



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

Simulación dinámica de fluidos computacional (CFD) del tanque lamella de  
la planta piloto de UDLA PARK

AUTOR

Nicolás Alejandro Argüello Prócel

AÑO

2020



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

Simulación dinámica de fluidos computacional (CFD) del tanque lamella de la  
planta piloto de UDLA PARK

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

Profesor guía

M. Sc. Santiago Daniel Piedra Burgos

Autor

Nicolás Alejandro Argüello Prócel

Año

2020

## DECLARACIÓN DEL PROFESOR GUÍA

"Declaro haber dirigido el trabajo, Simulación dinámica de fluidos computacional (CFD) del tanque lamella de la planta piloto de UDLA PARK, a través de reuniones periódicas con el estudiante Nicolás Alejandro Argüello Procel, en el semestre 2020-2, 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".

A handwritten signature in blue ink, appearing to be 'Santiago Daniel Piedra Burgos', written in a cursive style.

Santiago Daniel Piedra Burgos

Master of Science in Hydro Science and Engineering

C.I. 1715384150

## DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, Simulación dinámica de fluidos computacional (CFD) del tanque lamella de la planta piloto de UDLA PARK, del estudiante Nicolás Alejandro Argüello Procel, en el semestre 2020-2, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".



---

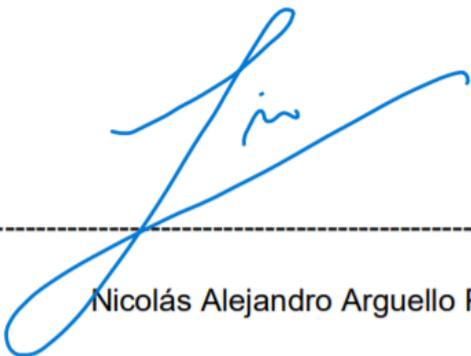
Daniel Hernán Hidalgo Villalba

Magister en Ciencias Especialización Ingeniería Ambiental

C.I.1801914449

## DECLARACIÓN DE 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.”



-----

Nicolás Alejandro Arguello Prócel

C.I. 1716761844

## **AGRADECIMIENTOS**

Agradezco a toda mi familia y amigos por su constante apoyo y comprensión.

A mi docente guía Santiago Piedra y mi docente corrector Daniel Hidalgo por compartir sus conocimientos y experiencias.

## **DEDICATORIA**

A mi familia.

A "Los C".

A Carla Oñate.

## Resumen

En el presente proyecto, se realizó la simulación CFD de una tanque lamella a través del software de código abierto Open FOAM, utilizando el algoritmo de solución PIMPLE y el solver interFoam en conjunto con las herramientas MPI. Se utilizó el software AutoCAD para representar la geometría del tanque, este tiene una tubería de entrada de 2 pulgadas de diámetro y una tubería de salida de 1 pulgada de diámetro, una altura de 0.82 metros, tiene 6 placas con una inclinación de 65 grados y una distancia de 0.06 metros entre cada placa y la tolva tiene una altura de 0.18 metros y una inclinación de 35 grados. Posteriormente se representó esta geometría en el software SALOME, donde adicionalmente se generó un mallado simple y se definieron las diferentes caras (entrada, salida, atmosfera y paredes). Se exportó la geometría al software Open FOAM, se definieron las condiciones de borde del sistema y se refinó la malla a través de la utilidad snappyHexMesh. La simulación duró 150 segundos y mediante las diferentes herramientas de Open FOAM, se determinó que el tanque lamella entra en estado estacionario después de 90 segundos de simulación, el tiempo de retención hidráulica promedio es de 2.25 minutos y que el tanque remueve el 61% de las partículas en el agua residual.

## Abstract

In the presented project, a CFD simulation of a lamellar tank was performed through the open source software Open FOAM, using the PIMPLE solution algorithm and the interFoam solver in conjunction with MPI tools. Through the AutoCAD tools the geometry of the tank was represented, it has a 2-inch diameter inlet pipe and a 1-inch diameter outlet pipe, a height of 0.82 meters, it has 6 plates with an inclination of 65 degrees and a distance of 0.06 meters between each plate and the hopper has a height of 0.18 meters and a 35 degree tilt. Then the geometry was represented in the SALOME software, where additionally a simple mesh was generated for it and the different faces (entrance, exit, atmosphere, and walls) were defined. The geometry was then exported to the Open FOAM software, the system boundary conditions were defined, and the mesh was refined through the snappyHexMesh utility. The simulation lasted 150 seconds and using the different Open FOAM tools, it was determined that the lamellar tank enters on steady state after 90 seconds of simulation, the average hydraulic retention time is 2.25 minutes, and that the tank removes 61% of the solids on the waste water.

## Contenido

1.	Introducción .....	1
1.1.	Antecedentes.....	1
1.2.	Objetivos.....	4
1.2.1.	Objetivo General.....	4
1.2.2.	Objetivos Específicos .....	4
1.3.	Justificación .....	4
1.4.	Alcance.....	6
2.	Marco Teórico.....	7
2.1.	Sedimentación.....	7
2.1.1.	Sedimentación tipo 2 .....	7
2.2.	Tanque lamella .....	7
2.3.	Computational Fluid Dynamics .....	8
2.3.1.	Pre-procesamiento .....	9
2.3.2.	Procesamiento.....	9
2.3.3.	Post-procesamiento.....	9
2.3.4.	Ecuaciones gobernantes de flujo de fluidos .....	10
2.4.	AutoCAD.....	11
2.5.	SALOME.....	11
2.5.1.	NETGENPLUGIN .....	11
2.5.2.	Archivo STL .....	11
2.5.3.	Archivo UNV .....	11
2.6.	OpenFOAM .....	12
2.6.1.	Carpetas base de casos en OpenFOAM .....	12
2.6.2.	Tipos de parches geométricos.....	14
2.6.3.	Algoritmo PIMPLE .....	15
2.6.4.	BoundingBox .....	17
2.6.5.	Comandos OpenFOAM .....	17
2.6.6.	Número de Courant .....	21
3.	Metodología.....	21
3.1.	Preprocesamiento .....	22

3.1.1.	Generación de Geometría en AutoCAD .....	22
3.1.2.	Generación de Geometría en SALOME .....	22
3.1.3.	Generación del mallado de la geometría en SALOME .....	23
3.1.4.	Generación de archivos STL y archivos UNV .....	26
3.1.5.	Generación de mallado hexaedrico a través de <i>snappyHexMesh</i> .....	26
3.1.6.	Bounding Box .....	28
3.1.7.	Línea de comandos <i>snappyHexMesh</i> .....	28
3.1.8.	Malla refinada .....	28
3.1.9.	Condiciones de borde.....	31
3.2.	Procesamiento.....	33
3.2.1.	Línea de comandos <i>interFoam</i> .....	33
3.3.	Post-procesamiento.....	34
4.	Resultados y discusión .....	34
4.1.	Comparación en procesamiento con solver estándar y solver <i>mpiexec -n 4</i> .....	34
4.2.	Análisis de convergencia .....	35
4.3.	Análisis de velocidades .....	36
4.4.	Determinación de tiempo de retención hidráulico (TRH)..	39
5.	Conclusiones .....	42
6.	Recomendaciones .....	43
7.	Referencias.....	44

## **1. Introducción**

### **1.1. Antecedentes**

Los primeros intentos de gestión y manejo de aguas residuales ocurrieron entre los años 3500 – 2500 A.C., donde se tenía un simple sistema de recolección y posterior aislamiento (Laughlin, 1999). Con el pasar de los años, el tratamiento de aguas residuales se tornó en un tema más importante, con la revolución industrial llegó un aumento exponencial en el interés de la gestión y tratamiento de aguas residuales (Lofrano & Brown, 2010).

Con el crecimiento demográfico y el aumento de la contaminación (en los años 1800s), la calidad del agua comenzó a tener una mayor exigencia por las autoridades debido a su fuerte relación con la salud pública (Lofrano & Brown, 2010; Shaikh, Ziauddin Ahammad, David Graham, 2013). En la actualidad, el tratamiento de aguas residuales domesticas ha evolucionado, tanto en su amplitud, como en su complejidad, existen estructuras especializadas para este proceso, conocidas como plantas de tratamientos de aguas residuales o PTAR, estas están diseñadas para que cada nivel de tratamiento elimine un contaminante con un tamaño o estructura específico, el cual podría dañar al siguiente nivel de tratamiento (Lofrano & Brown, 2010). Los niveles de tratamiento se separan de la siguiente manera:

- **Pretratamiento:** El primera etapa en el tratamiento de aguas residuales tiene como propósito principal, remover objetos flotantes (como trapos, mazorcas de maíz, trozos de madera) y partículas minerales pesadas (arena y gravilla), para evitar que las tuberías de ingreso a la planta se tapen con solidos grandes, el cual permitirá que las siguientes etapas de la planta se desarrollen de manera efectiva (Mara, 2004). Algunos ejemplos de pretratamiento son, desarenadores, canales Parshal, canales Ventury y rejillas gruesas y delgadas (INEN, 1992; Mara, 2004)

- Tratamiento primario: En esta fase del tratamiento se da la remoción de sólidos pesados por sedimentación, esta es una herramienta utilizada en una gran cantidad de procesos de tratamientos de aguas residuales (Tchobanoglous, Burton, & Stensel, 2003). Existen diversos tipos de sedimentación, la sedimentación a través de la simple acción de la gravedad, la sedimentación asistida de un químico coagulante (coagulación floculación), flotación, entre otros (Ghawi, 2016; Tchobanoglous et al., 2003). Los tanques sedimentadores lamella se han convertido en una alternativa para realizar este proceso de manera efectiva, existen recientes estudios sobre este, los cuales demuestran que es un proceso sedimentador lamella, los cuales demuestran su efectividad tanto en la remoción de sólidos (la cual proviene de sus placas inclinadas, estas permiten que todo flujo turbulento entrante se vuelva laminar), como en costos (Tarpagkou & Pantokratoras, 2014; Weiss, 2016).
- Tratamiento secundario: En esta parte del tratamiento se puede utilizar el factor biológico como herramienta o se puede utilizar la vía química para eliminar los contaminantes más persistentes en el sistema, aunque ahora se considera al tratamiento químico como un tratamiento terciario o complementario (Samer, 2015). Los tratamientos químicos eliminan los contaminantes a través de reacciones como la precipitación, neutralización, adsorción química y desinfección. En un tratamiento biológico, se requiere que un consorcio microbiano crezca y esto se puede lograr a través de diferentes formas de proliferación, en lecho adjunto (a través del uso de bio carriers) o en lecho suspendido (lodos activados) (Halling-Sorensen & Jorgensen, 1993; Lofrano & Brown, 2010; Shaikh, Ziauddin Ahammad, David Graham, 2013).
- Tratamiento terciario: Conocidos también como tratamientos complementarios son utilizados para eliminar DBO residual, sólidos suspendidos, nutrientes inorgánicos y en algunos casos, la proliferación descontrolada de microorganismos (Moran, 2018). Existen varias tecnologías para tratar los puntos anteriormente expuestos, entre estos

está la nitrificación biológica, remoción biológica de fósforo, adsorción de carbono, desnitrificación, dosimetría de sales metálicas, filtración, camas de juncos y desinfección (química, física y por radiación) (Moran, 2018; Tchobanoglous et al., 2003).

El tratamiento de aguas residuales ha tenido un gran avance en los últimos años, sin embargo, se cree que siempre existirá un espacio para las mejoras, ya que el error humano siempre estará presente, y el tratamiento de aguas residuales no es una excepción, pudiendo existir muchos tipos de problemas, como por ejemplo, la formación de lodos filamentosos (bulking), bajos rendimientos, generación de malos olores y de espuma (Meriç, Kaptan, Övez, & Eser Okten, 2002). Se cree que algunos de estos problemas se pueden atribuir a contrariedades de origen hidráulico, sin embargo, subsanarlos puede llegar a ser una tarea particularmente intrincada, ya que hallar el origen de un problema hidráulico a simple vista es implícitamente imposible.

La tecnología siempre ha sido una herramienta invaluable para el ser humano, debido a que permite un análisis completo tanto de macroelementos como de microelementos. A principios del año 1973, un grupo de personas en Imperial College empezó el desarrollo de un programa que sea capaz de predecir el movimiento de un flujo simple, después de años de perfeccionamiento y transformaciones, las simulaciones dinámicas de fluidos computacional o por sus siglas en inglés CFDs, han atraído la atención de muchos diseñadores de plantas de tratamientos de aguas residuales recientemente, esta herramienta ha evolucionado de ser costosa, capaz de analizar únicamente geometrías de una y dos dimensiones con flujos de una dimensión y estáticos, a ser asequible, logrando el análisis tanto de sistemas como geometrías complejas (tridimensionales), flujos tridimensionales y con las configuraciones adecuadas, estos procesos se pueden desarrollar óptimamente (Jamshed, 2015; Khalil, 2012). Con esta herramienta, se pueden llegar a observar diferentes fenómenos que ocurren dentro de un tanque de una PTAR (Samstag et al., 2016), por ejemplo en un tanque sedimentador, una simulación CFD permite el análisis de

velocidades en cada eje del plano cartesiano, esto permite a su vez determinar la eficiencia del tanque, con ayuda de estudios complementarios tales como la determinación de las velocidades de sedimentación de las partículas dentro del agua residual, finalmente se podría determinar el tiempo de retención hidráulico o TRH, el cual afecta directamente a la eficiencia del proceso (Jordan, Harasek, Gößnitzer, Zibuschka, & Chen, 2018; Xanthos et al., 2011).

## **1.2. Objetivos**

### **1.2.1. Objetivo General**

- Simular el funcionamiento hidráulico del tanque lamella de la planta piloto de UDLA PARK.

