



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

SIMULACIÓN DE DINÁMICA DE FLUIDOS COMPUTACIONAL (CFD) DE
LA FASE PRELIMINAR DE LA PLANTA PILOTO DE TRATAMIENTO DE
AGUAS RESIDUALES DE UDLAPARK

AUTOR

Andrea Cristina Parra Cevallos

AÑO

2020



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

SIMULACIÓN DE DINÁMICA DE FLUIDOS COMPUTACIONAL (CFD) DE LA FASE
PRELIMINAR DE LA PLANTA PILOTO DE TRATAMIENTO DE AGUAS RESIDUALES
DE UDLAPARK

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

Profesor Guía

M.Sc. Santiago Daniel Piedra Burgos

Autora

Andrea Cristina Parra Cevallos

Año

2020

DECLARACIÓN DEL PROFESOR GUÍA

"Declaro haber dirigido el trabajo, Simulación de dinámica de fluidos computacional (CFD) de la fase preliminar de la Planta Piloto de Tratamiento de Aguas Residuales de Udlapark, a través de reuniones periódicas con el estudiante Andrea Cristina Parra Cevallos, en el semestre 202020, 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"



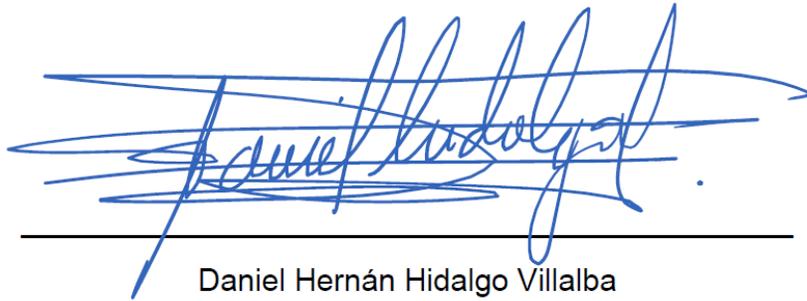
Santiago Daniel Piedra Burgos

Master of Science in Hydro Science and Engineering

CI.1715384150

DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, Simulación de dinámica de fluidos computacional (CFD) de la fase preliminar de la Planta Piloto de Tratamiento de Aguas Residuales de Udlapark, del Andrea Cristina Parra Cevallos, en el semestre 202020, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".



Daniel Hernán Hidalgo Villalba

Magister en Ciencias Especialización Ingeniería Ambiental

CI.1801914449

DECLARACIÓN DE AUTORÍA 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.”



Andrea Cristina Parra Cevallos

CI.1723228019

AGRADECIMIENTO

A mi familia, por ser un soporte para culminar esta meta.

A mis padres, Lucía y José por siempre confiar en mí.

A mis hermanos, Fernanda, Esteban y mis cuñados por su ejemplo y apoyo incondicional; Meri por cuidarme.

A mi tutor, Ing. Santiago Piedra y a mi docente corrector

Ing. Daniel Hidalgo por compartir sus conocimientos y por toda la motivación durante este proceso.

A mis demás profesores por su dedicación y cariño.

A mis amigas y primos, por darme ánimos para seguir y compartir esta experiencia. Finalmente, a todas las personas estuvieron ahí apoyándome para que este sueño se haga realidad.

DEDICATORIA

Dedico este trabajo a mi madre Lucía y mi hermana Fernanda
por enseñarme a no rendirme y porque a pesar
de la distancia siempre estamos juntas.
Finalmente, a mi sobrina Valentina con mucho amor.

RESUMEN

Hoy en día, uno de los principales desafíos en el área de la depuración de aguas residuales es el diseño exitoso y una operación compleja en los procesos de tratamiento, las cuales deben garantizar altas eficiencias de tratamiento para cumplir los criterios de calidad de con los efluentes. La dinámica de fluidos computacional (CFD) se utiliza para predecir el comportamiento e interacciones entre los diversos procesos de tratamiento para que el diseño ayude a cumplir con los objetivos previstos.

En el presente estudio, se realizó la simulación del comportamiento hidráulico de la fase preliminar de la Planta Piloto de Aguas Residuales de Udlapark que está compuesta por una canaleta parshall y desarenador de flujo horizontal, utilizando el software de acceso libre OpenFoam empleando el solucionador “interFoam”. La geometría de los casos fue definida de acuerdo con especificaciones de estas unidades ya existentes.

Posteriormente, para la refinación del mallado se utilizó la herramienta “snappyHexMesh” que ayuda en la estabilidad y en la resolución de los algoritmos de solución. Además, se especificó las condiciones de borde en base a las características de cada uno de los casos.

Finalmente, el post-procesamiento de los datos se realizó mediante el software ParaView a partir de donde se obtuvo la interfase gráfica lo cual determina el comportamiento hidráulico, los vectores de velocidad y la estimación del tiempo de retención.

ABSTRACT

Nowadays, one of the main challenges in wastewater sector is the successful design and complex operation of the treatment processes, which must have high treatment efficiencies to meet the effluent quality criteria. Computational Fluid Dynamics (CFD) is used to predict behavior and interactions between various treatment processes, thereby helping design objectives meet their goal.

In the present study was carried out the simulation of the hydraulic behavior of the preliminary phase of the Udlapark Wastewater Treatment Plan (WTP), which is composed of a parshall flume and a horizontal flow desander, the same one used by free access software "OpenFoam" by using the "interFoam" solver. The geometry of the cases was defined according to the specifications of these units already determined. Subsequently, the "snappyHexMesh" tool was used to refine the mesh, which helps in the stability and resolution of the solution algorithms. In addition, the boundary conditions were specified according to the characteristics of each of the cases.

Finally, the post-processing of the data was carried out using the "ParaView" software from which the simulation was obtained. Where it was possible to determine the hydraulic behavior, the speed vectors, and the estimation of the retention time.

ÍNDICE

1. INTRODUCCIÓN	1
1.1 Antecedentes	1
1.2 Problemática	2
1.3 Objetivos	3
1.3.1 Objetivo General	3
1.3.2 Objetivos Específicos	4
1.4 Alcance	4
1.5 Justificación	4
2. MARCO TEÓRICO	6
2.1 Aguas Residuales	6
2.2 Clasificación de Aguas Residuales	6
2.3 Tratamiento de aguas residuales	7
2.3.1 Tipos de tratamientos de aguas residuales	7
2.3.2 Operaciones físicas en tratamiento preliminar	8
2.3.2.1 Rejillas	8
2.3.2.2 Canal Parshall	8
2.3.2.3 Partes canal parshall	9
2.3.2.4 Desarenador horizontal	10
2.3.2.5 Vertedero Sutro	11
2.3.2.6 Velocidad de Manning	12

2.4 Computational Fluids Dynamics (CFD's)	12
2.4.1 Estudio de interacción de fluidos	13
2.4.2 Preprocesamiento.....	13
2.4.2.1 Salome.....	14
2.4.3 Solver	14
2.4.3.1 Ecuaciones gobernantes	14
2.5 Open Foam.....	15
2.5.1 Solver InterFoam	16
2.5.2 Método de Volumen Fluido (VOF)	16
2.5.1.1 Ecuación de continuidad.....	17
2.5.1.2 Ecuación de momento	17
2.5.1.3 Ecuación de interfase	18
2.5.2 Algoritmo de solución	18
2.5.3 Estructura caso OpenFoam	19
2.5.3.1 Directorios Requeridos	20
2.5.3.1.1 Archivos principales directorio “system”.....	21
2.5.3.2 Control tiempo de simulación.....	22
2.5.3.3 Carpetas obtenidas de simulación.....	23
2.5.4 Descripción de utilidades ejecutadas.....	23
2.6 Unidades y dimensiones.....	24
2.7 Condiciones de borde	25
2.7.1 Tipos de parches	25
2.7.2 Tipos de entrada:.....	26
2.8 Generación de Mallado.....	26

2.8.1 Descripción mallado	26
2.8.1.1 polyMesh	27
2.8.2 Forma de celdas	27
2.8.3 Generación de volumen de control	29
2.8.4 Refinamiento de mallado	31
2.9 Número de Courant	32
2.10 Post-procesamiento	32
3. METODOLOGÍA.....	33
3.1 Preprocesamiento	34
3.1.1 Geometría.....	34
3.1.2.1 Definición de grupos	38
3.1.3 Generación de mallado.....	39
3.1.3.1 Generación de archivos UNV y STL	41
3.1.4 blockMesh.....	41
3.1.5 snappyHexMesh	43
3.2 Procesamiento	44
3.2.1 Condiciones de borde	45
3.2.1.1 Directorio “0”	45
3.2.1.2 Directorio “system”	47
3.2.2 Simulación en paralelo	48
3.2.3 Esquemas numéricos	49
3.2.3.1 fvSchemes.....	49
3.2.4 Control y solución de algoritmos.....	51
3.2.4.1 fvSolution	51

3.2.5 Residuales.....	53
3.3 Post-Procesamiento.....	54
4. RESULTADOS Y ANÁLISIS	54
4.1 Residuales	54
4.2 Simulación canaleta parshall.....	56
4.3 Simulación desarenador.....	58
4.4 Tiempo de residencia.....	61
5. CONCLUSIONES Y RECOMENDACIONES.....	63
5.1 Conclusiones	63
5.2 Recomendaciones.....	64
REFERENCIAS.....	65

NOMENCLATURA

Símbolo	Unidad	Parámetro
U	m/s	Velocidad en el eje x
V	m/s	Velocidad en el eje y
W	m/s	Velocidad en el eje z
μ	kg/m*s	Viscosidad dinámica
p	Pa	Presión
g	m/s ²	Gravedad
ρ	kg/m ³	Densidad
t	s	Tiempo
F	kg/m*s	Fuerza
τ_{ii}	-	Tensores de viscosidad y turbulencia
$f\sigma_i$	N/m	Tensión superficial
∇	-	Nabla
V_m	m/s	Velocidad de Manning
R	m	Radio hidráulico
N	-	Factor de rugosidad de Manning
S	-	Pendiente

1. INTRODUCCIÓN

1.1 Antecedentes

La evolución del campo del tratamiento de aguas residuales concierne directamente con aspectos de salud, saneamiento y cuidado del medio ambiente, especialmente en el constante crecimiento de los centros urbanos. Los métodos de tratamiento de efluentes se fueron desarrollando en respuesta a la preocupación por la salud pública y las condiciones desfavorables causadas por las descargas de aguas residuales no tratadas a los cuerpos hídricos (Metcalf and Eddy, 2014). Durante el siglo XX, las diferentes unidades de tratamiento y conducción tanto de agua potable como residuales eran efectuadas a partir de cálculos hidráulicos simples; lo que puede provocar distintos problemas en su operación, desencadenando una vida útil de altos costos de operación y mantenimiento, donde se evidencia la importancia de las herramientas técnicas de simulación que se están utilizando a nivel mundial (Metcalf and Eddy, 2014).

El desarrollo de la simulación mediante dinámica de fluidos computacional (Computational Fluid Dynamics- CFD) comenzó con la llegada de la computadora digital a partir del año 1950. Inicialmente, se empleó en la aeronáutica y la industria aeroespacial convirtiéndose en una herramienta esencial y a su vez permitiéndole expandirse hacia otras industrias como la automotriz, la generación de energía, etc. (Moukalled, Mangani, y Darwish, 2016).

Dentro de la ingeniería ambiental, la simulación CFD es aplicada en el aprovechamiento de energías renovables como una forma rentable de simulación del funcionamiento de turbinas eólicas y paneles solares (Khandve, 2016).

Adicionalmente, es ampliamente utilizada para el análisis y resolución de problemas hidráulicos en el tratamiento de aguas residuales. Con la ayuda de esta tecnología, de simulación CFD se ha convertido en una alternativa para el análisis de fluidos donde se puede visualizar los distintos componentes que son partes de los sistemas de control y movimiento de los fluidos en estado estacionario o turbulento (Tič y Lovrec, 2012). Sin embargo, todavía es necesario fomentar el empleo de esta tecnología para el manejo de procesos físicos, químicos y biológicos en las plantas de tratamiento de aguas residuales (PTAR); dado que el sistema de depuración es complejo y todos los componentes individuales deben trabajar bajo estricta sincronía para lograr los objetivos de rendimiento, lo cual depende principalmente de un diseño adecuado y la optimización de flujos dinámicos, niveles de agua y tiempo de residencia en los distintos tratamientos (Samstag et al., 2016).

El desarrollo continuo y el aumento de la tecnología ha permitido perfeccionar la codificación para una simulación CFD, integrando con mejores interfaces para el usuario, lo que ha llevado a que la academia promueva el uso de CFD como una herramienta de diseño común en la ingeniería.

1.2 Problemática

El crecimiento demográfico ligado al desarrollo de la industria, así como también al uso de nuevas tecnologías de extracción de recursos naturales han incrementado evidentemente el aporte y el nivel de contaminación de los vertidos líquidos, en su mayoría sin ningún tipo de tratamiento previo a su descarga a los sistemas hídricos, desencadenando distintos impactos ambientales (Organización de Naciones Unidas [ONU], 2018).

En el Ecuador, solo el 61,88% de las aguas residuales cuenta con un sistema de depuración. Actualmente el país tiene 421 PTAR municipales donde el 49,88% se encuentran en la Región Sierra, el 30,64% en la Región Costa, 19 % en la Región

Amazónica y el 0,48% se encuentran en la Región Insular, cifras determinadas en el año 2015, sin embargo, no existen datos estadísticos sobre si estas tienen una correcta operación y control de las mismas (INEC, 2016).

En ocasiones, los gobiernos no reconocen la necesidad de adoptar medidas para reducir la contaminación del agua, aumentar la calidad de agua y mejorar significativamente el tratamiento de aguas residuales. Con respecto a la situación del país, el tratamiento de aguas residuales no es una prioridad puesto su infraestructura y operación es costosa, por ende, este tipo de inversiones quedan fuera del alcance de muchos municipios. (Ramón, 2010).

También, una planta de tratamiento de aguas residuales, involucran distintas operaciones unitarias, las cuales se rigen bajo estrictas técnicas experimentales y de diseño con el propósito de controlar los parámetros de funcionamiento, el consumo de potencia, homogeneidad de fluidos, entre otros. Partiendo principalmente de ensayos experimentales los cuales se logran obteniendo correlaciones empíricas mediante la relación entre variables de diseño y parámetros de funcionamiento. Sin embargo, dada la complejidad, la diversidad de resultados de presión y gradientes de velocidad en un análisis experimental, no es suficiente para evaluar detalladamente el comportamiento del fluido bajo distintas condiciones (Chambergo, Valverde, Pachas y Yépez, 2017).

1.3 Objetivos

1.3.1 Objetivo General

- Simular el funcionamiento hidráulico de la fase preliminar de la Planta Piloto de Tratamiento de Aguas Residuales de Udlapark

1.3.2 Objetivos Específicos

- Establecer las condiciones de borde de la fase preliminar de la Planta Piloto de Tratamiento de Aguas Residuales de Udlapark
- Estimar el tiempo de residencia real en la fase preliminar de la Planta Piloto de Tratamiento de Aguas Residuales de Udlapark

1.4 Alcance

En el presente estudio se realizó el modelamiento del comportamiento hidráulico del agua que ingresa a la Planta de Tratamiento de Aguas Residuales Piloto de Udlapark en su fase preliminar. Inicialmente, se efectuó la descripción de las condiciones de borde iniciales del sistema, lo que incluye la geometría correspondiente a las distintas operaciones unitarias de la fase preliminar con la ayuda del software Salome para el pre-procesamiento del caso. Posteriormente, se ejecutó la simulación CFD con el software Open Foam (Open Source Field Operation and Manipulation) para lograr en el estudio de la interpretación de los datos del fluido y determinar el tiempo de residencia.

1.5 Justificación

Algunos de los problemas tecnológicos más importantes que afectan globalmente a la humanidad son el abastecimiento de la energía y de agua de calidad en cantidades suficientes para el bienestar y desarrollo. La energía proveniente mayoritariamente de fuentes no renovables y por lo tanto limitadas, provocan graves problemas ambientales. Actualmente, estas energías son claramente insuficientes para abastecer a todos los países, sobre todo si se tiene en cuenta la creciente

demanda de las economías emergentes. En el caso de los recursos hídricos mayormente utilizados hoy en día, son renovables, a diferencia de los recursos energéticos que provienen del ciclo natural del agua. Con el aumento de la escasez de agua y la presión sobre los recursos hay una mayor probabilidad de conflictos y problemas de contaminación que originan complicaciones en la salud de las personas, por lo cual es importante brindar un tratamiento adecuado a las aguas residuales (Herrerias, 2013).

