepípedo de lado 2 con ejes $\{\xi, \eta, \zeta\}$ en el centro, estas funciones son:

$$N_i = \frac{1}{8} \cdot (1 + \xi \cdot \xi_i) \cdot (1 + \eta \cdot \eta_i) \cdot (1 + \zeta \cdot \zeta_i) \quad \begin{cases} \xi_i = -1,1 \\ \eta_i = -1,1 \\ \zeta_i = -1,1 \end{cases} \tag{III-103}$$

Mientras que para 20 nodos se definen las funciones de forma de los vértices y de los nodos centrales:

$$N_i = \frac{1}{8} \cdot (1 + \xi \cdot \xi_i) \cdot (1 + \eta \cdot \eta_i) \cdot (1 + \zeta \cdot \zeta_i) \cdot (\xi \cdot \xi_i + \eta \cdot \eta_i + \zeta \cdot \zeta_i - 2) \quad \begin{cases} \xi_i = -1,1 \\ \eta_i = -1,1 \\ \zeta_i = -1,1 \end{cases}$$

$$N_i = \frac{1}{4} \cdot (1 - \xi^2) \cdot (1 + \eta \cdot \eta_i) \cdot (1 + \zeta \cdot \zeta_i) \quad \begin{cases} \xi_i = 0 \\ \eta_i = -1,1 \\ \zeta_i = -1,1 \end{cases} \tag{III-104}$$

$$N_i = \frac{1}{4} \cdot (1 + \xi \cdot \xi_i) \cdot (1 - \eta^2) \cdot (1 + \zeta \cdot \zeta_i) \quad \begin{cases} \xi_i = -1,1 \\ \eta_i = 0 \\ \zeta_i = -1,1 \end{cases}$$

$$N_i = \frac{1}{4} \cdot (1 + \xi \cdot \xi_i) \cdot (1 + \eta \cdot \eta_i) \cdot (1 - \zeta^2) \quad \begin{cases} \xi_i = -1,1 \\ \eta_i = -1,1 \\ \zeta_i = 0 \end{cases}$$

6. Lectura/ela_sup.f90

Se calculan y ensamblan los apoyos elásticos que se disponen únicamente en la hipótesis de transporte.

7. Cargas/General/fuerza.f90

Se calculan los vectores de fuerzas nodales.

8. Solver/solver.f90

Se procede a la resolución del sistema de ecuaciones.

9. Escritura/wricor.f90

Escribe a fichero los nodos de la malla.

10. Control/eltask.f90 'WRIMES'

Escribe a fichero las conectividades de los elementos.

11. Escritura/wrires.f90

Tras la resolución, se generan los ficheros *.tim, *.sal, *.flavia.msh y *.flavia.res.

La siguiente tabla recoge la parte final del contenido generado al resolver el cajón ejemplo en el fichero *.sal, en la que se reflejan los parámetro relativos al número de ecuaciones resueltas, a la memoria RAM empleada y al tiempo de cálculo requerido, entre otros.

```

P E S O   P R O P I O

  A C T I V A D O

Calculando grados de libertad nodales...
Inicializando la matriz de rigidez global...

#####
>>>> PERMANENT MEMORY INFORMATION FOR:  THERMAL
-----

*** SOLVER TYPE           : SK_CG METHOD           (DIAG.  PREC.)
*** NUMBER OF D.O.F.     :          774399
*** NUMBER OF EQUATIONS  :          763794
*** KRYLOV DIMENSION     :             50
*** NONZERO MATRIX COEFFICIENTS : 108687256
*** TOTAL PERMANENT MEMORY : 1335.197 (Mb)

Calculando y ensamblando las matrices de rigidez elementales ...
Calculando y ensamblando el vector de fuerzas ...
Resolviendo el sistema de ecuaciones ...

```

3. Los Cajones Multicelulares

```
Escribiendo los datos de la malla ...
Escribiendo los resultados obtenidos ...
El proceso ha finalizado ...
Tiempo total de ejecucion en segundos: 0.59930E+04
```

Tabla III-16. Ejemplo de fichero *.sal tras resolución en Calsef

La siguiente tabla recoge parte del contenido del fichero .flavia.res generado.

```
Gid Post Results File 1.0
#CALSEF 2001 by Dr. Francisco Zarate and Ing. Daniel Di Capua
Result "Desplazamientos" "Análisis_de_carga" 1 Vector OnNodes
ComponentNames "Desp-X" "Desp-Y" "Desp-Z"
Values
    1 -0.2223815E-03 0.1590599E-03 0.0000000E+00
    2 -0.2248182E-03 0.1577794E-03 0.0000000E+00
    3 -0.2211680E-03 0.1564480E-03 0.0000000E+00
...
    254965 0.0000000E+00 0.0000000E+00 0.1994300E+05
    255258 0.0000000E+00 0.0000000E+00 -0.4974232E+04
End Values
Result "Tensiones_SET1" "Análisis_de_carga" 1 Matrix OnNodes
ComponentNames "Tens-X" "Tens-Y" "Tens-Z" "Tens-XY" "Tens-YZ" "Tens-XZ"
Values
    1 -0.2415887E+06 -0.2434933E+06 -0.2370565E+06 -0.2078001E+04
0.5198567E+03 -0.1204021E+04
    2 -0.2344181E+06 -0.2398058E+06 -0.2330043E+06 0.1843956E+04
0.6833530E+04 0.1610483E+04
    3 -0.2273161E+06 -0.2327004E+06 -0.2344709E+06 0.2647364E+04
0.9443187E+03 -0.1354282E+04
...
    258131 -0.1403733E+04 -0.6310271E+03 -0.2432875E+04 0.7802416E+03 -
0.7507780E+02 -0.3182897E+03
    258132 -0.1776871E+04 -0.2431387E+03 -0.2378952E+04 0.4110960E+03 -
0.1132894E+03 -0.2485226E+03
    258133 -0.1397642E+04 -0.2741078E+03 -0.2038677E+04 0.3585304E+02 -
0.2931571E+02 -0.1854336E+03
End Values
```

Tabla III-17. Ejemplo de fichero *.flavia.res tras resolución en Calsef

12. Control/eltask.f90 'FLUJON'

Calcula las tensiones suavizadas.

13. Lectura/closfile.f90

Cierra los ficheros y finaliza el proceso.

Post-proceso

Se puede realizar un fichero por lotes para la generación automática de imágenes del proceso. En dicho fichero se indican uno por uno los resultados a mostrar sobre el cajón y la vista elegida para hacer la captura de imagen, así como el fichero en el que se guardará. Es posible realizar cortes de la malla, para lo cual hay que seleccionar tres puntos del cajón que definen los planos de corte.

```
escape escape escape escape View Zoom points -19.6275 27.75 0.000000 58.88250000000001
-9.25 0.000000 escape
escape escape escape escape Geometry escape
escape escape escape escape View Rotate Angle 270 90
escape escape escape escape View Hardcopy gif C:/Temp/GifTmp/Fig_Geometria_Planta.gif
escape
escape escape escape escape View Rotate Angle 45 45
escape escape escape escape View Hardcopy gif
C:/Temp/GifTmp/Fig_Geometria_Isometrico.gif escape
escape escape escape escape Meshing MeshView
escape escape escape escape View Rotate Angle 270 90
escape escape escape escape View Hardcopy gif C:/Temp/GifTmp/Fig Malla Planta.gif
```

```

escape
escape escape escape escape View Rotate Angle 45 45
escape escape escape escape View Hardcopy gif C:/Temp/GifTmp/Fig_Malla_Isometrico.gif
escape
escape escape escape escape Postprocess
escape escape escape escape displaystyle bodybound
escape escape escape escape results geometry original geometry deformation
Desplazamientos |Desplazamientos|
escape escape escape escape View Rotate Angle 45 45
escape escape escape escape View Hardcopy gif
C:/Temp/GifTmp/Fig_Deformada_Isometrico.gif escape
escape escape escape results analysissel Analisis_de_carga 1 ContourFill
Desplazamientos |Desplazamientos|
escape escape escape escape View Rotate Angle 45 45
escape escape escape escape View Hardcopy gif
C:/Temp/GifTmp/Fig_Desp_Res_Isometrico.gif escape
escape escape escape escape Results Geometry NoResults Geometry Original
escape escape escape escape results analysissel Analisis_de_carga 1 ContourFill
Tensiones_SET1 Tens-X
escape escape escape escape View Rotate Angle 45 45
escape escape escape escape View Hardcopy gif C:/Temp/GifTmp/Fig_Tens_X_Isometrico.gif
escape
escape escape escape escape results analysissel Analisis_de_carga 1 ContourFill
Tensiones_SET1 Tens-XY
escape escape escape escape View Rotate Angle 45 45
escape escape escape escape View Hardcopy gif C:/Temp/GifTmp/Fig_Tens_XY_Isometrico.gif
escape
escape escape escape escape results analysissel Analisis_de_carga 1 ContourFill
Tensiones_SET1 Tens-XZ
escape escape escape escape View Rotate Angle 45 45
escape escape escape escape View Hardcopy gif C:/Temp/GifTmp/Fig_Tens_XZ_Isometrico.gif
escape
escape escape escape escape results analysissel Analisis_de_carga 1 ContourFill
Tensiones_SET1 Tens-YZ
escape escape escape escape View Rotate Angle 45 45
escape escape escape escape View Hardcopy gif C:/Temp/GifTmp/Fig_Tens_YZ_Isometrico.gif
escape
escape escape escape escape results analysissel Analisis_de_carga 1 ContourFill
Tensiones_SET1 Si-Tensiones_SET1___
escape escape escape escape View Rotate Angle 270 0
escape escape escape escape DoCut CutPlane ThreePoints
0.0 0.0 0.275
19.6275 0.0 0.275
19.6275 18.5 0.275
escape escape escape escape Select CutSets NO_CUTSETS escape
escape escape escape escape Select VolumeSets NO_VOLUMESETS escape
escape escape escape escape Select CutSets C_CutSet_1_VolumeSet_1 escape
escape escape escape escape View Rotate Angle 270 90
escape escape escape escape View Hardcopy gif
C:/Temp/GifTmp/Fig_Tens_Si_CortePlanta_losa.gif escape
...
escape escape escape escape Select CutSets NO_CUTSETS escape
escape escape escape escape Select VolumeSets ALL_VOLUMESETS escape
escape escape escape escape View Rotate Angle 270 90
escape escape escape escape DoCut CutPlane ThreePoints
15.9175 0.0 0.0
15.9175 18.5 0.0
15.9175 18.5 17.2
escape escape escape escape Select CutSets NO_CUTSETS escape
escape escape escape escape Select VolumeSets NO_VOLUMESETS escape
escape escape escape escape Select CutSets C_CutSet_9_VolumeSet_1 escape
escape escape escape escape View Rotate Angle 0 0
escape escape escape escape View Hardcopy gif
C:/Temp/GifTmp/Fig_Tens_Si_CortePerfil_centro2.gif escape
escape escape escape escape Select CutSets NO_CUTSETS escape
escape escape escape escape Select VolumeSets ALL_VOLUMESETS escape
escape escape escape escape quit

```

Tabla III-18. Fichero por lotes para generación de imágenes en post-proceso en GiD

De tal modo que en línea de comandos se puede realizar la llamada a GiD para que procese el fichero y genere las imágenes en la ruta especificada. En este caso, es preciso indicar la opción '+g' para que se muestre el programa y se puedan realizar las capturas de pantalla.

```

"<ruta GiD>\gid.exe" -b+g "<ruta proyecto>\proyecto.gid\proyectoGif.bat" -n
"<ruta proyecto>\proyecto.gid"

```

Tabla III-19. Orden de generación de imágenes en línea de comandos

3. Los Cajones Multicelulares

Las siguientes figuras ilustran sobre el cajón los resultados obtenidos, empleando las opciones de post-procesado de GiD.

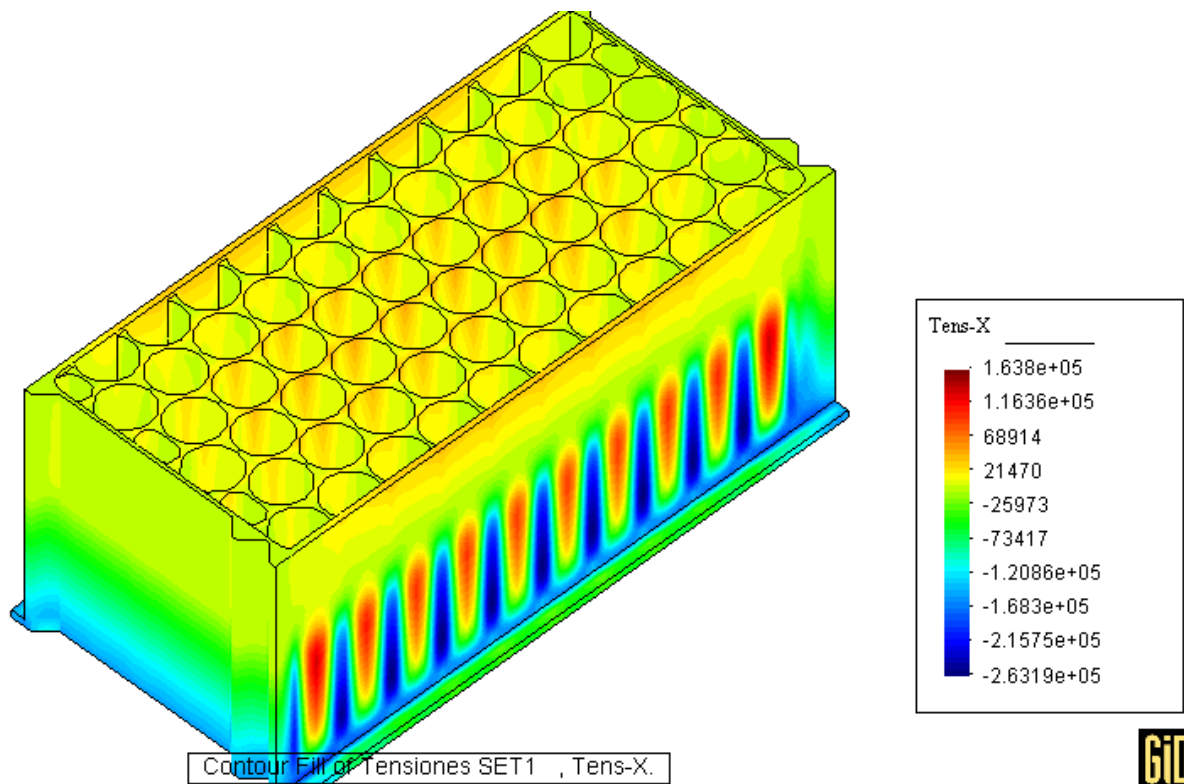


Tabla III-20, Resultado de tensiones normales en el cajón en una hipótesis de cálculo

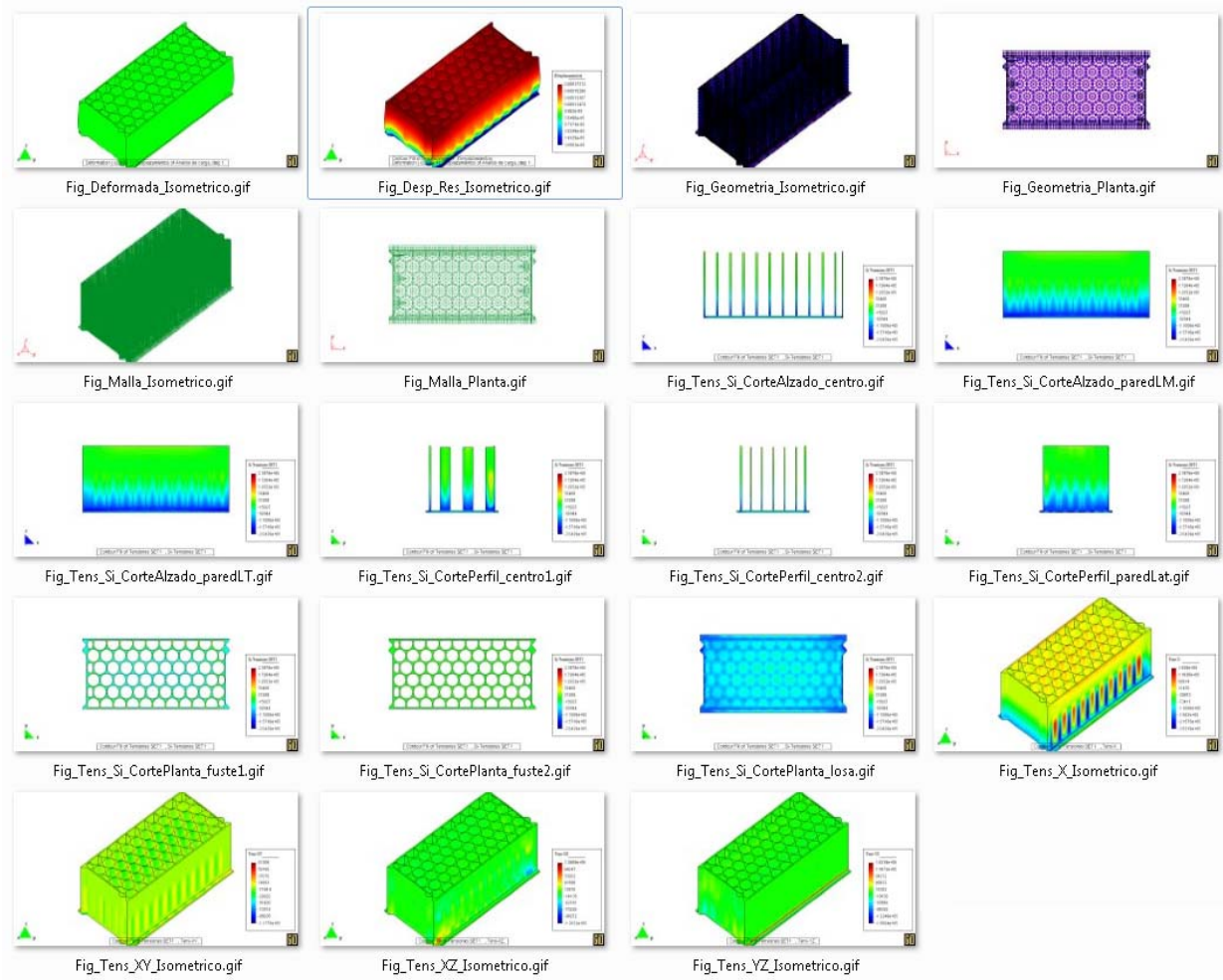


Tabla III-21. Colección de imágenes de postproceso generadas automáticamente en GiD

III.5.- Generación de esfuerzos en secciones críticas

Una vez que se ha realizado el análisis tridimensional de la estructura bajo diversas situaciones de carga mediante el método de los elementos finitos, los resultados que se obtienen directamente son desplazamientos y tensiones.

A fin de realizar el armado de las distintas paredes del cajón es imprescindible calcular los esfuerzos equivalentes a dichas tensiones en las diferentes zonas de armado. En concreto, se buscarán los momentos flectores y esfuerzos axiales con los que determinar el armado longitudinal horizontal y vertical.

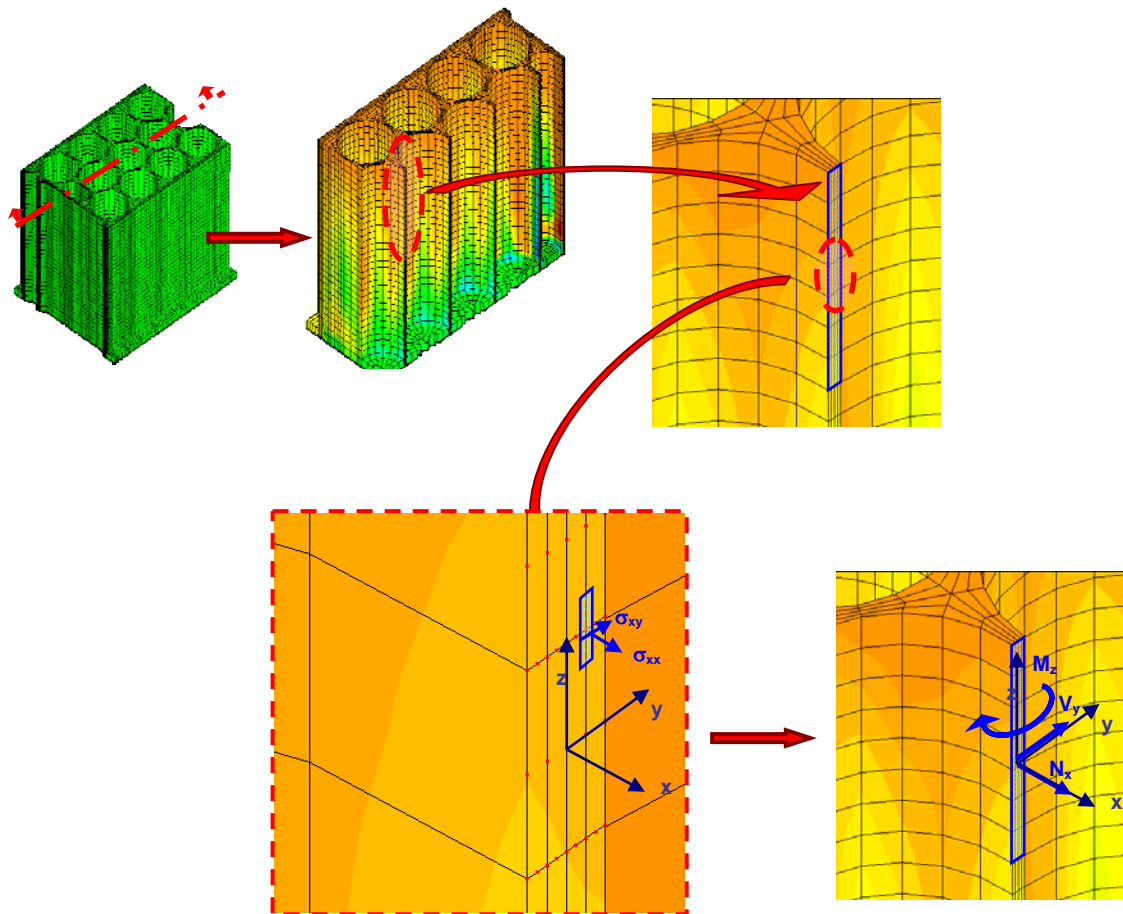


Figura III-136. Planteamiento general de obtención de esfuerzos en paredes verticales

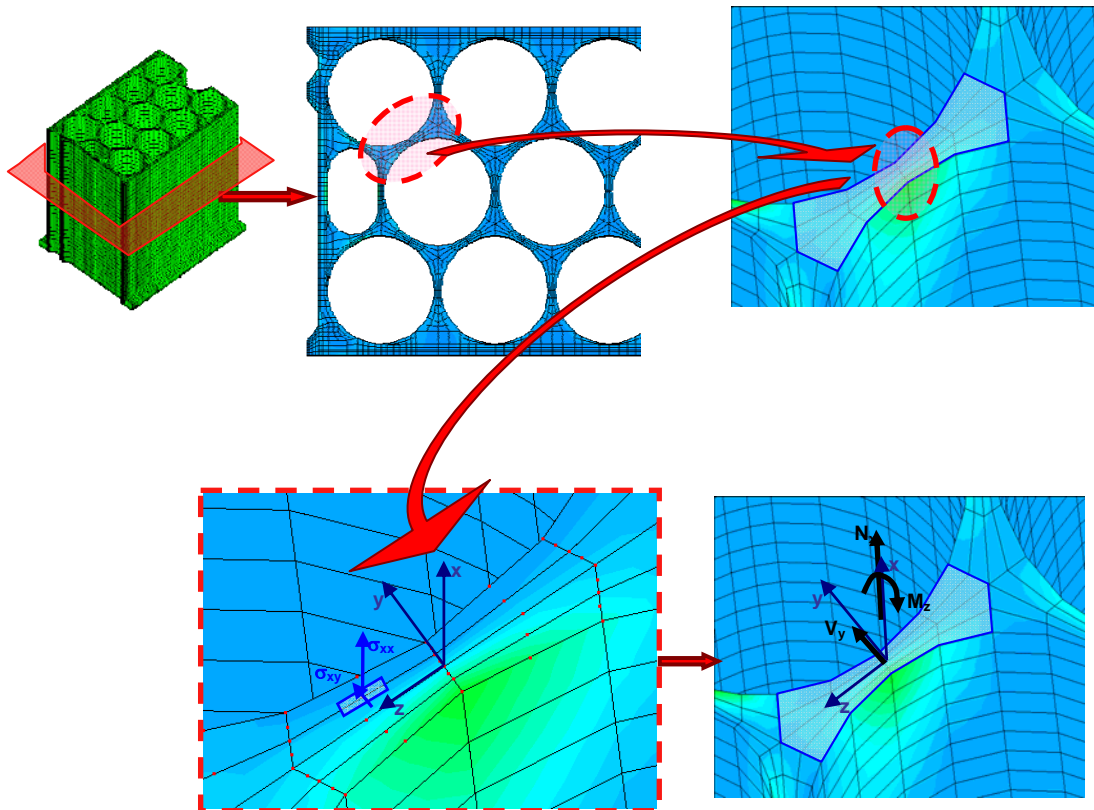


Figura III-137. Planteamiento general de obtención de esfuerzos en paredes horizontales

III.5.1.- Generación de gargantas

Finalizada la generación de la geometría, en cada tipología de celda se han definido las posiciones de las gargantas, por lo que mediante un algoritmo que recorre todas y cada una de las celdas dispuestas en la sección se seleccionan dichas gargantas y se escriben a ficheros.

Para ello se emplean las siguientes clases que contienen la información precisa para generar cada tipo de garganta.

Las clases **CGargantaVertical** y **CColGargantasVerticales**

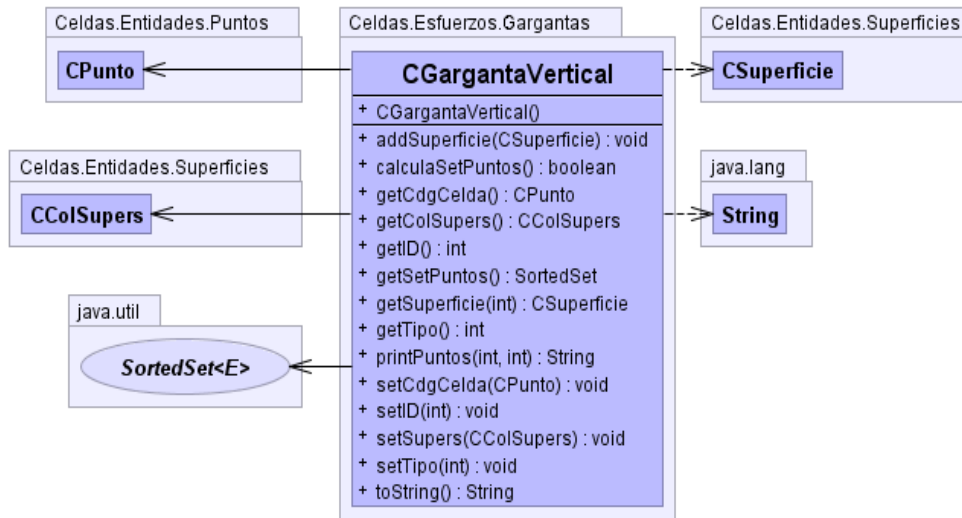


Figura III-138. Diagrama UML de la clase **CGargantaVertical**

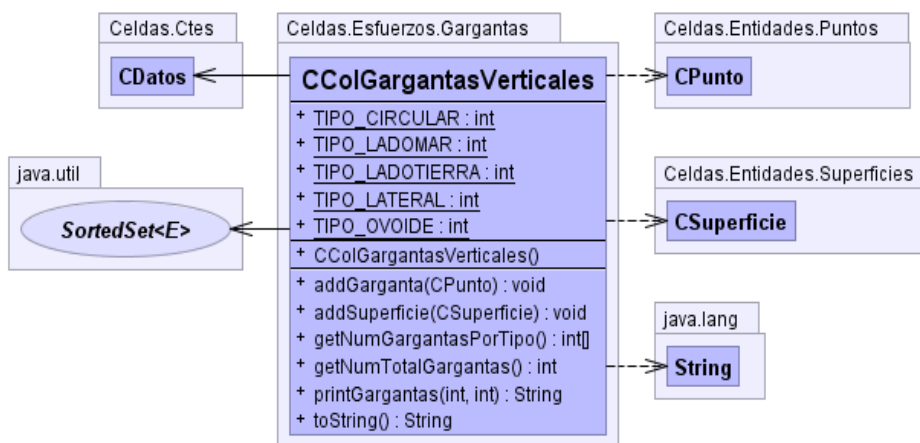


Figura III-139. Diagrama UML de la clase **CColGargantasVerticales**

Para determinar las gargantas verticales se recorren todas las superficies verticales de cada uno de los cubos generados en cada una de las alturas, seleccionando aquéllas que fueron marcadas como superficie con garganta (**Figura III-60**). La garganta queda definida por el punto central en la base del fuste y por todas las superficies de las diferentes alturas (una o garganta simple, dos o garganta doble, por rebanada).

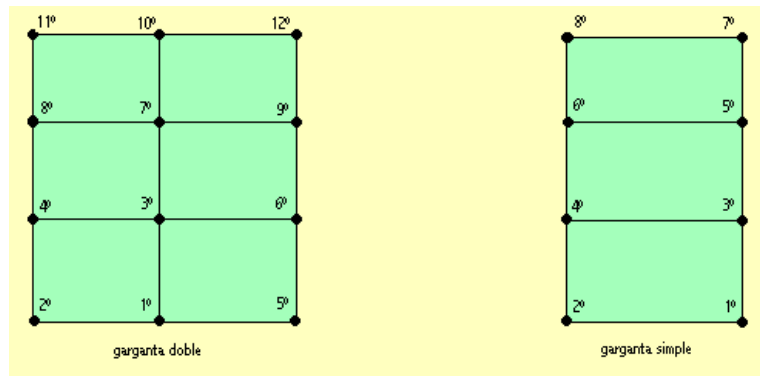


Figura III-140. Orden de numeración de los puntos que definen las gargantas verticales

Las clases `CGargantaHorizontal` y `CColGargantasHorizontales`, `CMuro` y `CColMuro`

Las gargantas horizontales son iguales en todas las rebanadas del fuste, por lo que se determinan en la base del mismo. Hay que distinguir dos tipos:

- Gargantas interiores formadas entre celdas circulares adyacentes:

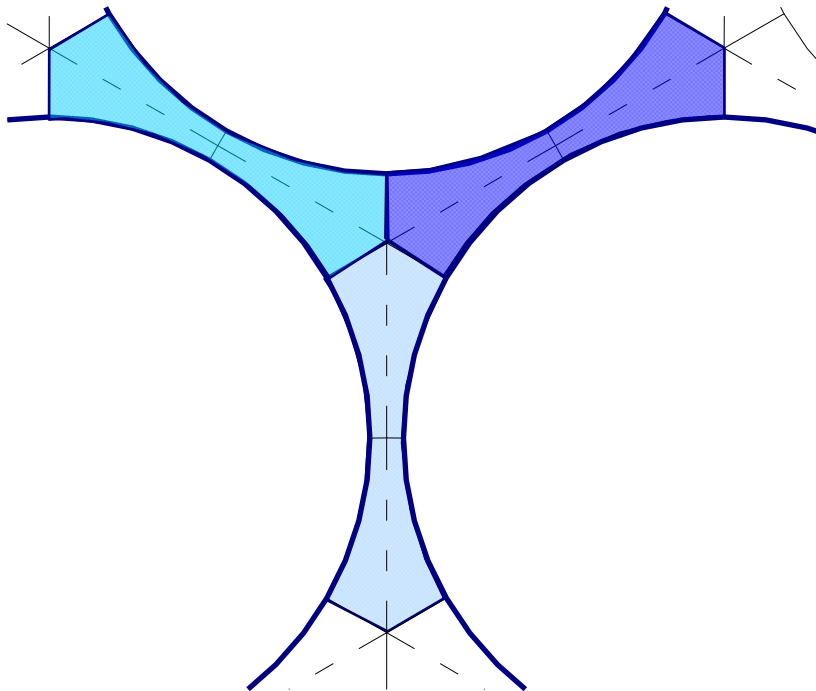


Figura III-141. Geometría de las gargantas horizontales interiores

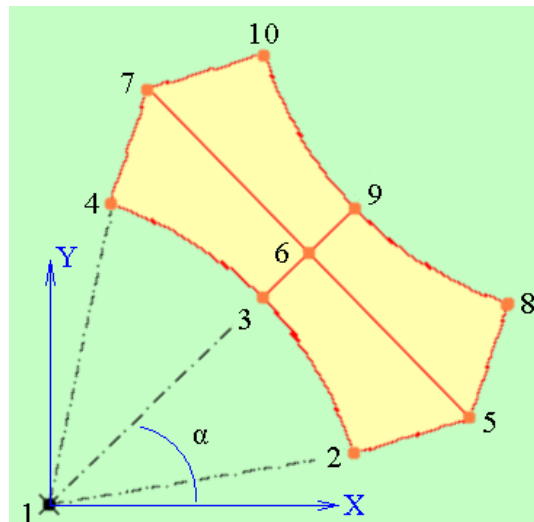


Figura III-142. Garganta horizontal interna y puntos que la definen

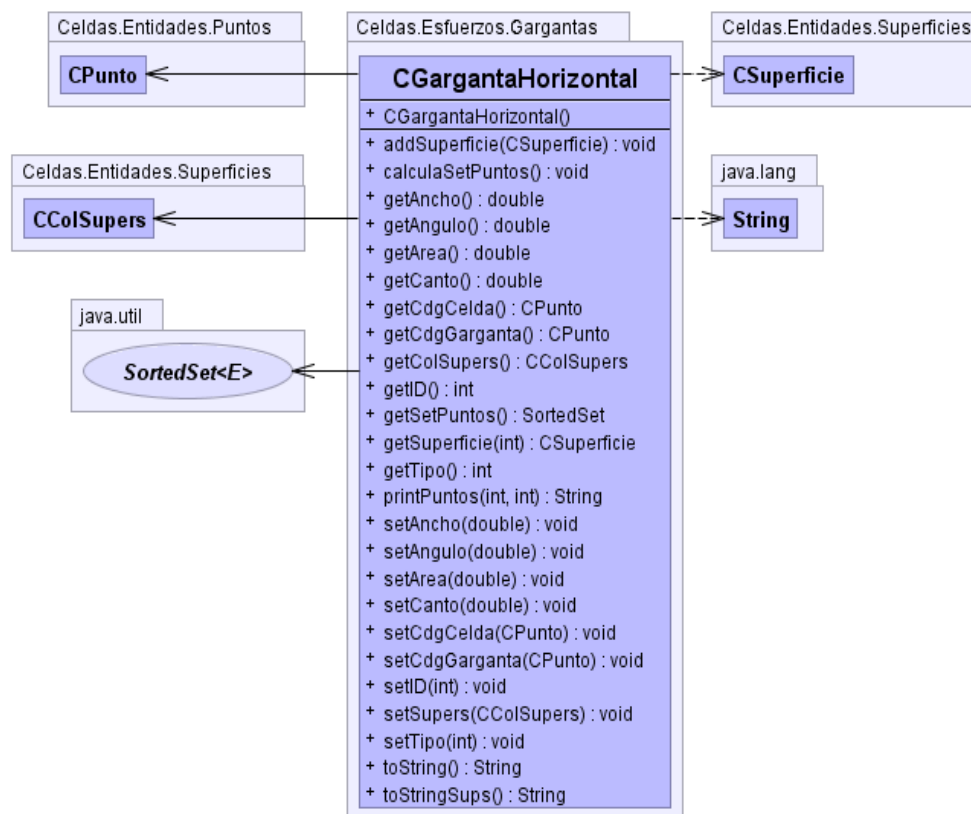


Figura III-143. Diagrama UML de la clase CGargantaHorizontal

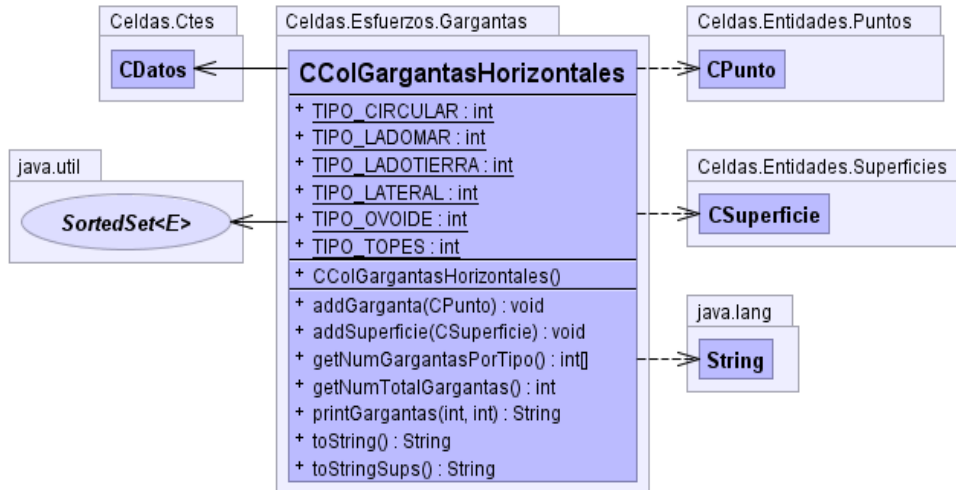


Figura III-144. Diagrama UML de la clase CColGargantasHorizontales

- Gargantas ubicadas en las paredes exteriores

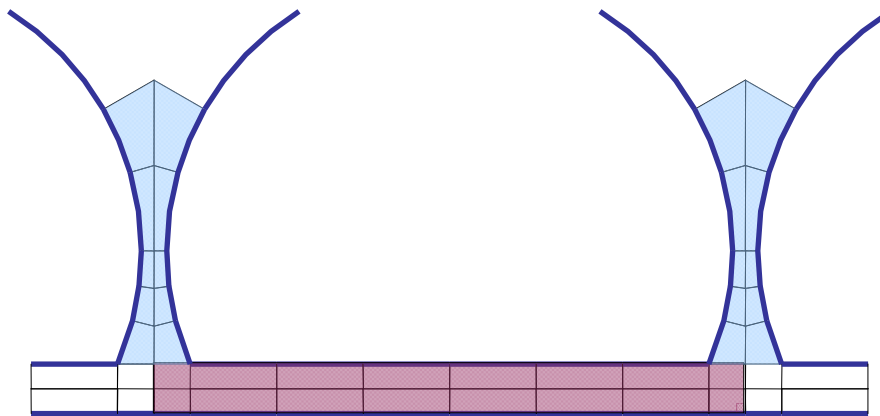


Figura III-145. Geometría de las gargantas horizontales de pared

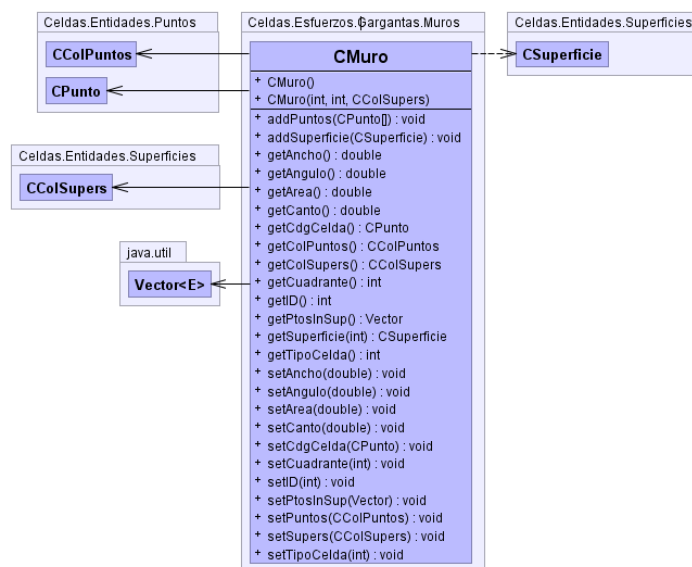


Figura III-146. Diagrama UML de la clase CMuro

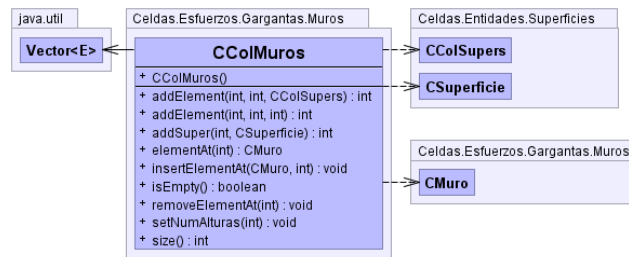


Figura III-147. Diagrama UML de la clase CColMuros

En ambos tipos de gargantas se debe conocer el ángulo que forman con la horizontal, así como su área y centro de gravedad y el orden de aparición de los puntos que las definen. Para determinar las gargantas horizontales se recorren todas las superficies horizontales de cada uno de los cubos generados en la base del fuste, seleccionando aquéllas que fueron marcadas como superficie con garganta (**Figura III-60**). La garganta queda definida por el punto central en la base del fuste y por todas las superficies planas que lo conforman.

En cualquier caso, a la hora de realizar el armado, tanto gargantas interiores como en exteriores se tratarán como una sección rectangular equivalente.

La clase CSalidaGarg

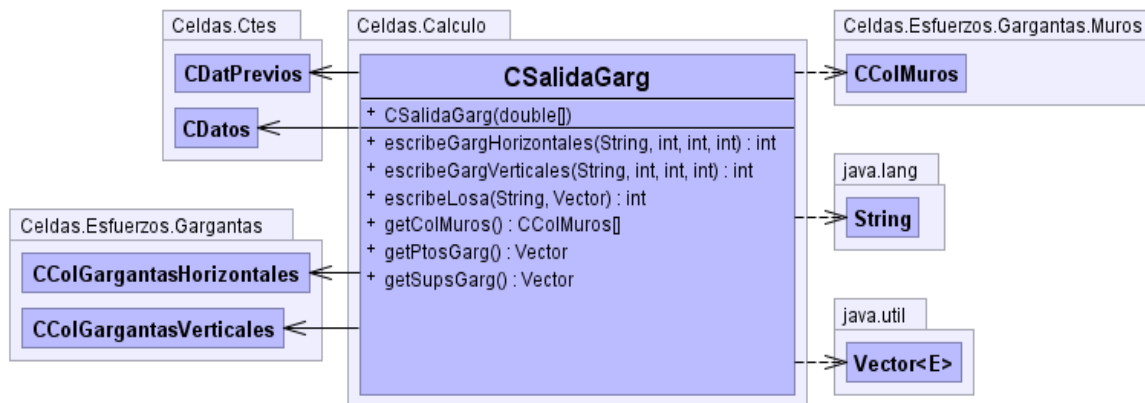


Tabla III-22. Diagrama UML de la clase CSalidaGarg

Tras la creación de las colecciones de gargantas, se escriben en los ficheros gargantasVert.gar (**Tabla III-23**) y gargantasHorz.gar (**Tabla III-24**).

```

número de gargantas: 196
Tramos de fuste: 15
número de rebanadas: 30
nodos por hexaedro: 20
elementos por lado: 1
elementos por altura: 2

Garganta numero: 1 Tipo: 0
Punto      Coor_x      Coor_y      Coor_z
5448 1.0775 18.1 0.55
5431 1.0775 18.3 0.55
5412 1.0775 18.5 0.55
8250 1.0775 18.1 1.1
8233 1.0775 18.3 1.1
8214 1.0775 18.5 1.1
    
```



```

...
39532 1.0775 18.1 17.5
39521 1.0775 18.3 17.5
39502 1.0775 18.5 17.5

Garganta numero: 2 Tipo: 0
Punto      Coor_x      Coor_y      Coor_z
5252 3.86 18.1 0.55
5234 3.86 18.3 0.55
5227 3.86 18.5 0.55

...
Garganta numero: 196 Tipo: 4
Punto      Coor_x      Coor_y      Coor_z
5086 35.295 17.28238562010067 0.55
3730 35.395 17.28238562010067 0.55
3768 35.495 17.28238562010067 0.55

...
37646 35.295 17.28238562010067 17.2
36888 35.395 17.28238562010067 17.2
36926 35.495 17.28238562010067 17.2

```

Tabla III-23. Fichero con gargantas verticales

```

número de gargantas: 205
Tramos de fuste: 15
número de rebanadas: 30
nodos por hexaedro: 20
elementos por lado: 1
elementos por altura: 2

Garganta numero: 1
Tipo: 0
Angulo: 90.0
Area: 0.7917333218311879
Ancho: 2.005
Canto: 0.4
Cdg: 1.1049244015014792 18.282308696719603
Superficies: 12
-2.220446049250313E-16 18.5 -2.220446049250313E-16 18.3 0.125 18.3 0.125 18.5 0.025
0.125 18.5 0.125 18.3 0.25 18.3 0.25 18.5 0.025
0.25 18.5 0.25 18.3 0.49236913856813125 18.3 0.49236913856813125 18.5
0.04847382771362611
0.49236913856813125 18.5 0.49236913856813125 18.3 1.0775 18.3 1.0775 18.5
0.11702617228637291
1.3700654307159317 18.3 1.0775 18.3 1.0775 18.1 1.3700654307159317 18.116352287597987
0.05612102911103873
1.0775 18.5 1.0775 18.3 1.3700654307159317 18.3 1.3700654307159317 18.5
0.05851308614318651
1.662630861431865 18.3 1.3700654307159317 18.3 1.3700654307159317 18.116352287597987
1.662630861431865 18.1 0.056121029111038695
1.3700654307159317 18.5 1.3700654307159317 18.3 1.662630861431865 18.3
1.662630861431865 18.5 0.05851308614318646
2.005 18.3 1.662630861431865 18.3 1.662630861431865 18.1 2.005 18.1
0.06847382771362602
1.662630861431865 18.5 1.662630861431865 18.3 2.0049999999999963 18.3 2.005 18.5
0.06847382771362603
1.0775 18.3 0.49236913856813125 18.3 0.49236913856813125 17.85763086143187 1.0775 18.1
0.16112974705878205
1.662630861431865 18.1 1.3700654307159317 18.116352287597987 1.0774999999999981 18.1
1.662630861431865 17.85763086143187 0.048887688836704676

Garganta numero: 2
Tipo: 0
Angulo: 90.0
Area: 1.4839999999999995
Ancho: 3.71
Canto: 0.4
Cdg: 3.8599999999999963 18.3
Superficies: 16
2.005 18.3 2.005 18.1 2.3070892730804378 18.1 2.3070892730804378 18.3
0.060417854616088076
2.005 18.5 2.005 18.3 2.3070892730804378 18.3 2.3070892730804378 18.5
0.060417854616088076
...

Garganta numero: 205
Tipo: 5
Angulo: 90.0
Area: 1.0347645000000003
Ancho: 1.023
Canto: 1.0115
Cdg: 39.68028077113198 17.951140385565996

```

3. Los Cajones Multicelulares

```
Superficies: 12
39.7665 18.5 39.255 18.5 39.255 18.3 39.7665 18.331416666666666 0.09426518749999968
39.7665 18.331416666666666 39.255 18.3 39.255 18.1 39.7665 18.162833333333333
0.09426518749999968
39.7665 18.162833333333333 39.255 18.1 39.255 17.99745584915977 39.7665 17.99425
0.0693408540773881
39.7665 17.99425 39.255 17.99745584915977 39.255 17.63992073463022 39.7665
17.825666666666667 0.13455479304093224
39.7665 17.825666666666667 39.255 17.63992073463022 39.255 17.28238562010067 39.7665
17.657083333333333 0.1345547930409323
39.7665 17.657083333333333 39.255 17.28238562010067 39.255 16.977 39.7665 17.4885
0.12121755984074492
40.278 18.5 39.7665 18.5 39.7665 18.331416666666666 40.278 18.416666666666668
0.06442768750000058
40.278 18.416666666666668 39.7665 18.331416666666666 39.7665 18.162833333333333 40.278
18.333333333333332 0.06442768750000148
40.278 18.333333333333332 39.7665 18.162833333333333 39.7665 17.99425 40.278 18.25
0.06442768749999958
40.278 18.25 39.7665 17.99425 39.7665 17.825666666666667 40.278 18.166666666666668
0.06442768750000058
40.278 18.166666666666668 39.7665 17.825666666666667 39.7665 17.657083333333333 40.278
18.083333333333332 0.06442768750000147
40.278 18.083333333333332 39.7665 17.657083333333333 39.7665 17.4885 40.278 18.0
0.06442768749999969
```

Tabla III-24. Fichero con gargantas horizontales

La información contenida en estos ficheros permite reconstruir las gargantas: su numeración, ubicación y orientación.