### **1.2.2. Objetivos Específicos**

- Establecer las condiciones de borde del tanque lamella de la planta piloto de UDLA PARK
- Estimar el tiempo de residencia real en el tanque lamella de la planta piloto de UDLA PARK

## **1.3. Justificación**

El agua residual sin tratamiento puede ser un problema, no solamente ambiental (daño en la calidad del agua y consecuentemente daño en la biota acuática) sino también para los seres humanos (las aguas residuales domésticas pueden causar enfermedades relacionadas con materia fecal) (Mara, 2004). En América Latina, menos del 14% de las aguas residuales recolectadas por los sistemas de alcantarillado son tratadas antes de ser descargadas a un cuerpo de agua (Organización Panamericana de la Salud, 2011). Sin embargo, se ha observado con el pasar de los años que el tratamiento de aguas residuales ya es un tema muy trascendental, en Ecuador, se ha observado que algunos Gobiernos Autónomos Descentralizados (GADs) que tratan sus aguas residuales antes de descargarlos a un cuerpo hídrico aledaño. Según un estudio realizado por el INEC, se demuestra que casi el 62% de los GADs municipales realizar

tratamientos de aguas residuales, mientras que el restante 38% no lo realiza (INEC, 2016).

Para el diseño adecuado de una PTAR, son necesarios los cálculos de diseño como, el dimensionamiento del tanque, la cantidad de agua que ingresa al sistema, el tiempo de retención del líquido, entre otros factores (Wiesmann et al., 2007), sin embargo, se considera que la calidad de los diseños pueden ser perfeccionados en mayor grado, si se aplica uno de los siguientes métodos de verificación, se puede emplear un modelo físico del tanque a escala de laboratorio o una simulación Computational Fluid Dynamics (CFDs) (Versteeg & Woltering, 1990). El modelo físico se puede aplicar cuando se requiera evaluar el funcionamiento de un proceso de lodos activados, sin embargo, esto no suele ser viable, ya que la inversión requerida para llevarlo a cabo es muy alta y por lo tanto, el dinero invertido puede ser no recuperable en un corto periodo de tiempo (Prieto et al., 2016), adicionalmente se debe tomar en cuenta que no se pueden realizar muchas iteraciones del experimento, a menos que se cuente con el presupuesto necesario, mientras que en una simulación CFD, la inversión requerida es considerablemente menor comparada con un modelo físico, debido a que se requiere una menor cantidad de recursos económicos y humanos, por lo cual se pueden realizar un número indefinido de iteraciones según sean requeridas (Jamshed, 2015), estos métodos pueden ser aplicados de manera separada, no obstante, pueden ser utilizados en conjunto (Singh, Fu, Yang, & Xiong, 2019).

Los tanques sedimentadores primarios son muy conocidos en el campo del tratamiento de aguas ya que son utilizados en toda PTAR (Mara, 2004), por lo que los estudios con CFDs sobre sedimentadores primarios no es algo nuevo, adicionalmente las simulaciones CFD juegan un papel indispensable en su diseño, ya que se pueden analizar variables complejas que a simple vista no se pueden examinar, tales como pérdidas de carga, remoción de arena, velocidades de sedimentación, entre otras (Ganjare & Patwardhan, 2019; Xanthos et al., 2011). El uso de tanques sedimentadores tipo lamella ha aumentado considerablemente por su alto rendimiento y bajo costo monetario

(Weiss, 2016), sin embargo, se ha observado que no existen muchos trabajos investigativos que utilicen CFDs para analizar el funcionamiento de los mismos (Tarpagkou & Pantokratoras, 2014).

#### 1.4. Alcance

El presente proyecto, planea simular el funcionamiento hidráulico del tanque lamella perteneciente a la planta piloto de UDLA PARK, que está ubicada en el laboratorio de Ingeniería Ambiental de la Universidad de las Américas, localizada en la vía a Nayón. La planta piloto es el conjunto de varios trabajos de titulación realizados por los estudiantes de la universidad.



*Figura 1:* Planta piloto de UDLA PARK

La simulación del tanque se realizó el software OpenFOAM. Primero se proyectaron las dimensiones del tanque lamella en el software de diseño gráfico AutoCAD, las dimensiones del tanque fueron obtenidos en los planos de diseño del tanque, posteriormente se exportó la geometría al software SALOME donde a su vez se generó el mallado correspondiente a cada una de las caras del tanque (entrada, salida, atmosfera y paredes), se exportó el mallado al software OpenFOAM y a través de esta la misma herramienta, se refino el mallado anterior. Se definieron las condiciones de borde del sistema y las condiciones

iniciales de la simulación a través de un software de edición de texto. Finalmente se observaron los resultados en el software de visualización ParaView.

## **2. Marco Teórico**

### **2.1. Sedimentación**

El proceso de sedimentación de partículas suspendidas o coloidales en aguas residuales por gravedad es una de las operaciones físicas más usadas en el tratamiento de aguas residuales (Tchobanoglous et al., 2003). En esta operación intervienen algunas variables, tales como la velocidad de sedimentación de la partícula, fuerza de arrastre, las dimensiones del tanque de sedimentación y la velocidad crítica de sedimentación (Tchobanoglous et al., 2003). Dentro de este proceso pueden existir cuatro tipos de sedimentación por gravedad:

1. Sedimentación de partículas discretas (sedimentación tipo 1)
2. Sedimentación de partículas floculantes (sedimentación tipo 2)
3. Sedimentación obstaculizada (sedimentación tipo 3)
4. Sedimentación comprimida (sedimentación tipo 4)

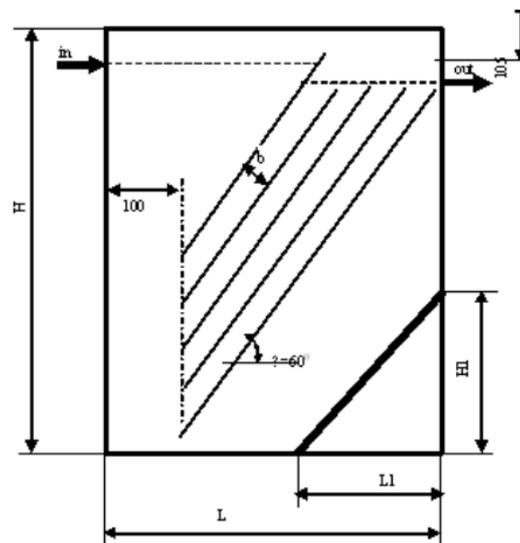
(Ghawi, 2016)

#### **2.1.1. Sedimentación tipo 2**

Típicamente la sedimentación funciona únicamente con la gravedad y el peso de las partículas suspendidas en el agua, sin embargo, existen partículas más pequeñas y por consiguiente más livianas, por lo que su sedimentación por simple acción de la gravedad no es posible. Una metodología que permite la sedimentación de partículas livianas consiste en agrupar estas partículas antes de que entren al tanque sedimentador, esto se realiza a menudo con la adición de coagulantes - floculantes, los cuales permiten la aglomeración de las partículas, facilitando su sedimentación (Kang, 2015).

## **2.2. Tanque lamella**

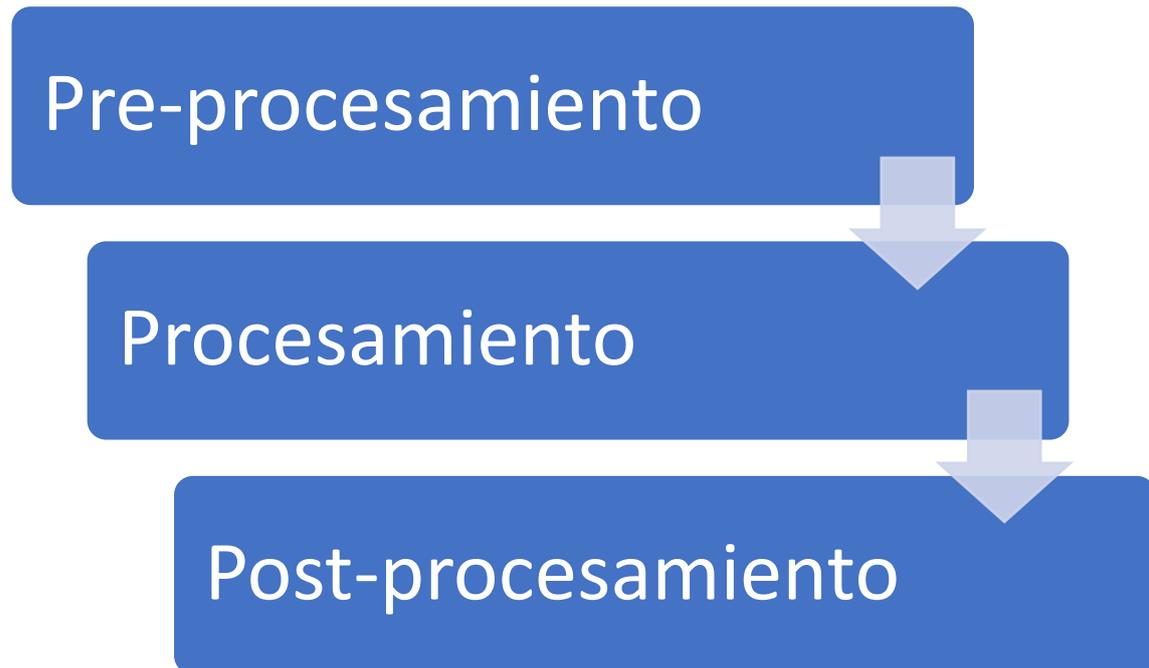
Los tanques lamella se han convertido en tanques de uso regular en los tratamientos de aguas residuales, se ha observado su uso como tanques de sedimentación primaria, secundaria y terciaria. La teoría bajo los tanques lamella es que el área de sedimentación afecta a la eficiencia de sedimentación, más que el tiempo de retención hidráulico (Robescu, Mandiş, & Robescu, 2010). El funcionamiento del tanque lamella se basa en el incremento de la capacidad de los tanques, a través del uso de placas inclinadas en un ángulo predeterminado, se recomienda una inclinación 55 grados (Robescu et al., 2010; Tchobanoglous et al., 2003).



*Figura 2:* Bosquejo de tanque de sedimentación lamella.  
Adaptado de (Robescu et al., 2010)

### 2.3. Computational Fluid Dynamics

Conocido ahora como CFD, es el conjunto de metodologías que permite la simulación numérica de flujo de fluidos, se la reconoce como una simulación ya que se utiliza a la computadora como herramienta para resolver numéricamente las leyes involucradas en la mecánica de fluidos (Hirsch, 2007). El modelamiento matemático del flujo de fluidos se puede separar en tres fases principales:



*Figura 3:* Diagrama de procesos para la elaboración de un CFD.

Adaptado de (Moukalled, Mangani, & Darwish, 2016)

### **2.3.1. Pre-procesamiento**

En esta fase se utilizan las aplicaciones CAD (Computer-Assisted Design), para la definición de la geometría (siendo el más común el AutoCAD) y las especificaciones del caso (Hirsch, 2007). En esta fase también se define el mallado (también conocido como discretización del dominio) que se aplicara a la geometría, además de las condiciones de borde del sistema (Moukalled et al., 2016).

### **2.3.2. Procesamiento**

Dentro de esta fase, se utiliza el software especializado para resolver numéricamente las ecuaciones gobernantes de sistema (el comportamiento del agua dentro de la geometría definida anteriormente) (Hirsch, 2007; Moukalled et al., 2016).

### **2.3.3. Post-procesamiento**

Finalmente se analizan los resultados, típicamente con la ayuda de software de visualización o renderización del agua alrededor del sistema (Haider, 2018; Hirsch, 2007). En el presente proyecto, se utilizarán las aplicaciones de visualización ParaView y Gnuplot.

#### 2.3.4. Ecuaciones gobernantes de flujo de fluidos

La mayoría de las ecuaciones utilizadas en las simulaciones de dinámica de fluidos pertenecen al grupo de ecuaciones diferenciales parciales de segundo orden, conocidas como las ecuaciones de Navier-Stokes, las cuales son utilizadas comúnmente para describir el flujo de un fluido (Hirsch, 2007). Estas ecuaciones pueden describir un sistema tridimensional, siendo la Ecuación 1, la representación del balance de fuerzas en el eje X, donde el termino al lado izquierdo de la ecuación representa la inercia, el primer término del lado derecho de la ecuación representa el gradiente de presión, el segundo término representa la viscosidad del fluido y el ultimo termino representa las fuerzas del fluido involucrado (Eslahpazir, Krull, & Krühne, 2019), a su vez las Ecuaciones 2 y 3 representan los balances en los ejes Y y Z respectivamente.

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

Ecuación 1

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

Ecuación 2

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

Ecuación 3

Donde:

U, V y W: Representan las velocidades en los diferentes ejes X, Y y Z respectivamente

$\rho$ : Densidad

P: Presión

$\mu$ : Viscosidad

## **2.4. AutoCAD**

AutoCAD es una herramienta desarrollada por Auto Desk, en donde se pueden crear geometrías en dos y tres dimensiones. A través de sus diferentes herramientas, se pueden generar planos a partir de la geometría instaurada (Autodesk, 2010).

## **2.5. SALOME**

Se describe a SALOME como un software de código abierto que ofrece una plataforma genérica para el pre-procesamiento de simulaciones numéricas. En este software se pueden generar geometrías y el mallado de las mismas, adicionalmente en este software se pueden generar los archivos STL y UNV de las diferentes caras de la geometría (Velten, 2010).

### **2.5.1. NETGENPLUGIN**

Esta herramienta del software SALOME le permite al usuario generar mallados de entidades geométricas de una, dos o tres dimensiones. El algoritmo de NETGEN, se encarga de convertir caras en elementos triangulares y salidos en elementos tetraédricos (Open Cascade, 2007).

### **2.5.2. Archivo STL**

También conocidos como StereoLithography files, son archivos desarrollados por sistemas de modelamiento en tres dimensiones, los cuales contienen el mallado (conformado por únicamente triángulos) de la superficie de un sólido tridimensional, este tipo de archivos se han convertido en un estándar de descripción de las diferentes caras de un objeto tridimensional en todo sistema de representación gráfica, el archivo STL debe definir qué lado de la superficie representada es la que enfrenta al interior del objeto (Szilvási-Nagy & Mátyási, 2003).

