



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

SIMULACIÓN MEDIANTE DINÁMICA DE FLUIDOS COMPUTACIONAL  
(CFD) DEL EFLUENTE EN EL CLARIFICADOR SECUNDARIO DE LA  
PTAR UDLAPARK

AUTOR

Martín André Villavicencio Robalino

AÑO

2020



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

SIMULACIÓN MEDIANTE DINÁMICA DE FLUIDOS COMPUTACIONAL (CFD)  
DEL EFLUENTE EN EL CLARIFICADOR SECUNDARIO DE LA PTAR  
UDLAPARK

Trabajo de Titulación presentado en conformidad con los requisitos establecidos  
para optar por el título de Ingeniero Ambiental en Prevención y Remediación.

Profesor Guía

MSc. Santiago Daniel Piedra Burgos

Autor

Martín André Villavicencio Robalino

Año

2020

## DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido el trabajo, Simulación mediante dinámica de fluidos computacional (CFD) del efluente en el clarificador secundario de la PTAR UDLAPARK, a través de reuniones periódicas con el estudiante Martín André Villavicencio Robalino, en el semestre 2020-1, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.



---

Santiago Daniel Piedra Burgos

Máster en Hidrociencias e Ingeniería

C.I. 1715384150

## DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, Simulación mediante dinámica de fluidos computacional (CFD) del efluente en el clarificador secundario de la PTAR UDLAPARK, del estudiante Martín André Villavicencio Robalino, en el semestre 2020-1, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.



Daniel Hernán Hidalgo Villalba

Magister en Ciencias Especialización Ingeniería Ambiental

C.I. 1801914449

## DECLARACIÓN DE AUTORIA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

A handwritten signature in blue ink, written in a cursive style, positioned above a horizontal line.

---

Martín André Villavicencio Robalino

C.I. 172566010

## RESUMEN

Este trabajo se centra en la aplicación de la dinámica de fluidos computacional (CFD) para simular el comportamiento hidráulico del flujo en el clarificador secundario de la planta de tratamiento de aguas residuales (PTAR) de UDLAPARK por medio de la plataforma de código abierto de OpenFOAM, la cual incluye los métodos de cálculo numérico que resuelven iterativamente las ecuaciones de Navier-Stokes para predecir el movimiento del fluido.

En principio, se desarrolló el pre-procesamiento con la definición del dominio mediante la construcción de la geometría circular con sus respectivas dimensiones reales y la generación del mallado computacional en el software SALOME. A continuación, se ejecutó la simulación para el clarificador secundario donde se especificaron las regiones de interés del flujo y las condiciones iniciales del sistema referentes a los parámetros de velocidad y presión. Se busco alcanzar el estado estacionario del modelo hidráulico en un tiempo de simulación de 300 segundos, utilizando el solucionador para flujos incompresibles “interFoam”, con el modelo de turbulencia RANS (Reynolds-Averaged Navier-Stokes) y el algoritmo de solución “PIMPLE” que se caracteriza por resaltar el cálculo de la velocidad y presión con mayor estabilidad.

Finalmente se alcanzó la convergencia numérica a partir de los 225 segundos y se pudo llevar a cabo el post-procesamiento utilizando la herramienta visual ParaView en donde fue posible observar la proyección tridimensional del comportamiento hidráulico del clarificador secundario y se realizó el análisis del campo de flujo por medio de la aplicación visual del campo vectorial de la velocidad el cual determino la distribución y dirección del flujo.

Palabras clave: CFD, mallado computacional, solucionador, convergencia, campo vectorial, campo de flujo.

## ABSTRACT

This work focuses on the application of computational fluid dynamics (CFD) to simulate the hydraulic operation of flow in the secondary clarifier of the UDLAPARK wastewater treatment plant (WWTP) using the open source platform OpenFOAM, which includes numerical calculation methods that iteratively solve the Navier-Stokes equations to predict the motion of the fluid.

In principle, the pre-processing was developed with the definition of the domain through the construction of the circular geometry with its respective real dimensions and the generation of the computational mesh in the SALOME software. Then, the simulation for the secondary clarifier was executed where the flow regions of interest and the initial conditions of the system regarding the speed and pressure parameters were specified. The aim was to reach the stationary state of the hydraulic model in a simulation time of 300 seconds, using the solver for incompressible flows "interFoam", with the turbulence model RANS (Reynolds-Averaged Navier-Stokes) and the solution algorithm "PIMPLE" which is characterized by highlighting the calculation of speed and pressure with greater stability.

Finally, numerical convergence was reached after 225 seconds and post-processing could be carried out using the visual tool ParaView where it was possible to observe the three-dimensional projection of the hydraulic behavior of the secondary clarifier and the analysis of the flow field was carried out by the visual application of the vector field of the velocity which determined the distribution and direction of the flow.

Keywords: CFD, mesh, solver, numerical convergence, vector field, flow field.

# ÍNDICE

1. INTRODUCCIÓN.....	1
1.1. Antecedentes .....	1
1.2. Marco referencial.....	3
1.3. Objetivos .....	5
1.3.1. Objetivo general.....	5
1.3.2. Objetivos específicos .....	5
1.4. Alcance .....	5
1.5. Justificación .....	6
2. MARCO TEÓRICO .....	8
2.1. Planta de tratamiento de aguas residuales.....	8
2.2. Clarificador secundario .....	9
2.2.1. Decantador circular de flujo vertical .....	9
2.3. SALOME .....	9
2.3.1. NETGEN.....	10
2.3.2. Malla “Mesh” .....	10
2.4. Dinámica de fluido computacional “CFD” .....	10
2.4.1. Ecuaciones de Navier-Stokes .....	11
2.5. Método de volumen finito .....	12
2.6. OpenFOAM.....	13
2.6.1. Unidades y dimensiones.....	13

2.6.2. Solucionador “interFoam” .....	14
2.6.3. Algoritmo “PISO” .....	15
2.6.4. Algoritmo “SIMPLE” .....	15
2.6.5. Algoritmo de solución “PIMPLE” .....	16
2.6.6. Ecuaciones gobernantes .....	16
2.6.7. Modelo de turbulencia “RANS” ( <i>Reynolds Averaged Navier-Stokes</i> ) ...	18
2.6.7.1. Transporte de esfuerzo cortante “k-omegaSST” .....	19
2.6.8. Directorios requeridos.....	19
2.6.9. Archivos de “ <i>system</i> ” .....	20
2.6.9.1. controlDict.....	21
2.6.10. Condiciones de borde .....	23
2.6.10.1. Base .....	24
2.6.10.2. Primitivos .....	25
2.6.10.3. Derivados .....	25
2.7. Generación mallado computacional .....	26
2.7.1. blockMesh.....	27
2.7.2. surfaceFeatureExtract.....	27
2.7.3. snappyHexMesh .....	28
2.7.3.1. Flujo de trabajo.....	28
2.8. Simulación en paralelo.....	31
2.8.1. MPI .....	31
2.8.2. Descomposición del dominio .....	32
2.8.3. Reconstrucción del dominio.....	32
2.9. ParaView .....	33

3. METODOLOGÍA .....	34
3.1. Pre-Procesamiento .....	35
3.1.1. Geometría.....	35
3.1.2. Generación del mallado computacional .....	38
3.1.2.1. SALOME MESH .....	38
3.1.2.2. snappyHexMesh (OpenFOAM) .....	40
3.2. Procesamiento .....	41
3.2.1. Condiciones iniciales y de borde .....	42
3.2.2. Esquemas numéricos .....	45
3.2.3. Control y solución de algoritmos.....	48
3.2.4. Cómputo en paralelo.....	50
3.3. Post-Procesamiento .....	53
3.3.1. Visualización .....	53
4. RESULTADOS Y ANÁLISIS .....	54
4.1. Residuales de la simulación .....	54
4.2. Campo de flujo.....	55
4.3. Distribución del circuito de flujo .....	61
5. CONCLUSIONES Y RECOMENDACIONES.....	62
5.1 Conclusiones.....	62
5.2 Recomendaciones .....	64
REFERENCIAS .....	66

## ÍNDICE DE TABLAS

Tabla 1. <i>Simbología de las ecuaciones de Navier-Stokes</i> .....	12
Tabla 2. <i>Conjunto de unidades utilizadas en OpenFOAM</i> .....	14
Tabla 3. <i>Simbología de las ecuaciones gobernantes</i> .....	18
Tabla 4. <i>Definición del control de tiempo para simulación</i> .....	21
Tabla 5. <i>Definición de escritura para el registro del tiempo de simulación</i> .....	22
Tabla 6. <i>Lista de parches de límites geométricos</i> .....	24
Tabla 7. <i>Lista de parches de límites complejos</i> .....	24
Tabla 8. <i>Lista de condiciones de contorno de tipo primitivo</i> .....	25
Tabla 9. <i>Lista de condiciones de límites derivados del tipo primitivo</i> .....	25
Tabla 10. <i>Dimensionamiento de la geometría del clarificador secundario</i> .....	35
Tabla 11. <i>Información de malla computacional</i> .....	38
Tabla 12. <i>Condiciones de borde del clarificador secundario</i> .....	42
Tabla 13. <i>Esquema de discretización “fvSchemes” del clarificador secundario</i> .....	45
Tabla 14. <i>Esquema de solución “fvSolution” del clarificador secundario</i> .....	48
Tabla 15. <i>Configuración del algoritmo de solución “PIMPLE”</i> .....	50
Tabla 16. <i>Configuración de “controlDict” para tiempo y escritura de simulación</i> .....	52
Tabla 17. <i>Tramos del circuito hidráulico</i> .....	62

# ÍNDICE DE FIGURAS

<i>Figura 1.</i> Bloque genérico de malla con vértices enumerados.....	27
<i>Figura 2.</i> Malla esquemática para snappyHexMesh.....	29
<i>Figura 3.</i> División y eliminación de celdas por región en el proceso de mallado por snappyHexMesh.....	29
<i>Figura 4.</i> Ajuste de la superficie en el proceso de mallado por snappyHexMesh..	30
<i>Figura 5.</i> Adición de capa en el proceso de mallado por snappyHexMesh. ....	30
<i>Figura 6.</i> Esquema metodológico para simulación CFD. ....	34
<i>Figura 7.</i> Estructura decantadora de sólidos del clarificador secundario. ....	36
<i>Figura 8.</i> Tanque del clarificador secundario.....	36
<i>Figura 9.</i> Estructura rompe presión para la entrada del afluente al clarificador secundario.....	37
<i>Figura 10.</i> Rebosadero para salida de agua del clarificador secundario.....	37
<i>Figura 11.</i> Tubería de salida para el efluente del clarificador secundario. ....	37
<i>Figura 12.</i> Geometría construida del clarificador secundario. ....	38
<i>Figura 13.</i> Malla de la geometría de inlet. ....	39
<i>Figura 14.</i> Malla de la geometría de outlet. ....	39
<i>Figura 15.</i> Malla de la geometría de atmosphere. ....	40
<i>Figura 16.</i> Malla de la geometría de walls. ....	40
<i>Figura 17.</i> Mallado final del clarificador secundario.....	41
<i>Figura 18.</i> Vista superior del contorno de nivel de agua en el clarificador secundario .....	51
<i>Figura 19.</i> Vista lateral del contorno de nivel de agua en el clarificador secundario .....	52
<i>Figura 20.</i> Monitoreo de residuales de las velocidades en los componentes (x,y,z). .....	54

<i>Figura 21.</i> Rango de velocidades del agua en el eje “z” del clarificador secundario .....	55
<i>Figura 22.</i> Distribución de la velocidad en las paredes del clarificador secundario en ParaView .....	56
<i>Figura 23.</i> Campo de flujo por velocidades “z” del clarificador secundario en ParaView .....	57
<i>Figura 24.</i> Visualización del campo vectorial del clarificador secundario con isolíneas de flujo en ParaView. ....	58
<i>Figura 25.</i> Visualización del campo vectorial lateral del clarificador secundario con isolíneas de flujo en ParaView. ....	59
<i>Figura 26.</i> Visualización de dirección y distribución de velocidades del clarificador secundario en ParaView. ....	60
<i>Figura 27.</i> Esquema del circuito de flujo y zonas muertas del clarificador secundario. ....	61

## **1. INTRODUCCIÓN**

### **1.1. Antecedentes**

Por generaciones el desarrollo y crecimiento del ser humano y sus civilizaciones se han encontrado ligadas a los residuos líquidos producidos, generando poco a poco la contaminación de cuerpos hídricos. A través del tiempo los cuerpos hídricos han sufrido cargas más elevadas de contaminación de carácter industrial y doméstico, a causa de que las fuentes hídricas son utilizadas para drenar todas las descargas líquidas del ser humano (Burroughs & Burroughs, 2011).

La contaminación del agua ha provocado una serie de problemas ambientales y sociales en todo el mundo, razón por la que el hombre se ha encontrado en la necesidad de crear, diseñar y mejorar a su paso técnicas de depuración y tratamientos que sean más eficientes, para lograr mitigar y controlar los posibles impactos ambientales que generan las descargas de aguas residuales, los cuales hoy en día presentan una mayor complejidad debido a la creación de nuevas sustancias contaminantes (Raghuvanshi, Bhakar, Sowmya, & Sangwan, 2017).

Uno de los enfoques principales que presenta el tratamiento de aguas residuales es velar por la calidad de vida con respecto a la salud de la población. El incremento poblacional ha sido el motivo por el cual se potenció el desarrollo de sistemas de tratamiento de aguas residuales con la finalidad de reducir al máximo los impactos ambientales, tomando en cuenta los riesgos para la salud de la población (Lozano, 2012).

La aplicación de ingeniería que el ser humano ha realizado para la construcción y desarrollo de las plantas de tratamiento de aguas residuales (PTAR's) constan de varias operaciones unitarias estructuradas con el fin de transformar y reducir la carga orgánica presente en aguas residuales, por medio de procesos biológicos y físico-químicos basados en una serie de niveles que, de manera ordenada, empiezan por el pretratamiento, tratamiento primario, tratamiento secundario y terciario (Menéndez Gutiérrez & Garcia Zumalacarregui, 2013), dependiendo de la necesidad para la remoción de los agentes contaminantes presentes en el agua residual (Metcalf & Eddy, 2014). El grado de tratamiento deseado varía dependiendo de las propiedades del cuerpo de agua receptor para realizar una auto purificación de manera natural. Dicha capacidad para lograr la auto purificación depende especialmente de su propio caudal, el contenido que posee de oxígeno y su capacidad para reoxigenar (Makisha & Kulakov, 2017). Por lo tanto, el tratamiento de las aguas residuales en las urbes se basa en producir un efluente reusable para el ambiente y por otro lado un fango que puede ser reutilizado dependiendo de su disposición final (WEF, 2010).

En los primeros procesos de ingeniería para el diseño y construcción de los distintos sistemas de tratamiento de aguas residuales durante el siglo XX se presenciaban métodos de cálculo hidráulico simples, en donde solo se apreciaba el movimiento del fluido, dejando de lado factores importantes tales como las condiciones de borde del sistema que garantizarían la eficiencia a largo plazo (Findikakis, 2018). Sin embargo, el transcurso del tiempo generó una demanda más estricta hacia la optimización del diseño y construcción de estructuras hidráulicas para el tratamiento del agua (Findikakis, 2016). En consecuencia, el hombre se encaminó hacia el componente del modelamiento matemático con la finalidad de definir con mayor exactitud las condiciones a las que se encuentra enfrentado el flujo a tratar en cada uno de los procesos unitarios del sistema (Weiss, Plósz, Essemiani, & Meinhold, 2007), de tal manera que el producto final se vería afectado positivamente y

mejorado en el aspecto de tratamiento de efluentes (Ting, Miedema, & Xiuhan, 2019).

Posteriormente el ser humano dio un gran salto en el avance tecnológico computacional como consecuencia del desarrollo urbanístico e industrial que conllevaron a la creación de nuevos sistemas innovadores, nuevas estructuras hidráulicas y nuevas demandas de optimización para el diseño de plantas de tratamiento de aguas residuales (De Clercq & Vanrolleghem, 2002). Este avance ha dado como resultado la aplicación de la modelación matemática por CFD gracias a la accesibilidad de computadores, que contribuyen a resolver ciertos problemas de mecánica de fluidos (Jeppsson & Diehl, 1996), de tal manera que la hidráulica experimental posee una amplia gama de aplicación en donde los resultados se encuentran sujetos a efectos de escala (Samstag et al., 2016).