La simulación CFD es una combinación de física y tecnología relacionada con fluidos, aplicaciones de computación, matemáticas y mecánica. Además, es empleada para predecir el comportamiento de fluidos, la transferencia de calor y de masa además de resolver reacciones químicas y ecuaciones matemáticas complejas que gobiernan estos procesos usando un algoritmo numérico, Obteniendo como resultado una simulación del comportamiento de un fluido bajo condiciones establecidas. Dentro de la ingeniería es utilizada principalmente en estudios conceptuales de nueva tecnología, resolución de problemas y rediseño de unidades de tratamiento. También es utilizada como alternativa en lugar de métodos experimentales, reduciendo costo y tiempo requerido en los procesos de diseño. La percepción en el proceso de construir un sistema ha cambiado con el análisis de CFD que permite tener una visión del funcionamiento y en qué condiciones puede desarrollarse el proyecto; evaluando distintos parámetros en muy poco tiempo hasta alcanzar un resultado ideal. Garantizando eficiencia y logrando corregir errores de funcionamiento (Raman, Dewang y Raghuwanshi, 2018).

2. MARCO TEÓRICO

2.1 Aguas Residuales

Son aguas que han cambiado sus propiedades físicas, químicas y biológicas como resultado de su uso en actividades antrópicas (Amotey y Bani, 2011).

2.2 Clasificación de las Aguas Residuales

Las aguas residuales se clasifican de la siguiente manera:

Domésticas: son el resultado del metabolismo de las personas, la utilización de agua de las actividades tanto de viviendas como comercios y escurrería que llega a la red de alcantarillado.

Industriales: son efluentes de distinta composición generados a partir de diferentes industrias (agropecuarias, minería, textiles, petróleo, farmacéuticas, entre otras...)

Municipales/Urbanas: Son la combinación de aguas utilizadas por actividades antrópicas de zonas residenciales, comerciales, industrias y aguas lluvia (European Environment Agency [EEA], 2018).

A continuación, se puede observar un diagrama acerca de la clasificación de las aguas residuales:

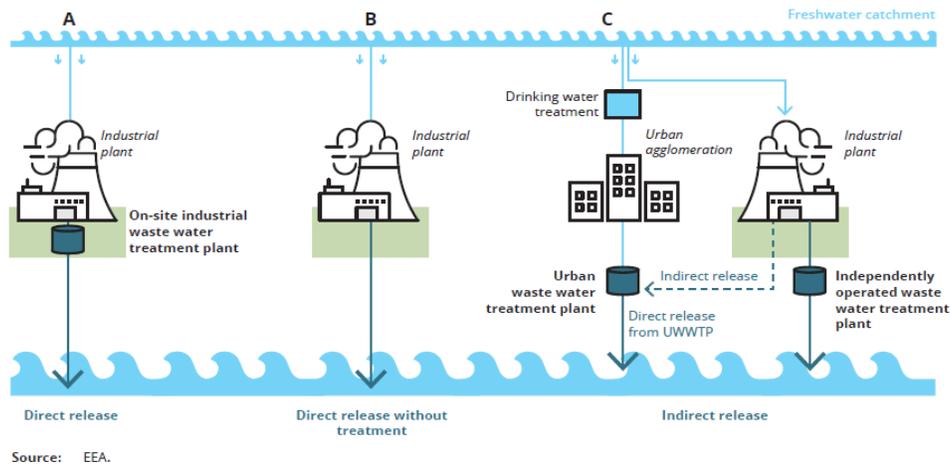


Figura 1. Clasificación de las aguas residuales

Tomado de (EAE, 2018)

2.3 Tratamiento de aguas residuales

Es el proceso y la tecnología que es utilizada para la eliminación de los contaminantes que se encuentran en las aguas residuales, para retornar al medio ambiente y asegurar la salud pública (Amotey y Bani, 2011)

2.3.1 Tipos de tratamientos de aguas residuales

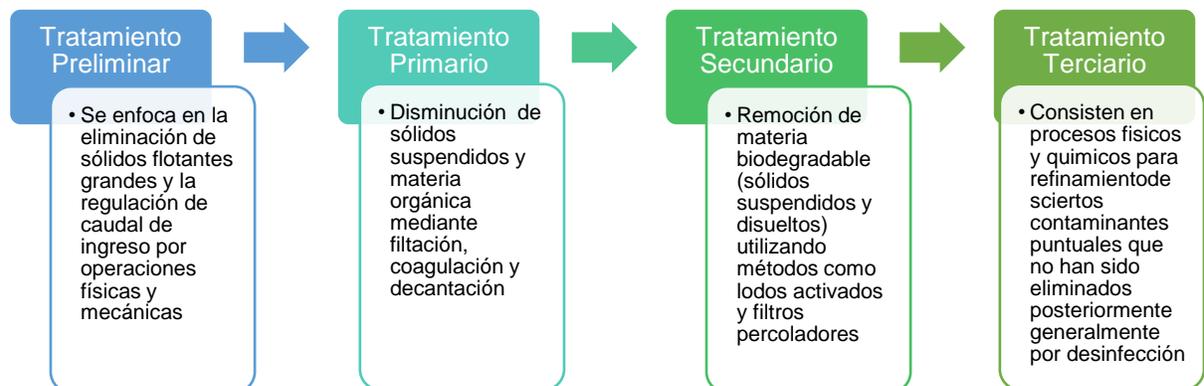


Figura 2. Tipos de tratamientos de aguas residuales

Adaptado de (Metcalf and Eddy, 2014)

2.3.2 Operaciones físicas en tratamiento preliminar

2.3.2.1 Rejillas

Es una operación unitaria física utilizada para retención de sólidos gruesos (basura, ramas, bolsas) que puedan perturbar o dañar el funcionamiento de las unidades posteriores, su mantenimiento puede ser de manera manual o automática (Metcalf and Eddy, 2014)

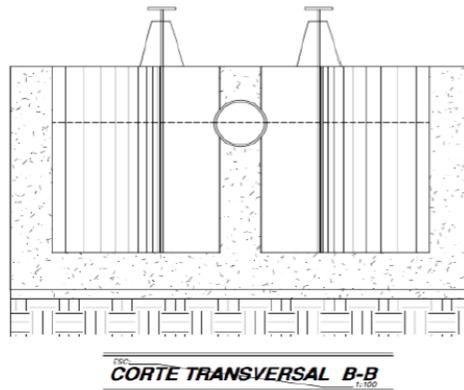


Figura 3. Diseño de rejillas

Tomado de (Fierro, 2018)

2.3.2.2 Canal Parshall

Es una estructura hidráulica que permite medir la cantidad de agua que pasa por una sección transversal de un canal abierto determinado en régimen crítico;

manteniendo un nivel casi constante de agua en la zona inicial y en la garganta se relaciona la altura del agua con la cota del vertedero. Es común en industrias que requieren medir descargas de aguas residuales además de su fácil mantenimiento y su bajo costo económico (González y Hurtado, 2012).

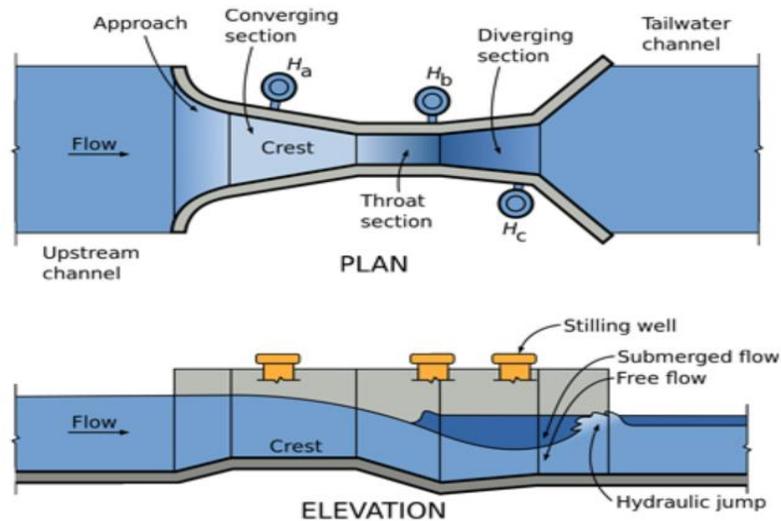


Figura 4. Zonas de Canaleta Parshall

Tomado de (González y Hurtado, 2012)

2.3.2.3 Partes Canaleta Parshall

Consta de las siguientes cuatro partes principales:

1. **Transición de entrada:** la parte inferior se eleva sobre el fondo original del canal con una pendiente suave (relación 1:4) y las paredes laterales se van cerrando en línea rectangular
2. **Sección convergente:** está conformado por paredes paralelas, el fondo es horizontal y el ancho va disminuyendo

3. **Garganta:** la parte inferior vuelve a bajar para terminar con otra pendiente ascendente en la sección divergente
4. **Sección divergente:** zona de salida del caudal donde las paredes se van ensanchando de forma rectangular en la base del canal con una pendiente ligeramente ascendente

(Goyena y Fallis, 2019)

2.3.2.4 Desarenador horizontal

Es el tipo de desarenador más antiguo en utilizado en el tratamiento de aguas residuales; estas unidades se proyectan de modo que la velocidad sea controlada lo más cercana posible a 0,3 m/s y proporcione suficiente tiempo como para que se sedimenten las partículas de arena en el fondo del canal.

La velocidad de diseño permite la circulación a través de la unidad de la mayor parte de la materia orgánica en suspensión de todas aquellas partículas orgánicas hayan sedimentado, pero permite la sedimentación de la arena pesada. Generalmente, los desarenadores se proyectan para eliminar todas las partículas de arena desde 0.21mm a 0.15 mm de diámetro. La longitud del canal vendrá dada por la velocidad de sedimentación y el área de la sección transversal lo cual está relacionado con el caudal; además se toma en cuenta una longitud adicional para que se reduzca la turbulencia que se produce a la entrada y a la salida del canal (Metcalf and Eddy, 2014).

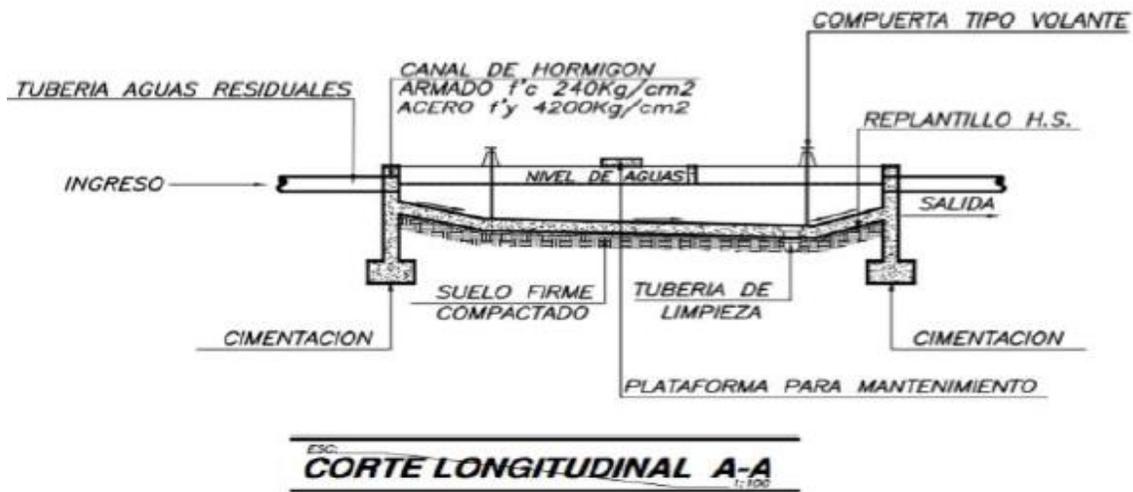


Figura 5. Diseño de desarenador horizontal

Tomado de (Fierro, 2018)

2.3.2.5 Vertedero Sutro

También llamados vertederos proporcionales, son dispositivos utilizados en plantas de tratamiento de aguas residuales, se emplea para la regulación de caudal el cual es proporcional a la carga. Se encuentra compuesto de una sección rectangular adherida a una sección curva, la cual garantiza una proporcionalidad y la velocidad de fluido constante (Juárez, 2017).

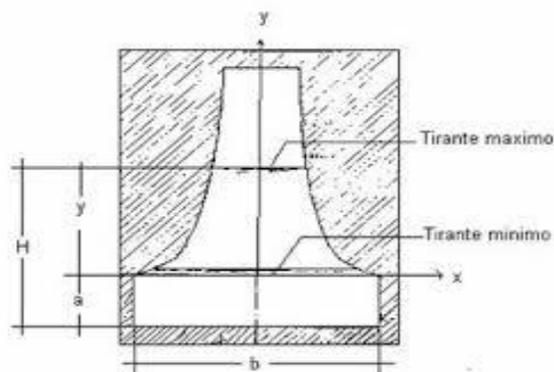


Figura 6. Diseño Vertedero Sutro

Tomado de (Juárez, 2017)

2.3.2.6 Velocidad de Manning

La fórmula de Manning (1895) es aplicada para estimar la velocidad del fluido en canales abiertos (Aldama y Ocón, 2002) como se presenta a continuación:

$$V = \frac{1}{n} R^{\frac{2}{3}} \times S^{\frac{1}{2}}$$

Siendo:

V= velocidad media de la sección transversal (m/s)

n= factor de rugosidad de Manning

R= radio hidráulico

S= pendiente del fondo del canal

2.4 Computational Fluids Dynamics (CFD 's)

También conocido como Dinámica de Fluidos Computacional, es el análisis de la interacción de fluidos en la ingeniería mediante una simulación computacional, utilizando modelamiento matemático (resolución de ecuaciones que gobiernan el movimiento de los fluidos) y métodos numéricos (métodos discretización, solvers, parámetros números y generación de malla) (Zuo, 2015).

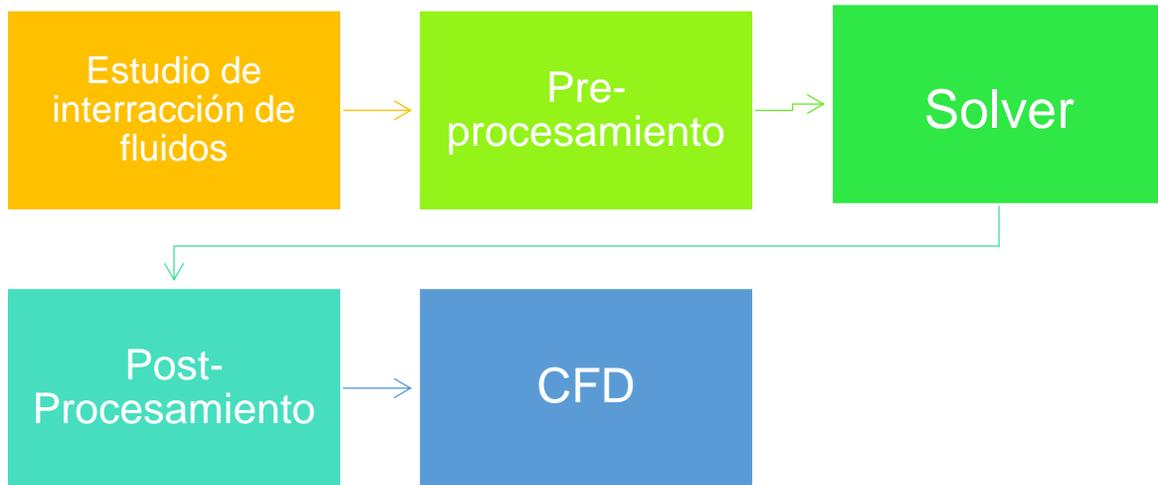


Figura 7. Proceso de dinámica de fluidos computacional

2.4.1 Estudio de interacción de fluidos

Para trasladar un problema de ingeniería a una simulación de dinámica de fluidos computacional (CFD) es necesario conocer las propiedades físicas del fluido y sus interacciones determinadas en el estudio regida por la mecánica de fluidos (Zuo, 2015).

2.4.2 Preprocesamiento

Implica principalmente en la creación de la geometría del caso de estudio utilizando un software para generar el diseño (CAD), siendo uno de los pasos más largos para la simulación de fluidos donde se enfrentan desafíos de complejidad de acuerdo con la geometría. Posteriormente, se puede dar paso a la generación del mallado y configuración de condiciones de borde específicas al caso (Ali, Tyacke, Watson, Tucker y Shahpar, 2019).

2.4.2.1 Salome

Es un software de código abierto, el cual proporciona una plataforma genérica de pre y post procesamiento para simulación numérica. Se basa en una geometría abierta y flexible. Provee un modelamiento numérico de problemas hidráulicos, el desarrollo de soluciones y su integración de procesos de diseño manejado bajo contornos específicos. Facilita la integración de nuevos componentes en sistemas heterogéneos para el cálculo numérico. Además, establece prioridad para el empleo de “solvers multifásicos” mediante una interfase gráfica fácil de utilizar que ayuda en la reducción de costos y ensayos (Salome Platform, 2014).