### **2.5.3. Archivo UNV**

En el software de creación de geometrías y mallado llamado SALOME, se puede exportar la geometría en este formato universal, el cual permite que OpenFOAM pueda representar gráficamente la geometría, editar y observar las diferentes caras de la misma, analizar sus condiciones de borde y analizar sus propiedades (Schulz, Andrade, & Lobosco, 2011)

## **2.6. OpenFOAM**

OpenFOAM es una herramienta de código abierto que permite el desarrollo de aplicaciones ejecutables que utilizan funcionalidades contenidas en aproximadamente 100 directorios C++. Este software contiene aproximadamente 250 aplicaciones que pueden caer en dos categorías: solvers, los cuales están diseñados para resolver un problema específico dentro de la mecánica de fluidos, o una utilidad, la cual está diseñada para realizar una tarea que implique la manipulación de datos (Greenshields, 2018).

### **2.6.1. Carpetas base de casos en OpenFOAM**

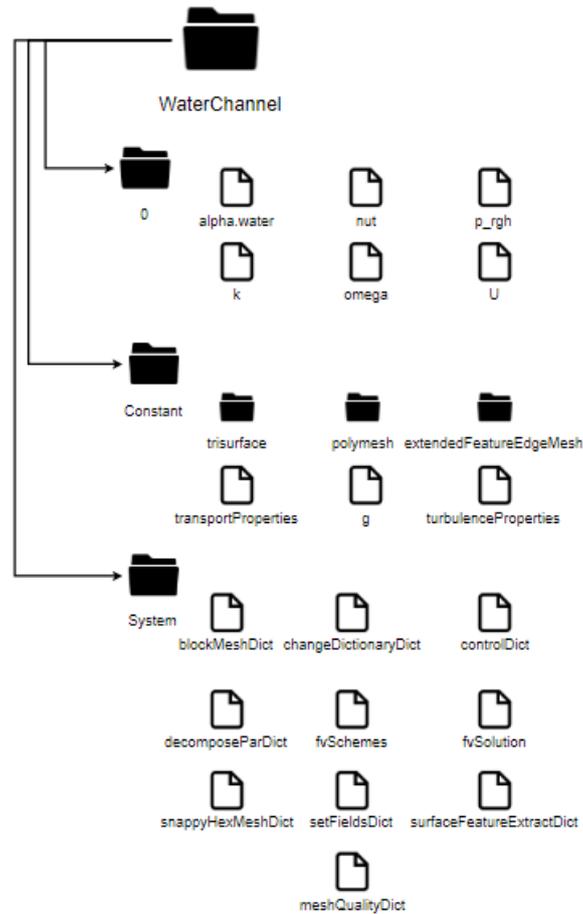


Figura 4: Estructura básica de casos en OpenFOAM.

Adaptado de (Jordan, Harasek, Zibuschka, & Chen, 2018).

En la carpeta 0 se describen las condiciones iniciales de la simulación. Cada uno de los archivos describe una propiedad y puede ser ajustada a las condiciones requeridas, por ejemplo, el archivo `alpha.water` define la interfase entre agua y gas, el archivo `U` controla la velocidad del fluido dentro del sistema (Greenshields, 2018). La carpeta *constant* suele consistir en varios subdirectorios (`trisurface`, `polyMesh` y `extendedFeatureEdgeMesh`) y algunos archivos (`g`, `transportProperties` y `turbulenceProperties`), estos describen las propiedades de algunos materiales, la física y la química envueltas en la simulación, por ejemplo, en el subdirectorio `trisurface` se almacenan los archivos STL de las caras de la geometría, al igual que el archivo STL del volumen entero. Finalmente, en la carpeta *system*, en este directorio se definen el solver y el método de volúmenes finitos a utilizar, a través de los tres archivos principales:

- *fvSchemes*: En este archivo se definen los esquemas de discretización a utilizar en la resolución de ecuaciones en el caso.
- *fvSolution*: Se realizan ajustes a los métodos de acoplamiento de presiones y velocidades, los métodos numéricos para resolver diferentes cantidades y determinar tolerancias para la convergencia de datos.
- *controlDict*: Se especifica el momento temporal en el cual empieza la simulación (*startFrom*), el momento temporal en el cual termina la simulación (*stopAt*), el escalón temporal (*deltaT*), intervalo de almacenamiento de datos (*writeInterval*), formato de almacenamiento de datos (*writeFormat*), precisión de almacenamiento de datos (*writePrecision*) y definir si al cambiar algún parámetro mencionado anteriormente afecta a la simulación (*runTimeModifiable*). (Jordan, Harasek, Zibuschka, et al., 2018).

## 2.6.2. Tipos de parches geométricos

*Patch*: Tipo genérico que no contiene información geométrica o topológica sobre el mallado, usado típicamente en la entrada y salida del sistema (Greenshields, 2018).

*Wall*: Para los parches que son considerados como paredes sólidas, requeridas para algunos modelos físicos, como modelos turbulentos (Greenshields, 2018).

### 2.6.2.1. Valores en parches

Cada grupo de celdas de la geometría debe pertenecer a un tipo de parche y a un tipo de entrada, con el objetivo de definir la resolución numérica del sistemas a simular (Greenshields, 2018).

*fixedValue*: El parámetro esta fijo en un valor definido por el usuario

*fixedGradient*: El gradiente normal se encuentra especificado

*zeroGradient*: El gradiente es igual a cero

(Greenshields, 2018).

### 2.6.3. Algoritmo PIMPLE

Este algoritmo es una combinación entre el algoritmo SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) y PISO (Pressure implicit with Splittinf of Operations). Como se puede observar el diagrama de flujo, se utilizan un esquema Euleriano implícito de primer orden (Garcia-Alcaide et al., 2017). Este algoritmo puede aplicarse con numero de Courant altos, a su vez, realiza iteraciones y sub-relajamiento en la resolución en relaciones entre presión-velocidad dentro de un tiempo establecido, el algoritmo termina cuando existe un numero de residuales mínimos (Vita & Atter, 2016).

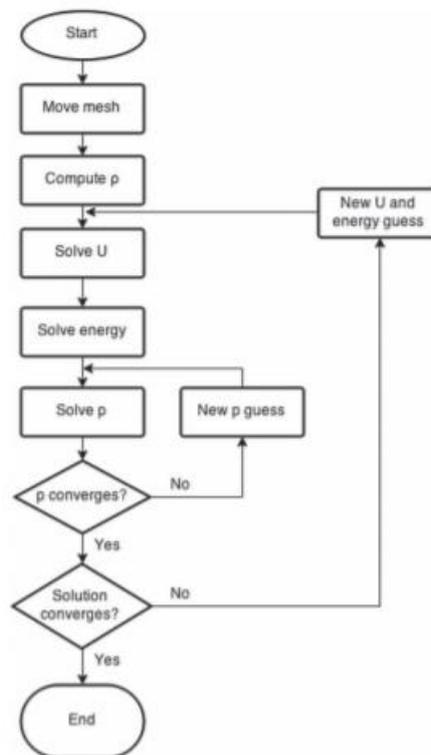


Figura 5: Diagrama de flujo de algoritmo PIMPLE.

Adaptado de (Garcia-Alcaide et al., 2017).

#### 2.6.3.1. Algoritmo SIMPLE

Este algoritmo realiza un procedimiento predictor-corrector con un corrector sucesivo de presiones a través de la siguiente ecuación (Ecuación 4):

$$p = \bar{p} + p'$$

Ecuación 4

Donde:

p: Presión real

 $\bar{p}$ : Presión estimada

P': Presión corregida

Igualmente se pueden realizar estas correcciones en las velocidades en los ejes de coordenadas en un sistema bidimensional:

$$u = \bar{u} + u'$$

Ecuación 5

$$v = \bar{v} + v'$$

Ecuación 6

Donde:

u: Velocidad en el eje X

v: Velocidad en el eje Y

Se pueden relacionar las ecuaciones 4 y 5 con la ecuación 3 y substituyendo el resultado en la ecuación de continuidad, teniendo como resultado a la ecuación de corrección de presiones de Poisson de una forma elíptica (Chung, 2002).

$$p'_{.ii} = -\frac{\rho}{\Delta t} \left( \frac{\partial v_i}{\partial x_i} - \frac{\partial \bar{v}_i}{\partial x_i} \right) = \frac{\rho}{\Delta t} \frac{\partial \bar{v}_i}{\partial x_i}$$

Ecuación 6

### 2.6.3.2. Algoritmo PISO

Se observó que el algoritmo SIMPLE requiere un proceso iterativo extenso, por lo que se desarrolló este algoritmo el cual puede obtener soluciones sin la necesidad de realizar iteraciones y minimizando el esfuerzo computacional. En este procedimiento la conservación de masa está diseñada para ser resuelta dentro de las ecuaciones de predicción y corrección, las ecuaciones gobernantes dentro de este algoritmo son la ecuación de momento (Ecuación 7) y la ecuación de corrección de presiones (Ecuación 8) (Chung, 2002):

$$\frac{\rho}{\Delta t} (v_j^{n+1} - v_j^n) = -s_{i,j,i}^{n+1} - p_j^{n+1}$$

Ecuación 7

$$p_j^{n+1} = -\frac{\rho}{\Delta t} (v_j^{n+1} - v_j^n) - s_{i,j,i}^{n+1}$$

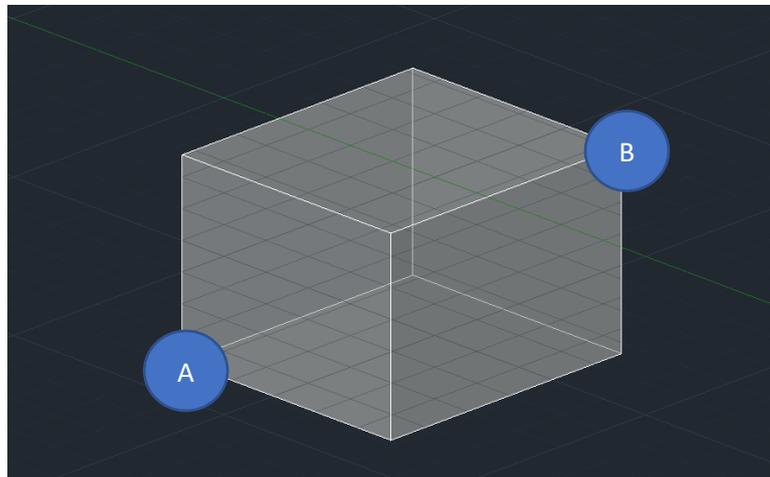
Ecuación 8

Donde:

$s_{i,j,i}$ : Se refiere a la suma de las derivadas de los términos de convección y viscosidad

#### 2.6.4. BoundingBox

Se define como un bloque en el cual se encuentran los límites de la geometría, esta se representa a través de dos coordenadas, como se observa en la *Figura 6*:



*Figura 6*: Representación gráfica del concepto de BoundingBox.

Adaptado de (Greenshields, 2018).

#### 2.6.5. Comandos OpenFOAM

##### 2.6.5.1. Solver interFoam

Este solver es un algoritmo de solución de dos fases, está basado en el método de volumen del fluido o Volume of Fluid (VOF), el cual utiliza ecuaciones de transporte para determinar las fracciones de volumen relativo de las dos fases en cada celda individual. La lógica tras el método VOF es que la interfase entre las dos fases no es calculada, sino que aparece como una propiedad de las fases (Greenshields, 2018).

### 2.6.5.2. **ideasUnvToFoam:**

Este comando perteneciente al software OpenFOAM, el cual convierte un archivo I-deas UNV a formato legible por OpenFOAM. Adicionalmente, crea una carpeta titulada *polymesh* en donde se encuentran diversos archivos que describen a la geometría en términos de caras, cada cara es asignada a una celda “owner” y una celda “neighbour”, con el objetivo de que cada cara pueda ser descrita por owners y neighbours (Greenshields, 2018).

#### 2.6.5.2.1. **Archivos en polymesh**

*Tabla 1*

Archivos contenidos en carpeta *polymesh*

Adaptado de (Greenshields, 2018).

<b><i>polymesh</i></b>	
<b><i>Título archivo</i></b>	<b><i>Descripción</i></b>
<b><i>points</i></b>	Una lista de vectores cuyas coordenadas describen los vértices de cada celda
<b><i>faces</i></b>	Una lista de caras, cada lista contiene los vectores que conforma la cara descrita
<b><i>owner</i></b>	Una lista con etiquetas de “owner”, cada una de estas etiquetas pertenece a una sola cara
<b><i>neighbor</i></b>	Una lista de etiquetas con grupos de celdas
<b><i>boundary</i></b>	Una lista de parches, cada parche tiene un directorio asignado con el mismo nombre del parche

#### 2.6.5.3. **DecomposePar**

Este comando realiza una partición del mallado generado en diferentes partes, según fue definido por el usuario en el diccionario de OpenFOAM titulado *decomposeParDict*. Dentro de este archivo existen dos apartados principales, el primer apartado *numberOfSubdomains* es en donde se definen el número de procesadores que posee la computadora y el segundo apartado *method* es en donde se define el método con el que se va a repartir la geometría y la subsecuente asignación de los diferentes procesadores definidos anteriormente, en la mayoría de estos métodos, se requeriría un mayor nivel de especificación en cuanto a la repartición de la geometría, para este apartado existen varias opciones o métodos: *Simple*, *Hierarchical*, *Scotch* y *Manual* (Greenshields, 2018).

#### **2.6.5.3.1. Scotch**

En este método no se requiere ningún tipo de especificación (a menos que el usuario desee especificar a que procesador se debe asignar mayores cantidades de la geometría), ya que aquí el algoritmo reparte la geometría de forma equitativa a cada procesador (Greenshields, 2018).