La simulación de fluidos posee la característica de realizar una serie de cálculos complejos que ahorran experimentaciones de alto costo con modelos a escala, no obstante, la validación de los resultados numéricos se complementan con un análisis experimental del modelo final comparando las soluciones para reducir el número de experimentos (Liscia, López, Ruiz Diaz Cordoso, & De Dios, 2018), por consiguiente el tan solo modificar los datos de entrada o el diseño del modelo, admite la obtención de una multitud de configuraciones reduciendo elocuentemente el tiempo de estudio (Fonseca, Marques, Azevedo, & Rodrigues, 2014).

## **1.2. Marco referencial**

A nivel país tan solo se trata el 29.3% de aguas residuales que ingresan a las redes de alcantarillado, valor del cual tan solo se encuentran destinadas en un 43.8%

hacia ríos, 30.3% hacia quebradas y el 25.9% distribuidos entre descargas al suelo y otros (INEC, 2016a). Esto quiere decir que la disposición final del agua residual no tratada representa un gran riesgo ambiental para la capacidad de carga de ríos y quebradas. El agua residual tratada se distribuye en las 421 plantas de tratamiento operadas por los 133 GAD Municipales de todo el Ecuador, teniendo que alrededor del 50% de las plantas se encuentran en la región sierra, sin embargo, la cantidad no es el único déficit presente en el tratamiento de aguas residuales sino también la eficiencia de rendimiento de dichos sistemas de tratamiento (INEC, 2016b).

El precario rendimiento de eficiencia puede deberse a un diseño incorrecto o a una mala operación de la planta, no obstante, en la realidad nacional los principales factores apuntan hacia la carencia de información y precaria normativa correspondiente para el diseño, que no se encuentra adecuada con las condiciones ambientales del país. Por lo general el déficit de eficiencia se debe a que el 50% de las plantas destinadas para tratamiento de aguas residuales son prediseñadas para ser multipropósito, en donde no presentan un rango específico de operación, puesto que su diseño no ha tomado en cuenta las características específicas del agua residual a tratar (Metcalf & Eddy, 2014), de tal manera que realizan sus descargas con una alta cantidad de carga orgánica lo que perjudica no solo al cuerpo receptor de descargas sino también al funcionamiento de la planta, provocando que la eficiencia de los demás tratamientos se reduzcan (Findikakis, 2018).

El ineficaz manejo de una planta de tratamiento de aguas residuales genera un malgasto de recursos económicos y operacionales, aclarando que no genera ningún efecto positivo al medio ambiente. Por lo tanto, la aplicación de una simulación CFD del diseño de un sistema de tratamiento de aguas previo a su construcción, es una manera de comprobar el correcto funcionamiento del sistema (Ting et al., 2019), y también una herramienta para lograr identificar y corregir cualquier falla de diseño

con el fin de ahorrar recursos y mejorar la eficiencia de tratamiento (Jeppsson & Diehl, 1996).

### **1.3. Objetivos**

#### **1.3.1. Objetivo general**

- Simular el funcionamiento hidráulico del clarificador secundario de la planta de tratamiento de aguas residuales de UDLAPARK.

#### **1.3.2. Objetivos específicos**

- Establecer las condiciones de borde del clarificador secundario de la PTAR UDLAPARK
- Estimar el tiempo de residencia real del clarificador secundario de la PTAR UDLAPARK

### **1.4. Alcance**

El presente estudio pretende modelar el trayecto hidráulico del agua residual de una PTAR a través del clarificador secundario. En la fase inicial del estudio se programa la descripción total de las condiciones del sistema a las que se verá enfrentado el fluido y su comportamiento considerando las condiciones de borde del clarificador de la PTAR y su caudal de diseño. Subsiguientemente se ejecutará la simulación

CFD para lograr la interpretación de los datos del flujo y estimación del tiempo de retención hidráulico.

### **1.5. Justificación**

En las últimas tres décadas se ha desarrollado el campo de estudio de la mecánica de fluidos para optimizar el diseño de plantas de aguas residuales (Gomez, 2018). En un principio se aplicaban metodologías simples de análisis matemáticos y experimentaciones a escala piloto, que no lograban garantizar con exactitud la eficiencia del sistema, puesto que no contemplaban la identificación de potenciales errores y fallas en un sistema a escala real (Ferziger & Peric, 2002).

Por otra parte, cabe recalcar que la ejecución de una experimentación es indispensable y obligatoria por ser la única vía metodológica para determinar un error o un acierto para las condiciones del sistema. Sin embargo, el papel que juegan los factores económicos y temporales presentan un gran peso y riesgo que es indispensable para llevar a cabo cualquier proyecto (Findikakis, 2018). Esto se debe a la magnitud de implementación de recursos, los cuales han provocado una evolución en el campo de la ingeniería al desarrollar la experimentación simulada por computador con el objetivo de disminuir dicha demanda económica y temporal. Consecuentemente, el desarrollo de herramientas computacionales para realizar modelaciones y simulaciones hidráulicas ha generado la solución para la optimización de recursos, puesto que garantiza una gran calidad con respecto al funcionamiento de cada operación unitaria del sistema y además su implementación resguarda la inadecuada implementación de recursos y capital (Sadrehaghighi, 2019).

La simulación “CFD” posee entre sus cualidades la implementación a bajo costo, mayor seguridad, menor consumo de recursos y presenta la posibilidad de realizar repeticiones indefinidas en un corto plazo a comparación de las experimentaciones tradicionales o físicas (Ferziger & Peric, 2002). En efecto, las experimentaciones físicas demandan una mayor inversión de dinero y se encuentran sujetas a la disponibilidad de tiempo y magnitud de recursos. Por lo que el desarrollo de pruebas experimentales físicas puede presentar un mayor riesgo sin resultados asegurados.

Para el desarrollo del diseño de una planta de aguas residuales se debe tener claro que todas las operaciones unitarias se rigen por parámetros específicos dependiendo su función (Das et al., 2016). En cuanto al enfoque del tratamiento secundario del sistema, se debe especificar que tal fase de operación es la responsable de realizar la principal descontaminación del efluente. En consecuencia, tanto el reactor, como el clarificador del tratamiento secundario cumplen un rol muy importante para garantizar un afluente descontaminado y clarificado. Por esta razón es necesario hacer estudios de optimización de la eficiencia del sistema, con el fin de provocar que el impacto ambiental originado por las descargas de efluentes en los cuerpos de agua se reduzca significativamente (Do-Quang, Cockx, Liné, & Roustan, 1998).

En contraste se puede inducir que la simulación CFD involucra un menor esfuerzo económico con mejores resultados, dado que su aplicación figura una potencial reducción de recursos e impactos que afectan al medio ambiente (Novak, Guinot, Jeffrey, & Reeve, 2018). En síntesis, el modelamiento hidráulico del clarificador secundario proveerá de información base, que será esencial para determinar si su deficiente funcionamiento se encuentra relacionado con la hidráulica del sistema.

## **2. MARCO TEÓRICO**

### **2.1. Planta de tratamiento de aguas residuales**

Los sistemas de tratamiento de aguas residuales comprenden una serie de niveles operacionales que constituyen procesos de carácter físico, químico y biológico, que se rigen por los fenómenos de movimiento y transporte hidráulico. Los procesos que pretende establecer el sistema se categorizan por etapas de pretratamiento, tratamiento primario, secundario y terciario.

En el tratamiento preliminar se realizan procesos físicos de separación o eliminación de elementos sólidos que pueden causar daños en los equipos de los posteriores sistemas debido a su gran tamaño. Consecuentemente el tratamiento primario busca remover aquellas fracciones de los sólidos en suspensión y sedimentables que el tratamiento preliminar no pudo eliminar, además dicha fase es capaz de realizar el tratamiento por métodos físicos o químicos (Menéndez Gutiérrez & Garcia Zumalacarregui, 2013)

En la etapa del tratamiento secundario se presenta la acción de microorganismos que transforman la materia orgánica compleja en materia estable y disminuyen la demanda de oxígeno, lo que genera masas de lodo que se conducen hacia un decantador secundario el cual realiza una separación adecuada para obtener un efluente clarificado. Finalmente se puede incorporar un último nivel de tratamiento catalogado como terciario el cual se caracteriza por su potencial de alta desinfección para aquellos remanentes orgánicos y sólidos suspendidos, principalmente por métodos de cloración y desinfección UV (Lozano, 2012).

## **2.2. Clarificador secundario**

Operación unitaria caracterizada por ser un proceso físico que cumple la función de realizar una separación de la materia sólida del medio líquido, en donde se presenta una diferencia de densidades y masas entre el agua y los elementos en suspensión, lo que termina complementando el desempeño del reactor secundario para producir un efluente que no presente elevadas concentraciones de sólidos en suspensión (Amy et al., 2017).

### **2.2.1. Decantador circular de flujo vertical**

Estructura definida para emplear procesos netamente físicos que permiten la sedimentación de las partículas por gravedad. Su configuración establece el ingreso de agua por medio de una campana deflectora que se localiza en el centro de la estructura, de tal manera que obliga al agua a introducirse por la parte inferior del tanque y su característica principal es que realiza una atenuación de la energía del flujo eliminando las turbulencias, ya que rompe la velocidad del agua. La disposición de la salida del agua presenta un rebosadero dentado como canal de recolección periférica para asegurar que el desagüe del efluente sea homogéneo (Lozano, 2012).

## **2.3. SALOME**

Es un software de código abierto que comprende una plataforma de carácter genérico enfocado en el preprocesamiento CFD para realizar una simulación numérica. Se fundamenta por presentar una arquitectura flexible y abierta para

poder formar geometrías. SALOME también posee la capacidad de realizar módulos de post procesamiento (Holzinger, 2020).

### **2.3.1. NETGEN**

Es un complemento de SALOME que funciona como un generador automático de mallas para entidades geométricas en 1D, 2D y 3D. Realiza una división para las caras de la geometría en elementos triangulares y una división para los sólidos en elementos tetraédricos (Schöberl, Egger, Becirovic, & Huber, 2007).

### **2.3.2. Malla “Mesh”**

La geometría se sometió a un proceso de mallado como parte integral para la solución numérica la cual debe cumplir una serie de criterios con la finalidad de garantizar una solución válida y precisa. En principio la malla se define automáticamente por celdas poliédricas en tres dimensiones y de carácter arbitrario, que se encuentran determinadas por caras poligonales, es decir, dichas celdas pueden presentar una cantidad unitaria ilimitada de caras, donde no existe un límite en la cantidad de bordes ni restricción alguna en su formación correspondiente para cada cara que se haya creado (Greenshields, 2019).

## **2.4. Dinámica de fluido computacional “CFD”**

Pertenece a una ramificación asociada a la mecánica de fluidos en donde se aplican análisis numéricos y algoritmos con el fin de analizar el comportamiento hidráulico,

siendo las computadoras la herramienta principal que lleva a cabo los cálculos necesarios para simular la interacción entre líquidos y gases con relación a superficies definidas por condiciones de borde. La CFD trabaja con tres ecuaciones generales de donde parte la resolución del comportamiento de los fluidos y hacen referencia a la ecuación de continuidad, de la cantidad de movimiento y de la conservación de la energía (Greenshields, 2019).

El procedimiento realiza una discretización de una zona o región en un espacio determinado, creando un mallado computacional de la geometría con la que se requiere trabajar, de tal manera que se generan ciertos volúmenes de control y se dispone a resolver cada volumen de control por medio de las ecuaciones referentes a la conservación que se discretizan de manera iterativa hasta obtener residuales cada vez más pequeños y estables (Sadrehaghighi, 2019).

#### **2.4.1. Ecuaciones de Navier-Stokes**

Las ecuaciones de Navier-Stokes son las ecuaciones gobernantes de la dinámica de fluidos computacional. Dichas ecuaciones describen como la temperatura, velocidad, presión y densidad de un fluido en movimiento se encuentran relacionadas. Las ecuaciones son una extensión de las ecuaciones de Euler, sin embargo, incluyen los efectos de viscosidad en el flujo (Holzinger, 2020).

- Ecuación de continuidad de flujo incompresible

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

- Ecuación de Navier-Stokes de flujo incompresible para componente x

$$\rho * \left( \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = - \frac{\partial P}{\partial x} + \rho g_x + \mu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right)$$

- Ecuación de Navier-Stokes de flujo incompresible para componente y

$$\rho * \left( \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = - \frac{\partial P}{\partial y} + \rho g_x + \mu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right)$$

- Ecuación de Navier-Stokes de flujo incompresible para componente z

$$\rho * \left( \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = - \frac{\partial P}{\partial z} + \rho g_x + \mu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right)$$

(Jayanti, 2018)

Siendo:

Tabla 1

*Simbología de las ecuaciones de Navier-Stokes.*

$v$	velocidad	$g$	gravedad
$u$	vector de velocidad en dirección x	$P$	presión
$v$	vector de velocidad en dirección y	$\rho$	densidad
$w$	vector de velocidad en dirección z	$\mu$	tensor de estrés
$t$	tiempo		

## 2.5. Método de volumen finito

Se presenta como una técnica numérica la cual se encarga de transformar las ecuaciones diferenciales parciales que representan las leyes de conservación sobre los volúmenes diferenciales en ecuaciones algebraicas discretas sobre los

volúmenes finitos, elementos o celdas. El primer paso en cuanto al proceso de solución es la discretización del dominio geométrico y cabe recalcar que se discretiza en elementos no superpuestos. El sistema de ecuaciones algebraicas realiza una resolución para llegar al cálculo de los valores de la variable dependiente para todos los elementos. Además, en dicho método se caracteriza su estricta conservación, propiedad inherente que ofrece resultados preferibles porque puede formularse en el espacio físico para mallas poligonales no estructuradas (Moukalled, Mangani, & Darwish, 2016).

## **2.6. OpenFOAM**

Es un software CFD que se encuentra en código abierto que se especializa en una amplia gama de atributos que busca resolver problemas tales como el flujo de fluidos con interacciones químicas, turbulencias y transferencias de calor, también temas de acústica, mecánica de sólidos y electromagnetismo. De tal manera que OpenFOAM es una terminal que desarrolla ejecutables de aplicaciones que se caracterizan por su funcionalidad de empaquetamiento hallada en una colección que alberga alrededor de 100 bibliotecas (Haider, 2018).

Las aplicaciones que comprende OpenFOAM se dividen por dos categorías, la primera hace referencia a los solucionadores, que están diseñados para resolver problemas concretos en relación con la mecánica de fluidos, y en segundo lugar se presentan las utilidades que se han diseñado con la finalidad de realizar tareas que involucran toda la transferencia y manipulación de datos (Greenshields, 2019).

### **2.6.1. Unidades y dimensiones**

En la terminal de OpenFOAM las unidades físicas hacen referencia a las dimensiones y se encuentran configuradas dentro de la clase “dimensionSet”. Por otro lado, las cantidades escalares o vectoriales se tratan de forma implícita por los tipos de datos (Holzinger, 2020).

Tabla 2

*Conjunto de unidades utilizadas en OpenFOAM.*

<b>dimensionSet</b>		
<b>Propiedad</b>	<b>Unidad</b>	<b>Símbolo</b>
Masa	Kilogramos	kg
Longitud	Metro	m
Tiempo	Segundo	s
Temperatura	Kelvin	K
Moles	Mol	Mol
Corriente	Amperio	A
Intensidad lumínica	Candela	Cd

Adaptado de (Holzinger, 2020).

### **2.6.2. Solucionador “interFoam”**

Solucionador de dos fluidos inmiscibles isotérmicos incompresibles que se caracteriza por presentar un enfoque hacia el volumen modificado del fluido, en donde se puede apreciar la incorporación de un término de flujo de compresión interfacial con el propósito de mitigar los efectos relacionados con las manchas numéricas que presenta la interfaz (Haddadi et al., 2018). El solucionador se encuentra dentro de las bibliotecas y utilidades de C ++ pertenecientes a

OpenFOAM, y su ámbito de aplicación se encuentra relacionado con la investigación de flujo multifásico, por lo tanto, el solucionador está diseñado para realizar una discretización de volúmenes finitos correspondientes a un conjunto genérico de ecuaciones diferenciales parciales (Moukalled et al., 2016). Además, presenta una serie de características muy útiles para la simulación como la facilidad de paralelización, la disponibilidad y accesibilidad de las utilidades previas y posteriores en cuanto al procesamiento, la oportunidad de elegir esquemas de discretización temporal y espacial, y el monitoreo de residuales (Jayanti, 2018).