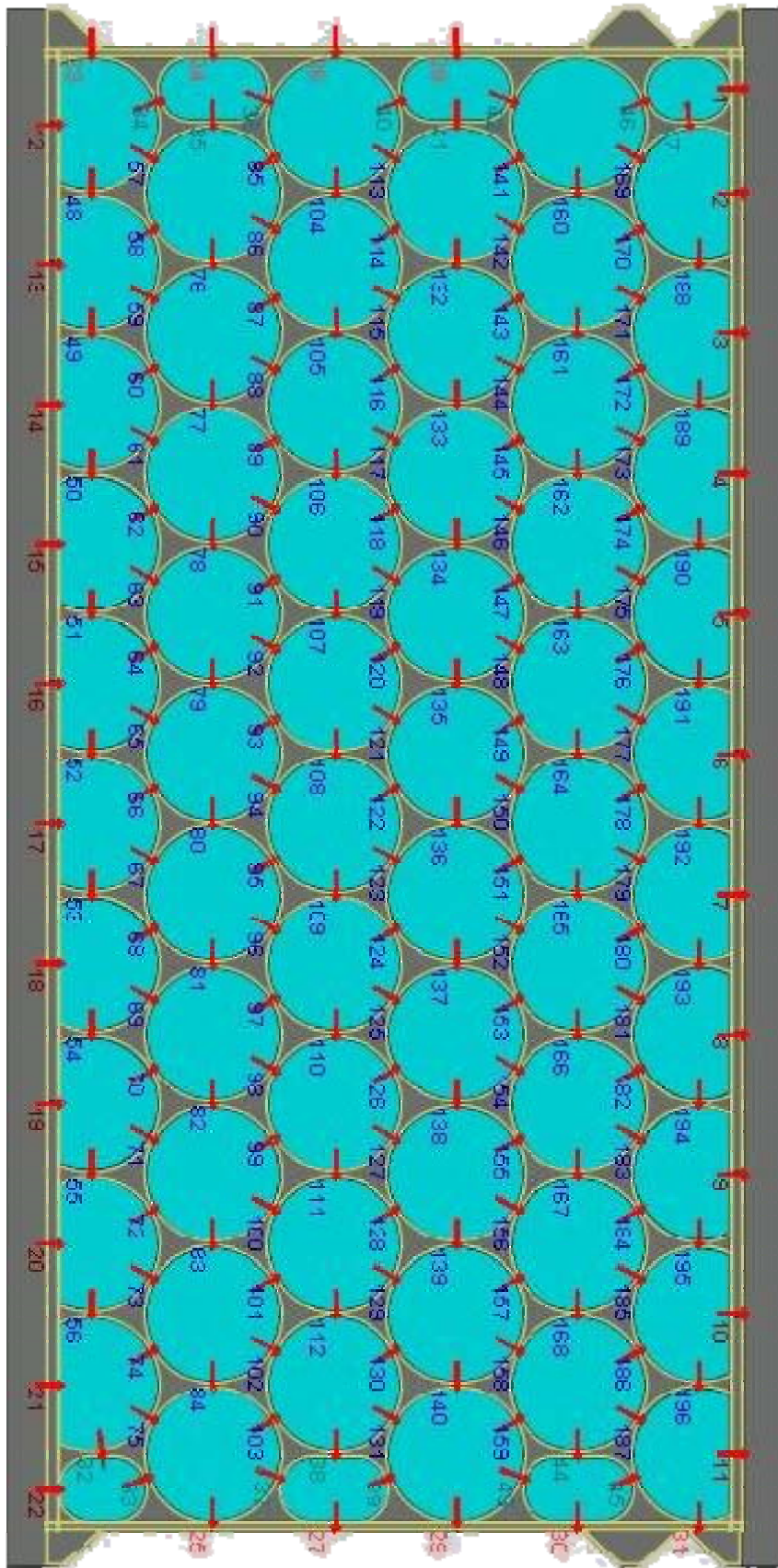


Figura III-148. Ejemplo de gargantas verticales en un cajón

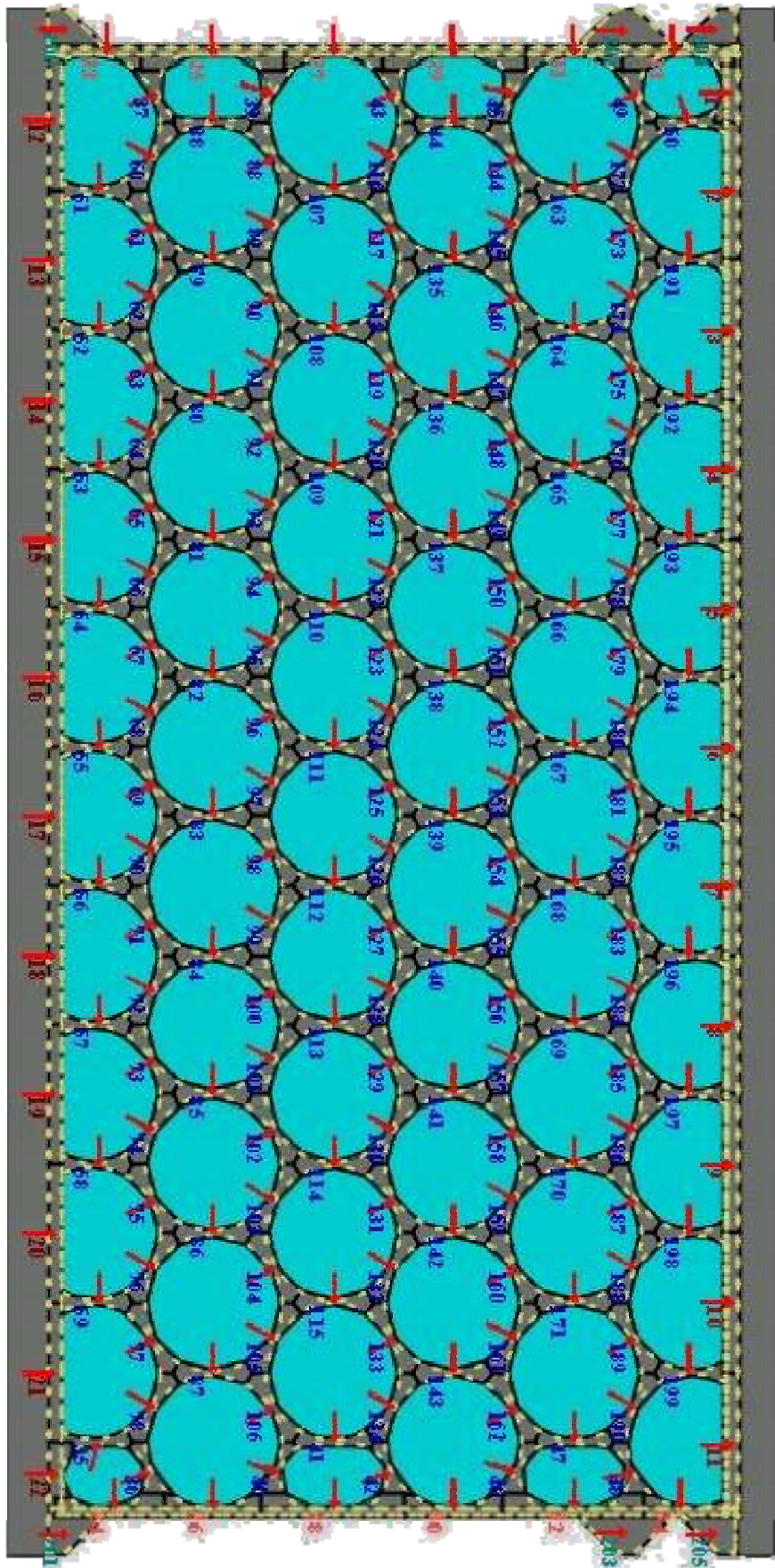


Figura III-149. Ejemplo de gargantas horizontales en un cajón

III.5.2.- Esfuerzos horizontales en gargantas verticales

Para la obtención del armado horizontal, se calculará el momento flector M_z y el axil N_x equivalentes a las tensiones normales σ_{xx} que se desarrollan tanto en las paredes interiores (entre celdas aligeradas) como en las exteriores.

La Figura 6.1 muestra las direcciones de actuación de la tensión normal y de los esfuerzos, mientras que su equivalencia se recoge en la ecuación (III-105). Igualmente se realiza la integración del resto de tensiones para determinar los esfuerzos cortantes y el momento torsor.

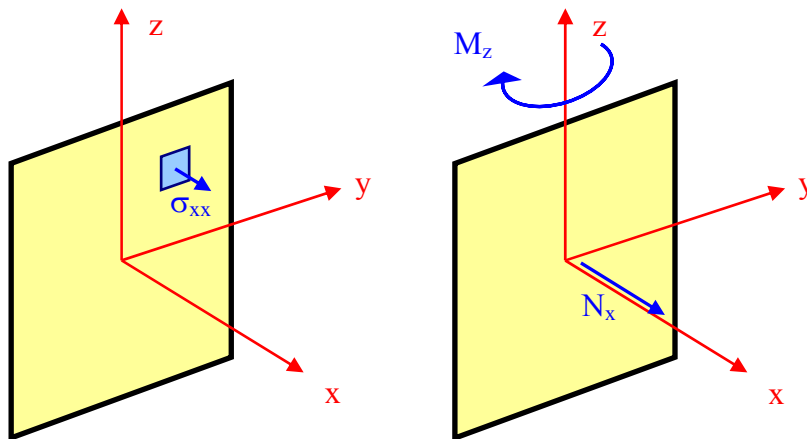


Figura III-150. Equivalencia entre tensiones normales y esfuerzos

$$\begin{aligned} N_x(x) &= \int_A \sigma_{xx} \, da \\ M_z(x) &= \int_A \sigma_{xx} \cdot y \, da \end{aligned} \quad \text{(III-105)}$$

En las paredes interiores las superficies más críticas son las *gargantas*, las de mínima distancia entre celdas circulares u ovoidales adyacentes. En las paredes exteriores, la zona de armado será también la de mínima distancia entre los paramentos exteriores y las celdas.

Por tanto, para el cálculo de los esfuerzos horizontales (M_z, N_x) en las paredes verticales, las cinco zonas de integración de tensiones serán las secciones rectangulares de canto igual al espesor de la pared:

1. Lado Tierra
2. Lado Mar
3. Pared Lateral
4. Gargantas celdas circular-circular
5. Gargantas celdas circular-ovoidal

3. Los Cajones Multicelulares

El ancho de dichas secciones es igual a la distancia vertical entre nodos. Por tanto, cada garganta estará dividida en secciones rectangulares, denominadas *subgargantas*.

Una vez que se disponen de las tensiones discretas en los nodos se procede a realizar las integrales para generar los esfuerzos en cada subgarganta. En esencia, para cada hipótesis *I* resuelta, el proceso es el siguiente:

1. Son dato los ficheros de mallado (*Hipotesis_I.gid/Hipotesis_I.flavia.msh*) y de resultados (*Hipotesis_I.gid/Hipotesis_I.flavia.res*) así como el fichero de las posiciones de los distintos puntos que definen las secciones o gargantas (*gargantas/gargantasVert.gar*).

El fichero de mallado contiene información acerca de los nodos y los elementos de la malla de utilizada: las posiciones de los nodos expresadas en el sistema global de coordenadas cartesianas y los elementos con sus conectividades. El fichero de resultados contiene información acerca de los desplazamientos, reacciones y las 6 componentes del tensor de tensiones en cada nodo. En el fichero de gargantas aparecen los 4 vértices de cada subgarganta con sus tres coordenadas correspondientes.

2. El proceso comienza generando las ecuaciones de los planos definidos por los puntos de las gargantas. A continuación se hace un barrido de los nodos de la malla identificando aquéllos que pertenecen a cada garganta puesto que van a ser los que afecten a los resultados. Además se generan los sistemas de referencia locales asociados a cada sección, cuya orientación es la indicada en la Figura 6.1.

3. Cada nodo lleva asociado una fracción del área de la subgarganta, tal y como se muestra en la **Figura III-151** para el caso de elementos de orden cuadrático. Expresando las coordenadas de los nodos y los tensores de tensiones en el sistema local de la subgarganta se realizan las integrales numéricas de las tensiones para calcular los esfuerzos mediante sumatorios extendidos a todos los nodos que la componen, ecuación (III-106).

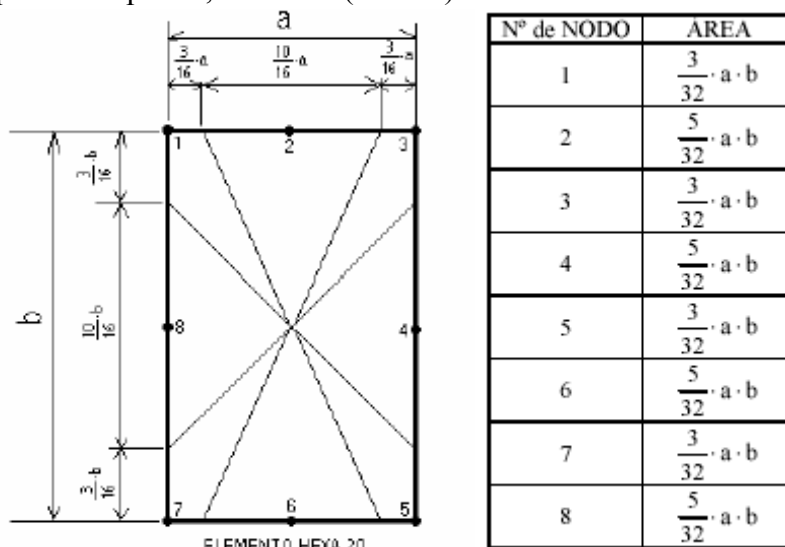


Figura III-151. Reparto de área para cada nodo en elementos cuadráticos

$$\iint_A f(y, z) dA \approx \sum_{i=1}^{NNodos} a_i \cdot f(y_i, z_i) \quad (\text{III-106})$$

4. Se genera un fichero de esfuerzos (*Hipotesis_I.arm/Hipotesis_I.esfHorzs.txt*) conteniendo las 6 componentes de esfuerzos (axil, cortantes, flectores y torsor) para cada subgarganta.

```

Esfuerzo Nx
Garganta - Valor en las distintas rebanadas
1 -94744.58090624967 -112277.53840421834 -121323.6162948433 -127179.5580718746 -
130526.06346249946 -132139.11646093716 -132482.26657812452 -131893.26640624978 -
130582.28042187478 -128642.30912499978 -126020.33140624958 -122372.73229687456 -
117285.43704687465 -110961.49792187483 -103962.69890625017 -96601.06949999982 -
89018.6789062501 -81300.98721875012 -73499.1733906251 -65649.6960156249 -
57763.063937499894 -49873.26756250007 -41972.72515625006 -34079.611390624945 -
26191.985496875037 -18310.530790625024 -10463.311081250049 -2699.478518749985
6096.908137499971 8645.52847499999
...
196 -95521.74912499796 -150697.9388281218 -199536.05437499576 -236746.02218749505 -
259380.91515624442 -270922.01687499456 -275864.87140624423 -277452.6889843696 -
277334.1491406196 -275907.9281249946 -272724.94554686936 -266860.6966406194 -
257579.97023436963 -245100.38585937026 -230354.36890624627 -214247.92890624586 -
197403.2428124968 -180190.2817968721 -162785.69484374733 -145271.6674218722 -
127652.78515624754 -109987.78465624822 -92284.566601561 -74676.23720312354 -
57369.22828906157 -40696.06918749934 -24986.369484374667 -10080.688678515398
4230.361555468655
Esfuerzo Vy
Garganta - Valor en las distintas rebanadas
1 -4786.655307499984 -4940.635412187483 -5983.51840703123 -5545.461503124983 -
5199.123165468729 -5392.117159140612 -6032.216646640607 -6890.135641406238 -
7766.796144843737 -8545.521056249985 -9170.755726406222 -9598.678909374967 -
9805.44588343747 -9825.222562812483 -9726.033506718764 -9541.561029218734 -
9280.995937500013 -8947.613374062514 -8551.702184062513 -8104.590460468737 -
7622.759593593739 -7110.83032656251 -6570.997297812509 -5983.456342968739 -
5314.740952656258 -4510.737844531257 -3503.2516232812663 -2134.1120701874897 -
79.29298335937455 1509.8161535624984
...
Esfuerzo Mz
Garganta - Valor en las distintas rebanadas
1 3910.8092375000256 8488.585692593751 13413.775846499972 18077.764925624946
21917.40720687491 24829.834485156174 26906.074582812387 28291.435379687427
29113.301423437413 29461.057685937412 29393.065948437365 28967.200967187375
28251.39414062487 27268.20771406242 26027.754212499996 24571.718296874926 22952.1942125
21218.179075 19408.869053124996 17557.577957812446 15688.12746406245 13832.225084374995
12012.444599999995 10264.913835937463 8626.817323437497 7139.709090624994
5838.624067187511 4722.995554687463 3795.448068749965 1597.2988612499878
...
196 105.21335859363126 181.5250742185624 282.3716289059996 403.63643359345133
518.6243945309195 616.9845234371535 695.1002499996446 752.4441015621414
789.7846914058907 808.4261914058914 809.9060976558935 795.8224609371507
767.886265624664 727.9020234371809 677.6976054684528 619.1821015622218
554.1785312497461 484.3207929685186 411.0425234372915 335.7695429685641
259.50342968733753 183.59581992173753 109.08441796863576 37.54325937490887 -
29.265837890693007 -88.91035429692124 -138.28010585940126 -171.49144505859954 -
186.83084763670666
Esfuerzo Mx
...

```

Tabla III-25. Fichero de esfuerzos horizontales en gargantas verticales

Para cada hipótesis de cálculo resuelta se presentan los esfuerzos obtenidos, en términos de axil y flector en N y N·m, respectivamente, y de forma *reducida*, o adimensional:

$$\nu = \frac{N}{b \cdot d \cdot f_{cd}} \quad \mu = \frac{M}{b \cdot d^2 \cdot f_{cd}} \quad (\text{III-107})$$

donde b es el ancho de la sección, d es el canto útil y f_{cd} es la resistencia de cálculo del hormigón. Los valores positivos del axil indican tracción. Los valores positivos del

momento indican tracción de las fibras $y>0$ y compresión de las fibras $y<0$, **Figura III-152.**

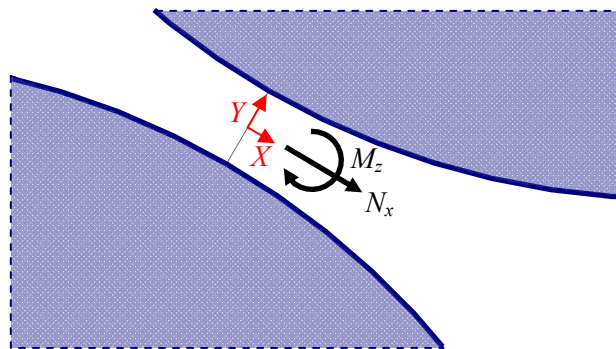


Figura III-152. Criterio de esfuerzos positivos

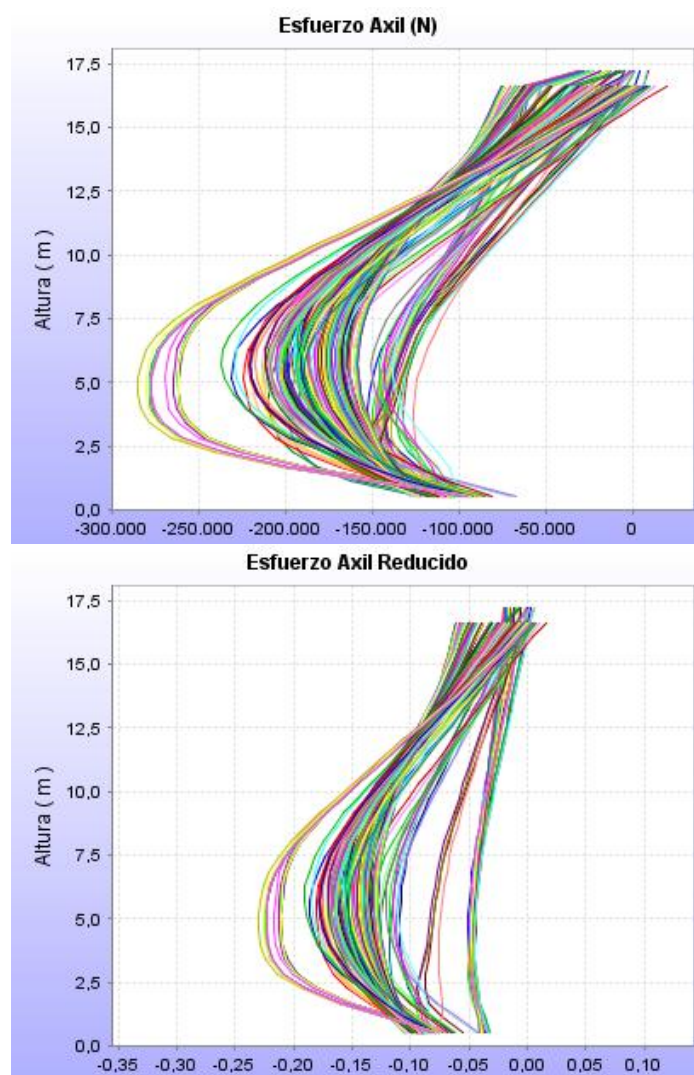


Figura III-153. Gráficas de esfuerzos axil y axil reducido en gargantas verticales

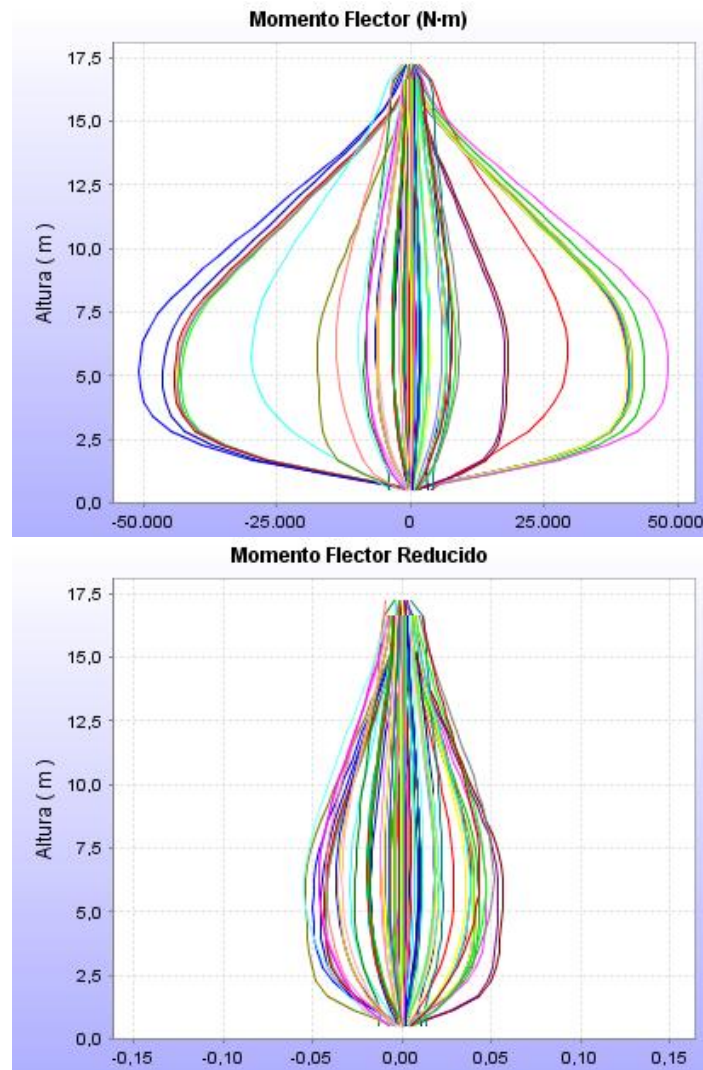


Figura III-154. Gráficas de momento flector y flector reducido en gargantas verticales

III.5.3.- Esfuerzos verticales en gargantas horizontales

Una vez que se conocen las gargantas horizontales y una vez que se disponen de las tensiones discretas en los nodos procede a realizar las integrales de manera numérica. En esencia, para cada hipótesis I resuelta, el proceso es el siguiente:

1. Son dato los ficheros de mallado (*Hipotesis_I.gid/Hipotesis_I.flavia.msh*) y de resultados (*Hipotesis_I.gid/Hipotesis_I.flavia.res*) así como el fichero de las posiciones de los distintos puntos que definen las gargantas verticales interiores (*gargantas/gargantasHorz.gar*) y las gargantas verticales exteriores correspondientes a las paredes (*Hipotesis_I.arm/Hipotesis_IMuros.gar*). El fichero de mallado contiene información acerca de los nodos y los elementos de la malla de utilizada: las posiciones de los nodos expresadas en el sistema global de coordenadas cartesianas y los elementos con sus conectividades. El fichero de resultados contiene información acerca de los desplazamientos, reacciones y las 6 componentes del tensor de tensiones en cada nodo. En el fichero de gargantas aparecen los diversos vértices de cada subgarganta con sus tres coordenadas correspondientes.

- El proceso comienza generando las ecuaciones de los planos definidos por los puntos de las gargantas. A continuación se hace un barrido de los nodos de la malla identificando aquéllos que pertenecen a cada garganta puesto que van a ser los que afecten a los resultados. Además se generan los sistemas de referencia locales asociados a cada sección, cuya orientación es la indicada en la **Figura III-155** en el caso de las gargantas interiores. Para las exteriores, se emplea un eje local Y perpendicular a la pared y que va en el sentido indicado en la **Figura III-149**.

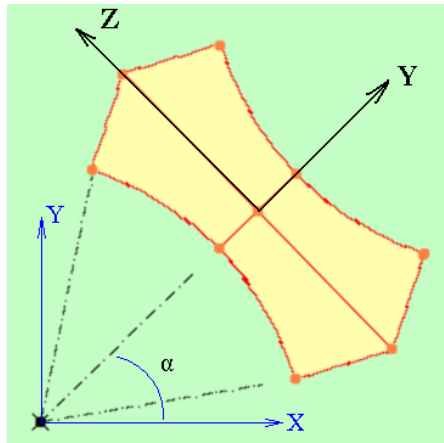


Figura III-155. Ejes locales en garganta interior

- Expresando las coordenadas de los nodos y los tensores de tensiones en el sistema local de la subgarganta se realizan las integrales numéricas de las tensiones para calcular los esfuerzos mediante sumatorios extendidos a todos los nodos que la componen (ecuación (III-106)). En general, y ante la dificultad de calcular el área de influencia de cada nodo se opta por suponer que el diferencial de área es igual para todos los que caen dentro de la subgarganta.
- Se genera un fichero de esfuerzos (*Hipotesis_I.arm/Hipotesis_I.esfVerts.txt*) conteniendo las 6 componentes de esfuerzos (axil, cortantes, flectores y torsor) para cada subgarganta.

Esfuerzo Nx	
Muro: 1	
0	-491905.3531983867
1	-552856.0207006088
2	-544251.3241825659
3	-538153.1729920005
4	-529432.6698689192
5	-518533.4885091052
6	-505965.3714041798
7	-492073.27128281706
8	-477093.93382195424
9	-461056.9785688462
10	-443961.1678357587
11	-425547.8792242863
12	-405407.130785597
13	-383341.7925525446
14	-359964.66861901624
15	-335739.53402696643
16	-311011.3698450989
17	-285954.5501094098
18	-260770.8453501236
19	-235573.7986344896
20	-210484.5967998473
21	-185636.48410002622
22	-161201.8929618246
23	-137285.52697337553
24	-114105.69076410151

```

25 -91775.5421143951
26 -70448.31157321495
27 -50240.680464945865
28 -30882.535616804296
29 -11021.761519353908
30 -975.3487166866998
...
Esfuerzo Vy
Muro: 1
0 9182.287136543855
1 23530.27088775248
2 27539.819668276257
3 23676.415921216016
4 17006.994643410126
5 10278.69511619197
...
Esfuerzo Mz
...
Muro: 205
0 -4145.695187821711
1 -10990.563379278385
2 -15557.59641686691
3 -18841.46033979437
4 -18143.36928549138
5 -15382.74668120042
6 -12384.850936875617
7 -10206.826332564902
8 -9060.039398644725
9 -8899.696775212264
10 -9305.167550141838
11 -9824.628368606958
12 -9932.83497802016
13 -9578.19628622849
14 -8747.709931026066
15 -7848.399420510763
16 -7030.972236312073
17 -6421.50140470524
18 -5952.397767269742
19 -5635.151874843035
20 -5372.978605506355
21 -5156.9698243585335
22 -4885.967736305073
23 -4520.353360630955
24 -3959.8081508554465
25 -3199.0745996760256
26 -2325.5900828627614
27 -1681.4914870178375
28 -1505.836203982087
29 -1127.5737310366621
30 -314.4246514172472
Esfuerzo Mx

```

Tabla III-26. Fichero de esfuerzos verticales en gargantas horizontales

Para cada hipótesis de cálculo resuelta se determinan los esfuerzos, en términos de axil y flector en N y $N \cdot m$, respectivamente, y de forma *reducida*, o adimensional.

Los valores positivos del axil indican tracción. Los valores positivos del momento indican tracción de las fibras $y > 0$ y compresión de las fibras $y < 0$, Figura 6.9.

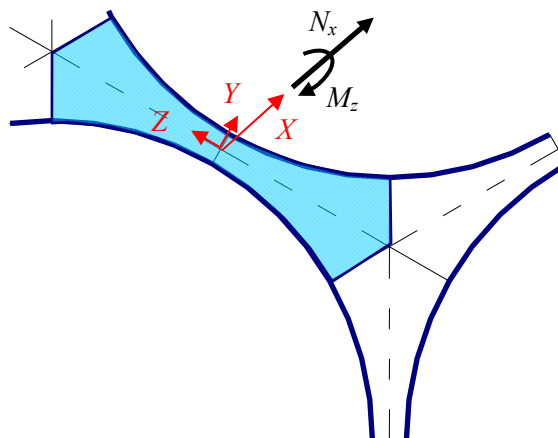


Figura III-156. Criterio de esfuerzos positivos

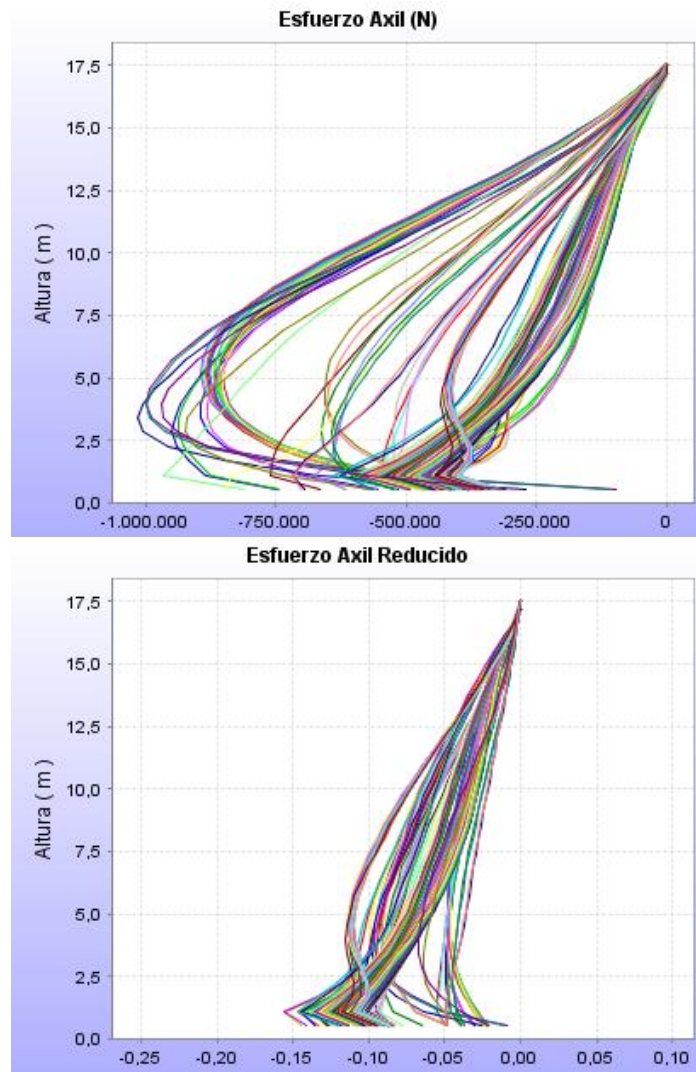
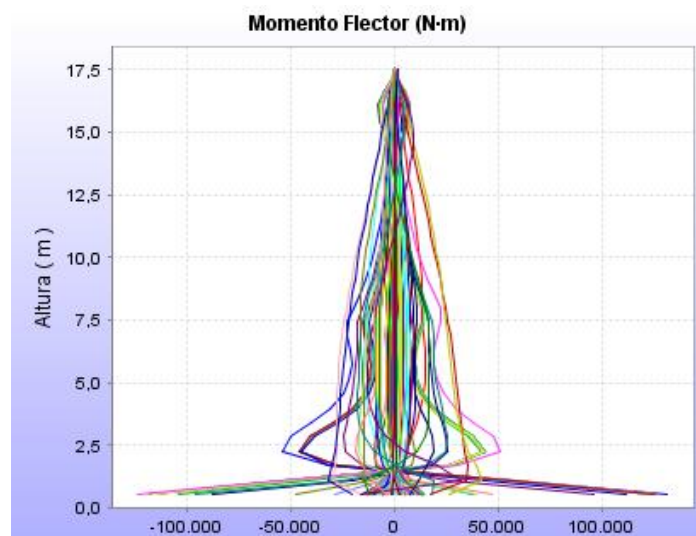


Figura III-157. Gráficas de esfuerzos axil y axil reducido en gargantas horizontales



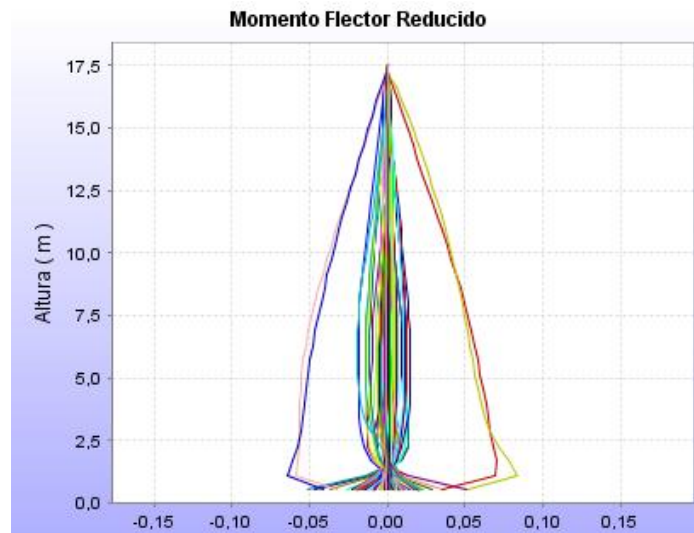


Figura III-158. Gráficas de momento flector y flector reducido en gargantas horizontales

III.5.4.- Validación de resultados

Los algoritmos empleados para la conversión de las tensiones discretas en nodos pertenecientes a una superficie común en esfuerzos resultantes sobre dicha superficie han sido convenientemente validados frente a problemas de solución conocida.

En concreto, se ha realizado el análisis de una viga en voladizo bajo diversos estados de carga, de solución conocida, y mallada con elementos hexaédricos de 20 nodos.

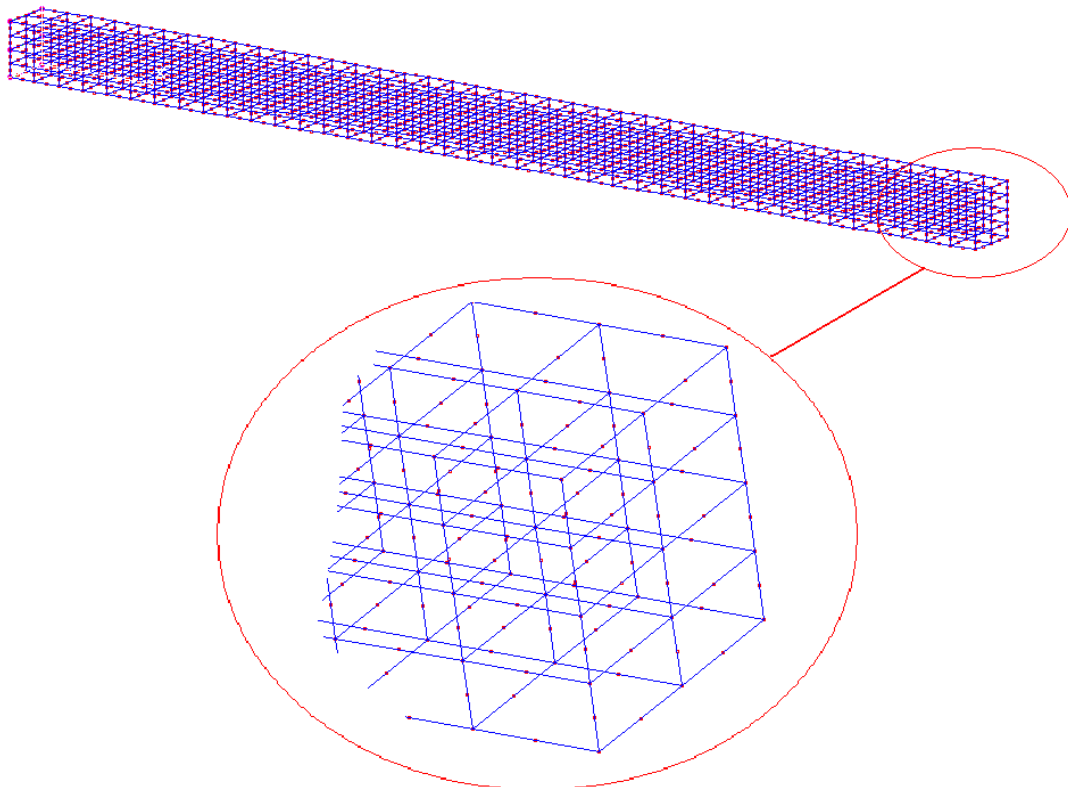


Figura III-159. Modelo de viga para validación de algoritmos

3. Los Cajones Multicelulares

Las secciones que se han marcado para obtener datos corresponden con las que están situadas equiespaciadas con las coordenadas longitudinales $X= 5 \text{ m}$, 10 m , 15 m , 20 m , 25 m , 30 m y 35 m tal y como se observa en la figura siguiente:

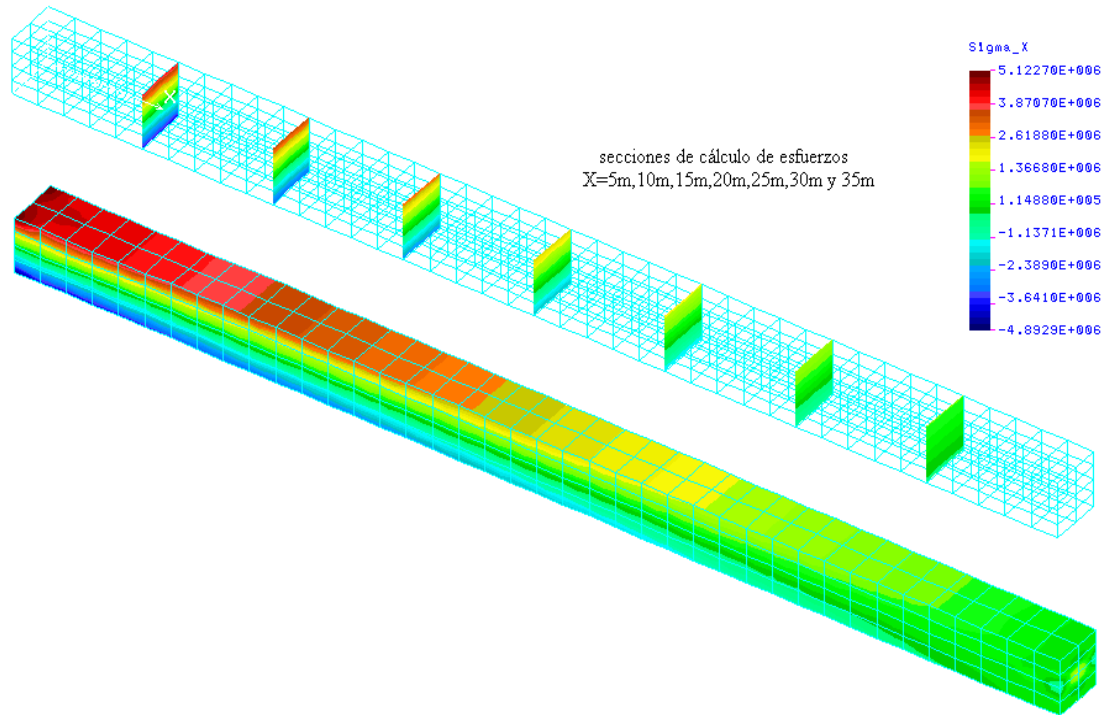


Figura III-160. Tensiones normales en secciones de estudio

Para el caso de tracción (carga de 350 kN) y flexión en el plano XY (carga vertical 150 kN en la punta)

		sección X:	5 m	10 m	15 m	20 m	25 m	30 m	35 m
N_x (N)	TEORIA		350000	350000	350000	350000	350000	350000	350000
	algoritmo		349964	350000	350000	350000	350000	350000	350001
V_y (N)	TEORIA		-150000	-150000	-150000	-150000	-150000	-150000	-150000
	algoritmo		-150000.2	-150003	-150004	-149998	-150004	-150004	-149999
V_z (N)	TEORIA		0	0	0	0	0	0	0
	algoritmo		1e-6	1e-5	1e-5	1e-6	1e-6	1e-5	1e-6
M_y (N.m)	TEORIA		0	0	0	0	0	0	0
	algoritmo		1e-11	1e-11	1e-12	1e-11	1e-8	1e-10	1e-10
M_z (N.m)	TEORIA		5250000	4500000	3750000	3000000	2250000	1500000	750000
	algoritmo		5598430	4799300	3999250	3200000	2399390	1599450	799000
M_x (N.m)	TEORIA		0	0	0	0	0	0	0
	algoritmo		1e-7	1e-5	1e-7	1e-6	1e-6	1e-6	1e-6

Tabla III-27. Comparativa de resultados según algoritmo propuesto

Si bien el axil y el cortante obtenidos por el algoritmo arrojan valores muy cercanos a la solución real, en el caso del momento flector los errores alcanzan el 6%, por el lado de la seguridad. El error se ve reducido a medida que la densidad de la malla aumenta, no obstante.

III.5.5.- Esfuerzos en la losa

El algoritmo de integración de esfuerzos en las gargantas no se hace extensible a la losa de fondo ni a los voladizos.

Los diagramas de esfuerzos obtenidos en la losa mediante la integración de tensiones del modelo tridimensional se ven afectados, por las condiciones de contorno impuestas en el terreno para la resolución del problema y por la continuación vertical del fuste o no. Por este motivo no se han obtenido resultados de aplicación del modelo tridimensional y se ha realizado un estudio clásico mediante placas circulares empotradas.

Así, la losa de fondo se considera formada por un conjunto de placas circulares del radio igual al de las celdas, empotradas en el borde de unión con las paredes verticales y sometidas a la carga uniforme determinada por la ecuación (III-68).

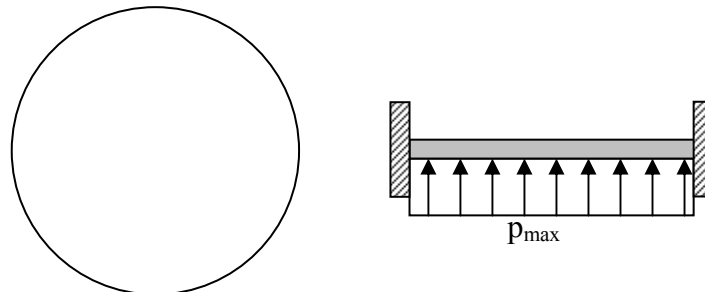


Figura III-161. Placa circular empotrada bajo carga uniforme

La solución del momento flector por unidad de longitud en la dirección radial de la placa, tomando origen en el centro de la misma viene dada por la siguiente expresión:

$$M_r(r) = \frac{P}{16} \cdot [R^2 \cdot (1 + \nu) - r^2 \cdot (3 + \nu)] \quad \text{(III-108)}$$

mientras que el esfuerzo cortante por unidad de longitud es:

$$Q(r) = \frac{P}{2} \cdot r \quad \text{(III-109)}$$

En cuanto a los esfuerzos en los voladizos, se calcula el momento flector en la sección de referencia que se encuentra a una distancia del paramento exterior igual a 0.15 veces el ancho del muro. El cortante se evalúa a una distancia del paramento igual al canto útil.

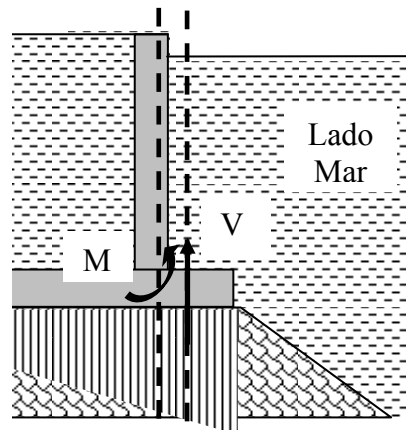


Figura III-162. Secciones de referencia para el cálculo de esfuerzos en el voladizo

III.6.- Armado de los elementos estructurales

III.6.1.- El armado según la EHE

Uno de los objetivos del diseño estructural es garantizar un comportamiento estructural adecuado. La estructura debe ser lo suficientemente fuerte (resistente) para soportar de forma segura todas las cargas que previsiblemente pudieran actuar sobre ella.

Mediante el **Método de los Estados Límite** se definen todas las situaciones límite potenciales, que pudieran dejar a la estructura fuera de servicio (*pérdida de resistencia o de funcionalidad*). Se introduce el nivel de seguridad adecuado al caso (coeficientes de seguridad; minoración de resistencia de los materiales, mayoración de cargas). Así, la Estructura Ideal de Cálculo que resulta será mucho más crítica que la proyectada.

En la práctica se dimensiona de forma que se esté en la **situación límite de rotura** y se comprueba que no se alcancen las otras situaciones límite.

La actual normativa de Hormigón Estructural EHE, en el Artículo 42º - *Estado Límite de Agotamiento frente a sollicitaciones normales* define los esfuerzos normales como aquéllos que producen en cada sección tensiones normales, paralelas a la directriz de la pieza. Son, por tanto, el momento flector y el esfuerzo axial. La sección resistente del hormigón se obtiene de las dimensiones de la pieza.

El estado límite último de agotamiento resistente de una sección de hormigón armado sometida a flexión simple o compuesta se produce por aplastamiento del hormigón comprimido. En este instante, la capacidad resistente última de la sección es equivalente a los esfuerzos últimos que provocan esta situación. Por tanto, la filosofía del método de diseño en rotura consiste en dimensionar la sección resistente de tal manera que su capacidad última coincida con los efectos de las sollicitaciones normales de cálculo.

III.6.2.- Planteamiento del problema

En el problema concreto de diseño de las gargantas horizontales o verticales de hormigón armado, las dimensiones de la sección transversal ancho b y canto h , son especificadas para cada garganta según su tipología en los ficheros *.gar.

También son conocidas las características resistentes de los materiales que se van a disponer: la resistencia a compresión del hormigón, f_{ck} , y la resistencia a tracción/compresión del acero, f_{yk} . Un ejemplo de hormigón que se emplea en la construcción del cajón es: **HA - 30 / B / 20 / IIIb + Qb**, para el que $f_{ck} = 30 \text{ MPa}$. Así mismo, el acero a emplear suele ser **B-400-S**, por lo que $f_{yk} = 400 \text{ MPa}$.

La distancia a la que es colocado el armado respecto de los paramentos exteriores se denomina recubrimiento. El recubrimiento geométrico es la distancia entre la superficie exterior de la armadura y la superficie de hormigón más cercana. Viene prefijado por la Instrucción EHE, en función del ambiente en el que el elemento estructural va a desempeñar su función (Tabla 37.2.4). Para un ambiente de tipo Qb se deja al proyectista fijar un valor que garantice la protección de las armaduras frente a la acción agresiva ambiental. En base a la experiencia en proyectos similares es habitual dotar de un recubrimiento geométrico a las paredes exteriores en contacto con el mar de **5 cm**, mientras que para las interiores, las cuales estarán en contacto con el material de relleno, es suficiente con disponer **4 cm**.

El recubrimiento mecánico (d_1 y d_2) es la distancia medida desde el centro de gravedad de la barra.

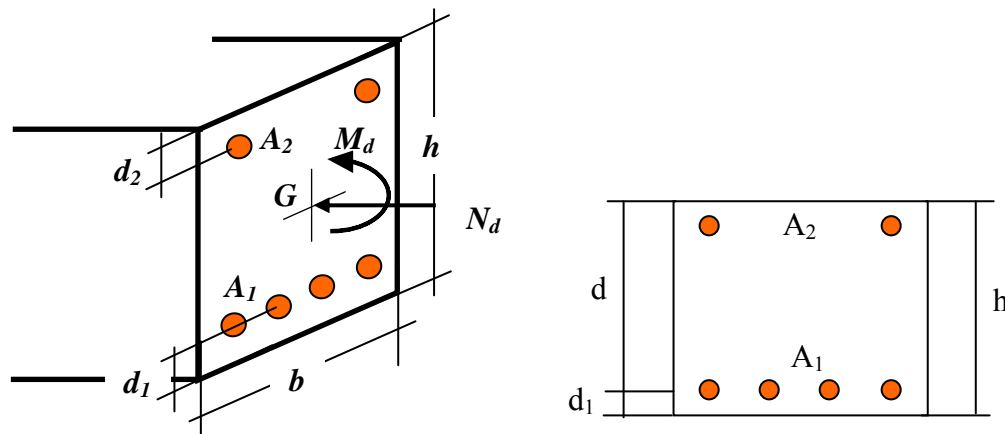


Figura III-163. Variables principales de la sección

Una vez que se han calculado los esfuerzos sobre cada subgarganta, se conocen el momento flector (M_d) y el esfuerzo axial (N_d) de cálculo y, según el Método de los Estados Límite, la situación de agotamiento de la sección se producirá bajo estos esfuerzos. Por tanto, habrá que calcular la armadura de acero (A_1 y A_2) que es necesario disponer en la sección para que su capacidad resistente sea justamente la capacidad última: *los esfuerzos últimos que resiste la sección son justamente los de cálculo que la solicitan*.

En general, se denomina A_1 a la armadura de acero traccionada, o en su caso, menos comprimida, y A_2 a la armadura comprimida, o en su caso, menos traccionada. d_1 es el recubrimiento mecánico de A_1 y d_2 el de A_2 . La distancia de la fibra más comprimida (en su caso, la menos traccionada) hasta el centro de gravedad de la armadura A_1 se denomina *canto útil* d , **Figura III-163**.

En las armaduras, σ_1 y ε_1 son tensión normal y deformación normal, en la armadura de tracción, respectivamente, mientras que σ_2 y ε_2 son la tensión normal y deformación normal en la armadura de compresión. En el hormigón, a su vez, σ_c y ε_c son la tensión normal y deformación normal, en la fibra más comprimida de la sección.

Según los esfuerzos que solicitan la sección, existen diversas formas de trabajo de la misma, las cuales se ven reflejadas en la posición de la línea neutra, medida a través de la profundidad x , **Figura III-164**. Así, cabe distinguir inicialmente entre:

- Situaciones en las que toda la sección trabaja a tracción. La profundidad de la línea neutra varía en el intervalo $-\infty \leq x \leq 0$. El problema es de tracción simple (axil de tracción, sin flector) si $x = -\infty$ o compuesta (axil de tracción predominante sobre el momento flector) si $-\infty < x \leq 0$.

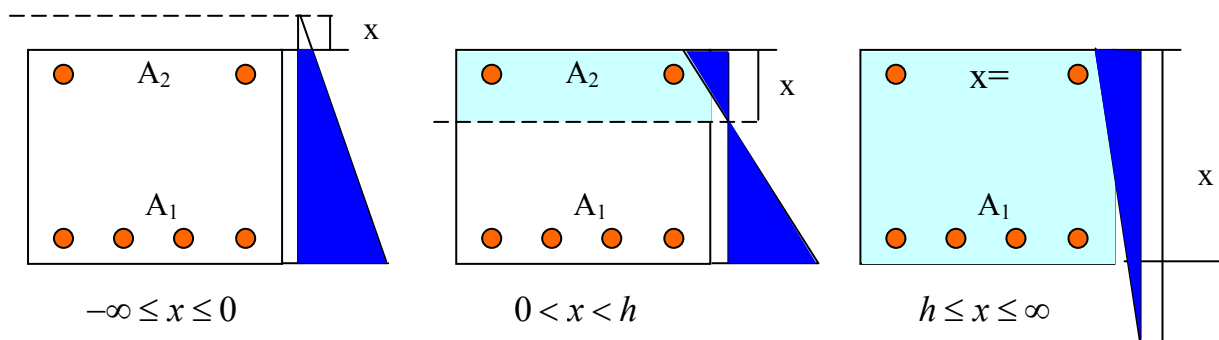


Figura III-164. Formas de trabajo de la sección y leyes lineales de deformación

- Situaciones en las que una parte trabaja a compresión y otra a tracción. La profundidad de la línea neutra varía en el intervalo $0 < x < h$. El problema es de flexión simple si no hay axil y flexión compuesta, si actúan conjuntamente el flector y el axil, siendo el primero el predominante.
- Situaciones en las que toda la sección está comprimida. La profundidad de la línea neutra varía en el intervalo $h \leq x \leq \infty$. El problema es de compresión simple si $x = \infty$ (axil de compresión, sin flector) o compuesta (axil de compresión, predominante sobre el momento flector) si $0 \leq x < \infty$.

III.6.3.- Hipótesis de Cálculo

1. Sobre la deformación de la sección y las ecuaciones de compatibilidad

- Se suponen pequeñas deformaciones.
- Se dan las condiciones de adherencia perfecta, de modo que las deformaciones normales (alargamientos y contracciones) en las armaduras son iguales a las que se producen en el hormigón que las envuelve.
- Se admite la hipótesis de Navier-Bernoulli, por la que la sección permanece plana tras la deformación, por lo que la ley de deformaciones normales varía linealmente a lo largo de la misma (Figura 7.2).