#### **2.6.5.4. snappyHexMesh**

El comando *snappyHexMesh* genera una malla alrededor de toda la geometría (utilizando únicamente hexaedros y hexaedros partidos) a partir de la malla de fondo utilizando como molde los archivos STL de las caras de la geometría, este es un método de refinamiento de malla, este procedimiento se puede separar en siete fases principales:

- Creación de malla hexaédrica: Se debe generar una malla de fondo que constituya únicamente de hexaedros y debe cubrir a toda la geometría. Se puede generar la malla en el directorio *blockMeshDict* o cualquier otra herramienta de generación de mallado, las proporciones de aspecto de cada celda debe ser de aproximadamente 1.

- División de celdas en bordes característicos: Las divisiones se realizan según lo especifique el usuario en el apartado `castellatedMeshControls` dentro del archivo `snappyHexMeshDict`, los bordes característicos pueden ser extraídos de los archivos STL de las caras de la geometría a través del comando `SurfaceFeatureExtract`.
- División de celdas en superficies: Las divisiones de las celdas (refinamiento) se realizan según lo especifique el usuario en el apartado `refinementMeshControls` dentro del archivo `snappyHexMeshDict`.
- Remoción de celdas: Una vez realizado la selección de los superficies y bordes empieza un proceso de remoción de celdas, según un punto definido por el usuario, el cual debe estar dentro de la región, este punto se define en el apartado `locationInMesh` del archivo `snappyHexMeshDict`.
- Refinamiento de celdas en regiones especificadas: Aquellas celdas que se encuentran dentro de una o más regiones de los volúmenes especificados, pueden ser refinados en mayor grado, el nivel de refinamiento puede ser especificado por el usuario dentro del apartado `refinementRegions`.
- Encajado en superficies: Una vez borradas las celdas seleccionadas, se empiezan a encajar los puntos dentro de las superficies para la generación de la malla conforme.
- Capas del mallado: La malla resultante puede ser apta para simulaciones, sin embargo, se pueden generar celdas irregulares, donde el usuario según sus propias especificaciones definidas en el apartado `addLayersControl`, puede añadir capas adicionales de mallado.

(Guerrero, 2018).

#### **2.6.5.5. SetFields**

Este comando lee el bloque definido en el directorio `setFieldsDict`, el cual define el nivel del agua con el que empieza el sistema, este nivel se define al igual que el `BoundingBox` (Greenshields, 2018).

#### **2.6.5.6. SurfaceFeatureExtract**

Este comando extrae los bordes cuyo ángulo sea menor al ángulo definido por el usuario en el directorio titulado *surfaceFeatureExtractDict*, en el apartado *includedAngle* (Greenshields, 2018).

#### **2.6.5.7. ReconstructPar**

Reconstruye un caso resuelto en paralelo (Greenshields, 2018).

#### **2.6.5.8. mpiexec -n 4 comando -parallel**

MPI o Message Passing Interface, es el conjunto de funciones y programas que facilitan la comunicación entre procesadores, el cual permite la ejecución de un programa en paralelo, o sea, repartir la ejecución de un comando entre los procesadores definidos por el usuario, típicamente esta colección de funciones se utilizan en programas C++ (Vargas-Félix, 2015).

#### **2.6.6. Número de Courant**

Este se define como un número de una dimensión, el cual mide cuanta información atraviesa una celda de una malla en un tiempo dado. Por ejemplo, dentro de un sistema resuelto con un método Euleriano, el número de Courant no puede ser mayor a 1, ya que esto implica que la información está pasando por más de una celda de la malla por cada paso temporal, lo cual afectaría a la estabilidad del modelo, este número es obtenido a través de la Ecuación 9 (Giraldo, 1986).

$$C = \frac{u\Delta t}{\Delta x}$$

Ecuación 9

Donde:

u: Velocidad característica de onda dentro del sistema

$\Delta t$ : Paso temporal del modelo numérico

$\Delta x$ : Espacio entre cada celda en la malla en el modelo numérico

### **3. Metodología**

### 3.1. Preprocesamiento

#### 3.1.1. Generación de Geometría en AutoCAD

Tomando en cuenta la complejidad de la geometría del tanque tipo lamella, se proyectaron primero las dimensiones del tanque en AutoCAD para extraer posteriormente los vértices necesarios para representar la geometría en SALOME (Anexo 1) (*Tabla 2*):

*Tabla 2*

Dimensiones del tanque lamella

<b>Parámetro</b>	<b>Valor</b>	<b>Dimensión</b>
<b>Espacio entre placas</b>	0.06	Metros
<b>Número de placas</b>	6	N/A
<b>Tubería entrada (diámetro)</b>	2	Pulgadas
<b>Tubería salida (diámetro)</b>	1	Pulgadas
<b>Largo de placas diagonales</b>	0.6	Metros
<b>Longitud total del tanque</b>	0.86	Metros
<b>Altura del tanque (sin tolva)</b>	0.82	Metros
<b>Altura tolva</b>	0.18	Metros
<b>Inclinación de las placas</b>	65	Grados
<b>Inclinación de la tolva</b>	35	Grados

#### 3.1.2. Generación de Geometría en SALOME

La geometría en SALOME se realizó a través de vértices basados en el modelo generado en AutoCAD. Una vez generada la geometría, se agrupan las diferentes caras para que cumplan un determinado papel en la simulación, siendo:

- Entrada o inlet: Cara o caras de la geometría por la que entra el agua.
- Salida u outlet: Cara o caras de la geometría por la que sale el agua.

- Atmosfera o atmosphere: Cara o caras de la geometría que no tienen interacción con el agua.
- Pared o walls: Cara o caras de la geometría con las que el agua interactúa.

### **3.1.3. Generación del mallado de la geometría en SALOME**

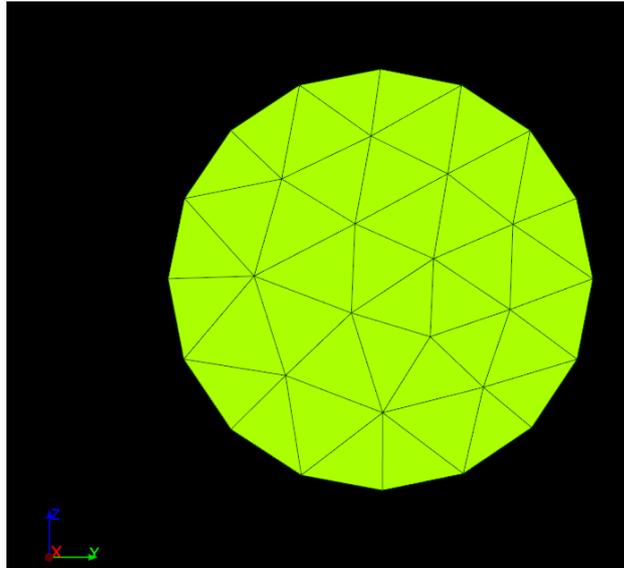
El mallado del tanque se realizó a través de los algoritmos de generación de mallado conocido como “NETGEN 1D – 2D - 3D” en donde se reconstruye cada cara de la geometría utilizando figuras geométricas según el algoritmo lo determine, SALOME genera triángulos, cuadrados, cuadriláteros, tetraedros y hexaedros (Open Cascade, 2007). Se puede especificar el número de figuras geométricas que conformaría la malla perteneciente a la geometría, sin embargo, para el presente trabajo, solo se requiere de una malla sencilla, así que se dejara las configuraciones que vienen por defecto.

#### **3.1.3.1. Malla resultante**

La malla generada por el software SALOME contiene únicamente triángulos, con un total de 40,492 triángulos.

##### **3.1.3.1.1. Entrada**

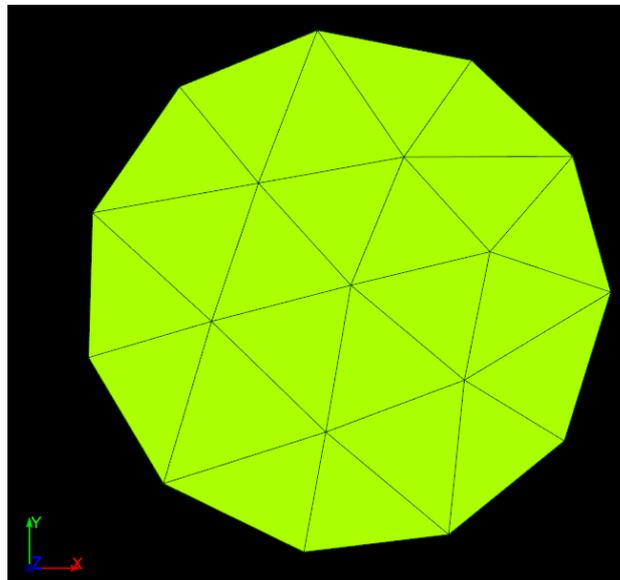
La cara “Entrada” está conformada únicamente por triángulos (40).



*Figura 7:* Malla SALOME de la cara “entrada”

### 3.1.3.1.2. Salida

Esta cara está conformada únicamente por triángulos (23).



*Figura 8:* Malla SALOME de la cara “salida”

### 3.1.3.1.3. Atmosfera

Esta cara está conformada únicamente por triángulos (1,145).

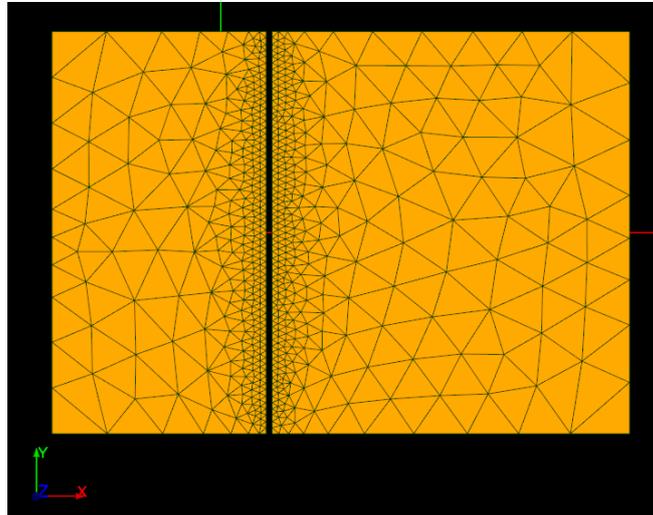


Figura 8: Malla SALOME de la cara "atmosfera"

#### 3.1.3.1.4. Paredes

Esta cara está conformada únicamente por triángulos (39,284).

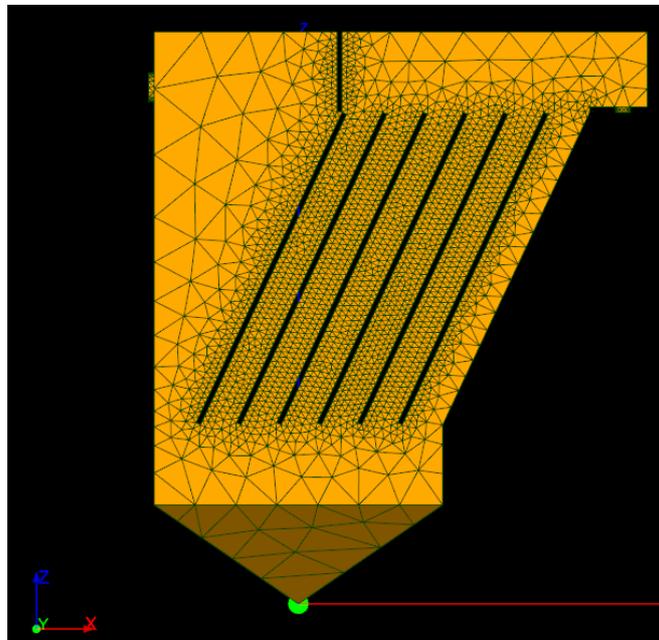


Figura 9: Malla SALOME de la cara "paredes"

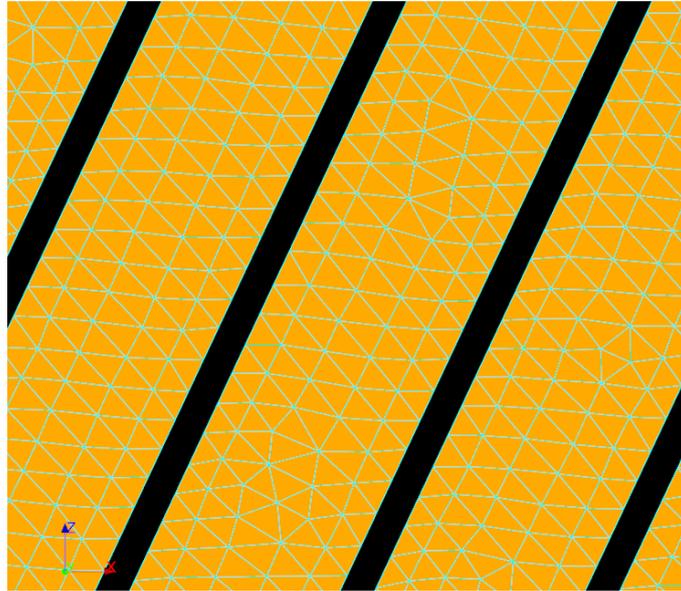


Figura 10: Malla SALOME de la cara “paredes”, acercamiento a placas

#### 3.1.4. Generación de archivos STL y archivos UNV

Se extrajeron los archivos STL de todas las caras de la geometría (inlet, outlet, walls y atmosphere) y la geometría entera, adicionalmente se extrajo el archivo UNV de la geometría entera únicamente. Se debe tomar en cuenta que se debe etiquetar correctamente los archivos STL para su posterior procesamiento en los siguientes pasos del trabajo.

#### 3.1.5. Generación de mallado hexaedrico a través de *snappyHexMesh*

##### 3.1.5.1. Generación de mallado hexaédrico de fondo

A través del directorio *blockMeshDict* ubicado en la carpeta *system* de la carpeta principal del presente proyecto, se definió una caja a través de ocho vértices en un eje de coordenadas, se debe tomar en cuenta que la geometría debe estar dentro del bloque de fondo (Greenshields, 2018).