### **2.6.3. Algoritmo “PISO”**

Es un algoritmo de presión implícita que corresponde a un operador dividido enfocado para casos que presentan inestabilidad, el cual propone una resolución sin iteraciones y sus pasos de tiempo se caracterizan por ser grandes, por ende, un esfuerzo de cálculo menor. Para la dinámica de fluidos computacional se aplica con el fin de resolver las ecuaciones de Navier-Stokes, y su procedimiento se basa en el cálculo de presión de velocidad para dichas ecuaciones con el propósito de solventar las problemáticas de un flujo compresible e inestable, sin embargo, también se adapta para problemáticas en estado estacionario. El algoritmo PISO se caracteriza por implicar un paso de tiempo predictor y dos pasos correctores que esperan cumplir y satisfacer la conservación de la masa aplicando los pasos referentes a la predicción y la corrección (Venier, Pairetti, Damian, & Nigro, 2017).

### **2.6.4. Algoritmo “SIMPLE”**

Este algoritmo utiliza una metodología semi implícita para las ecuaciones que se encuentran asociadas a la presión y es únicamente aplicado para estudios que se

encuentran en estado estacionario. El procedimiento numérico aplica para resolver de igual manera las ecuaciones de Navier-Stokes con el fin de solventar las problemáticas de flujo de fluidos y también de la transferencia de calor que se puede presentar, en esta caso el algoritmo presenta una metodología iterativa en donde primero establece las condiciones de contorno para proceder a realizar los cálculos de los gradientes de velocidad y presión, por lo que se dispone a resolver la ecuación de momento ya discretizada para computar el campo vectorial de la velocidad intermedia, además, realiza los cálculos asociados a los flujos de masa y presión no corregidos para actualizar dichos campos para finalmente llegar a una corrección de tales variables, incluyendo la velocidad en cada parte del volumen (Xiang, Yu, Yuan, & Sun, 2017).

#### **2.6.5. Algoritmo de solución “PIMPLE”**

El algoritmo de solución PIMPLE se encuentra conformado por la combinación de PISO y SIMPLE, el cual se resuelve de manera iterativa y se aplica al caso de estudio transitorio. En principio se debe apreciar el funcionamiento de tal algoritmo como la metodología aplicada al SIMPLE en cuanto a los pasos de tiempo, en donde se tendrá a los correctores externos como las iteraciones, de tal manera que cuando llegue a la convergencia numérica podrá continuar al siguiente paso de tiempo hasta que se finalice la solución. Es por tal razón que el algoritmo PIMPLE presenta una mejor estabilidad, sobre todo cuando nos encontramos frente a pasos de tiempo largos donde el comportamiento de la solución demuestra una inherente inestabilidad (Ferziger & Peric, 2002).

#### **2.6.6. Ecuaciones gobernantes**

El comportamiento de los fluidos se rige por las siguientes tres ecuaciones fundamentales:

- Ecuación de continuidad

$$\frac{\partial \rho}{\partial t} + \rho \frac{\partial u_i}{\partial x_i} = 0$$

Se asume que para fluidos incompresibles la densidad es constante.

(Holzinger, 2020)

- Ecuación de momentum

$$\frac{\partial(\rho u_j)}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i u_i) = -\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} (\tau_{ij} + \tau_{tij}) + \rho g_j + f \sigma_i$$

( I )      ( II )      ( III )      ( IV )      ( V )      ( VI )

(Holzinger, 2020)

Donde:

I: Cambio local con el tiempo

II: Convección de momento

III: Fuerza superficial

IV: Intercambio de momento dependiente molecular (difusión)

V: Fuerza de masa

VI: Tensión superficial

- Ecuación para la interfase

$$\frac{\partial \alpha}{\partial t} + \frac{\partial(\alpha u_j)}{\partial x_j} = 0$$

(Holzinger, 2020)

Siendo:

Tabla 3

*Simbología de las ecuaciones gobernantes.*

$\mathbf{u}$	velocidad	$g$	gravedad
$\boldsymbol{\tau}_{ij}$	tensores de viscosidad y turbulencia	$P$	presión
$\sigma$	tensión superficial constante	$\rho$	densidad
$f\sigma_i$	tensión superficial	$\mu$	tensor de estrés
$t$	tiempo	$\alpha$	(fluido 1, densidad 1); 0 (fluido 2, densidad 2)

Los tensores de viscosidad se aplican para modelar la parte de la tensión en un punto dentro de algún material que se puede atribuir a la velocidad de deformación (Bergadà Grañó, 2012).

### 2.6.7. Modelo de turbulencia “RANS” (*Reynolds Averaged Navier-Stokes*)

El modelo se basa en descomponer las variables de interés como la presión y velocidad, en un valor fluctuante y otro intermedio. El uso del promedio de Reynolds

dependerá implícitamente de la estabilidad que posean los valores promedio. Dicho modelo realiza una abismal simplificación de datos en donde es posible que mucha información que se encuentren originalmente dentro de las ecuaciones de Navier-Stokes se pierda (Robertson, Choudhury, Bhushan, & Walters, 2015).

Las ecuaciones de Navier-Stokes se someten a una reformulación en función de aquellas variables promedio y se genera un nuevo conjunto de ecuaciones, que se catalogan como las ecuaciones promedio de Reynolds y se diferencian de las originales por comprender una variable adicional, motivo por el cual requiere de un modelo adicional para poder cerrar el modelo RANS (Richmond, 2019). El modelo presenta una menor demanda computacional para realizar los cálculos, puesto que la resolución de las ecuaciones originales de Navier-Stokes no se realiza de manera directa (Moukalled et al., 2016).

#### **2.6.7.1. Transporte de esfuerzo cortante “k-omegaSST”**

Corresponde al modelo adicional que requiere el promedio de Reynolds. Es un modelo conformado por dos ecuaciones para la energía cinética de la turbulencia y la tasa de disipación específica de la turbulencia. Como objetivo requiere superar aquellas deficiencias que presenta su modelo estándar de “k-omega”, en cuanto a la dependencia de los valores de flujo libre que presenta omega y la cinética de la turbulencia, además es capaz de realizar la captura de la separación del flujo y dicho modelo presenta una gran capacidad de ejecución para simulaciones que involucren flujos curvos y rotación (Holzinger, 2020).

#### **2.6.8. Directorios requeridos**

El caso de estudio presenta un conjunto mínimo de archivos y directorios con los que se operara la simulación hidráulica, todo empieza desde un directorio raíz que se encuentra conformado por tres subdirectorios que corresponde a la carpeta “0”, “*constant*” y “*system*” (Moukalled et al., 2016).

El primer directorio se caracteriza por contener las condiciones iniciales y los límites en relación con todas las cantidades variables, en donde se calibran las velocidades y presiones con respecto al fluido. Con respecto a la carpeta “*constant*” se especifica que alberga todos aquellos archivos relacionados con cantidades constantes, como la información del mallado de la geometría que se encuentra dentro del subdirectorio “*polymesh*”, pero también se estipulan las propiedades y parámetros que establecerán la conducta del fluido. Por último se localiza la carpeta “*system*” en donde se hallan todos los archivos que manipularan y controlaran el solucionador y otras herramientas dinámicas para la simulación (Holzinger, 2020).

### **2.6.9. Archivos de “*system*”**

Existen tres archivos clave dentro del directorio que tienen la capacidad de controlar el solucionador para la simulación y que se puedan realizar los cálculos.

- *controlDict*: Contiene los controles que están relacionados con el tiempo de la simulación en donde se ven involucrados los pasos de tiempos, el intervalo de salida, entre otros, y comprende también la programación del tiempo de registro de la información con la que correrá la simulación (Greenshields, 2019).
- *fvSchemes*: En tal archivo se delimitan los esquemas para la discretización del volumen finito (Greenshields, 2019).

- *fvSolution*: El archivo aloja los controles relacionados con el solucionador, los algoritmos para la resolución y los límites de tolerancia del caso (Holzinger, 2020).

También existen archivos adicionales que contienen funciones para complementar la simulación con mayor rapidez y exactitud hacia los resultados, para el caso de estudio se presenta el “decomposeParDict” que es utilizado para establecer el número de subdominios con los que la computadora va a trabajar la simulación con el fin de correr en paralelo, por el otro lado, se presenta la archivo “setFieldsDict” la cual es una herramienta necesaria para establecer las cantidades de campo, lo que quiere decir que se determina a que nivel de agua se encuentra nuestra geometría por medio de las coordenadas (Greenshields, 2019).

#### 2.6.9.1. controlDict

Tabla 4

*Definición del control de tiempo para simulación.*

<b>startFrom</b>	Tiempo de inicio para la simulación	<b>firstTime</b>	Simulación empieza desde el primer paso de tiempo
		<b>startTime</b>	Simulación empieza desde un tiempo descrito
		<b>latestTime</b>	Simulación empieza desde el último paso de tiempo
<b>stopAt</b>	Determina el final de la simulación	<b>endTime</b>	Simulación se paraliza cuando se alcanza un tiempo específico

	<b>writeNow</b>	Simulación se paraliza posterior al completarse el paso de tiempo actual y se guarda la información
<b>endTime</b>		Tiempo de finalización para la simulación
<b>deltaT</b>		Paso de tiempo de la simulación si tal usa paso de tiempo fijos
<b>adjustTimeStep</b>		Controla si los pasos de tiempo son de longitud variable o fija
<b>runTimeModifiable</b>		Controla si la terminal puede leer algunos diccionarios cuando empieza cada paso de tiempo
<b>maxCo</b>		Puede imponer un límite superior en el tamaño del paso del tiempo (cuando la simulación se configura con un paso de tiempo ajustable)
<b>maxDeltaT</b>		Puede proporcionar un límite superior y manual para el paso de tiempo máximo (cuando la simulación se configura con un paso de tiempo ajustable)

Adaptado de (Holzinger, 2020).

Tabla 5

*Definición de escritura para el registro del tiempo de simulación.*

		<b>runTime</b>	Se registran los datos por cada intervalo de escritura en segundos
<b>writeControl</b>	Tiempo de escritura de datos en el archivo	<b>adjustableRunTime</b>	Permite que el solucionador acomode el paso del tiempo

	<b>timeStep</b>	Registra los datos de cada paso de tiempo estipulado por "writeInterval"
<b>writeInterval</b>		Es un valor que puede controlar el intervalo de escritura de los datos
<b>writeFormat</b>		Posee el control de cómo se escriben los datos en la memoria
<b>writePresicion</b>		Puede controlar la precisión de los valores escritos en la memoria
<b>writeCompression</b>		Determina si se comprimen o no los archivos registrados
<b>timeFormat</b>		Especifica el formato a utilizarse para escribir las carpetas de pasos de tiempo
<b>timePrecision</b>		Determina la cantidad de decimales para el tiempo
<b>purgeWrite</b>		Esta disposición puede controlar si se deben eliminar los pasos de tiempo antiguos

Adaptado de (Holzinger, 2020).

### 2.6.10. Condiciones de borde

Son la agrupación de restricciones que prescriben las condiciones de contorno para la malla computacional, ya que tal comprende ciertas caras de su geometría que se encuentra limitante al dominio físico, por lo tanto, son necesarias para lograr completar un cierre de las ecuaciones gobernantes del sistema de la dinámica de fluidos computacional. Todo caso de estudio demanda el ingreso de condiciones iniciales para poder configurar las variables que presenta el flujo y colocar los valores iniciales en los nodos del dominio del flujo (Holzinger, 2020).

### 2.6.10.1. Base

Tabla 6

*Lista de parches de límites geométricos.*

Parches	
symmetry plane	El contorno límite que se presente es de tipo plano de simetría y se modela solo la mitad del dominio
empty	Corresponde a una simulación bidimensional para que los límites que se encuentran paralelo al plano puedan determinarse
wedge	Simplifica los problemas para geometría de tipo axisimétrica, en donde solo se modela una parte en forma de cuña de la geometría
cyclic	Corresponde a límites cíclicos
processor	Genera un límite para los subdominios formados durante la descomposición del dominio físico

Adaptado de (Holzinger, 2020).

Tabla 7

*Lista de parches de límites complejos.*

Parches	
patch	Es un parche de tipo genérico para todos los límites
wall	Es de tipo especial y necesario para aplicar modelos de pared cuando se modela turbulencias

Adaptado de (Holzinger, 2020).

### 2.6.10.2. Primitivos

Tabla 8

*Lista de condiciones de contorno de tipo primitivo.*

	Condiciones límite
fixedValue	Prescripción directa del valor de una cantidad
fixedGradient	Prescripción directa del gradiente de una cantidad
zeroGradient	Prescribe a cero el gradiente de una cantidad

Adaptado de (Holzinger, 2020).

### 2.6.10.3. Derivados

Tabla 9

*Lista de condiciones de límites derivados del tipo primitivo.*

	Condiciones límite
inletOutlet	Comportamiento depende de dirección del flujo. Cuando el flujo se direcciona hacia afuera, se aplica la condición “zeroGradient”. Si el flujo se direcciona hacia adentro, se establece “fixedValue”
surfaceNormalFixedValue	Se ocupa de la dirección del vector correspondiente a la superficie del parche y se indica un valor positivo cuando la cantidad se alinea en la misma dirección que el vector y un valor negativo cuando es opuesta

pressureInletOutletVelocity	Combinación de condiciones estándar para el flujo de salida, pero para el flujo de entrada la velocidad puede encontrar su propio valor
totalPressure	Condiciona un valor fijo a partir de la velocidad y presión
fixedFluxPressure	Ajusta el gradiente de presión logrando que la condición de velocidad coincida con el flujo límite
noSlip	Condiciona un antideslizamiento en paredes fijas

Adaptado de (Holzinger, 2020).

## 2.7. Generación mallado computacional

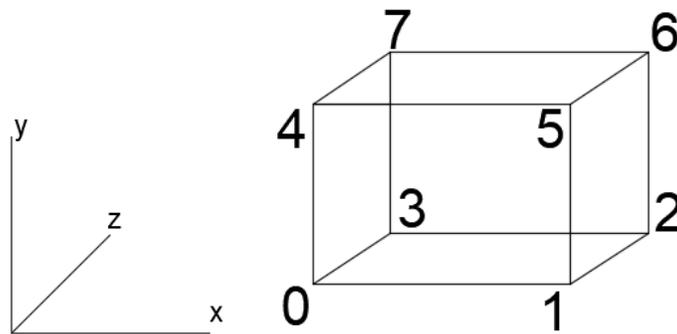
La transformación del dominio original hacia un contorno discretizado se lleva a cabo utilizando el comando “ideasUnvToFoam” para importar la geometría realizada en SALOME en formato UNV hacia la terminal.

OpenFOAM procede a definir la malla por medio de varios archivos, los cuales se encuentran en el directorio de “constant”, específicamente en el subdirectorio “polyMesh” y contiene los siguientes archivos (Holzinger, 2020):

- *boundary*: Posee una librería en donde se hallan todas las caras que conforman a los parches de límite o fronterizo.
- *faces*: Posee la definición de todas las caras.
- *neighbour*: Posee una librería con las celdas vecinas de todas las caras.
- *owner*: Posee una librería de las celdas pertenecientes de las caras.
- *points*: Posee una librería con las coordenadas de todos los puntos.

### 2.7.1. blockMesh

Herramienta de preprocesamiento aplicada a la geometría para crear una malla, la cual define y combina la geometría en relación con otras mallas del archivo original. En teoría la malla computacional creada a partir de la geometría se fundamenta en la generación de un bloque genérico que se define por los números de cada uno de sus vértices, los cuales se encuentran enumerados en el sentido antihorario en los planos de las coordenadas x-y, comenzando por el vértice en la coordenada z local. Por medio de aquellos vértices se procede a la construcción de los bloques y un bloque se configura utilizando una lista que comprenda los 8 vértices que se ordenaran para lograr coincidir con los vértices locales. Los vértices locales juegan un papel muy importante para determinar la clasificación de la malla y también la cantidad de celdas (Holzinger, 2020).



*Figura 1.* Bloque genérico de malla con vértices enumerados.

### 2.7.2. surfaceFeatureExtract

Es una herramienta que permite manipular la superficie de malla para lo cual es necesario trabajar con los archivos “STL” de la geometría, ya que tales archivos poseen un lenguaje de teselación de superficie que se aplica previo al

procesamiento de la creación de una malla con “snappyHexMesh”. De tal manera que su funcionalidad es extraer las propiedades y características de los “STLs” que no son nada más que los grupos geométricos de toda la estructura, por lo tanto, se definen los bordes de las características que se encuentran en el archivo (Holzinger, 2020).

### **2.7.3. snappyHexMesh**

Es una herramienta extremadamente fuerte que se aplica para crear mallas tridimensionales de carácter hexadecimal que se divide automáticamente con el propósito de refinar el mallado ajustándolo a la superficie, en donde se necesita obligatoriamente partir de las geometrías en formato “STL”, para que la malla se vaya adaptando aproximadamente a la superficie deseada por medio del refinamiento iterativo de una malla base y la transformación de la malla hexagonal que resulta de la superficie. Se deben realizar especificaciones para determinar la calidad de la malla (Greenshields, 2019).