2.4.3 Solver

Es una técnica de solución numérica la cual requiere alta potencia computacional. Existen múltiples solucionadores disponibles, que varían en eficiencia y capacidad para incorporar la integración de las ecuaciones gobernantes, discretización (conversión de ecuaciones integrales en ecuaciones algebraicas) y resolución de ecuaciones algebraicas. (Raman et al., 2018).

2.4.3.1 Ecuaciones gobernantes

El comportamiento de los fluidos se encuentra regido por leyes físicas que controlan los principios de conservaciones que se traducen en relaciones matemáticas descritas como ecuaciones diferenciales parciales necesarias para una simulación CFD. La ecuación de Navier Stokes expresa los principios de conservación de masa, momento y energía como se presenta a continuación:

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

Inercia

Gradiente

Viscosidad

Fuerza Fluido

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

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

(Moukalled et al., 2016)

(Ecuación 1)

2.5 Open Foam

Para la simulación de este caso se utiliza OpenFoam (Open Source Field And Manipulation) es un software libre perteneciente a una biblioteca C++ (contiene algoritmos, iteradores y funciones) que puede ser utilizada para resolver problemas de mecánica de fluidos y continuidad con un enfoque en la discretización del volumen finito. También incluye varios solucionadores listos, utilidades y aplicaciones que pueden ser usados directamente.

El núcleo de estas bibliotecas consta de un conjunto de objetos que permiten al programador manipular geometrías, generación de mallas y técnicas de discretización a un alto nivel de codificación (Moukalled et al., 2016).

A continuación, se presenta la estructura general de OpenFoam:

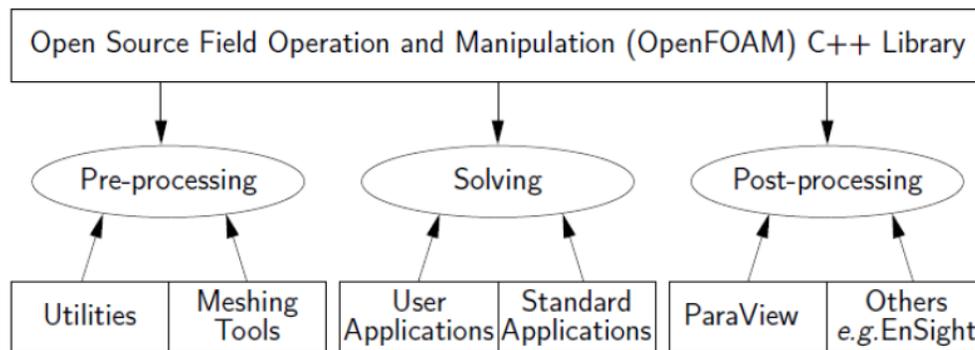


Figura 8. Descripción general de la estructura de OpenFoam

Tomado de (Greenshields, 2019)

2.5.1 Solucionador InterFoam

El módulo InterFoam utiliza el Método de Volúmenes Finitos (VOF) para resolver problemas de flujo transitorio de gases inmiscibles e incompresibles en condiciones isotérmicas (Fonseca, Pelliccioni, Tim y Aquite, 2014).

2.5.2 Método de Volumen Fluido (VOF)

Es uno de los solucionadores de tipo más utilizados para entender la interacción de un fluido con su entorno. Esta interfase se encuentra representada indirectamente por un campo numérico que describe la fracción de volumen de agua dentro de cada celda computacional (Moukalled et al., 2016). Además, permite discretizar y resolver numéricamente ecuaciones diferenciales en torno a cada punto de esta malla se construye un volumen finito. En cada volumen se toma como referencia un centroide para realizar los cálculos de las variables como presión, velocidad. Finalmente, mediante a la interpolación se puede calcular en las fronteras de cada volumen; está determinada por la siguiente ecuación (Maldonado, 2018):

$$\frac{\partial F}{\partial t} + U \frac{\partial F}{\partial x} + V \frac{\partial F}{\partial y} + W \frac{\partial F}{\partial z} = 0$$

(Ecuación 2)

2.5.1.1 Ecuación de continuidad

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0$$

(Ecuación 3)

Sabiendo que un fluido incompresible se asume que $p = \text{const}$

$$\nabla \cdot u = 0$$

(Ecuación 4)

(Holzinger, 2020)

2.5.1.2 Ecuación de momento

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

(Ecuación 5)

Siendo

u = velocidad

g = gravedad

ρ = presión

τ_{ij} = tensores de viscosidad y turbulencia

$f_{\sigma i}$ = tensión superficial

2.5.1.3 Ecuación de interfase

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

(Ecuación 6)

2.5.2 Algoritmo de solución

La mayoría de los solucionadores de CFD utilizan el algoritmo PIMPLE el cual intenta ajustar el campo de presión durante la simulación. Es la combinación de los algoritmos “pressure-implicit split-operator” (PISO) y el “semi-implicit method for pressure-linked equations” (SIMPLE). Dichos algoritmos son procesos iterativos para la resolución de ecuaciones de momento y conservación en estado transitorio y estacionario (Holzinger, 2020).

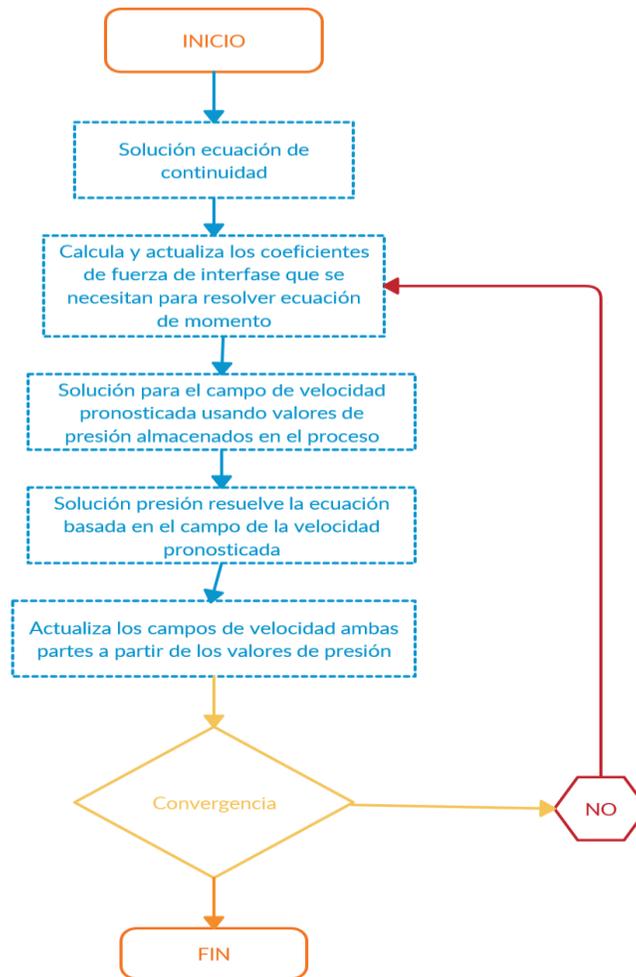


Figura 9. Proceso de solución de algoritmo PIMPLE

Adaptado de (Maldonado, 2018)

2.5.3 Estructura caso OpenFoam

Un caso de OpenFoam contiene un conjunto de archivos y directorios estructurados utilizados, los cuales describen a continuación:

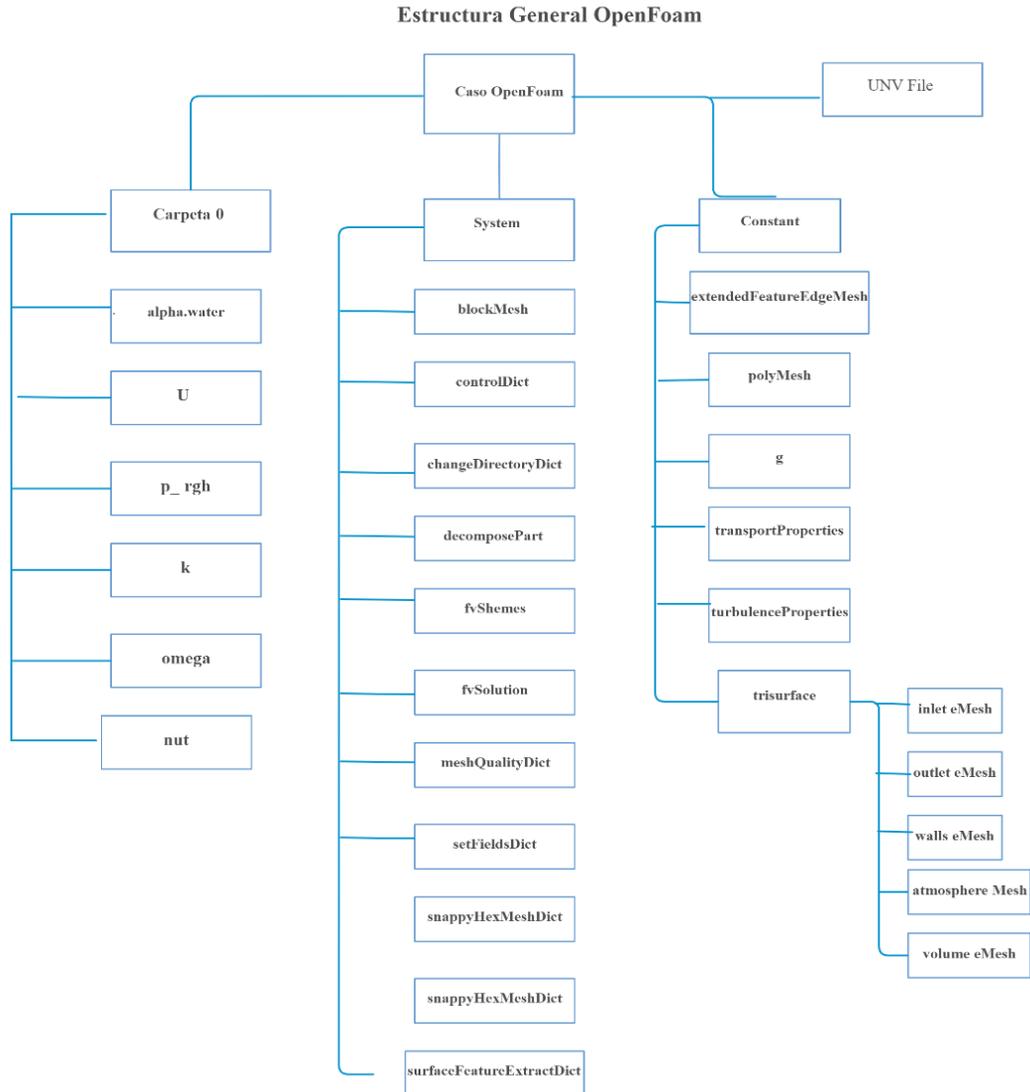


Figura 10. Estructura caso OpenFoam

Adaptado de (Greenshields, 2019)

2.5.3.1 Directorios Requeridos

En el directorio “0” contiene directorios individuales de datos para campos particulares de variables como velocidad, presión. Los datos pueden ser valores

iniciales y condiciones de contorno que se utilizan para definir las características de acuerdo al caso de estudio (Greenshields, 2019).

El directorio “constant” contiene la descripción completa de las propiedades y comportamiento del fluido. También contiene un subdirectorio llamada “polyMesh” donde se encuentra información específica del mallado (Greenshields, 2019).

Finalmente, el directorio “system” se describe los parámetros asociados con el procedimiento de solución (Greenshields, 2019).

2.5.3.1.1 Archivos principales directorio “system”

A continuación, se presenta la descripción de archivos que deben encontrar en la carpeta “system” para poder realizar el cálculo:

Tabla 1. Descripción general de directorio “system”

Tomado de (Greenshields, 2019)

fvSchemes	Define los esquemas de discretización de volumen finito
fvSolution	Contiene los controles relacionados con el solucionador matemático, los algoritmos de resolución y las tolerancias
controlDict	Contiene los controles relacionados con el tiempo y el intervalo de salida de los datos

2.5.3.2 Control tiempo de simulación

En sección se enlista los controles más importantes con respecto al ajuste de tiempo de simulación dentro del archivo “controlDict”

Tabla 2. Apartados de archivo “controlDict”

Tomado de (Holzinger, 2020)

startFrom	Controla el tiempo de inicio de la simulación
startTime	Especifica el tiempo de inicio de la simulación
endTime	Especifica el tiempo de finalización de la simulación
writeNow	Genera una carpeta para registrar los datos de simulación
deltaT	Intervalo de tiempo de simulación
adjustTimeStep	Permite ajustar el tiempo de simulación con relación al número de Courant
maxCo	Número de Courant máximo
writeControl	Controla el registro de datos en la simulación
writePrecision	Controla el registro de datos en el disco local

2.5.3.3 Carpetas obtenidas de simulación

Al instante de iniciada la simulación se fabrican una serie de carpetas temporales asociadas a los intervalos de tiempo configurado en la carpeta “controlDict” donde se almacenaron los nuevos valores de las variables iniciales.

2.5.4 Descripción de utilidades ejecutadas

Tabla 3. Comandos utilizados en la simulación

Tomado de (Greenshields, 2019)

ideasUnvToFoam	Conversión de malla de formato unv a foam
blockMesh	Generador de malla multi bloque
checkMesh	Verifica la calidad del mallado
surfaceFeatureExtract	Extrae y refina las características del mallado
decomposePar	Descompone el mallado para la ejecución en foam
snappyHexMesh	Generación de mallado hexadecimal. Refina y ajusta a la superficie
setFields	Establece los valores de celdas y parches
Touch	Genera un archivo para la visualización de la simulación

2.6 Unidades y dimensiones

OpenFoam utiliza un formato I/O para un conjunto de dimensiones básicas definidos por 7 valores escalares delimitados por corchetes en cada uno de los archivos de la siguiente manera (Greenshields, 2019):

[0 0 0 0 0 0 0]

Tabla 4. Dimensiones básicas OpenFoam
Tomado de (Greenshields, 2019)

N°	Propiedad	Unidad SI
1	Masa	Kilogramo (kg)
2	Longitud	Metro (m)
3	Tiempo	Segundos (s)
4	Temperatura	Kelvin (K)
5	Cantidad	Moles (mol)
6	Corriente	Amperio (A)
7	Intensidad luminosa	Candela (cd)

Las propiedades físicas generalmente se especifican bajo dimensiones asociadas a las unidades generales.

Tabla 5. Unidades OpenFoam

Tomado de (Greenshields, 2019)

Propiedades	Unidad	Formato O/I
Velocidad	m/s	[0 1 -1 0 0 0 0]
Presión	Kg/m. s ²	[1 -1 -2 0 0 0 0]

2.7 Condiciones de borde

Los distintos parches de la geometría necesitan ser especificados para la resolución numérica de acuerdo con el caso que se desea simular.

2.7.1 Tipos de parches

Patch: tipo genérico para todas las condiciones de borde, se utiliza para declarar la entrada (inlet) y la salida (outlet)

Wall: específico tipo de parche utilizado como paredes en modelos de turbulencia

Empty: empleado en planos en 3D y no requieren solución

(Holzinger, 2020)

2.7.2 Tipos de entrada:

fixedValue: especifica un valor fijo a la variable

fixedGradient: la gradiente de una variable en función del tiempo

zeroGradient: la gradiente de una variable permanece constante en función del tiempo

InletOutlet: depende la dirección del flujo si los vectores se dirigen afuera funciona de igual manera que zeroGradient.

pressureInletOutletVelocity: La velocidad y presión son descritas de la misma forma en la entrada y la salida

(Holzinger, 2020)

2.8 Generación de Mallado

El mallado es parte fundamental de la solución numérica y debe satisfacer ciertos criterios con el objetivo de asegurar un resultado válido y preciso (Moukalled et al., 2016).

2.8.1 Descripción de mallado

Por defecto, Open Foam define una malla de celdas poliédricas en 3D, es decir, las celdas pueden tener un número ilimitado de caras donde no hay límite en el número de bordes ni ninguna restricción en su alineación. En la carpeta “polyMesh” se alberga la estructura general del mallado, donde se puede manipular la malla principalmente cuando el dominio geométrico es complejo (Greenshields, 2019).

2.8.1.1 polyMesh

Este subdirectorio almacena los archivos que proporcionan toda la información acerca del mallado.