2. Sobre la ley de comportamiento del hormigón

Se asume (Instrucción EHE - Artículo 39.5) que la ley de comportamiento del hormigón sigue la ley conocida como *parábola-rectángulo*, ecuación (III-110):

$$\sigma(\varepsilon) = \begin{cases} 0.85 \cdot f_{cd} \cdot \left[2 \cdot \frac{\varepsilon}{\varepsilon_0} - \left(\frac{\varepsilon}{\varepsilon_0} \right)^2 \right] & 0 \leq \varepsilon \leq \varepsilon_0 \\ 0.85 \cdot f_{ck} & \varepsilon_0 \leq \varepsilon \leq \varepsilon_c \end{cases} \quad \text{(III-110)}$$

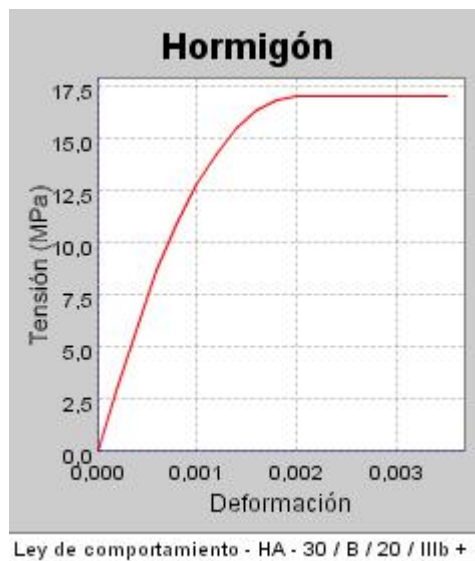


Figura III-165. Ley de comportamiento del hormigón

donde $\varepsilon_0 = 0.002$ (deformación de rotura del hormigón por compresión simple), $\varepsilon_c = 0.0035$ (deformación de rotura del hormigón en flexión) y $f_{cd} = f_{ck} / \gamma_c$ es la resistencia de cálculo a compresión. Como coeficiente de minoración de la resistencia se toma $\gamma_c = 1.5$.

3. Sobre la ley de comportamiento del acero

Como relación de comportamiento del acero de dureza natural de las armaduras pasivas se adopta el diagrama birrectilíneo, cuyo tramo lineal tiene pendiente igual al módulo de elasticidad del acero, $E_s = 200000 \text{ MPa}$. La deformación del acero en tracción se limita al 10 por 1000, máxima deformación plástica admitida. En compresión se considera la misma deformación límite que en el hormigón, 0.0035. Ambas ramas horizontales comienzan en el punto de límite elástico $\varepsilon_y = \frac{f_{yd}}{E_s}$, donde $f_{yd} = \frac{f_{yk}}{\gamma_s}$ es la resistencia de cálculo del acero. Como coeficiente de minoración de la resistencia se toma $\gamma_s = 1.15$.

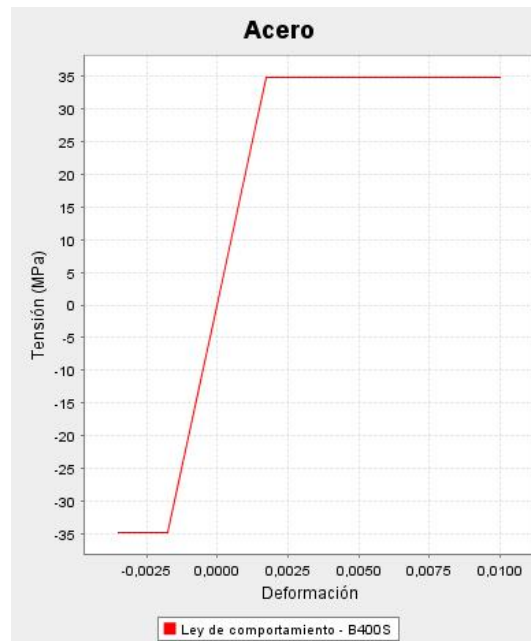


Figura III-166. Ley de comportamiento del acero

4. Sobre las ecuaciones de equilibrio entre fuerzas y esfuerzos

Se admite la hipótesis de pequeños desplazamientos, por lo que el equilibrio entre las acciones externas y los esfuerzos internos se plantea en la configuración indeformada de la pieza.

5. Sobre las ecuaciones de equivalencia estática entre tensiones y esfuerzos

Se desprecia la contribución de las tensiones que se originan en el hormigón traccionado, que en condiciones de rotura está fisurado, y es el acero el que pasa a soportar las tensiones de tracción.

III.6.4.- Caracterización de la ley de deformaciones en el Estado Límite de Agotamiento Resistente

Se relacionará la deformación límite, en situación de rotura, con la profundidad de la línea neutra. A una sección dada con unas armaduras dadas le corresponden unos esfuerzos últimos resistidos (y su ley de deformaciones límite asociada) distintos según sea la profundidad de la línea neutra. Por esta razón, se definen los llamados *dominios de deformación*, para asociar la forma de trabajo de la sección a los esfuerzos y deformaciones sobre la misma, recorriendo de manera continua desde la tracción simple hasta la compresión simple. La **Figura III-167** muestra los distintos dominios en función de la ley de deformaciones. La **Tabla III-28** recoge los valores límite entre dominios.

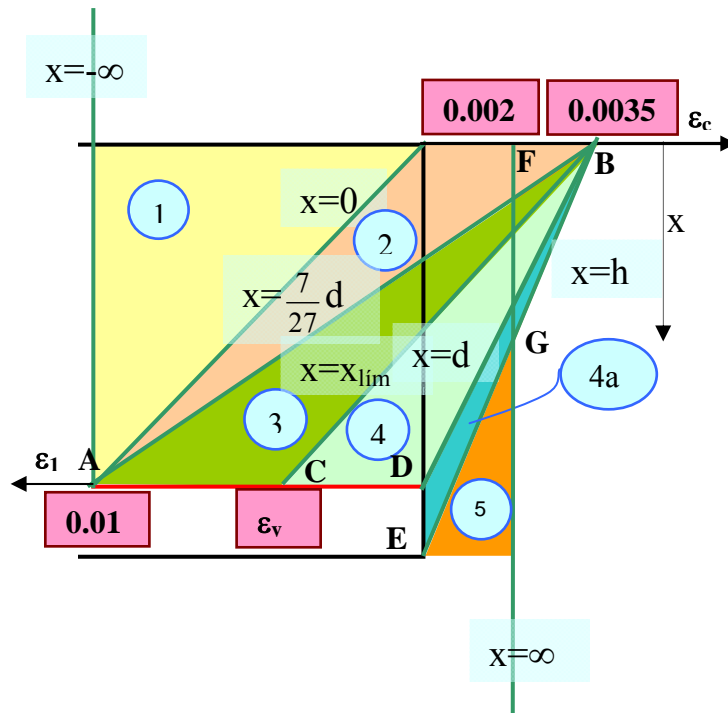


Figura III-167. Dominios de deformación de la EHE

Dominios	Intervalo de profundidad de la línea neutra
1	$-\infty \leq x \leq 0$
2	$0 < x < \frac{7}{27}d$
3	$\frac{7}{27}d \leq x \leq \frac{0.0035}{0.0035 + \varepsilon_y}d$
4	$\frac{0.0035}{0.0035 + \varepsilon_y}d < x \leq d$
4.a	$d < x < h$
5	$h \leq x \leq \infty$

Tabla III-28. Límites de los dominios

Tracción - DOMINIO 1

Para cualquier profundidad x de la línea neutra, el estado límite implica una ley de deformación que pivota en el punto de alargamiento del acero más traccionado $\varepsilon_s = 0.01$ (A). El problema de tracción simple se corresponde con la recta $x = -\infty$, mientras que la recta $x = 0$ delimita el dominio 1 de deformación, el cual incluye todas las rectas que pasan por A y presentan leyes de deformación en tracción, con una profundidad de la línea neutra en el intervalo: $-\infty \leq x \leq 0$.

Flexión. DOMINIO 2

Para los valores iniciales de la profundidad x de la línea neutra dentro de la sección, la fibra más contraída de hormigón no alcanza la deformación límite

$\varepsilon_c = 0.0035$ (B). La recta $x = 0$ delimita inferiormente el dominio 2 de deformación. La recta límite superior AB se obtiene de la siguiente relación de triángulos:

$$\frac{x}{0.0035} = \frac{d-x}{0.01} \Rightarrow x = \frac{7}{27}d$$

por tanto, el dominio 2 incluye todas las rectas que pasan por A y presentan leyes de deformación en flexión, con una profundidad de la línea neutra en el intervalo:

$$0 < x < \frac{7}{27}d.$$

Flexión. DOMINIO 3

Ahora las deformaciones pivotan entorno a la deformación límite $\varepsilon_c = 0.0035$

(B). La recta $x = \frac{7}{27}d$ delimita inferiormente el dominio 3 de deformación. La recta límite superior BC se obtiene de la siguiente ecuación:

$$x_{lim} = \frac{0.0035}{0.0035 + \varepsilon_y}d$$

Por tanto, el dominio 3 incluye todas las rectas que pasan por B y presentan leyes de deformación en flexión, con una profundidad de la línea neutra en el intervalo:

$$\frac{7}{27}d \leq x \leq \frac{0.0035}{0.0035 + \varepsilon_y}d.$$

Flexión. DOMINIO 4

Las deformaciones pivotan entorno a la deformación límite $\varepsilon_c = 0.0035$ (B). La recta $x = x_{lim}$ delimita inferiormente el dominio 4 de deformación. La recta límite superior BD es $x = d$, de modo que la deformación en el acero es nula. Por tanto, el dominio 4 incluye todas las rectas que pasan por B y presentan leyes de deformación en flexión, con una profundidad de la línea neutra en el intervalo: $\frac{0.0035}{0.0035 + \varepsilon_y}d < x \leq d$.

Flexión. DOMINIO 4.a

Las deformaciones pivotan entorno a la deformación límite $\varepsilon_c = 0.0035$ (B). La recta $x = d$ delimita inferiormente el dominio 4a de deformación. La recta límite superior BE es $x = h$, de modo que toda la sección resulta comprimida. Por tanto, el dominio 4.a incluye todas las rectas que pasan por B y presentan leyes de deformación en flexión, con una profundidad de la línea neutra en el intervalo: $d < x < h$.

Compresión - DOMINIO 5

Las deformaciones pivotan entorno a la deformación $\varepsilon_0 = 0.002$ en el punto G, a una profundidad dada por la siguiente relación de triángulos:

$$\frac{h}{0.0035} = \frac{x_G}{0.0035 - 0.002} \Rightarrow x_G = \frac{3}{7}h$$

De este modo, la deformación en la fibra más contraída oscila entre $0.002 \leq \varepsilon_c \leq 0.0035$. La recta $x = h$ delimita inferiormente el dominio 5 de deformación. La recta límite superior es $x = \infty$, en el caso de la compresión simple.

Tensiones y Deformaciones

Conocido el dominio de trabajo, la ley de deformaciones permite conocer a través de las leyes de comportamiento de hormigón y acero las tensiones que sufren las distintas fibras de la sección. La **Tabla III-29** recoge de forma resumida las correspondientes expresiones de las deformaciones y tensiones de las armaduras y de la fibra más contraída del hormigón. Se llama $\xi = \frac{x}{d}$ a la profundidad *reducida* de la línea neutra y $\delta_2 = \frac{d_2}{d}$ al recubrimiento mecánico reducido de la armadura menos traccionada.

III.6.5.- Ecuaciones de campo en el Estado Límite de Agotamiento Resistente

Los problemas de dimensionamiento de una sección de hormigón bajo unos esfuerzos de cálculo dados y de análisis de secciones armadas con unas cuantías de acero determinadas pueden plantearse a partir de los datos indicados en la figura siguiente.

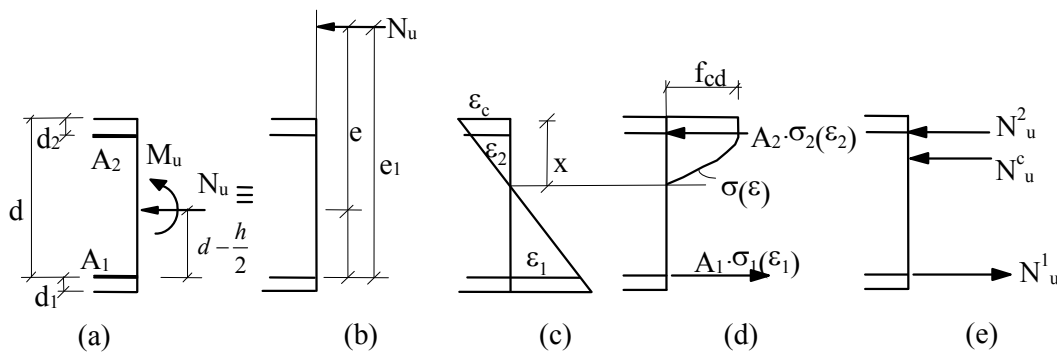


Figura III-168. Equilibrio de fuerzas y momentos en la sección

El equilibrio de fuerzas y momentos respecto de la armadura de tracción conduce a las ecuaciones siguientes:

$$N_u = N_u^1 + N_u^2 + N_u^c \tag{III-111}$$

$$M_{u1} = M_u + N_u \cdot \left(d - \frac{h}{2} \right) = M_u^1 + M_u^2 + M_u^c$$

O bien, empleando variables reducidas:

$$v_u = v_u^1 + v_u^2 + v_u^c \tag{III-112}$$

$$\mu_{u1} = \mu_u + \nu_u \cdot \left(1 - \frac{\delta}{2}\right) = \mu_u^1 + \mu_u^2 + \mu_u^c$$

donde el momento reducido y el axil reducido se han indicado en la ecuación (III-107).

La relación entre el esfuerzo axil y el momento flector se denomina excentricidad, medida respecto del centro de gravedad de la sección bruta (sin contemplar la presencia de las armaduras). Se define además la excentricidad respecto de las armaduras de acero más traccionada:

$$e = \frac{M_u}{N_u} \tag{III-113}$$

$$e_1 = d - \frac{h}{2} + e$$

De tal modo que la ecuación de equilibrio de momentos puede describirse como:

$$M_{u1} = M_u + N_u \cdot \left(d - \frac{h}{2}\right) = N_u \cdot \left(d - \frac{h}{2} + e\right) = N_u \cdot e_1 \tag{III-114}$$

Para cada dominio, se deducirán los distintos términos de esfuerzos axiles y momentos flectores reducidos debidos a las armaduras y al hormigón, en función de la variación de la profundidad de la línea neutra. Así, por ejemplo, la **Figura III-169** recoge el esquema de armado en la segunda parte del dominio 2.

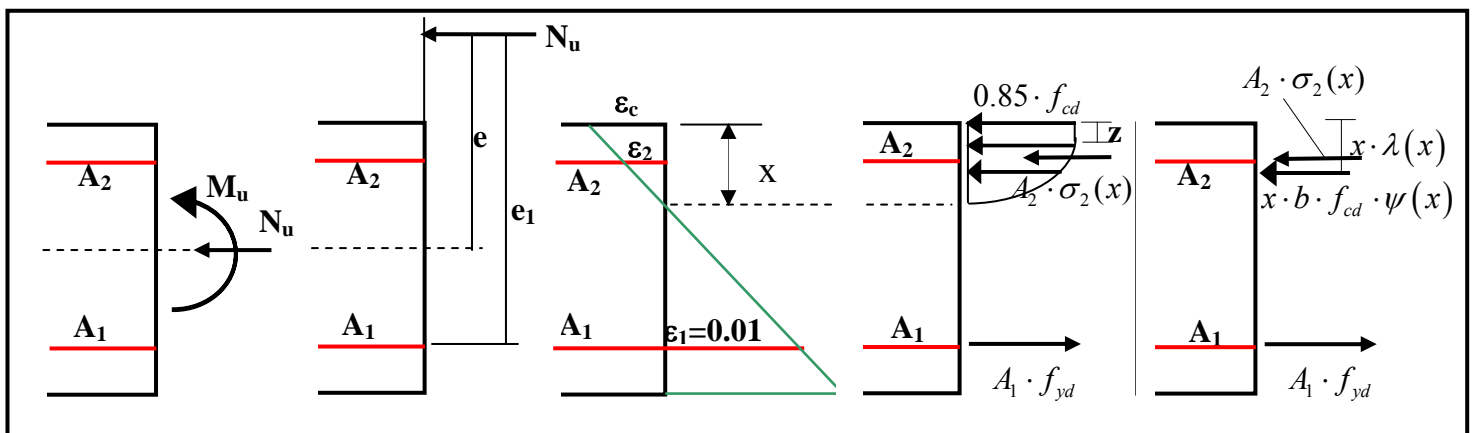


Figura III-169. Esquema de armado del dominio 2.b

Los esfuerzos últimos M_u y N_u son estáticamente equivalentes a las fuerzas en las armaduras y a la fuerza de compresión en el hormigón. En el caso del ejemplo en el dominio 2.b serían:

- Equilibrio de fuerzas
- Equilibrio de momentos respecto de la armadura A_1 :

$$N_u = -A_1 \cdot f_{yd} + A_2 \cdot \sigma_2(x) + b \cdot x \cdot f_{cd} \cdot \psi(x) \tag{III-115}$$

$$N_u \cdot e_1 = A_2 \cdot \sigma_2(x) \cdot (d - d_2) + b \cdot x \cdot f_{cd} \cdot \psi(x) \cdot (d - x \cdot \lambda(x))$$

La fuerza de compresión en el hormigón es $b \cdot x \cdot f_{cd} \cdot \psi(x)$ y la distancia entre el punto de aplicación de esta fuerza y la fibra más contraída de la sección es $x \cdot \lambda(x)$. Por tanto:

$$\begin{aligned} b \cdot x \cdot f_{cd} \cdot \psi(x) &= \int_{A_c} \sigma(x') \cdot dA_c \\ b \cdot x \cdot f_{cd} \cdot \psi(x) \cdot x \cdot \lambda(x) &= \int_{A_c} \sigma(x') \cdot x' \cdot dA_c \end{aligned} \quad \text{(III-116)}$$

Integrando estas expresiones para cada dominio se obtienen las ecuaciones de equilibrio en fuerzas y momentos que permiten el dimensionamiento de la sección. En la Tabla III-30 se recogen los términos en fuerzas y momentos del hormigón y de las armaduras de las ecuaciones de equilibrio.

La última fila de la tabla indica el número total de aligeramientos que presenta la sección.

DOM	1		2				3	4	4.a	5	
	1.1	1.2	2.1	2.2	2.3	2.4	3.1	4.1	4.a.1	5.1	5.2
ξ_0	$-\infty$	$-\left \frac{\varepsilon_y - 0.01 \cdot \delta_2}{0.01 - \varepsilon_y} \right $	0	δ_2	$\frac{1}{6}$	$\frac{\varepsilon_y + 0.01 \cdot \delta_2}{0.01 + \varepsilon_y}$	$\frac{7}{27}$	$\frac{0.0035}{0.0035 + \varepsilon_y}$	1	δ	$\frac{0.002 - \frac{3}{7} \cdot \delta \cdot \varepsilon_y}{0.002 - \varepsilon_y}$
ξ_1	$-\left \frac{\varepsilon_y - 0.01 \cdot \delta_2}{0.01 - \varepsilon_y} \right $	0	δ_2	$\frac{1}{6}$	$\frac{\varepsilon_y + 0.01 \cdot \delta_2}{0.01 + \varepsilon_y}$	$\frac{7}{27}$	$\frac{0.0035}{0.0035 + \varepsilon_y}$	1	$\frac{h}{d}$	$\frac{0.002 - \frac{3}{7} \cdot \delta \cdot \varepsilon_y}{0.002 - \varepsilon_y}$	∞
ε_c	-		$\frac{0.01 \cdot \xi}{1 - \xi}$				0.0035			$\frac{0.002 \cdot \xi}{\xi - \frac{3}{7} \cdot \delta}$	
ε_1	0.01						$0.0035 \cdot \frac{1 - \xi}{\xi}$	$0.0035 \cdot \frac{\xi - 1}{\xi}$	$\frac{0.002 \cdot (\xi - 1)}{\xi - \frac{3}{7} \cdot \delta}$		
ε_2	$0.01 \cdot \frac{\xi + \delta_2}{\xi + 1}$		$0.01 \cdot \frac{\delta_2 - \xi}{1 - \xi}$	$0.01 \cdot \frac{\xi - \delta_2}{1 - \xi}$			$0.0035 \cdot \frac{\xi - \delta_2}{\xi}$			$\frac{0.002 \cdot (\xi - \delta_2)}{\xi - \frac{3}{7} \cdot \delta}$	
σ_c	0		$< 0.85 \cdot f_{cd}$			$0.85 \cdot f_{cd}$					
σ_1	$f_{yd}(T)$							$f_{yd} \frac{\varepsilon_1}{\varepsilon_y}(T)$	$f_{yd} \frac{\varepsilon_1}{\varepsilon_y}(C)$		$f_{yd}(C)$
σ_2	$f_{yd}(T)$	$f_{yd} \frac{\varepsilon_2}{\varepsilon_y}(T)$		$f_{yd} \frac{\varepsilon_2}{\varepsilon_y}(C)$		$f_{yd}(C)$					

Tabla III-29. Deformaciones y tensiones de armaduras y hormigón

3. Los Cajones Multicelulares

DOM	1		2.a		2.b		3	4	4a	5	
	1.1	1.2	2.1	2.2	2.3	2.4				5.1	5.2
v_u^1	ω_1		$-\omega_1$				$-\omega_1 \frac{\varepsilon_1}{\varepsilon_y}$		$\omega_1 \frac{\varepsilon_1}{\varepsilon_y}$	ω_1	
μ_u^1	0		0								
v_u^2	ω_2	$\omega_2 \frac{\varepsilon_2}{\varepsilon_y}$	$\omega_2 \frac{\varepsilon_2}{\varepsilon_y}$			ω_2					
μ_u^2	$\omega_2 \cdot (1 - \delta_2)$	$\omega_2 \cdot \frac{\varepsilon_2}{\varepsilon_y} \cdot (1 - \delta_2)$	$\omega_2 \cdot \frac{\varepsilon_2}{\varepsilon_y} \cdot (1 - \delta_2)$			$\omega_2 \cdot (1 - \delta_2)$					
$\psi(\xi)$	0		$\frac{17(3-8 \cdot \xi) \cdot \xi}{12(1-\xi)^2}$		$\frac{272 \cdot \xi - 17}{300 \cdot \xi}$		$\frac{289}{420} = 0.688095$		$\frac{17}{420} \frac{1029 \cdot \xi^2 - 882 \cdot \xi \cdot \delta + 125 \cdot \delta^2}{(7 \cdot \xi - 3 \cdot \delta)^2}$		
$\lambda(\xi)$	0		$\frac{4-9 \cdot \xi}{12-32 \cdot \xi}$		$\frac{1-22 \cdot \xi + 171 \cdot \xi^2}{20 \cdot \xi \cdot (-1+16 \cdot \xi)}$		$\frac{99}{238} = 0.415966$		$\frac{3}{14} \frac{2401 \cdot \xi^2 - 2058 \cdot \xi \cdot \delta + 185 \cdot \delta^2}{1029 \cdot \xi^2 - 882 \cdot \xi \cdot \delta + 125 \cdot \delta^2}$		
v_u^c	0		$\frac{17(3-8 \cdot \xi) \cdot \xi^2}{12(1-\xi)^2}$		$\frac{(272 \cdot \xi - 17)}{300}$		$\frac{289}{420} \cdot \xi$		$\frac{17 \cdot \delta}{420} \frac{1029 \cdot \xi^2 - 882 \cdot \xi \cdot \delta + 125 \cdot \delta^2}{(7 \cdot \xi - 3 \cdot \delta)^2}$		
μ_u^c	0		$\frac{17(4-12 \cdot \xi + 3 \cdot \xi^2) \cdot \xi^2}{16(1-\xi)^2}$		$\frac{-17}{2000}(7-114 \cdot \xi + 57 \cdot \xi^2)$		$\frac{17}{5880} \cdot \xi \cdot (238 - 99 \cdot \xi)$		$\frac{17}{5880} \left(147 \cdot \delta \cdot (2 - \delta) + \frac{128 \cdot \delta^3 \cdot (6 \cdot \delta - 7)}{(7 \cdot \xi - 3 \cdot \delta)^2} \right)$		

Tabla III-30. Fuerzas y momentos de las armaduras y del hormigón para cada dominio

Nota: En el dominio 1 el axil es positivo de tracción. En el resto es positivo de compresión.

Equilibrio de fuerzas: $v_u = v_u^1 + v_u^2 + v_u^c$

Equilibrio de momentos: $\mu_{u1} = \mu_{u1}^1 + \mu_{u1}^2 + \mu_{u1}^c$

III.6.6.- Dimensionado en el Estado Límite de Agotamiento Resistente por flexión

En el apartado anterior se han expresado las dos ecuaciones de equilibrio en los distintos dominios que ligan las cuantías mecánicas de las armaduras con los esfuerzos de cálculo, que en el Estado Límite de Agotamiento Resistente se hacen coincidir con los últimos resistidos por la sección. Además de las dos incógnitas de las áreas de acero, existe una tercera, la profundidad de la línea neutra, que debe ser fijada siguiendo algún criterio.

En función de este criterio pueden adoptarse distintas soluciones de dimensionamiento. La que se va a utilizar es el conocido como criterio de *armado óptimo*, aquél en el que se consigue que, para los esfuerzos dados, los materiales trabajen lo más cerca posible de sus capacidades resistentes máximas.

La profundidad óptima de la línea neutra se obtendrá minimizando la cantidad de acero dispuesto en la sección, suponiendo que trabaja en el dominio 3:

$$\xi_{\text{opt}} = \frac{1 + \delta_2}{4 \cdot \lambda} = \frac{119}{198} \cdot (1 + \delta_2) \quad \xi_{\text{opt}} \leq \frac{0.0035}{0.0035 + \varepsilon_y} \quad (\text{III-117})$$

El momento resistido por el hormigón en la situación óptima es:

$$\mu_{u1,\text{opt}}^c = \xi_{\text{opt}} \cdot \psi \cdot (1 - \xi_{\text{opt}} \cdot \lambda) \quad (\text{III-118})$$

Determinación de las cuantías mecánicas

En el caso de momentos reducidos de cálculo μ_{d1} inferiores a este valor $\mu_{u1,\text{opt}}^c$, el hormigón por sí solo es capaz de resistir dicho momento, sin la necesidad de la armadura de compresión, por lo que el armado óptimo se consigue en el dominio 2 o en el 3 *sin armadura de compresión*.

Conocido el dominio, se determinará la profundidad de la línea neutra a partir de la ecuación de equilibrio en momentos:

$$\mu_{d1} = \xi \cdot \psi(\xi) \cdot (1 - \xi \cdot \lambda(\xi)) \rightarrow \xi^* \quad (\text{III-119})$$

Conocida la profundidad de la línea neutra, la cuantía de tracción será:

$$\omega_1 = \xi^* \cdot \psi(\xi^*) - \nu_d \quad (\text{III-120})$$

En el caso de que la cuantía resulte negativa, se dice que la sección está *sobredimensionada*. Por sí mismo el hormigón es capaz de resistir los esfuerzos solicitantes.

Por otra parte, en el caso de que el momento reducido de cálculo μ_{d1} supere el valor del momento óptimo, para seguir manteniendo la armadura de tracción en fluencia, es decir, sin entrar en el dominio 4, habrá que disponer armadura de compresión, que también estará en fluencia, si se mantiene el armado en el punto de diseño óptimo $\xi = \xi_{opt}$. Ahora las ecuaciones se transforman en:

$$\nu_d = -\omega_1 + \omega_2 + \xi_{opt} \cdot \psi \quad \mu_{d1} = \omega_2 \cdot (1 - \delta_2) + \xi_{opt} \cdot \psi \cdot (1 - \xi_{opt} \cdot \lambda) \quad (\text{III-121})$$

donde las únicas incógnitas son las cuantías:

$$\omega_2 = \frac{\mu_{d1} - \mu_{u1,opt}^c}{(1 - \delta_2)} \quad \omega_1 = \omega_2 + \xi_{opt} \cdot \psi - \nu_d \quad (\text{III-122})$$

Por otro lado, en el dominio 1, el armado óptimo se consigue haciendo que las dos armaduras trabajen a la tensión de cálculo, en tracción. Para lo cual, la profundidad mínima de la línea neutra requerida, en valor absoluto es:

$$\xi = \frac{\varepsilon_y - 0.01 \cdot \delta_2}{\varepsilon_y - 0.01} \quad (\text{III-123})$$

Entonces:

$$\begin{aligned} \nu_d &= \omega_1 + \omega_2 \quad \mu_{d1} = \omega_2 \cdot (1 - \delta_2) \\ \omega_2 &= \frac{\mu_{d1}}{(1 - \delta_2)} \quad \omega_1 = \omega_2 - \nu_d \end{aligned} \quad (\text{III-124})$$

Determinación del armado

La solución alcanzada en términos de cuantías de acero tiene equivalencia inmediata con el área de acero a disponer. Así, para ambas armaduras:

$$A = \frac{\omega \cdot b \cdot d \cdot f_{cd}}{f_{yd}} \quad (\text{III-125})$$

Sin embargo, hay que determinar el número de redondos que se colocarán, utilizando para ello los diámetros de la serie comercial indicada en la Instrucción EHE-08, que en su Artículo 32 indica que los diámetros nominales de las barras corrugadas se ajustarán a la serie siguiente: 6-8-10-12-14-16-20-25-32 y 40 mm. Así, si ϕ es el diámetro de la barra a emplear, se tomará el valor entero superior:

$$n = E \left[\frac{4 \cdot A}{\pi \cdot \phi^2} \right] + 1 \quad (\text{III-126})$$

Partiendo de la experiencia adquirida en otros proyectos similares, se suele emplear como diámetros de armado de las gargantas $\phi 12$ para las interiores y $\phi 16$ para las exteriores, tanto en horizontal como en vertical.

Cuantías mínimas

Existen dos razones que obligan a mantener unas cuantías mínimas en flexión. La primera es la de incrementar la seguridad de piezas que presenten rotura frágil. La segunda es la de disponer de una cantidad suficiente de armadura para controlar la fisuración.

La EHE-08 indica para el problema de flexión simple o compuesta que la cuantía de acero mínima debe ser $A_1 \geq 0.04 \cdot \frac{f_{cd}}{f_{yd}} \cdot A_c$.

En los cajones multicelulares con celdas interiores circulares achaflanadas hay dos tipologías estructurales de referencia con comportamientos básicos bien diferenciados:

- Tipo 1: Placas empotradas en tres bordes en paredes exteriores
- Tipo 2: Cilindros empotrados en la base en celdas interiores

En ambos casos cuando las cargas son repartidas, uniformes o triangulares, los momentos de diseño verticales negativos son máximos en el borde inferior empotrado y los positivos los son a una distancia que se puede acotar entre dos y tres metros del mismo (el sentido positivo y negativo del esfuerzo depende del de la carga aplicada). En el resto de la altura los momentos verticales pasan de positivos a negativos pero con valores muy pequeños o por lo menos mucho más pequeños que los anteriores. La respuesta horizontal es totalmente diferente y por ello queda fuera de estas reflexiones.

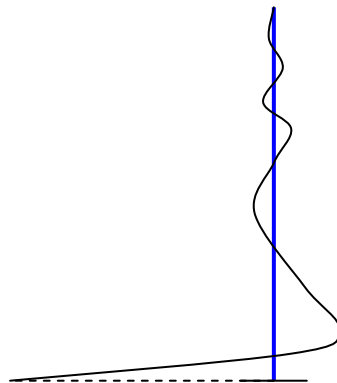


Figura III-170. Gráfica habitual de la ley de momentos flectores en paredes verticales

Ello lleva a considerar, de acuerdo con la normativa vigente y teniendo en cuenta la confianza de los esfuerzos verticales de diseño asociados con los distintos casos de carga aplicados, que se satisfacen las condiciones establecidas por la norma que permiten reducir las cuantías mínimas verticales. Esta cuestión podría acarrear ciertas dudas cuando la simulación del comportamiento de la estructura se realiza con modelos estructurales simples o simplificados, placas y tubos por ejemplo, pero se

refuerza cuando se constata este efecto mediante un cálculo tridimensional más complejo.

De hecho, estudios tridimensionales no lineales [16] demuestran, a partir del índice de daño en el hormigón, que no sería necesaria la disposición de armaduras verticales, desde un punto de vista mecánico.

Se admite, por tanto, para la armadura vertical de las celdas interiores, una reducción de armadura mínima mediante el coeficiente $\alpha = 1.5 - 12.5 \cdot \frac{A_1 \cdot f_{yd}}{A_c \cdot f_{cd}}$. Si bien, se reforzarán los primeros metros, susceptibles de quedar sometidos a mayores esfuerzos por causas no previstas.

Para compresión simple o compuesta, con ambas armaduras en compresión, se debe cumplir para cada una de ellas: $0.05 \cdot N_d \leq A_s \cdot f_{yc,d} \leq 0.5 \cdot f_{cd} \cdot A_c$, siendo $f_{yc,d} = f_{yd} \leq 400 \text{ MPa}$. Si el armado es simétrico el área total de las armaduras cumplirá: $0.1 \cdot N_d \leq A_s \cdot f_{yc,d} \leq f_{cd} \cdot A_c$

Por otro lado, la misma Norma indica en su Tabla 42.3.5 la cuantía geométrica de tracción mínima para aceros de B 400 S de $A_{1,\min} = 0.004 \cdot A_c$ en el caso de pilares, $A_{1,\min} = 0.0033 \cdot A_c$ para vigas, $A_{1-2,\min} = 0.004 \cdot A_c$ en muros en posición horizontal, a repartir en ambas caras al 50%. Por último, $A_{1,\min} = 0.0012 \cdot A_c$ en muros en posición vertical.

La cuantía de compresión mínima es el 30% de la mínima de tracción.

Distancias máximas y mínimas entre armaduras

La EHE-08 da unas limitaciones en lo relativo a las separaciones máximas y mínimas permisibles entre los redondos que constituyen una misma armadura. Las distancias mínimas están dadas en base a un correcto hormigonado de la pieza, de modo que todas las barras o grupos de barras queden perfectamente envueltas por el hormigón. De otra parte, se establecen unas distancias máximas atendiendo a lograr una distribución lo más uniforme posible de las barras de una misma armadura y para que no haya concentraciones en unas zonas y otras se queden sin armar.

Con carácter general, para barras aisladas (en casos de grupos de barras se tratarán como una barra con un diámetro equivalente), la distancia entre dos redondos consecutivos debe ser mayor o igual que el mayor de los tres valores siguientes:

- Dos centímetros.
- El diámetro del redondo que sea mayor.
- 1.25 veces el tamaño máximo del árido.

y menor o igual que el menor de los dos valores siguientes:

- 30 centímetros

- Tres veces el espesor bruto de la parte de la sección del elemento, alma o alas, en las que vayan situados.

III.6.7.- Dimensionado en el Estado Límite de Agotamiento Resistente por cortante

El cálculo de las armaduras a disponer en las piezas para resistir el esfuerzo cortante o, si no son necesarias dichas armaduras, las comprobaciones a realizar, se efectúa aplicando las recomendaciones recogidas en el Artículo 44 de la EHE. Concretamente, para que una sección rectangular no se agote por tensiones tangenciales debidas al esfuerzo cortante, si en la sección actúa un cortante efectivo V_d , éste debe satisfacer simultáneamente las dos ecuaciones siguientes:

$$V_d \leq V_{u1} = 0.30 f_{cd} b d \quad (\text{III-127})$$

$$V_d \leq V_{u2} = \begin{cases} 0.12 \xi (100 \rho_l f_{ck})^{1/3} b d & \text{en piezas sin armadura de cortante} \\ 0.10 \xi (100 \rho_l f_{ck})^{1/3} b d + 0.90 A_{90} f_{y90,d} & \text{en piezas con armadura de cortante} \end{cases}$$

donde se ha supuesto que se disponen cercos verticales con un área A_{90} y que no existen armaduras activas. ρ_l es la cuantía geométrica de la armadura traccionada de flexión y

$$\xi = 1 + \sqrt{\frac{200}{d}} \quad (\text{III-128})$$

con d en milímetros. En las ecuaciones anteriores, V_{u1} es el cortante de agotamiento por compresión oblicua en el alma y V_{u2} es el cortante de agotamiento por tracción en el alma. La comprobación correspondiente al agotamiento por tracción en el alma se efectúa para una sección situada a una distancia de un canto útil del borde del apoyo directo. En piezas que no necesiten armadura de cortante no es necesaria la comprobación por compresión oblicua en el alma.

El proceso de cálculo del área necesaria es sencillo. Primeramente se supone que no es necesaria la armadura de cortante y se realiza la comprobación adecuada. Si no es satisfactoria, se supone que sí debe llevar armadura y su área A_{90} se despeja de la comprobación correspondiente, debiéndose comprobar, además, que no se agota el alma por compresión oblicua.

La cuantía mínima necesaria que establece la EHE debe ser tal que cumpla la relación

$$\sum A_{90} f_{y90,d} \geq 0.02 f_{cd} b \quad (\text{III-129})$$

Respecto de la disposición de las armaduras de cortante, la colocación de cercos o estribos se prolongará más allá de la sección en la que teóricamente dejen de ser necesarios en una longitud igual a medio canto de la pieza, y la separación s_t entre los

mismos deberá cumplir las siguientes condiciones (al objeto de asegurar un adecuado confinamiento del hormigón):

$$s_t \leq \begin{cases} 0.75 d < 300 \text{ mm} & \text{si } V_d \leq V_{u1} / 5 \\ 0.60 d < 300 \text{ mm} & \text{si } V_d / 5 \leq 2V_{u1} / 3 \\ 0.30 d < 200 \text{ mm} & \text{si } V_d > 2V_{u1} / 3 \end{cases} \quad (\text{III-130})$$

III.6.8.- Obtención de los resultados de armado

Establecidas las ecuaciones de armado de cada sección rectangular, se procede a determinar el armado en todas las gargantas para todas y cada una de las hipótesis de cálculo resueltas.

Las clases `CArmadoHorizontal` y `CArmadoVertical`

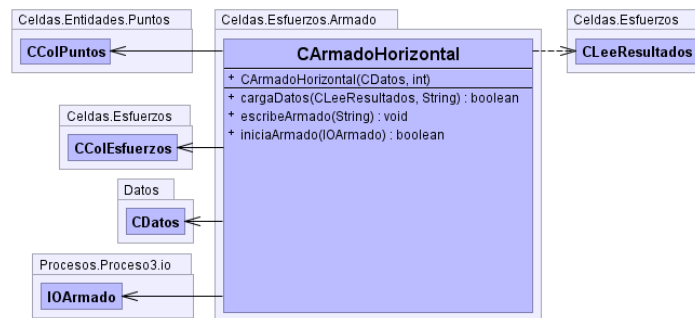


Figura III-171. Diagrama UML de la clase `CArmadoHorizontal`

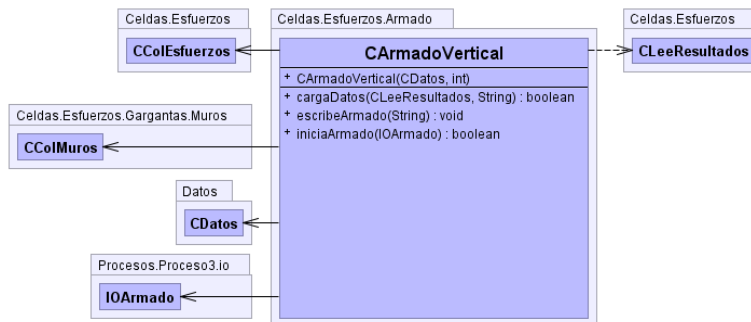


Figura III-172. Diagrama UML de la clase `CArmadoVertical`

Una vez realizado el armado horizontal de las gargantas verticales, se escribe en ficheros `*.armVert.txt`, indicando para cada cota de altura:

Esfuerzo axial, Momento flector, axial reducido, flector reducido, cuantía 1, cuantía 2, cuantía mínima 1, cuantía mínima 2, área 1, área 2, área mínima 1, área mínima 2, número de redondos 1, número de redondos 2, separación 1, separación 2, signo.

```

Número de Gargantas: 196
Número de Rebanadas: 30
Garganta: 1, Tipo: 2
Rebanadas: 30
1 -94744.58090624967 3910.8092375000256 -0.033384277979651183 0.004005860290472386 0.0
0.0 0.07190203347938438 0.03595101673969219 0.0 0.0 4.69333333333333175E-4
2.34666666666666587E-4 5.0 3.0 0.1115 0.235 1.0
2 -112277.538840421834 8488.585692593751 -0.03784210933745157 0.008316859184159302 0.0
0.0 0.07190203347938438 0.03595101673969219 0.0 0.0 4.9066666666666649E-4
2.4533333333333243E-4 5.0 3.0 0.10299999999999998 0.17966666666666664 1.0
...
Garganta: 196, Tipo: 3
Rebanadas: 29
1 -95521.74912499796 105.21335859363126 -0.08040551273148216 6.150238413863023E-4 0.0
0.0 0.06441223832528235 0.06441223832528235 0.0 0.0 1.75999999999999625E-4
1.75999999999999625E-4 2.0 2.0 0.492 0.492 1.0
2 -150697.9388281218 181.5250742185624 -0.12133489438657512 0.0010149684325157298 0.0
0.0 0.06441223832528235 0.06441223832528235 0.0 0.0 1.83999999999999607E-4
1.83999999999999607E-4 2.0 2.0 0.27549999999999997 0.27549999999999997 1.0
...
29 4230.361555468655 -186.83084763670666 0.003406088208912066 -0.0010446348163620394
0.0 0.003409975090412369 0.06441223832528235 0.06441223832528235 0.0 9.74093484327167E-
6 1.83999999999999585E-4 1.83999999999999585E-4 2.0 2.0 0.2754999999999963
0.2754999999999963 -1.0
    
```

Tabla III-31. Fichero de armado horizontal de gargantas verticales

Una vez realizado el armado vertical de las gargantas horizontales, se escribe en ficheros *.armHorz.txt, indicando para cada cota de altura:

Esfuerzo axil, Momento flector, axil reducido, flector reducido, cuantía 1, cuantía 2, cuantía mínima 1, cuantía mínima 2, área 1, área 2, área mínima 1, área mínima 2, número de redondos 1, número de redondos 2, separación 1, separación 2, signo.

```

Número de Muros: 205
Número de Alturas: 31
Muro: 1 Tipo: 2
1 -491905.3531983867 13560.912639952652 -0.04926492535712803 0.004090789332842192 0.0
0.0 0.03143006809848088 0.009429020429544265 0.0 0.0 7.2179999999999988E-4
2.1653999999999963E-4 7.0 2.0 0.2744285714285709 0.9904999999999982 1.0
2 -552856.0207006088 7655.403399643029 -0.055369209576521534 0.0023093314880297685 0.0
0.0 0.03143006809848088 0.009429020429544265 0.0 0.0 7.2179999999999988E-4
2.1653999999999963E-4 7.0 2.0 0.2744285714285709 0.9904999999999982 1.0
...
31 -975.3487166866998 -31.90603493913819 -9.7682372050466637E-5 -9.624784912910738E-6
0.0 0.0 0.009429020429544265 0.03143006809848088 0.0 0.0 2.1653999999999963E-4
7.2179999999999988E-4 2.0 7.0 0.9904999999999982 0.2744285714285709 -1.0
Muro: 2 Tipo: 2
1 -556126.8720353884 131197.87100534435 -0.030100286430649187 0.021388752327750962
0.004397558288411325 0.0 0.03143006809848088 0.009429020429544265 1.868713370775689E-4
0.0 0.0013356000000000002 4.0068000000000005E-4 12.0 4.0 0.29716666666666663 0.9155 1.0
2 -694840.2895445216 58522.39867891045 -0.03760813006984929 0.00954071191382463 0.0 0.0
0.03143006809848088 0.009429020429544265 0.0 0.0 0.0013356000000000002
4.0068000000000005E-4 12.0 4.0 0.29716666666666663 0.9155 1.0
...
Muro: 205 Tipo: 5
1 -701498.393318107 -4145.695187821711 -0.0485556154848888 -3.0478235886904486E-4 0.0
0.0 0.008407952157750124 0.02802650719250041 0.0 0.0 2.793864150000009E-4
9.312880500000028E-4 3.0 9.0 0.32900000000000107 0.10166666666666703 -1.0
2 -717354.2449524069 -10990.563379278385 -0.049653109994461236 -8.080019587248692E-4
0.0 0.0 0.008407952157750124 0.02802650719250041 0.0 0.0 2.793864150000009E-4
9.312880500000028E-4 3.0 9.0 0.32900000000000107 0.10166666666666703 -1.0
...
31 -968.5812022335197 -314.4246514172472 -6.704228672440519E-5 -2.3115806301205343E-5
0.0 0.0 0.008407952157750124 0.02802650719250041 0.0 0.0 2.793864150000009E-4
9.312880500000028E-4 3.0 9.0 0.32900000000000107 0.10166666666666703 -1.0
    
```

Tabla III-32. Fichero de armado vertical de gargantas horizontales

Finalmente se realiza la unificación de armadura horizontal y vertical, tomando el máximo de acero obtenido por alturas en gargantas del mismo tipo, que se armarán del mismo modo.

Para cada garganta y altura se obtienen datos de armado horizontal:

3. Los Cajones Multicelulares

- **cm²/m**: área de acero perimetral o circunferencial a disponer por metro de altura.
- **cm²**: área de acero perimetral o circunferencial a disponer en cada tramo de altura definido.
- **nFi/m**: número de barras, sin redondear, a disponer por metro de altura.
- **nFi**: número de barras, sin redondear, a disponer en cada tramo de altura definido.
- **kg/m**: kg de acero a disponer en cada m de altura, sin considerar las longitudes de solape y anclaje de armaduras.

Con todos los resultados obtenidos en cada subgarganta se realiza un agrupamiento en varios tramos de altura, de manera que se ofrecen resultados cada 3 ó 4 m, con consideración especial en el arranque del fuste (2 a 3 m).

	Cota 0,55 - 1,10		Cota 1,10 - 1,68		Cota 1,68 - 2,25		Cota 2,25 - 2,82		Cota 2,82 - 3,40		Cota 3,40 - 3,98		Cota 3,98 - 4,55		Cota 4,55 - 5,13		Cota 5,13 - 5,70		Cota 5,70 - 6,28		
	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	
cm²/m																					
LadoTierra	8,53	8,53	8,54	8,54	8,54	8,54	8,54	8,54	8,54	8,96	8,54	9,51	8,54	9,62	8,54	9,46	8,54	9,17	8,54	8,82	
LadoMar	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	8,5	
Lateral	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	
Circular	3,20		3,20		3,20		3,20		3,20		3,20		3,20		3,20		3,20		3,20		
CircularCha	3,20		3,20		3,20		3,20		3,20		3,20		3,20		3,20		3,20		3,20		
CircularOvo	3,20		3,20		3,20		3,20		3,20		3,20		3,20		3,20		3,20		3,20		
CircularOvoCha	3,56		3,57		3,57		3,57		3,57		3,57		3,57		3,57		3,57		3,57		
Ovoide	3,56		3,57		3,57		3,57		3,57		3,57		3,57		3,57		3,57		3,57		
cm²																					
LadoTierra	4,69	4,69	4,91	4,91	4,91	4,91	4,91	4,91	4,91	5,15	4,91	5,47	4,91	5,53	4,91	5,44	4,91	5,27	4,91	5,07	
LadoMar	4,7	4,7	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	4,9	
Lateral	2,9	2,9	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	3,1	
Circular	1,76		1,84		1,84		1,84		1,84		1,84		1,84		1,84		1,84		1,84		
CircularCha	1,76		1,84		1,84		1,84		1,84		1,84		1,84		1,84		1,84		1,84		
CircularOvo	1,76		1,84		1,84		1,84		1,84		1,84		1,84		1,84		1,84		1,84		
CircularOvoCha	1,96		2,05		2,05		2,05		2,05		2,05		2,05		2,05		2,05		2,05		
Ovoide	1,96		2,05		2,05		2,05		2,05		2,05		2,05		2,05		2,05		2,05		
nFi/m																					
LadoTierra	7,5	7,5	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,9	7,6	8,4	7,6	8,5	7,6	8,4	7,6	8,1	7,6	7,8	
LadoMar	7,5	7,5	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	
Lateral	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	
Circular	2,8		2,8		2,8		2,8		2,8		2,8		2,8		2,8		2,8		2,8		
CircularCha	2,8		2,8		2,8		2,8		2,8		2,8		2,8		2,8		2,8		2,8		
CircularOvo	2,8		2,8		2,8		2,8		2,8		2,8		2,8		2,8		2,8		2,8		
CircularOvoCha	3,2		3,2		3,2		3,2		3,2		3,2		3,2		3,2		3,2		3,2		
Ovoide	3,2		3,2		3,2		3,2		3,2		3,2		3,2		3,2		3,2		3,2		
nFi																					
LadoTierra	4,10	4,10	4,30	4,30	4,30	4,30	4,30	4,30	4,30	4,60	4,30	4,80	4,30	4,90	4,30	4,80	4,30	4,70	4,30	4,50	
LadoMar	4,1	4,1	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	
Lateral	2,6	2,6	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	2,7	
Circular	1,60		1,60		1,60		1,60		1,60		1,60		1,60		1,60		1,60		1,60		
CircularCha	1,60		1,60		1,60		1,60		1,60		1,60		1,60		1,60		1,60		1,60		
CircularOvo	1,60		1,60		1,60		1,60		1,60		1,60		1,60		1,60		1,60		1,60		
CircularOvoCha	1,70		1,80		1,80		1,80		1,80		1,80		1,80		1,80		1,80		1,80		
Ovoide	1,70		1,80		1,80		1,80		1,80		1,80		1,80		1,80		1,80		1,80		
kg/m																					
LadoTierra	6,66	6,66	6,75	6,75	6,75	6,75	6,75	6,75	6,75	7,01	6,75	7,46	6,75	7,55	6,75	7,46	6,75	7,19	6,75	6,92	
LadoMar	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	6,7	
Lateral	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	
Circular	79,55		79,55		79,55		79,55		79,55		79,55		79,55		79,55		79,55		79,55		
CircularCha	39,77		39,77		39,77		39,77		39,77		39,77		39,77		39,77		39,77		39,77		
CircularOvo	19,89		19,89		19,89		19,89		19,89		19,89		19,89		19,89		19,89		19,89		
CircularOvoCha	11,36		11,36		11,36		11,36		11,36		11,36		11,36		11,36		11,36		11,36		
Ovoide	11,36		11,36		11,36		11,36		11,36		11,36		11,36		11,36		11,36		11,36		

Figura III-173. Ejemplo de tabla resumen de armado horizontal por alturas y tipos de garganta

resumen fi/m	fi	uds	LonExt	LonInt	0,55	2,5	2,5	6,5	6,5	10,5	10,5	14,5	14,5	17,5	kg ext	0,55-2,5	kg ext	2,5-6,5	kg ext	6,5-10,5
					ext	int	ext	int	ext	int	ext	int	ext	int		ext		int		
LadoTierra	12	1	43,99	39,25	7,6	7,6	7,6	8,5	7,6	7,6	7,6	7,6	7,6	7,6	578,8	516,4	1187,2	1184,8	1187,2	1059,3
LadoMar	12	1	50,58	39,25	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	7,6	665,5	516,4	1365,1	1059,3	1365,1	1059,3
Lateral	12	2	18,40	22,53	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,8	299,4	366,6	614,2	752,1	614,2	752,1
Circular	12	32	11,38		2,8		2,8		2,8		2,8		2,8	7,6	1765,2		3620,9		3620,9	
CircularCha	12	16	10,69		2,8		2,8		2,8		2,8		2,8	10,0	829,1		1700,7		1700,7	
CircularOvo	12	8	11,38		2,8		2,8		2,8		2,8		2,8	9,1	441,3		905,2		905,2	
CircularOvoCha	12	4	10,69		3,2		3,2		3,2		3,2		3,2	10,1	236,9		485,9		485,9	
Ovoide	12	4	7,90		3,2		3,2		3,2		3,2		3,2	7,0	175,1		359,1		359,1	
hasta 17,20															kg Acero:	6390,7	kg Acero:	13234,5	kg Acero:	13109,1
																		m3 hor fuste	2874,58	

Figura III-174. Ejemplo de tabla resumen de armado horizontal por tramos unificados y tipos de garganta

Del mismo modo, para cada garganta y altura se obtienen datos de armado vertical:

3. Los Cajones Multicelulares

- **cm²/m**: área de acero vertical a disponer por metro de pared o de celda aligerada.
- **cm²**: área de acero total a disponer en cada pared o entorno a cada celda aligerada.
- **nFi/m**: número de barras, sin redondear, a disponer en cada altura por metro de pared o de celda aligerada.
- **nFi**: número de barras, sin redondear, a disponer en cada altura en cada pared o entorno a cada celda aligerada.
- **kg/m**: kg de acero a disponer en cada altura por metro de pared o de celda aligerada (este dato es orientativo y puede variar con las longitudes de solape y anclaje de armaduras).