#### **2.7.3.1. Flujo de trabajo**

##### **I. Creación malla de fondo**

Por medio de “blockMesh” se genera una malla de fondo construida por celdas hexagonales en su totalidad, es decir, celdas en forma de cubo. También se presentan muchas intersecciones con relación a los bordes de las celdas de la malla computacional de fondo con la tri superficie (Greenshields, 2019).

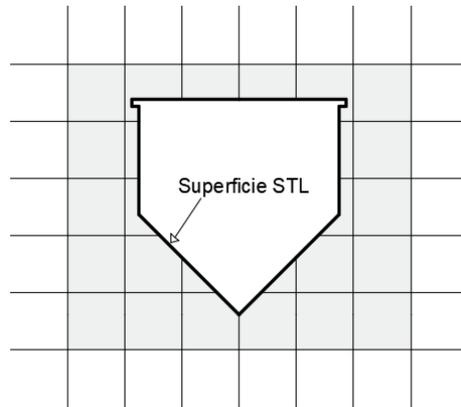


Figura 2. Malla esquemática para snappyHexMesh.

## II. snappyHexMesh

- *Castellating*

Divide las celdas de la malla de fondo, en donde aquellas celdas que se encuentran fronterizas a la superficie se someten a un refinamiento, razón por las que eliminan celdas de la malla de fondo dentro de la estructura (Greenshields, 2019).

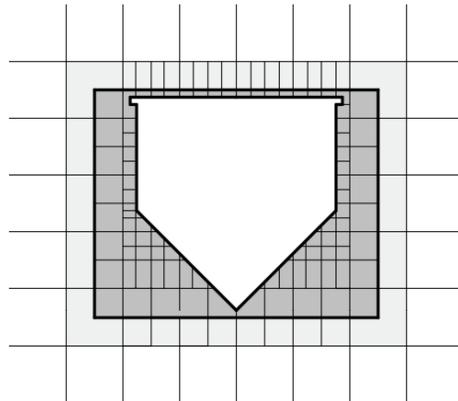
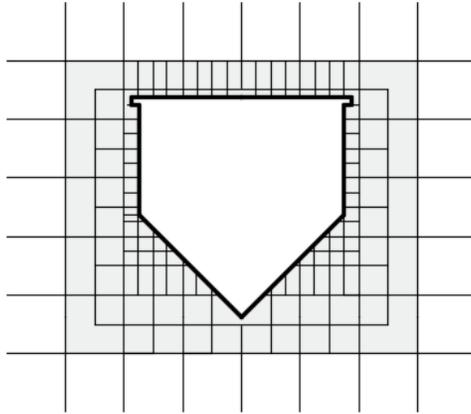


Figura 3. División y eliminación de celdas por región en el proceso de mallado por snappyHexMesh.

- *Snapping*

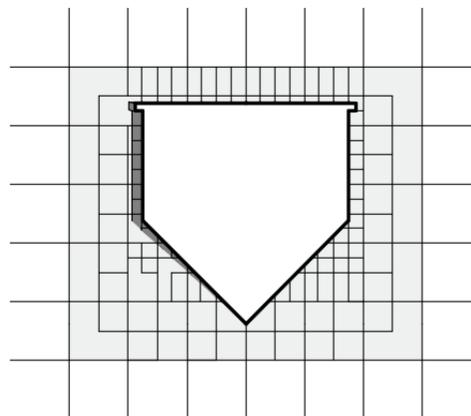
Es el proceso en donde se ajustan las celdas, provocando que la malla de fondo sobrante se someta a una modificación con la finalidad de reconstruir la superficie de la estructura (Greenshields, 2019).



*Figura 4.* Ajuste de la superficie en el proceso de mallado por snappyHexMesh.

- Capas de malla

Se incorporan celdas de forma hexaédrica como adiconantes para la superficie de contorno de la estructura para validar la calidad de la malla (Greenshields, 2019).



*Figura 5.* Adición de capa en el proceso de mallado por snappyHexMesh.

## **2.8. Simulación en paralelo**

La aplicación metodológica para realizar una computación en paralelo se fundamenta esencialmente en la descomposición del dominio físico, es decir, los campos y la geometría relacionada se fragmenta en varias partes dependiendo de la asignación a los procesadores del computador para su posterior solución, en donde las instancias del solucionador procederán a ejecutar el caso de estudio en un subdominio cada una. Además, el cómputo en paralelo se puede ejecutar no solo para los solucionadores, sino que también se pueden aplicar a varias herramientas de previo y posterior procesamiento (Towara, Schanen, & Naumann, 2015).

### **2.8.1. MPI**

Se conoce como una especificación estándar de interfaz que contiene librerías de paso de mensajes la cual aborda la capacidad de modelar una programación en paralelo, logrando que la información se movilice del espacio en donde se encuentran las direcciones del proceso de simulación hacia otro proceso por medio de la ejecución cooperativa operacional en cada uno de los procesos, lo que genera una sincronización equitativa entre procesos permitiendo obtener una exclusión mutua (Towara et al., 2015).

La herramienta que activa las librerías se reconoce como “mpixec” y puede hacer uso de las librerías tanto de Intel como de Microsoft dependiendo las preferencias del usuario. Por otra parte, la manera correcta de simular el comando en la terminal de OpenFOAM dependerá de la cantidad de procesadores a disposición y de la selección del solucionador o herramienta de previo o posterior procesamiento (Holzinger, 2020). Se presenta un ejemplo del comando completo:

“mpiexec -n 4 “herramienta a usar” -parallel”

Donde:

mpiexec: Activador de librerías

n 4: El número de subdominios de procesamiento

parallel: Especificación de cómputo en paralelo

### **2.8.2. Descomposición del dominio**

El comando “decomposeParMesh” es aplicado para realizar la tarea de descomposición en donde se fragmenta el dominio de estudio en la cantidad correcta de subdominios configurado por el usuario, para cada uno de los procesos en paralelo y dichos procesos paralelos realizan su dominio independientemente, compartiendo la información de capas límite al finalizar cada paso de tiempo (Greenshields, 2019).

### **2.8.3. Reconstrucción del dominio**

Aplica necesariamente el comando “reconstructParMesh” para poder visualizar los resultados, puesto que la herramienta se encarga de realizar una recopilación de toda la información que albergan las carpetas de cada procesador y vuelve a juntar todas las partes obteniendo un dominio reconstruido como el original utilizando los datos de paso de tiempo generados (Holzinger, 2020).

## 2.9. ParaView

Es un programa de código abierto para el post procesamiento gráfico de datos que trabaja en función de las herramientas de visualización “VTK”, y se utiliza para generar visualizaciones 3D con la finalidad de realizar un análisis técnico cuantitativo y cualitativo, como la representación aerodinámica de los campos vectoriales correspondientes a las variables de estudio (Farahani, 2018).

Filtros predefinidos para visualización estructural:

- Countour: Tiene la capacidad de extraer las isocurvas o isosuperficies del campo escalar (Moreland, 2014).
- Clip: Recorta la estructura de datos, logrando una visualización de los valores intrínsecos de su campo en la profundidad de su estructura de información (Farahani, 2018).
- Slice: Realiza una división de la estructura de datos en relación con su plano predefinido (Farahani, 2018).

Filtros vectoriales para visualización de magnitud y dirección:

- Glyph: Define la visualización del campo vectorial como la velocidad, por medio de flechas para indicar su densidad, dirección y magnitud (Moreland, 2014).
- Stream Tracer: Define la visualización del campo vectorial como líneas continuas de corriente para indicar su densidad, dirección y magnitud (Moreland, 2014).

- Warp By Vector: Crea una mejor visualización en cuanto a desplazamiento o velocidades de acuerdo con la estructura de datos de un campo vectorial (Moreland, 2014).

### 3. METODOLOGÍA

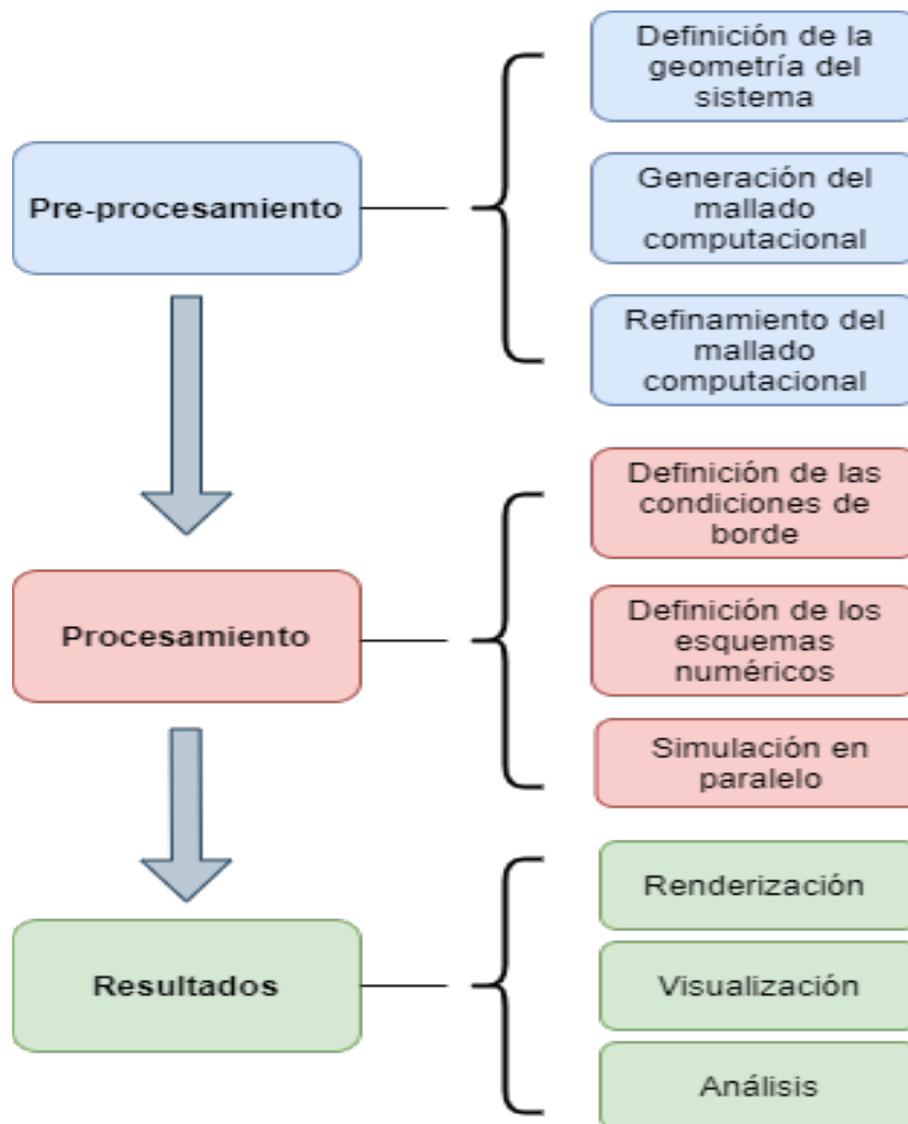


Figura 6. Esquema metodológico para simulación CFD.

### 3.1. Pre-Procesamiento

Es el punto de arranque para la resolución hidráulica del sistema, en donde es necesario alimentar de toda la información inicial para obtener una representación que asemeje las condiciones reales que se desean simular, acorde a los parámetros del sistema.

#### 3.1.1. Geometría

En un principio se procedió a extraer las dimensiones de las estructuras del clarificador secundario de los planos de construcción AutoCAD, y se realizaron las adecuaciones necesarias para establecer correctamente el diseño final en SALOME, el cual abarca la entrada, superficies laterales, superficies base y salida del sistema.

La construcción de la geometría del clarificador secundario en SALOME se basó en la creación de varios sólidos de forma cilíndrica y cónica dentro del plano. La estructura se compone por cinco sólidos independientes que se someten a traslaciones, corte, fusión y partición. Se presentan de la siguiente manera:

Tabla 10

*Dimensionamiento de la geometría del clarificador secundario*

ID	Solidos	Dimensiones	
		Díámetro (m)	Altura (m)

---

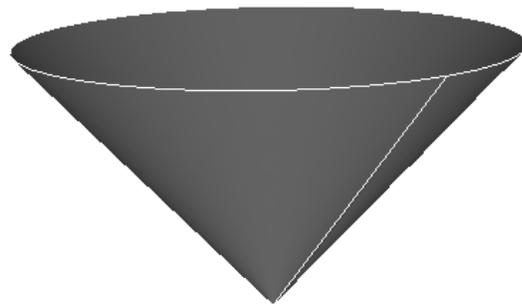
0	Cono	1.4	1.4
1	Cilindro I	1.4	1.5
2	Cilindro II	0.36	1.5
3	Cilindro III	1.5	0.1
4	Cilindro IV	0.0508	0.2

---

*Nota. La estructura general del clarificador secundario presenta una altura total de 3 metros y un ancho de 3 metros.*

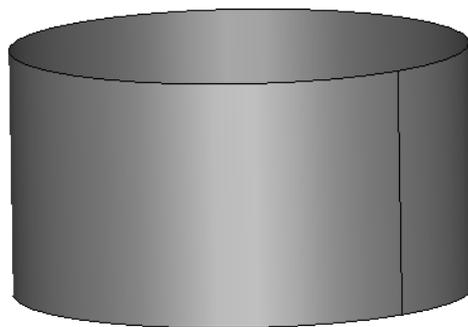
---

- Cono:



*Figura 7.* Estructura decantadora de sólidos del clarificador secundario.

- Cilindro I:



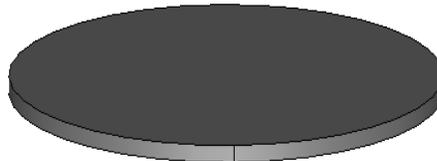
*Figura 8.* Tanque del clarificador secundario.

- Cilindro II:



*Figura 9.* Estructura rompe presión para la entrada del afluente al clarificador secundario.

- Cilindro III:



*Figura 10.* Rebosadero para salida de agua del clarificador secundario.

- Cilindro IV: Tubería de salida para el efluente clarificado



*Figura 11.* Tubería de salida para el efluente del clarificador secundario.

Para conseguir un único sólido final, se fusionaron los sólidos 1,3 y 4, los cuales conforman el tanque, rebosadero y salida, consecuentemente se utilizó el sólido 2 para realizar un corte desde el tope central del nuevo cuerpo sólido, obteniéndose la configuración para la entrada de agua. Finalmente, se aplicó la operación de partición para crear un compuesto a partir del sólido 0 y el sólido fusionado con corte por los sólidos 1, 2, 3 y 4.

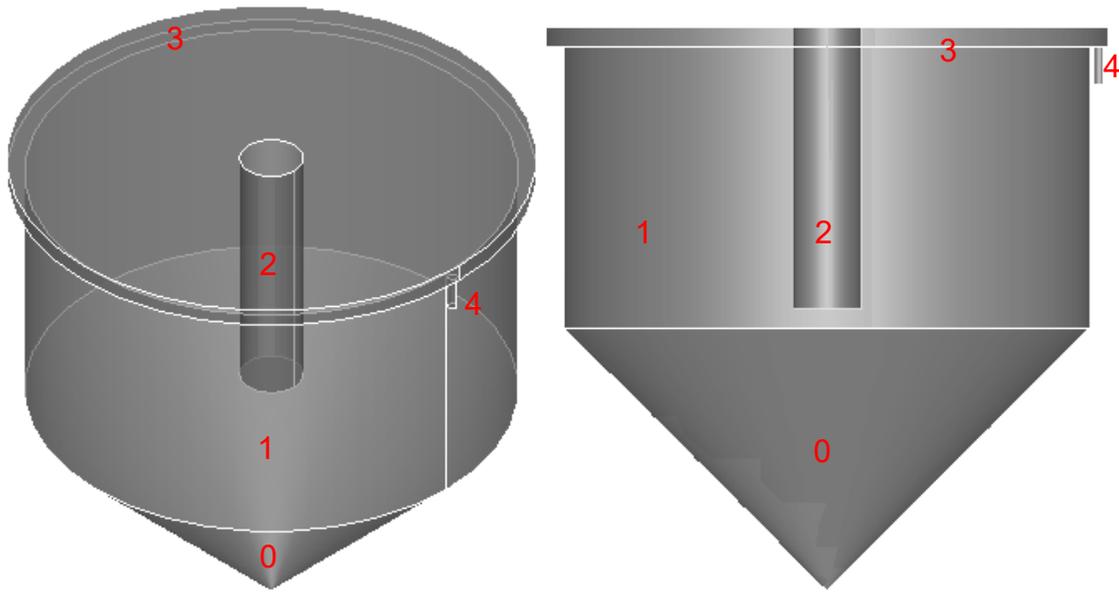


Figura 12. Geometría construida del clarificador secundario.