Points: contiene la lista de las coordenadas de todos los puntos

Faces: contiene la definición de todas las caras

Owner: se relaciona con las caras internas del mallado definidas previamente

Neighbour: proporciona índice de celdas cercanas

Boundary: lista de caras que forman el límite de los parches

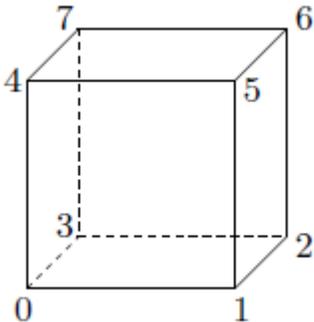
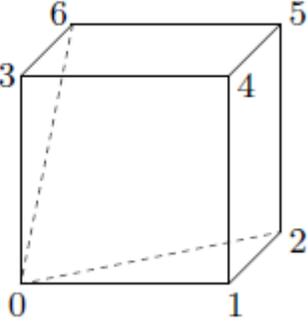
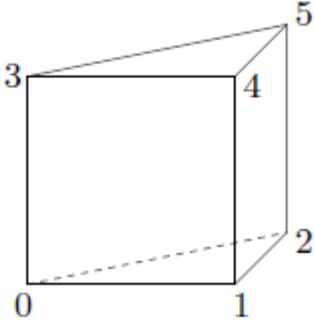
(Cardiff, 2017)

2.8.2 Forma de celdas

OpenFoam admite cualquier forma de celdas, sin embargo, para un mejor desarrollo de la simulación es recomendable el uso de formas de celdas definidas a continuación:

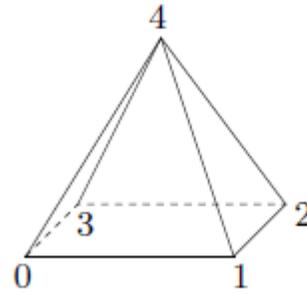
Tabla 6. Tipos de mallado

Tomado de (Greenshields, 2019)

Tipo de Celda	Código	Forma
Hexahedron	Hex	
Wedge	Wedge	
Prism	Prism	

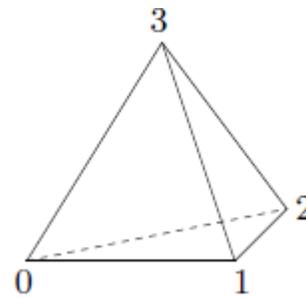
Pyramid

Pyr



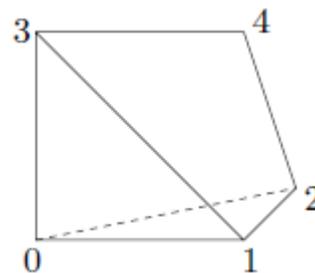
Tetrahedron

Tet



Tet-wedge

TetWedge



2.8.3 Generación de volumen de control

Con la ayuda del archivo blockMesh se permite fabricar mallas paramétricas con gradientes y bordes curvas. La utilidad de esta herramienta es crear el mallado de

elementos de dos dimensiones o la generación y el mallado del volumen de control en geometría tridimensional. Se fundamenta en descomponer el dominio en un cierto número de bloques tridimensionales. Los contornos de los bloques pueden estar compuestas por líneas rectas, arcos o curvas, estando cada uno de los bloques compuesto por sus 8 vértices (Reyes, 2019).

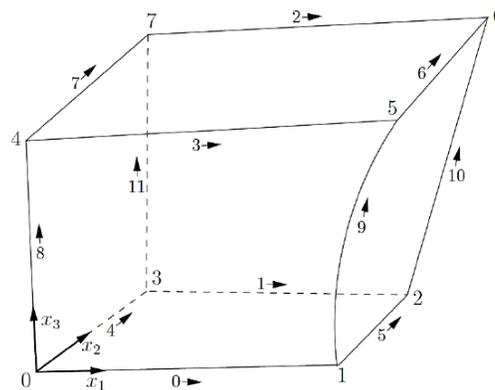


Figura 11. Distribución de vértices de blockMesh

Tomado de (Greenshields, 2019)

Tabla 7. Descripción archivo blockMesh

Tomado de (Greenshields, 2019)

Parámetros blockMesh	Definición
converToMeters	Factor de escala con respecto a metro
Vértices	Lista de coordenadas de cada uno de los vértices de los distintos bloques
Edges	Especifica el tipo de contorno de cada uno de los bloques
Block	Define cada bloque a partir de sus 8 vértices

Boundary	Permite establecer las caras de cada bloque que pertenecen a una de las superficies de contorno del volumen de control
mergePatchPairs	Define los criterios de unión de distintos bloques

2.8.4 Refinamiento de mallado

OpenFoam dispone de una aplicación propia de mallado de superficies, conocida como “snappyHexMesh”, la cual genera mallas tridimensionales compuestas por celdas hexaédricas a partir de geometrías en 3 dimensiones en formato STL. La malla se adapta aproximadamente a la superficie mediante un proceso iterativo, dando origen a una malla inicial. La especificación del nivel de refinamiento de la malla es flexible, permitiendo una buena calidad de mala final (Reyes, 2019).

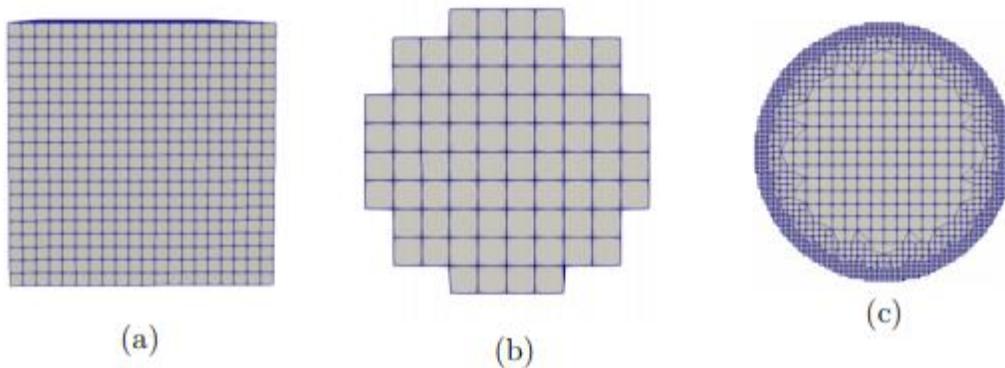


Figura 12. snappyHexMesh

Tomado de (Gullberg, 2017)

En la figura anterior se puede observar el mallado después de correr el blockMesh, snappyHexMesh y el resultado final de refinamiento.

2.9 Número de Courant

Establece un criterio de estabilidad en análisis numéricos mediante CFD. Por lo general es recomendable no superar el valor de $Co=1$, lo cual se cree que cada paso de tiempo de partículas del fluido se mueve de una celda a otra, de manera que si $C>1$, las partículas del fluido se desplazan a través de 2 o más celdas en cada paso de tiempo. En el archivo “controlDict” se puede definir este parámetro de la siguiente manera dependiendo del caso de estudio:

```
maxCo 1;
```

```
maxAlphaCo 1;
```

```
maxDeltat 1;
```

Esta herramienta permite llevar a cabo simulaciones estables a pesar de poseer altos maxCo obteniendo resultados en menor tiempo; por el contrario, valores menores de maxCo permiten más estabilidad en la simulación y resultados precisos (Reyes, 2019).

2.10 Post-procesamiento

La principal herramienta de post-procesamiento es el software de libre acceso “ParaView” con la capacidad para la visualización de grupos de datos en 2 y 3 dimensiones. Una vez finalizada la simulación se puede observar los datos de salida, la geometría predeterminada del fluido donde se proyectarán en un conjunto

de datos representados en 3D, donde se podrán evaluar los resultados de tiempo de residencia y el comportamiento hidráulico del caso (Muttenthaler, 2017).

3. METODOLOGÍA

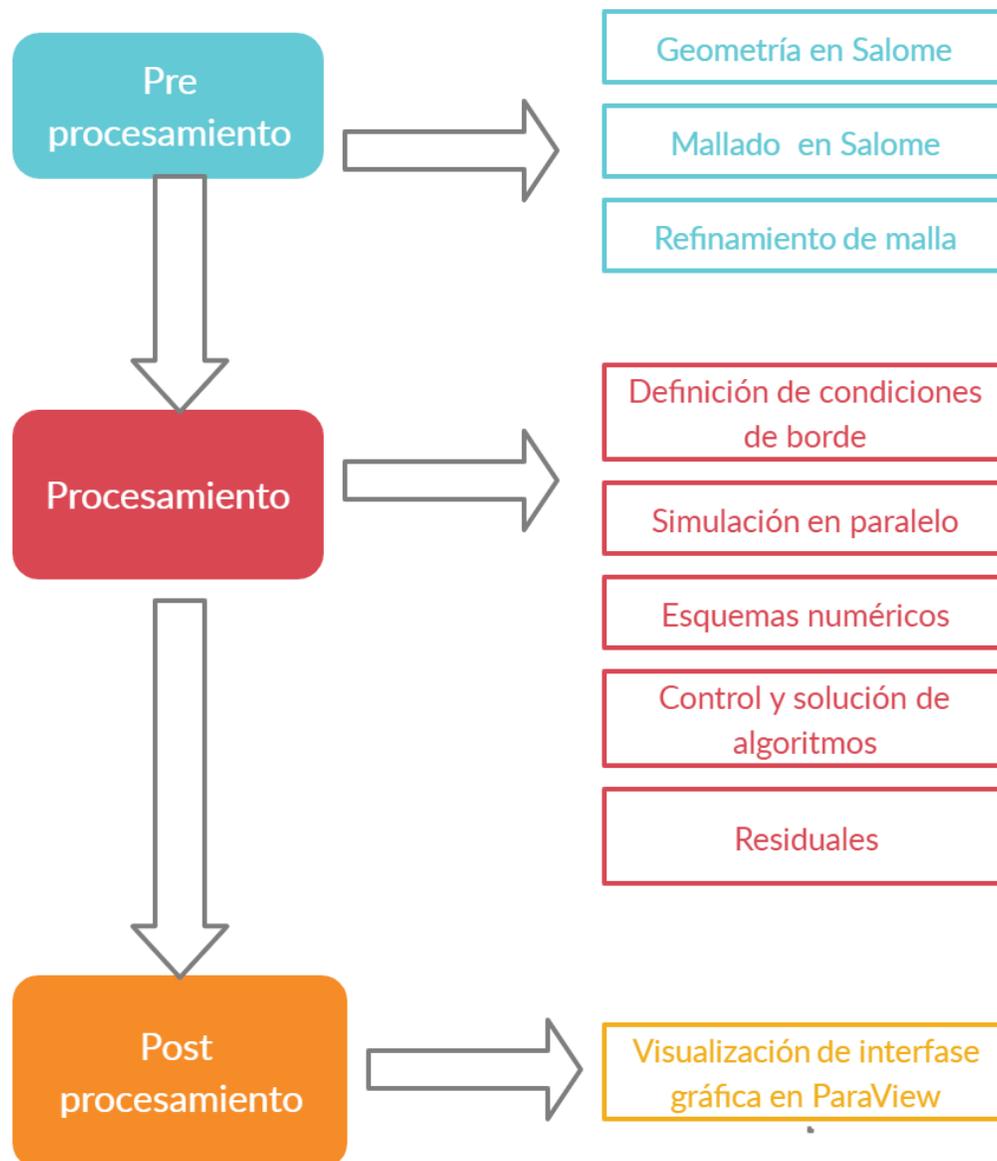
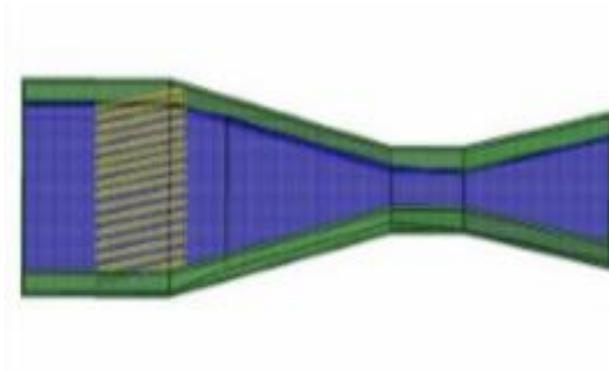
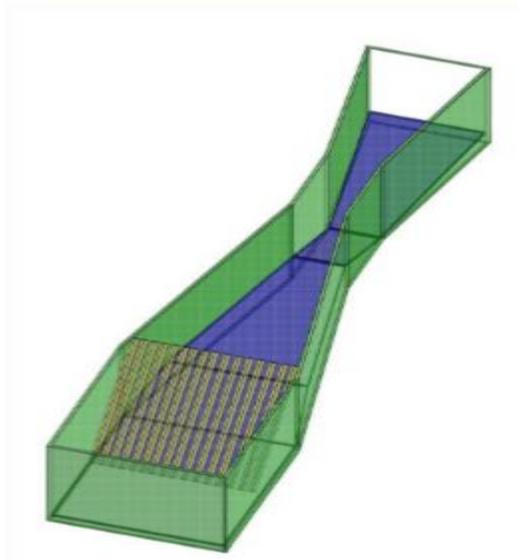


Figura 13. Esquema de etapas desarrolladas en la metodología

3.1 Preprocesamiento

3.1.1 Geometría

La geometría fue definida a partir del diseño de la fase preliminar de la Planta de Tratamiento Piloto de Aguas Residuales de Udlapark (Fierro, 2018), con un caudal de diseño de 40 l/min como se muestra a continuación:



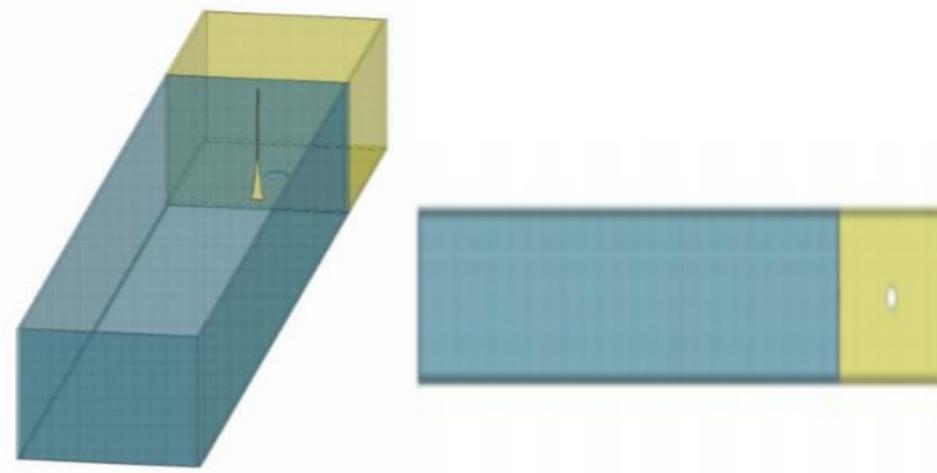


Figura 14. Geometría del caso

Tomado de (Fierro, 2018)

En las siguientes tablas se presenta las especificaciones que se utilizaron en el diseño de la geometría de la canaleta parshall y el desarenador con veredero sutro:

Tabla 8. Especificaciones de canal de aproximación y canaleta parshall

Variable	Eje	Unidad	Valor
Largo	Y	m	0.22
Longitud	X	m	2.25
Altura	Z	m	0.21
Ø tubería de entrada		in	2
Numero de rejillas		N	13
Ancho de barras		mm	5

Espacio entre barras	mm	15
----------------------	----	----

Tabla 9. Especificaciones desarenador

Variable	Eje	Unidad	Valor
Ancho	Y	m	0.2
Largo	X	m	1.23
Profundidad	Z	m	0.2
Ø tubería de salida		in	2
Volumen		m ³	0.049

3.1.2 Geometría en Salome

Se incorporó al diseño original un canal de aproximación de 1 m de largo, ya que este garantiza un flujo uniforme del afluente al momento de llegar a la canaleta parshall (Lopera, 2010) .El diseño fue efectuado principalmente por vértices que posteriormente formaron caras y finalmente se formó un solo sólido. Además, se realizó el corte de los espacios de las rejillas después del canal de aproximación.