	Cota 0,55		Cota 1,10		Cota 1,68		Cota 2,25		Cota 2,82		Cota 3,40		Cota 3,98		Cota 4,55		Cota 5,13		
	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	
cm²/m																			
LadoTierra	10,07	7,36	7,36	7,36	7,36	7,36	7,36	7,36	7,36	7,36	7,36	7,36	7,36	7,36	7,36	7,36	7,36	7,36	
LadoMar	8,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	7,4	
Lateral	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	4,6	
Circular	3,68		3,68		3,68		3,68		3,68		3,68		3,68		3,68		3,68		
CircularLat	3,68		3,68		3,77		3,68		3,68		3,68		3,68		3,68		3,68		
ChaffanTierra	3,68		3,68		3,68		3,68		3,68		3,68		3,68		3,68		3,68		
EsquinaTierra	3,68		3,68		3,68		3,68		3,68		3,68		3,68		3,68		3,68		
ChaffanMar	3,68		3,68		3,68		3,68		3,68		3,68		3,68		3,68		3,68		
EsquinaMar	3,68		5,85		5,16		4,94		4,83		4,72		4,32		3,94		3,68		
Ovoide	3,68		5,85		5,16		4,94		4,83		4,72		4,32		3,94		3,68		
EsquinaOvo	3,68		3,68		3,68		3,68		3,68		3,68		3,68		3,68		3,68		
Topes	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	18,61	
cm²																			
LadoTierra	395,29	288,97	288,97	288,97	288,97	288,97	288,97	288,97	288,97	288,97	288,97	288,97	288,97	288,97	288,97	288,97	288,97	288,97	
LadoMar	330,2	289,0	289,0	289,0	289,0	289,0	289,0	289,0	289,0	289,0	289,0	289,0	289,0	289,0	289,0	289,0	289,0	289,0	
Lateral	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	84,7	
Circular	41,61		41,61		41,61		41,61		41,61		41,61		41,61		41,61		41,61		
CircularLat	41,65		41,65		42,59		41,65		41,65		41,65		41,65		41,65		41,65		
ChaffanTierra	39,31		39,31		39,31		39,31		39,31		39,31		39,31		39,31		39,31		
EsquinaTierra	39,32		39,32		39,32		39,32		39,32		39,32		39,32		39,32		39,32		
ChaffanMar	39,31		39,31		39,31		39,31		39,31		39,31		39,31		39,31		39,31		
EsquinaMar	39,32		62,52		55,09		52,77		51,61		50,45		46,12		42,09		39,32		
Ovoide	28,82		45,79		40,35		38,65		37,80		36,95		33,78		30,83		28,82		
EsquinaOvo	24,38		24,38		24,38		24,38		24,38		24,38		24,38		24,38		24,38		
Topes	52,49	61,29	52,49	61,29	52,49	61,29	52,49	61,29	52,49	61,29	52,49	61,29	52,49	61,29	52,49	61,29	52,49	61,29	
nFi/m																			
LadoTierra	8,9	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	
LadoMar	7,4	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	
Lateral	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	
Circular	4,7		4,7		4,7		4,7		4,7		4,7		4,7		4,7		4,7		
CircularLat	4,7		4,7		4,8		4,7		4,7		4,7		4,7		4,7		4,7		
ChaffanTierra	4,7		4,7		4,7		4,7		4,7		4,7		4,7		4,7		4,7		
EsquinaTierra	4,7		4,7		4,7		4,7		4,7		4,7		4,7		4,7		4,7		
ChaffanMar	4,7		4,7		4,7		4,7		4,7		4,7		4,7		4,7		4,7		
EsquinaMar	4,7		7,4		6,6		6,3		6,1		6,0		5,5		5,0		4,7		
Ovoide	4,7		7,4		6,6		6,3		6,1		6,0		5,5		5,0		4,7		
EsquinaOvo	4,7		4,7		4,7		4,7		4,7		4,7		4,7		4,7		4,7		
Topes	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	
nFi																			
LadoTierra	350,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	256,00	
LadoMar	292,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	256,0	
Lateral	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	75,0	
Circular	53,00		53,00		53,00		53,00		53,00		53,00		53,00		53,00		53,00		
CircularLat	54,00		54,00		55,00		54,00		54,00		54,00		54,00		54,00		54,00		

Figura III-175. Ejemplo de tabla resumen de armado vertical por alturas y tipos de garganta

resumen	fi	uds	LonExt	LonInt	0,55 ext	2,5 int	2,5 ext	6,5 int	6,5 ext	10,5 int	10,5 ext	14,5 int	14,5 ext	17,5 int	kg ext	0,55-2,5 int	kg ext	2,5-6,5 int
LadoTierra	12	1	39,25	39,25	8,9	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	604,7	441,7	906,0	906,0
LadoMar	12	1	39,25	39,25	7,4	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5	502,8	441,7	906,0	906,0
Lateral	12	2	18,4	22,53	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	4,1	261,2	319,8	535,8	656,1
Circular	10	32	11,31		4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	2045,0		4194,9	
CircularLat	10	8	11,31		4,8	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	522,1		1048,7	
ChaffanTierra	10	8	10,69		4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	483,2		991,2	
EsquinaTierra	10	2	10,69		4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	120,8		247,8	
ChaffanMar	10	8	10,69		4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	483,2		991,2	
EsquinaMar	10	2	10,69		7,4	6,3	6,3	6,3	6,3	6,3	6,3	6,3	6,3	6,3	190,2		332,2	
Ovoide	10	4	7,83		7,4	6,3	6,3	6,3	6,3	6,3	6,3	6,3	6,3	6,3	278,6		486,6	
EsquinaOvo	10	2	6,63		4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	4,7	74,9		153,7	
Topes	16	4	2,82		9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	322,9		662,3	
Doble Tope	16	2	3,29		9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	9,3	188,3		386,3	
hasta 17,20															kg Acero: 7281,3		kg Acero: 14310,7	

Figura III-176. Ejemplo de tabla resumen de armado vertical por tramos unificados y tipos de garganta

Tal y como se ha mencionado, la solera y los voladizos se arman conforme a los resultados de esfuerzos máximos en determinadas secciones de estudio.

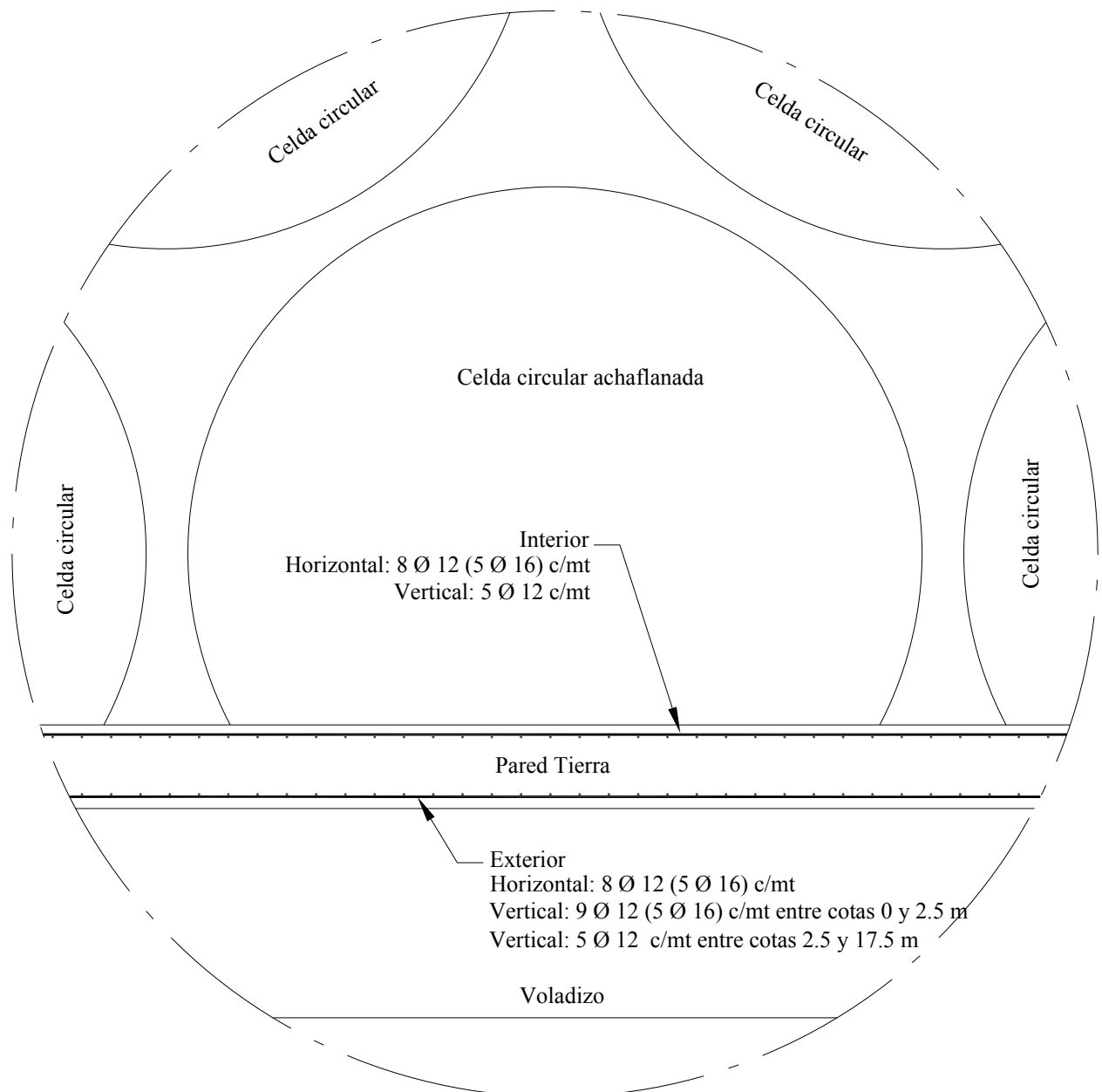
3. Los Cajones Multicelulares

Longitudinal		Transversal		Voladizos
Superior	Inferior	Superior	Inferior	Inferior
12.06 cm ² /m	12.06 cm ² /m	12.06 cm ² /m	12.06 cm ² /m	18.56 cm ² /m
6φ16 pm	6φ16 pm	6φ16 pm	6φ16 pm	6φ16 pm + 4φ16 pm // 6φ12 pm // 9φ10 pm

Figura III-177. Ejemplo de tabla resumen de armado longitudinal de losa y voladizos

Esquemas de armado

Una vez que se han obtenido los resultados finales de armado, se procede a generar los esquemas generales de armado por tipologías de celdas o paramentos.



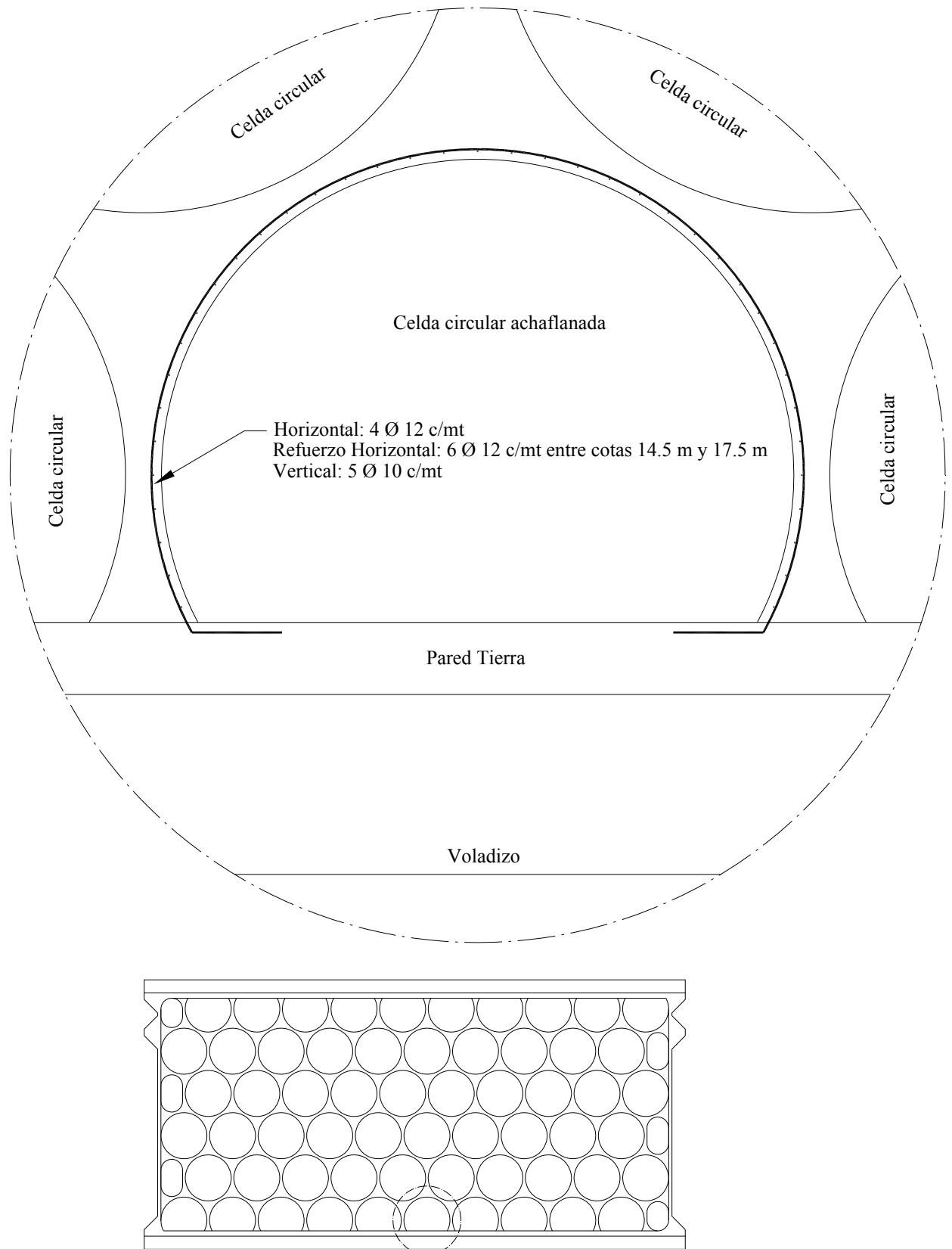


Figura III-178. Ejemplo de esquema de armado en pared y celdas circulares achaflanadas del Lado Tierra

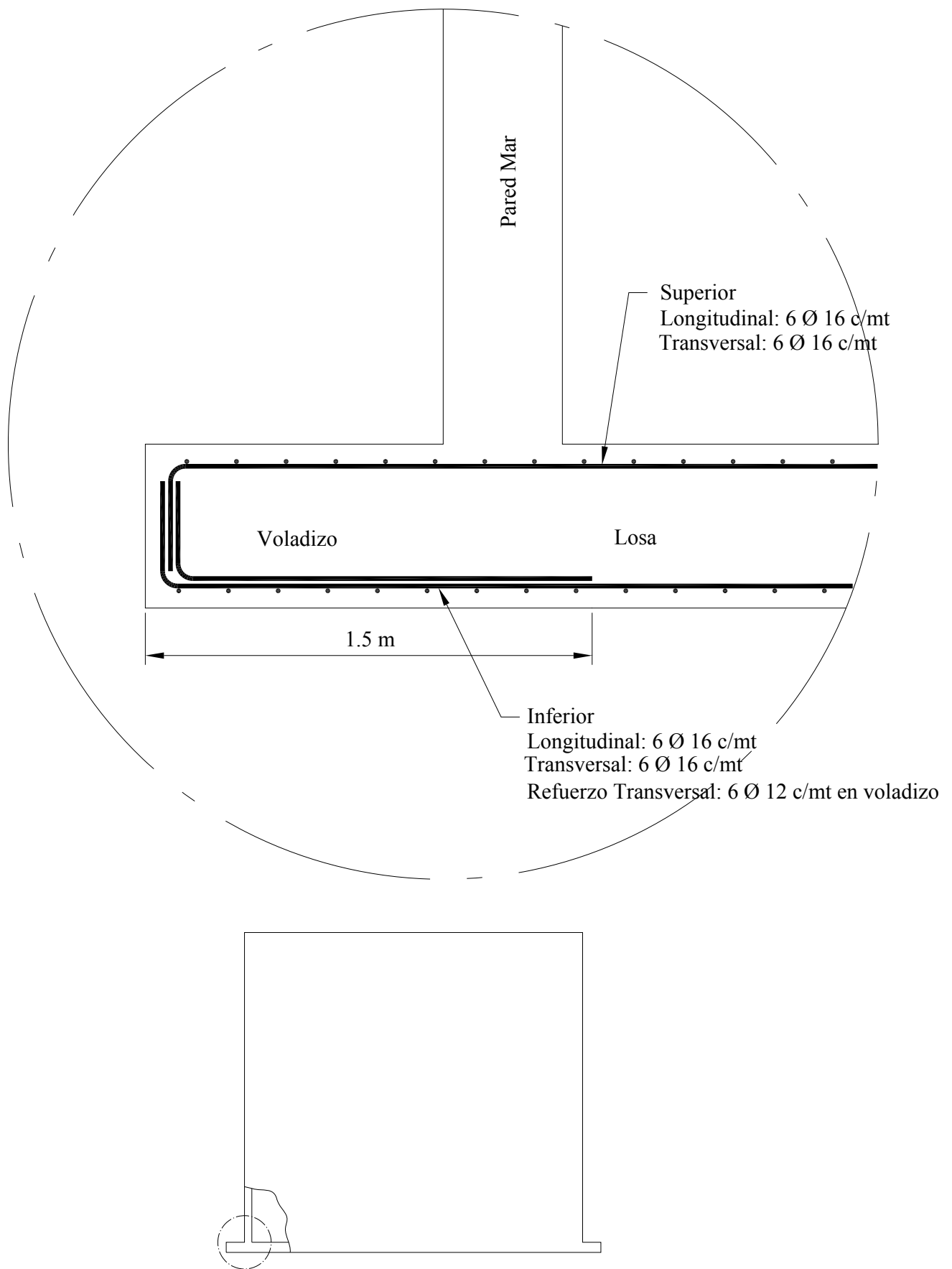


Figura III-179. Ejemplo de armado de solera y voladizos

Capítulo IV. Un Framework para el cálculo de cajones

IV.1.- Descripción del framework para el cálculo de cajones

El desarrollo de la aplicación de cálculo de cajones flotantes multicelulares para obras portuarias pretende establecer un entorno de trabajo familiar para que el usuario pueda diseñar un cajón flotante, siguiendo las diferentes etapas de cálculo a medida que se realizan.

En la aplicación se suceden las tres etapas *básicas* de los procesos de cálculo:

- **El preproceso:**
 - Definición de las variables del problema (predimensionado de la estructura, parámetros del cálculo, etc.).
 - Definición automática de las secciones de análisis según etapas constructivas y sus solicitaciones. Cálculo de las leyes de presiones correspondientes a cada etapa (lastre, relleno, etc.).
 - Definición libre de cargas: distribución de presiones definidas por el usuario.
 - Definición automática de los puntos de mayor solicitación (gargantas) para posterior análisis de los resultados.
 - Generación automática de malla. Malla de elementos hexaédricos de 20 nodos (cuadráticos). Posibilidad de refinado de malla (definición del tamaño máximo de malla).
- **El cálculo por Elementos Finitos (M.E.F.):** a partir de la malla de E.F. generada con las condiciones de contorno y cargas aplicadas.
- **El postproceso:** análisis gráfico y numérico de los resultados tensionales y de esfuerzos dando paso finalmente al armado del cajón para cada etapa constructiva.

El proceso esquematizado es el siguiente:

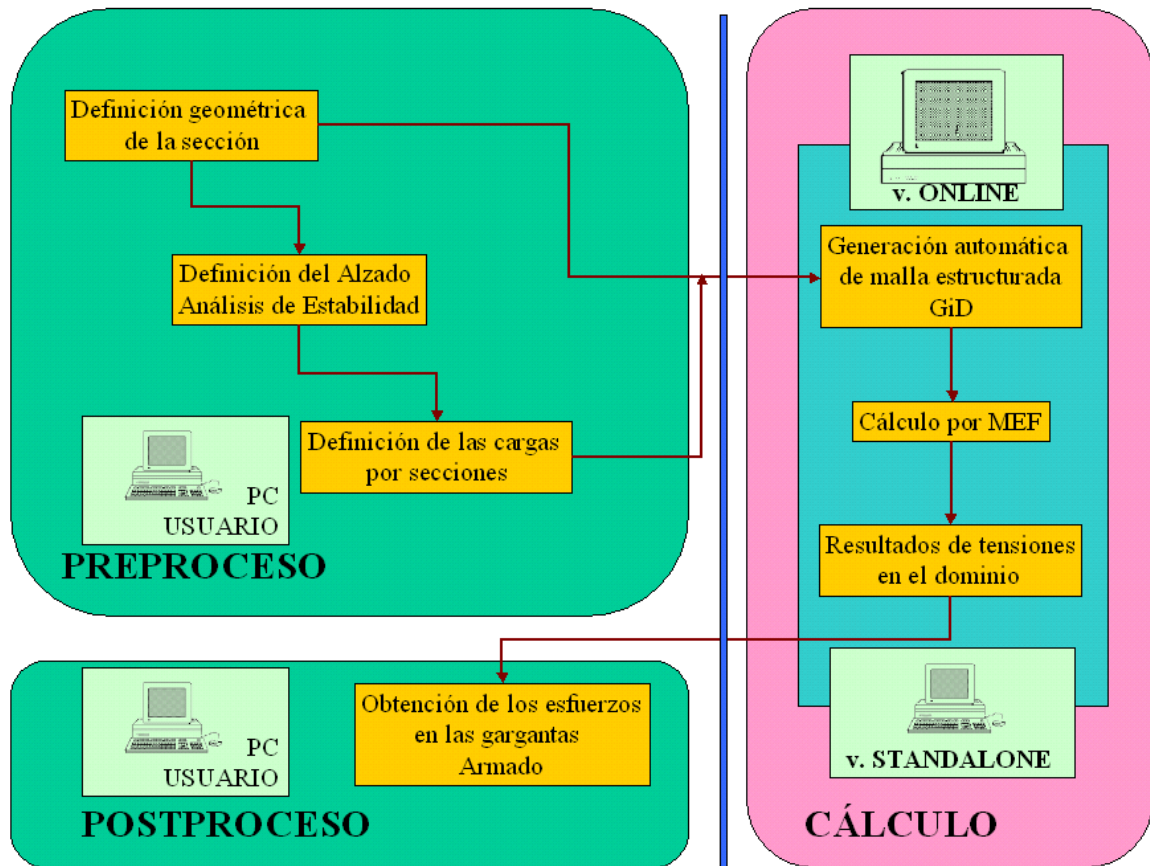


Figura IV-1. Esquema global de la estructura de la aplicación

En el preproceso, el usuario debe definir la geometría del cajón. Como ésta depende de un gran número de variables (eslora, manga, altura de fuste, número de celdas a disponer en cada dirección y espesores de paredes y gargantas, entre otras) y los valores que se les asignen pueden ser incompatibles entre sí o, si lo son, no responder a los requerimientos exigidos, la aplicación permite especificar un intervalo de variación para cada dato.

Posteriormente, presenta al proyectista todas las soluciones compatibles con esos datos, de entre las que debe elegir la que más se ajuste a las especificaciones requeridas. A continuación, se chequea la estabilidad naval del diseño elegido, tanto en la fase de construcción como en la de transporte y, de no ser satisfactoria, se remite de nuevo a la definición de geometría. En caso de resultar satisfactoria, se continúa con la determinación de las cargas básicas actuantes, tanto en construcción como en transporte y en servicio.

Una vez que el usuario ha identificado las hipótesis a resolver, las leyes de comportamiento de los materiales, las características del armado y el grado de finura de la discretización en elementos finitos que desea utilizar, se establece una conexión con el servidor a través de Internet, a quien se le envía la petición del cálculo y los datos necesarios. A partir de este momento, el usuario queda a la espera de que finalicen los cálculos para pasar a la etapa de postproceso, aunque en todo momento puede conocer el estado del proceso mediante las correspondientes peticiones al servidor.

En la fase de cálculo se realizan las tareas de generación automática de la malla de elementos finitos mediante GiD y, para cada hipótesis de carga, el cálculo de tensiones y desplazamientos utilizando CALSEF, la determinación de esfuerzos en las gargantas y la obtención de las armaduras. Si es necesario, se realiza una homogeneización de las armaduras necesarias en cada hipótesis para obtener un único esquema de armado. Tras finalizar todos los cálculos, el servidor manda un aviso (e-mail, sms, etc.) al usuario y queda a la espera de que éste le solicite los resultados obtenidos.

Para finalizar, en la fase de postproceso, el proyectista puede analizar los resultados en su ordenador cliente, que previamente habrá solicitado y recibido del servidor. Para ello, tiene la posibilidad de visualizar tensiones, desplazamientos y esquemas de armado, tanto gráficamente como mediante listados. Si los resultados son satisfactorios, puede generar automáticamente una memoria detallada del proyecto, incluyendo los planos de armado, con lo que se daría por finalizado el proyecto. En caso contrario, puede modificar cualquiera de los datos (geometría, características de los materiales, valores de las cargas, recubrimientos de las armaduras, etc.) y repetir de nuevo los cálculos.

Este capítulo pretende realizar una descripción del *servidor de cálculo*, así como de su funcionamiento, estructura, programación y de las opciones que permite. El servidor proporciona una interfaz genérica de I/O de forma que la aplicación cliente es el encargado de enviar (input) los datos correctos para el cálculo y de interpretar los resultados de la salida (output).

El servidor se ha diseñado como un sistema genérico de gestión de procesos en donde no se necesita conocer en ningún momento qué procesos se van a realizar, de qué naturaleza son ni quien es el cliente que envía y recibe los datos.

La implementación básica del servidor ofrece una serie de funcionalidades básicas a las aplicaciones clientes aunque por sí mismo no es capaz de realizar ninguna operación ya que no conoce la naturaleza de los datos ni del proceso a realizar, ni dispone de las herramientas necesarias para ello. Por tanto el servidor de cálculo se constituye como un *Framework* que se debe instanciar.

El *Framework* desarrollado cuenta, entre otras, con las siguientes funciones genéricas asociadas a cualquier proceso:

- Sistema de recepción de datos y envío de resultados (I/O)
- Sistema de registro, validación y autenticación del usuario
- Sistema de descarga de archivos
- Sistema de mensajería como aviso de fin de proceso
- Cola de espera para el inicio de proceso
- Cola de espera para el inicio de reprocesado si es necesario
- Sistema de información del estado y de los errores producidos en el proceso
- Sistema de información del estado del servidor

Al realizar una instancia del *Framework*, se define el comportamiento exacto de cada una de sus funciones. Es decir, a partir del comportamiento genérico que ofrece

cada una de las características del servidor, éstas se particularizan para un uso específico o para un proceso concreto. De esta forma, el código necesario para la instanciación es mínimo, ya que únicamente se indica qué particularidades tiene el proceso concreto frente al comportamiento genérico. En algunos casos, ni siquiera es necesario añadir código más allá del envoltorio necesario para la instanciación de la clase que ofrece la funcionalidad, debido a que el comportamiento por defecto que ofrece el *Framework* es suficiente para las necesidades del proceso.

IV.1.1.- Análisis de los requisitos

El primer paso a la hora de realizar el diseño del *Framework* consiste en evaluar las necesidades que tiene el sistema. Es decir, hay que realizar un **análisis detallado de los requisitos** que permiten identificar la funcionalidad que el sistema debe ofrecer. Esto significa que es necesario tener muy claro lo que se desea que haga el *Framework* antes de empezar a diseñarlo.

Para realizar los diagramas que permiten identificar todas las necesidades, realizar el diseño e indicar los pasos necesarios para ejecutar cualquiera de la operaciones que ejecutará el *Framework*, se recurrirá al **Lenguaje Unificado de Modelado (UML)**.

Existe un único usuario de sistema (o *actor*) que utilizará el servidor, al que se denominará *cliente del Framework*. Será el encargado de particularizar todas las operaciones que puede realizar el servidor para un uso concreto, es decir, de instanciar el *Framework*.

Una vez definido el *actor*, se deben identificar todas las operaciones que puede llevar a cabo, lo que se conoce en terminología UML como **casos prácticos**. La identificación se realiza mediante un **diagrama de caso práctico** que permite modelizar el aspecto exterior del sistema e ilustrar las interacciones con el mundo exterior.

El *actor* debe poder llevar a cabo las siguientes operaciones:

- Comprobación del estado del servidor
- Registro, validación y autenticación del usuario
- Envío de datos desde el cliente al servidor (input) y desde el servidor al cliente (output)
- Descarga de los archivos de resultados
- Envío de petición de inicio de proceso
- Envío de petición de inicio de reproceso
- Envío de petición de parada del proceso de cálculo
- Obtener información del estado y de los errores del proceso

Además existe un *actor* denominado **Administrador** que podrá llevar a cabo las siguientes operaciones:

- Registro de usuarios
- Eliminación de usuarios
- Detener cálculos

- Eliminar los datos de los cálculos
- Comprobación del estado del servidor de cálculo

Todos estos requisitos se plasman gráficamente en los siguientes diagramas de *caso de uso*:

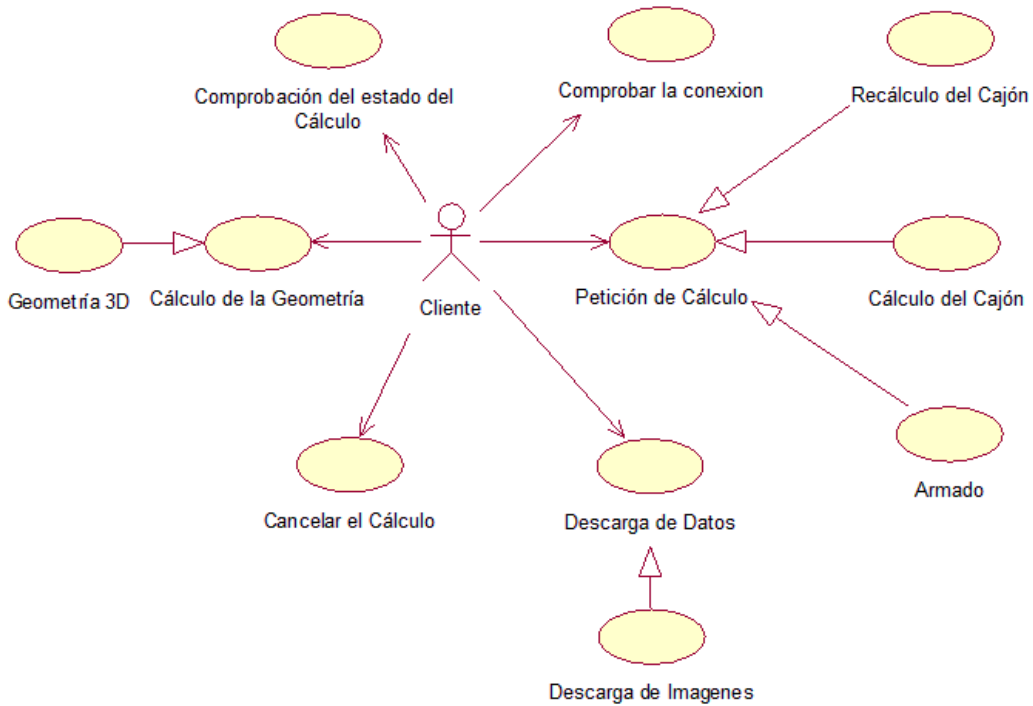


Figura IV-2. Casos de uso de la aplicación para el actor “Usuario”

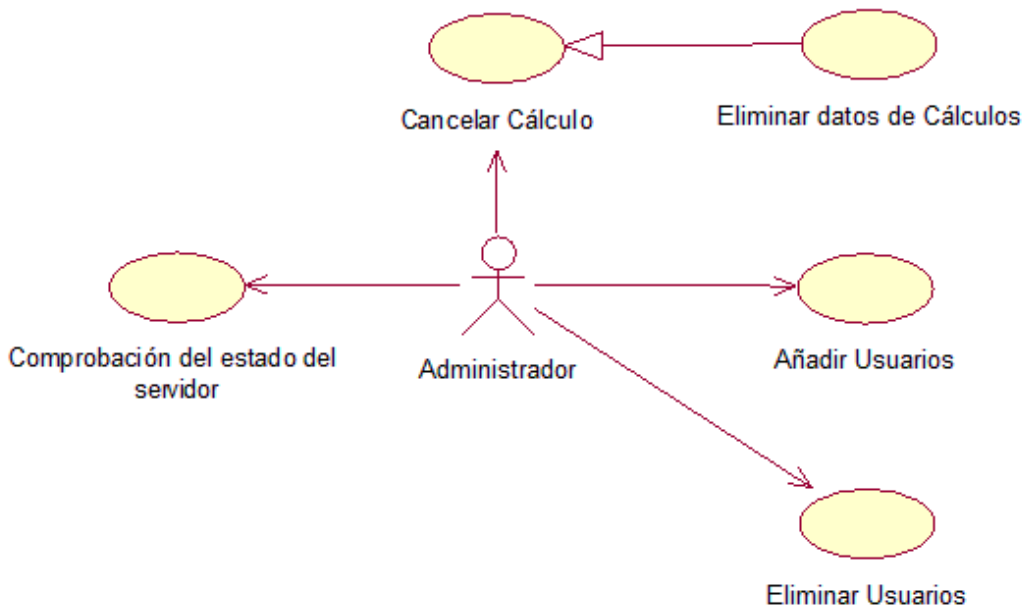


Figura IV-3. Casos de uso de la aplicación para el actor “Administrador”

Como se observa en el diagrama, cada *actor* está vinculado a una serie de formas ovaladas que representan el sistema, cada una de ellas corresponde a un caso práctico de uso que el *actor* puede ejecutar.

IV.1.2.- Tecnología empleada

Una vez consolidados los requisitos, el siguiente paso es la selección de la tecnología que se va a utilizar para desarrollar e implementar el sistema. Los factores más importantes que conviene tener en cuenta a la hora de escoger una tecnología u otra incluyen los siguientes:

- Tecnología basada en estándares
- Compatibilidad con desarrollo rápido
- Extensibilidad
- Compatibilidad con herramientas
- Implementación de bajo coste y ajena al proveedor
- División de roles durante el desarrollo y el diseño de la aplicación.

Considerando todos estos factores, se elige la tecnología **Java** para el desarrollo de todos los componentes del servidor.

Hay que definir la ubicación en la que se almacenarán los datos relativos a los procesos y los clientes que utilizarán el servidor. La opción más adecuada para almacenar entidades constantes relacionadas entre si es utilizar una base de datos relacional a la que se puede acceder desde programas en lenguaje Java con la ayuda del API JDBC.

Por último, se deben definir las herramientas necesarias para desarrollar e implementar el sistema. Se utilizará **Tomcat 5.0** como servidor web básico y contenedor jsp/servlet. Se elige **MySQL** como base de datos relacional. Ambos son productos gratuitos y de código abierto bajo licencia GNU.

IV.1.3.- Arquitectura del Framework

Una vez determinada la finalidad del sistema, se debe buscar el modelo arquitectónico adecuado que permita desarrollar el *Framework*. Conviene tener en cuenta los siguientes factores:

- **División** de la aplicación, con un diseño modular basado en componentes de forma que las partes individuales del sistema resulten fáciles de entender y no interactúen con otras partes, lo que se denomina **desdoblamiento** de los componentes de una aplicación.
- **Mantenimiento**. Cuanto más simple sea el diseño, más sencillo será su mantenimiento y más complicado resultará cometer errores.
- **Separación entre contenido y presentación**, de forma que se pueda trabajar en cada uno de estos aspectos de forma independiente.

Teniendo en cuenta estos factores, se puede utilizar el paradigma MVC/Model2 en la implementación del sistema ya que permite dividir en niveles independientes en función de la lógica y de la presentación y dividir el nivel lógico en componentes relacionados con cada uno de los casos de uso.

Es importante identificar los componentes que integrarán el modelo de la aplicación, la parte que almacena los datos y permite acceder y actualizar los componentes del sistema de forma sencilla. Los componentes se pueden clasificar en las siguientes categorías:

- **Componentes de lógica de negocios.** Son los encargados de implementar las diversas entidades del sistema, como puede ser la descarga, comprobación de estado del servidor, etc. También se pueden denominar **Objetos de negocio (Business Objects)**
- **Componentes de estado del sistema.** Ofrecen los objetos del estado del sistema almacenados en el modelo relacional base (en este caso, son **Beans** que almacenan la información que sirve de comunicación entre las diferentes capas de la aplicación). También se denominan **Entidades**
- **Componentes lógicos de acceso a datos.** Son los componentes que implementan la lógica de acceso a datos para transformar los registros de la base de datos relacional en componentes de estado del sistema y almacenar el estado de los componentes de estado del sistema de nuevo en la base de datos. Este tipo de componentes se denominan **Objetos de acceso a datos**.

Las solicitudes que provienen del cliente se pueden clasificar como solicitudes de *recuperación* de datos y *modificación* de datos. Dichas solicitudes se dirigen a las clases de acción adecuadas por medio de los *servlets*, en función de la información de asignación que se envíe por parte del cliente.

- Las solicitudes de modificación de datos se manipulan con los objetos de negocio que, a su vez, utilizan los objetos de acceso a datos para modificar los datos relacionales base.
- En las solicitudes de recuperación de datos, se utilizan directamente los objetos de acceso a datos para conseguir una entidad o grupo de entidades, que se almacenan en un atributo de solicitud (“get”)

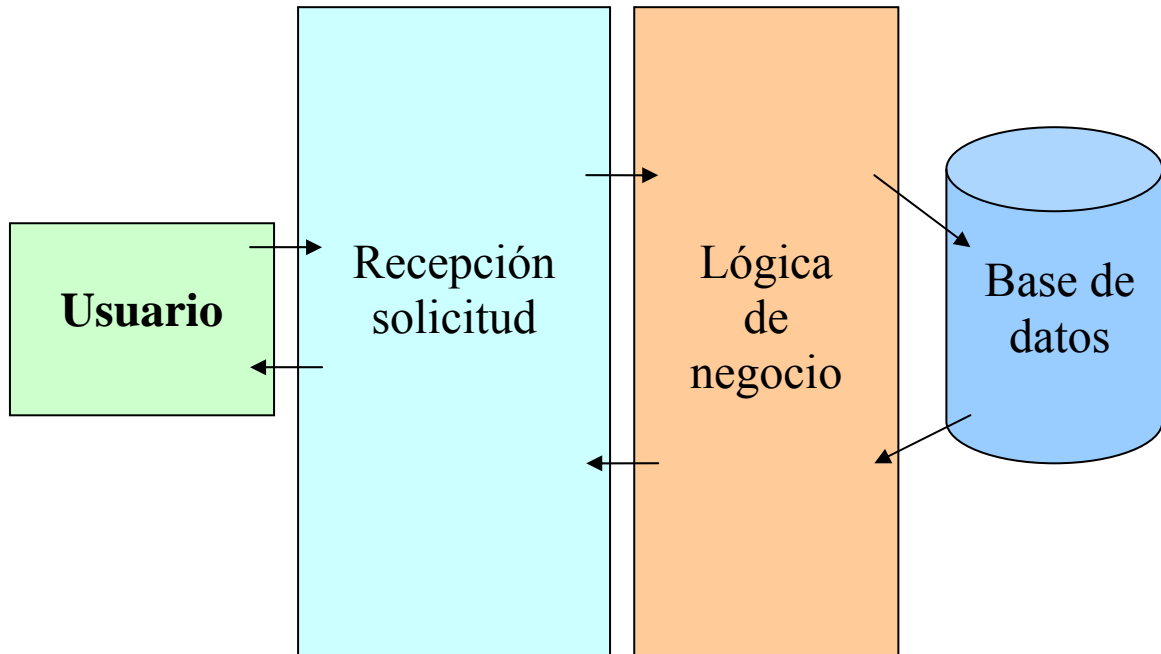


Figura IV-4. Gráfico de la estructura

De esta forma cada uno de los componentes tiene una finalidad específica: solamente ejecuta una serie de operaciones relacionadas y su papel dentro del conjunto es claro y conciso.

IV.1.4.- Seguridad

Los aspectos principales relativos a la seguridad del *Framework* que se plantea son los siguientes:

- **Autenticación.** Las diferentes partes que se comunican entre si deben probar sus credenciales (demostrar que son quienes dicen ser) por medio de la autenticación.
- **Autorización.** Una vez autenticado el cliente, es necesario determinar los recursos del *Framework* a los que tiene acceso. Se puede conseguir mediante Listas de Controles de Acceso (ACL) que muestran el “permiso” específico necesario para acceder a cada uno de los recursos.
- **Integridad de datos.** La integridad de datos asegura que los datos enviados por una parte no interfieren con terceras partes antes de llegar a su destino.
- **Privacidad de los datos.** La privacidad de los datos asegura que la información enviada por una de las partes no puede ser leída por nadie más antes de llegar a su destino.

IV.1.5.- Identificación de entidades

El siguiente paso consiste en identificar los elementos de datos (**entidades**) que determinan el estado de la aplicación y representan los datos constantes que se almacenan en la base de datos. Con ayuda del análisis de requisitos iniciales y de las

parcelas de seguridad, se pueden distinguir cuatro entidades principales que se utilizarán en el sistema.

- **CDatosRetorno**: entidad que almacena todos los datos que deben ser enviados al cliente como respuesta del fin de un proceso
- **CDatosCalculo**: entidad que almacena todos los datos necesarios para iniciar un proceso
- **CDatosStatus**: entidad utilizada en la comprobación de actividad del servidor, almacena tanto la petición de una operación asociada al proceso del servidor como los datos de validación del usuario.
- **CUserData**: entidad que representa al usuario que desea realizar el proceso, almacena todos sus datos de registro (uso legal del programa), proceso a realizar y una serie de datos estadísticos.

La entidad CDatosRetorno

Se implementa en el paquete `com.celdas.Data`. Esta clase es la encargada de almacenar los datos que se utilizan para identificar al usuario de forma única. Estos datos vienen definidos por los campos:

```
private int m_userID;  
private int m_userProc;  
private int m_statusCode;  
private int m_iLicencias;  
private Timestamp m_tsPeticion = null;  
private Timestamp m_tsInicio = null;  
private Timestamp m_tsFinal = null;
```

Figura IV-5. Campos de la entidad CDatosRetorno

Este grupo de datos se envía al cliente como respuesta a los eventos de:

- Inicio de cálculo
- Fin de cálculo
- Petición de estado del cálculo
- Petición de descarga de los resultados de un cálculo

La entidad CDatosCalculo

Se implementa en el paquete `com.celdas.Data`. Esta clase almacena todos los datos que el motor de cálculo necesita para poder iniciar los algoritmos. Este conjunto de datos se envían desde el programa cliente y se establecen en las etapas anteriores a la solicitud de cálculo. La entidad queda definida gracias a los campos:

```
private int m_userID;
private int m_userProc;
private String m_userMail;
private String m_userSerial;
private String m_szRutaProyecto;

// Datos proceso 3
private IOCalculo m_datIOCalculo = null;
private IOArmado m_datIOArmado = null;
private IOGeometria m_datIOGeometria = null;

// Datos proceso 2
private IOEstabilidad m_datIOEstabilidad = null;
private IOTransporte m_datIOTransporte = null;
private IOHundimiento m_datIOHundimiento = null;
private IORelleno m_datIORelleno = null;
private IOEmpujeMar m_datIOEmpujeMar = null;
private IOEmpujeTerreno m_datIOEmpujeTerreno = null;
private IOServicio m_datIOServicio = null;
private IORotura m_datIORotura = null;
private IOExplotacion m_datIOExplotacion = null;
private IOCombinaciones m_datIOCombinaciones = null;
```

Figura IV-6. Campos de la entidad CDatosCalculo

Este conjunto de datos se envía al servidor una única vez al inicio del proceso de cálculo. Si alguno de los campos no almacena datos correctos o es *null* el proceso de cálculo no finalizará correctamente.

La entidad CDatosStatus

Se implementa en el paquete `com.celdas.Data`. La clase `CDatosStatus` almacena todos los datos que el servidor necesita para autenticar cada una de las llamadas que el cliente realiza al servidor. Esta clase no realiza un autenticado del usuario, de eso ya se encarga otro grupo de servlets, únicamente valida la llamada actual desde el cliente. La validación se encarga de comprobar que tanto el *userID* como el *userProc* del usuario correspondan a datos reales de procesos de cálculo lanzados, además se encarga de verificar si el número de licencias de cálculo (número de ejecuciones) que restan al usuario es mayor que cero y, por último, se intenta comprobar si el software Cliente es un software legal mediante una validación del número de serie y clave.

La clase `CDatosStatus` viene definida fundamentalmente por los siguientes campos:

```
/**Indice del proceso del usuario*/
private int m_szUserID = -1;
private int m_szUserProc = -1;
/**Nombre del usuario*/
private String m_szUserName = "";
/**e-mail del usuario*/
private String m_szUserMail = "";
/**Serial del programa*/
private String m_szId_CD = "";
/**Licencias de cálculo que quedan*/
private int m_iLicencias = -1;
```

Figura IV-7. Campos de la entidad CDatosStatus

También se realizan otras tareas relacionadas con la inicialización de datos para los procesos de cálculo y la descarga de archivos, pero estas otras tareas son siempre dependientes de la funcionalidad principal de la clase que se ha descrito antes.

La entidad CUserData

Se implementa en el paquete `com.celdas.Data`. La clase `CDatosStatus` almacena todos los datos necesarios para el registro del usuario en el sistema. Estos datos garantizarán la legalidad del usuario así como la legalidad del Cliente que utiliza. Al inicio de la ejecución del Cliente se hace una comparación de estos datos frente a los almacenados en la base de datos, si ambos grupos de datos son iguales se asume que el usuario dispone de un software legal.

Los datos almacenados esta entidad son:

```

/**Indice del proceso del usuario*/
private int m_iUserID = -1;
/**Tratamiento Mr - Mrs*/
private boolean m_bMr = true;
/**Nombre del usuario*/
private String m_szUserName = "";
/**Ocupacion*/
private int m_iCmbOcupacion = 0;
/**Otra Opcuacion que no aparece en la lista*/
private String m_szOcupacion = "";
/**Organizacion*/
private int m_iCmbOrganizacion = 0;
/**Nombre de la Organizacion*/
private String m_szOrganizacion = "";
/**Departamento de la Organizacion*/
private String m_szDpto = "";
/**e-mail del usuario*/
private String m_szMail = "";

/**Telefono del usuario*/
private String m_szPhone = "";
/**Direccion del usuario*/
private String m_szDireccion = "";
/**Ciudad del usuario*/
private String m_szCiudad = "";
/**Codigo Postal del usuario*/
private String m_szCodPostal = "";
/**Provincia del usuario*/
private String m_szProvincia = "";
/**Pais del usuario*/
private String m_szPais = "";
/**ID del CD*/
private String m_szUserldCD = "";
/**Codigo de Activacion*/
private String m_szActCode = "";
/**IP del usuario*/
private String m_szUserIP = "";
/**Tipo de licencia*/
private int m_tipoLic = 0;

```

Figura IV-8. Campos de la entidad CUserData

IV.1.6.- Servicios

El servidor de cálculo necesita de la implementación de un conjunto de servicios basados en tecnología *servlet*. Los servicios permitirán recibir la información desde el cliente e iniciar su procesado.