### 3.1.2. Generación del mallado computacional

#### 3.1.2.1. SALOME MESH

Por medio de la geometría se generó un mallado de finura moderada utilizando las herramientas del software SALOME.

Tabla 11

*Información de malla computacional.*

Parámetros	Total
Nodos	2930

Bordes	422
Caras triangulares	3434
Volúmenes tetraédricos	11818

Consecuentemente el mallado se sometió a una diferenciación de grupos que determinaron a las regiones de interés del sistema por medio de la selección de sus caras, resultando cuatro grupos de interés que hacen referencia a la entrada (*inlet*), salida (*outlet*), atmósfera (*atmosphere*) y paredes (*walls*).

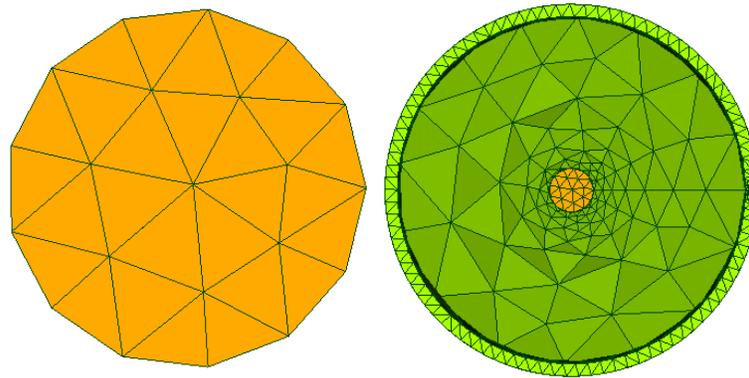


Figura 13. Malla de la geometría de *inlet*.

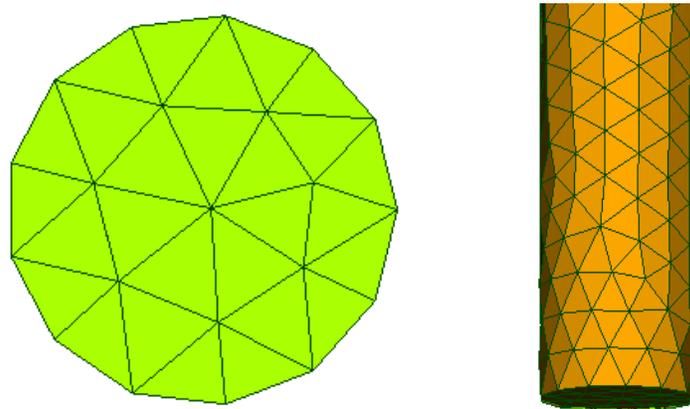


Figura 14. Malla de la geometría de *outlet*.

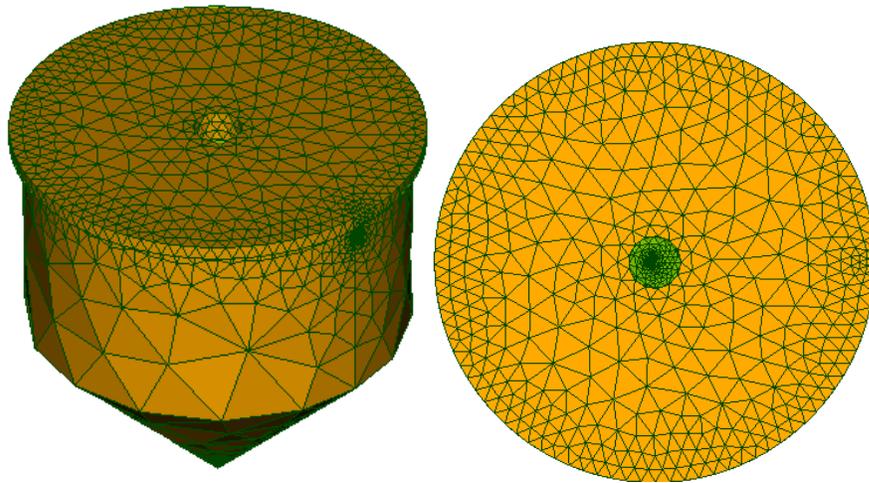


Figura 15. Malla de la geometría de *atmosphere*.

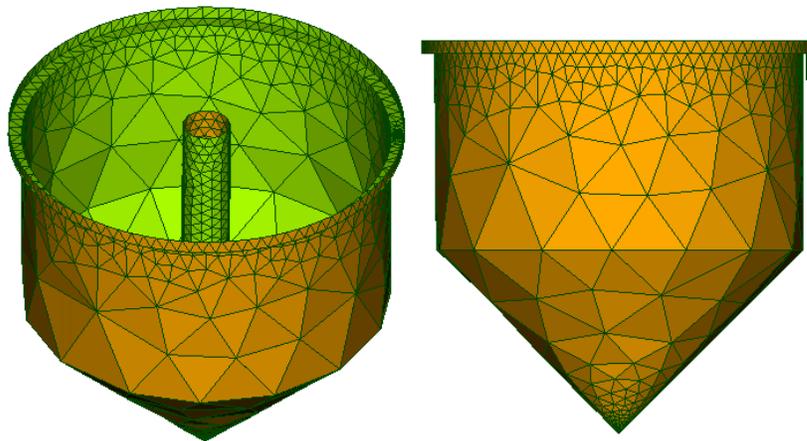
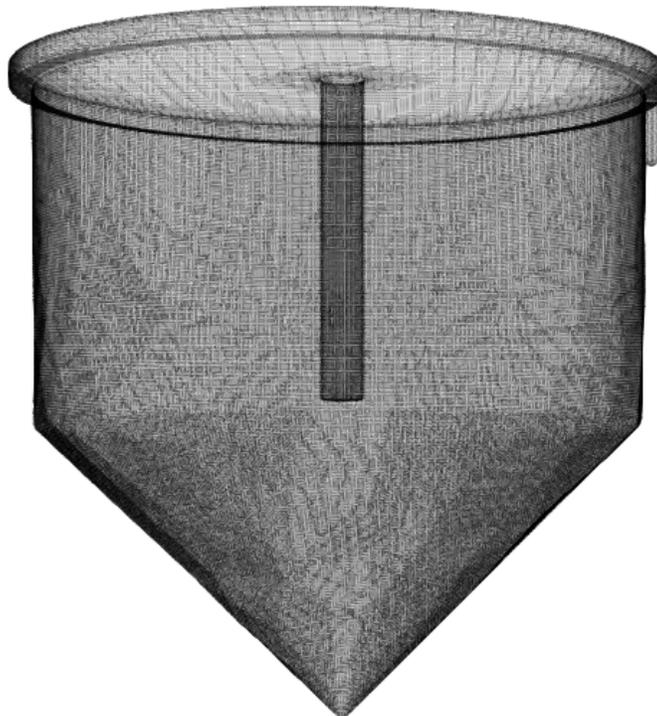


Figura 16. Malla de la geometría de *walls*.

### 3.1.2.2. snappyHexMesh (OpenFOAM)

Se procedió a realizar el refinamiento y ajuste de la malla, en donde primero se configuró el recurso “blockMesh” de OpenFOAM para construir la malla de fondo y así poder descomponer la geometría del dominio en un conjunto de bloques

hexaédricos, los cuales contendrán las superficies y subsuperficies de la geometría. Posteriormente se configuró el subdirectorio “snappyHexMesh” donde se definieron las entidades geométricas, las características para el refinamiento de malla del volumen y las superficies, también se programaron los controles para el ajuste de las mallas superficiales y de las mallas de capa límite, y por último se delimitó métricamente la configuración para la calidad final del nuevo mallado.



*Figura 17.* Mallado final del clarificador secundario

### **3.2. Procesamiento**

En este apartado es donde se ejecuta la simulación por medio del solucionador “interFoam”, el cual se integra y ejecuta en el sistema computacional de OpenFOAM, teniendo que modificar y controlar los parámetros necesarios de solución.

### 3.2.1. Condiciones iniciales y de borde

Generada la malla se configuraron los campos iniciales que se especifican principalmente en las condiciones de entrada y los límites estructurales físicos, los cuales llegan a coincidir con las limitantes del volumen completo. Razón dio paso a la introducción de los lineamientos de contorno para las cuatro regiones que componen el clarificador secundario.

Las condiciones de entrada se refieren al estado del fluido, en condiciones de presión y velocidad al comienzo del periodo de simulación. Las condiciones de entrada que se establecieron cumplieron la tarea de determinar el estado del fluido en el área de estudio, en sentido que tales condiciones representaron una gran influencia en la calidad de la simulación.

Tabla 12

*Condiciones de borde del clarificador secundario.*

Región	Condiciones		
	Agua	Presión	Velocidad
<i>inlet</i>	fixedValue	fixedFluxPressure	fixedValue
<i>outlet</i>	zeroGradient	fixedFluxPressure	inletOutlet
<i>atmosphere</i>	inletOutlet	totalPressure	pressureInletOutletVelocity
<i>walls</i>	zeroGradient	fixedFluxPressure	noSlip

Las condiciones de contorno se trabajaron desde el preprocesamiento, debido a que toda la información que generó la malla de la geometría se encontró disponible

en los archivos de OpenFOAM. Consecuentemente se establecieron las propiedades del fluido para el caso de estudio, el cual estableció que se trabajara con agua (alpha.water) como un líquido incompresible.

- ***Inlet***

Se configuró la entrada de agua hacia el sistema como un modelo despresurizado, lo cual indica su condición “fixedFluxPressure”, donde se muestra que el gradiente de presión es igual a cero pero que presenta la dinámica de las fuerzas gravitacionales y de tensión superficial. El modelo referente a la velocidad de ingreso se condicionó de acuerdo con el valor del caudal y de su velocidad especificando la cantidad y dirección de flujo.

La entrada de agua hacia el sistema es por una tubería de dos pulgadas que se dirige hacia la estructura cilíndrica central rompe presión, con un caudal de 0.8 l/s. Para hallar la velocidad de entrada fue necesario resolver y despejar la siguiente ecuación de caudal:

$$Q = A * V$$

$$A = \pi * r^2$$

$$Q = (\pi * r^2) * V$$

Resolución:

Área de entrada:

$$A = \pi * (2.54cm)^2$$

$$A = \pi * 6.4516 cm$$

$$A = 20.26 cm$$

$$A = 0.20 m$$

Velocidad:

$$Q = A * V$$

$$V = Q/A$$

$$V = \frac{0.0008 \text{ m}^3/\text{s}}{0.2 \text{ m}}$$

$$V = 0.004 \text{ m/s}$$

Siendo:

$Q$ : Caudal ( $\text{m}^3/\text{s}$ )

$A$ : Área ( $\text{m}^2$ )

$V$ : Velocidad ( $\text{m/s}$ )

$r$ : Radio ( $\text{cm} \rightarrow \text{m}$ )

El agua ingresa verticalmente hacia abajo, por lo tanto, su valor de entrada se especificó en la coordenada "z" con un signo negativo (0 0 -0.004).

- **outlet**

Las condiciones referentes a la presión de salida representaron de igual manera un ámbito despresurizado que contempló a la gravedad y la tensión superficial. El modelo de velocidad se rigió por la orientación del flujo de agua, de modo que se otorgó el mismo valor referente a la velocidad de entrada, puesto que la configuración del funcionamiento del clarificador secundario indicó que la liberación del flujo se realiza por una tubería de dos pulgadas al igual que la tubería de entrada.

- **atmosphere**

Se condicionó la región como una atmósfera abierta, en donde el flujo entra en contacto con el aire y los parámetros de presión reflejan un valor directo en relación con la velocidad local y la presión total especificada, mientras que para su modelo

de velocidad se estableció una composición de condiciones para su flujo de entrada y salida.

- **walls**

En las paredes de la geometría, se aplicó la condición de límite de presión de flujo en donde se establece un gradiente de presión igual a cero, presentando fuerzas de gravedad y de tensión. Su modelo de velocidad se configuró de acuerdo con los parches específicos de paredes en donde se estableció que no posea un deslizamiento “noSlip”.

### 3.2.2. Esquemas numéricos

En la sección numérica se definió el método de discretización para volúmenes finitos, utilizando el solucionador “interFoam” para los cálculos durante toda la simulación.

Tabla 13

*Esquema de discretización “fvSchemes” del clarificador secundario.*

Esquemas		interFoam
ddtSchemes	default	Euler
gradSchemes	default	Gauss linear
divSchemes	div(rhoPhi,U)	Gauss linearUpwind grad(U)
	div(phi,alpha)	Gauss vanLeer
	div(phirb,alpha)	Gauss linear
	“div\(\phi,(k \omega)\)”	Gauss upwind

	$\text{div}(((\rho * \nu_{\text{Eff}}) * \text{dev2}(\text{T}(\text{grad}(\text{U}))))))$	Gauss linear
laplacianSchemes	default	Gauss linear corrected
interpolationSchemes	default	linear
snGradSchemes	default	corrected

Los subdirectorios de discretización albergan una variedad de esquemas numéricos que desarrollan los cálculos de los derivados en las ecuaciones mientras se ejecuta la simulación. A continuación, se presentan la división de categorías y sus especificaciones:

- **Esquema de tiempo “*ddtSchemes*”**

La especificación de la primera derivada para el caso de estudio utilizó el esquema “Euler”, el cual se describe como de primer orden, implícito y acotado, puesto que se ha tomado en cuenta el régimen estacionario para el modelo, asimilando que los parámetros de los gradientes se mantendrán invariables con respecto al tiempo de simulación (Moukalled et al., 2016).

- **Esquema de gradiente “*gradSchemes*”**

El subdirectorio predeterminado que se aplicó hace referencia a “*Gauss linear*”, logrando determinar la discretización para el volumen finito en donde es necesario realizar una interpolación lineal pertinente a los ejes centrales de las caras y celdas para la integración gaussiana (Moukalled et al., 2016).

- **Esquema de divergencia “*divSchemes*”**

El esquema posee términos de divergencia y se identifica por la condición de convección representado por “div (phi)” donde phi denota el flujo de velocidad en las caras de las celdas con densidad constante para la integración gaussiana (Greenshields, 2019).

- **Esquema de gradiente normal de superficie “*snGradSchemes*”**

Se aplicó la evaluación del esquema en la cara de una celda para establecer una gradiente normal, el cual realiza un cálculo de segundo orden, restando el valor ubicado en el centro de las celdas, del valor en el centro del otro lado de la cara y se divide para la distancia. Su determinación por defecto se presenta como el esquema “*corrected*”, el cual se aplicó como una corrección al componente ortogonal donde se incrementa el ángulo entre los lados del mallado ofreciendo una mejor estabilidad (Moukalled et al., 2016).

- **Esquema laplacianos “*laplacianSchemes*”**

El esquema aloja una serie de términos laplacianos “ $\nabla$ ”, que representan un término de difusión para las ecuaciones gobernantes de momento por medio de la discretización gaussiana, la cual requirió de un esquema de interpolación, presentando una integración gaussiana corregida de segundo orden que indica parámetros limitantes y conservadores (Moukalled et al., 2016).

- **Esquema de interpolación “*interpolationSchemes*”**

Se aplicaron interpolaciones de valores que parten de los centros de las celdas hacia los ejes centrales de las caras, en donde se utilizó la interpolación lineal por defecto para el flujo de velocidad ( $\phi$ ) (Greenshields, 2019).

### 3.2.3. Control y solución de algoritmos

El solucionador es un recurso numérico que concede paso a la integración de las ecuaciones que gobiernan al sistema, la discretización de tales ecuaciones y la resolución de ecuaciones algebraicas (Jayanti, 2018). A continuación, se presenta la configuración desarrollada para el caso de estudio:

Tabla 14

*Esquema de solución “fvSolution” del clarificador secundario.*

Factores	Parámetros	
alpha.water	nAlphaCorr	1
	nAlphaSubCycles	1
	cAlpha	1
	MULESCorr	Yes
	nLimiterIter	3
	solver	smoothSolver
	smoother	symGaussSeidel
	tolerance	1e-8
	relTol	0
	pcorr	solver
preconditioner		GAMG
tolerance		1e-5

	relTol	0
	smoother	GaussSeidel
	maxIter	50
	solver	GAMG
	tolerance	5e-9
p_rgh	relTol	0.01
	smoother	GaussSeidel
	maxIter	50
p_rghFinal	tolerance	5e-9
	relTol	0
	solver	smoothSolver
	smoother	symGaussSeidel
(U k omega)	nSweeps	1
	tolerance	1e-6
	relTol	0.1

Se definieron los solucionadores lineales de control para las ecuaciones discretizadas, en donde se encuentran dos métodos numéricos que resolverán la ecuación matricial para velocidad y presión. Se definió “GAMG” como el solucionador para múltiples rejillas y “PCG” como la metodología de proyección simétrica con una matriz definida (Holzinger, 2020). También se utilizó “GAMG” como un pre-acondicionante para la corrección de presión (pcorr) que se activa por la incorporación de la malla hacia los directorios (Greenshields, 2019).