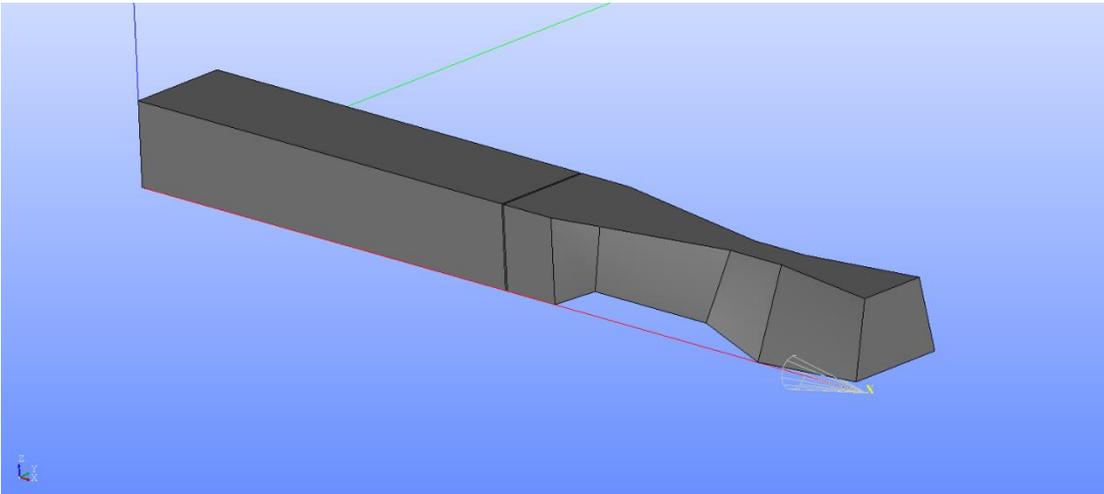


Figura 15. Geometría Canaleta Parshall en Salome

El diseño del desarenador fue realizado principalmente por la fusión de varios sólidos; el vertedero sutro fue definido por distintos vértices hasta formar una cara, la misma que fue extraída de la fusión de sólidos antes mencionada.

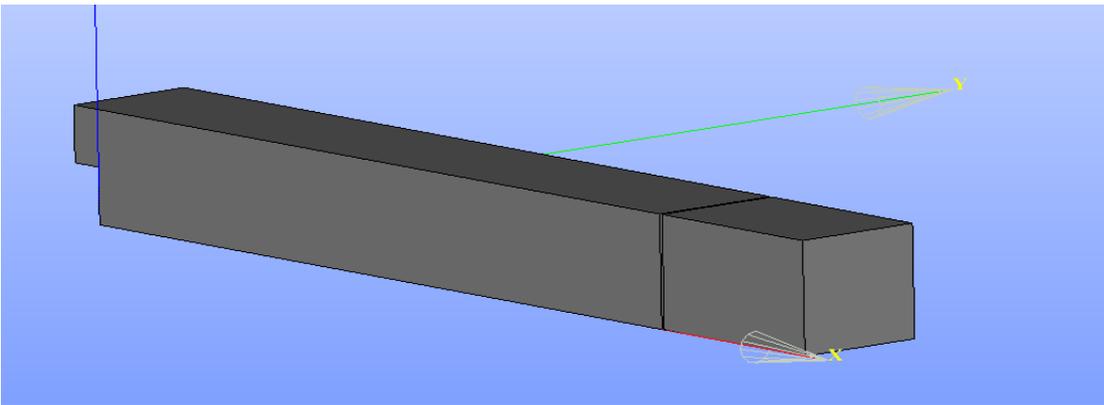


Figura 16. Geometría Desarenador en Salome

3.1.2.1 Definición de grupos

A continuación, se muestra la definición de distintos tipos de caras del sólido los cuales influyen en el comportamiento del fluido dentro de la simulación.

Tabla 10. Definición de grupos de geometría de la canaleta parshall

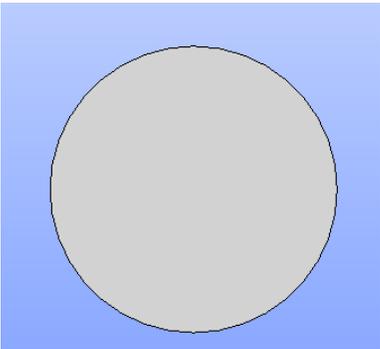
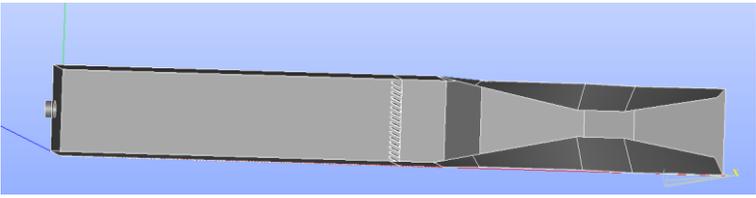
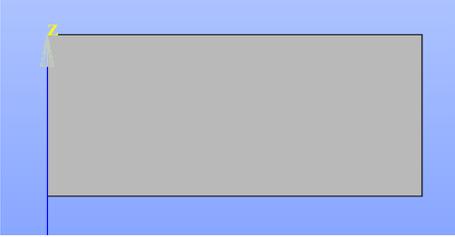
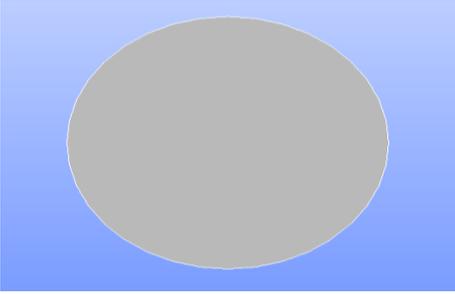
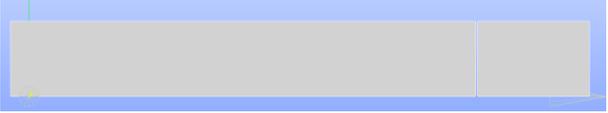
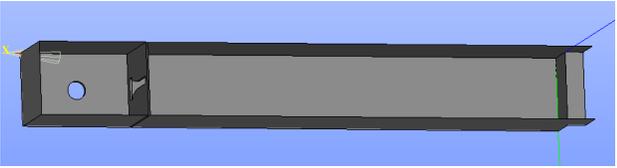
Inlet	Cara por donde ingresa el agua	
Outlet	Cara por donde sale el agua	
Atmosphere	Caras por donde no tienen la interacción con el agua	
Walls	Caras con las que el agua interactúa	

Tabla 11. Definición de grupos de geometría desarenador

Inlet	
Outlet	
Atmosphere	
Walls	

3.1.3 Generación de mallado

Una vez generada la geometría de la canaleta parshall y el desarenador con el vertedero sutro, se generó una malla fina como se observa en las Figuras 17 y 18 a continuación:

Tabla 12. Especificaciones de mallado en Salome

	Canaleta Parshall	Desarenador
Celdas	94 226	82 244
Forma	Tetraedros	Tetraedros

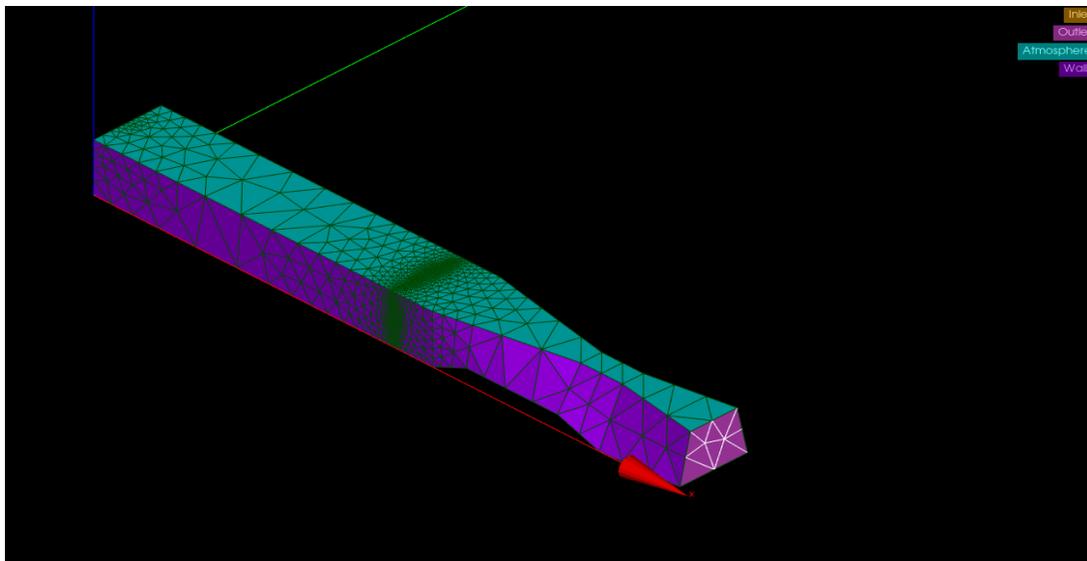


Figura 17. Mallado de la Canaleta Parshall en Salome

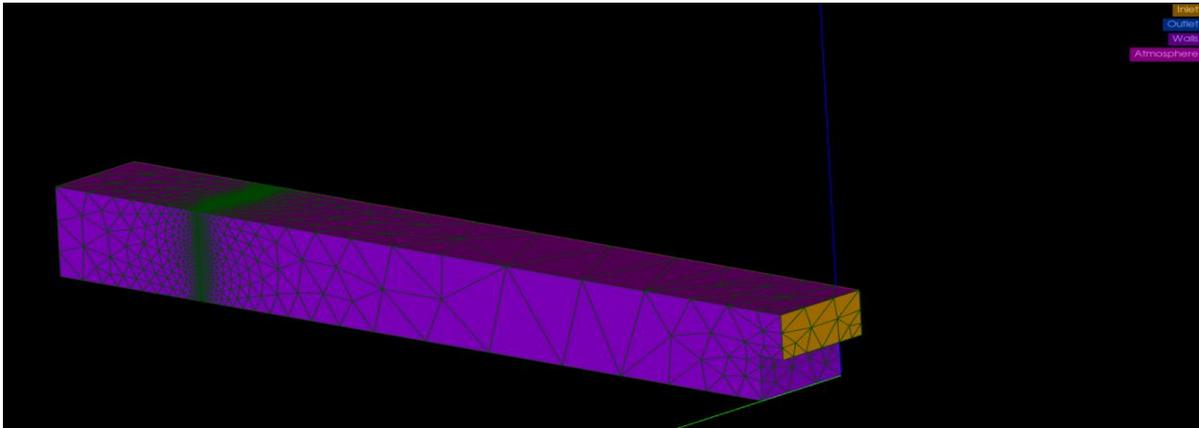


Figura 18. Mallado del Desarenador en Salome

3.1.3.1 Generación de archivos UNV y STL

Se exportó la geometría completa en formato UNV y las caras definidas (inlet, outlet, atmosphere, walls y volume) en el punto anterior en formato STL que fueron colocadas en la carpeta “ trisurface” para su posterior utilización en la generación del snappyHexMesh.

3.1.4 blockMesh

Es una de las herramientas básicas para la generación del mallado. Se utilizó con el objetivo de descomponer la geometría del dominio para generar bloques hexaédricos a partir de vértices definidos (ejes x, y,z) en el archivo “blockMeshDict” a partir de la geometría de los archivos STL de la carpeta trisurface. Las definiciones de bloques están contenidas en la lista llamada “blocks”. La definición de los vectores que proporciona el número de celdas requeridas en cada dirección, fueron calculadas en base a las dimensiones de la geometría. El “blockMesh” fue utilizado

como herramienta para la generación del mallado inicial para la ejecución de snappyHexMesh (Gullberg, 2017).

Tabla 13. Especificaciones archivo “blockMesh”

	Canaleta Parshall	Desarenador
Vértices		
	(-0.1 -0.1 -0.1) // 0	(-0.1 -0.1 -0.1) // 0
	(2.35 -0.1 -0.08) // 1	(1.6 -0.1 -0.1) // 1
	(2.35 0.36 -0.08) // 2	(1.6 0.3 -0.1) // 2
	(-0.1 0.36 -0.1) // 3	(-0.1 0.3 -0.1) // 3
	(-0.1 -0.1 0.42) // 4	(-0.1 -0.1 0.3) // 4
	(2.35 -0.1 0.42) // 5	(1.6 -0.1 0.3) // 5
	(2.35 0.36 0.42) // 6	(1.6 0.3 0.3) // 6
	(-0.1 0.36 0.42) // 7	(1.6 0.3 0.3) // 7
blocks hex	(45 5 5)	(50 7 7)

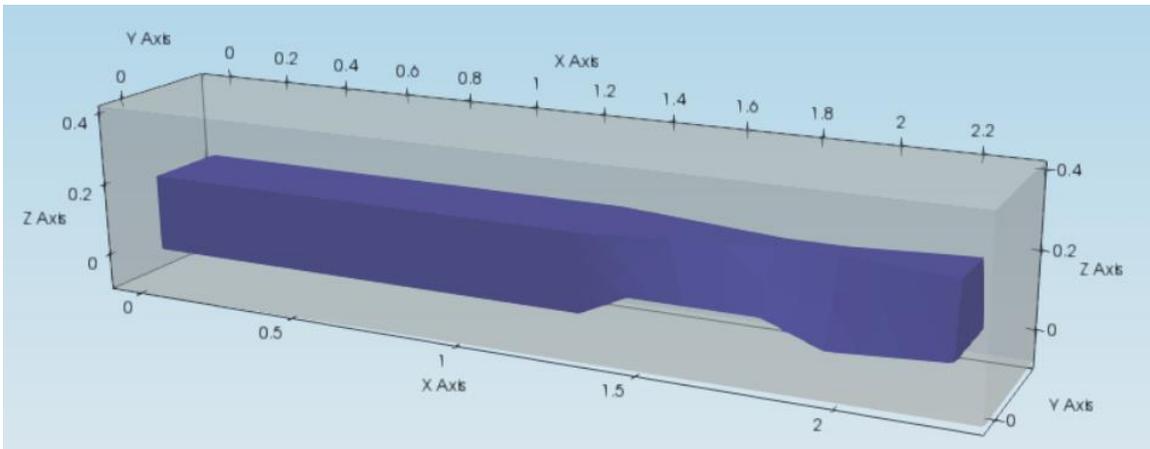


Figura 19. Volumen de control de la Canaleta Parshall

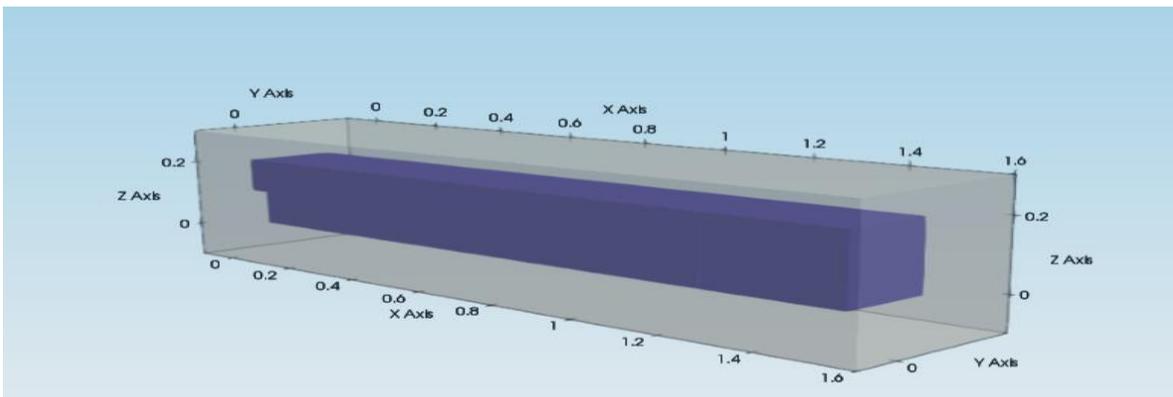


Figura 20. Volumen de control del Desarenador

3.1.5 snappyHexMesh

Esta herramienta se utilizó para el refinamiento del mallado tridimensional de ambos casos. Inicialmente se generaron 1227 370 celdas en 470 segundos en el caso de la canaleta parshall y 1391 950 celdas en 558 segundos en el caso del desarenador.