Para el presente proyecto, los vértices de la malla anteriormente mencionada son los siguientes:

Tabla 3

Coordenadas malla hexaédrica de fondo

Numero de vértice	Coordenadas		
	X	Y	Z
0	-0.3818	-0.3750	-0.1016
1	0.7182	-0.3750	-0.1016
2	0.7182	0.3750	-0.1016
3	-0.3818	0.3750	-0.1016
4	-0.3818	-0.3750	1.1984
5	0.7182	-0.3750	1.1984
6	0.7182	0.3750	1.1984
7	-0.3818	0.3750	1.1984

Para que la malla hexaédrica definida en el directorio *blockMeshDict* pueda ser utilizado para la aplicación del *snappyHexMesh*, esta debe constituirse de cuadrados casi perfectos (Greenshields, 2018), entonces se deben calcular el número de cuadrados en cada eje (X, Y y Z) según las dimensiones de la malla de fondo, para este proyecto, se realizaron cuadrados de 0.04 m, estos parámetros deben ser definidos en el directorio *blockMeshDict*, al igual que el orden en el cual los vértices serán armados por OpenFOAM el ultimo termino que se observa en la *Figura 11*, indica las proporciones de los bloques, en este caso es 1x1x1, el cual indica que los cuadrados perfectos (Wolfdynamics, n.d.):

```
blocks
(
  hex (0 1 2 3 4 5 6 7) (28 19 33) simpleGrading (1 1 1)
);
```

*Figura 11*: Definición de hex y cantidad de cuadrados por eje en malla hexaédrica de fondo

### 3.1.5.2. DecomposeParDict

El método elegido para este trabajo de titulación es el método scotch, ya que la geometría del tanque a analizar es irregular y los otros métodos resultarían en una partición deficiente, se decidió que el algoritmo de este método asigne de forma equitativa las particiones de la geometría a los cuatro procesadores que posee la computadora utilizada. Se observa que el método utilizado es *scotch*:

```

1  /*----- C++ -----*/
2  =====
3  \ \ \ \ \ Field      | OpenFOAM: The Open Source CFD Toolbox
4  \ \ \ \ \ Operation  | Version: 6.x
5  \ \ \ \ \ And        | Web: www.OpenFOAM.org
6  \ \ \ \ \ Manipulation |
7  *-----*/
8  FoamFile
9  {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     object       decomposeParDict;
14 }
15
16 // ***** //
17
18 numberOfSubdomains 4;
19
20 method           scotch;
21

```

Figura 12: Configuración elegida para *decomposeParDict*

### 3.1.6. Bounding Box

Para el presente proyecto, las coordenadas de los puntos “A” y “B”, son las siguientes:

- A: X = -0.25    Y = -0.3000    Z = 0.000
- B: X = 0.6053    Y = 0.3000    Z = 1.0006

### 3.1.7. Línea de comandos *snappyHexMesh*

Posterior a la definición del método de partición de los procesadores, se ingresaron los siguientes comandos a la terminar de la aplicación OpenFOAM, donde a su vez se utilizarán las herramientas MPI:

- *decomposePar*
- *mpiexec -n 4 snappyHexMesh -parallel*
- *reconstructPar > constant*

### 3.1.8. Malla refinada

La malla refinada resultante contiene un total 466,776 hexaedros.

### 3.1.8.1. Entrada

Esta cara está conformada por 72 elementos.

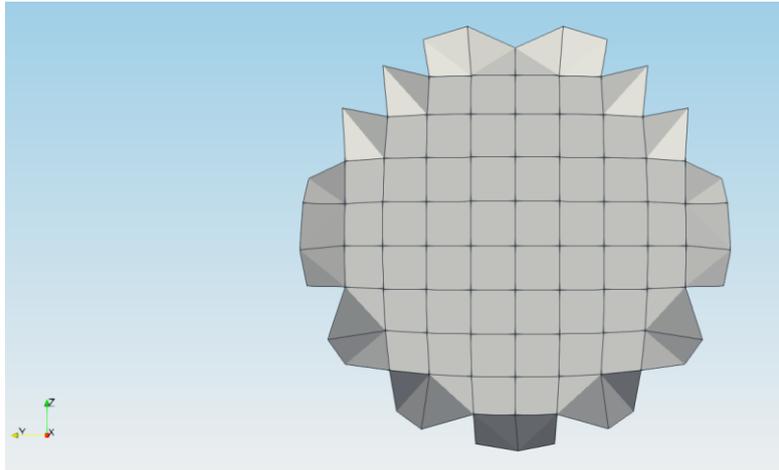


Figura 13: Malla refinada de la cara "entrada"

### 3.1.8.2. Salida

Esta cara está conformada por 20 elementos.

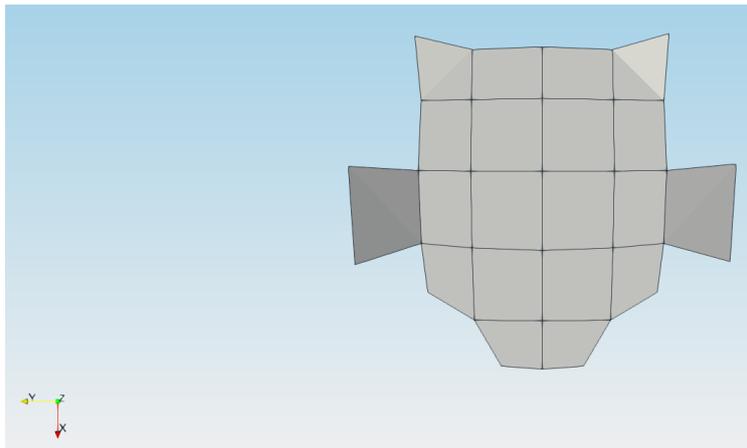
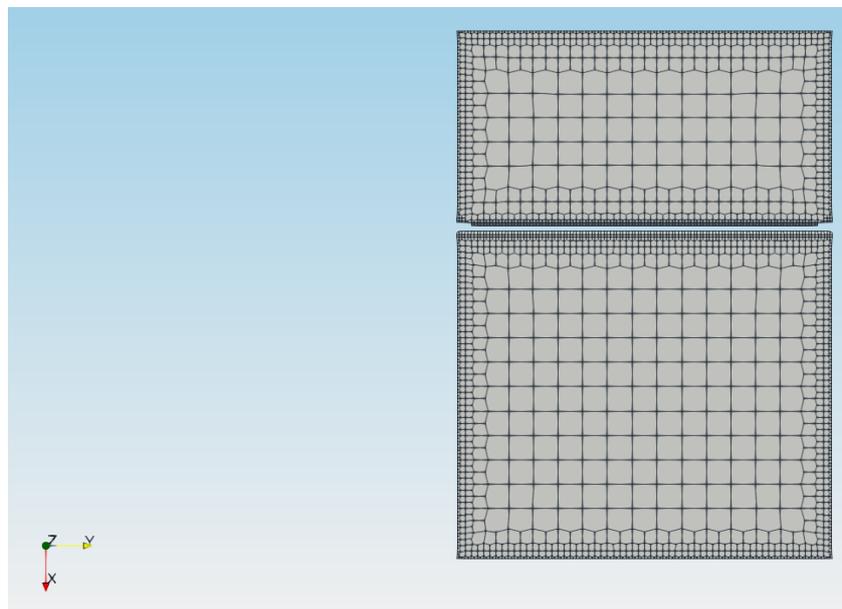


Figura 14: Malla refinada de la cara "salida"

### 3.1.8.3. Atmosfera

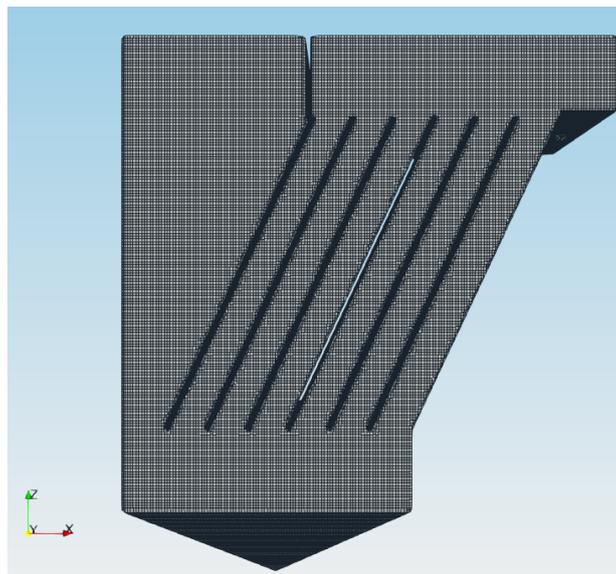
Esta cara está conformada por 2,346 elementos.



*Figura 15:* Malla refinada de la cara "atmosfera"

#### 3.1.8.4. Paredes

Esta cara está conformada por 269,816 elementos.



*Figura 16:* Malla refinada de la cara "paredes"

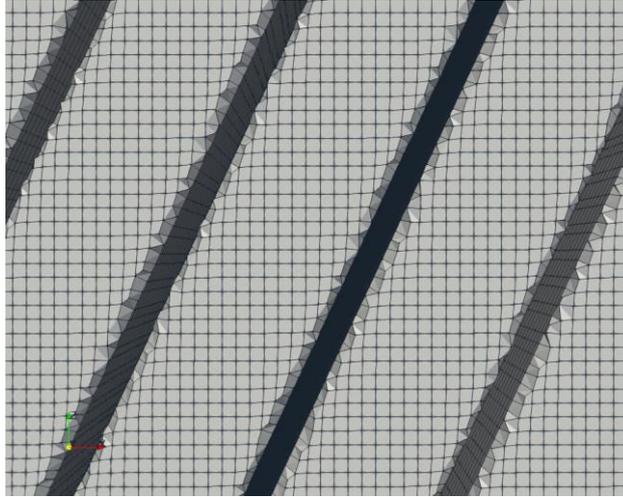


Figura 17: Malla refinada de la cara “paredes”

### 3.1.9. Condiciones de borde

#### 3.1.9.1. Carpeta 0

Tabla 4

Condiciones de borde en alpha.water

<b>Campo de borde</b>	Inlet	Type	FixedValue
		Value	0.5
	Outlet	Type	ZeroGradient
		Value	0
	Atmosphere	Type	InletOutlet
		Inlet Value	0
		Value	0
	Walls	Type	ZeroGradient

Tabla 5

Condiciones de borde de U

<b>Campo de borde</b>	Inlet	Type	FlowRateInletVelocity
		VolumetricFlowRate	0.000667
	Outlet	Type	FlowRateOutletVelocity
		VolumetricFlowRate	-0.000667
	Atmosphere	Type	pressureInletOutletVelocity
		Value	(0, 0, 0)
	Walls	Type	noSlip

En el archivo alpha.water, se definió un valor de 0.5 en el campo de la entrada del sistema, ya que el agua no ingresa al sistema por toda la cara, por lo que se asumió que el agua entraría únicamente por la mitad de la cara. Mientras que en el archivo U, se definieron los caudales en la entrada y salida del sistema, el cual fue 40 litros por minuto positivo a la entrada y 40 litros por minuto negativo a la salida. El resto de los archivos dentro de esta carpeta (k, nut, omega y p\_rgh) se mantuvieron con los valores que vienen por defecto.

### 3.1.9.2. Carpeta constant

A través del comando snappyHexMesh se generó una carpeta polymesh con sus respectivos archivos al igual que el subdirectorio extendedFeatureEdgeMesh con los archivos respectivos. Dentro del subdirectorio trisurface se anexaron los archivos STL y emesh respectivos de cada cara de la geometría.

### 3.1.9.3. Carpeta system

Dentro del archivo blockMeshDict, se establecieron los vértices para la generación de la malla de fondo para la aplicación del comando snappyHexMesh, al igual que las proporciones de aspecto de las celdas (Guerrero, 2018) (Tabla 3). En decomposeParDict, se definió el método de partición del mallado como scotch y se definió que se utilizan los cuatro

procesadores de la computadora (*Figura 12*). Para el archivo fvSchemes se mantuvieron los esquemas de discretización que fueron definidos en la carpeta base del estudio (Greenshields, 2018). En el archivo fvSolutions se decidió aplicar el algoritmo de solución PIMPLE y se añadió el predictor de momento. Dentro de setFieldsDict, se establecieron los puntos que definirán el nivel del agua en el sistema son los siguientes:

- A: X = -0.25    Y = -0.3000    Z = 0.000
- B: X = 0.6053    Y = 0.3000    Z = 0.8606

El tanque empezó la simulación casi lleno ya que este proyecto no se enfoca en simular como se llena el tanque, sino analizar el tiempo de retención hidráulico, el cual requiere que el tanque se encuentre lleno. Finalmente en el archivo controlDict se definió un tiempo de corrida de 150 segundos, además se observaran los gráficos de residuales cada 50 segundos (50, 100 y 15 segundos de simulación) de simulación para constatar el momento en el cual el sistema entra en estado estacionario, un deltaT de 0.005 para asegurar la exactitud de los cálculos realizador por OpenFOAM, un numero de Courant de 1 para asegurar la estabilidad numérica del modelo, finalmente se añadió la función de rastreo de residuales para su análisis en los tiempos definidos anteriormente (Giraldo, 1986; Greenshields, 2018).

## **3.2. Procesamiento**

### **3.2.1. Línea de comandos interFoam**

Una vez definidas las condiciones de borde del sistema, se ingresaron los siguientes comandos en la terminal de la aplicación OpenFOAM, con la asistencia de las herramientas MPI:

- decomposePar
- mpiexec -n 4 interFoam -parallel > log
- reconstructPar
- foamLog log

### **3.3. Post-procesamiento**

Como se mencionó en el anterior apartado, se analizaron las gráficas de velocidades en gnuPlot en los tiempos 0, 50, 100 y 150 para determinar el tiempo en el cual el sistema entra en estado estacionario. A través de la aplicación ParaView, se examinaron las velocidades en las dimensiones X y Z (se utilizó los datos de velocidades una vez el sistema haya entrado en estado estacionario) con el objetivo de calcular las velocidades promedio del agua en las placas inclinadas (este deberá ser realizado en las 6 placas inclinadas) del tanque, el mismo se realizó con la formula del teorema de Pitágoras. El porcentaje de eficiencia del tanque (el cual es 61% según su diseño) se determinó comparando las velocidades inclinadas calculadas con las velocidades de sedimentación establecidas en el diseño del tanque (si la velocidad del agua es menor a la velocidad de sedimentación, un cierto porcentaje de partículas sedimentan en el tanque).