Se pueden diferenciar los siguientes servicios públicos:

- Instalación del servidor: creación de la base de datos y asignación de los permisos necesarios. Sólo disponible para el administrador.
- Registro de usuario
- Comprobación del estado del servidor
- Inicio de un proceso de cálculo
- Comprobación del estado de un proceso determinado de cálculo
- Cancelación de un proceso de cálculo
- Descarga de los datos generados por un proceso de cálculo
- Generación del armado de una sección
- Generación de la *Vista3D* de un cajón
- Generación de la *geometría* básica de un cajón

Además se han implementado los siguientes servicios privados, es decir, servicios que únicamente están disponibles por el propio servidor:

- Inicialización de la base de datos
- Test de funcionamiento del servidor
- Ejecución del ciclo de eventos
- Ejecución del ciclo de envío de email

El funcionamiento de todos los servlets, tanto públicos como privados, sigue el mismo esquema:

1. Todos los servicios extienden la clase `javax.servlet.http.HttpServlet`
2. El procesamiento de la información se realiza a través de la función `doPost`

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)  
throws ServletException, IOException
```

Figura IV-9. Función `doPost`

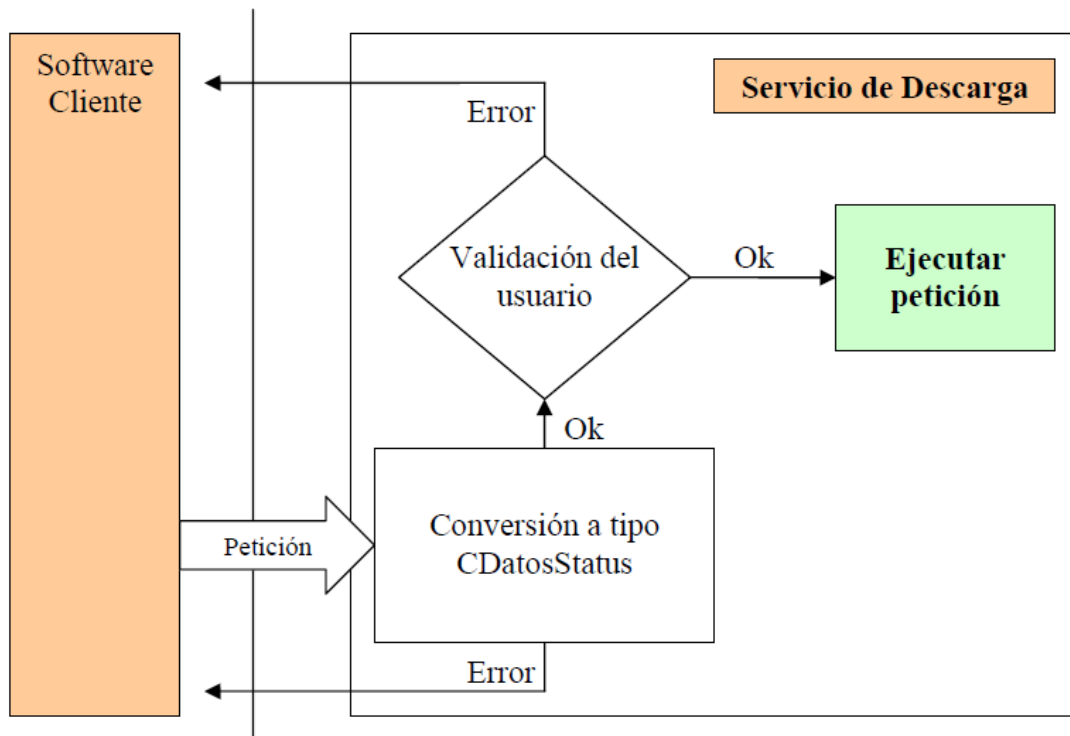


Figura IV-10. Servicio de descarga

Para explicar el funcionamiento de los servicios públicos, y a modo de ejemplo, se explicará en profundidad el funcionamiento del servicio de **descarga de los datos calculados**, el funcionamiento de este servicio se puede extender a los demás servicios públicos. Más tarde se explicará el funcionamiento de los servicios privados utilizando como ejemplo el **servicio de gestión de los ciclos de eventos**.

Servicio de descarga de datos

El **servicio de descarga de datos** está basado en la clase *DownloadServlet* que se encuentra en el paquete `com.auc`. El servicio *DownloadServlet* extiende a **HttpServlet**, de esa forma se obtiene el funcionamiento principal del sistema de descargas: *escuchar* las peticiones del cliente sin importar que tipo de cliente sea.

La estructura de la clase es la siguiente:

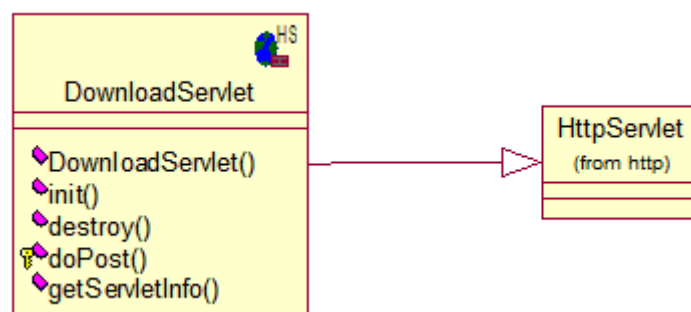


Figura IV-11. Diagrama de clases del servicio de descarga

4. Un Framework para el cálculo de cajones

El método `doPost` es un método heredado de la clase `HttpServlet` que realiza el procesado de la petición del usuario. En esta función se almacena toda la lógica de la clase. Esta función supone el punto de enlace del servicio con el cliente y permite recibir las peticiones, procesarlas y devolver una respuesta. La lógica del servicio de descarga se encuentra en esta función: permite recibir las peticiones, procesarlas y enviar una respuesta adecuada al cliente.

El diagrama de flujo muestra el funcionamiento de esta función:

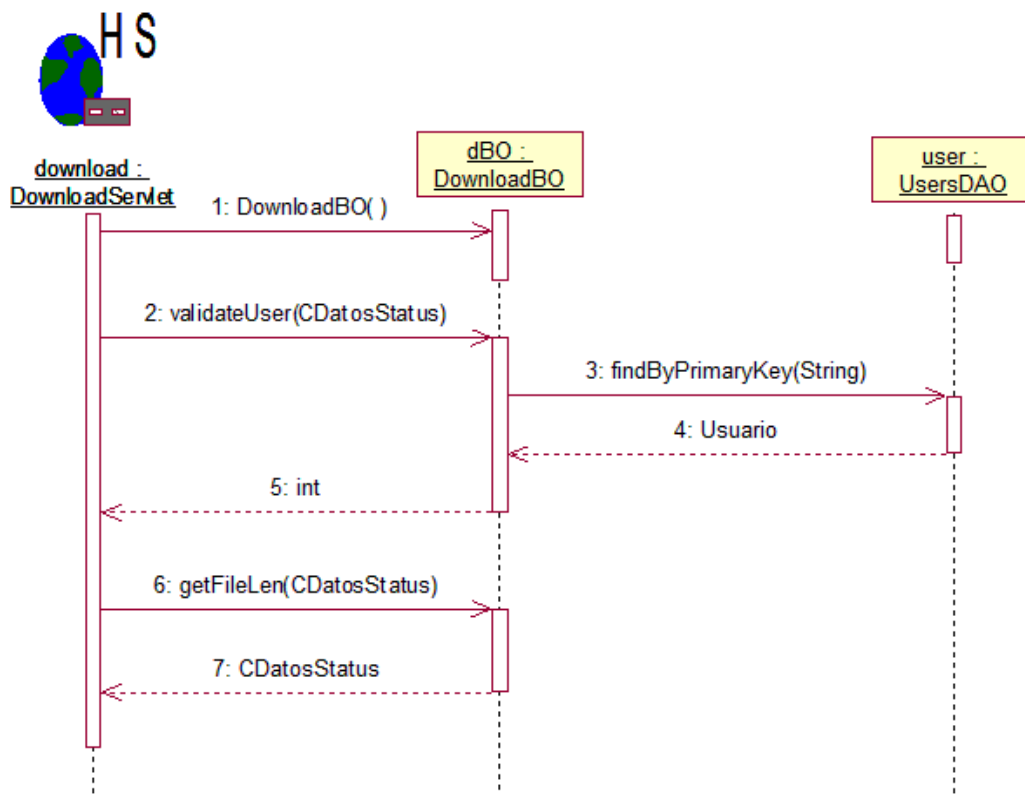


Figura IV-12. Diagrama de secuencia del servicio de descarga para la petición del tamaño del archivo

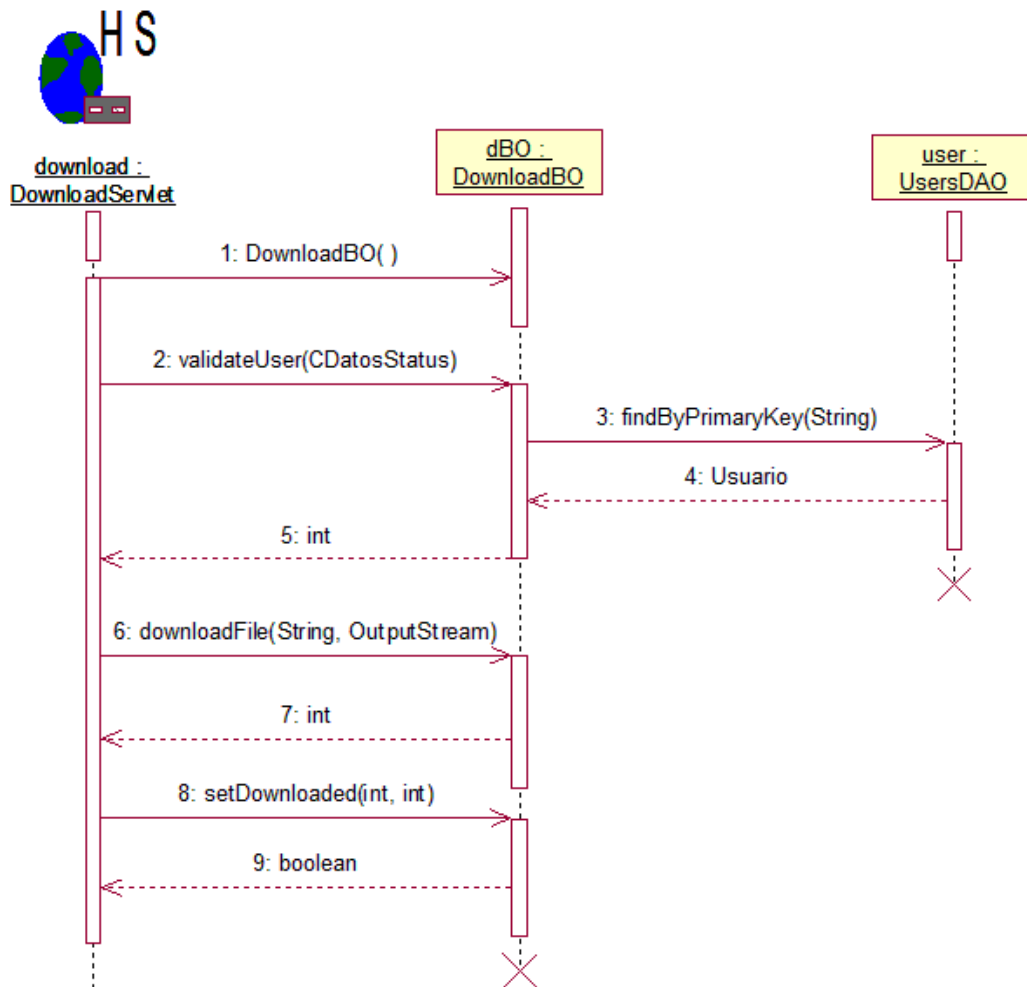


Figura IV-13. Diagrama de secuencia del servicio de descarga para la descarga real del archivo

Como se muestra en los gráficos, la función `doPost` de la clase `DownloadServlet` realiza sucesivas llamadas a la siguiente capa de la aplicación, la capa de *BusinessObjects* (BO). El servlet únicamente tiene conocimiento de los elementos de su propia capa y de los elementos públicos de los objetos instanciados de la capa BO y en ningún momento tiene permitido acceder a elementos de capas inferiores o superiores. Esta forma de estructurar el código permite crear capas lógicas de forma que un nivel no tiene nunca conocimiento del funcionamiento interno de cualquier otro nivel, únicamente conoce los puntos de entrada / salida de información.

Mediante un *Diagrama de Actividad* se observa de forma más detallada el funcionamiento del Servicio de Descarga:

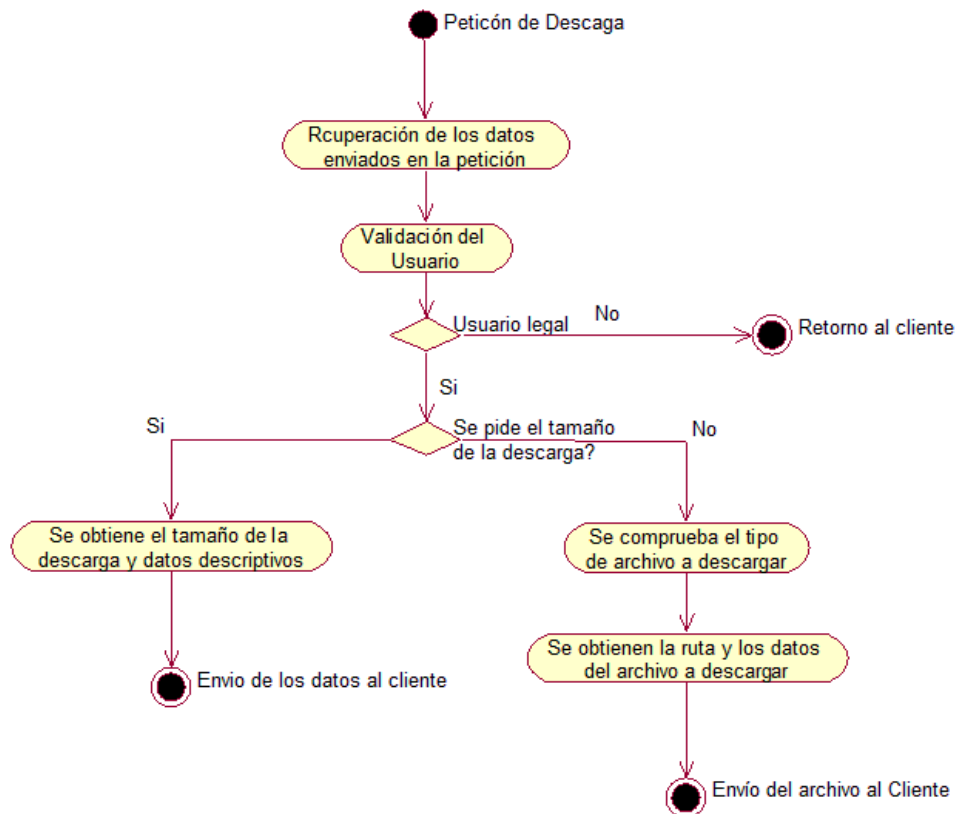


Figura IV-14. Diagrama de actividad del servicio de descarga

La primera parte del Servicio de Descarga, implementado a través de la función `doPost` del servlet, consiste en recuperar los datos enviados desde el cliente. La validación consiste en una lectura del objeto desde el buffer de entrada y una conversión (*cast*) al tipo de dato esperado. En caso de error (objeto nulo) se lanza una excepción y no se continúa con el resto del proceso, en caso de excepción durante la recepción de los datos o durante la conversión (generalmente *ClassCastException*) el servicio dispone de un sistema de gestión de excepciones que permite una salida rápida y no bloqueante para el sistema, de esa forma se devuelve el control al *Cliente* de una forma transparente y se muestra el mensaje de error adecuado.

```

Object obj = objin.readObject();
CDatosStatus dat = (CDatosStatus)obj;
if( dat == null ) {
    CLogout.getInstance().log("Error en la recepcion de los datos", CLogout.DEBUG);
    throw new ServletException( "Error en la recepción de los datos" );
}
  
```

Figura IV-15. Excepción en caso de error en la descarga

El siguiente paso consiste en la **validación del usuario**, en la que se debe autenticar el cliente que está solicitando el proceso de cálculo para asegurar que:

- a) El software que utiliza el usuario es legal
- b) El usuario está intentando descargar un proceso de cálculo que ya ha finalizado, que ha dado error o que pertenece a otro usuario

La clase `DownloadServlet` realiza llamadas a la siguiente capa de la aplicación donde estará centralizada la **lógica del negocio**, es decir, la lógica de la aplicación. Para ello se llama a la capa *BusinessObjects*, objeto `DownloadBO`, allí se encuentran las funciones necesarias para la validación del usuario y las funciones necesarias para llamar a la siguiente capa (Capa DAO).

```
CLogOut.getInstance().log("Validar usuario", CLogOut.DEBUG);
CLogOut.getInstance().log("ID = " + dat.getUserID(), CLogOut.DEBUG);
CLogOut.getInstance().log("Proc = " + dat.getUserProc(), CLogOut.DEBUG);
int iRet = dBO.validateUser( dat );
if( iRet == StatusCodes.USER_VALIDADO ) {
    ... Código en caso de que el usuario sea validado
}
```

Figura IV-16. Comprobación de usuario válido

Una vez pasado este punto, se diferencian dos tipos de peticiones:

- a) Petición del tamaño del archivo.
- b) Petición de descarga del archivo.

Para el primer tipo de petición la ejecución se basa en hacer una llamada a la función *getFileLen* de la capa BO y devolver el dato que se ha obtenido al usuario.

```
if( dat.getFileLen() ) // Se esta pidiendo el tamaño de los archivos
{
    dat = dBO.getFileLen(dat);
    // otros datos
    dat.setLicencias( dBO.getNumLicencias( dat ) );
    ObjectOutputStream out = new ObjectOutputStream( response.getOutputStream() );
    out.writeObject( dat );
    out.flush();
    out.close();
}
```

Figura IV-17. Obtención del tamaño del archivo

El segundo tipo de petición consiste en la descarga real del archivo, para ello se necesita identificar el tipo de cálculo que ha realizado el usuario (ya que según el tipo de cálculo se envía un archivo u otro) y el *mime type* del archivo que se desea descargar.

```
CLogOut.getInstance().log("se descarga un archivo", CLogOut.DEBUG);
// PETICION 2 - DESCARGA DEL ARCHIVO
String szFile = "";
int iHip = dat.getDownloadFile();
int fileType = dat.getDownloadFileType();
if( fileType == CDatosStatus.HIPOTESIS )
    szFile = Rutas.RutaZipHip( dat.getUserID(), dat.getUserProc(), iHip );
else if( fileType == CDatosStatus.ARMADO )
    szFile = Rutas.RutaZipArm( dat.getUserID(), dat.getUserProc(), iHip );
else if( fileType == CDatosStatus.IMAGENES )
    szFile = Rutas.RutaZipGif( dat.getUserID(), dat.getUserProc(), iHip );
else
    throw new ServletException("Petición de archivo no correcta");

CLogOut.getInstance().log("Ruta del archivo a descargar = " + szFile, CLogOut.DEBUG);
ServletOutputStream sos = response.getOutputStream();
// Mime Type del archivo
String contentType = getServletContext().getMimeType( szFile );
response.setContentType( contentType );
/** String szFile: nombre del archivo a descargar
 * ServletOutputStream sos: stream de salida
 */
CLogOut.getInstance().log("Inicio descarga", CLogOut.DEBUG);
dBO.downloadFile( szFile, sos );
CLogOut.getInstance().log("Fin descarga", CLogOut.DEBUG);
dBO.setDownloaded( dat.getUserID(), dat.getUserProc() );
```

Figura IV-18. Descarga real del archivo

Capa BusinessObjects: DownloadBO

El **servicio de descarga de datos** está basado en la clase `DownloadServlet` que se encuentra en el paquete `com.auc`. El servicio puede recibir dos tipos de peticiones (tamaño del archivo y descarga real del archivo) pero desconoce la lógica que hay que ejecutar tras esas peticiones. La lógica de la aplicación se encuentra en la capa **BusinessObjects** (también llamada *BO*).

En la aplicación la capa BO será la capa intermedia entre los *hotspots* (servicios que interactúan con el usuario) y la capa de almacenamiento de datos. Es decir, el nivel BO va a recibir un conjunto de datos que provienen del sistema de servicios y va *decidir* que hay que hacer con ellos según el tipo de petición, si lo necesita se puede ayudar de un sistema de persistencia de datos del que únicamente conoce su interfaz pública, que en nuestro caso es un base de datos relacional que se gestiona por la capa DAO.

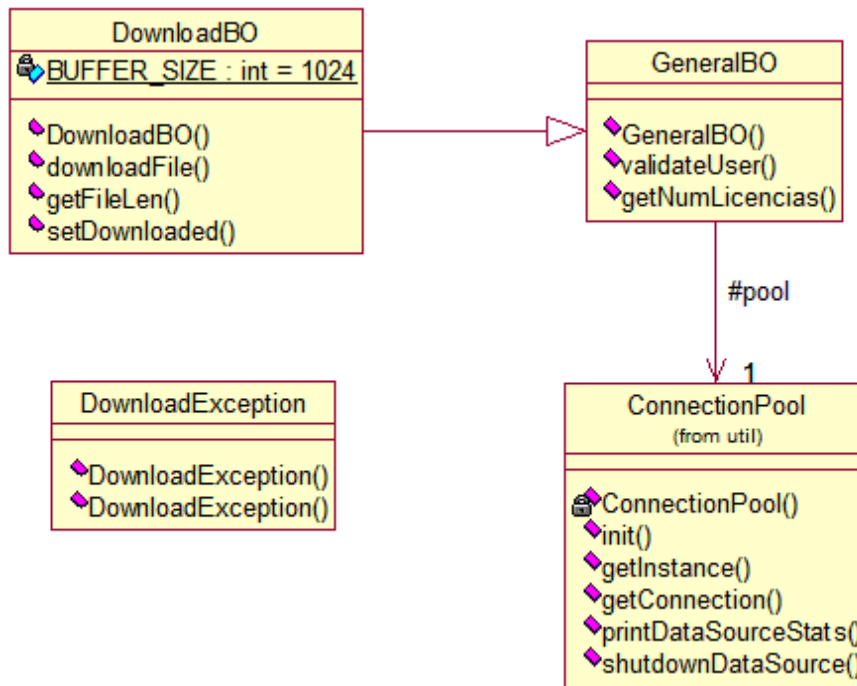


Figura IV-19. Diagrama de clases para el objeto DownloadBO

```

/**
 * Funcion para descargar un archivo
 * @param filename nombre del archivo a descargar
 * @param out stream de salida
 * @throws DownloadException Si sucede cualquier error
 */
public void downloadFile( String filename, OutputStream out )
    throws DownloadException ;

/**
 * Funcion que obtiene el tamaño de un archivo
 * @param dat Objeto de tipo CDatosStatus a completar
 * @return objeto de tipo CDatosStatus con los campos relativos al tamaño del archivo completados
 */
public CDatosStatus getFileLen(CDatosStatus dat);

/**
 * Establece como marcado el archivo de resultados del proceso de cálculo para el usuario
 * que ha hecho la petición para el id y el proc adecuado
 * @param iID Identificador del usuario que ha hecho la petición
 * @param iProc Identificador del proceso de cálculo
 * @throws DownloadException
 */
public void setDownloaded( int iID, int iProc )
    throws DownloadException ;
  
```

Figura IV-20. Cabeceras de los métodos de DownloadBO

El método `DownloadFile()` define el comportamiento que debe tener la aplicación ante una llamada al Servicio de Descarga cuando se quiere realizar una descarga real del archivo y no una petición del tamaño del archivo. Por otro lado, `getFileLen` define el comportamiento que debe tener la aplicación ante una llamada al Servicio de Descarga cuando se quiere realizar una petición del tamaño del archivo. Finalmente, `setDownloaded` establece con la marca de *Archivos descargados* en la base de datos a los archivos que se ha realizado la petición. Se necesita indicar el *ID* (identificador único del usuario) y el *iProc* (identificador del proceso de cálculo para el usuario indicado).

Una vez que se han definido los prototipos de la interfaz pública que establece el comportamiento el objeto `DownloadBO`, se analiza en profundidad cual debe ser la *lógica de negocio* que debe implementar.

En el objeto `DownloadServlet` se ha definido que el primer paso en la descarga de un archivo es realizar la petición de su tamaño y, a continuación, se realiza la descarga real del archivo. Hay que tener en cuenta que el servicio de descarga únicamente es responsable del envío del archivo como un stream de bytes, es decir, no tiene en cuenta nunca de qué tipo es el cliente que realiza la petición de descarga ni qué gestión va a realizar con el stream que recibe. Aunque en el diseño del software Cajones Multicelulares de Hormigón Armado el cliente que se encarga de recibir el stream realiza una comprobación minuciosa del tipo de archivo que se envía así como de la integridad de los datos enviados, de esa forma se puede asegurar que el archivo que se recibe está en perfectas condiciones para la siguiente etapa de cálculo.

Capa Acceso a Datos: `DownloadDAO`

La capa BO realiza llamadas a la capa de Acceso a Datos desde la que obtiene o establece todos los datos necesarios para poder operar. La capa DAO se encarga de la *persistencia* de la aplicación, es decir, conserva en un medio seguro la información necesaria para asegurar la ejecución de la aplicación.

La persistencia se asegura mediante una base de datos de tipo **MySQL**.

La capa `DownloadBO` únicamente necesita un acceso a la base de datos con el que indicar si el archivo ha sido descargado o no, para lo cual accede a la clase `DownloadDAO`.

```
UPDATE CALCULO_TERMINADO SET DESCARGADO=? WHERE USER_ID=? AND PROC=?"
```

Figura IV-21. Consulta SQL

El único método que se requiere en esta capa DAO para el proceso de descarga es `update`. Con este método se actualiza el estado de los archivos a `ARCHIVO_DESCARGADO`, que más tarde se utilizará para dejar de enviar al usuario avisos por email (sistema *notifier*) sobre el cálculo finalizado.

En el método `update` se define la consulta SQL que se ejecuta en la base de datos reemplazando en tiempo de ejecución las interrogaciones por sus valores correctos:

- **DESCARGADO**: **1** si se quiere marcar el archivo como descargado o **0** como no descargado.
- **USER_ID**: identificador del usuario en la base de datos.
- **PROC**: identificador del proceso de cálculo en la base de datos.

El par `USER_ID / PROC` identifica de forma única al proceso de cálculo que un usuario determinado ha solicitado.

Para poder ejecutar esta consulta SQL primero se debe obtener una conexión a la base de datos mediante su *pool* de conexiones. Esta conexión se obtiene en la capa BO y

se envía a DAO a través del constructor, la única labor que debe realizar la capa DAO es comprobar si la conexión está todavía en activo.

El siguiente paso es sustituir en tiempo de ejecución los parámetros dinámicos por los valores adecuados y ejecutar la consulta. Hay que señalar que debido a que se está ejecutando la aplicación web en un entorno *multi-threaded*, se debe asegurar que la consulta se ejecuta en el mismo *thread* desde la que se llama.

```
synchronized( con ) {
    if( ps.executeUpdate() != 1 )
        throw new NoSuchEntityException("error.removed.download");
}
```

Figura IV-22. Ejecución sincronizada de la consulta SQL

Gestión del ciclo de eventos

La gestión del ciclo de eventos es una de las partes más importantes de la aplicación. La gestión queda definida a partir de un servlet de inicio que se configura para ejecutarse cada 5 min. En cada ejecución se comprueba el estado del cálculo, se envían los emails necesarios y se inicia un nuevo proceso de cálculo si es necesario.

Para que se pueda definir como un servlet de inicio, se debe incluir el siguiente código en el archivo **web.xml**:

```
<servlet>
  <servlet-name>CicloEventos</servlet-name>
  <servlet-class>com.auc.CicloEventos</servlet-class>
  <load-on-startup>10</load-on-startup>
</servlet>
```

Figura IV-23. Registro de CicloEventos como servlet de inicio

Con este código se indica:

- El nombre del servlet que se está configurando.
- La clase Java que se debe ejecutar cuando se haga referencia al nombre del servlet.
- La prioridad de inicio, indica el orden de ejecución de este servlet en el momento de arranque del servidor.

Toda la lógica del ciclo de eventos queda incluida en la clase `CicloEventos`

Clase `CicloEventos`

La clase `CicloEventos` contiene toda la lógica de la gestión del sistema de eventos de la aplicación. Se entienden por eventos los siguientes casos:

- Ejecución de un cálculo
- Ejecución de un rearmado
- Obtención de las imágenes del proceso
- Inicio de un nuevo proceso de cálculo
- Envío de emails a los usuarios con el estado del cálculo.

La función principal de la clase `CicloEventos` es la función `init`, mediante la cual se indica al sistema el tiempo que debe pasar entre dos ejecuciones del ciclo de eventos (30*1000 milisegundos), es decir, 30 segundos. A continuación se realiza una llamada a la función `start()` definida en la clase padre `CicloEventosServlet`.

En la **Figura IV-24** se puede observar la dependencia (extensión) de la clase `CicloEventos` con `CicloEventosServlet`, clase que realmente almacena la lógica de todo el proceso. De forma que `CicloEventos` únicamente personaliza la función `init` según su necesidad. Este es un ejemplo de extensión del framework según la necesidad del usuario.

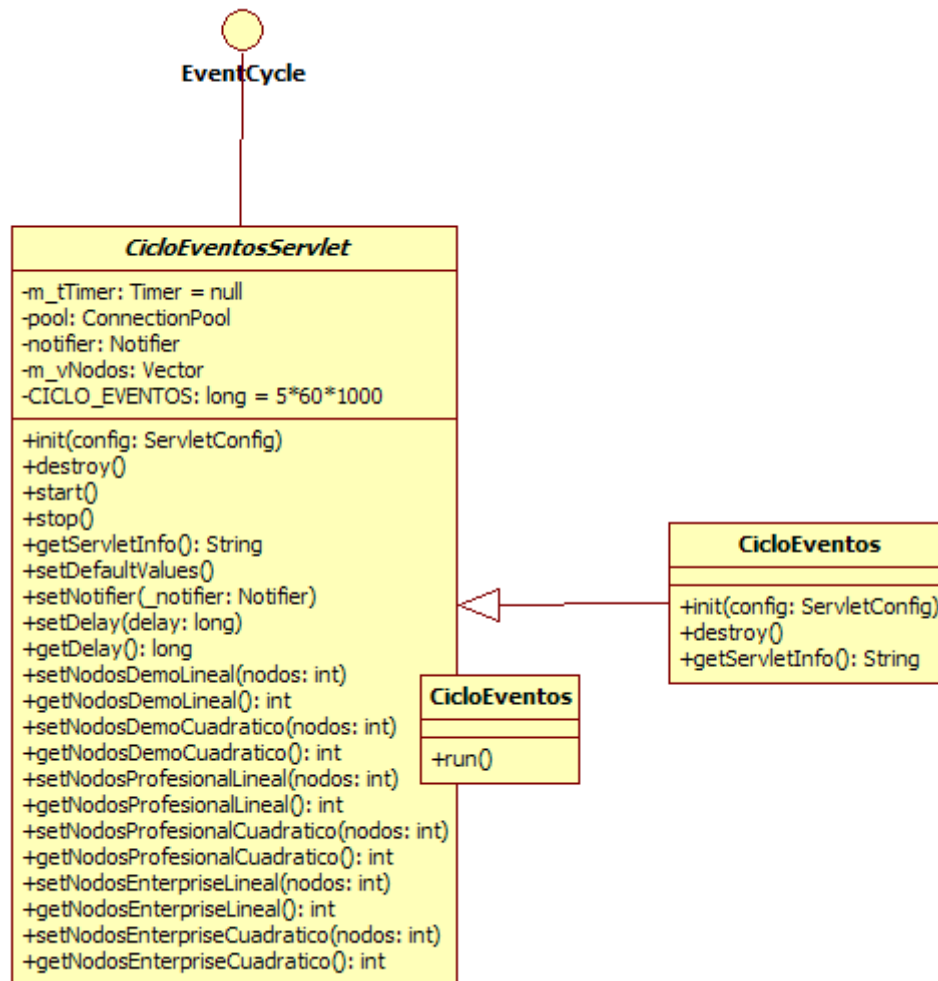


Figura IV-24. Diagrama de herencia de la clase `CicloEventos`

Clase `CicloEventosServlet`

En este servlet se almacena toda la lógica del ciclo de eventos. Además para mayor generalización, implementa una interfaz común entre todas clases relacionadas con eventos denominada `EventCycle`.

La lógica de `CicloEventosServlet` se divide en tres partes:

- Inicio del sistema
- Configuración del sistema.

- Ejecución del sistema.

La primera parte de la lógica de la clase se dedica a extender los métodos de `HttpServlet`, lo que permitirá realizar una primera configuración por defecto del sistema y establecer los parámetros necesarios para el servlet de inicio.

Para la segunda parte (configuración del sistema), se dedican gran parte de los métodos que se incluyen en la clase. Esta configuración se puede dar de dos formas diferentes: durante el proceso de inicio del servlet mediante un archivo de propiedades o mediante codificación en cualquier momento del ciclo de vida de la aplicación.

La tercera parte consiste en la ejecución real del sistema con llamadas a cada tipo de evento según el estado del sistema.

Funciones de inicio del sistema

Las funciones de inicio del sistema extienden a `HttpServlet` y permiten utilizar la clase como servlet, además se incluyen las funciones necesarias para realizar una configuración por defecto de todos los parámetros del sistema.

```
public void init(ServletConfig config) throws ServletException {
    super.init(config);

    pool = ConnectionPool.getInstance();

    this.setDefaultValues();
}

public void destroy() {
    stop();
}
```

Figura IV-25. Función `init` de inicio del sistema

Durante el proceso de inicio (función `init()`) se configuran dos valores importantes: el pool de conexiones a la base de datos y los valores por defecto de la aplicación.

```
public void start() {
    CLogOut.reset();
    CLogOut.getInstance().log("*****");
    CLogOut.getInstance().log("*** Parametros del Servidor de Calculo ***");
    CLogOut.getInstance().log("*****");
    CLogOut.getInstance().log(" ");
    CLogOut.getInstance().log("Nodos Enterprise Lineal = " +
((Integer)m_vNodos.elementAt(4)).intValue());
    CLogOut.getInstance().log("Nodos Enterprise Cuadratico = " +
((Integer)m_vNodos.elementAt(5)).intValue());
    CLogOut.getInstance().log(" ", CLogOut.DEBUG);
    CLogOut.getInstance().log("*****", CLogOut.DEBUG);
    CLogOut.getInstance().log("*** Iniciando el Servidor de Calculo ***", CLogOut.DEBUG);
    CLogOut.getInstance().log("*****", CLogOut.DEBUG);
    CLogOut.getInstance().log(" ", CLogOut.DEBUG);
    m_tTimer = new Timer();
    m_tTimer.schedule(new CicloEventos(), CICLO_EVENTOS, CICLO_EVENTOS );
}
```

Figura IV-26. Función `start` de inicio del sistema

4. Un Framework para el cálculo de cajones

La llamada a la función `start` se realiza desde la clase `CicloEventos`, clase hija de `CicloEventosServlet` y por tanto ya tendremos todos los parámetros de la aplicación configurados. Como podemos ver en la función, se hace un *log* de los valores que se han utilizado como configuración, además se inicia el *Timer* (reloj que controla el ciclo de eventos) con un valor de `CICLO_EVENTOS`, es decir, 5 min.

```
public void stop() {
    try {
        CLogOut.getInstance().log(" ", CLogOut.DEBUG);
        CLogOut.getInstance().log("*****",
CLogOut.DEBUG);
        CLogOut.getInstance().log("***   Finalizando el Servidor de Calculo   ***",
CLogOut.DEBUG);
        CLogOut.getInstance().log("*****",
CLogOut.DEBUG);
        CLogOut.getInstance().log(" ", CLogOut.DEBUG);
        m_tTimer.cancel();
        m_tTimer = null;
    } catch( Exception e ){
    }
}
```

Figura IV-27. Función `stop` de inicio del sistema

Al igual que en la función `start`, en `stop` de nuevo se hace un *log* de la acción que se realiza, el *Timer* se detiene y se anula, con lo que el sistema de Ciclo de Eventos queda detenido y eliminado.

```
public void setDefaultValues() {
    CRes mRes = new CRes("com.auc.cajsrv.business.CicloEventosServlet", "");

    try { CICLO_EVENTOS = Long.parseLong( mRes.getStr("CicloEventos") ); }
    catch( Exception nfe ) { nfe.printStackTrace(); CICLO_EVENTOS = 5*60*1000; }

    // Lectura de los nodos
    m_vNodos = new Vector();
    int iNodos;

    try { iNodos = Integer.parseInt( mRes.getStr("NodosEnterpriseLineal") ); }
    catch( Exception nfe ) { nfe.printStackTrace(); iNodos = 480000; }
    m_vNodos.add(new Integer(iNodos));

    try { iNodos = Integer.parseInt( mRes.getStr("NodosEnterpriseCuadratico") ); }
    catch( Exception nfe ) { nfe.printStackTrace(); iNodos = 240000; }
    m_vNodos.add(new Integer(iNodos));

    notifier = new SendMail();
}
```

Figura IV-28. Función `setDefaultValues` de inicio del sistema

Los valores que se indican en la función de configuración del sistema son los relacionados con el número de nodos máximo que cada usuario puede calcular según el tipo de licencia que haya comprado del programa. Además se establece un valor por defecto para el tiempo de ejecución del ciclo de eventos y para el notificador.

Funciones de configuración del sistema

La mayoría de funciones de configuración están asociadas a un *JavaBean* interno de la clase, por tanto son las típicas funciones `get/set` de recogida y establecimiento de datos para uso interno de la clase: son funciones dedicadas a configurar el máximo

número de nodos que se permite calcular según sea el tipo de licencia que el usuario ha comprado, a configurar el tiempo que debe pasar entre la ejecución de dos ciclos de eventos, y a establecer el notificador cuando suceda algún evento, por defecto ese notificador es un sistema de envío de email.

Ejecución del sistema

En la clase `CicloEventosServlet` se ha definido de forma interna la clase `CicloEventos`, encargada de almacenar la lógica de la acción que se debe realizar en cada ciclo después de comprobar el estado de la aplicación.

Cada vez que se ejecuta el `timer` se crea una instancia nueva de la clase interna `CicloEventosServlet.CicloEventos` y se ejecuta. El proceso es automático gracias a que extiende a `TimerTask`, que a su vez implementa la interfaz `Runnable`, es decir, es un sistema *multi-threaded* que se ejecuta en paralelo a la aplicación principal. Cada vez que el `timer` se activa, se lanza un nuevo *thread* que ejecuta un nuevo Ciclo de Eventos, llamando automáticamente a la función `run`.

En la lógica de esta función se diferencian dos partes:

- Obtención de los datos del estado del cálculo desde la base de datos
- Tratamiento de los datos.

Para la primera parte se necesita una consulta SQL con la que obtener el estado desde la base de datos. Se obtienen todos los registros de la tabla `ESTADO_CALCULO` de los que únicamente interesan los valores de `USER_ID`, `PROC`, `STATUS`. El orden de obtención de los datos es el natural de la tabla, es decir, el primer registro de la tabla será el primer registro que se va a obtener.

Para poder obtener los registros primero se establece una conexión con la base de datos llamando al *pool* de conexiones que se ha definido. Por defecto, el *pool* gestiona un mínimo de 10 conexiones que se pueden ampliar si se necesitan más, pero siempre tendrá un mínimo de 10 conexiones en estado activo.

```
private class CicloEventos extends TimerTask {
    public void run() {
        CLogOut.getInstance().log("Evento", CLogOut.INTERNAL);
        Connection con = null;
        PreparedStatement ps = null;
        ResultSet rs = null;
        String sql = "SELECT USER_ID, PROC, STATUS FROM ESTADO_CALCULO";
        try {
            con = pool.getConnection();
            if( con.isClosed() ) {
                throw new IllegalStateException("error.unexpected");
            }
            // Se prepara la sentencia SQL
            ps = con.prepareStatement( sql );
            // Se ejecuta la sentencia SQL
            synchronized( con ) {
                rs = ps.executeQuery();
            }
        }
    }
}
```

Figura IV-29. Ejecución sincronizada de la consulta SQL

4. Un Framework para el cálculo de cajones

Una vez que se ha ejecutado la consulta de manera sincronizada, se procede a iterar sobre los registros. Si durante la iteración sobre los registros aparece uno que cumpla con las condiciones de estado (tratamiento de datos, fase dos), la iteración se detiene y se cierra la conexión con la base de datos volviendo al estado de espera hasta el siguiente ciclo.

En el diagrama de estados siguiente se muestran los diferentes estados que puede atravesar la aplicación durante el ciclo de eventos:



Figura IV-30. Diagrama de estados durante el ciclo de eventos

- Cálculo no iniciado. Se procede a iniciar el cálculo
- Cálculo iniciado. No se realiza ningún proceso adicional, se cierra la conexión, y se espera a que finalice el cálculo.
- Cálculo de rearmado no iniciado. Se procede a iniciar el armado.
- Creación de imágenes no iniciado. Se procede a iniciar la obtención de imágenes.
- Creación de imágenes iniciado. No se realiza ninguna acción, esperando a que finalice el proceso.
- Otro estado.