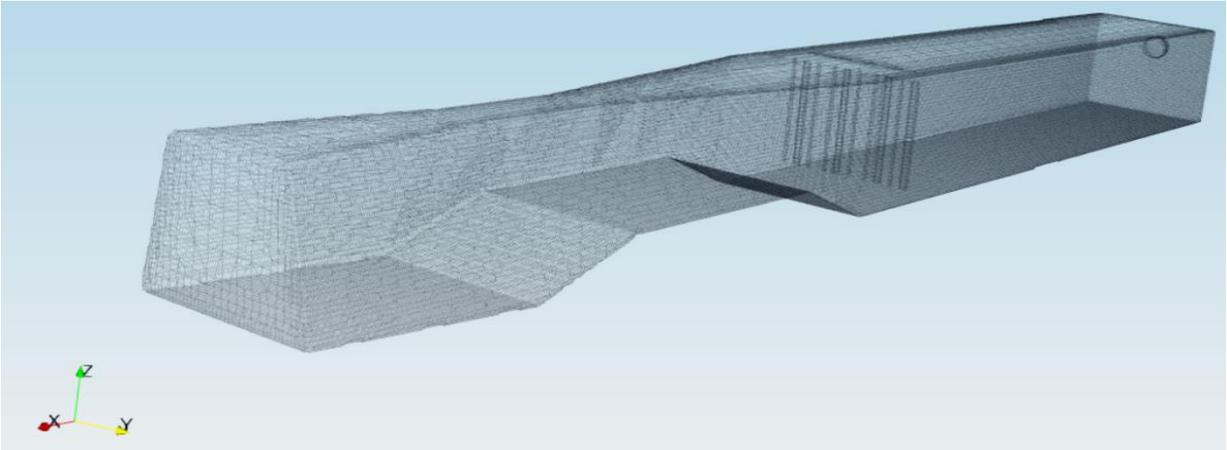


Figura 21. snappyHexMesh de la Canaleta Parshall

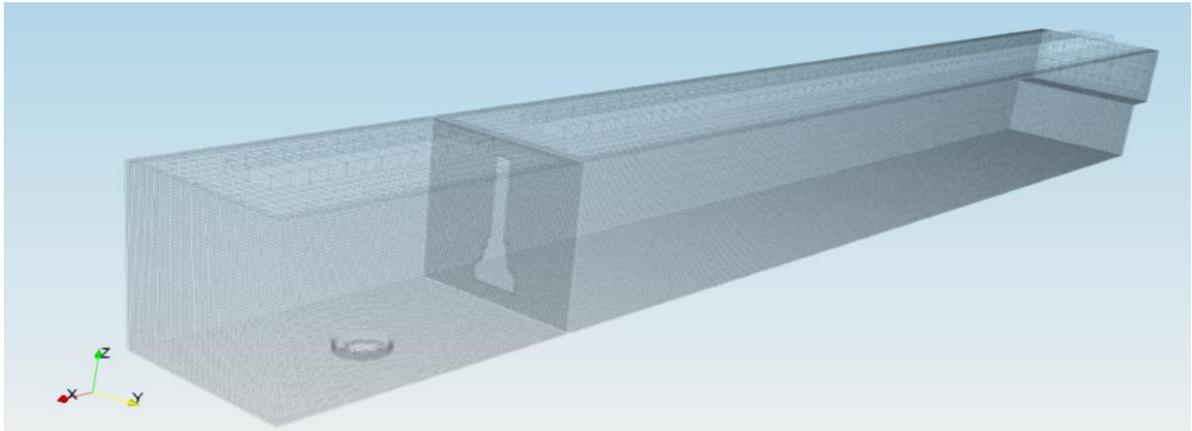


Figura 22. snappyHexMesh del Desarenador

3.2 Procesamiento

3.2.1 Condiciones de borde

Una vez generado el mallado, se inicia con la configuración de las condiciones de contorno de las carpetas `u`, `p_rgh` y `alpha water` que se encuentran en el subdirectorio “0”

3.2.1.1 Directorio “0”

Tabla 14. Condiciones de borde archivo “U” canaleta parshall

	internalField	uniform (0 0 0)	
		Type	Value
Archivo U	Inlet	fixedValue	uniform (0.32 0 0)
	Outlet	inletOutlet	uniform (0.32 0 0)
	Atmosphere	presurreInletOutletVelocity	uniform (0 0 0)
	Walls	-	-

Tabla 15. Condiciones de borde archivo “U” desarenador

	internalField	uniform (0 0 0)	
		Type	Value
Archivo U	Inlet	fixedValue	uniform (0.26 0 0)
	Outlet	inletOutlet	uniform (0.26 0 0)

Atmosphere	pressureInletOutletVelocity	uniform (0 0 0)
Walls	-	-

El parámetro `internalField` define las condiciones iniciales de la simulación como constante (`uniform 0 0 0`) en todo el dominio. La velocidad en la entrada (`inlet`) fue definida como uniforme en el eje x. La velocidad de los distintos casos fue calculada a partir de la velocidad de Manning que toma en cuenta la sección transversal de la entrada del agua.

Tabla 16. Condiciones de borde archivo “alpha.water” canaleta parshall

	<code>internalField</code>	<code>uniform 0</code>	
		Type	Value
Archivo alpha water	Inlet	<code>fixedValue</code>	<code>uniform 0.5</code>
	Outlet	<code>zeroGradient</code>	<code>uniform 0</code>
	Atmosphere	<code>inletOutlet</code>	<code>uniform 0</code>
	Walls	<code>zeroGradient</code>	<code>uniform 0</code>

Tabla 17. Condiciones de borde archivo “alpha.water” desarenador

	internalField	uniform 0	
		Type	Value
Archivo alpha water	Inlet	fixedValue	uniform 1
	Outlet	zeroGradient	uniform 0
	Atmosphere	inletOutlet	uniform 0
	Walls	zeroGradient	uniform 0

Dentro del archivo “alpha water” se definió un valor de 0.5 en la entrada de la canaleta parshall ya que el agua no entra por toda la sección transversal de tubería. El resto de los archivos k, nut, omega y p_rgh no se realizó ninguna modificación.

3.2.1.2 Directorio “system”

A continuación, se presenta la configuración del archivo “controlDict” para ambos casos de simulación.

Tabla 18. Configuración archivo “controlDict”

Application	interFoam
startTime	0

endTime	60/80
deltaT	0.001
writeControl	adjustableRunTime
writeInterval	1
purgeWrite	0
writePrecision	6
timePrecisión	6
adjustTimeStep	No
maxCo	3
maxAlphaCo	2
maxDelta T	1

3.2.2 Simulación en paralelo

OpenFoam es capaz de hacer simulaciones en paralelo, no existe una diferencia significativa en el cálculo con un proceso único con múltiples procesadores (Holzinger, 2020). En el archivo “decomposePar” divide el cálculo del dominio de acuerdo con la geometría. Para esta simulación se especificó dos núcleos para la resolución en el eje “x”. Finalmente, para empezar la simulación se colocó el comando “`mpiexec -n2 interFoam -parallel`”

Para la simulación de la canaleta parshall y el desarenador se definió de acuerdo con su comportamiento hidráulico estos no funcionan llenos en su totalidad por ende en el archivo “setFields” se colocó (0 0 0).

3.2.3 Esquemas numéricos

3.2.3.1 fvSchemes

Establece los esquemas numéricos de todas las operaciones que realiza el solver donde incluye las ecuaciones para cada una de las variables que se pretenden calcular durante la simulación. A continuación, se detalla el archivo “fvSchemes” del caso:

Tabla 19. Configuración archivo “fvSchemes”

Grupo		
ddtSchemes	Euler	
gradSchemes	Gauss linear	
	div (rhoPhi,U)	Gauss linearUpwind grad
divSchemes	div (phi, alpha)	Gauss vanLeer
	div (phirb, alpha)	Gaus linear
laplacianSchemes	Default	Gauss linear corrected
interpolationSchemes	Default	Linear
snGradSchemes	Default	Corrected

En el apartado “`ddtSchemes`” define la primera derivada y el esquema de discretización de cada término (Tabla 19). Generalmente se encuentra configurado como estado transitorio o estático; para el cual se puede escoger los esquemas que se detallan a continuación (Greenshields, 2019):

Tabla 20. Opciones de configuración apartado “`ddtSchemes`”

<code>Euler</code>	Transitorio borderizado
<code>steadyState</code>	Define el tiempo de la derivada en cero
<code>Backward</code>	Transitorio, potencialmente no borderizado
<code>localEuler</code>	Pseudo- transitorio para la solución en estado estático
<code>crankNicolson</code>	Requiere de un coeficiente ψ . Donde, si $\psi=1$ entonces es CrankNicolson, si $\psi=0$, es Euler.

“`gradSchemes`” define términos de gradiente, se utiliza como entrada el método de “Gauss” que se basa en el volumen estándar de discretización de la integración gaussiana que requiere la interpolación de valores desde el centro de las celdas (Greenshields, 2019).

“`divSchemes`” contiene los posibles esquemas de evaluación del operador divergencia. “`div (phi, alpha)`” establece la velocidad del flujo en las caras de las celdas con una densidad constante (Greenshields, 2019).

“`laplacianSchemes`” contiene términos laplacianos ∇ que establecen la variación en tres ejes coordinados. Este apartado utiliza el esquema de Gauss de discretización

que utiliza un coeficiente de difusión para la interpolación lineal de los datos (Greenshields, 2019).

“interpolation Schemes” contiene términos que son interpolaciones de los valores típicos de los centros de las celdas los cuales se dividen en dos categorías: la primera de esquemas numéricos generales y la segunda esquemas utilizados principalmente en conjunto con discretización Gauss (divergencia) en términos del flujo del fluido (Greenshields, 2019).

Finalmente, el apartado “snGradSchemes” componente de gradiente normal a una cara de la celda sustrayendo el calor del centro de la celda (Greenshields, 2019).

3.2.4 Control y solución de algoritmos

Constituye la parte central del recurso numérico que se encarga de la integración de las ecuaciones que gobiernan el sistema: la discretización y la resolución de estas ecuaciones en forma iterativa (Ruiz, 2015).

3.2.4.1 fvSolution

En este apartado se debe especificar los controles para la solución del algoritmo, métodos de solución y tolerancia (Greenshields, 2019).

Tabla 21. Configuración archivo “fvSolution”

Factores	Parámetros
nAlphaCorr	1
nAlphaSubCycles	1

	cAlpha	1
alpha.water	MULESCorr	Yes
	nLimiterIter	3
	Solver	smoothSolver
	Smoother	symGaussSeidel
	Tolerance	1e-8
	relTol	0
<hr/>		
	Solver	GAMG
p_rgh	Tolerance	5e-9
	relTol	0.01
<hr/>		
	Tolerance	5e-9
p_rgh Final	relTol	0
<hr/>		
	Solver	smoothSolver
	Smoother	symGaussSeidel
U/k/ omega	nSweeps	1
	Tolerance	1e-6
	relTol	0.1
<hr/>		

El primer subdirectorio “solver” especifica que para cada solucionador lineal se utiliza las distintas ecuaciones discretizadas; refiriéndose al método de solución de la matriz de “solver”(Greenshields, 2019).

La sintaxis para la entrada dentro del archivo “fvSolution” empieza con una palabra definida para cada variable (alpha.water, p_rgh ,p_rgh final, U, omega y K) que va a hacer resuelta en las distintas ecuaciones (Greenshields, 2019).

El algoritmo de solución PIMPLE utiliza “nCorrectors” para establecer el número de veces que el algoritmo resuelve la ecuación de presión y momento en cada paso, desde que inicia la simulación. Además, el apartado “ nOuterCorrectors” se utiliza para habilitar la solución de las ecuaciones del sistema en el registro del tiempo (Greenshields, 2019).

Tabla 22. Configuración de algoritmo de solución

PIMPLE	
momentumPredictor	Yes
nCorrectors	2
nNonOrthogonalCorrectors	0

3.2.5 Residuales

En cualquier proceso de solución iterativa es importante poder determinar cuando la solución puede considerarse lo suficientemente correcta o cuando se puede estimar que el error se encuentra dentro del rango de tolerancia o incluso de la precisión que tienen las ecuaciones de conservación hasta que se encuentren

satisfechas (Moukalled et al., 2016). Este método se utiliza para el análisis de las condiciones de contorno para las geometrías definidas. En el terminal se colocó el comando “`mpiexec -n2 interFoam -parallel >log`” lo cual crea una carpeta donde se almacenan estos datos para ser visualizados en el software “gnuplot”

3.3 Post-Procesamiento

La simulación de la interfase gráfica fue plasmada en el software “ParaView” donde se puede observar el comportamiento de la canaleta parshall y el desarenador de acuerdo con la geometría establecida y las condiciones de borde.

4. RESULTADOS Y ANÁLISIS

En este apartado, se presenta el análisis de los resultados obtenidos en las simulaciones de los distintos casos antes mencionados. Los mismos que son representados gráficamente mediante distintas herramientas del software ParaView, obteniéndose las gráficas de dirección, comportamiento del fluido, así como los vectores y perfiles de velocidad.

4.1 Residuales

Durante el proceso de solución se monitoreó la convergencia mediante la revisión de los residuales graficando las velocidades en el eje x como se puede observar en las Figuras 23 y 24 de ambos casos. Las simulaciones encontraron convergencia a los 60 segundos en el caso de la canaleta parshall y 80 segundos en el caso del desarenador.

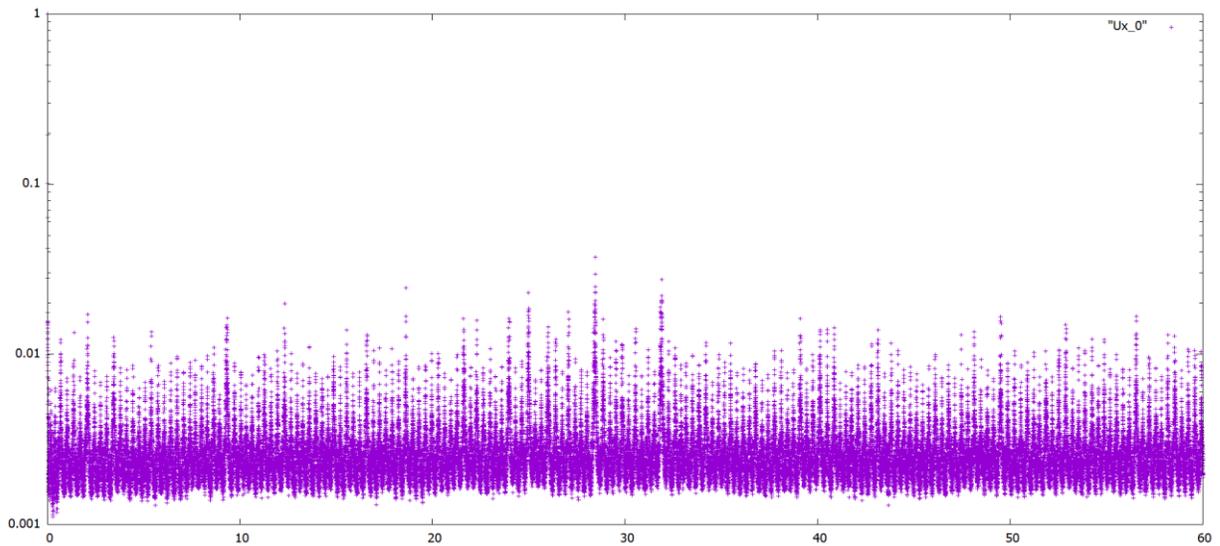


Figura 23. Residuales obtenidos en la simulación de la canaleta parshall

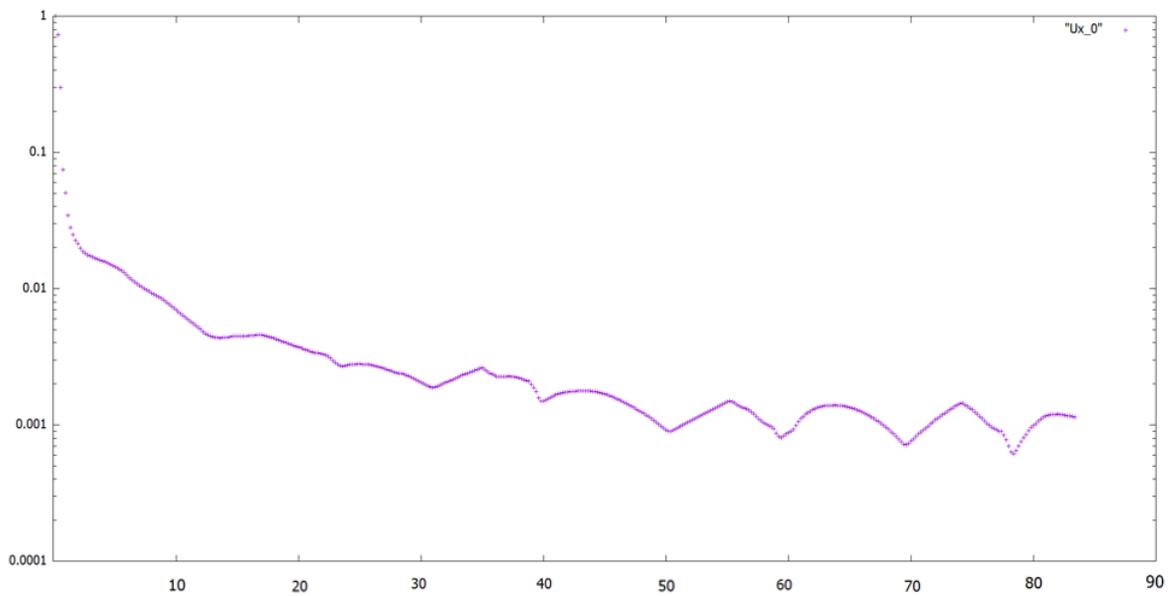


Figura 24. Residuales obtenidos en la simulación del desarenador

4.2 Simulación Canaleta Parshall

En la Figura 25, se observan las líneas de flujo iniciando su trayectoria de izquierda a derecha. Lo cual nos indica que tanto en la entrada como en la salida se generan las velocidades más altas, sin embargo, en la zona central existe una variación de velocidad debido a los obstáculos previamente definidos en la geometría del caso.