Para la determinación del tiempo de retención hidráulico del tanque, se utilizaron los gráficos de velocidades en los ejes X y Z y a través de la herramienta AutoCAD, se trazaron todas las posibles rutas que pudo tomar el agua dentro del tanque, posteriormente se midió cada tramo individual, finalmente se calculo el tiempo de retención en cada tramo dividiendo la longitud del mismo para la velocidad correspondiente (la velocidad utilizada dependerá de la orientación del tramo, si el tramo tiene una orientación horizontal, la velocidad utilizada corresponde al eje X, mientras que si la orientación es vertical, la velocidad utilizada corresponde al eje Z, finalmente para los tramos cuya orientación sea inclinada, la velocidad utilizada correspondería a las calculadas anteriormente), el THR de cada ruta equivalió a la suma de tiempos de cada tramo involucrado en la ruta.

## **4. Resultados y discusión**

### **4.1. Comparación en procesamiento con solver estándar y solver mpiexec**

**-n 4**

Como se observa en la *Tabla 6*, el tiempo que se tarda la computadora sin las herramientas MPI en completar el comando snappyHexMesh es de 45 minutos y al utilizar las herramientas MPI, se demora 15 minutos. Mientras que al aplicar el solver interFoam sin las herramientas, tardó 40 minutos en generar la primera carpeta y utilizándolas se demoró entre 10 a 12 minutos por carpeta. A pesar de que al correr en paralelo se deba realizar un paso adicional antes de empezar la fase de post-procesamiento, que es la reconstrucción de las carpetas lo cual toma alrededor de 1 hora, realizar cualquiera de los procesos en paralelo es relativamente más rápido que realizarlos con un solo procesador.

*Tabla 6*

Comparación entre solver estándar y herramientas MPI

Aplicación	Parámetro	Solver estándar	Mpiexec -n 4
<b>interFoam</b> <b>Delta T: 0.005</b> <b>StartTime: 0</b> <b>EndTime: 150</b> <b>WriteInterval: 1</b>	Tiempo demorado en crear una carpeta	40 - 45 minutos	10 - 12 minutos
<b>snappyHexMesh</b>	Tiempo demorado en completar el comando	45 minutos	15 minutos

## 4.2. Análisis de convergencia

Como se observa en la *Figura 18*, las velocidades analizadas sobrellevan un incremento durante los primeros segundos de la simulación y llegan a su punto más alto durante el segundo 20 (velocidades fluctúan entre 0.09 y 0.1 m/s) y llegan a su punto más bajo en el segundo 80 (velocidades fluctúan entre 0.025 y 0.03 m/s). A partir del segundo 90 se observa la tendencia de la gráfica a un estado estacionario, es decir, que la velocidad no cambia a partir de este tiempo.

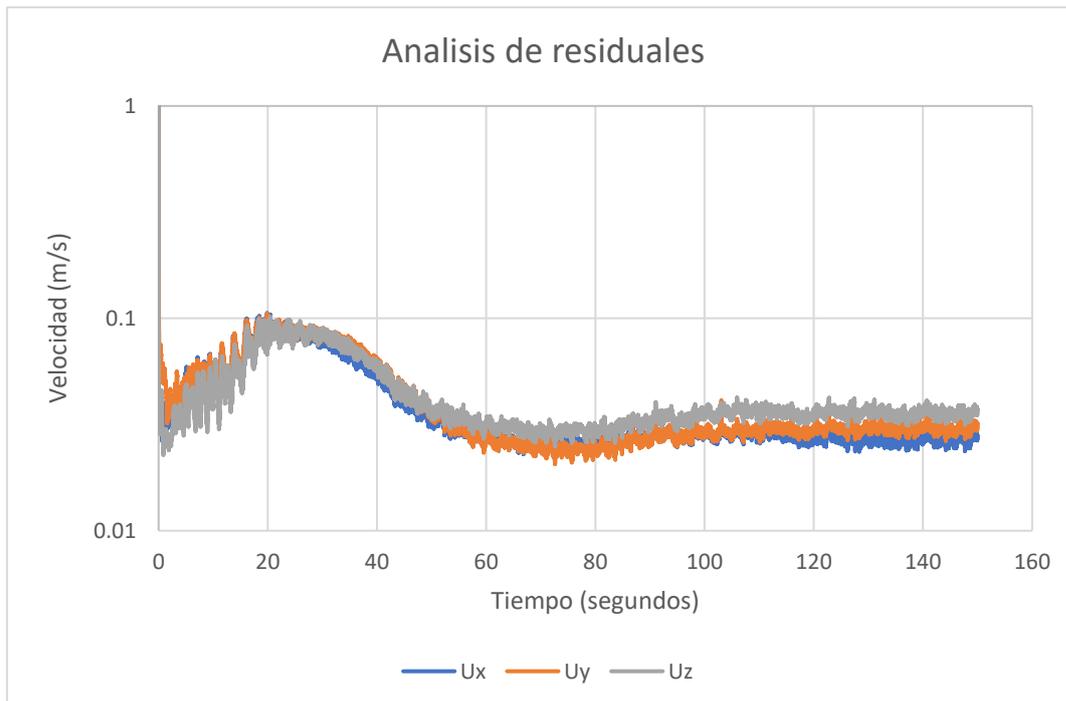
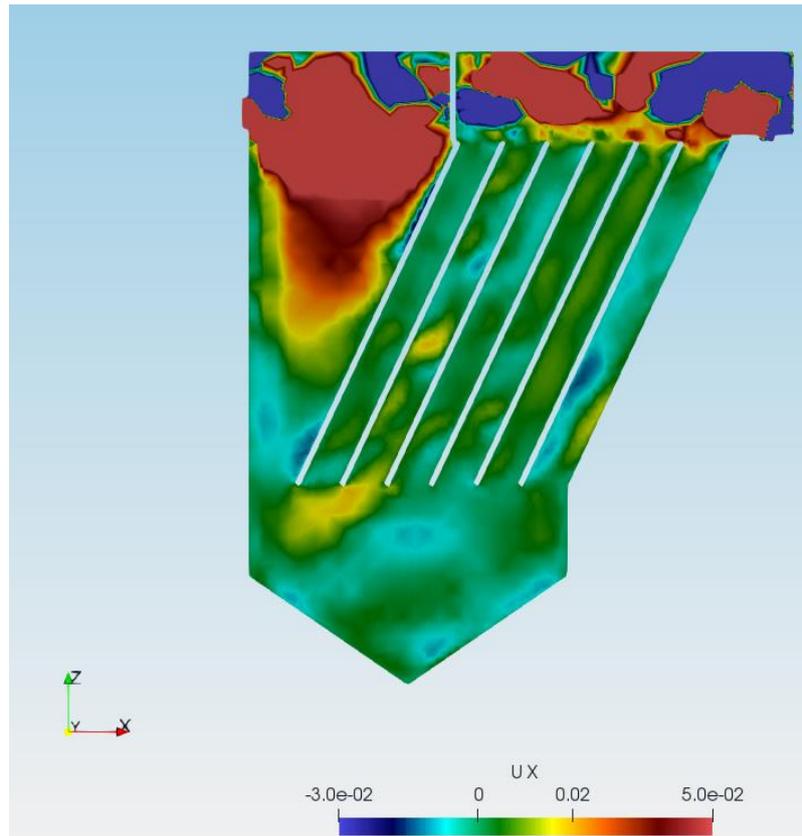


Figura 18: Análisis de velocidades en los ejes X, Y y Z

### 4.3. Análisis de velocidades

Tal como fue descrito en la metodología, el análisis de velocidades se realizó a partir de los 90 segundos de simulación. Según la Figura 19, la cual representa las velocidades en eje X en el segundo 104 de la simulación, las zonas azules y celestes representan velocidades horizontales negativas o velocidades hacia la izquierda, mientras que las zonas de color rojo, amarillo y verde representan velocidades horizontales positivas o velocidades hacia la derecha y según la Figura 20, la cual representa las velocidades en eje Z en el segundo 104 de la simulación las zonas azules, y celestes representan velocidades verticales negativas o velocidades hacia abajo, mientras que las zonas de color rojo, amarillo y verde representan velocidades verticales positivas o velocidades hacia arriba, en ambas figuras, se definió un rango de velocidades siendo la menor -0.03 m/s y la mayor siendo 0.05 m/s.



*Figura 19:* Grafico de velocidades en el eje X en el segundo 104 de la simulación

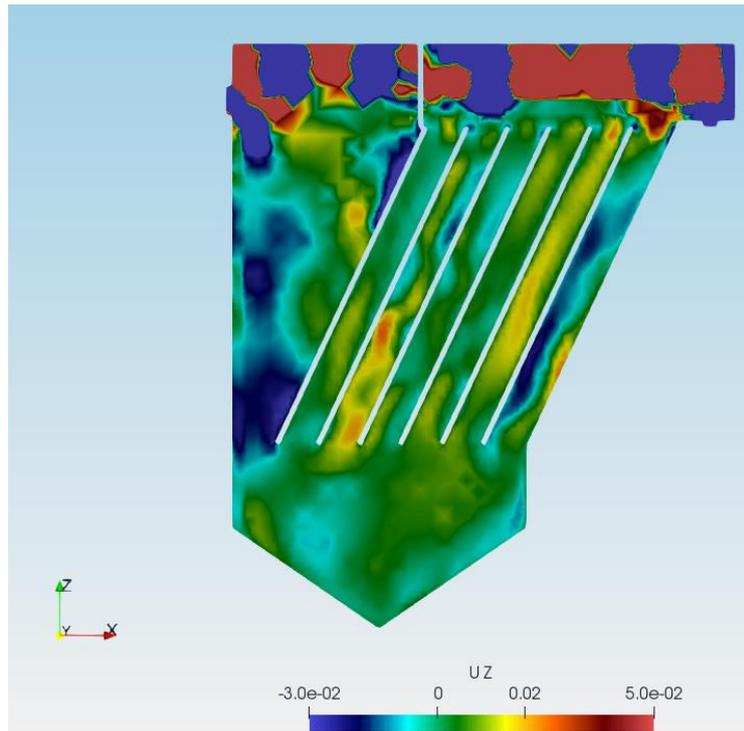


Figura 20: Grafico de velocidades en el eje Z en el segundo 104 de la simulación

En la *Tabla 8* se presentan las velocidades inclinadas calculadas, el orden de presentación de las placas empiezan de izquierda a derecha, siendo la placa 1 la que se encuentra al extremo izquierdo del tanque y la placa 6 aquella que se encuentra al extremo derecho del tanque (Figura 20), se observa que la mínima velocidad inclinada es de 0.0045 m/s en la placa 3, mientras que la mayor velocidad es de 0.0203 m/s en la placa 2, al realizar la respectiva comparación con los valores de porcentajes de remoción de la *Tabla 9* se puede determinar que el tanque lograría un porcentaje de remoción que no baja del 61%, conforme al diseño.

Tabla 8

Velocidades en placas inclinadas

Número de placa	Velocidad eje X (m/s)	Velocidad eje Z (m/s)	Velocidad inclinada (m/s)
1	-0.0045	0.01	0.0110
2	0.0035	0.02	0.0203
3	0.002	0.004	0.0045
4	0.003	0.015	0.0153
5	0.0025	0.015	0.0152
6	-0.009	-0.015	0.0175

Tabla 9

Porcentajes de remoción en base a velocidades de sedimentación

Remoción (Porcentaje)	Velocidad de sedimentación (m/s)
31%	0.0274
41%	0.0141
49%	0.0095
55%	0.007
61%	0.0054
66%	0.0034

#### 4.4. Determinación de tiempo de retención hidráulico (TRH)

Siguiendo los vectores observados en las figuras 19 y 20, se definió como entrada al punto A y el punto Q como salida, se definieron 21 tramos, de los cuales, A-B, C-D, C-E, C-F, C-G, C-H, C-I, J-P, K-P, L-P, M-P, N-P y O-P son tramos cuyo TRH fueron calculados con las velocidades en el eje X, mientras que B-C y P-Q son tramos cuyo TRH fueron calculados con las velocidades en el eje Z y D-J, E-K, F-L, G-M, H-N e I-O son tramos cuyo TRH fueron determinados con las velocidades inclinadas calculadas en el presente proyecto, definiéndose en total 6 rutas, las cuales son A-B-C-D-J-P-Q, A-B-C-E-K-P-Q, A-B-C-F-L-P-Q, A-B-C-G-M-P-Q, A-B-C-H-N-P-Q y A-B-C-I-O-P-Q (*Figura 21*).