```

// Bucle infinito para evitarlos codigos no conocidos
while( rs.next() ) {
    int status = rs.getInt(3);
    if( status == StatusCodes.CALCULO_NO_INICIADO ) {
        CLogOut.getInstance().log("***** INICIO DEL PROCESO DE CALCULO*");
        int id = rs.getInt(1);
        int proc = rs.getInt(2);
        SwingUtilities.invokeLater( new EvIniciarCalculo( id, proc, m_vNodos, notifier ) );
        break;
    } else if( status == StatusCodes.REARMADO_NO_INICIADO ) {
        CLogOut.getInstance().log("***** INICIO DEL PROCESO DE REARMADO*");
        int id = rs.getInt(1);
        int proc = rs.getInt(2);
        SwingUtilities.invokeLater( new EvIniciarRearmado( id, proc, notifier ) );
        break;
    } else if ( status == StatusCodes.IMAGENES_NO_INICIADO ) {
        CLogOut.getInstance().log("***** INICIO DEL PROCESO DE IMAGENES*");
        int id = rs.getInt(1);
        int proc = rs.getInt(2);
        SwingUtilities.invokeLater( new EvIniciarImagenes( id, proc, notifier ) );
        break;
    } else if ( status == StatusCodes.IMAGENES_YA_INICIADO ) {
        CLogOut.getInstance().log("Imagenes_YA iniciadas", CLogOut.INTERNAL);
        break;
    } else if ( status >= StatusCodes.CALCULO_PROCESO_GID_1 ) {
        CLogOut.getInstance().log("Calculo YA iniciado", CLogOut.INTERNAL);
        break;
    }
}
CLogOut.getInstance().log("Fin del turno actual de eventos", CLogOut.INTERNAL);

```

Figura IV-31. Secuencia de procesamiento de estados

Capítulo V. Cajones Realizados

V.1.- Introducción

A continuación se presentan los diferentes estudios realizados para proyectos de construcción o de estudio de viabilidad de cajones multicelulares para los puertos de Barcelona y Bilbao.

Estos estudios, que en la mayor parte de los casos han concluido satisfactoriamente con el cajón fabricado, fondeado y operando en servicio, han servido para validar los desarrollos teóricos que se han planteado durante la elaboración de la presente tesis doctoral, al tiempo que han obligado a una permanente y continua actualización y ampliación de los mismos, dado que en cada nuevo proyecto se ampliaba el abanico de requisitos técnicos a cubrir.

A pesar de los cambios requeridos, el marco de trabajo planteado inicialmente ha mostrado ser suficientemente flexible para permitir sin excesivo coste la incorporación de nuevas hipótesis de cálculo o de cambios en los algoritmos de generación de geometría multibloque.

V.2.- Puerto de Barcelona



Figura V-1. Vista del Puerto de Barcelona y de los diques y muelles objeto de estudio

V.2.1.- Abril 2004

Título del trabajo: ***DISEÑO DE UN CAJON MULTICELULAR TIPO PARA EL DIQUE DEL PRAT DEL PUERTO DE BARCELONA***



Figura V-2. Infografía del cajón

Datos geométricos del cajón

Eslora* _____	32.48 m
Manga _____	18.50 m
Altura Fuste _____	16.85 m
Canto losa de fondo _____	0.65 m
Altura total _____	17.50 m
Diámetro Celdas Circulares _____	3.81 m
Diámetro Celdas Ovoidales _____	1.80 m
Altura Celdas Ovoidales _____	1.27 m
Espesor gargantas interiores _____	20 cm
Espesor paredes longitudinales _____	40 cm
Espesor paredes transversales _____	30 cm
Vuelo transversal losa de fondo _____	1 m

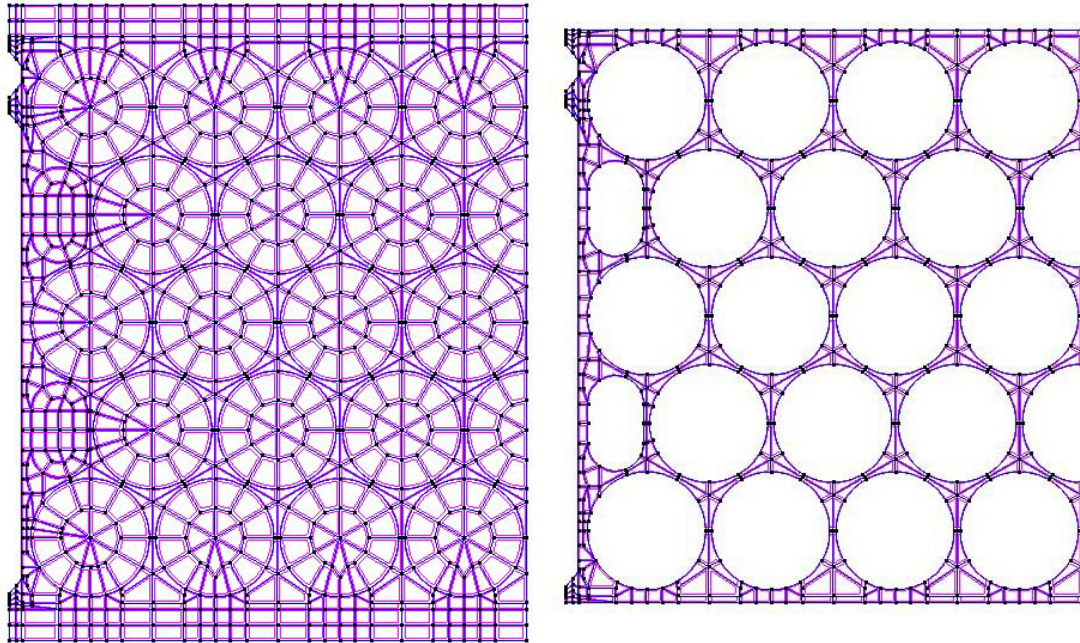


Figura V-3. Geometría modelada de media sección. (a) Losa, (b) Fuste.

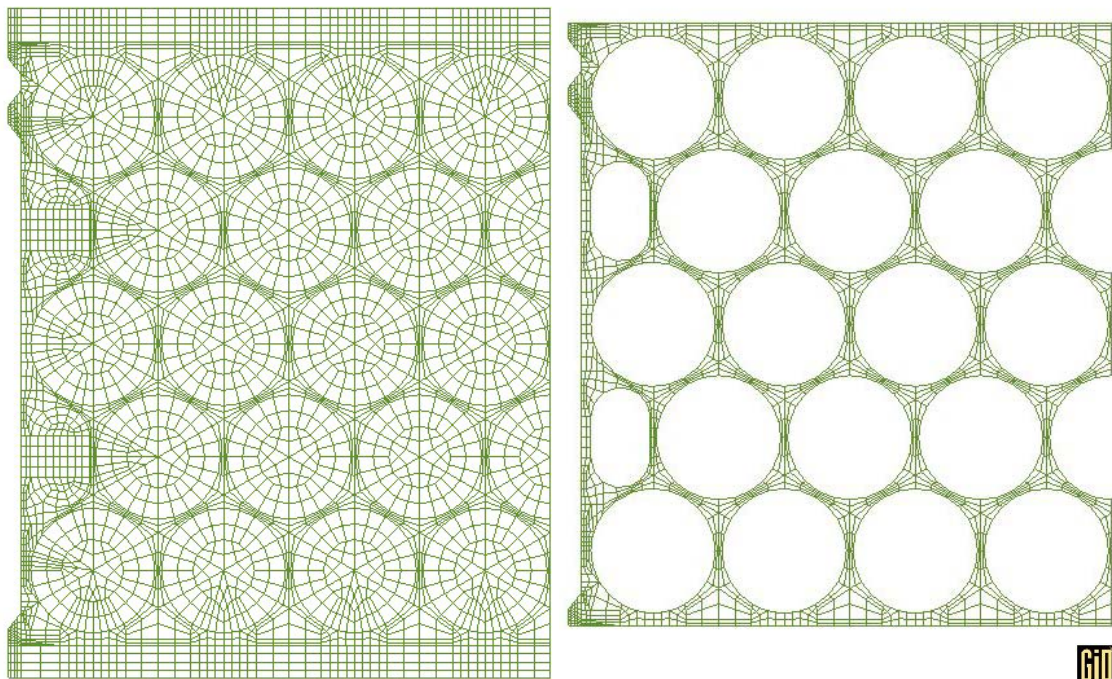


Figura V-4. Malla de media sección. (a) Losa, (b) Fuste.

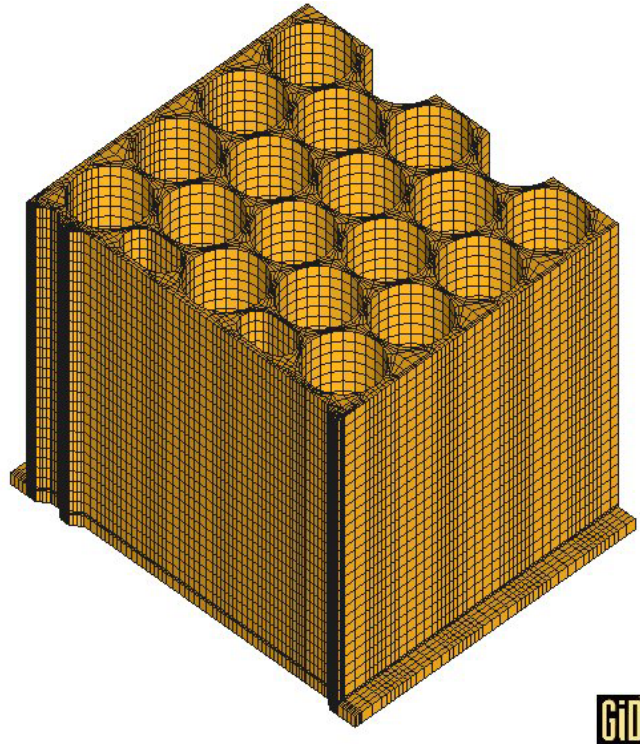


Figura V-5. Vista en 3D de la mitad simétrica del cajón.

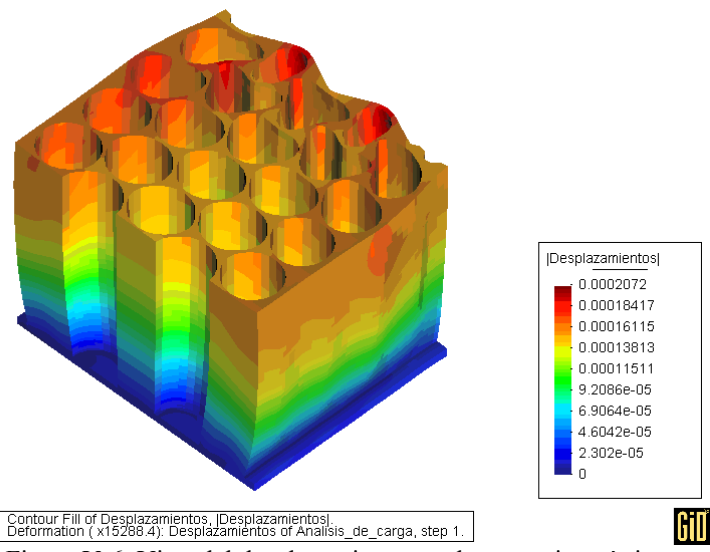


Figura V-6. Vista del desplazamiento resultante en isométrico

V.2.2.- Marzo 2005

Título del trabajo: *DISEÑO DE UN CAJON MULTICELULAR TIPO PARA EL MUELLE DEL PRAT (FASE II) DEL PUERTO DE BARCELONA*

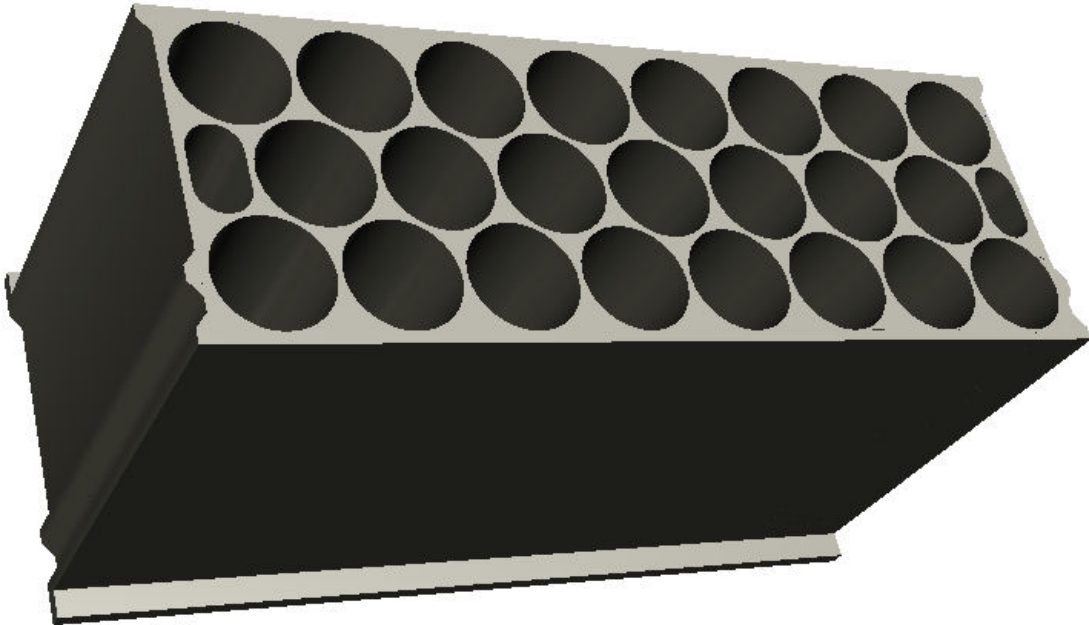


Figura V-7. Infografía del cajón

Datos geométricos del cajón

Eslora*	_____	32.48 m
Manga	_____	11.56 m
Altura Fuste	_____	16.85 m
Canto losa de fondo	_____	0.65 m
Altura total	_____	17.50 m
Diámetro Celdas Circulares	_____	3.81 m
Diámetro Celdas Ovoidales	_____	1.80 m
Altura Celdas Ovoidales	_____	1.27 m
Espesor gargantas interiores	_____	20 cm
Espesor paredes longitudinales	_____	40 cm
Espesor paredes transversales	_____	30 cm
Vuelo transversal losa de fondo	_____	1 m

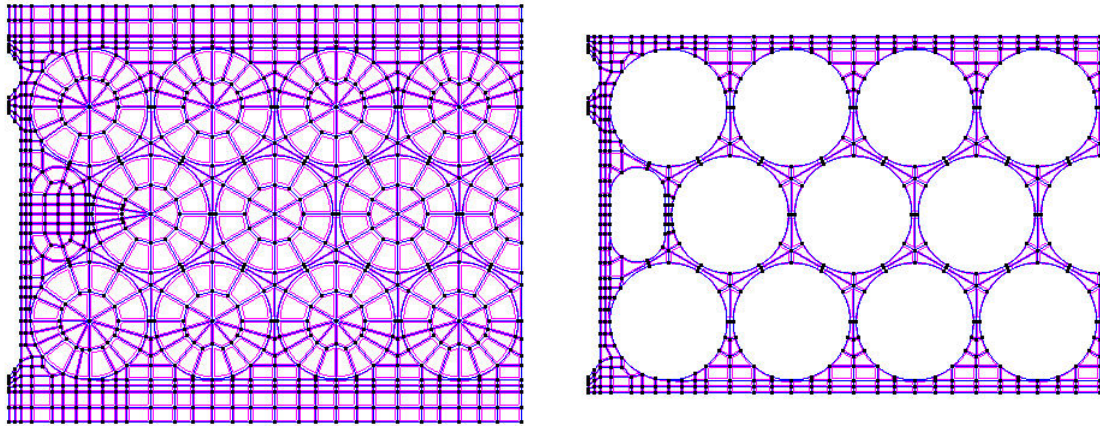


Figura V-8. Geometría modelada de media sección. (a) Losa, (b) Fuste.

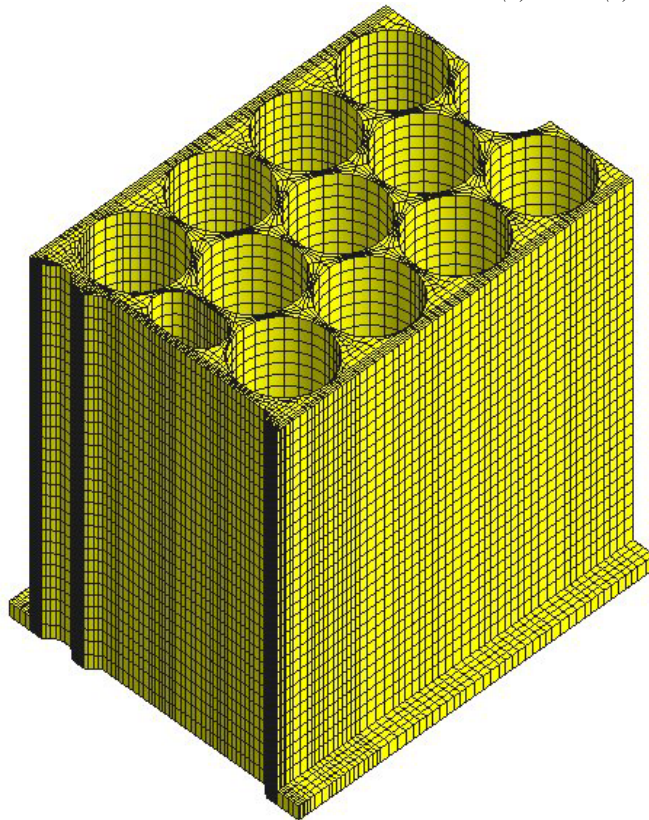


Figura V-9. Vista en 3D de la mitad simétrica del cajón

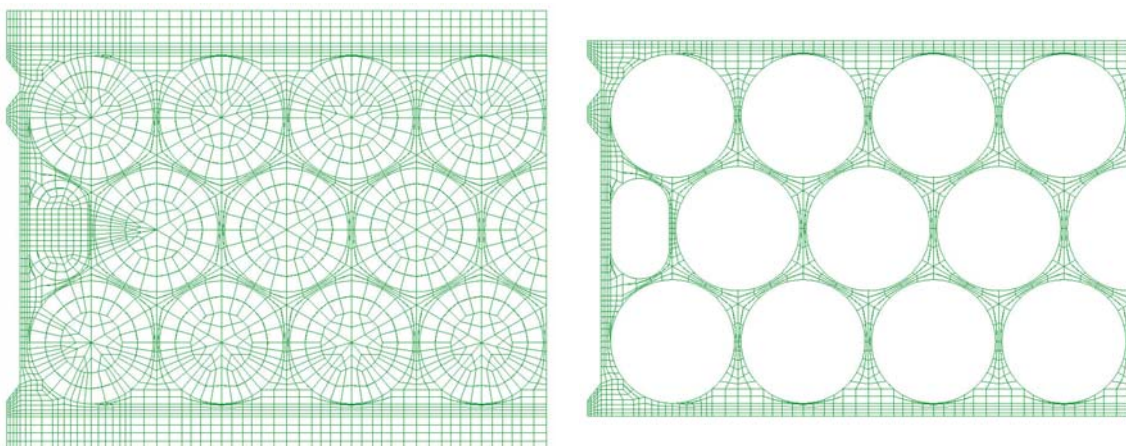


Figura V-10. Malla de media sección. (a) Losa, (b) Fuste

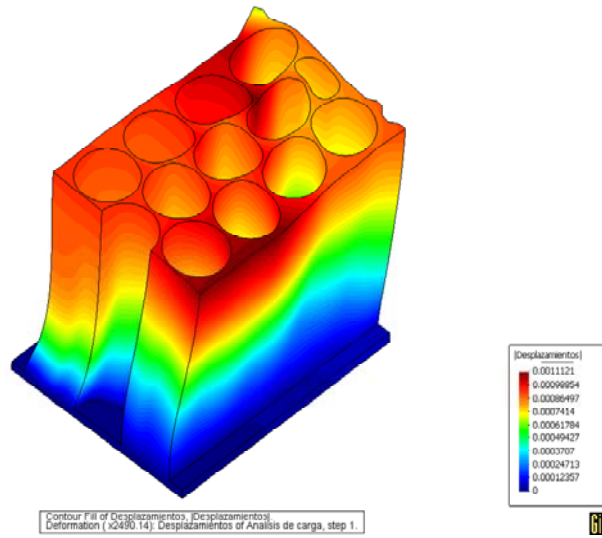


Figura V-11. Vista del desplazamiento resultante en isométrico

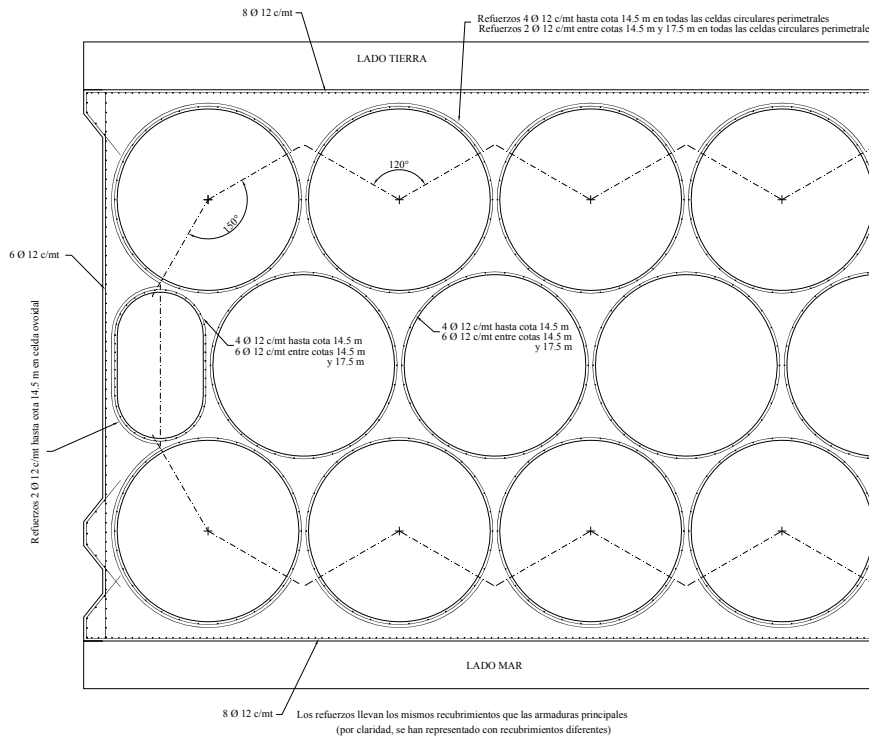


Figura V-12. Esquema de armado horizontal del fuste

V.2.3.- Marzo 2005

Título del trabajo: ***DISEÑO DE UN CAJON MULTICELULAR TIPO PARA EL MUELLE DEL PRAT (FASE I-A) DEL PUERTO DE BARCELONA***



Figura V-13. Infografía del cajón

Datos geométricos del cajón

Eslora* _____	39.254 m
Manga _____	18.5 m
Altura Fuste _____	16.65 m
Canto losa de fondo _____	0.55 m
Altura total _____	17.20 m
Diámetro Celdas Circulares _____	3.51 m
Diámetro Celdas Ovoidales _____	1.655 m
Altura Celdas Ovoidales _____	1.179 m
Altura Celdas Ovoidales Esquina _____	0.580 m
Achaflanamiento frontal _____	0.937 m
Espesor gargantas interiores _____	20 cm
Espesor paredes longitudinales _____	40 cm
Espesor paredes transversales _____	25 cm
Vuelo transversal losa de fondo _____	1 m

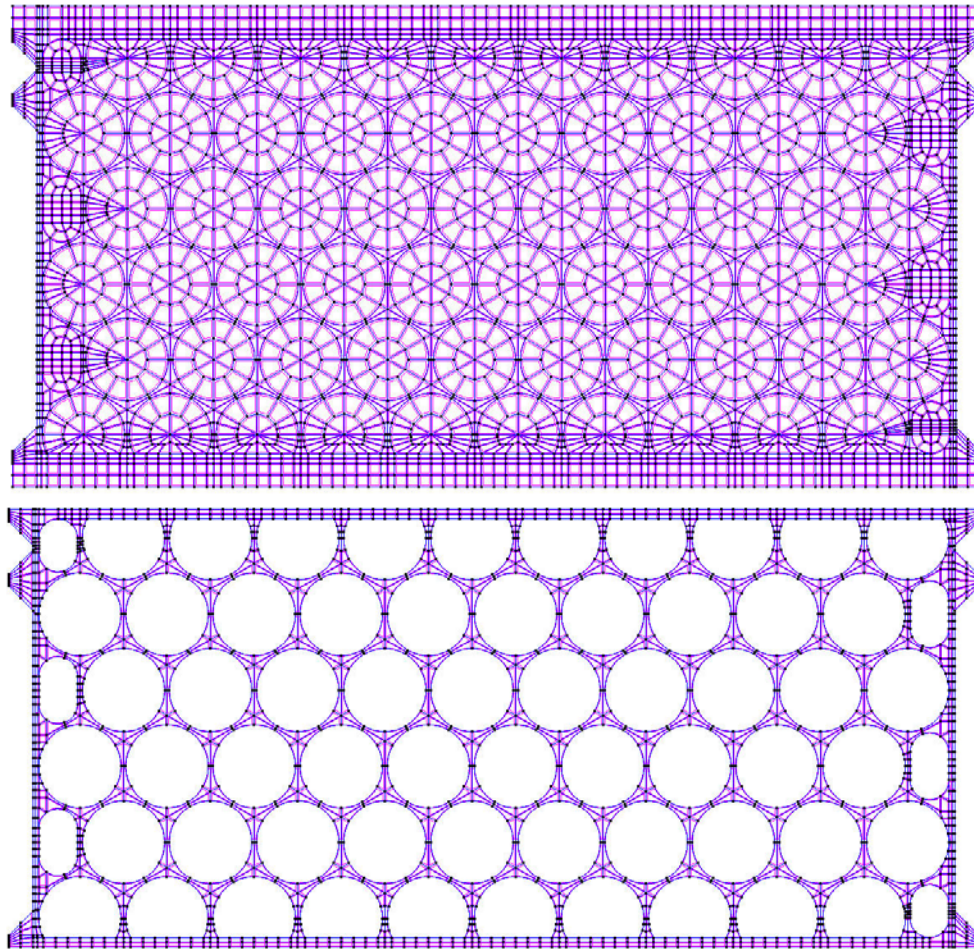


Figura V-14. Geometría modelada. (a) Losa, (b) Fuste.

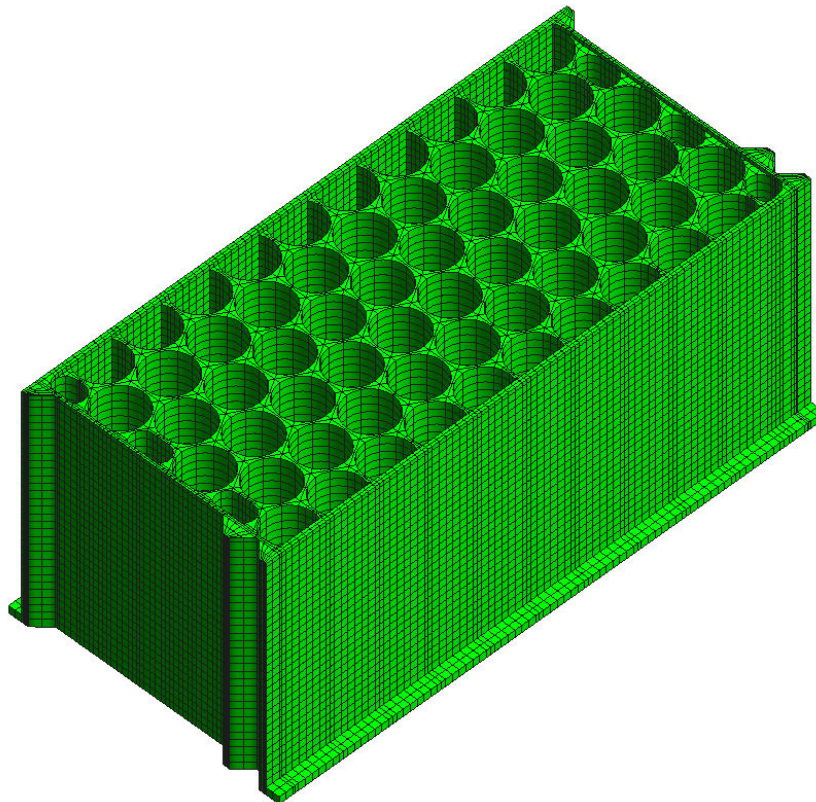


Figura V-15. Vista en 3D del cajón.

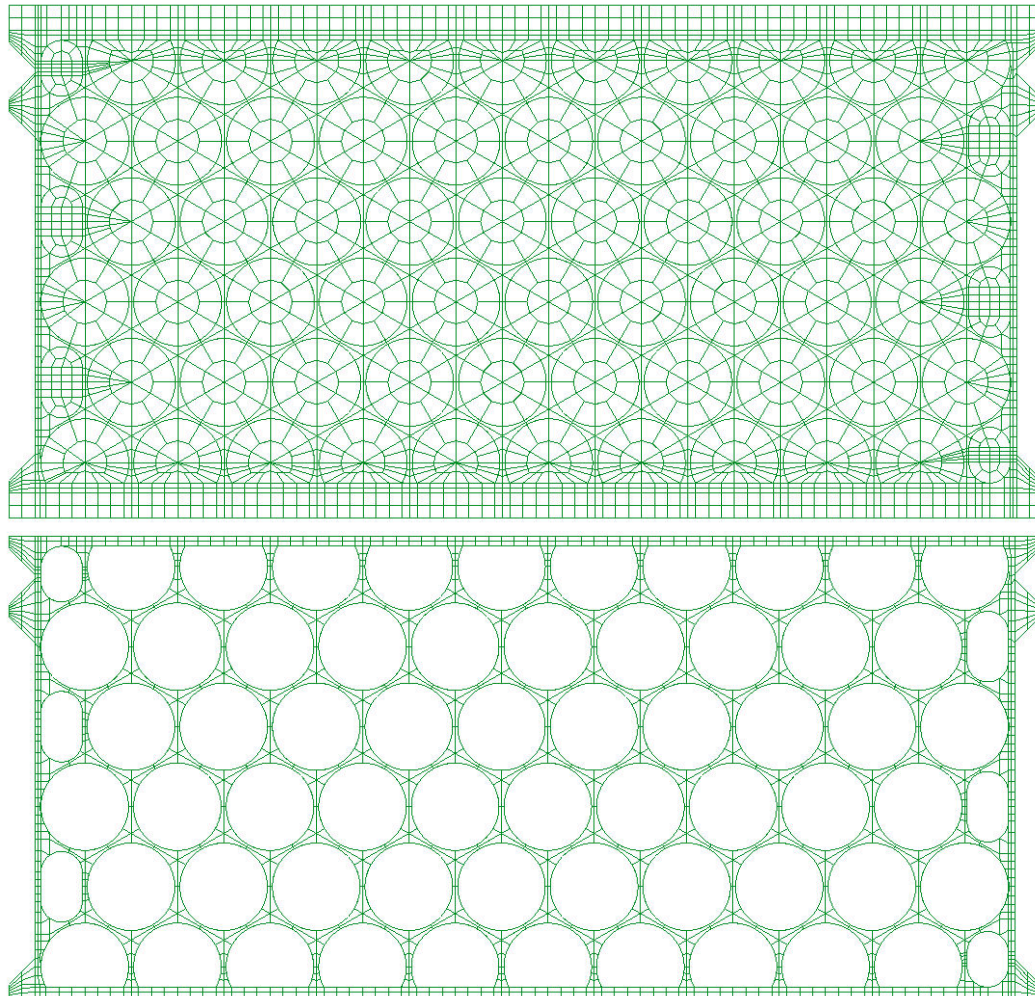


Figura V-16. Malla utilizada. (a) Losa, (b) Fuste.

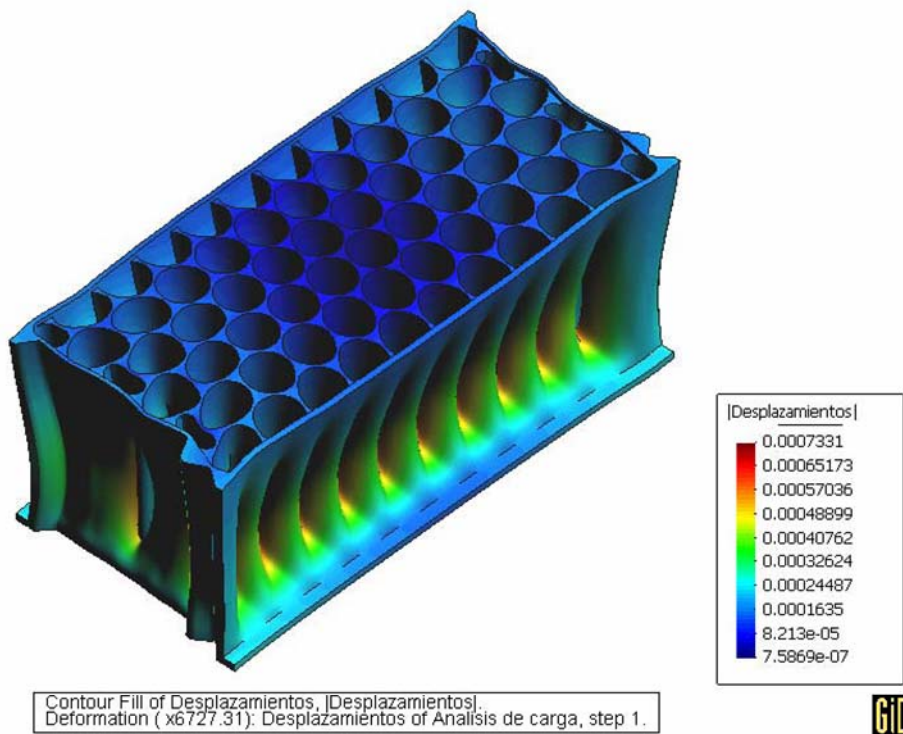


Figura 5.2.1.- Vista del desplazamiento resultante en isométrico

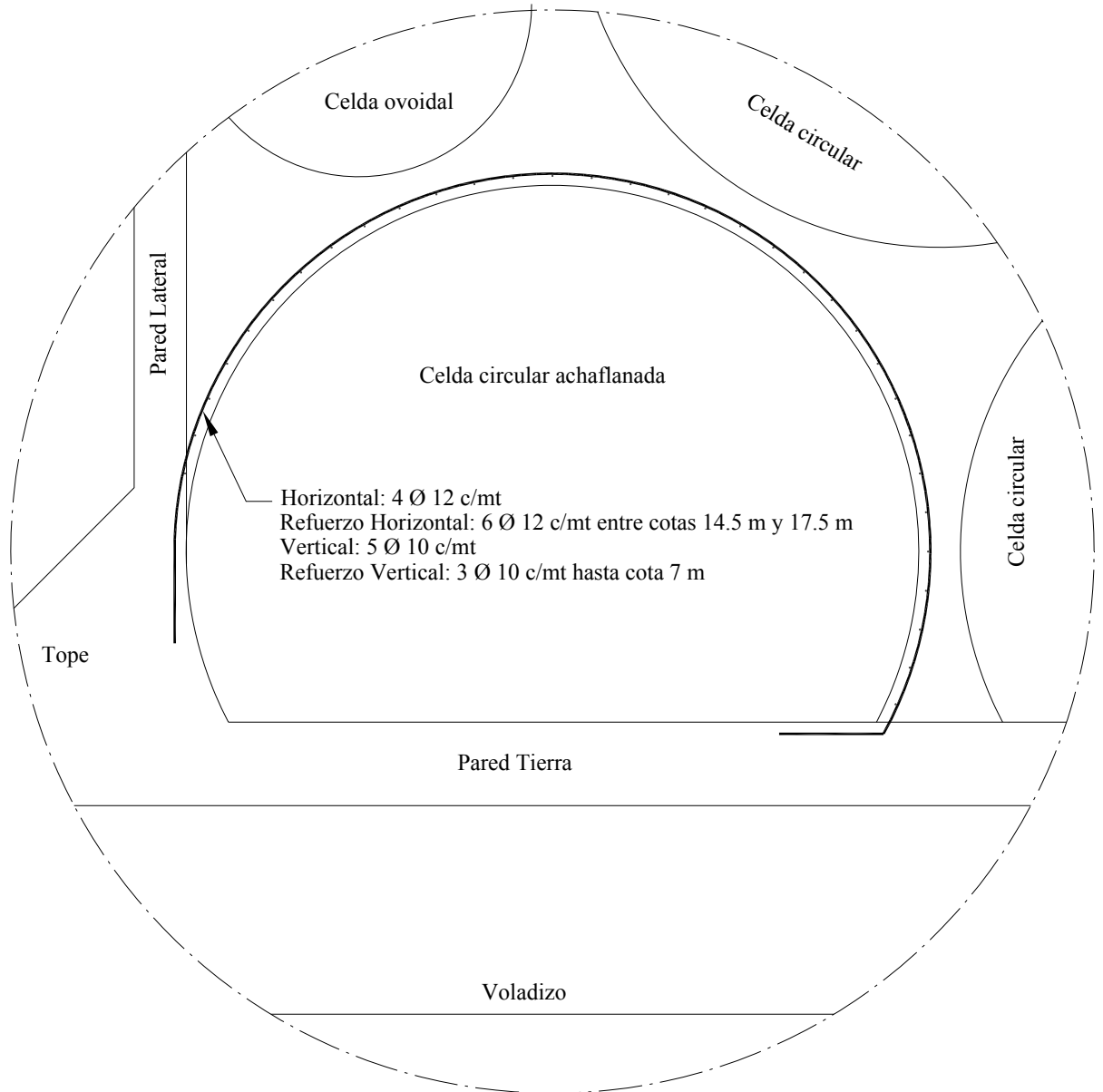


Figura V-17. Esquema de armado de las Celdas Circulares Achaflanadas de Esquina en Pared Tierra

V.2.4.- Junio 2008

Título del trabajo: ***DISEÑO DE UN CAJON MULTICELULAR TIPO PARA LA AMPLIACIÓN DEL MUELLE ADOSADO (2ª FASE B) DEL PUERTO DE BARCELONA***



Figura V-18. Infografía del cajón

Datos geométricos del cajón

Eslora* _____	39.254 m
Manga _____	18.5 m
Altura Fuste _____	16.65 m
Canto losa de fondo _____	0.55 m
Altura total _____	17.20 m
Diámetro Celdas Circulares _____	3.51 m
Diámetro Celdas Ovoidales _____	1.655 m
Altura Celdas Ovoidales _____	1.179 m
Altura Celdas Ovoidales Esquina _____	0.580 m
Achaflanamiento frontal _____	0.937 m
Espesor gargantas interiores _____	20 cm
Espesor paredes longitudinales _____	40 cm
Espesor paredes transversales _____	25 cm
Vuelo transversal losa de fondo _____	1 m

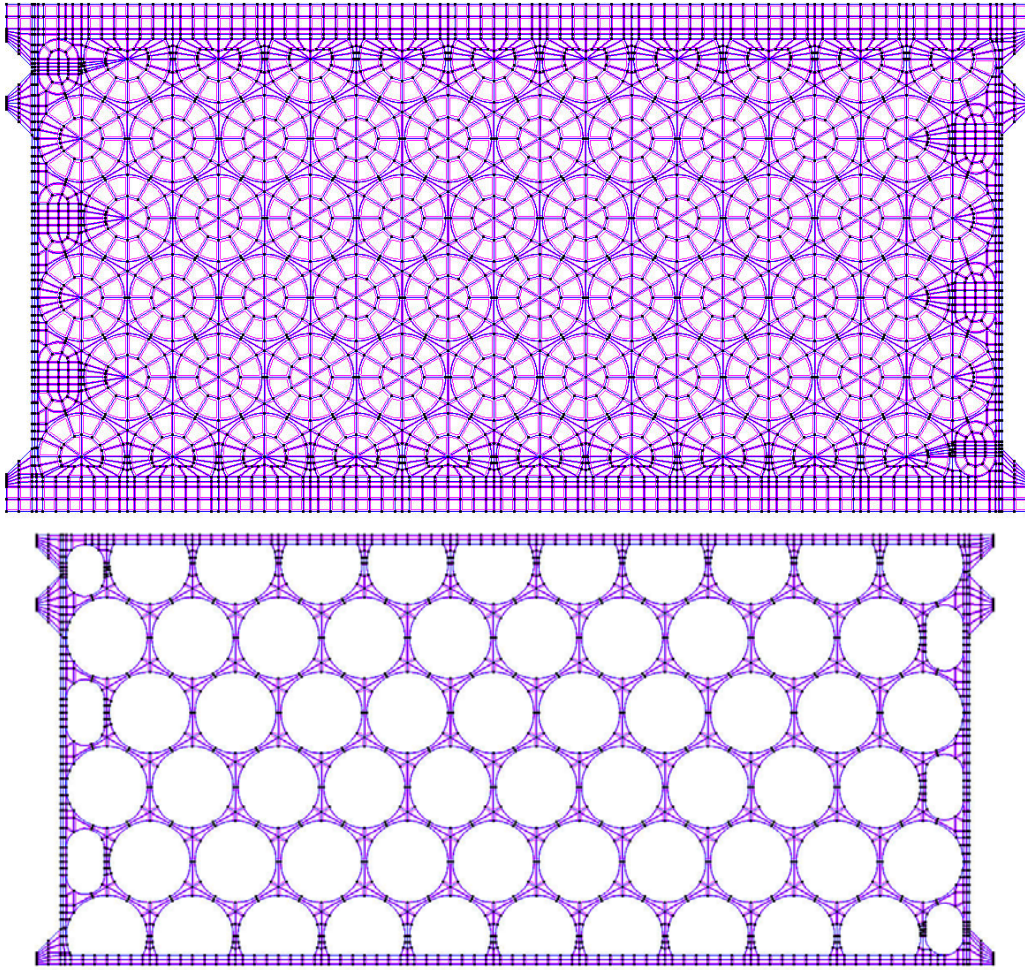


Figura V-19. Geometría modelada. (a) Losa, (b) Fuste.

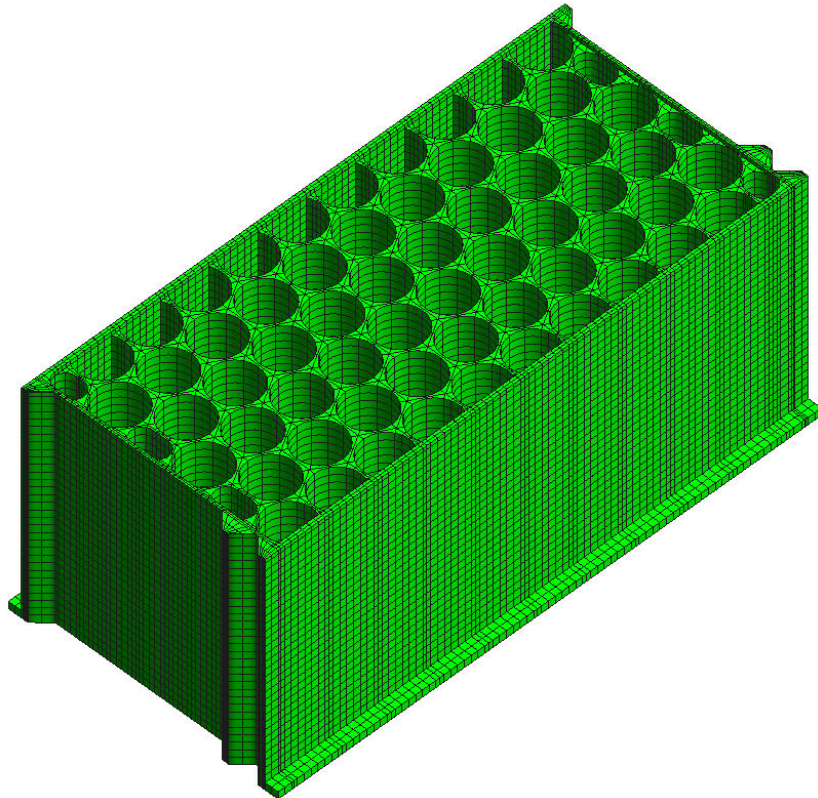


Figura V-20. Vista en 3D del cajón.

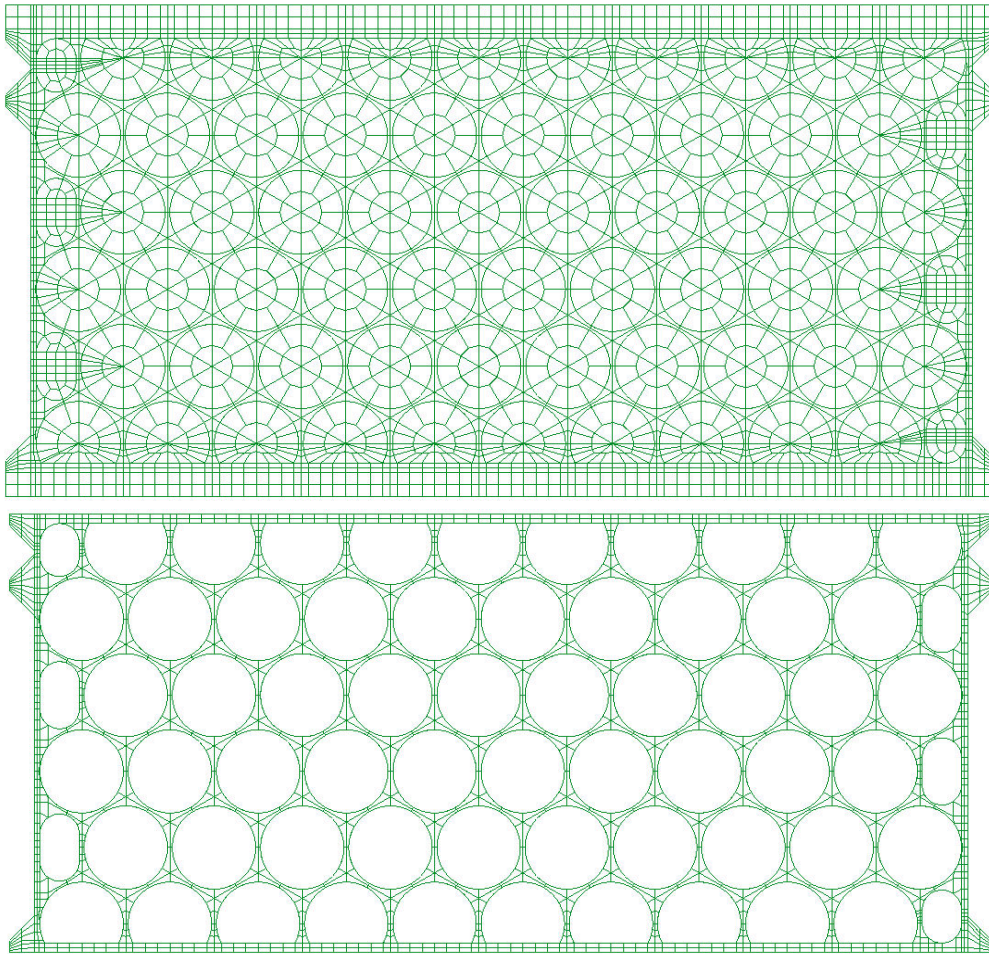


Figura V-21. Malla utilizada. (a) Losa, (b) Fuste.

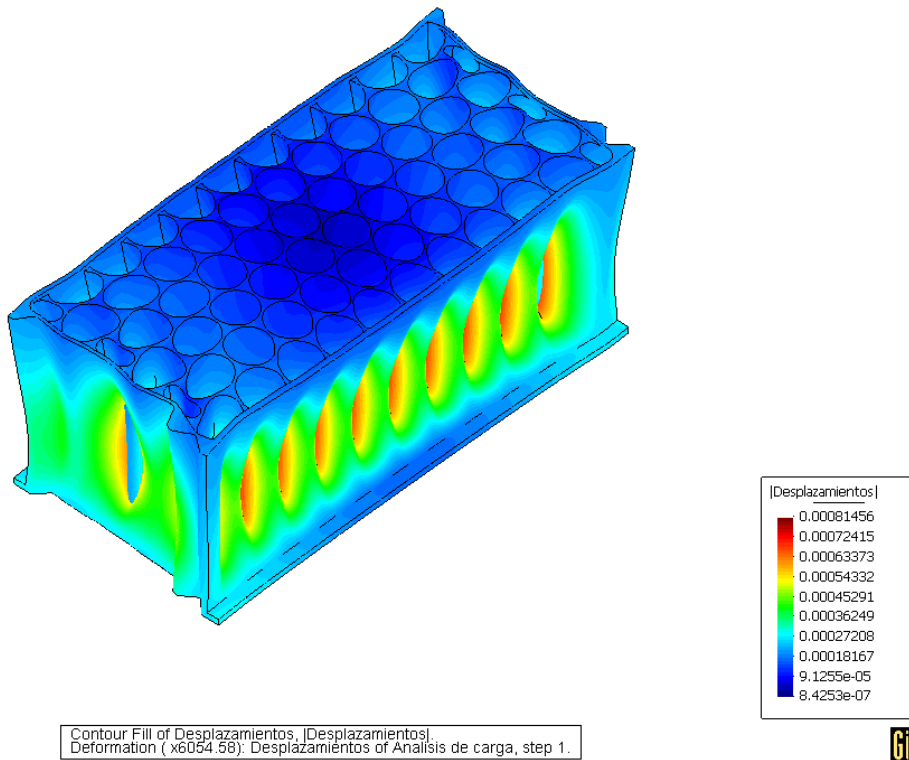


Figura V-22. Vista del desplazamiento resultante en isométrico

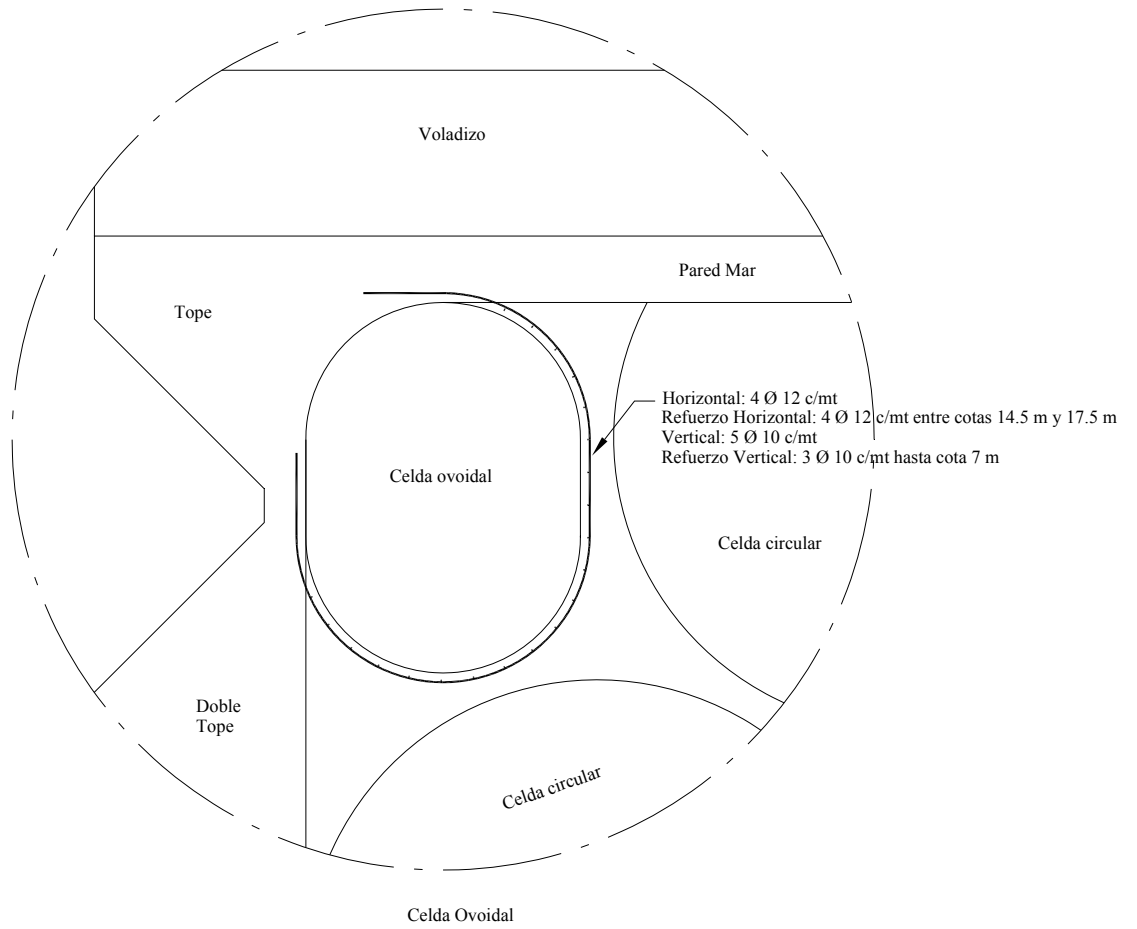


Figura V-23. Esquema de armado de las Celdas Ovoidales en esquina Lado Mar

V.2.5.- Octubre 2008

Título del trabajo: ***DISEÑO DE UN CAJON MULTICELULAR TIPO PARA LA AMPLIACIÓN DEL MUELLE ADOSADO (FASE II-C) TRAMO EXENTO DEL PUERTO DE BARCELONA***

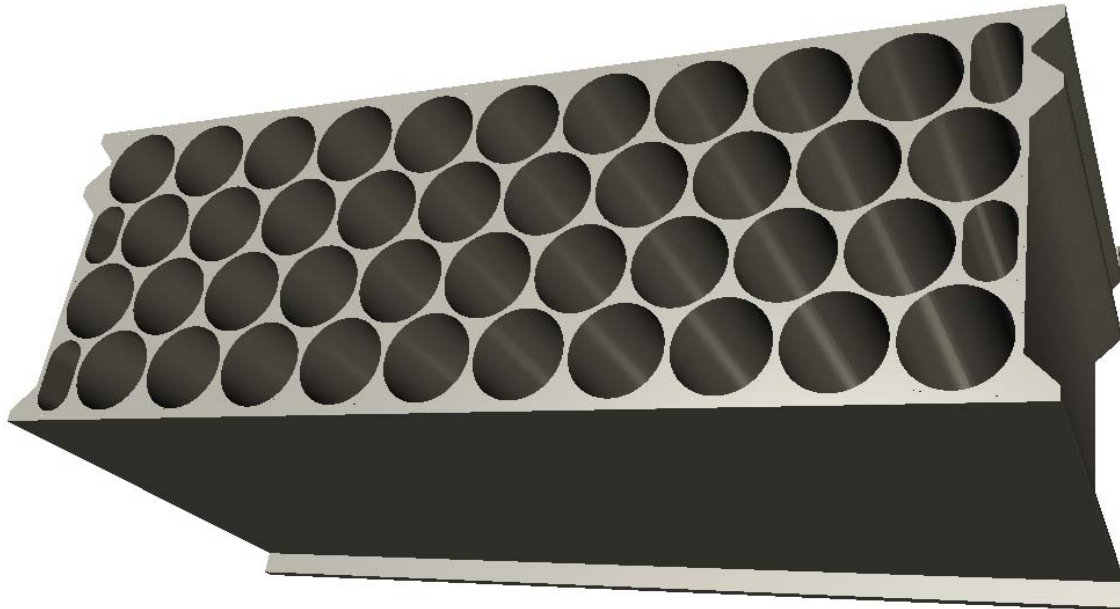


Figura V-24. Infografía del cajón

Datos geométricos del cajón

Eslora*	_____	39.255 m
Manga	_____	13.949 m
Altura Fuste	_____	18.45 m
Canto losa de fondo	_____	0.55 m
Altura total	_____	19.00 m
Diámetro Celdas Circulares	_____	3.51 m
Diámetro Celdas Ovoidales	_____	1.655 m
Altura Celdas Ovoidales	_____	1.179 m
Altura Celdas Ovoidales Esquina	_____	1.571 m
Espesor gargantas interiores	_____	20 cm
Espesor paredes longitudinales	_____	40 cm
Espesor paredes transversales	_____	25 cm
Vuelo transversal losa de fondo	_____	1.0 m

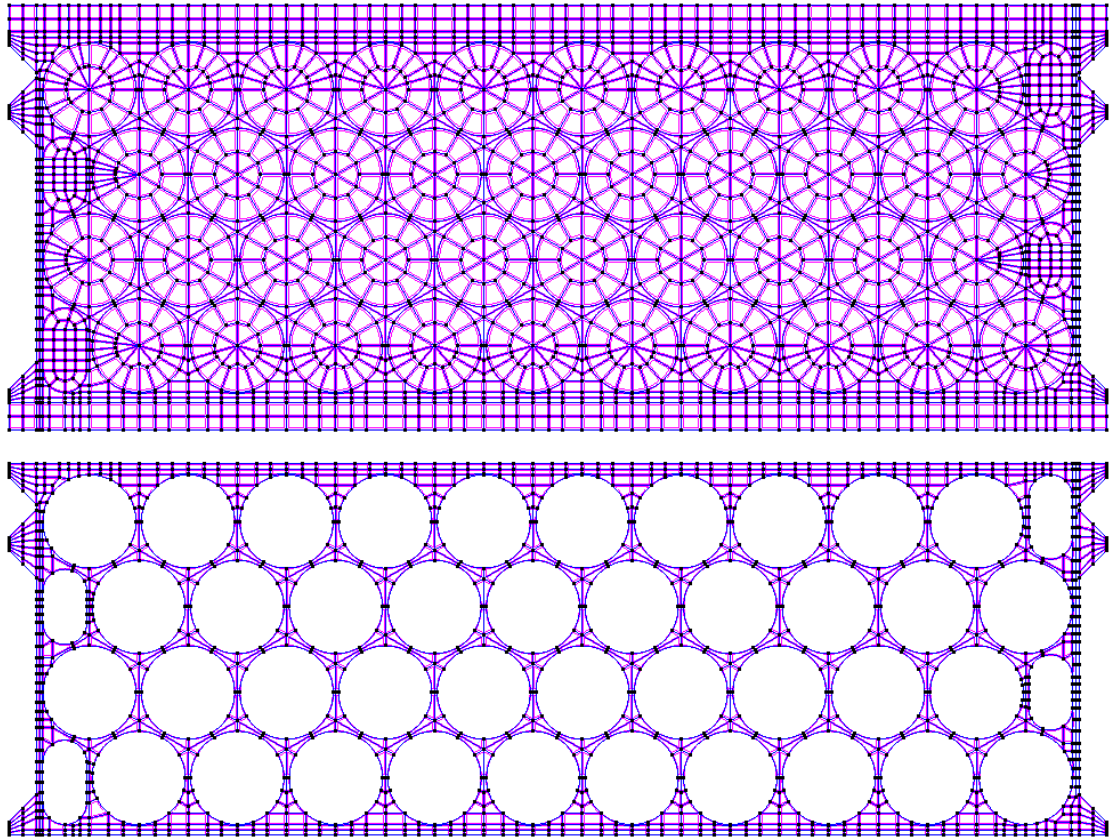


Figura V-25. Geometría modelada. (a) Losa, (b) Fuste.

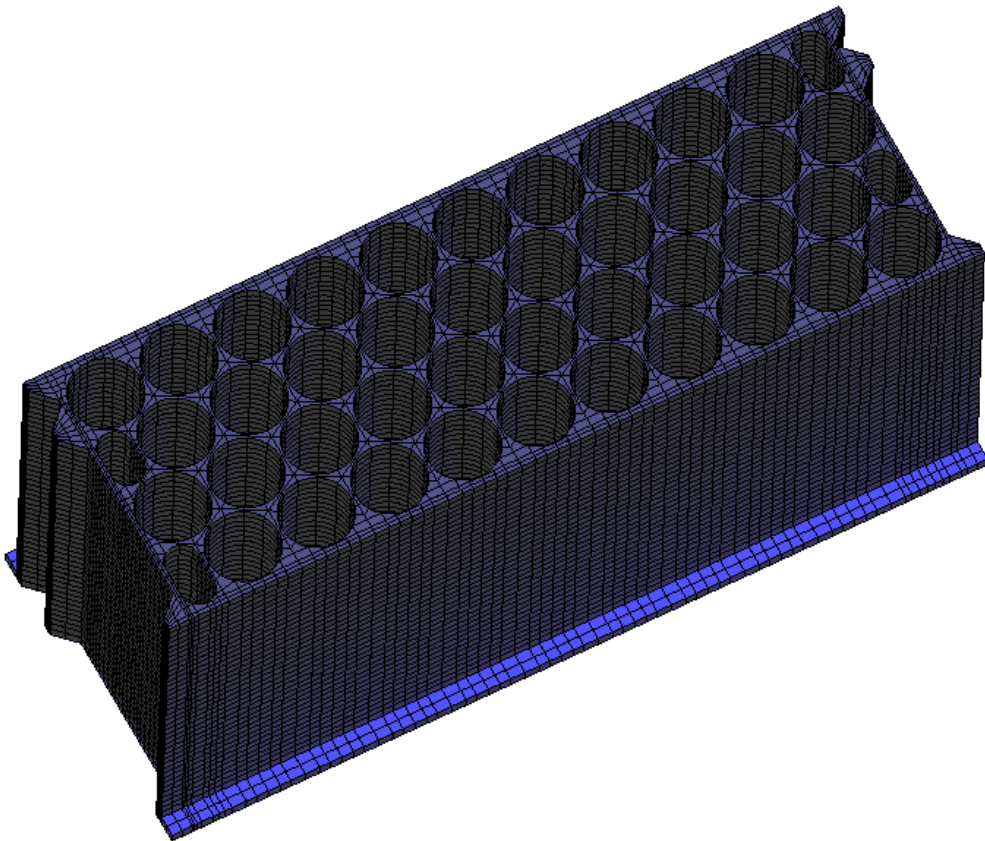


Figura V-26. Vista en 3D del cajón

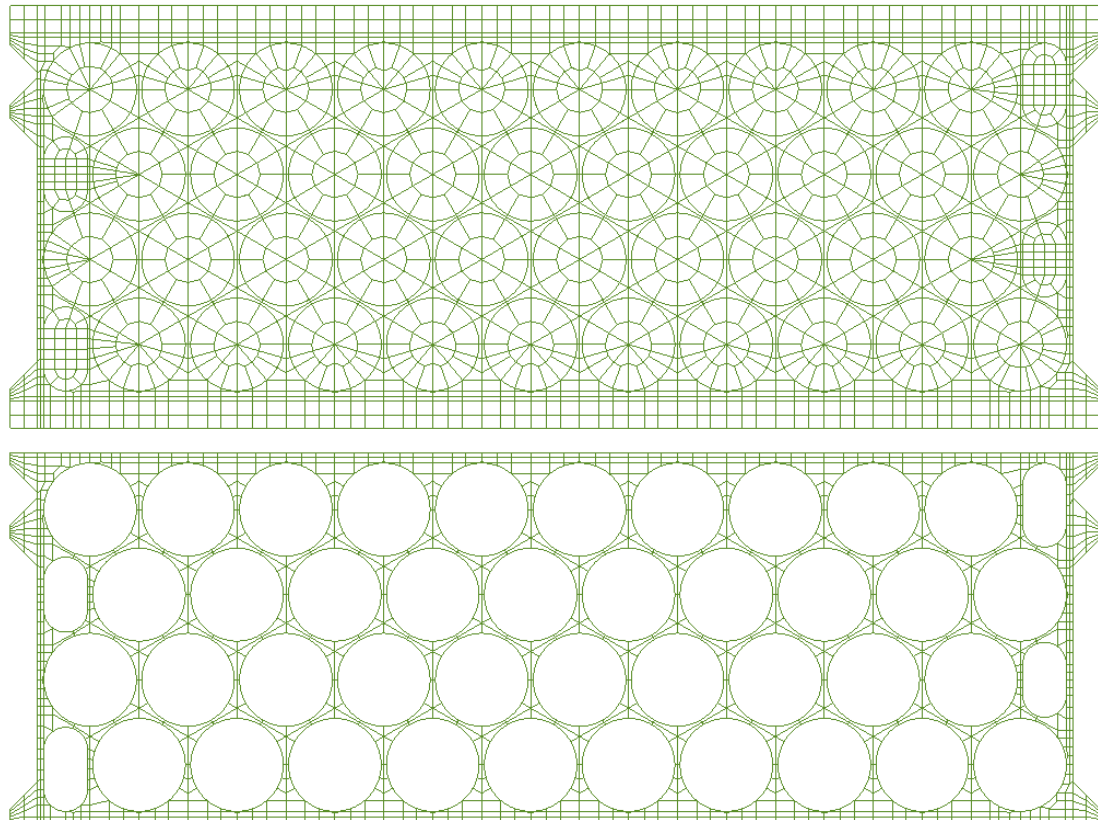


Figura V-27. Malla utilizada. (a) Losa, (b) Fuste.

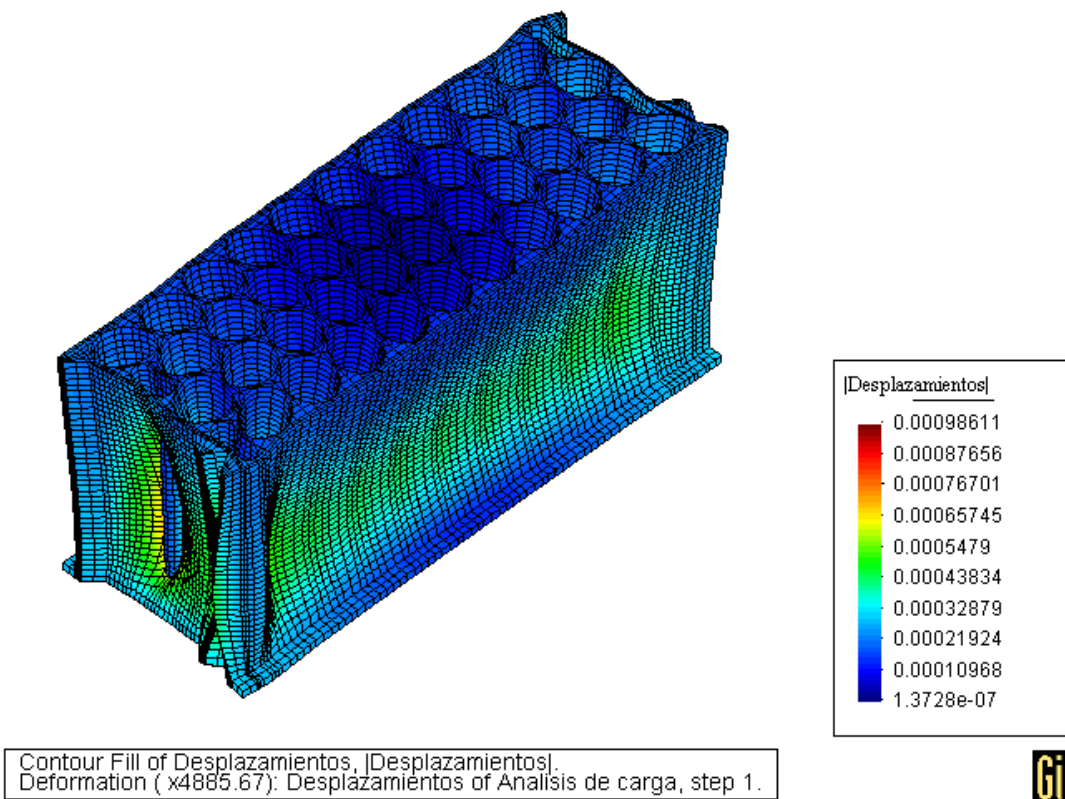


Figura V-28. Vista del desplazamiento resultante en isométrico

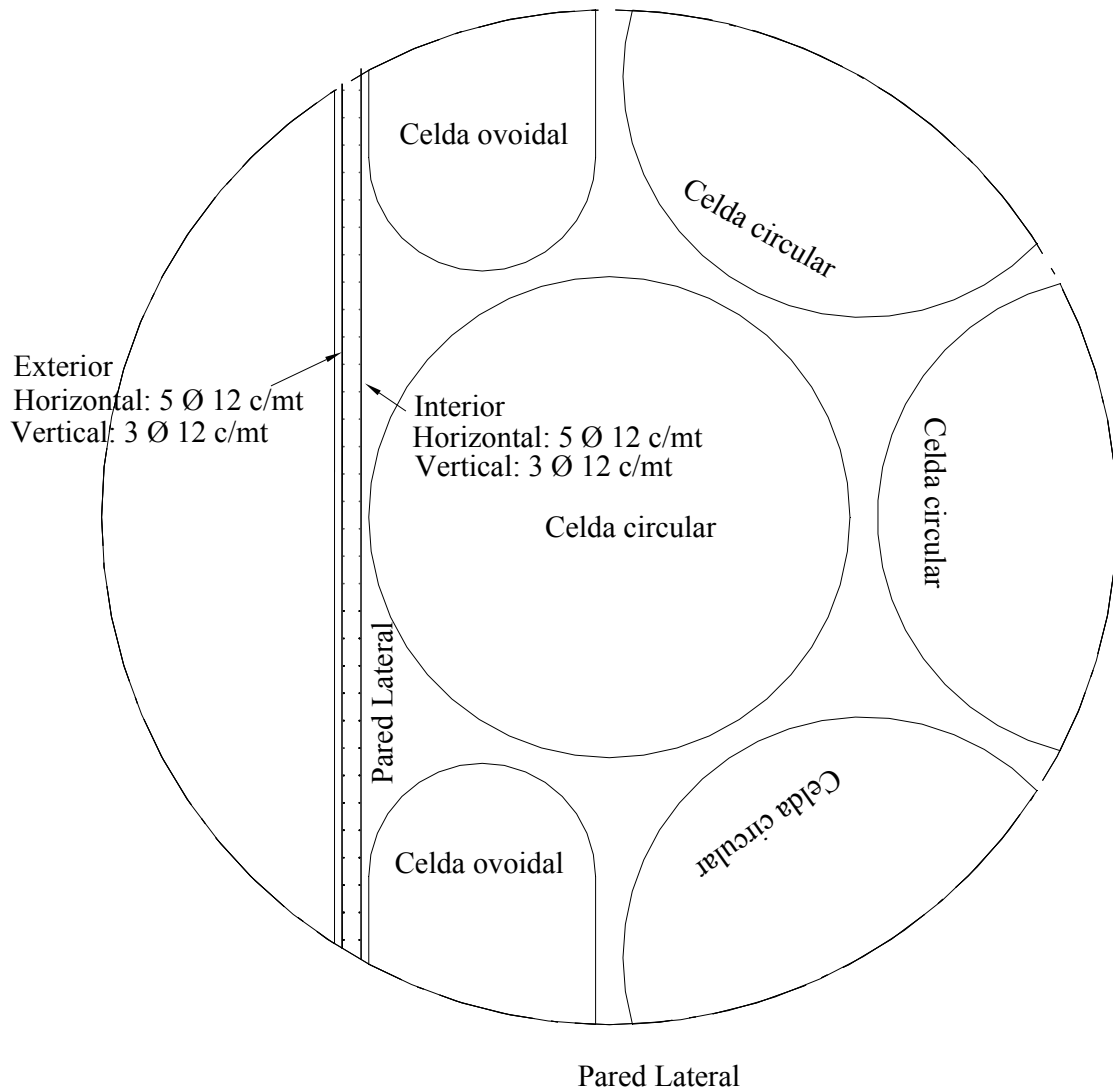


Figura V-29. Esquema de armado de la Pared Lateral

V.2.6.- Octubre 2008

Título del trabajo: ***DISEÑO DE UN CAJON MULTICELULAR TIPO PARA LA AMPLIACIÓN DEL MUELLE ADOSADO (FASE II-C) MUELLE INTERIOR DEL PUERTO DE BARCELONA***



Figura V-30. Infografía del cajón

Datos geométricos del cajón

Eslora* _____	37.40 m
Manga _____	7.323 m
Altura Fuste _____	5.75 m
Canto losa de fondo _____	0.25 m
Altura total _____	6.00 m
Diámetro Celdas Circulares _____	3.51 m
Diámetro Celdas Ovoidales _____	1.655 m
Altura Celdas Ovoidales Esquina _____	1.571 m
Espesor gargantas interiores _____	20 cm
Espesor paredes longitudinales _____	30 cm
Espesor paredes transversales _____	25 cm
Vuelo transversal losa de fondo _____	0.50 m

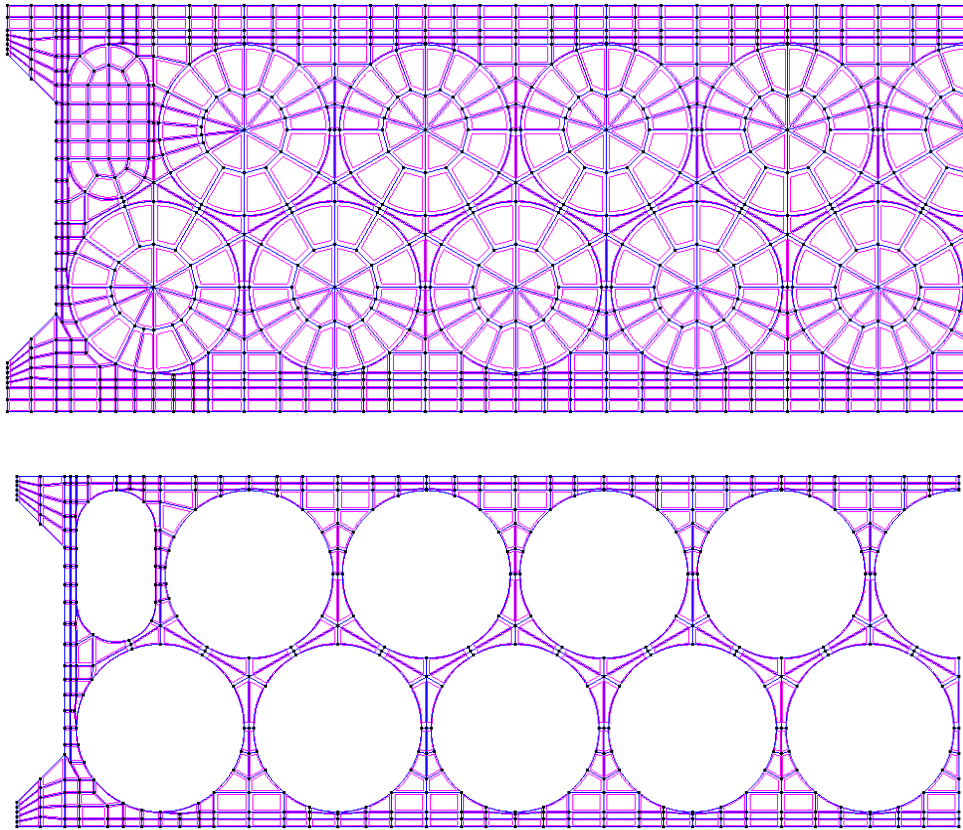


Figura V-31. Geometría modelada. (a) Losa, (b) Fuste.

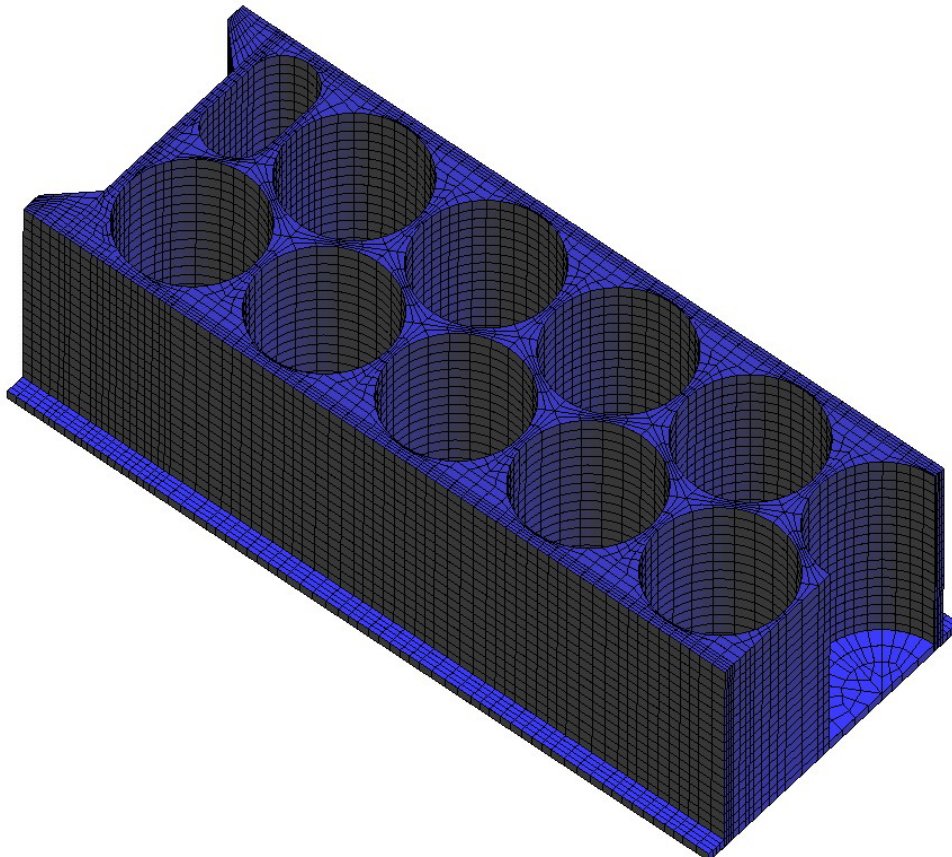


Figura V-32. Vista en 3D del cajón.

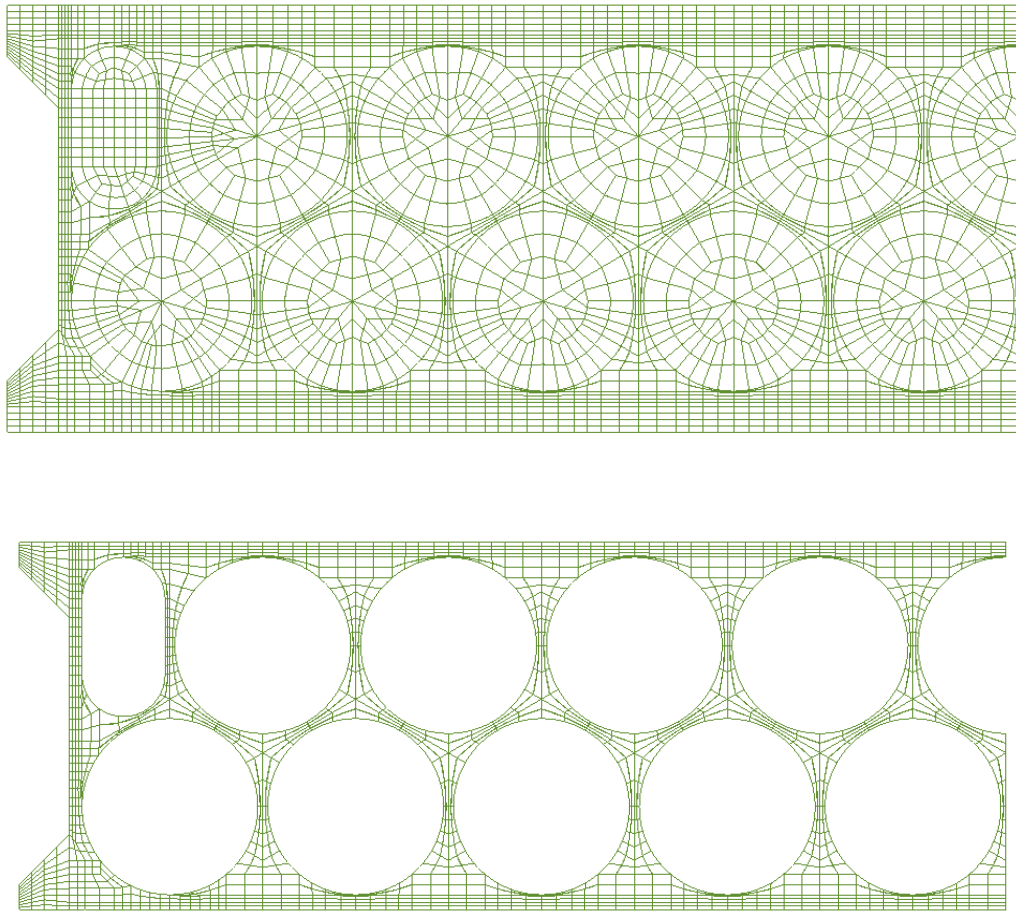
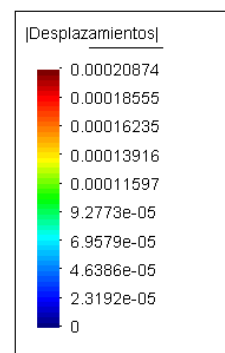
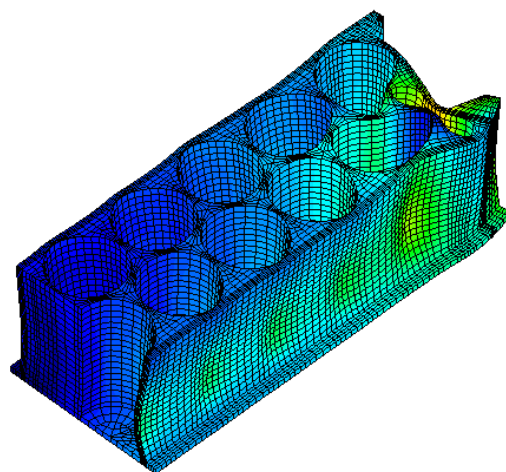


Figura V-33. Malla utilizada. (a) Losa, (b) Fuste.



Contour Fill of Desplazamientos, |Desplazamientos|.
Deformation (x10640.8): Desplazamientos of Analisis de carga, step 1.



Figura V-34. Vista del desplazamiento resultante en isométrico

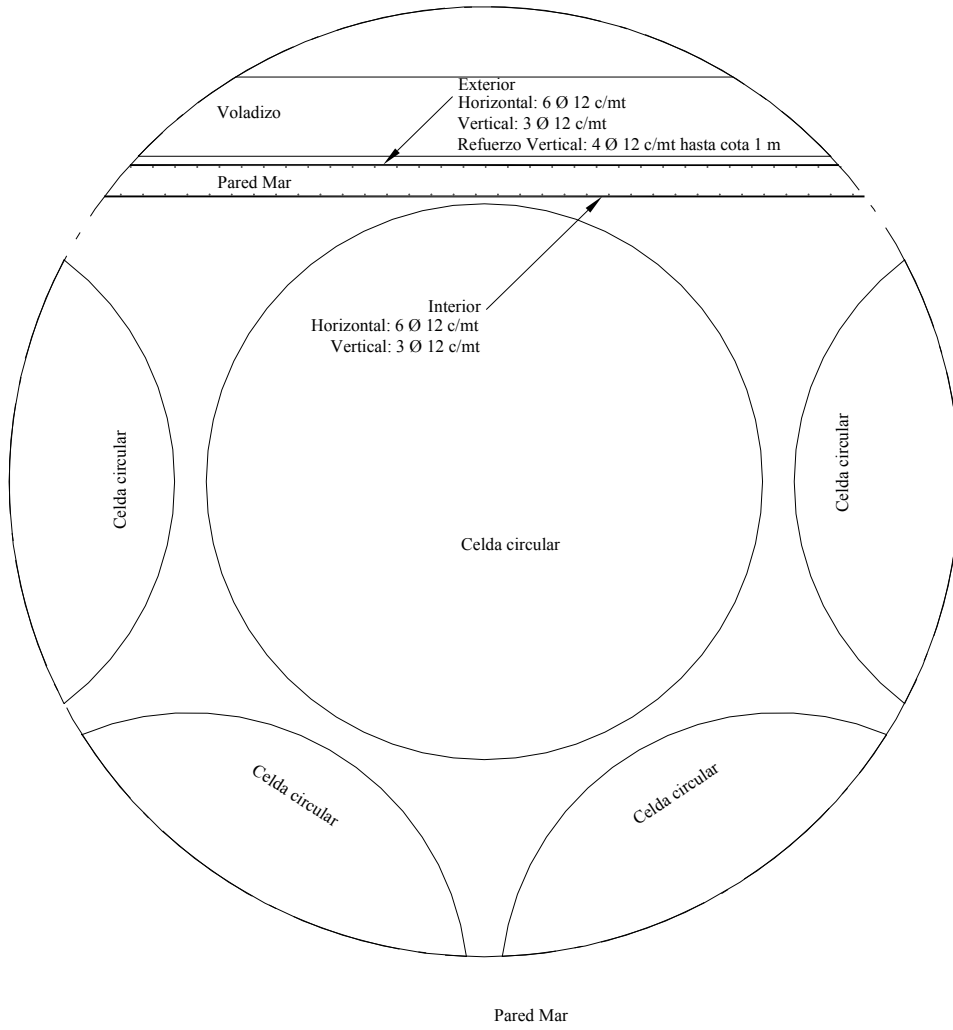


Figura V-35. Esquema de armado de la Pared Lado Mar

V.2.7.- Marzo 2009

Título del trabajo: **ESTUDIO DE VIABILIDAD DE UNA VIGA CANTIL DE ALTAS PRESTACIONES EN EL MUELLE DEL PRAT (FASE II) DEL PUERTO DE BARCELONA**

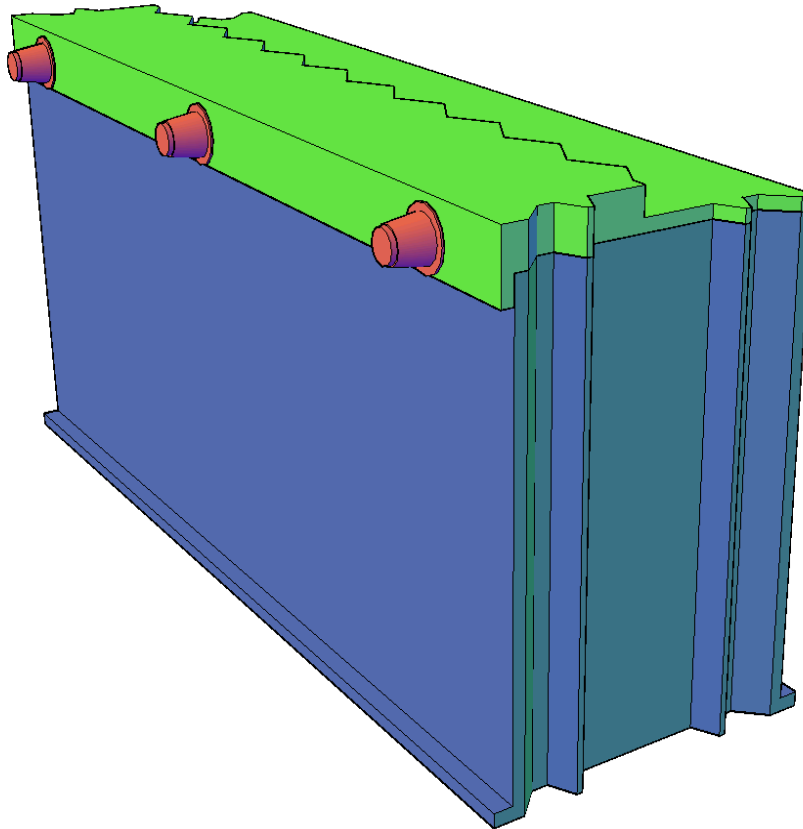


Figura V-36. Infografía del cajón

Datos geométricos del cajón

Eslora* _____	39.255 m
Manga _____	12.074 m
Altura Fuste _____	17.45 m
Canto losa de fondo _____	0.50 m
Altura total _____	18.00 m
Diámetro Celdas Circulares _____	3.51 m
Diámetro Celdas Ovoidales _____	1.655 m
Altura Celdas Ovoidales _____	1.179 m
Altura Celdas Ovoidales Esquina _____	0.580 m
Espesor gargantas interiores _____	20 cm
Espesor paredes longitudinales _____	40 cm
Espesor paredes transversales _____	25 cm
Vuelo transversal losa de fondo _____	0.743 m

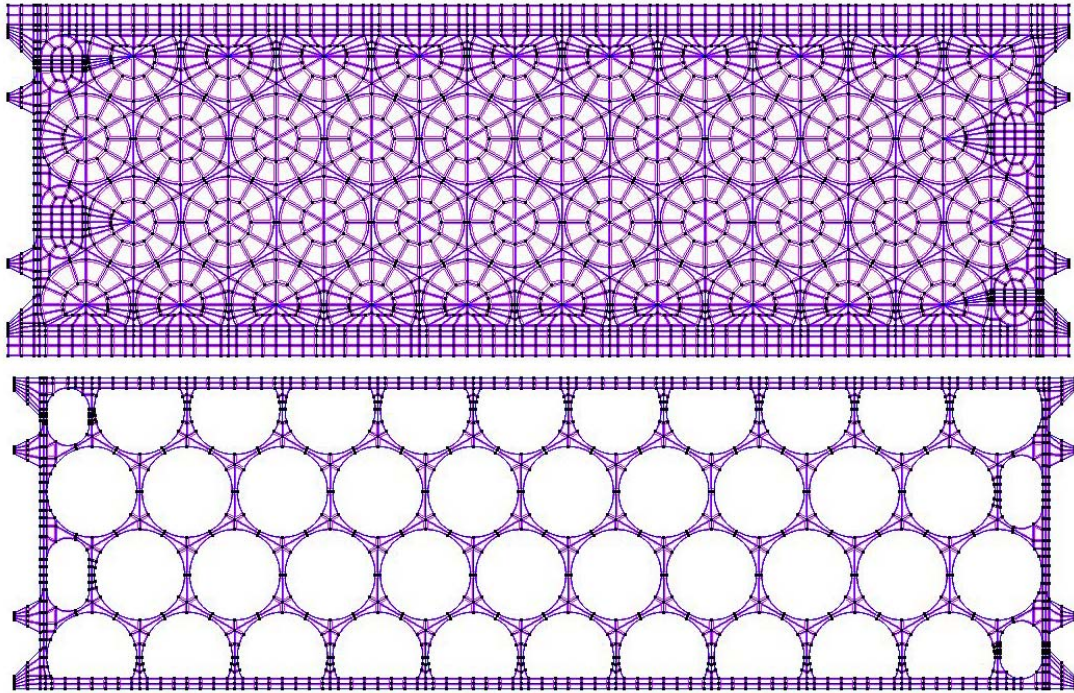


Figura V-37. Geometría modelada. (a) Losa, (b) Fuste.

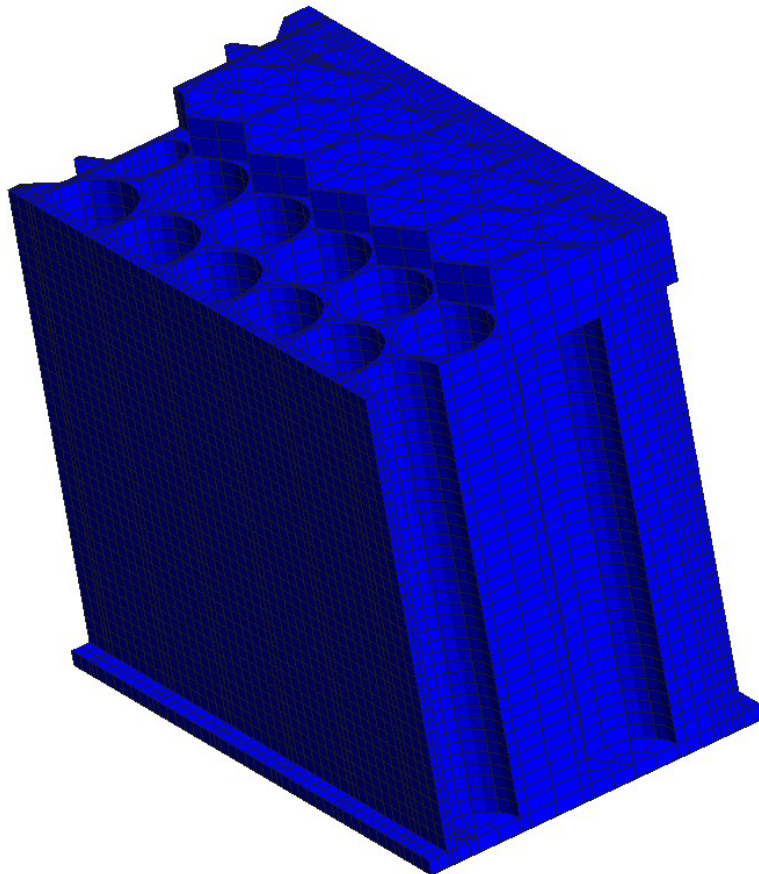


Figura V-38. Vista en 3D del cajón. Alternativa 1

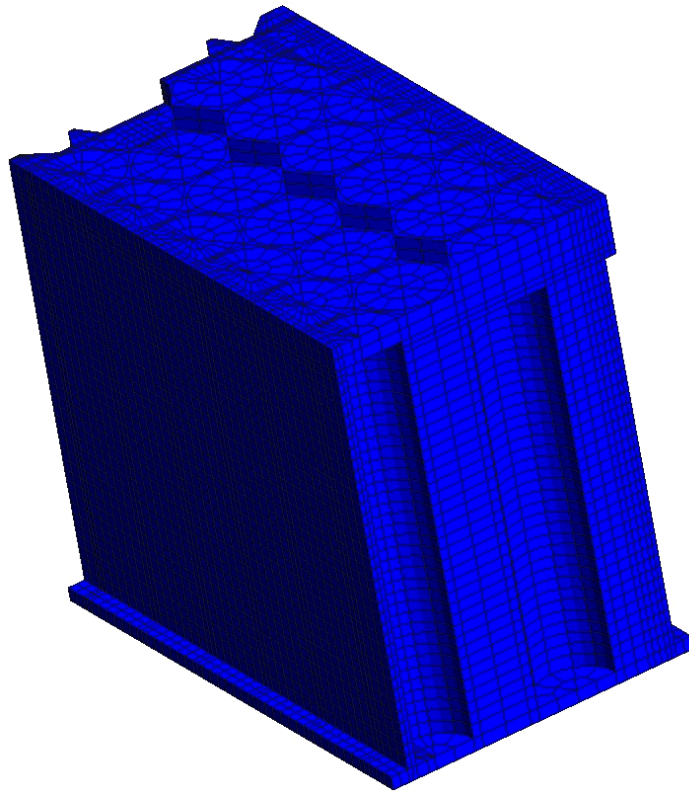


Figura V-39. Vista en 3D del cajón. Alternativa 2

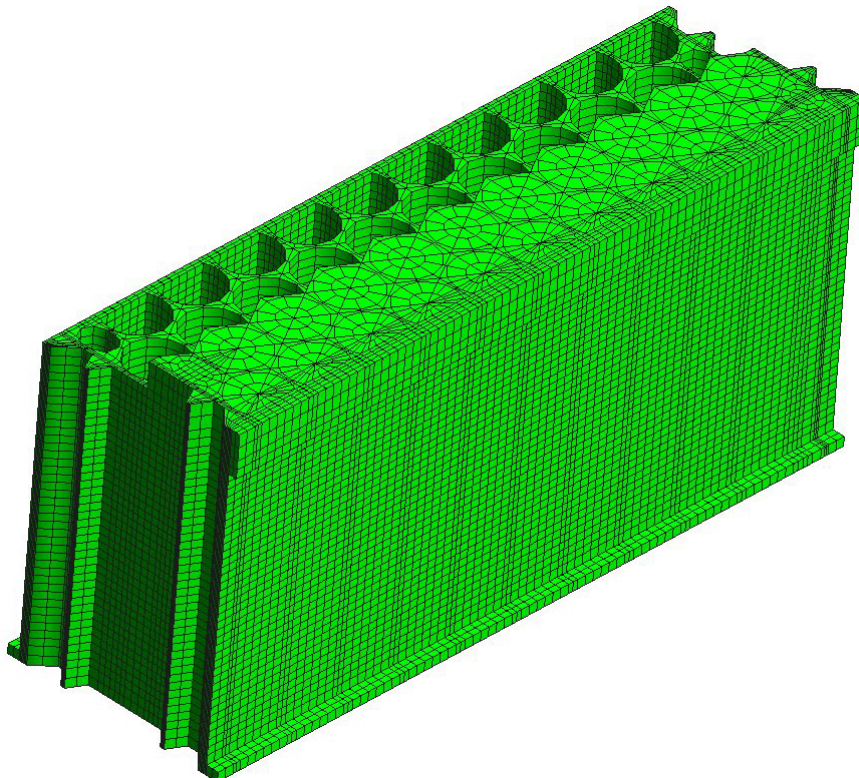


Figura V-40. Vista en 3D de la malla – alternativa 1

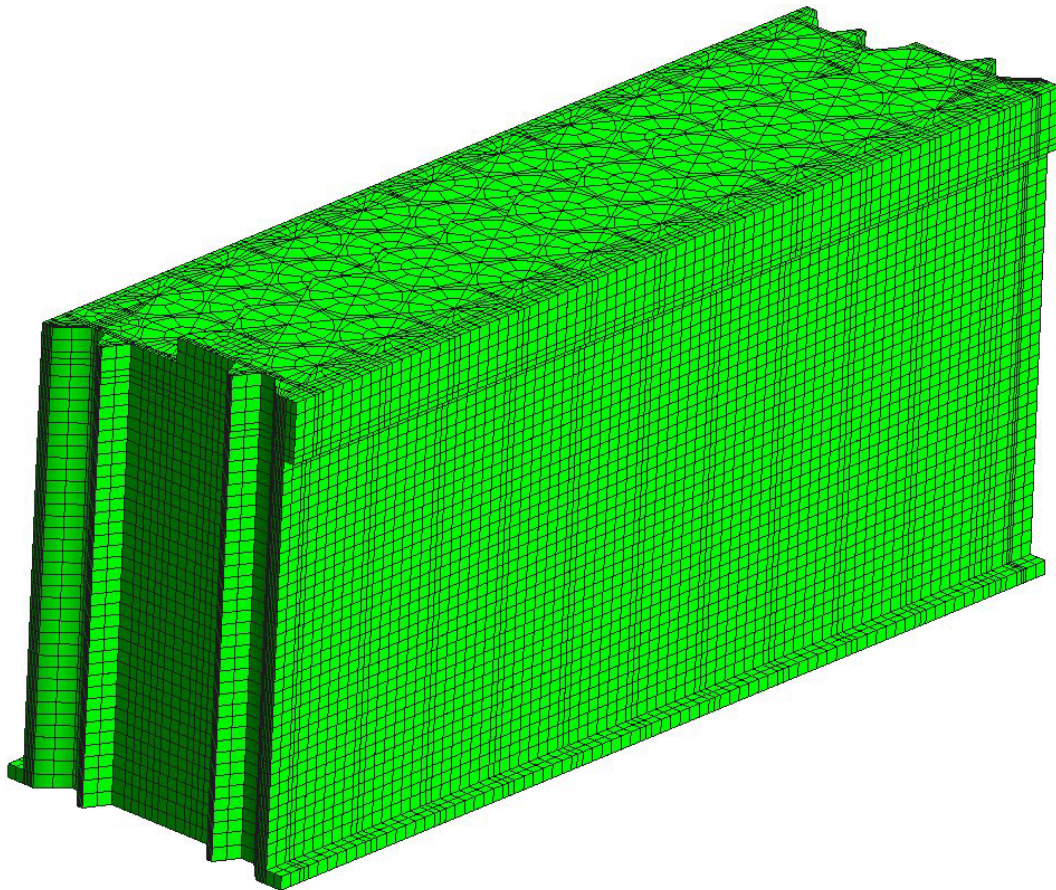


Figura V-41. Vista en 3D de la malla – alternativa 2

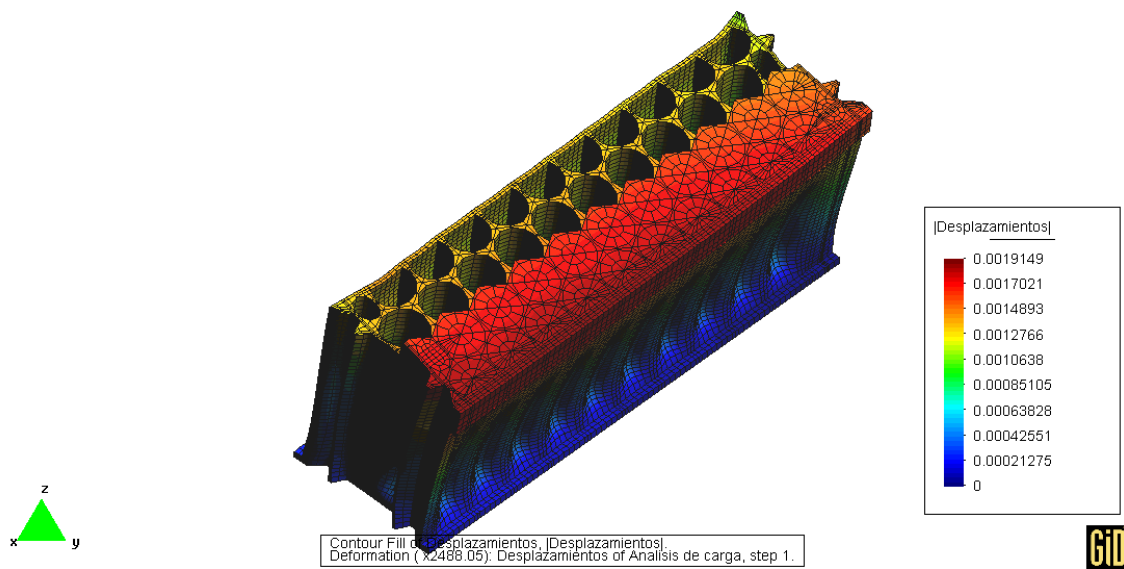


Figura V-42. Vista del desplazamiento resultante en isométrico – alternativa 1

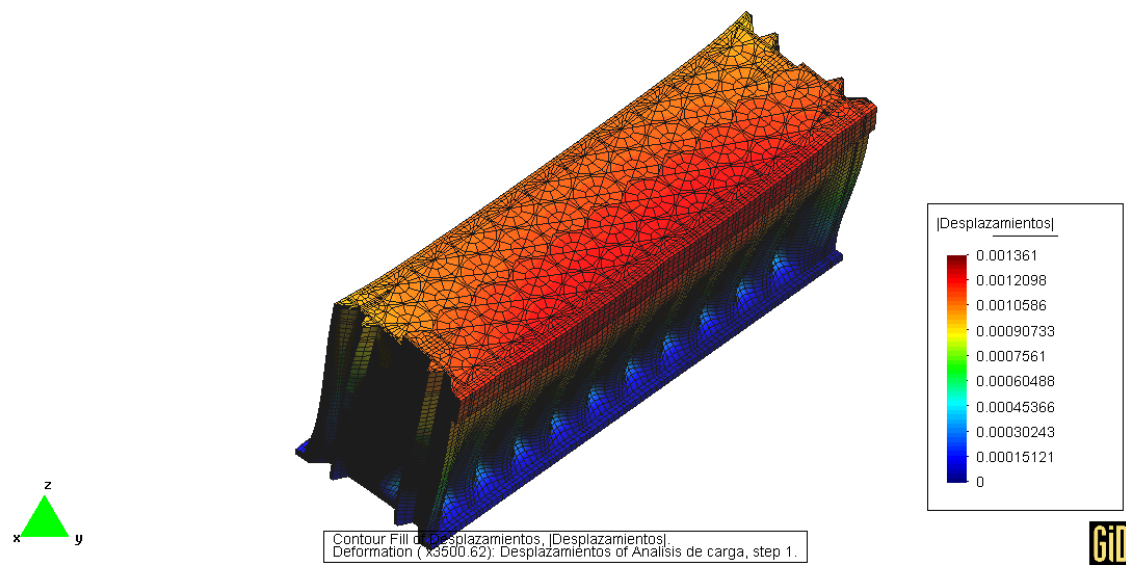


Figura V-43. Vista del desplazamiento resultante en isométrico – alternativa 2



Figura V-44. Vista del muelle de cajones fondeados



Figura V-45. Vista actual del muelle en servicio

V.3.- Puerto de Bilbao



Figura V-46. Vista del Puerto de Bilbao y de los muelles objeto de estudio

V.3.1.- Noviembre 2008

Título del Trabajo: ***DISEÑO DE UN CAJÓN MULTICELULAR TIPO PARA EL MUELLE AZ-2 ADOSADO AL DIQUE DE ZIERBENA DEL PUERTO DE BILBAO***

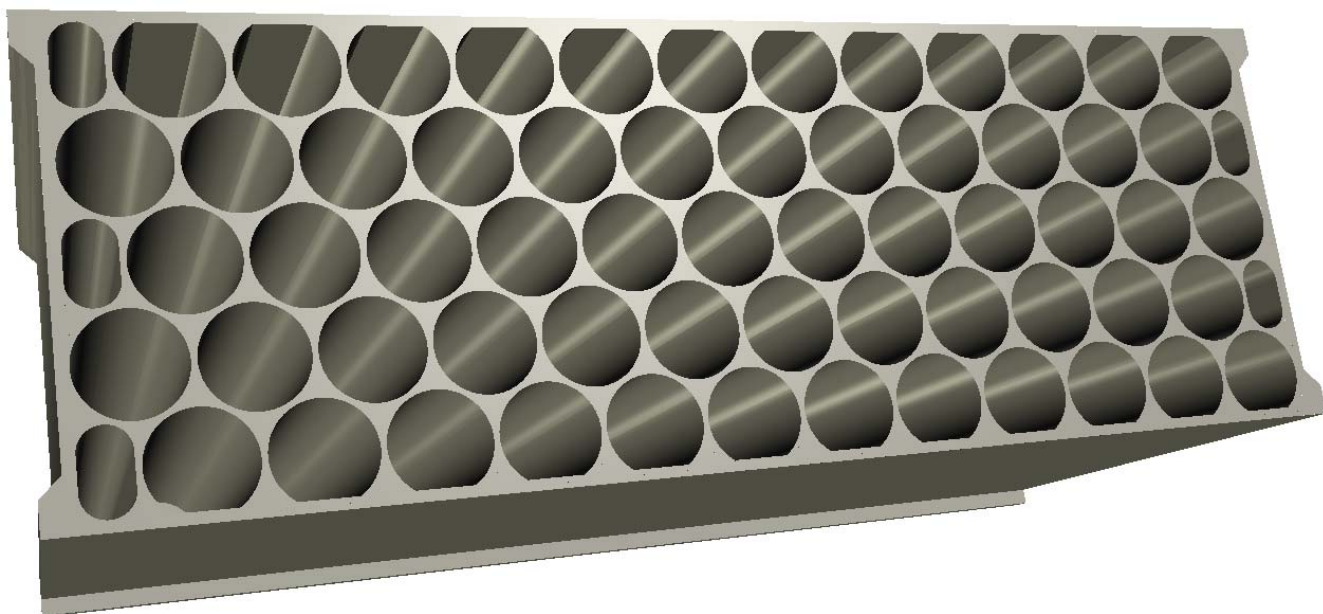


Figura V-47. Infografía del cajón

Datos geométricos del cajón

Eslora* _____	48.805 m
Manga _____	17.16 m
Altura Fuste _____	21.90 m
Canto losa de fondo _____	0.90 m
Altura total _____	24.00 m
Diámetro Celdas Circulares _____	3.65 m
Diámetro Celdas Ovoidales _____	1.725 m
Altura Celdas Ovoidales _____	1.2236 m
Altura Celdas Ovoidales Esquina _____	1.2609 m
Espesor gargantas interiores _____	20 cm
Espesor paredes longitudinales _____	40 cm
Espesor paredes transversales _____	44 cm
Vuelo transversal losa de fondo _____	0.72 m

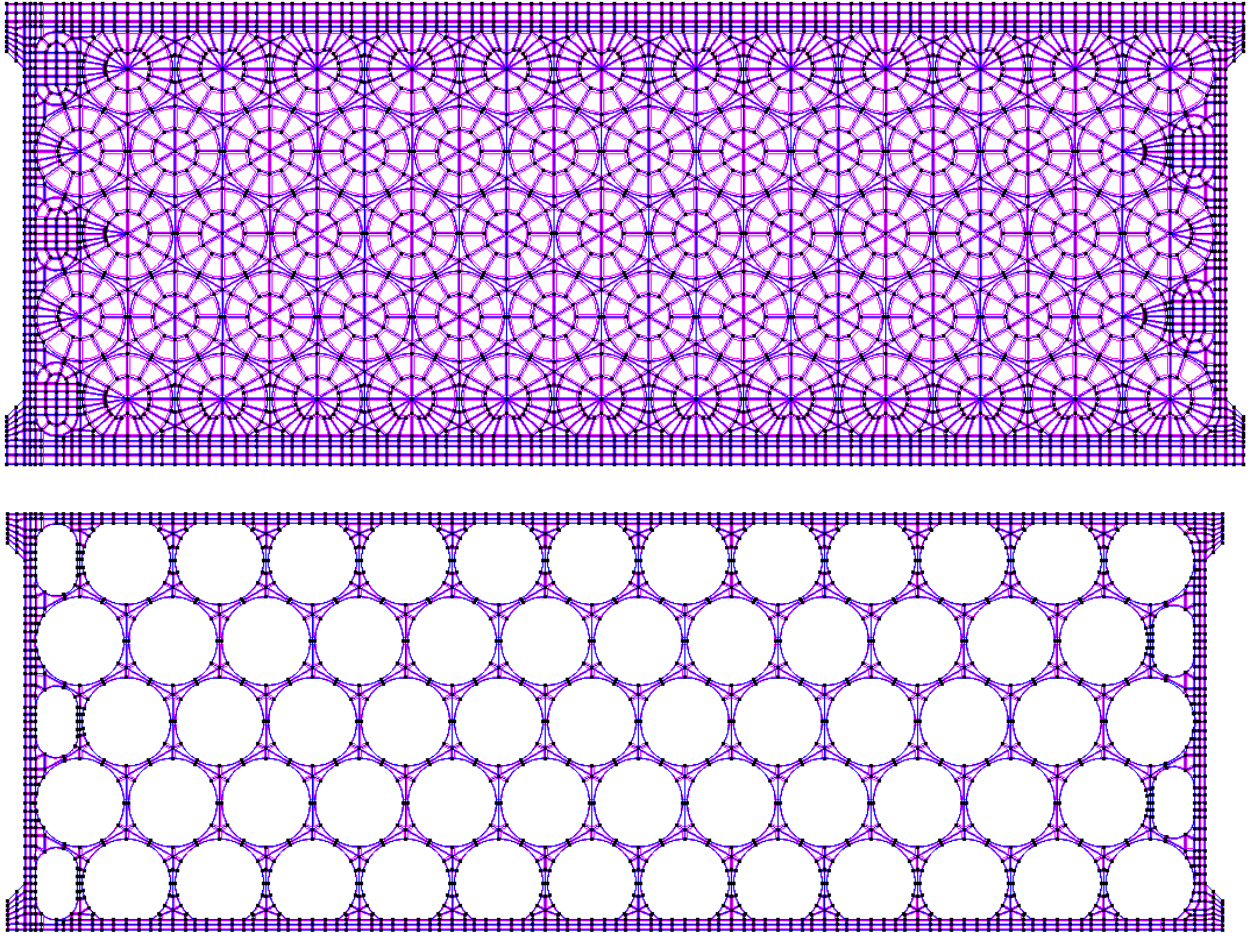


Figura V-48. Geometría modelada. (a) Losa, (b) Fuste

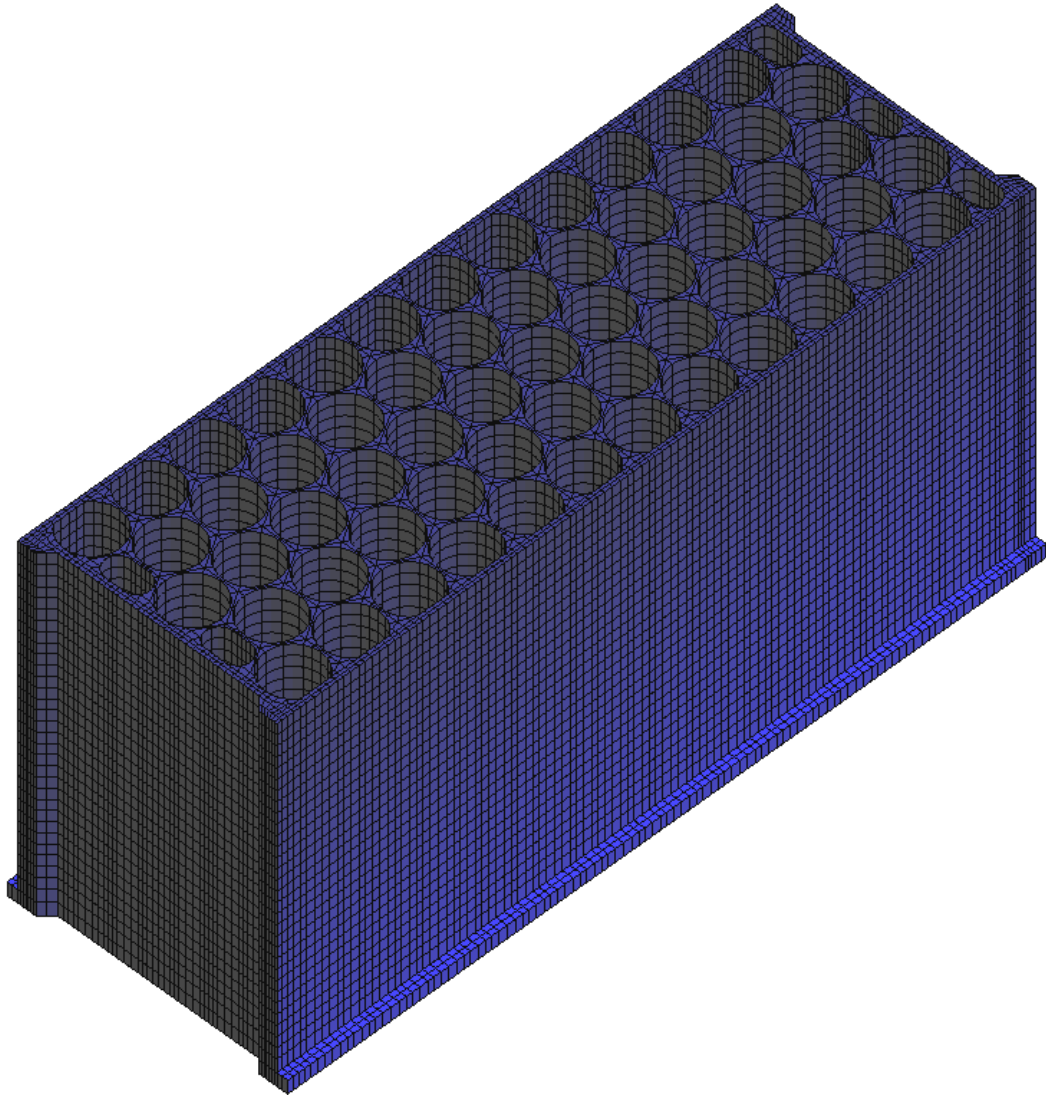
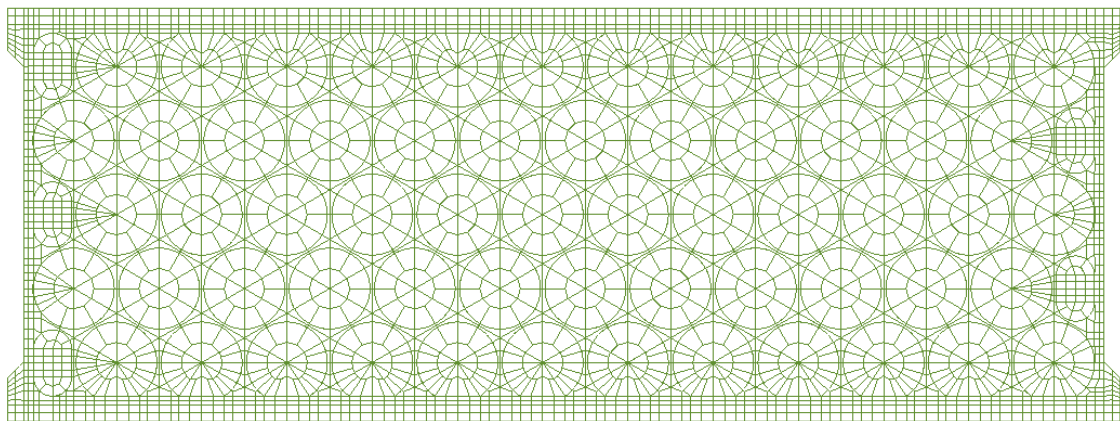


Figura V-49. Vista en 3D del cajón



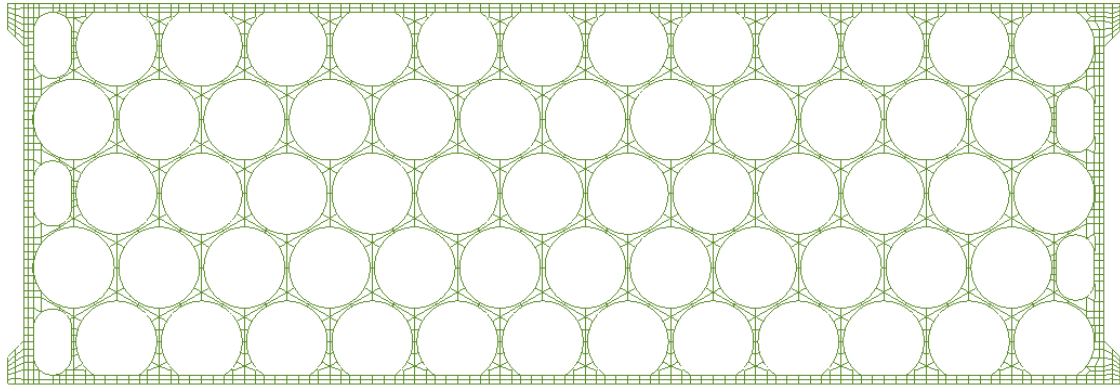
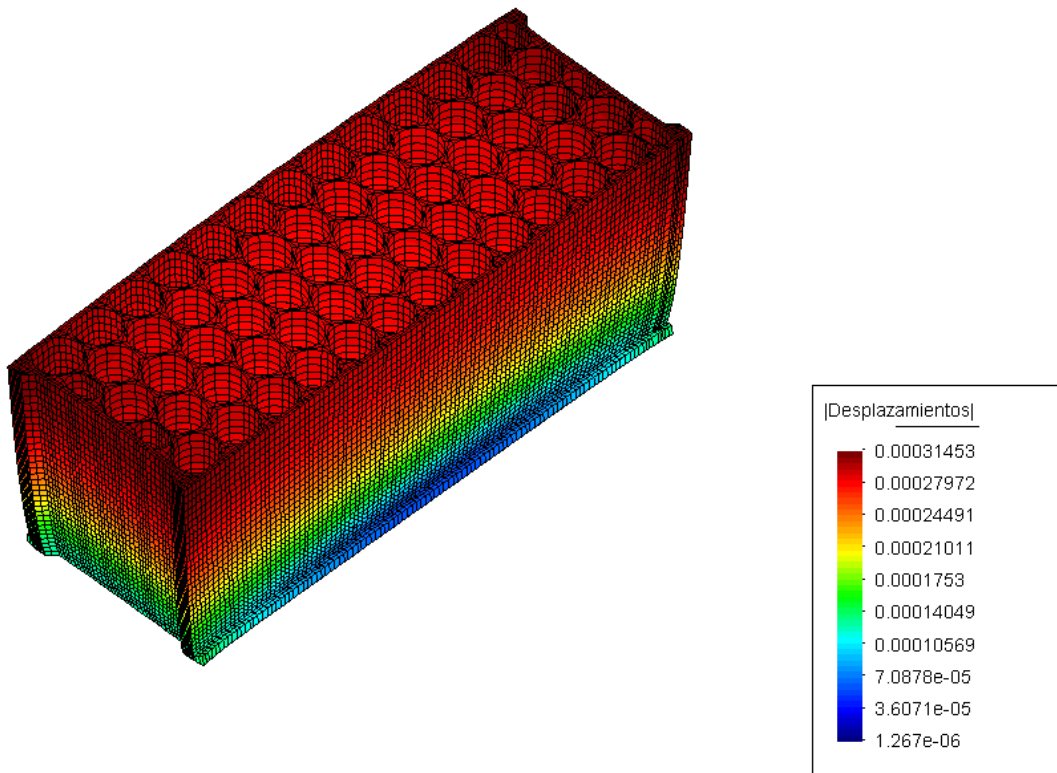


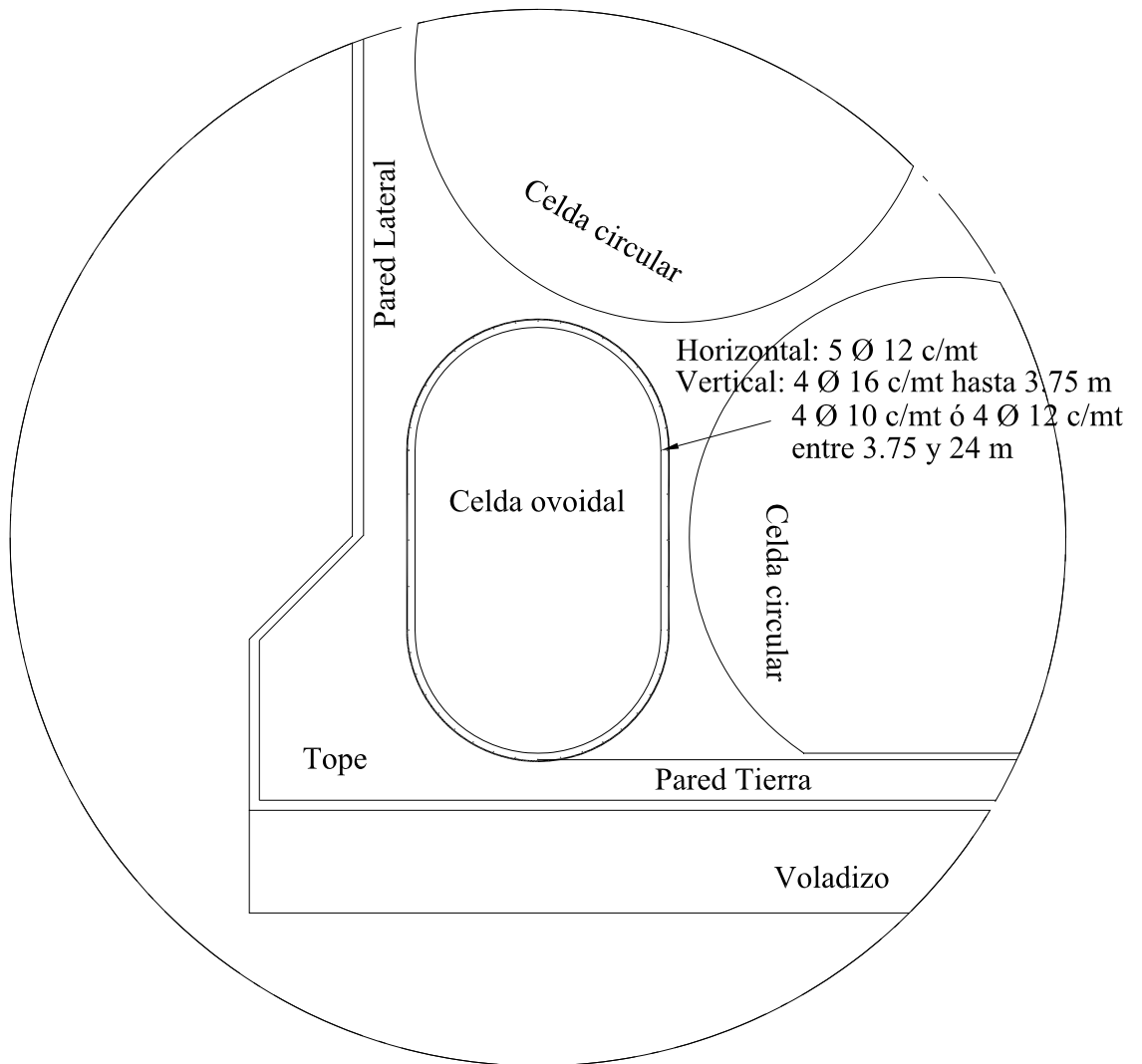
Figura V-50. Malla utilizada. (a) Losa, (b) Fuste.



Contour Fill of Desplazamientos, [Desplazamientos]
 Deformation (x18666.7): Desplazamientos of Analisis de carga, step 1.



Figura 5.1.1.- Vista del desplazamiento resultante en isométrico



Celda Ovoidal Esquina

Figura V-51. Esquema de armado de las Celdas Ovoidales Esquina



Figura V-52. Vista de cajones fondeados del muelle AZ2

V.3.2.- Junio 2009

Título del Trabajo: ***DISEÑO DE UN CAJÓN MULTICELULAR TIPO PARA MUELLE ADOSADO AL CONTRADIQUE DE ALGORTA DEL PUERTO DE BILBAO***