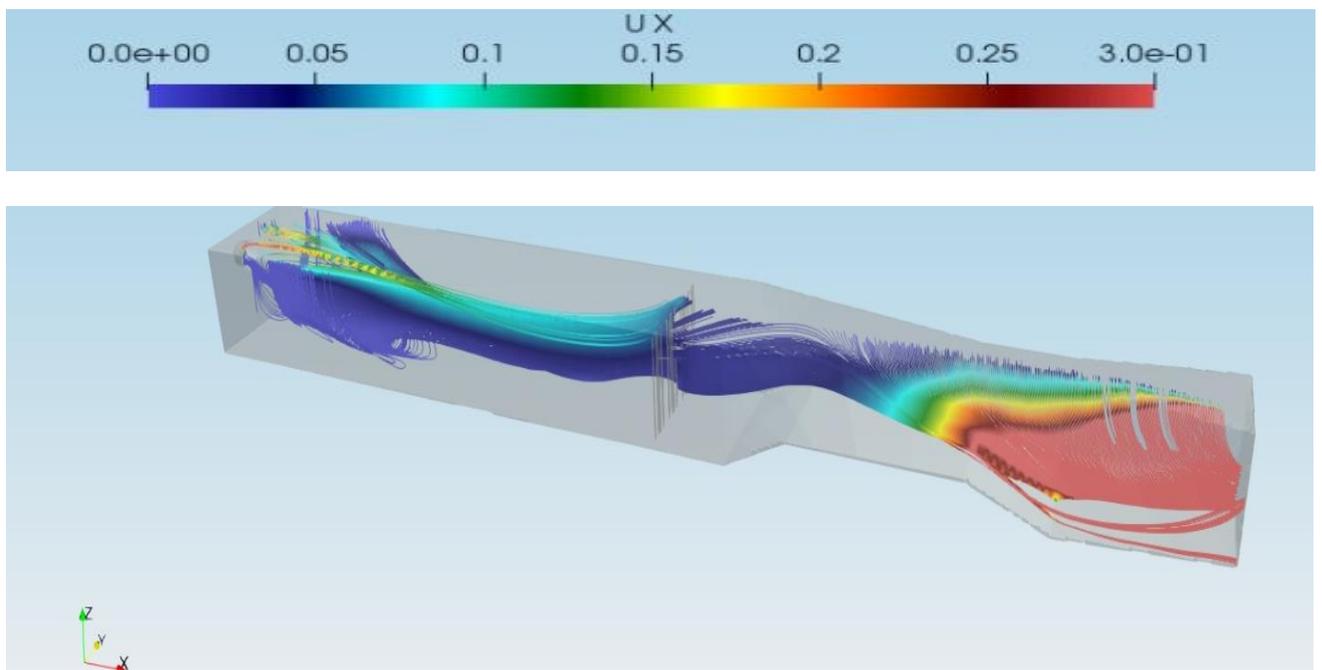


Figura 25. Líneas de flujo de la canaleta parshall

Con respecto, a los vectores de velocidad indicados a en la Figura 24, se evidencia que en la zona del resalto hidráulico inicia una transición hasta una velocidad uniforme en la salida, lo cual favorecería al ingreso de un flujo laminar a posteriores unidades de tratamiento (Hovány y Gabrić, 2014).

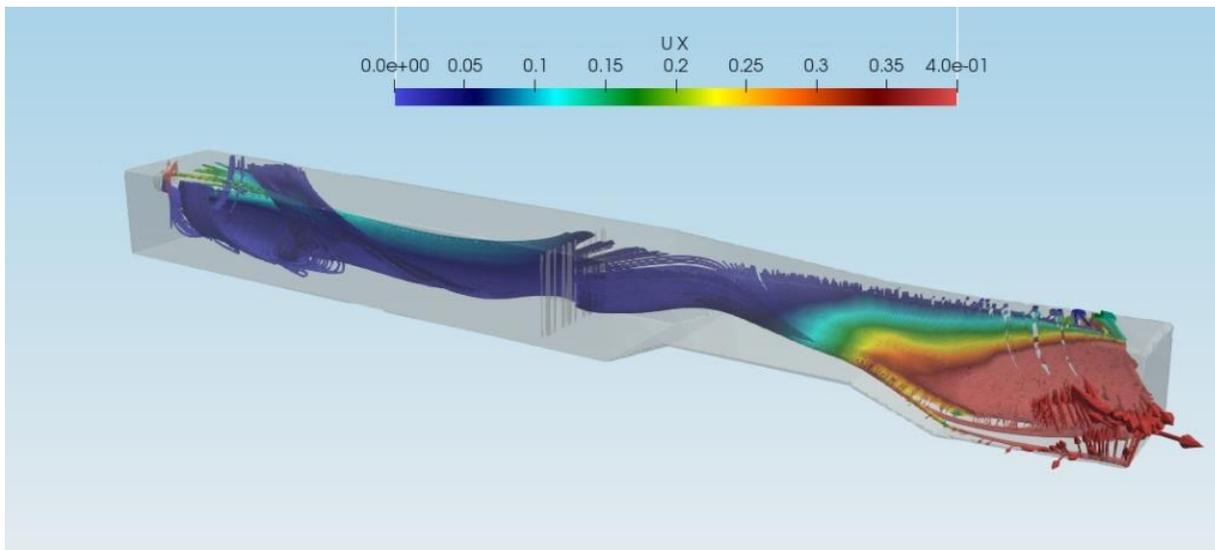


Figura 26. Vectores de velocidad de la canaleta parshall

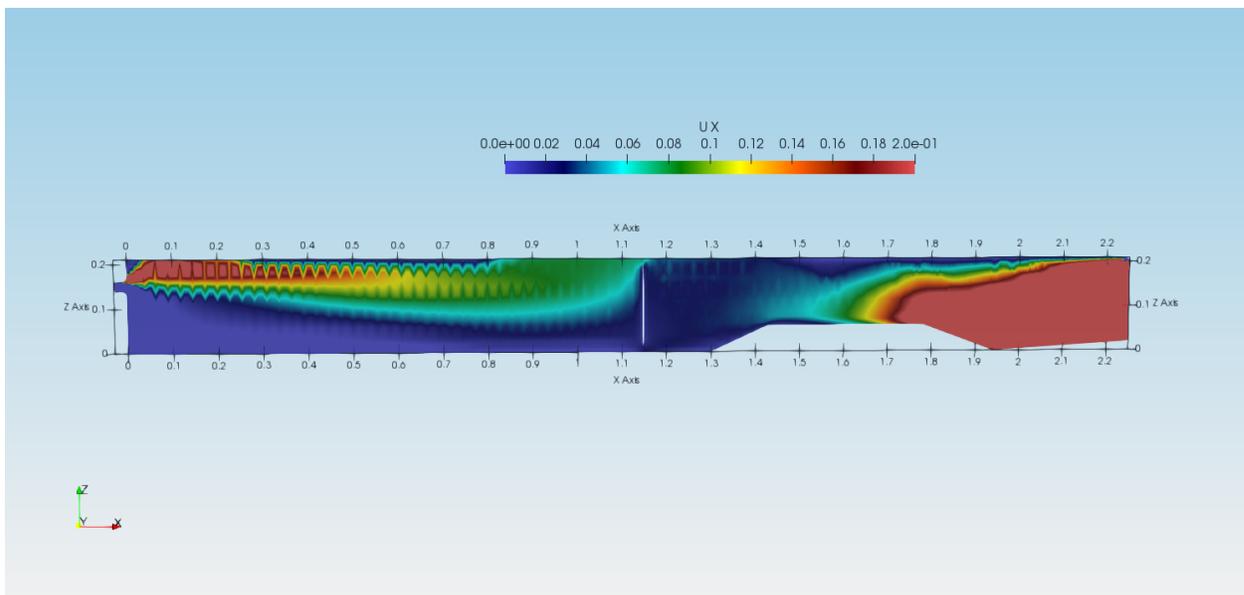


Figura 27. Velocidad (Ux) canaleta parshall

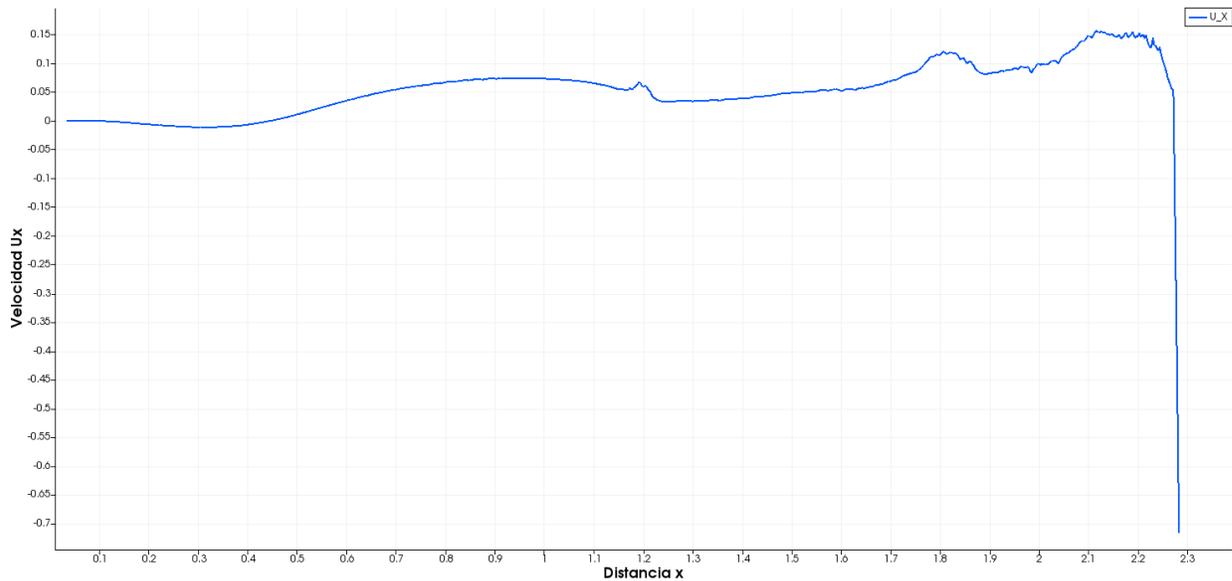


Figura 28. Gráficos velocidad (U_x) con respecto a la distancia (x)

En las Figuras 27 y 28 se observan las distintas velocidades (U_x) a lo largo de todo del canal parshall. Dado que la entrada del canal es mediante una tubería de 2 pulgadas de diámetro la velocidad inicial es mayor a 0,2 m/s, la cual se va reduciendo hasta llegar a una distancia de 1.1 m donde se encuentran las rejillas existiendo una ligera variación de la velocidad a causa de la pérdida de carga. Finalmente se observa como las velocidades van aumentando en la sección divergente.

4.3 Simulación desarenador

En la Figura 24, se observan líneas de flujo iniciando su trayectoria de izquierda a derecha con movimientos irregulares y variaciones en la velocidad del desarenador, lo que se significa que funciona bajo de un régimen turbulento.

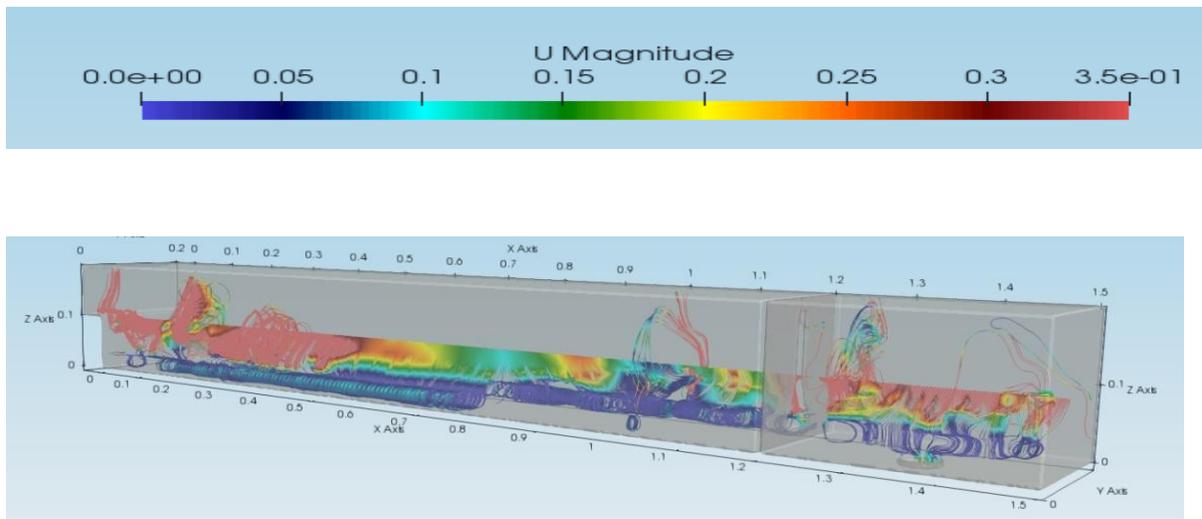


Figura 29. Líneas de flujo del desarenador

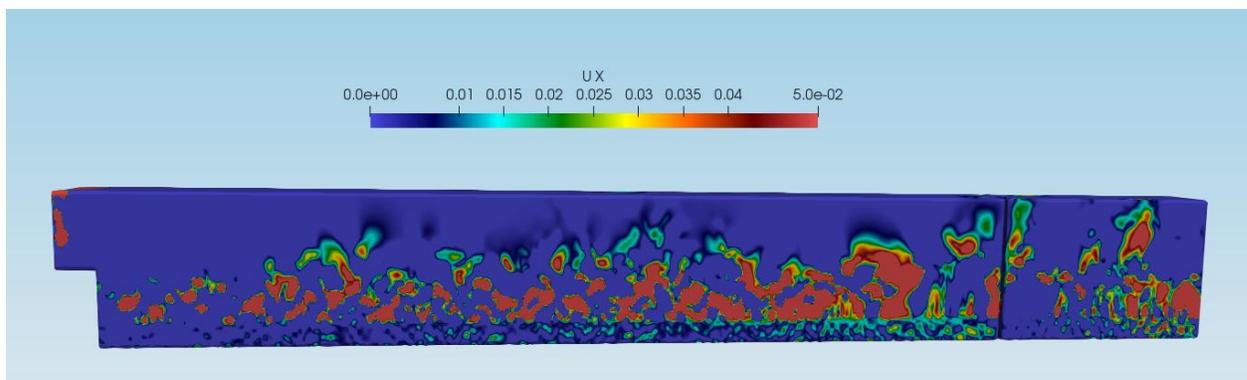


Figura 30. Velocidad (Ux) desarenador

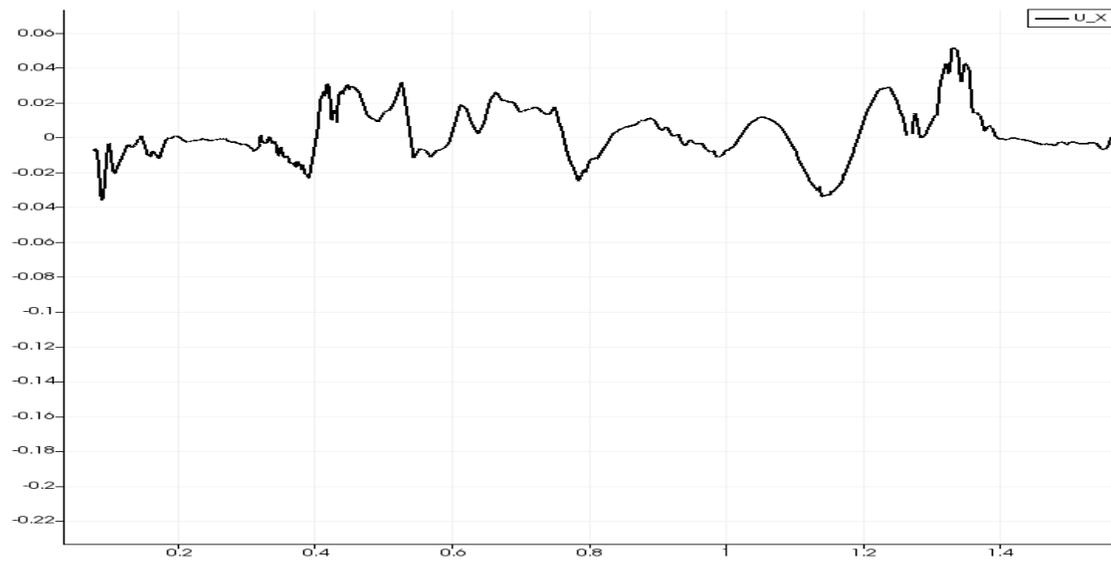


Figura 31. Gráfico velocidad (U_x) con respecto a la distancia (x)

De acuerdo a lo determinado por (Fierro, 2018) en el diseño del desarenador la velocidad horizontal teórica es de 0,027 m/s. En las Figuras 30 y 31 se observa que las velocidades en el eje (U_x) se encuentran en un rango de 0,01 m/s a 0.05 m/s. Se recomienda que esta velocidad sea menor o igual a 0.3 m/s para que las partículas tengan el suficiente tiempo para que se sedimenten (Metcalf and Eddy, 2014).