En cuanto al método iterativo se empleó “*smoothSolver*” para que la solución realice una reducción de la ecuación residual con el propósito de obtener la medida y análisis del error presente en la solución, motivo por el cual la tolerancia pretende

incluir coeficientes muy cercanos a cero (Holzinger, 2020). Además, fue necesario la aplicación de un suavizador al solucionador, en donde se utilizó el método de iteración simétrica “*symGaussSeidel*” y el método de iteración original “*GaussSeidel*”.

Por último, se estableció el algoritmo “PIMPLE” para la resolución de las ecuaciones gobernantes, especializándose en imponer la conservación de la masa, y corrigiendo la velocidad para la satisfacción de la conservación del momento. También se creó un ciclo de programación para todo el sistema de ecuaciones especificado por su intervalo de tiempo, representando la cantidad total de resoluciones del sistema.

Tabla 15

*Configuración del algoritmo de solución “PIMPLE”.*

<b>PIMPLE</b>	
momentumPredictor	Yes
nCorrectors	2
nNonOrthogonalCorrectors	0
pRefCell	0
pRefValue	0

### 3.2.4. Cómputo en paralelo

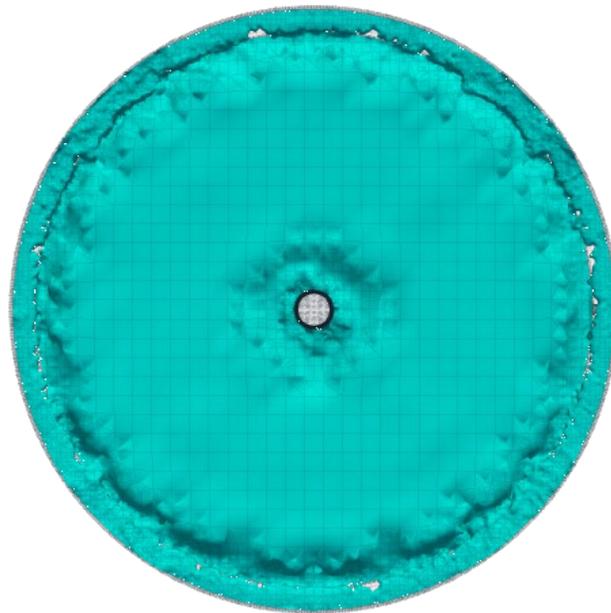
Se configuró el subdirectorio “*decomposeParDict*” para especificar los cuatro núcleos a disposición del computador para trabajar en la simulación del refinamiento

de malla “snappyHexMesh” y en la ejecución de solucionador “interFoam” por el método simple que ofrece OpenFOAM. El comando de ejecución de refinamiento de malla y del solucionador se presentan correspondientemente de la siguiente manera:

```
“mpiexec -n 4 snappyHexMesh -parallel”
```

```
“mpiexec -n 4 interFoam -parallel”
```

El caso de estudio requirió una simulación en donde el clarificador secundario se encuentre con un volumen de agua establecido, motivo por el cual se establecieron las coordenadas limitantes para indicar la altura de agua en la estructura, por medio del subdirectorio “*setFieldsDict*”.



*Figura 18.* Vista superior del contorno de nivel de agua en el clarificador secundario



*Figura 19.* Vista lateral del contorno de nivel de agua en el clarificador secundario

Por último, se calibró el tiempo y escritura de control necesaria para acercarse a la convergencia numérica y la estabilización de los residuales, factores que determinaron la calidad de la simulación.

Tabla 16

*Configuración de “controlDict” para tiempo y escritura de simulación.*

application	interFoam	purgeWrite	0
startFrom	startTime	writeFormat	ascii
startTime	0	writePrecision	8
stopAt	endTime	writeCompression	uncompressed

endTime	300	timeFormat	general
deltaT	0.005	timePrecision	6
writeControl	adjustableRunTime	runTimeModifiable	yes
writeInterval	1	adjustTimeStep	no
maxCo	3	maxAlphaCo	4
maxDeltaT	1	functions	include residuals

### 3.3. Post-Procesamiento

Obtenidos los resultados de la simulación CFD se procesó la visualización del comportamiento hidráulico del clarificador secundario y se procedió a analizar los datos modelados. En principio, se realizó la representación gráfica del volumen, superficies y componentes dinámicos del sistema, convirtiendo los datos discretizados del flujo en una proyección tridimensional.

#### 3.3.1. Visualización

Por medio de la interfaz gráfica de “ParaView” se pudo observar el volumen del clarificador secundario y sus parámetros del flujo. Se utilizaron los siguientes métodos visuales para comprender el comportamiento, distribución y dirección de las velocidades que alberga el campo de flujo del sistema:

- Vectores
- isolíneas de flujo
- Contornos
- Isosuperficies

## 4. RESULTADOS Y ANÁLISIS

### 4.1. Residuales de la simulación

La simulación requirió un tiempo de duración de 300 segundos con un paso de tiempo ( $\Delta T$ ) de 0.005 segundos para resolver cada iteración de la solución (tabla 16). Se comprobó el alcance de la convergencia numérica iterativa por medio del monitoreo de los residuales (Figura 20) correspondientes a los componentes de velocidad, los cuales se normalizaron a partir de los 225 segundos, indicando que el sistema hidráulico entro en estado estacionario.

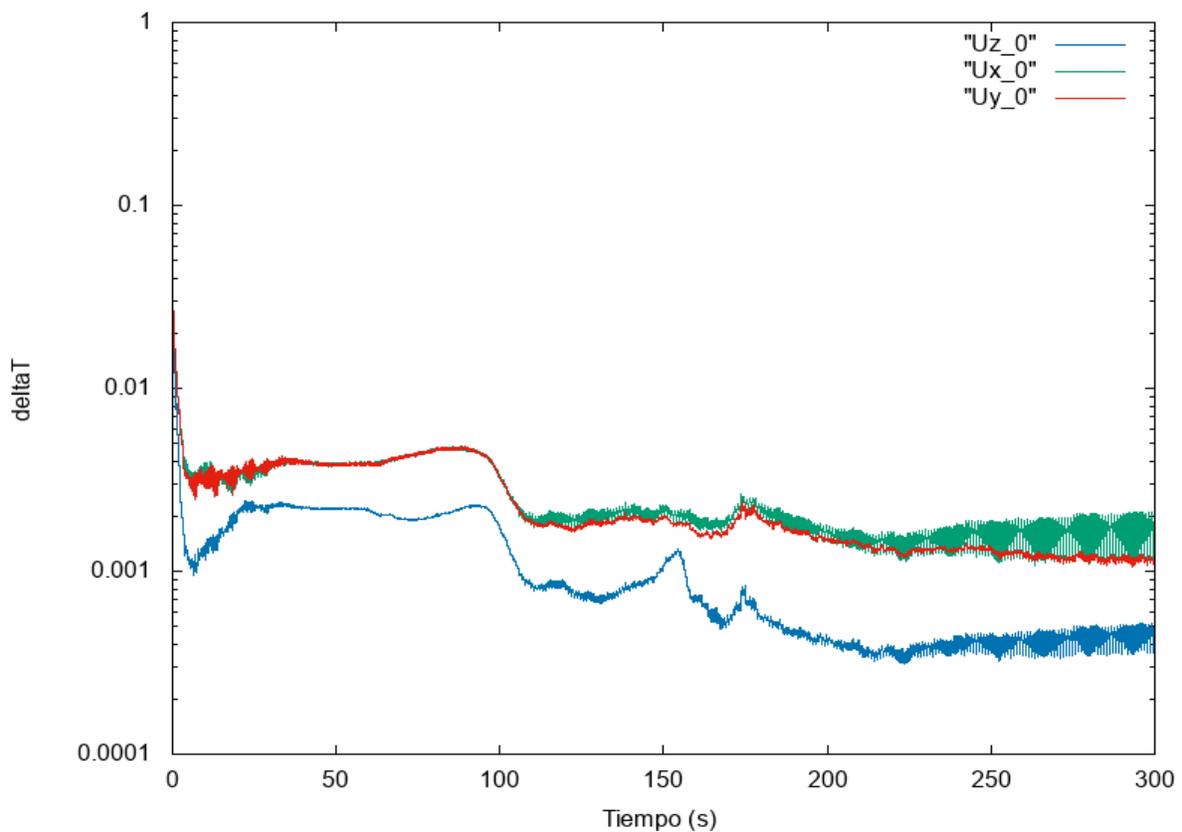


Figura 20. Monitoreo de residuales de las velocidades en los componentes (x,y,z).

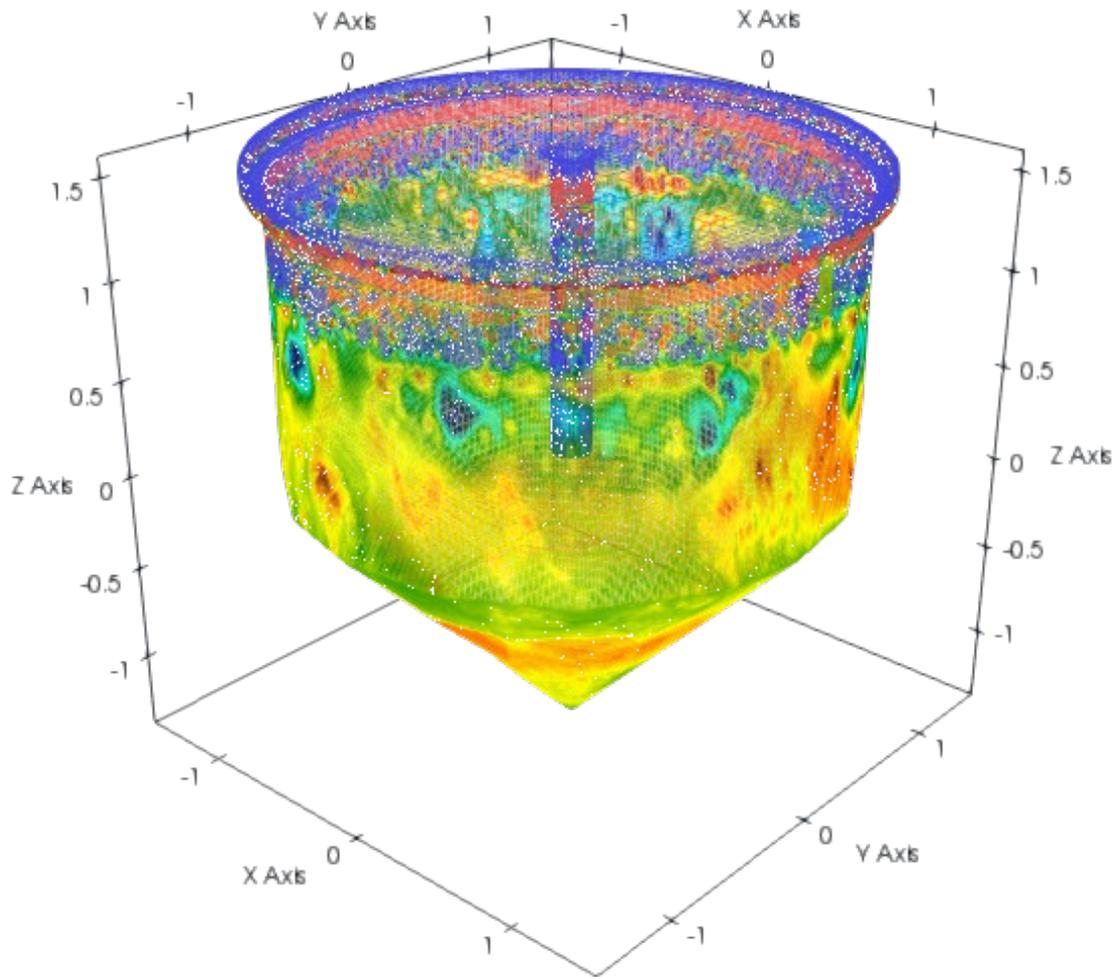
## 4.2. Campo de flujo

El clarificador secundario posee una altura total de 3 metros y un diámetro de 3 metros incluyendo un rebosadero. En cuanto a su configuración hidráulica, presenta un comportamiento de flujo vertical, empezando desde el ingreso del afluente hacia el fondo del tanque por la tubería de alimentación central. Consecuentemente el fluido se extiende a las zonas laterales del tanque y fluye ascensionalmente hasta llegar al límite superior con el rebosadero, seguidamente del límite, una parte del flujo se dirige hacia el vertedero de efluentes y la otra parte del flujo continúa su trayecto hacia el centro del tanque donde empieza su descenso hacia el fondo.

La representación del comportamiento del flujo dentro del sistema se interpretó por la magnitud de velocidad en el componente "z", donde los resultados de la simulación indican un rango de velocidades de 0 - 0.02 m/s, las cuales presentan un signo negativo cuando el flujo se dirige hacia abajo y un signo positivo cuando el flujo se dirige hacia arriba (Figura 21). Por lo tanto, la presencia de velocidades con valores de cero dentro del campo de flujo corresponde al estado en que el agua no recorre aquellas zonas, representando la transición de la dirección del flujo cuando dejan de subir o bajar.



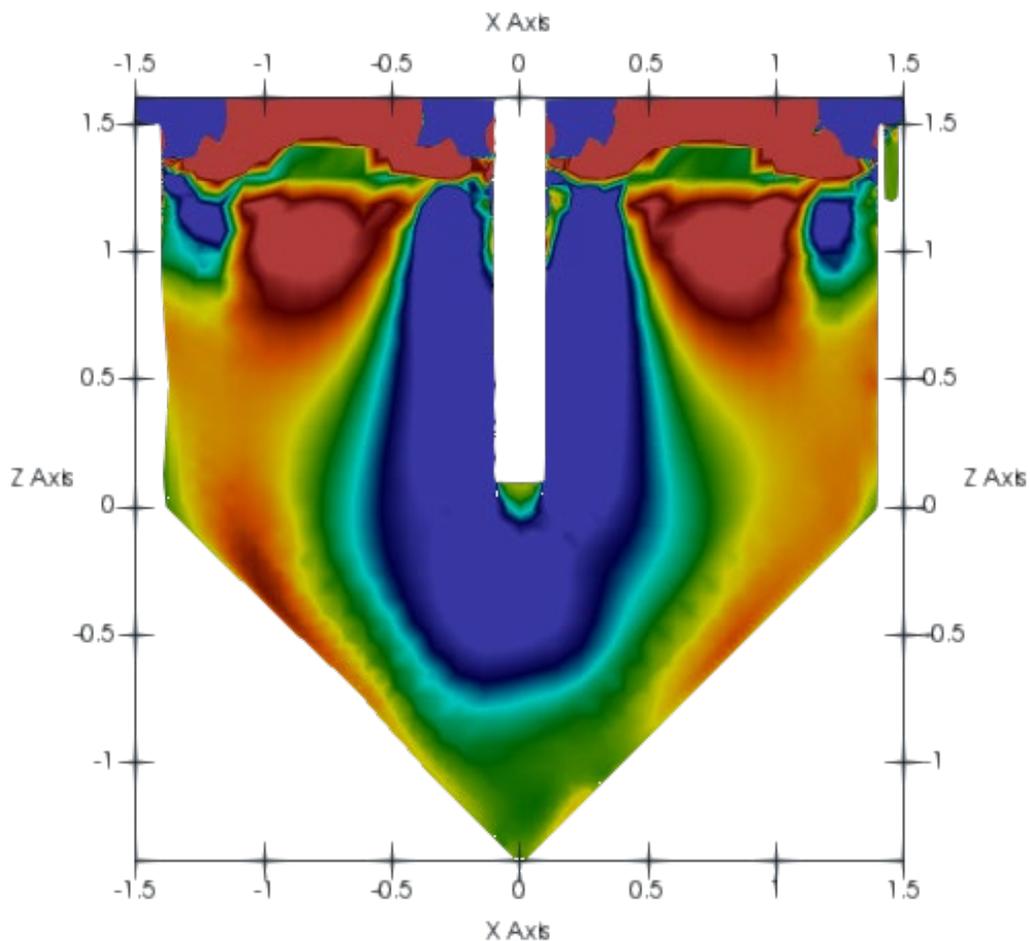
Figura 21. Rango de velocidades del agua en el eje "z" del clarificador secundario.