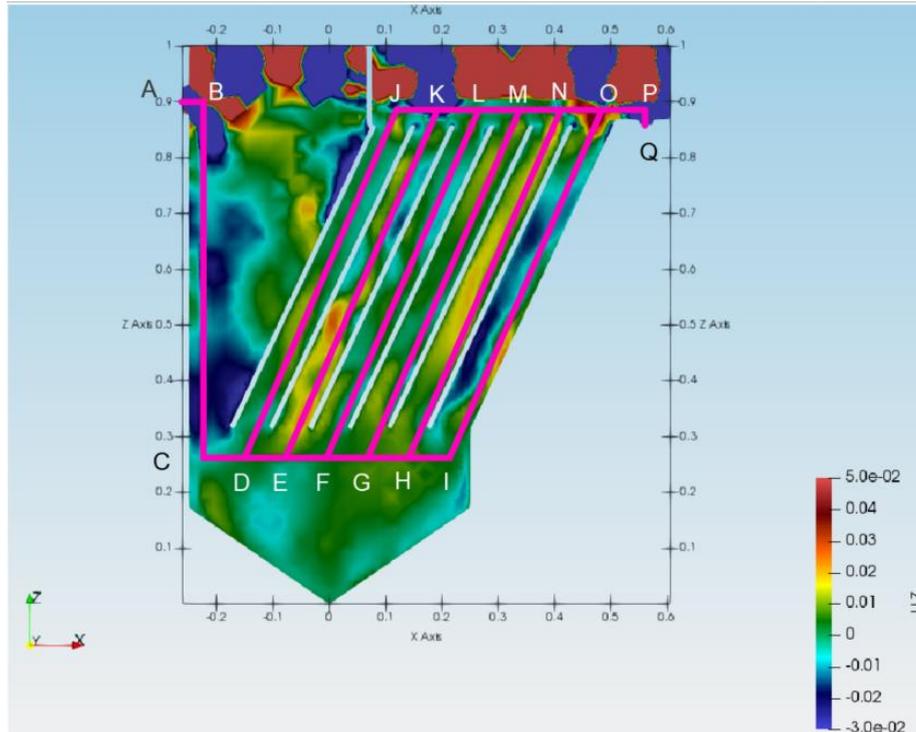


Figura 21: Diagrama de rutas dentro del tanque lamella

En la *Tabla 10*, se observan los TRH de cada tramo y posteriormente los TRH sumados de cada ruta posible, mientras que en la *Tabla 11*, se aprecian los TRH en minutos y el TRH promedio del sistema. El tramo con menor TRH es P-Q con un tiempo de 0.562 segundos, una longitud de 0.0281 metros y una velocidad promedio de 0.05 m/s y el tramo con mayor TRH es F-L con un tiempo de 153.33 segundos, una longitud de 0.6857 metros y una velocidad promedio de 0.004 m/s, la ruta con menor TRH es A-B-C-E-K-P-Q con un tiempo de 80.4 segundos o 1.34 minutos y una longitud total de 1.617 metros y la ruta con mayor TRH es A-B-C-F-L-P-Q con un tiempo de 212.85 segundos o 3.55 minutos y una longitud total de 1.76 metros, finalmente se obtuvo que el TRH promedio en el sistema es de 2.25 minutos.

Tabla 10

Tiempos de retención hidráulico (en segundos) en tramos

<b>Tramo</b>	<b>Longitud de tramo (m)</b>	<b>Velocidad promedio en tramo (m/s)</b>	<b>Tiempo retención hidráulico (s)</b>
<b>A-B</b>	0.0388	0.03	1.293333333
<b>B-C</b>	0.6433	0.025	25.732
<b>C-D</b>	0.0737	0.015	4.913333333
<b>D-J</b>	0.6857	0.011	62.53045761
<b>E-K</b>	0.6857	0.020	33.77176957
<b>F-L</b>	0.6857	0.004	153.3271812
<b>G-M</b>	0.6857	0.015	44.82561129
<b>H-N</b>	0.6857	0.015	45.09135424
<b>I-O</b>	0.6857	0.017	39.19885994
<b>C-E</b>	0.1474	0.015	9.826666667
<b>C-F</b>	0.2211	0.010	22.11
<b>C-G</b>	0.2948	0.015	19.65333333
<b>C-H</b>	0.3685	0.010	36.85
<b>C-I</b>	0.4422	0.008	58.96
<b>J-P</b>	0.4457	0.008	59.42666667
<b>K-P</b>	0.0737	0.008	9.2125
<b>L-P</b>	0.1474	0.015	9.826666667
<b>M-P</b>	0.2211	0.015	14.74
<b>N-P</b>	0.2948	0.015	19.65333333
<b>O-P</b>	0.0774	0.05	1.548
<b>P-Q</b>	0.0281	0.05	0.562
<b>A-B-C-D-J-P-Q</b>	1.9153		154.4577909
<b>A-B-C-E-K-P-Q</b>	1.617		80.39826957
<b>A-B-C-F-L-P-Q</b>	1.7644		212.8511812
<b>A-B-C-G-M-P-Q</b>	1.9118		106.806278
<b>A-B-C-H-N-P-Q</b>	2.0592		129.1820209
<b>A-B-C-I-O-P-Q</b>	1.9155		127.2941933

Tabla 11

Tiempos de retención hidráulicos (en minutos) en rutas

<b>Tramo</b>	<b>Tiempo retención hidráulico (minutos)</b>
<b>A-B-C-D-J-P-Q</b>	2.57
<b>A-B-C-E-K-P-Q</b>	1.34
<b>A-B-C-F-L-P-Q</b>	3.55
<b>A-B-C-G-M-P-Q</b>	1.78
<b>A-B-C-H-N-P-Q</b>	2.15
<b>A-B-C-I-O-P-Q</b>	2.12
<b>Promedio</b>	2.25

## 5. Conclusiones

La calidad de la malla hexaédrica generada en snappyhexmesh fue satisfactoria para la simulación realizada en el presente proyecto, puesto que las velocidades obtenidas en su correspondiente análisis representan las velocidades de diseño del tanque, asimismo el cálculo del porcentaje de remoción fue la misma calculada en el diseño del tanque. Otro factor que influyó en la exactitud de la simulación fue el uso del valor 0.005 en  $\Delta T$ , el cual permitió que los cálculos realizados por el programa sean representativos. El cálculo del TRH se considera el correcto ya que el tanque es de dimensiones pequeñas, por lo que un TRH alto no fuera el correspondiente.

El apartado de setFieldsDict permitió que el sistema entre en estado estacionario en un tiempo corto (en 90 segundos de simulación), ya que se omitió el proceso de llenado del tanque (al iniciar con la simulación con el tanque lleno), el cual naturalmente no es pertinente para el estudio.

Los archivos fvSchemes, fvSolutions, omega, nut, k y p\_rgh se mantuvieron con sus valores por defecto ya que se considera que estas no tuvieron un efecto en la simulación.

El método de partición de los procesadores “scotch” es ideal para la simulación de tanques cuyas geometrías son irregulares ya que el algoritmo reparte la geometría de manera equitativa en cada procesador, conjunto a las herramientas MPI, el proceso de simulación fue realizado óptimamente y en el menor tiempo posible.

## **6. Recomendaciones**

El uso de la malla hexaédrica es recomendable por la fiabilidad de los resultados que genera. En la mayoría de las simulaciones CFD, iniciar las mismas con un tanque lleno es ideal, de esta manera se pueden evitar archivos completamente irrelevantes para el estudio.

El uso de las herramientas MPI son indispensables para simulaciones ya que se pueden optimizar las horas de trabajo del computador y como ya ha sido mencionado, se pueden realizar un mayor número de iteraciones. El método de partición de procesadores puede variar de caso en caso, sin embargo, se puede generalizar que el método “scotch” se debe utilizar en geometrías irregulares, mientras que el método por defecto se puede aplicar en geometrías típicas, tales como tanques cilíndricos y rectangulares.

Se recomienda que el TRH sea calculado con las herramientas de OpenFOAM, puesto que el cálculo manual puede no ser confiable al utilizarse una fórmula sencilla.

## 7. Referencias

- Autodesk, I. (2010). *AutoCAD Architecture 2011, Manual del usuario*.
- Chung, T. J. (2002). Computational fluid dynamics. En *Cambridge University Press*. <https://doi.org/10.1016/B978-0-444-64046-8.00123-3>
- Eslahpazir, M., Krull, R., & Krühne, U. (2019). Computational fluid dynamics. En *Comprehensive Biotechnology*. <https://doi.org/10.1016/B978-0-444-64046-8.00123-3>
- Ganjare, A. V., & Patwardhan, A. W. (2019). CFD simulations of single-phase flow in settling tanks: comparison of turbulence models. *Indian Chemical Engineer*, 0(0), 1–14. <https://doi.org/10.1080/00194506.2019.1677514>
- Garcia-Alcaide, V. M., Palleja-Cabre, S., Castilla, R., Gamez-Montero, P. J., Romeu, J., Pamies, T., ... Milan, N. (2017). Numerical study of the aerodynamics of sound sources in a bass-reflex port. *Engineering Applications of Computational Fluid Mechanics*, 11(1), 210–224. <https://doi.org/10.1080/19942060.2016.1277166>
- Ghawi, A. H. (2016). *Department of Sanitary and Environmental Engineering A Numerical Model of Flow and Settling in Sedimentation Tanks in Potable Water Treatment Plants PhD Thesis Ing . Ali Hadi Ghawi Bratislava , Febrero 2008*. <https://doi.org/10.13140/RG.2.2.11654.37441>
- Giraldo, F. (1986). Time-Integrators. *Hilbert-Courant*, 221–533. [https://doi.org/10.1007/978-1-4612-4906-1\\_2](https://doi.org/10.1007/978-1-4612-4906-1_2)
- Greenshields, C. (2018). OpenFOAM User Guide. Recuperado en Enero de 2020 de: <https://cfd.direct/openfoam/user-guide/>
- Guerrero, J. (2018). *Mesh generation using snappyHexMesh*. 100.
- Haider, J. (2018). *OpenFOAM course for beginners*. (September), 135. Recuperado en Enero de 2020 de: <https://www.jibranhaider.weebly.com>
- Halling-Sorensen, B., & Jorgensen, S. (1993). *5. Attached Growth Reactors*.

- 153–234. [https://doi.org/10.1016/s0166-1116\(08\)70527-2](https://doi.org/10.1016/s0166-1116(08)70527-2)
- Hirsch, C. (2007). Numerical Computational of Internal & External Flows. En *British Library* (Vol. 1). <https://doi.org/10.1017/CBO9781107415324.004>
- INEC. (2016). *Estadística de información ambiental económica en gobiernos autónomos descentralizados municipales 2015 (agua y alcantarillado)*. 2015, 34. Recuperado en Enero de 2020 de:  
[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)
- INEN. (1992). Normas para estudio y diseño de sistemas de agua potable y disposición de aguas residuales para poblaciones mayores a 1000 habitantes. *Código de Práctica Ecuatoriano*, 1(Primera Edición), 5.
- Jamshed, S. (2015). Introduction to CFD. *Using HPC for Computational Fluid Dynamics*, (Diciembre 2015), 1–20. <https://doi.org/10.1016/b978-0-12-801567-4.00001-5>
- Jordan, C., Harasek, M., Gößnitzer, C., Zibuschka, S., & Chen, Y. (2018). *Tutorial Ten Residence Time Distribution*.
- Jordan, C., Harasek, M., Zibuschka, S., & Chen, Y. (2018). *Tutorial One Basic Case Setup*.
- Kang, J. (2015). *Flocculated sediment : what are the implications on erosion and sediment control ?* (Octubre).
- Khalil, E. E. (2012). CFD history and applications. *CFD Letters*, 4(2), 43–46.
- Laughlin, J. (1999). World of water 2000 : the past, present and future. In *Analytical Chemistry* (Vol. 44). <https://doi.org/10.1021/ac60309a733>
- Lofrano, G., & Brown, J. (2010). Wastewater management through the ages: A history of mankind. *Science of the Total Environment*, 408(22), 5254–5264. <https://doi.org/10.1016/j.scitotenv.2010.07.062>

- Mara, D. (2004). Domestic Wastewater Treatment in developed countries. En *Eathcan*.
- Meriç, S., Kaptan, D., Övez, S., & Eser Okten, H. (2002). Operational problems in small wastewater treatment plants: A case study. *Fresenius Environmental Bulletin*, 11(10 B), 910–914.
- Moran, S. (2018). Dirty water unit operation design: biological processes. En *An Applied Guide to Water and Effluent Treatment Plant Design* (pp. 171–202). <https://doi.org/10.1016/b978-0-12-811309-7.00014-x>
- Moukalled, F., Mangani, L., & Darwish, M. (2016). *The Finite Volume Method in Computational Fluid Dynamics*. [https://doi.org/10.1007/978-3-319-16874-6\\_21](https://doi.org/10.1007/978-3-319-16874-6_21)
- Open Cascade. (2007). Introduction to NETGENPLUGIN. Recuperado en Enero de 2020 de: <https://docs.salome-platform.org/latest/gui/NETGENPLUGIN/index.html>
- Organización Panamericana de la Salud. (2011). Agua y saneamiento: Evidencias para políticas públicas con enfoque en derechos humanos y resultados en salud pública. En *Twenty-third Street, N.W. Washington, D.C. 20037, E.U.A.*
- Prieto, D., Swinnen, N., Blanco, L., Hermosilla, D., Cauwenberg, P., Blanco, Á., & Negro, C. (2016). Drivers and economic aspects for the implementation of advanced wastewater treatment and water reuse in a PVC plant. *Water Resources and Industry*, 14, 26–30. <https://doi.org/10.1016/j.wri.2016.03.004>
- Robescu, D., Mandiș, C., & Robescu, D. (2010). Design lamellar secondary settling tank using numerical modeling. *UPB Scientific Bulletin, Series D: Mechanical Engineering*, 72(4), 211–216.
- Samer, M. (2015). Biological and Chemical Wastewater Treatment Processes. *Intech*, i, 13. <https://doi.org/http://dx.doi.org/10.5772/57353>