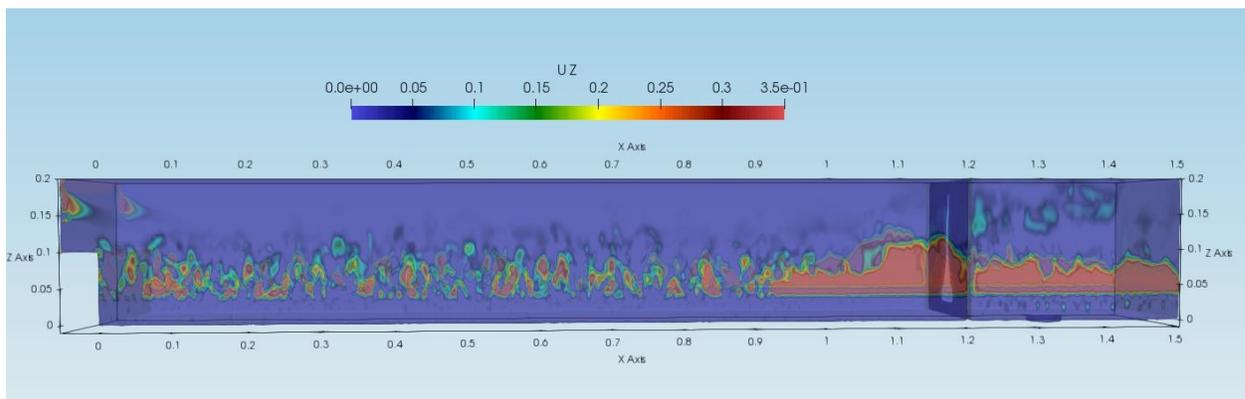


Figura 32. Velocidad (U_z) desarenador

A continuación, se observan las velocidades en el eje z a lo largo del tanque desarenador lo que representaría la velocidad de sedimentación de las partículas. Estas se encuentran en un rango 0.05 m/s a 0.35 m/s, en comparación con la velocidad de sedimentación teórica de 0,349 m/s definida por (Fierro, 2018). Además, dado que el fluido se encuentra en movimiento, la turbulencia genera velocidades altas que pueden afectar la caída de las partículas (Salinas, García, & Díaz, 2008).

4.4 Tiempo de residencia

A continuación, se presenta la división de tramos de la canaleta parshall y el desarenador de acuerdo con las velocidades y la distancia lo cual permite realizar el cálculo de tiempo de residencia obteniéndose valores de 0.48 segundos y 37,5 segundos respectivamente.

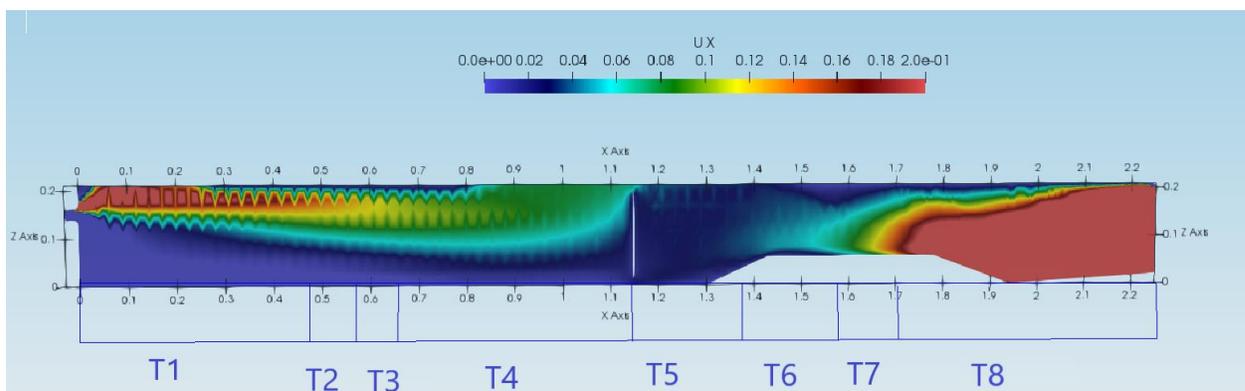


Figura 33. Definición de tramos para cálculo de tiempo de residencia de la canaleta parshall

Tabla 23: Definición de tiempo de residencia

	Tramos									
	Unidad	T1	T2	T3	T4	T5	T6	T7	T8	Total
Distancia	M	0.53	0.01	0.1	0.55	0.255	0.187	0.12	0.55	2.302
Velocidad	m/s	0.1	0.14	0.13	0.1	0.025	0.06	0.12	0.6	
TRH	S	0.053	0.001	0.013	0.055	0.006	0.011	0.014	0.330	0.48

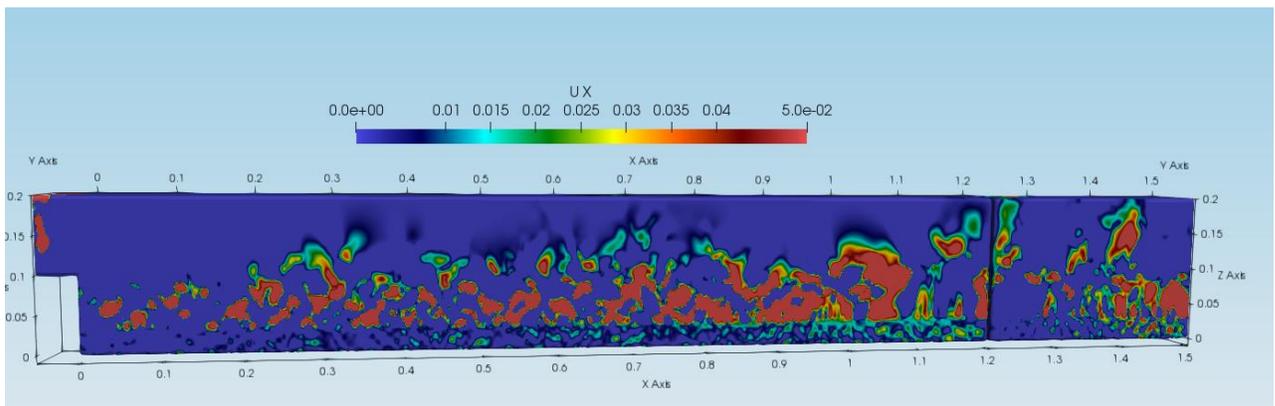


Figura 34. Definición de tiempo de residencia del desarenador

Tabla 24: Definición de tiempo de retención desarenador

VARIABLE	UNIDAD	VALOR
Distancia desarenador	m	1.5
Velocidad horizontal Promedio	m/s	0.04
Tiempo de retención	s	37,5

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

La geometría efectuada en Salome de los casos se basó en las especificaciones de diseño del modelo real. Sin embargo, existieron ciertas modificaciones en la interfaz gráfica, dado que, OpenFoam arrojaba errores en el reconocimiento de las estructuras internas de la canaleta parshall y de desarenador. Para lo cual, se trabajó con sólidos completos y realizando cortes de las estructuras internas para que facilite el reconocimiento de la función de cada una de las caras en los distintos casos.

Mediante la definición del dominio y las condiciones de borde, se logró obtener una convergencia a los 60 s en la canaleta parshall y 80 segundos en el desarenador.

Principalmente para la configuración de condiciones de borde se tomó en cuenta los parámetros de velocidad. Para esto, se estableció distintas regiones (inlet, outlet, atmosphere y walls) para definir el comportamiento hidráulico en los tanques.

Tanto el caso de la canaleta parshall y del desarenador en la entrada y la salida se definió como modelo despresurizado donde se determinó una velocidad de flujo uniforme de 0,26 m/s y 0,32 m/s, la cual fue calculada a partir de la velocidad de Manning considerando el caudal de diseño de 40 l/min y sus respectivas dimensiones. Esta fue utilizada como velocidad de partida en la simulación.

Dentro del archivo "alpha water" se definió un valor de 0.5 en la entrada de la canaleta parshall ya que el agua no entra por toda la sección transversal de tubería. El resto de los archivos k, nut, omega y p_rgh no se realizó ninguna modificación.

En la simulación de la canaleta parshall se obtuvo velocidades en el eje x (U_x) en un rango de 0,025 m/s a 0,3 m/s encontrando las velocidades más altas en los extremos y existiendo transiciones en las distintas zonas del canal lo que podría dificultar la entrada uniforme a la siguiente unidad de tratamiento.

En la simulación del desarenador se evaluaron la velocidad en el componente x obteniendo un rango de 0,01 m/s a 0.05 m/s y en el componente z con un rango menor a 0.05 m/s a 0.35 m/s. Lo cual evidencia una trayectoria irregular y variaciones en la velocidad del desarenador lo que podría producir resaltos hidráulicos que dificulten la sedimentación de las partículas y podría aumentar el tiempo de residencia.

El tiempo de retención fue calculado en función de la distancia y la velocidad en el eje x obteniendo 0.48 segundos para la canaleta parshall y 37,5 segundos para el desarenador en comparación con el tiempo de retención teórico obteniendo 0.58 segundos para la canaleta parshall y 41 segundos para el desarenador.

5.2 Recomendaciones

Se recomienda verificar que OpenFoam reconozca todas regiones establecidas para que no existan errores al momento de ejecutar la herramienta “blockMesh” y “snappyHexMesh”

En futuros casos de simulación, sería aconsejable trabajar en una unión de las estructuras en la simulación para obtener una perspectiva de cómo trabajan simultáneamente estas unidades ya que la fase preliminar en el tratamiento de aguas residuales es primordial para garantizar la efectividad y prevenir daños en los tratamientos siguientes. Puesto, que no se logro realizar por el lento procesamiento del ordenador, es aconsejable utilizar varios procesadores y correr el caso en paralelo.

REFERENCIAS

- Aldama, Á. A., & Ocón, A. R. (2002). Resistencia al flujo en canales y límites de aplicabilidad de la fórmula de Manning. In *Ingeniería Hidráulica en México* (Vol. 17).
- Ali, Z., Tyacke, J., Watson, R., Tucker, P. G., & Shahpar, S. (2019). Efficient preprocessing of complex geometries for CFD simulations. <https://doi.org/10.1080/10618562.2019.1606421>
- Amotey, P., & Bani, R. (2011). *Wastewater Management*. <https://doi.org/10.4324/9781351179430-5>
- Cardiff, P. (2017). Introduction to Meshing in OpenFOAM. In *University College Dublin*. Dublin.
- Chambergó, J. C., Valverde, Q., Pachas, A. A., & Yépez, H. (2017). Estudio del comportamiento fluido-dinámico de un agitador a escala reducida mediante simulación numérica. In *Información Tecnológica* (Vol. 28). <https://doi.org/10.4067/S0718-07642017000300005>
- European Environment Agency. (2018). *Industrial waste water treatment pressures on environment*. Recuperado el 23 de marzo de 2020 de [https://www.eea.europa.eu/publications/industrial-waste-water-treatment-pressures%0Ahttp://files/558/Rapport EEA Industrial waste water treatment – pressures on Europe’s environment.pdf](https://www.eea.europa.eu/publications/industrial-waste-water-treatment-pressures%0Ahttp://files/558/Rapport%20EEA%20Industrial%20waste%20water%20treatment%20pressures%20on%20Europe's%20environment.pdf)
- Fierro, S. (2018). *Elaboración de un modelo experimental de la fase preliminar de una planta tratamiento de aguas residuales* (Vol. 2). Universidad de las Américas.
- Fonseca, E., Pelliccioni, O., Tim, O y Aquite, W. (2014). Simulación del llenado de moldes con el uso de OpenFOAM. In *Jornadas de Investigación Encuentro Académico Industrial*. Caracas.

- Goyena, R., & Fallis, A. . (2019). Cálculo de la longitud mínima de aproximación para una canaleta parshall (Universidad Católica de Colombia; Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>
- Greenshields, C. J. (2019). *OpenFOAM*.
- Gullberg, R. (2017). *Computational Fluid Dynamics in OpenFOAM Mesh Generation and Quality*.
- Herreras, N. (2013). Two-Phase pipeflow simulations with OpenFoam. Norwegian University of Science and Technology.
- Holzinger, G. (2020). *OpenFoam A little User-manual*.
- Hovány, L., & Gabrić, O. (2014). Parshall Flume Calibration. In *Zbornik radova Građevinskog fakulteta* (Vol. 30). <https://doi.org/10.14415/konferencijagfs2014.087>
- INEC. (2016). *Estadística de Información Ambiental Económica en Gobiernos Autónomos Descentralizados* (Vol. 2015). Retrieved from www.ecuadorencifras.gob.ec
- Juárez, I. (2017). *Calibración de vertedero proporcional o sutro* (Universidad de San Carlos de Guatemala; Vol. 3). <https://doi.org/10.1017/CBO9781107415324.004>
- Khandve, P. (2016). *Application of CFD in Environment Engineering*.
- Lopera, A. (2010). *Manual De Operación Y Mantenimiento De Tirolesa*. Recuperado el 2 de mayo de 2020 <https://studylib.es/doc/7599167/manual-de-operación-y-mantenimiento-de-tirolesa>
- Maldonado, J. (2018). *Análisis matemático y simulación numérica del comportamiento de fluidos en un caldero pirotubular vertical*. Universidad Politécnica Salesiana.
- Metcalf and Eddy, I. (2014). *Wastewater Engineering (FIFTH EDIT)*. Boston: McGraw-Hill.

- Moukalled, F., Mangani, L., & Darwish, M. (2016). The finite volume method in computational fluid dynamics: An Advanced Introduction with OpenFOAM and Matlab. In *Fluid Mechanics and its Applications* (Vol. 113). https://doi.org/10.1007/978-3-319-16874-6_21
- Muttenthaler, L. (2017). *Scripted CFD simulations and postprocessing in Fluent and ParaVIEW*. <https://doi.org/10.13140/RG.2.2.11645.41446>
- Organización de Naciones Unidas (2018). Agua Limpia Y Saneamiento. Recuperado del 4 de marzo de 2020 de http://www.un.org/%0Ahttp://www.un.org/sustainabledevelopment/es/wp-content/uploads/sites/3/2016/10/6_Spanish_Why_it_Matters.pdf
- Pedroza, E y Hurtado, J.(2012). Variaciones en la calibración de un aforador Parshall. In *Tecnología y Ciencias del Agua* (Vol. 3). México DF.
- Raman, R. K., Dewang, Y.,y Raghuwanshi, J. (2018). A review on applications of computational fluid dynamics. In *International Journal of LNCT* (Vol. 2).
- Ramón, L. (2010). Infraestructura Sustentable: Las Plantas De Tratamiento De Aguas Residuales. In *Quivera* (Vol. 12).
- Reyes, J. (2019). *Estudio hidrodinámico de reducción de la resistencia al avance mediante la variación en la separación de cascós en una geometría a través de técnicas CFD*. Universidad Politécnica de Cartagena.
- Ruiz, K. (2015). *Análisis multidimensional de la influencia del techo en el comportamiento de recintos*. Universidad Industrial Santander.
- Salinas, H., García, J., y Díaz, C. (2008). Experimental analysis of turbulence effect in settling velocity of suspended sediments. In *Ingeniería, investigación y tecnología* (Vol. 9). Recuperado el 3 de junio de 2020 de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-77432008000100004

Salome Platform. (2014). Salome Documentation. Recuperado el 5 de junio de 2020 de <https://www.salome-platform.org/user-section/documentation/current-release>

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. In *Water Science and Technology* (Vol. 74). <https://doi.org/10.2166/wst.2016.249>

Tič, V., & Lovrec, D. (2012). Design of modern hydraulic tank using fluid flow simulation. In *International Journal of Simulation Modelling* (Vol. 11). [https://doi.org/10.2507/IJSIMM11\(2\)2.202](https://doi.org/10.2507/IJSIMM11(2)2.202)

Zuo, W. (2015). *Introduction of Computational Fluid Dynamics*. St Petersburg.