*Figura 22.* Distribución de la velocidad en las paredes del clarificador secundario en ParaView.

La dinámica de fluidos computacional del clarificador secundario demostró que su comportamiento hidráulico representa las condiciones adecuadas para que se realice la separación de sólidos del agua, contemplando primero la zona baja del tanque en la cual las velocidades del flujo son cero y oscilaciones de valores muy cercanos a cero (Figura 22) para que, en teoría, las partículas sólidas puedan aglomerarse y cumplir el proceso de espesamiento para mantenerse al fondo del tanque.

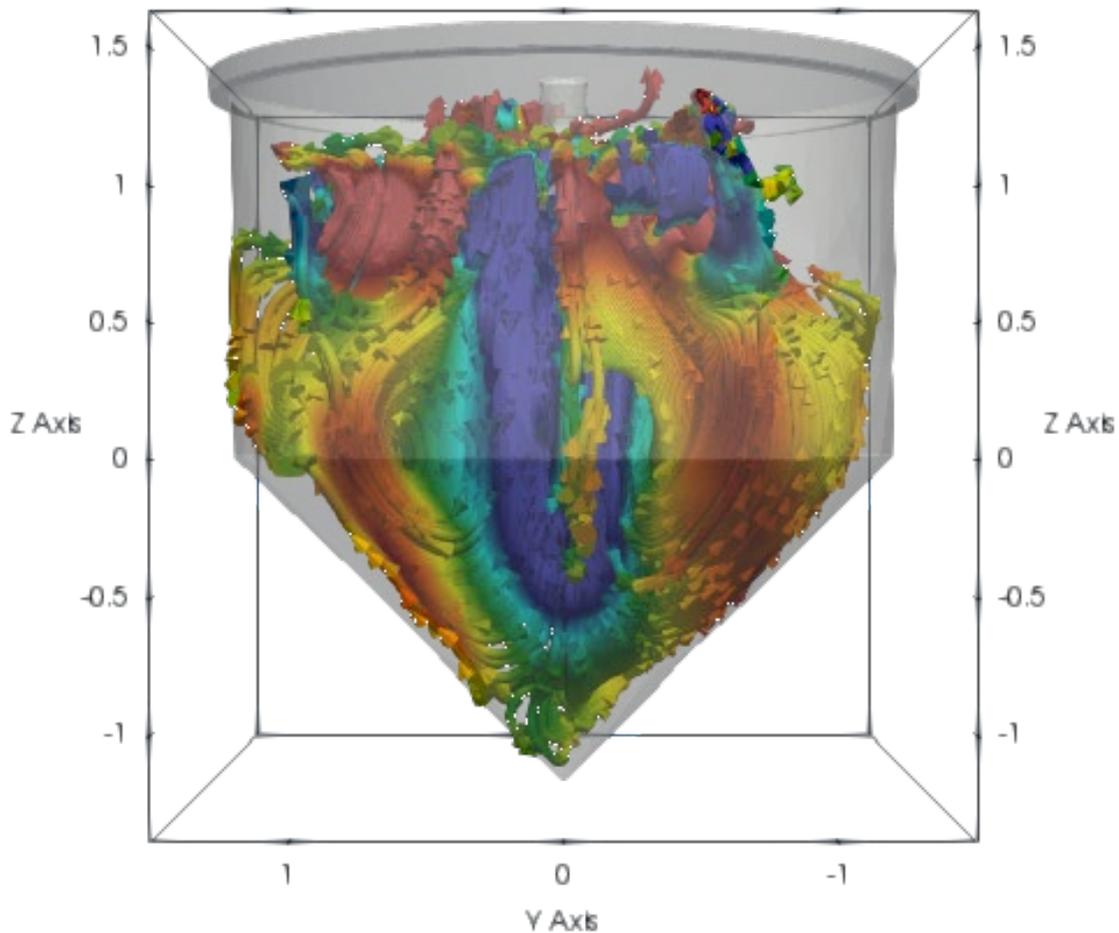
Correspondiente a la zona media del tanque clarificador se evidenció una correcta distribución de velocidades para el almacenamiento del agua, puesto que el modelo representa que el fluido posee una proporción adecuada de velocidades ascendentes en los flancos del tanque y velocidades descendentes en su centro (Figura 23), lo que genera un circuito limpio para el flujo. En consecuencia, de la distribución de velocidades de la zona baja y media, se presentó la zona superior del tanque con una interfaz de velocidades tope de ascenso y descenso (Figura 23), que en teoría provocarían la correspondiente separación de sólidos conduciéndolos hacia el fondo del tanque y por el otro lado la clarificación del efluente hacia el vertedero de salida.



*Figura 23.* Campo de flujo por velocidades “z” del clarificador secundario en ParaView.

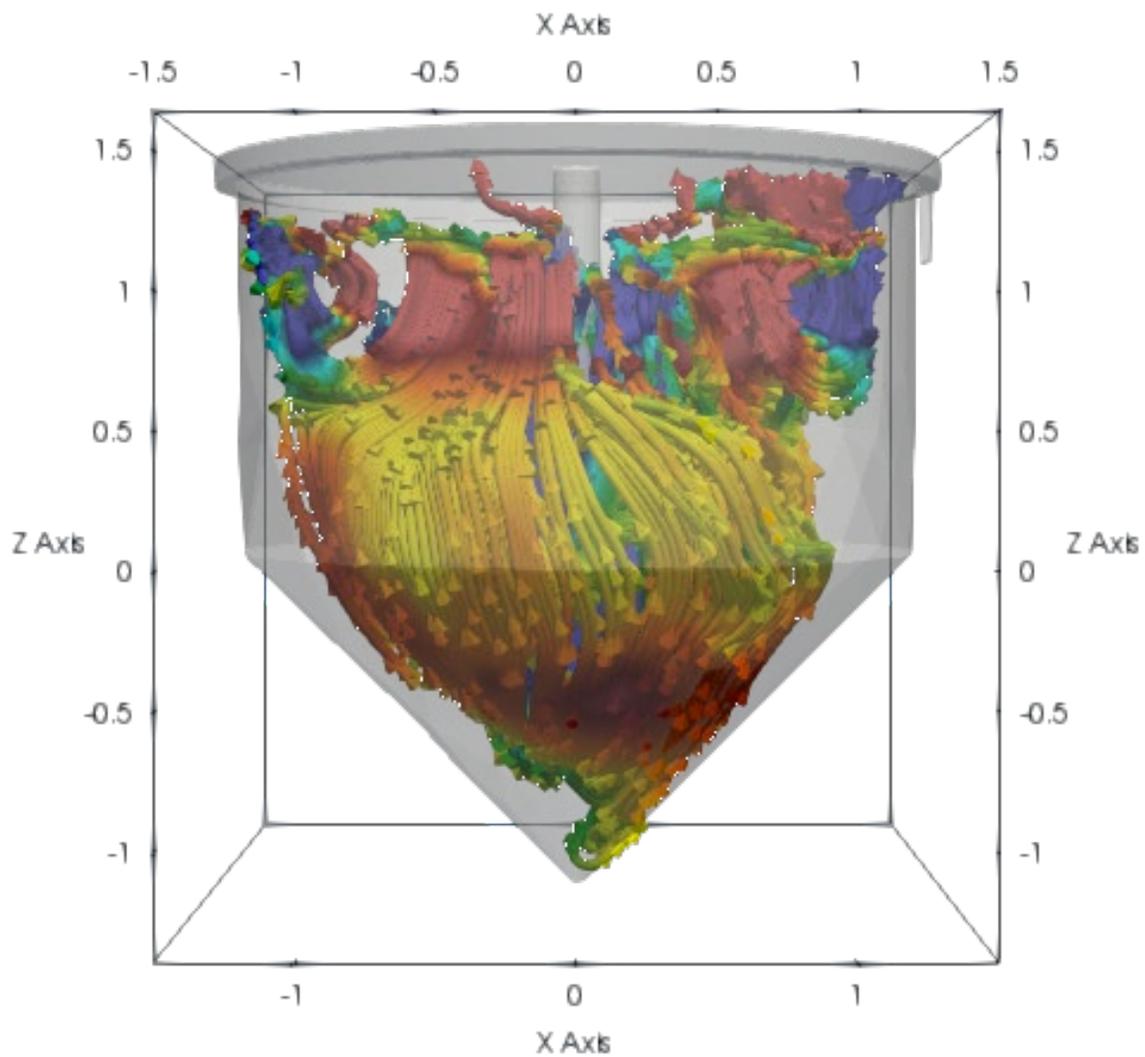
Se puede observar el campo vectorial unitario tridimensional en la dirección “z” de las velocidades dentro del clarificador secundario (Figura 24), para una mejor apreciación inicial de la distribución espacial y dirección del flujo.

De acuerdo con el campo vectorial (Figura 24) se puede visualizar la distribución central de las corrientes del fluido, las cuales presentan una estratificación con respecto a las velocidades descendentes en todo el alrededor de la tubería de alimentación, que a medida de su precipitación se empatan con el ingreso del flujo, logrando una incorporación y dirección homogénea.



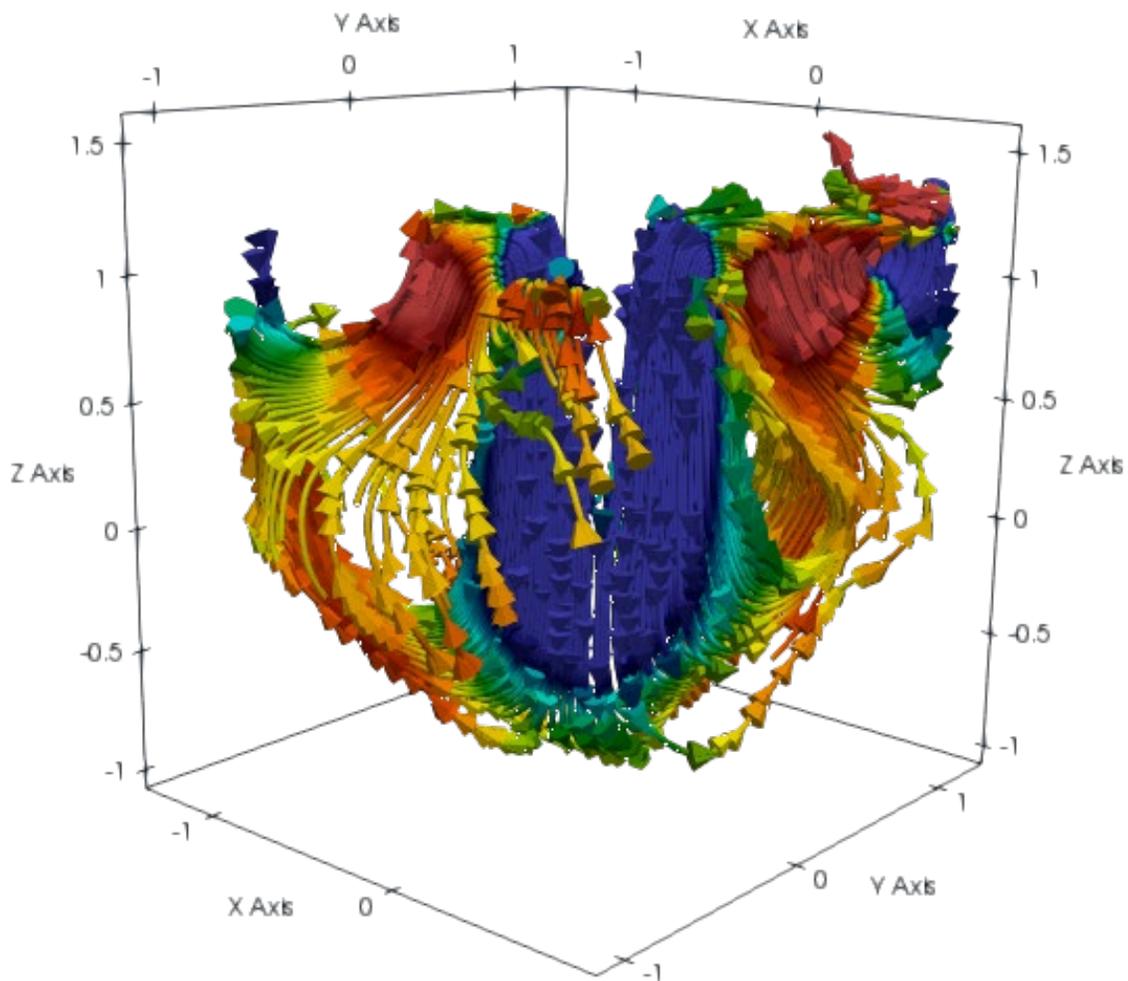
*Figura 24.* Visualización del campo vectorial del clarificador secundario con isolíneas de flujo en ParaView.

Conforme al campo vectorial del flujo en los costados del tanque (Figura 25), se observa la presencia de velocidades ascendentes con una distribución de velocidades medias (0.01 m/s) en la zona baja del tanque, velocidades medias bajas ( $\pm 0.005$  m/s) en la zona media y un aumento hacia la velocidad máxima (0.02 m/s) en la zona superior de separación y clarificación.



*Figura 25.* Visualización del campo vectorial lateral del clarificador secundario con isolíneas de flujo en ParaView.

La representación tridimensional del campo vectorial de la velocidad (Figura 26) indica que, en la zona superior de separación y clarificación del tanque el flujo muestra una tendencia direccional hacia la parte media entre los costados del tanque y la tubería de alimentación, concentrando las velocidades más elevadas de ascenso en los últimos 60 cm de altura del clarificador secundario, de donde se infiere que a partir de tal punto espacial se posibilita el desprendimiento del flujo hacia la salida y el continuo trayecto hacia el fondo del tanque, cumpliendo el circuito de flujo de un clarificador secundario circular de flujo vertical.



*Figura 26.* Visualización de dirección y distribución de velocidades del clarificador secundario en ParaView.

### 4.3. Distribución del circuito de flujo

El seguimiento de partículas considera toda la trayectoria del flujo desde su entrada hasta su salida del sistema. Teniendo en cuenta que el clarificador secundario presenta un flujo vertical ascendente y descendente, se establecieron seis tramos que representan la dirección del flujo con sus respectivas velocidades y las distancias que alcanzan dentro del tanque, logrando establecer el circuito hidráulico completo del sistema en el componente “z”. Además de predecir la distribución del flujo, se indican las zonas muertas representadas por las velocidades cero dentro del tanque en donde se infiere que el flujo no circula dado que las zonas muertas son la brecha que separa al flujo ascendente y descendente. A continuación, se presenta la esquematización del trayecto hidráulico con la simbología correspondiente a las velocidades de simulación.

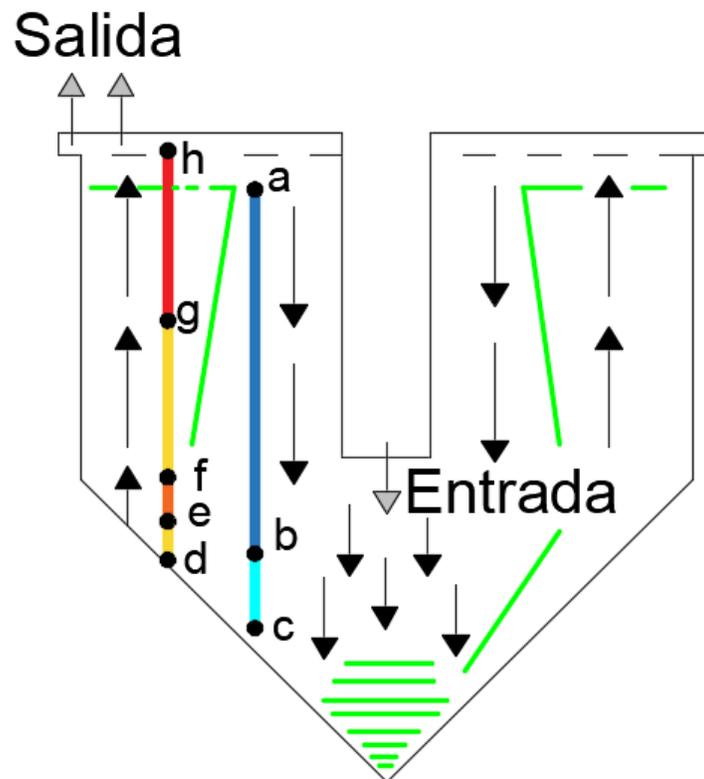


Figura 27. Esquema del circuito de flujo y zonas muertas del clarificador secundario.