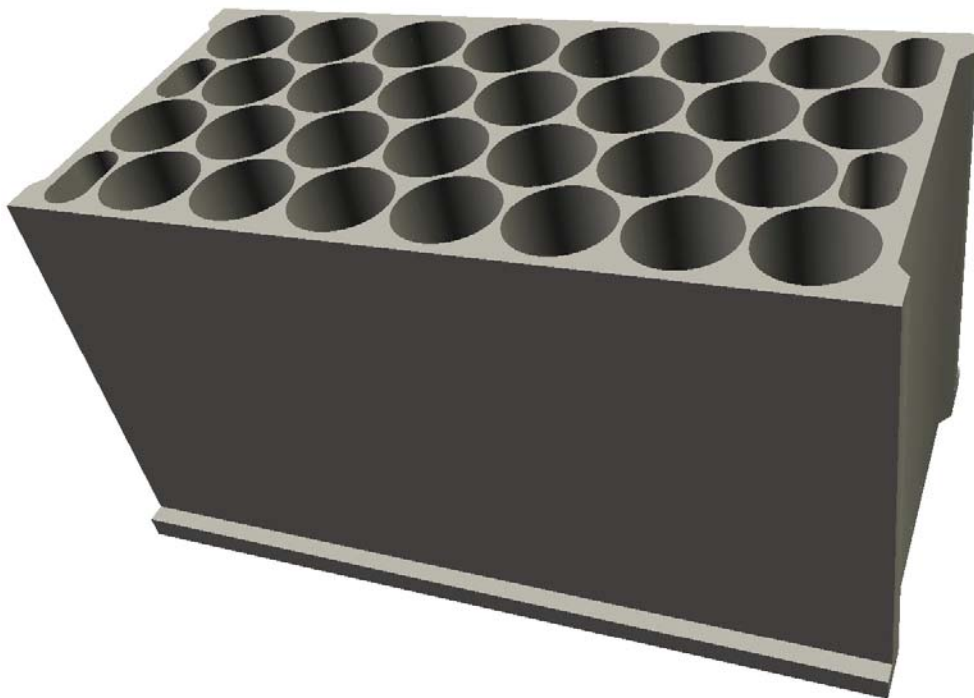


Figura V-53. Infografía del cajón

5. Cajones realizados

Datos geométricos del cajón

Eslora*	_____	28.625 m
Manga	_____	13.949 m
Altura Fuste	_____	15.10 m
Canto losa de fondo	_____	0.90 m
Altura total	_____	16.00 m
Diámetro Celdas Circulares	_____	3.51 m
Diámetro Celdas Ovoidales	_____	1.655 m
Altura Celdas Ovoidales	_____	1.179 m
Altura Celdas Ovoidales Esquina	_____	1.517 m
Espesor gargantas interiores	_____	20 cm
Espesor paredes longitudinales	_____	40 cm
Espesor paredes transversales	_____	50 cm
Vuelo transversal losa de fondo	_____	0.72 m

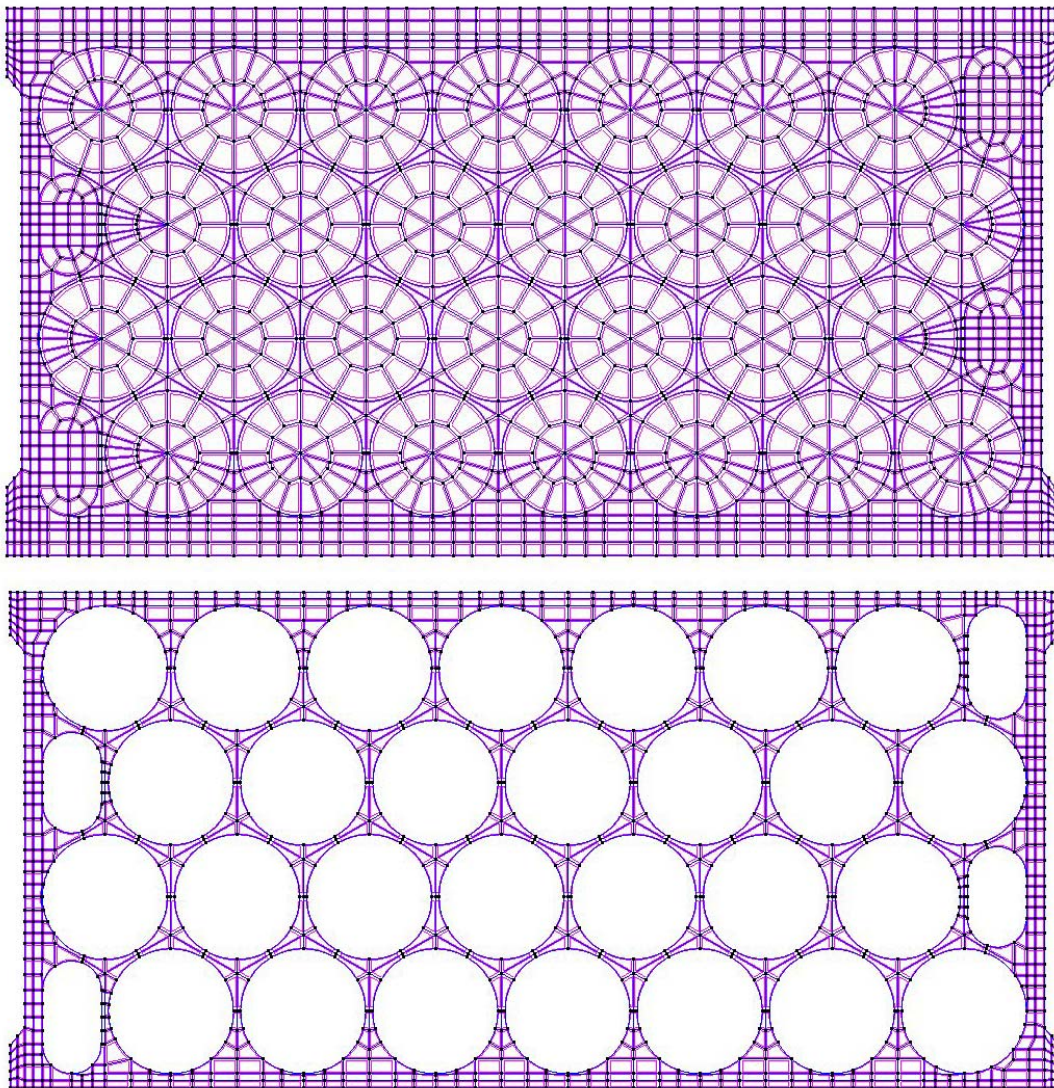


Figura V-54. Geometría modelada. (a) Losa, (b) Fuste

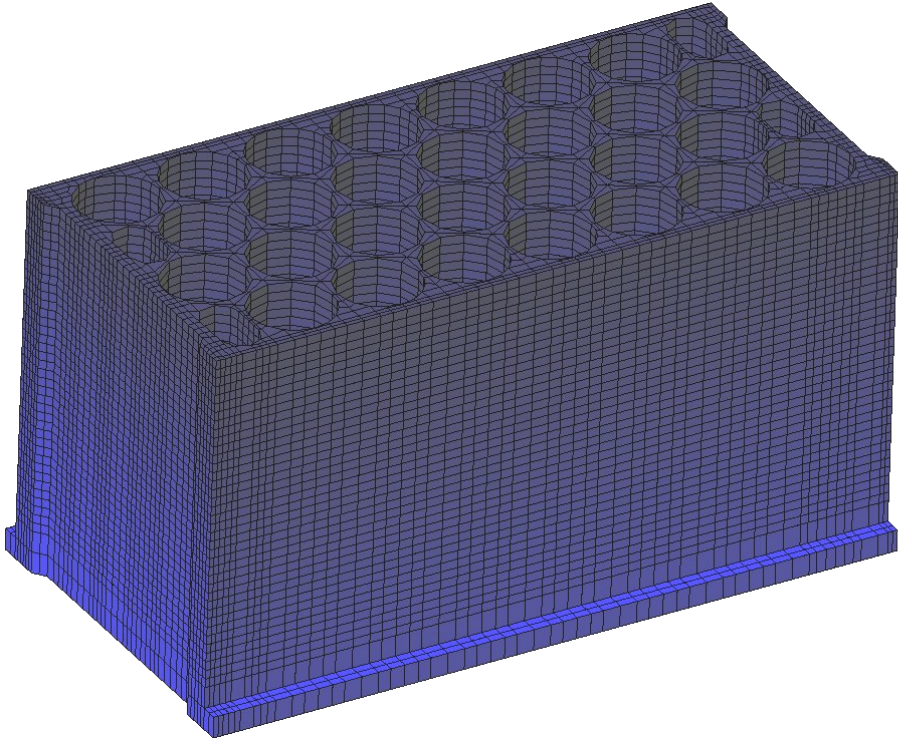


Figura V-55. Vista en 3D del cajón

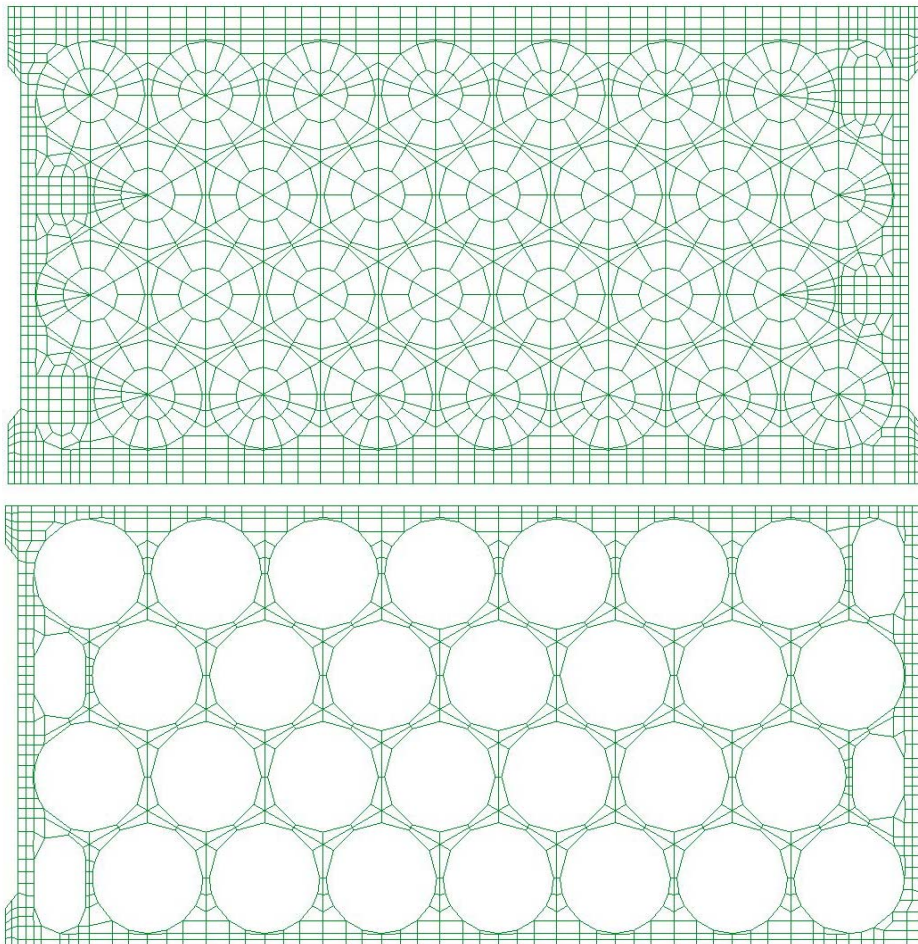


Figura V-56. Malla utilizada. (a) Losa, (b) Fuste.

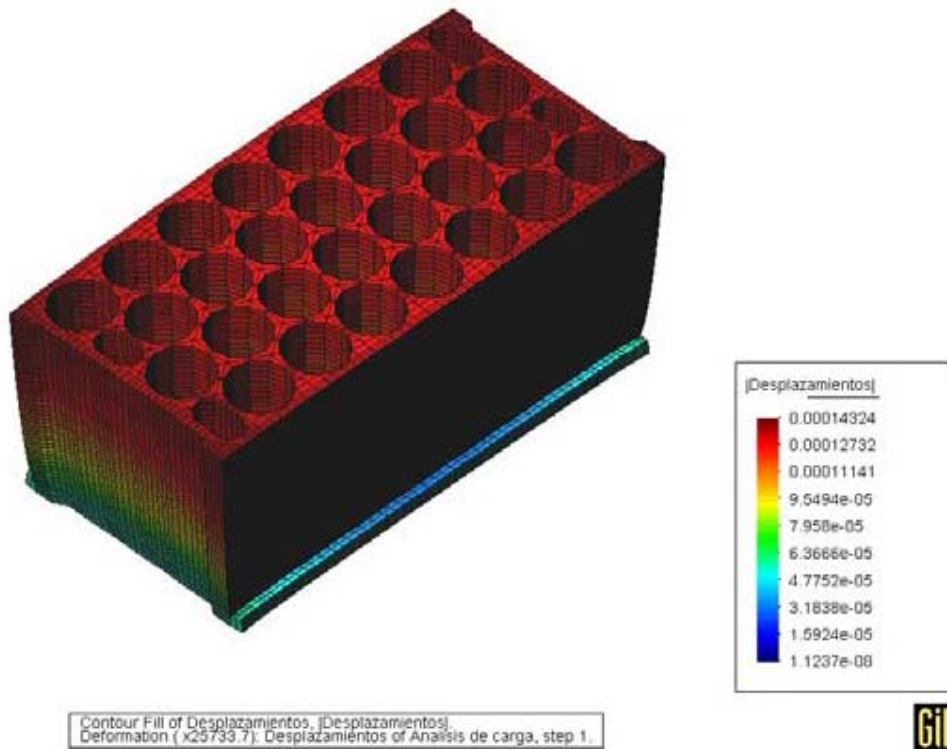


Figura V-57. Vista del desplazamiento resultante en isométrico

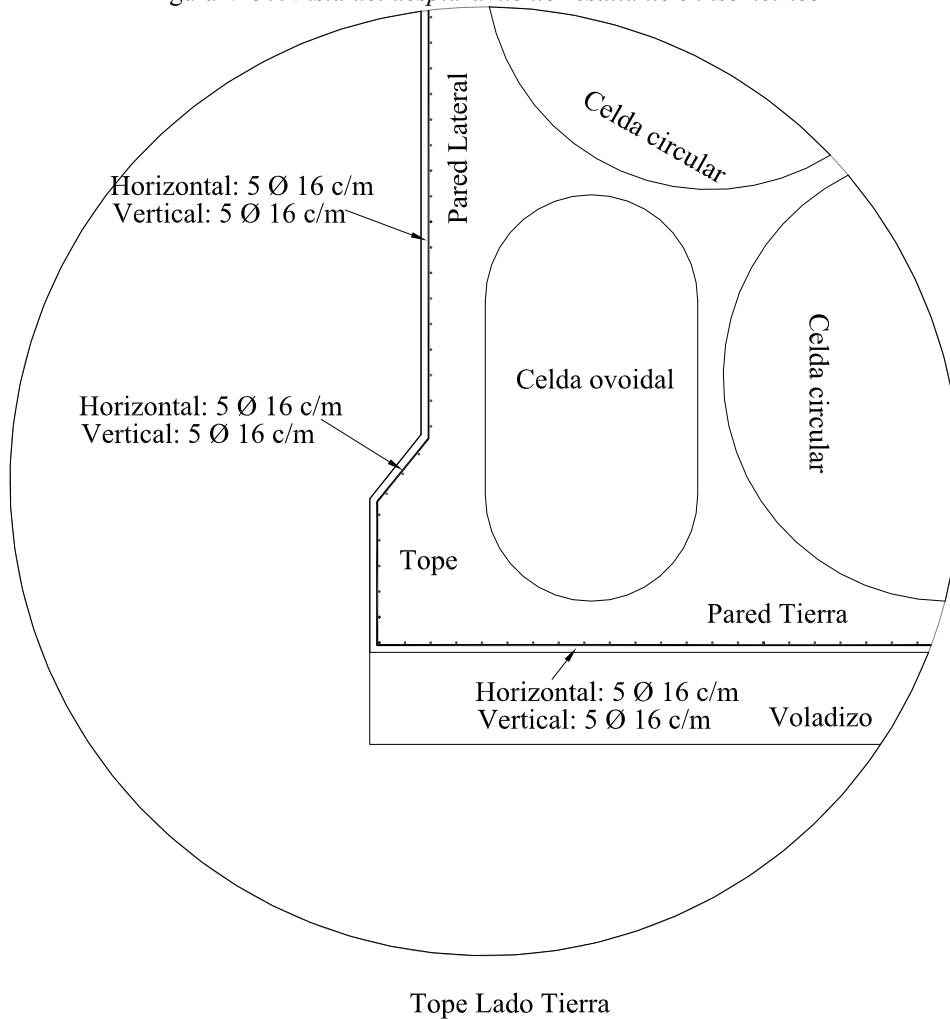


Figura V-58. Esquema de armado de las Celdas Ovoidales Esquina

Capítulo VI. Conclusiones y Líneas Futuras

VI.1.- Aportaciones originales

A lo largo del documento se han presentado las siguientes aportaciones originales, en relación a los cajones multicelulares portuarios.

1. Desarrollo de un tratamiento singular y unificado mediante un enfoque tridimensional a través del análisis automatizado por elementos finitos para el diseño, cálculo y armado de cajones multicelulares portuarios.
2. Desarrollo de una formulación explícita de las ecuaciones de geometría, centros de gravedad e inercia de flotación para cualquier variable que determina la sección transversal del cajón y la estabilidad en flotación.
3. Desarrollo de un algoritmo automático de creación de geometría multibloque para su inmediato mallado tridimensional estructurado con elementos hexaédricos.
4. Desarrollo de un algoritmo de ubicación de secciones críticas, denominadas gargantas, en paralelo con la generación de la geometría multibloque, de manera que en el mallado se ubiquen suficientes de nodos en estas secciones.
5. Desarrollo de un algoritmo de aplicación de condiciones de contorno en cargas en las distintas secciones del cajón
6. Desarrollo de un algoritmo de integración de las tensiones obtenidas en los nodos ubicados en las gargantas para la obtención de los esfuerzos para el armado de estas secciones.

En cuanto al marco de trabajo desarrollado, las aportaciones originarles que se han presentado son los siguientes:

7. Desarrollo de software único de integración de las etapas de pre-proceso, cálculo y post-proceso de un cajón multicelular, en versión standalone, o cliente-servidor.
8. Desarrollo de un entorno profesional de gestión mediante la implementación de servicios cliente/servidor: registro y comprobación de licencias, distribución de las hipótesis de cálculo y su correspondiente resolución por elementos finitos a diversos ordenadores para su cálculo y posterior armado tras integración de las tensiones obtenidas, la posibilidad de obtención de imágenes de los resultados generados, la descarga de ficheros, la comprobación periódica del estado del proceso y el aviso al usuario, etc.

VI.2.- Conclusiones

En primer lugar y, tal y como demuestran los trabajos profesionales realizados para distintas Autoridades Portuarias españolas, el enfoque realizado consistente en el desarrollo de los componentes precisos para la gestión integral de la información que involucra un proyecto completo de diseño, cálculo y armado de un cajón ha demostrado

ser de gran utilidad, dada la compleja información que cada uno de estos proyectos requiere.

Además ha demostrado ser muy versátil, dado que ha permitido la incorporación en sucesivos proyectos de modificaciones en la estructura del cajón y en las condiciones de carga.

La introducción y validación de los diversos algoritmos de automatización desarrollados específicamente para la interacción con software de terceros ha permitido realizar costosos procesos de cálculo de manera totalmente transparente y sin necesidad de usuarios cualificados, lo que tiene especial relevancia a la hora de minimizar los costes, el tiempo de cálculo y principalmente reducir al mínimo los posibles errores humanos.

Durante el desarrollo de la herramienta informática se han generado utilidades y servicios fácilmente extensibles a otras áreas de la ingeniería, por lo que el marco de trabajo presentado cumple con los requisitos que lo definen como tal: la posibilidad de su extensión e implementación en otros entornos.

VI.3.- Líneas futuras

Para finalizar, un proyecto de la envergadura del presentado en esta Tesis, al tiempo que permite resolver con éxito determinados problemas, genera grandes posibilidades de ampliación y aplicación a otros problemas, tanto del ámbito portuario, como de otros ámbitos de la ingeniería.

Dentro del ámbito portuario, se pueden presentar algunas posibilidades de ampliación.

La tipología de cajones puede extenderse a la de cajones de celdas cuadradas, e incluso pueden tratarse otras tipologías de cajones cilíndricos, o con modificaciones estructurales para disipación de energía.

Se puede extender el estudio de elementos finitos a estudios no lineal, para determinar el factor de daño, o para reflejar las condiciones de contorno adecuadas en el fondo, según el contacto del cajón. De esta manera sería posible realizar el armado de la losa a partir de los resultados obtenidos.

Sería posible extender los algoritmos de armado al dimensionamiento de la superestructura en caso de losas o de vigas cantiles.

Otra posible línea de trabajo es en el ámbito de la interacción del cajón con el terreno y con el mar. Dada la capacidad actual de los procesadores sería posible resolver mallas de mayor número de elementos. De emplear el *solver* actual sería preciso realizar una compilación a 64 bits poder utilizar mayor cantidad de memoria en la resolución, ya que actualmente está compilado en 32 bits y tan sólo redirecciona hasta 2 GBytes.

El ámbito de los problemas a resolver podría ampliarse, tratando problemas dinámicos debidos al impacto del oleaje sobre el cajón, entre otros.

Otra línea en la que se puede trabajar es en la paralelización de los cálculos para una misma hipótesis.

En lo que respecta al marco de trabajo, las posibilidades de ampliación pasan por actualizar la versión cliente actual con alguno de los frameworks presentados al inicio. La actualización natural sería realizar un cliente JavaFX. Si bien es multiplataforma y es posible su uso en ordenadores de sobremesa o portátiles con sistemas operativos Windows, Mac o Linux, por el momento no permite su uso en entornos móviles de *smartphones* y *tablets* con sistemas operativos iOS o Android. Si bien en Oracle están trabajando para hacer esto posible en un futuro inmediato.

Sí que es posible ya, mediante el uso de alguno de los frameworks como Vaadin, realizar versiones cliente con interfaz gráfica de usuario amigable para *smartphones* o *tablets*,

Como se ha comentado, el Framework es completamente versátil y es posible reemplazar el motor de cálculo implementado para cajones con otros motores para otras aplicaciones industriales específicas.

Bibliografía

- [1] J. R. Ventosa, *La Ingeniería portuaria en España. La lucha por someter al mar*. Revista del Ministerio de Fomento, 520, 2003, pp. 6-16
- [2] Eric Comerma Piña, *Diseño automatizado de cajones multicelulares flotantes para obras portuarias*, Tesina de Especialidad, Universidad Politécnica de Cataluña, 1999. Tutor: Benjamín Suárez Arroyo
- [3] *Anuario Estadístico del Sistema Portuario de Titularidad Estatal*, Puertos del Estado, 1999
- [4] Pau Esteban, Eric Comerma, “La prefabricación de cajones flotantes: tipologías y métodos constructivos”, 1º PREMIO Bettor MBT 99, ETSECCPB Mayo 1999.
- [5] *Manual para el diseño y la ejecución de cajones flotantes de hormigón armado para obras portuarias*, Puertos del Estado, Ministerio de Fomento, 2006
- [6] *Guía de buenas prácticas para la ejecución de obras marinas*, Puertos del Estado, Ministerio de Fomento, 2008
- [7] C. Rúa Costa, *El sistema portuario español*, UPC 2006
- [8] V. Negro, O. Varela, J. H. García, J. Santos, *Diseño de diques verticales*, CICCIP 2001
- [9] S. Takahashi, *Design of vertical breakwaters*, 25th International Conference on Coastal Engineering, ASCE, 1996
- [10] ROM 0.2-90 *Acciones en el Proyecto de Obras Marítimas y Portuarias* Ministerio de Obras Públicas y Urbanismo, 1990 [derogada]
- [11] ROM 2.0-11 *Recomendaciones para el proyecto y ejecución en las Obras de Atraque y Amarre, Criterios generales y factores de proyecto, Tomos I y II*, Puertos del Estado, Ministerio de Fomento, 2012
- [12] Y. Goda, *Random Seas and Design of Maritime Structures*, World Scientific, Singapore, pp. 1-443.
- [13] I. Hiroi, *On a method of estimating the force of waves*, Tokyo University Engineering Reports, X, 1919
- [14] M. Sainflou, *Essai sur les Diques Maritimes Verticales*, Annales des Ponts et Chaussées., Vol. 98 (1928) (Pt. 1, Tome 11)
- [15] Y. Goda, *New wave pressure formulae for composite breakwaters*, Proc. 14th Int. Conf. on Coastal Engineering, ASCE (1974), pp. 1702–1720
- [16] B. Suárez, A. D. Hanganu, J. M. Canet, *Optimum Design of Multicellular Reinforced Concrete Box Docks*, Journal of Structural Engineering, Vol. 128-5, 2002, pp. 603-611
- [17] FLAC/Slope, Itasca Consultores, S.L., <http://www.itascacg.com/flacslope/>
- [18] E. Pita, J. Vázquez, Comité Técnico de Ingeniería Portuaria, Asociación Técnica de Puertos y Costas, *Análisis de Estructuras Bidimensionales mediante programas de estabilidad de taludes*, Revista Puertos, 138, 2007
- [19] M. Carreiro, *Sobre la respuesta estática de suelos finos blandos bajo diques verticales*, Tesis Doctoral, ESICCP, UPM, 2007
- [20] M. Martín, *Sobre la respuesta dinámica del terreno bajo la acción del oleaje en cajones fondeados en suelos arcillosos*, Tesis Doctoral, ESICCP, UPM, 2010
- [21] E. Oñate, S. Idelsohn, F. Del Pin, R. Aubry, *The particle finite element method. An overview*, International Journal of Computational Methods, Vol 1, 2004, pp.

- 267-307
- [22] E. Oñate, B. Suárez, S. Idelsohn, R. Rossi, M. A. Celigueta, A. L. De Tetto, *Informe Técnico sobre la avería del Muelle del Prat*, CIMNE IT-514, Julio 2007
- [23] Cyes, *Obras Marítimas*, 2011
- [24] K. Tanimoto, S. Takahashi, *Design and construction of caisson breakwaters — the Japanese experience*, Coastal Engineering, 22, Issues 1–2, 1994, pp. 57-77
- [25] L. Franco, *Vertical breakwaters: the Italian experience*, Coastal Engineering, 22, Issues 1–2, 1994, pp. 31-55
- [26] *Ampliación del Puerto de Cartagena*, Arte y Cemento, 23, 2004, pp 102-103
- [27] *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Standard No.: 610.12-1990, The Institute of Electrical and Electronics Engineers, Inc., NY
- [28] W. Humphrey, *Managing the Software Process*. Reading, MA: Addison-Wesley. 1989, p. 247-286
- [29] I. Jacobson, G. Booch y J. Rumbaugh, *The Unified Software Development Process*, 1999, Addison-Wesley
- [30] *IEEE Guide for Developing Software Life Cycle Processes*, IEEE Standard No.: 1074-1997, The Institute of Electrical and Electronics Engineers, Inc., NY
- [31] R. Zavala, *Diseño de un Sistema de Información Geográfica sobre Internet*. Tesis de Maestría en Ciencias de la Computación. Universidad Autónoma Metropolitana-Azcapotzalco. México, D.F., 2000, <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>
- [32] K. C. Kang, *Feature-Oriented Development of Applications for a Domain*. Proceedings of the Fifth International Conference on Software Reuse, ICSR-5, pp. 354-355. 1998, Victoria, B.C., Canada. IEEE-CS.
- [33] L. Fuentes y A. Vallecillo, *Arquitecturas Software y Marcos de Trabajo*, GISUM: Grupo de Ingeniería del Software de la Universidad de Málaga, Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga. España, 2001
- [34] M. E. Markiewicz, C. J. P. Lucena, *Object Oriented Framework Development*, ACM CrossRoads, 2001, <http://www.acm.org/crossroads/xrds7-4/frameworks.html>
- [35] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley, 1999
- [36] R. E. Johnson, *Frameworks = (Components + Patterns)*, Communications of the ACM, 40 (10), 1997, pp. 39 - 42
- [37] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995
- [38] European Organization for Nuclear Research CERN, <http://home.web.cern.ch/about/birth-web>
- [39] World Wide Web Consortium W3C, <http://www.w3.org/Proposal.html>
- [40] World Wide Web Consortium W3C, http://www.w3.org/History/1994/WWW/Journals/CACM/screensnap2_24c.gif
- [41] Lynx, <http://lynx.isc.org/>
- [42] Mosaic, <http://www.ncsa.illinois.edu/Projects/mosaic.html>
- [43] <http://gladiator.ncsa.illinois.edu/Images/press-images/mosaic.6beta.tif>
- [44] A. Grosskurth, M. W. Godfrey, *A reference Architecture for Web Browsers, Proceedings of the 21st IEEE International Conference on Software Maintenance, ICSM'05*, 2005, pp. 661- 664

