

# Tabla de Contenido

<b>1</b>	<b>Introducción, planteamiento del problema y solución</b>	<b>1</b>
1.1	Introducción . . . . .	1
1.2	Planteamiento del problema . . . . .	3
1.3	Solución propuesta . . . . .	4
1.4	Objetivos de la Tesis . . . . .	5
1.5	Enfoque . . . . .	7
1.6	Impacto . . . . .	7
1.7	Producto . . . . .	8
<b>2</b>	<b>Marco Referencial</b>	<b>9</b>
2.1	Marco Conceptual . . . . .	9
2.1.1	Geometría Computacional . . . . .	9
2.1.2	Vacíos poligonales . . . . .	9
2.1.3	Triangulación Delaunay . . . . .	10
2.1.4	Grafo de adyacencia de triangulación de Delaunay . . . . .	11
2.1.5	Qhull . . . . .	12
2.1.6	Lenguaje de programación C++ . . . . .	12
2.1.7	Librería CGAL . . . . .	12
2.1.8	Conceptos ligados a la programación orientada a objetos . . . . .	13
2.2	Trabajos de investigación relacionados . . . . .	14
2.2.1	Algoritmo Delfin . . . . .	14
2.2.2	Algoritmo paralelo para vacíos poligonales en Triangulaciones de Delaunay . . . . .	16
2.2.3	Comparación de implementaciones para vacíos poligonales . . . . .	18
<b>3</b>	<b>Análisis de Requerimientos</b>	<b>20</b>
3.1	Metodología de trabajo . . . . .	20
3.1.1	Metodología de Investigación . . . . .	20
3.1.2	Metodología de Desarrollo . . . . .	20
3.2	Definición de Requerimientos . . . . .	22
3.2.1	Requerimientos Funcionales . . . . .	22
3.2.2	Requerimientos No Funcionales . . . . .	23
3.3	Usuario . . . . .	23
3.4	Funciones de la aplicación . . . . .	23
3.5	Casos de Uso . . . . .	23
3.5.1	Diagrama de Casos de Uso . . . . .	23
3.5.2	Caso de uso Encontrar Vacíos . . . . .	23
3.5.3	Caso de uso Visualizar Vacíos . . . . .	24
3.6	Diagrama de Secuencia . . . . .	24
3.7	Diagrama de Estados . . . . .	24
3.8	Modelo de dominio . . . . .	24

<b>4</b>	<b>Diseño</b>	<b>25</b>
4.1	Diseño del algoritmo . . . . .	25
4.2	Diseño de las clases básicas que requiere la implementación <b>Delfin ++</b> . . . . .	26
4.3	Diseño de la programación . . . . .	29
4.4	Diseño arquitectónico . . . . .	29
4.5	Diseño de la interfaz . . . . .	30
4.6	Diseño lógico . . . . .	31
4.6.1	Caso de uso reales . . . . .	31
4.6.2	Diagrama de colaboración . . . . .	31
4.6.3	Diagrama de clases . . . . .	31
4.7	Diseño de datos . . . . .	31
4.8	Diseño de pruebas . . . . .	34
4.8.1	Pruebas para el algoritmo diseñado . . . . .	34
4.8.2	Pruebas para la ejecución . . . . .	34
<b>5</b>	<b>Implementación</b>	<b>35</b>
5.1	Lenguajes de programación para la implementación . . . . .	35
5.1.1	Lenguaje C++ . . . . .	35
5.1.2	Lenguaje Python . . . . .	35
5.2	Herramienta de desarrollo . . . . .	36
5.2.1	Herramientas candidatas . . . . .	36
5.2.2	Herramienta seleccionada . . . . .	37
5.3	Hardware para evaluación y desarrollo . . . . .	38
5.4	Consideraciones de implementación . . . . .	38
<b>6</b>	<b>Pruebas</b>	<b>42</b>
6.1	Pruebas del algoritmo . . . . .	42
6.2	Pruebas de la ejecución . . . . .	49
6.3	Pruebas de comparación con <b>Delfin extendido</b> . . . . .	49
6.4	Pruebas para gráficos financieros . . . . .	57
<b>7</b>	<b>Conclusiones</b>	<b>59</b>
	<b>Bibliografía</b>	<b>61</b>
	<b>Anexos</b>	<b>63</b>
<b>A</b>	<b>Anexo A: Resultado pruebas para mayor volúmen de datos</b>	<b>64</b>
A.1	Prueba con 10000 puntos creados de la plantilla circular. . . . .	64
A.2	Prueba con 50000 puntos creados de la plantilla circular. . . . .	64
A.3	Prueba con 10000 puntos creados de la plantilla irregular. . . . .	65
A.4	Prueba con 100000 puntos creados de la plantilla irregular. . . . .	66
<b>B</b>	<b>Anexo B: Anexos del capítulo 3 analisis de requerimientos</b>	<b>68</b>
B.1	Diagrama de casos de uso . . . . .	68
B.2	Caso de uso Encontrar Vacíos . . . . .	68
B.3	Caso de uso Visualizar Vacíos . . . . .	70
B.4	Diagrama de secuencia del <b>Delfin ++</b> . . . . .	70
B.5	Diagrama de secuencia del <b>Delfin ++</b> . . . . .	71
B.6	Modelo de dominio del <b>Delfin ++</b> . . . . .	71

<b>C</b>	<b>Anexo C: Anexos del capítulo 4 diseño</b>	<b>72</b>
C.1	Caso de uso reales para el <b>Delfin ++</b> . . . . .	72
C.2	Caso de uso real ingresar parámetros y ejecutar implementación. . . . .	72
C.3	Caso de uso real visualizar resultados . . . . .	75
C.4	Diagrama de colaboración del <b>Delfin ++</b> . . . . .	76
C.5	Diagrama de clases del <b>Delfin ++</b> . . . . .	77
C.6	Pruebas del algoritmo diseñado . . . . .	77
C.7	Pruebas de ejecución del <b>Delfin ++</b> . . . . .	78
<b>D</b>	<b>Anexo D: Anexos del capítulo 5 implementación</b>	<b>80</b>
D.1	Código CGAL del <b>Delfin ++</b> donde se realiza la triangulación de Delaunay. . . . .	80
D.2	Código para clasificar a los vacíos por área. . . . .	81
D.3	Código para clasificar a los vacíos por arco más largo. . . . .	81
D.4	Código para unir dos vacíos en uno. . . . .	81
D.5	Código Python que lee un archivo de resultado con los puntos que conforman los polígonos que representan los espacios vacíos. . . . .	82
<b>E</b>	<b>Anexo E: Anexos del capítulo 6 pruebas</b>	<b>84</b>
E.1	Pruebas del algoritmo . . . . .	84