Tabla 17

*Tramos del circuito hidráulico.*

Tramos	Distancia (m)	Velocidad (m/s)
a-b	1.9	0.02
b-c	0.2	0.01
d-e	0.3	0.005
e-f	0.45	0.01
f-g	0.8	0.005
g-h	0.75	0.02

La distribución estratificada de las velocidades y su alcance longitudinal se determinaron individualmente para obtener el tiempo correspondiente de cada tramo y lograr establecer una estimación del tiempo de residencia real del flujo en el clarificador secundario.

## 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1 Conclusiones

La generación del mallado computacional inicial realizado en SALOME contemplo un total de 2930 nodos que representaron a la geometría del clarificador secundario de la PTAR UDLAPARK con una finura moderada sin errores, la cual permitió la aplicación de un procesamiento posterior de refinamiento y ajuste de malla en donde la ejecución de “snappyHexMesh” logró transformar los volúmenes tetraédricos en volúmenes hexaédricos, que en CFD se considera una malla de buen calidad,

puesto que la composición de una malla tridimensional hexaédrica proporciona una mayor compatibilidad con los algoritmos de solución para que la simulación represente el estado de las condiciones de borde lo más cercano a la realidad.

Para las condiciones de borde del caso de estudio se consideraron principalmente los escenarios de presión y velocidad a las que se somete el fluido, así es como se estableció la región “*atmosphere*” para el rebosadero (Figura 15), obteniendo una correcta definición de sistema abierto para la tapa de la estructura, en donde el flujo entra en contacto con el aire y los parámetros de presión apuntan valores directos en relación con la presión total del sistema. En la región que comprende a todas las paredes de la estructura “*walls*” (Figura 16), fue muy eficaz definir un límite de presión de flujo igual a cero y una condición de antideslizamiento para que la velocidad del fluido sea de cero en relación con el límite sólido, de modo que dicha condición determino la homogeneidad del fluido cerca del límite y evitó en lo posible fluctuaciones falsas en la región. Las condiciones de entrada del sistema contemplado por la región “*inlet*” representaron correctamente un modelo despresurizado con una velocidad de flujo de 0.004 m/s al ingreso del sistema, de modo que se proporcionaron valores coherentes con relación a las velocidades y su distribución dentro del sistema (Figura 23), ya que el rango de velocidades (Figura 21) y su dirección se encuentra dentro de un modelo teórico para realizar los procesos de sedimentación y clarificación. Por último, el condicionamiento de un modelo no presurizado que incluyo la determinación de la velocidad de salida por la orientación del flujo de agua en la región “*outlet*” fue el indicado para lograr que la simulación representara un sistema en estado estacionario.

El algoritmo “PIMPLE” respondió sin error alguno a los cálculos de solución iterativa para el caso de estudio, permitiendo ejecutar con éxito la simulación por 300 segundos, en donde se pudo observar que el algoritmo de solución alcanzo

adecuadamente la convergencia numérica por medio de la normalización de los residuales a partir de los 225 segundos (Figura 20), demostrando que el sistema entró en estado estacionario, lo cual fue un indicador muy importante que determinó la calidad de simulación y confiabilidad en los datos resultantes que representaron apropiadamente el trayecto hidráulico (Figura 26).

Se logró estimar el tiempo de residencia real del clarificador secundario de la PTAR UDLAPARK a través del planteamiento de la distribución del circuito de flujo (figura 27) en donde se realizó un seguimiento de partícula teórico y se obtuvo una estratificación definida por los rangos de velocidades y su alcance longitudinal, determinando el tiempo específico para cada sección trazada por el flujo de donde resulto un tiempo de 417.5 segundos es decir 6 minutos con 57 segundos estimados que la partícula de agua permanece dentro del sistema, no obstante, dicho cálculo solo contempla la carga hidráulica y no se ve afectado por la carga de sólidos que presenta el clarificador secundario.

## **5.2 Recomendaciones**

Se debería realizar otra simulación con un programa distinto para obtener resultados de otra fuente y poder compararlos computacionalmente con la finalidad de verificar los márgenes de error entre ambas simulaciones y determinar la aceptabilidad de los resultados.

Realizar una verificación y comparación de los planos de construcción del clarificador secundario con la estructura física que se encuentra en la PTAR UDLAPARK con el fin de obtener un dimensionamiento más preciso de la estructura, de manera que se pueda mejorar en lo posible la construcción de la geometría y el

desarrollo de la malla computacional para obtener una mejor simulación con valores más acercados a la realidad.

Se recomienda medir el tiempo de residencia real del clarificador secundario de la PTAR UDLAPARK utilizando trazadores hidráulicos que caractericen experimentalmente al flujo, ya que es muy importante comparar los resultados del comportamiento hidráulico con el modelo teórico desarrollado para poder identificar y solucionar problemas del circuito de flujo, además, se podría llegar a una estimación más real de la eficiencia del sistema y contemplar los errores de diseño de la estructura, ya que se ha verificado que el clarificador secundario funciona correctamente en relación al comportamiento hidráulico, presentando un campo de flujo que evidencia que no hay resuspensión de partículas.

## REFERENCIAS

- Amy, G., Brdjanovic, D., Comeau, Y., Ekama, G. A., Orozco Garcia, J. H., Gerba, C. P., ... Zeeman, G. (2017). Tratamiento biológico de aguas residuales. In C. M. López Vázquez, G. Buitrón Méndez, H. García, & F. J. Cervantes Carrillo (Eds.), *Ingeniería Química* (2017 IWA p, Vol. 33).  
<https://doi.org/10.17345/9788484241027>
- Bergadà Grañó, J. M. (2012). *Mecánica de fluidos : breve introducción teórica con problemas resueltos* (1ra ed.). Retrieved from  
[http://cataleg.upc.edu/record=b1498990~S1\\*cat%0Ahttp://upcommons.upc.edu/handle/2099.3/36611](http://cataleg.upc.edu/record=b1498990~S1*cat%0Ahttp://upcommons.upc.edu/handle/2099.3/36611)
- Burroughs, R., & Burroughs, R. (2011). Wastewater. In *Coastal Governance* (2nd ed.). [https://doi.org/10.5822/978-1-61091-016-3\\_3](https://doi.org/10.5822/978-1-61091-016-3_3)
- Das, S., Bai, H., Wu, C., Kao, J. H., Barney, B., Kidd, M., & Kuettel, M. (2016). Improving the performance of industrial clarifiers using three-dimensional computational fluid dynamics. *Engineering Applications of Computational Fluid Mechanics*, 10(1), 130–144. <https://doi.org/10.1080/19942060.2015.1121518>
- De Clercq, B., & Vanrolleghem, P. (2002). Computational fluid dynamics in wastewater treatment. *Ph.D. Thesis*, Vol. 67, pp. 15–18. Retrieved from  
<http://modeleau.fsg.ulaval.ca/fileadmin/modeleau/documents/Publications/pvr397.pdf>
- Do-Quang, Z., Cockx, A., Liné, A., & Roustan, M. (1998). Computational fluid dynamics applied to water and wastewater treatment facility modeling. *Environmental Engineering and Policy*, 1(3), 137–147.  
<https://doi.org/10.1007/s100220050015>
- Farahani, H. (2018). Paraview Documentation. *ResearchGate*, (October), 0–11.  
<https://doi.org/10.13140/RG.2.2.23268.63367>

- Ferziger, J. H., & Peric, M. (2002). *Computational Methods for Fluid Dynamics* (third, rev; J. H. Ferziger & M. Peric, Eds.). Berlin: Springer.
- Findikakis, A. (2016, February). *THE CONTRIBUTION OF ADVANCED SIMULATION TOOLS TO THE HYDRAULIC DESIGN OF WATER AND WASTEWATER TREATMENT FACILITIES*. (2), 2016. Retrieved from [https://www.iahr.org/PDF/Journals/Hydrolink/2016 free access/editorial 2.pdf](https://www.iahr.org/PDF/Journals/Hydrolink/2016%20free%20access/editorial%202.pdf)
- Findikakis, A. (2018). HYDRAULIC ENGINEERING IN THE SERVICE OF LEISURE ACTIVITIES. *HYDRAULIC ENGINEERING IN THE SERVICE OF LEISURE ACTIVITIES*, 34.
- Fonseca, P., Marques, N., Azevedo, J., & Rodrigues, V. (2014). Large Structures CFD Simulations For Water & Wastewater Treatment Plants CFD modelling Strategy Examples : In blueCAPE (Ed.), *STAR Global Conference* (pp. 1–26). Vienna: blueCape.
- Gomez, J. (2018). STUDY OF THE APPLICATION OF CFD IN THE MULTIPHASE FLOW SEPARATION. *ResearchGate*, (June). <https://doi.org/10.13140/RG.2.2.14412.92805>
- Greenshields, C. J. (2019). *OpenFOAM* (p. 237). p. 237. Londres: The OpenFOAM Foundation.
- Haddadi, B., Jordan, C., Nagy, J., Gößnitzer, C., Natarajan, V., Zibuschka, S., & Harasek, M. (2018). *OpenFOAM® Basic Training* (4th ed.; B. Haddadi, C. Jordan, & M. Harasek, Eds.). Retrieved from [www.youtube.com/channel/UCjdgpuxuAxH9BqheyE82Vvw%0Awww.youtube.com/channel/UCjdgpuxuAxH9BqheyE82Vvw%0Awww.OpenFOAM.org](http://www.youtube.com/channel/UCjdgpuxuAxH9BqheyE82Vvw%0Awww.youtube.com/channel/UCjdgpuxuAxH9BqheyE82Vvw%0Awww.OpenFOAM.org)
- Haider, J. (2018). *OpenFOAM course for beginners* (p. 135). p. 135. Retrieved from <https://www.jibranhaider.weebly.com>
- Holzinger, G. (2020). *OpenFOAM A little User-Manual* (pp. 1–316). pp. 1–316. Linz: ResearchGate.

- INEC. (2016a). *Documento Técnico “Estadística Ambiental Económica en Gobiernos Autónomos Descentralizados Municipales” Gestión de Agua Potable y Alcantarillado 2016*. Retrieved from [http://www.ecuadorencifras.gob.ec/documentos/web-inec/Encuestas\\_Ambientales/Municipios\\_2016/Documento tecnico APA 2016.pdf](http://www.ecuadorencifras.gob.ec/documentos/web-inec/Encuestas_Ambientales/Municipios_2016/Documento_tecnico_APA_2016.pdf)
- INEC. (2016b). *ESTADÍSTICA DE INFORMACIÓN AMBIENTAL ECONÓMICA EN GOBIERNOS AUTÓNOMOS DESCENTRALIZADOS MUNICIPALES 2015 (AGUA Y ALCANTARILLADO)* (Vol. 2015). Retrieved from [http://www.ecuadorencifras.gob.ec/documentos/web-inec/Encuestas\\_Ambientales/Municipios\\_2015/Documento\\_Tecnico-Gestion\\_de\\_Agua\\_y\\_Alcantarillado\\_2015.pdf](http://www.ecuadorencifras.gob.ec/documentos/web-inec/Encuestas_Ambientales/Municipios_2015/Documento_Tecnico-Gestion_de_Agua_y_Alcantarillado_2015.pdf)
- Jayanti, S. (2018). Computational Fluid Dynamics for Engineers and Scientists. In *Computational Fluid Dynamics for Engineers and Scientists*. <https://doi.org/10.1007/978-94-024-1217-8>
- Jeppsson, U., & Diehl, S. (1996). On the modelling of the dynamic propagation of biological components in the secondary clarifier. *Elsevier Science Ltd*, 34(5-6-6 pt 3), 85–92. [https://doi.org/10.1016/0273-1223\(96\)00632-4](https://doi.org/10.1016/0273-1223(96)00632-4)
- Liscia, S., López, A., Ruiz Diaz Cordoso, A., & De Dios, M. (2018). Interacción entre modelación física y numérica para el diseño de obras hidráulicas y validación de simulaciones CFD. *Rev. Soc. Cient. Paraguay*, 23, 227–240.
- Lozano, W. A. (2012). Diseño de Plantas de Tratamiento de Aguas Residuales. In *ResearchGate*. Retrieved from [298354134\\_Disenio\\_de\\_Plantas\\_de\\_Tratamiento\\_de\\_Aguas\\_Residuales](https://www.researchgate.net/publication/298354134_Disenio_de_Plantas_de_Tratamiento_de_Aguas_Residuales)
- Makisha, N., & Kulakov, A. (2017). Peculiarities of clarifiers' reconstruction at waste water treatment plants. *MATEC Web of Conferences*, 112, 10018. <https://doi.org/10.1051/mateconf/201711210018>

- Menéndez Gutiérrez, C., & Garcia Zumalacarregui, J. (2013). Plantas de tratamiento de aguas residuales. *ResearchGate*, (November 2015), 1–51.
- Moreland, K. (2014). The ParaView Tutorial. *Sandia National Laboratories*, p. 136. U. S.: Sandia National Laboratories.
- Moukalled, F., Mangani, L., & Darwish, M. (2016). The finite volume method in computational fluid dynamics. In *Fluid Mechanics and its Applications* (Vol. 113). [https://doi.org/10.1007/978-3-319-16874-6\\_21](https://doi.org/10.1007/978-3-319-16874-6_21)
- Novak, P., Guinot, V., Jeffrey, A., & Reeve, D. E. (2018). Hydraulic Modelling – an Introduction. In Taylor & Francis (Ed.), *Hydraulic Modelling – an Introduction* (1st Editio). <https://doi.org/10.1201/9781315272498>
- Raghuvanshi, S., Bhakar, V., Sowmya, C., & Sangwan, K. S. (2017). Waste Water Treatment Plant Life Cycle Assessment: Treatment Process to Reuse of Water. *Procedia CIRP*, 61, 761–766.  
<https://doi.org/10.1016/j.procir.2016.11.170>
- Richmond, G. (2019). *Modelos de turbulencia introductorio*. 1–16.  
<https://doi.org/10.13140/RG.2.2.21060.17289>
- Robertson, E., Choudhury, V., Bhushan, S., & Walters, D. K. (2015). Validation of OpenFOAM numerical methods and turbulence models for incompressible bluff body flows. *Computers and Fluids*, 123, 122–145.  
<https://doi.org/10.1016/j.compfluid.2015.09.010>
- Sadrehaghighi, I. (2019). *Essentials of CFD*.  
<https://doi.org/10.13140/RG.2.2.26586.44489/3>
- Samstag, R. W., Ducoste, J. J., Griborio, A., Nopens, I., Batstone, D. J., Wicks, J. D., ... Laurent, J. (2016). CFD for wastewater treatment: An overview. *Water Science and Technology*, 74(3), 549–563.  
<https://doi.org/10.2166/wst.2016.249>

- Schöberl, J., Egger, H., Becirovic, A., & Huber, M. (2007). *Netgen / NGSolve : Mesh Generator and Finite Element Solver*. Oslo.
- Ting, X., Miedema, S. A., & Xiuhan, C. (2019). Comparative analysis between CFD model and DHLLDV model in fully-suspended slurry flow. *Ocean Engineering*, 181(February), 29–42. <https://doi.org/10.1016/j.oceaneng.2019.03.065>
- Towara, M., Schanen, M., & Naumann, U. (2015). MPI-parallel discrete adjoint OpenFOAM. *Procedia Computer Science*, 51(1), 19–28. <https://doi.org/10.1016/j.procs.2015.05.181>
- Venier, C. M., Pairetti, C. I., Damian, S. M., & Nigro, N. M. (2017). On the stability analysis of the PISO algorithm on collocated grids. *Computers and Fluids*, 147, 25–40. <https://doi.org/10.1016/j.compfluid.2017.01.017>
- WEF. (2010). DESIGN OF MUNICIPAL WASTEWATER TREATMENT PLANTS. In American Society of Civil Engineers & Environmental and Water Resources Institute (Eds.), *Design of Municipal Wastewater Treatment Plants MOP 8, Fifth Edition* (Fifth Edit). Alexandria, Virginia: Mc Graw Hill.
- Weiss, M., Plósz, B. G., Essemiani, K., & Meinhold, J. (2007). Suction-lift sludge removal and non-Newtonian flow behaviour in circular secondary clarifiers: Numerical modelling and measurements. *Chemical Engineering Journal*, 132(1–3), 241–255. <https://doi.org/10.1016/j.cej.2007.01.004>
- Xiang, Y., Yu, B., Yuan, Q., & Sun, D. (2017). GPU Acceleration of CFD Algorithm: HSMAC and SIMPLE. *Procedia Computer Science*, 108, 1982–1989. <https://doi.org/10.1016/j.procs.2017.05.124>