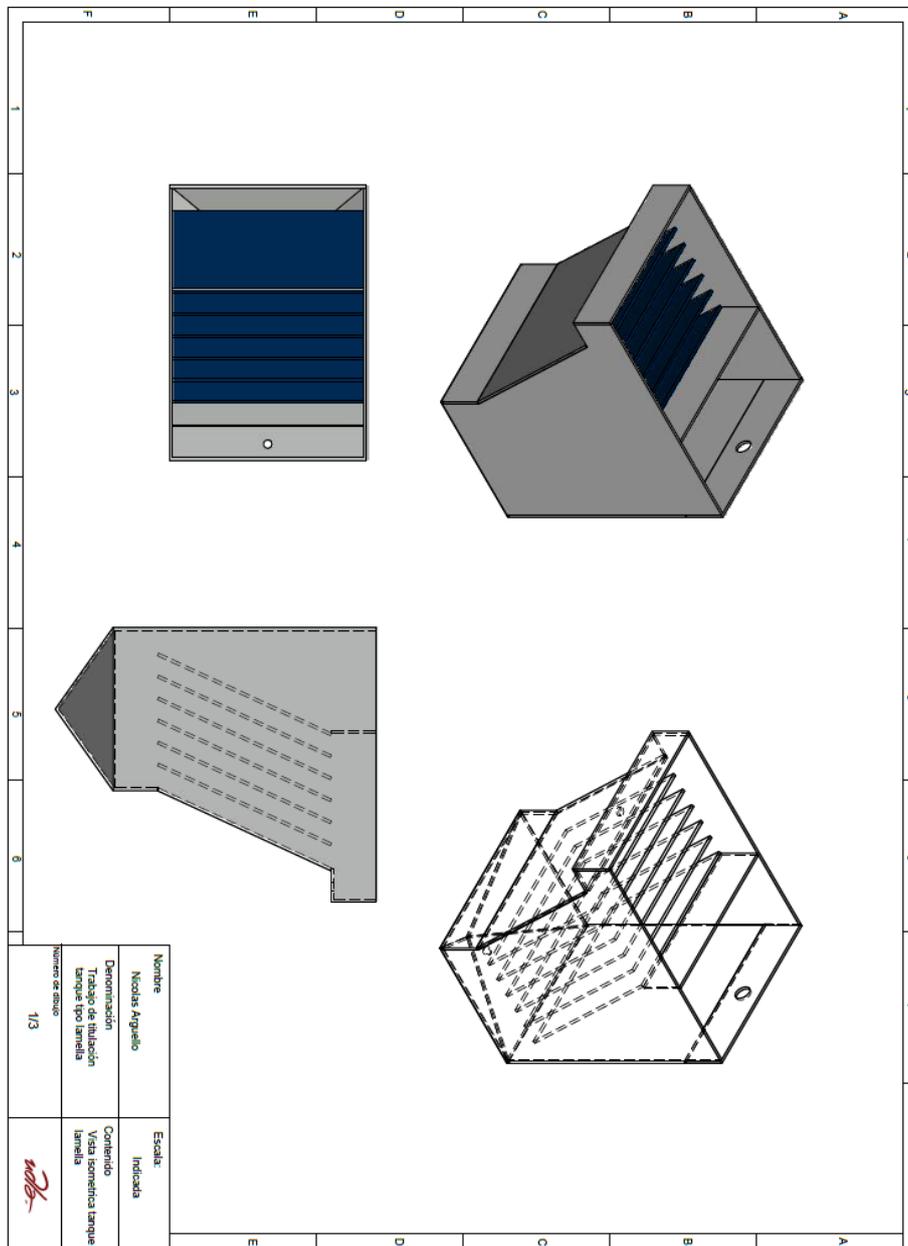
- Samstag, R. W., Ducoste, J. J., Griborio, A., Nopens, I., Batstone, D. J., Wicks, J. D., ... Laurent, J. (2016). CFD for wastewater treatment: An overview. *Water Science and Technology*, 74(3), 549–563. <https://doi.org/10.2166/wst.2016.249>
- Schulz, H. E., Andrade, A. L., & Lobosco, R. J. (2011). Hydrodynamics - Optimizing Methods and Tools. En *Hydrodynamics - Optimizing Methods and Tools*. <https://doi.org/10.5772/2371>
- Shaikh, Ziauddin Ahammad, David Graham, J. D. (2013). *Encyclopedia of Environmental Management Wastewater Treatment : Biological*. (Enero), 2645–2656. <https://doi.org/10.1081/E-EEM-120046063>
- Szilvási-Nagy, M., & Mátyási, G. (2003). Analysis of STL Files. *Mathematical and Computer Modelling*, 38(7–9), 945–960. [https://doi.org/10.1016/s0895-7177\(03\)90079-3](https://doi.org/10.1016/s0895-7177(03)90079-3)
- Tarpagkou, R., & Pantokratoras, A. (2014). The influence of lamellar settler in sedimentation tanks for potable water treatment - A computational fluid dynamic study. *Powder Technology*, 268, 139–149. <https://doi.org/10.1016/j.powtec.2014.08.030>
- Tchobanoglous, G., Burton, F. L., & Stensel, H. D. (2003). *Wastewater Engineering: Treatment and Reuse*.
- Vargas-Félix, M. (2015). *Cómputo en paralelo con OpenMP*. 1–34. Recuperado en Marzo de 2020 de: [http://personal.cimat.mx:8181/~miguelvargas/Course notes/Computo en paralelo con OpenMP 1.pdf](http://personal.cimat.mx:8181/~miguelvargas/Course%20notes/Computo%20en%20paralelo%20con%20OpenMP%201.pdf)
- Velten, K. (2010). Open Source Scientific Software ( OSSS ). *Program*, 1–16.
- Versteeg, D. J., & Woltering, D. M. (1990). A laboratory-scale model for evaluating effluent toxicity in activated sludge wastewater treatment plants. *Water Research*, 24(6), 717–723. [https://doi.org/10.1016/0043-1354\(90\)90027-4](https://doi.org/10.1016/0043-1354(90)90027-4)

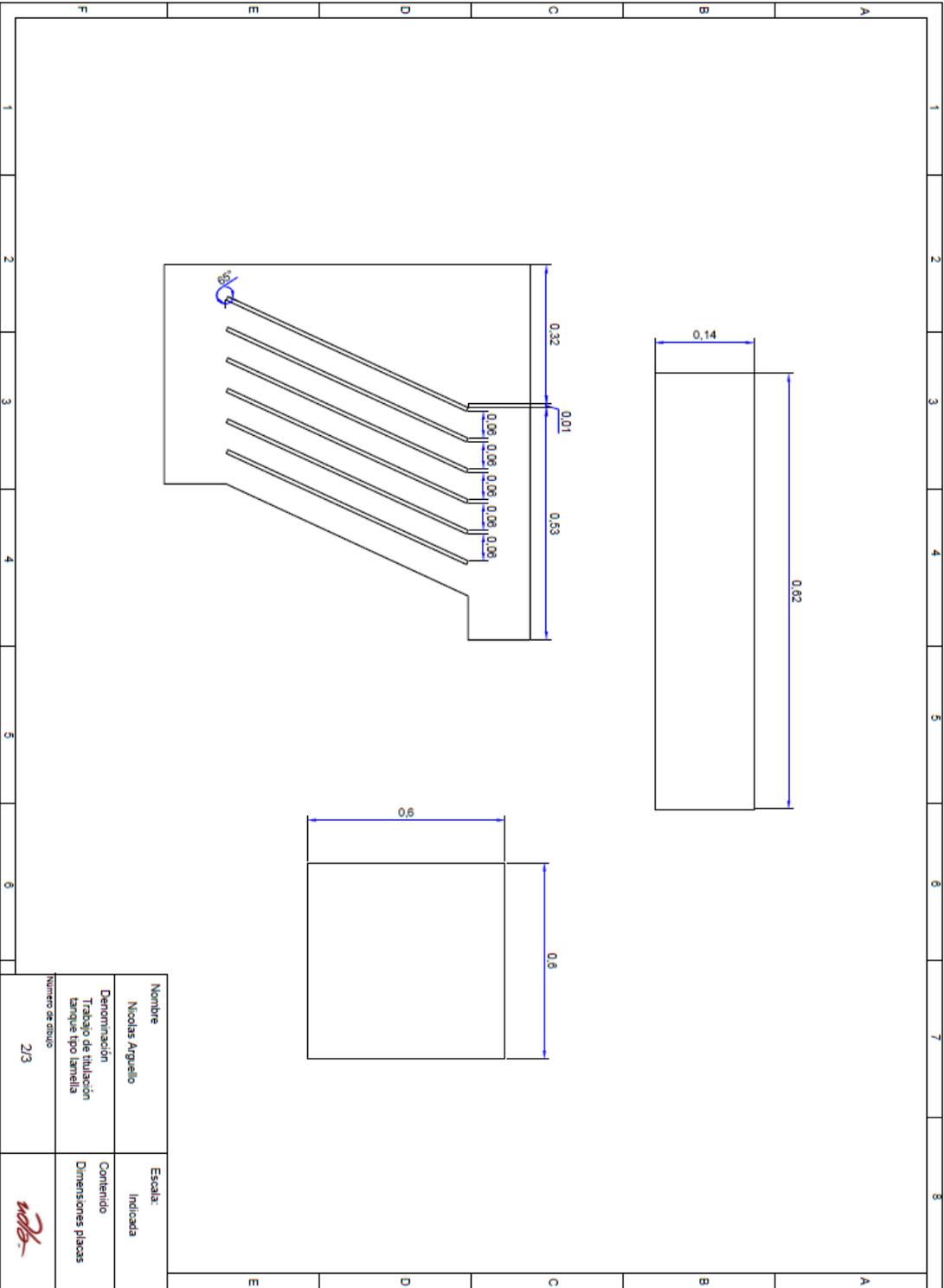
- Vita, P., & Atter, W. B. (2016). *Pimple algorithm and partitioned fsi solvers*. 2014.
- Weiss, G. (2016). Lamella settlers for treatment of urban storm runoff : Experience with model and prototype tests. *Novatech*, 1–6.
- Wolfdynamics. (n.d.). *Mesh generation using blockMesh*. 1–66.
- Xanthos, S., Gong, M., Ramalingam, K., Fillos, J., Deur, A., Beckmann, K., & McCorquodale, J. A. (2011). Performance Assessment of Secondary Settling Tanks Using CFD Modeling. *Water Resources Management*, 25(4), 1169–1182. <https://doi.org/10.1007/s11269-010-9620-1>

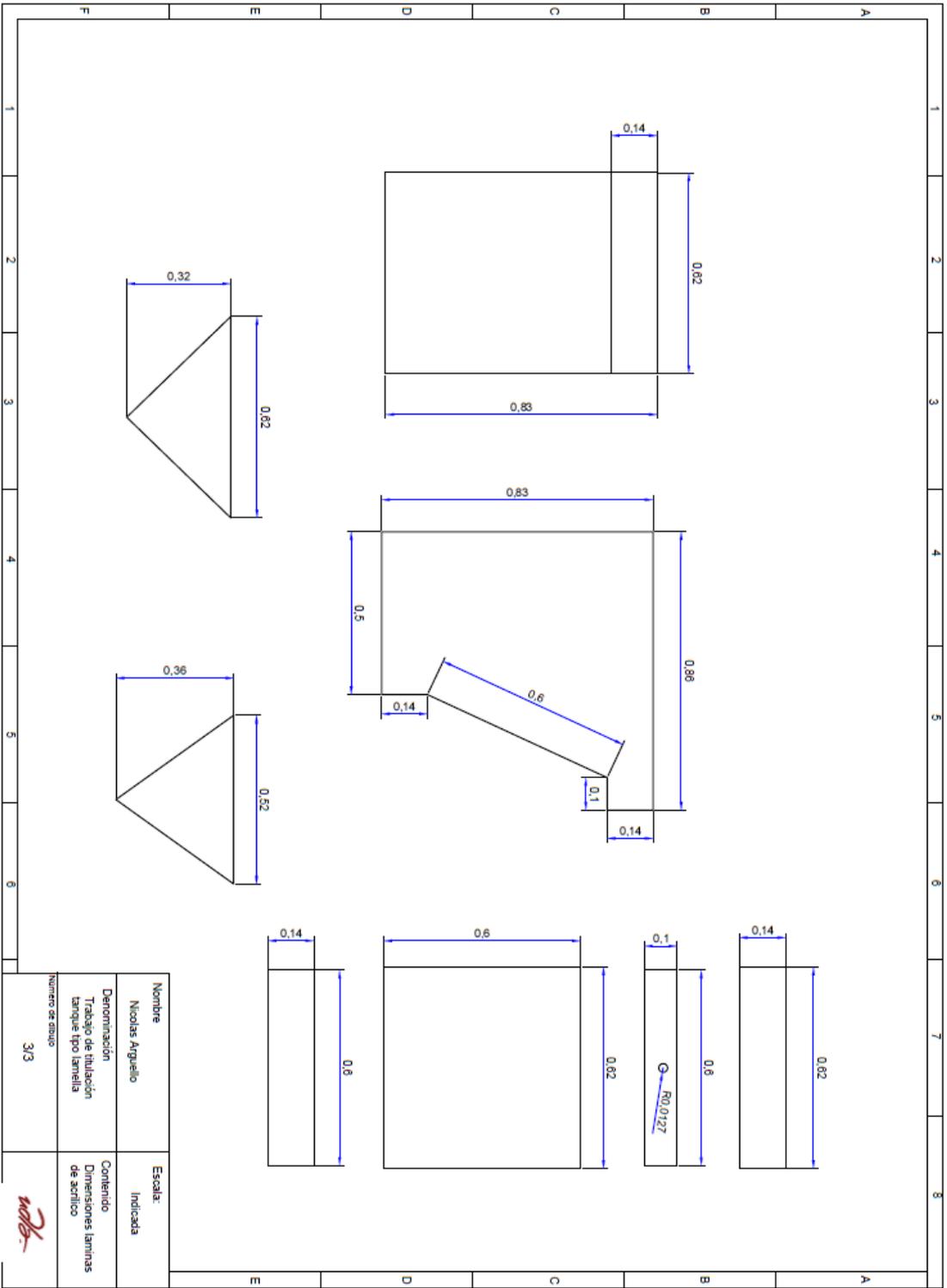
## Anexos

Anexo 1:

Planos tanque lamella (AutoCAD)







Nombre	Escalera
Nicolás Arguello	Indicada
Denominación	Contenido
Trabajo de instalación barraque tipo lamella	Dimensiones lamina de acrílico
Número de dibujo	
3/3	<i>mb</i>