- [45] Common Gateway Interface CGI, <http://www.w3.org/CGI>, <http://tools.ietf.org/html/rfc3875>
- [46] FastCGI, <http://www.fastcgi.com/devkit/doc/fcgi-spec.html>
- [47] SCGI, <http://www.python.ca/scgi/protocol.txt>
- [48] W3Techs Web Technology Surveys, <http://w3techs.com/technologies/>
- [49] Netscape History, http://www.holgermetzger.de/Netscape_History.html
- [50] Oracle iPlanet Web Server, <http://www.oracle.com/technetwork/middleware/iplanetwebserver-098726.html>
- [51] Oracle Glassfish Server, <http://www.oracle.com/technetwork/middleware/glassfish/overview/index.html>
- [52] Microsoft IIS Web Development Technologies, <http://msdn.microsoft.com/en-us/library/ms524911%28v=vs.90%29.aspx>
- [53] The Apache Software Foundation, <http://www.apache.org>
- [54] Apache Tomcat, <http://tomcat.apache.org>
- [55] Plataformas Java SE y EE, <http://www.oracle.com/technetwork/es/java/javase/overview/index.html>
<http://www.oracle.com/technetwork/es/java/javaee/overview/index.html>
- [56] Oracle WebLogic Suite, <http://www.oracle.com/us/products/middleware/application-server/weblogic-suite>
- [57] WebSphere software, IBM, <http://www.ibm.com/software/websphere>
- [58] JOnAS Open Source OSGi Enterprise Server, <http://jonas.ow2.org>
- [59] JBoss, <http://www.jboss.org>
- [60] Apache Geronimo, <http://geronimo.apache.org>
- [61] Jetty, Eclipse, <http://www.eclipse.org/jetty/>
- [62] Java Specification Requests JSR, <http://www.jcp.org/en/jsr/all>
- [63] Servlet API Overview, <http://www.codejava.net/java-ee/servlet/servlet-api-overview>
- [64] Oracle TopLink Developer's Guide, http://docs.oracle.com/cd/B25221_05/web.1013/b13593/undtldev010.htm
- [65] J.J. Garrett, *AJAX, A New Approach to Web Applications*, <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>
- [66] W3C, *Web Services Architecture*, <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis>
- [67] C.A. Morales, *Estado del Arte: Servicios Web*, Universidad Nacional de Colombia, 2010
- [68] Apache Struts, <http://struts.apache.org/>
- [69] Spring, <http://www.springsource.org>
- [70] Spring Framework: Definition, <http://blognextstar.wordpress.com/2013/02/08/spring-framework/>
- [71] Java Server Faces, <http://javaserverfaces.java.net>
- [72] Google Web Toolkit GWT, <https://developers.google.com/web-toolkit>
- [73] Vaadin, <https://vaadin.com/home>
- [74] H-M. Chen, Y-C. Lin, *Web-FEM: An internet-based finite-element analysis framework with 3D graphics and parallel computing environment*, *Advances in Engineering Software*, 39, 2008, pp. 55-68
- [75] J. Peng, H. K. H. Law, *Building finite element analysis programs in distributed services environment*. *Computers & Structures*, Vol 82-22, 2004, pp. 1813-1833
- [76] B. Eynard, S. Liénard, S. Charles, A. Odinet, *Web-based Collaborative*

- Engineering Support System: Applications in Mechanical Design and Structural Analysis*, Concurrent Engineering, Vol. 13-2, 2005, pp. 145-153
- [77] Thomas C. Hales, *The honeycomb conjecture*, Discrete and Computational Geometry 25, pp. 1–22, 2001
- [78] F.X. Unzaga, J.M. Vilchez, *Soluciones para vigas de atado en muelles de cajones flotantes*, Anales de Construcción y Materiales Avanzados, Vol.9, Master en Ingeniería Estructural y de la Construcción, UPC, 2009.
- [79] Teoría del Buque, Cesáreo Díaz Fernández, Barcelona 1972
- [80] Pierre Bouguer, *Traité du Navire*, 1746
- [81] *GiD The Personal Pre/Postprocesor*, versión 7, Internacional Center for Numerical Methods in Engineering – CIMNE, <http://www.gidhome.com>
- [82] L. Piegl, *On NURBS: a survey*. IEEE Computer Graphics and Applications, 11(1), 1991, pp. 55-71.
- [83] R. Ribó, G. Bugada, E. Oñate. *Some algorithms to correct a geometry in order to create a finite element mesh*. Computers & Structures, Vol. 80, 16-17, 2002, pp. 1399-1408
- [84] R. Ribó Rodríguez. *Desarrollo de un sistema integrado para tratamiento de geometría, generación de malla y datos para el análisis por el método de los elementos finitos*. PhD Thesis, UPC Barcelona, 2000
- [85] T. J. Baker, *Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation*. Engineering with Computers, Springer-Verlag, No. 5, 1989, pp. 161-175.
- [86] T.J. Baker, *Delaunay-Voronoi methods*, Handbook of Grid Generation, Chapter 16, 1999, J.F. Thompson, B.K. Soni and N.P. Weatherill (Eds.), CRC Press, Boca Raton, FL.
- [87] J. Peraire, J. Peiró, and K. Morgan, *Advancing Front Grid Generation*, Handbook of Grid Generation, Chapter 17, 1999, J.F. Thompson, B.K. Soni and N.P. Weatherill (Eds.), CRC Press, Boca Raton, FL.
- [88] J. H. Cheng, P. M. Finnigan, A. F. Hathaway, A. Kela, W. J. Schroeder, *Quadtree/octree meshing with adaptive analysis*. Numerical Grid Generation in Computational Fluid Dynamics, Pineridge Press, 1988, pp. 633-642.
- [89] S. E. Benzley, E. Perry, K. Merkley, B. Clark, and K. Sjaardema. *A comparison of all-hexahedral and all-tetrahedral finite element meshes for elastic and elasto-plastic analysis*. 4th Int. Meshing Roundtable (1995) 179-191
- [90] L. de la Cruz, *Metodologías de generación de mallas*, UNAM 2002
- [91] Fletcher, C.A.J., *Computational Techniques for Fluid Dynamics*, Vol II, Second Edition, Springer-Verlag, 1991
- [92] W.A. Cook, and W.R. Oakes. *Mapping Methods for Generating Three-Dimensional Meshes*, Computers in Mechanical Engineering, August 1982, pp. 67-72.
- [93] Zienkiewicz O.C., Phillips D.V., *An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by "Isoparametric" Co-ordinates*, International Journal for Numerical Methods in Engineering, vol. 3, 1971, pp. 519-528
- [94] E. Ruiz-Gironés, J. Sarrate, *Generation of structured hexahedral meshes in volumes with holes*, Finite Elements in Analysis and Design, Volume 46, Issue 10, October 2010, pp. 792-804.
- [95] Thompson, J.F. Warsi, Z.U.A., & Mastin, C.W., *Numerical Grid Generation, Foundations and Applications*, North-Holland, 1985
- [96] A. Gross, H. F. Fasel, *Multi-block Poisson grid generator for cascade simulations*, Mathematics and Computers in Simulation, Volume 79, Issue 3, 1

- 2008, pp 416-428.
- [97] Steinberg, S. & Roache, P.J., *Variational Grid Generation*, Num. Meth. for P.D.E.'s., 2, 1986, pp. 71-96.
- [98] Yerry M.A., Shephard M.S., "Automatic Three-Dimensional Mesh Generation by the Modified-Octree Technique", *International Journal for Numerical Methods in Engineering*, vol. 20, 1984, pp. 1965-1990
- [99] H. Zhang, G. Zhao, X. Ma, *Adaptive generation of hexahedral element mesh using an improved grid-based method*, *Computer-Aided Design*, Volume 39, Issue 10, 2007, pp 914-928.
- [100] M. Lai, S. Benzley and D. White, *Automated hexahedral mesh generation by generalized multiple source to multiple target sweeping*, *International Journal for Numerical Methods in Engineering*, John Wiley, Vol. 49, No. 1, 2000, pp. 261-275.
- [101] P.M. Knupp *Next-generation sweep tool: A method for generating all-hex meshes on two-and-one-half dimensional geometries*. In: *Proceedings 7th International Meshing Roundtable*. 1998. pp. 505–513.
- [102] Owen, Steven J., *A Survey of Unstructured Mesh Generation Technology*, *Proceedings 7th International Meshing Roundtable*, Dearborn, MI, October 1998.
- [103] S.B. Petersen, P.A.F. Martins, *Finite element remeshing: a metal forming approach for quadrilateral mesh generation and refinement*, *Int. J. Numer. Methods Eng.* 40, 1997, pp. 1449-1464
- [104] S.H. Lo, *Generating quadrilateral elements on plane and over curved surfaces*, *Computers & Structures*, Volume 31, Issue 3, 1989, pp. 421-426.
- [105] Peggy L. Baehmann, Scott L. Wittchen, Mark S. Shephard, Kurt R. Grice and Mark A. Yerry, *Robust Geometrically based, Automatic Two-Dimensional Mesh Generation*, *International Journal for Numerical Methods in Engineering*, Vol.24, 1987, pp.1043-1078
- [106] J.A. Talbert, and A.R. Parkinson, *Development of an Automatic, Two Dimensional Finite Element Mesh Generator using Quadrilateral Elements and Bezier Curve Boundary Definitions*, *International Journal for Numerical Methods in Engineering*, Vol.29, 1991, pp.1551-1567
- [107] A. C. Price, *MAT (Medial Axis Transform) and associated technologies for structured meshing*. *European Congress on Computational methods in Applied Science and Engineering*, 11-14 Sept. 2000
- [108] M.A. Price and C.G. Armstrong, *Hexahedral Mesh Generation by Medial Surface Subdivision: Part I*, *International Journal for Numerical Methods in Engineering*. Vol 38(19), 1995, pp.3335-3359
- [109] T. D. Blacker and M. B. Stephenson, *Paving: A new approach to automate quadrilateral mesh generation*. *International Journal for Numerical Methods in Engineering*, John Wiley & Sons, Vol. 32, 811-847, 1991
- [110] S. Mitchell. *A characterization of the quadrilateral meshes of a surface which admit a compatible hexahedral mesh of the enclosed volume*, 5th MSI Worksh. *Computational Geometry*, 1995
- [111] T. D. Blacker and R. J. Meyers, *Seams and Wedges in Plastering: A 3D Hexahedral Mesh Generation Algorithm*, *Engineering With Computers*, Springer-Verlag London Ltd., Vol. 9, pp. 83-93, 1993
- [112] T. J. Tautges, T. Blacker, and S. A. Mitchell, *The whisker weaving algorithm: A connectivity-based method for constructing all-hexahedral finite element meshes*, *Int. J. Numer. Methods in Eng.* 39 (1996), 3327-3349

- [113] N. A. Calvo and S. R. Idelsohn, *All-hexahedral element meshing by generating the dual mesh*, Computational Mechanics: New Trends and Applications (S. Idelsohn, E. Oñate, and E. Dvorkin, eds.), CIMNE, Barcelona, Spain, 1998
- [114] R. Schneiders, *A Grid-Based Algorithm for the Generation of Hexahedral Element Meshes*, Engineering With Computers. Vol.12, 1996, pp.168-177
- [115] A. Díaz Morcillo, *Método de mallado y algoritmos adaptativos de dos y tres dimensiones para la resolución de problemas electromagnéticos cerrados mediante el método de los elementos finitos*, Tesis Doctoral, Universidad Politécnica de Valencia, 2000
- [116] S.H. Huang, J. Shi, S. Maloo, E. Steinthorsson, *Automatic Generation of Quadrilateral Multi-Block Topology for FEA/CFD Applications*. IEEE ICIT'02, 2002, pp, 1300-1305
- [117] T. Schönfeld, P. Weinerfelt, C.B. Jenssen. *Algorithms For The Automatic Generation Of 2-D Structured Multi-Block Grids* NASA Conference Publication, 1995, pp. 561-561
- [118] I. Ergatoudis, B. M. Irons & O. C. Zienkiewicz, *Curved, isoparametric, "quadrilateral" elements for finite element analysis*. International Journal of Solids and Structures, 4(1), 1968, pp. 31-42.
- [119] Colecciones de Java
<http://docs.oracle.com/javase/7/docs/technotes/guides/collections/reference.html>
- [120] R. Ribó, *Descripción matemática de las NURBS*, 2003, UPC
- [121] A. R. Forrest, *On Coons and other methods for the representation of curved surfaces*. Computer Graphics and Image Processing, 1(4), 1972, pp. 341-359.
- [122] Formato de geometría de GiD
<http://listas.cimne.upc.edu/pipermail/gidlist/attachments/20040130/6117f3dd/attachment.pdf>
- [123] JavaBeans Specifications
<http://www.oracle.com/technetwork/java/javase/documentation/spec-136004.html>
- [124] Java Collections Framework
<http://docs.oracle.com/javase/6/docs/technotes/guides/collections/overview.html>
- [125] *ROM 0.5-05 Recomendación Geotécnica para las Obras Marítima y/o Portuaria*, Puertos del Estado, Ministerio de Fomento, 2005
- [126] *CALSEF[©] 2001*, Centro Internacional de Métodos Numéricos en Ingeniería,UPC, <http://www.cimne.com/calsef/frame.htm>
- [127] O.C. Zienkiewicz, R.L. Taylor, *El Método de los Elementos Finitos*, Volumen 1, CIMNE, 2004
- [128] M.D. García León, *Estrategias para la resolución de grandes sistemas de ecuaciones lineales. Métodos de Cuasi-Mínimo Residuo Modificados*, Tesis Doctoral, ULPGC, 2003
- [129] A N Krylov. *O chislennom reshenii uravnenia, kotorym v tekhnicheskikh system*. News of Academy of Sciences of the USSR, 1931, VII, Nr.4, pp. 491-539
- [130] B. F. Vajargah & M. Moradi, *Diagonal Scaling of Ill-Conditioned Matrixes by Genetic Algorithm*, Journal of Applied Mathematics, Statistics and Informatics, 8-1, 2012, pp. 49-53
- [131] J. R. Shewchuk, *An introduction to the conjugate gradient method without the agonizing pain*. School of Computer Science Carnegie Mellon University, 1994
- [132] *Librería MatMan*, Ramón Codina, UPC, <http://codina.rmee.upc.edu/>
- [133] C.A.R. Hoare, *Algorithm 63: partition*, Communications of the ACM, Vol 4, 7, 1961, p. 321

- [134] Y. SAAD. SPARSKIT: a basic tool-kit for sparse matrix computations (version 2), 1994, <http://www.cs.umn.edu/~saad>.
- [135] Y. SAAD, *ILUT: A dual Threshold Incomplete LU factorization*, Numerical Linear Algebra with Applications, Vol 1, 4, 1994, pp. 387-402
- [136] J.A. Garrido, J. Pereda, A. Foces y B. Suárez, *Una aplicación informática para el proyecto de cajones portuarios de hormigón armado a través de Internet*, Congreso de Métodos Numéricos en Ingeniería, 2005, pp. 170.



UVa

