



UNIVERSITAT
ROVIRA i VIRGILI

Universitat Rovira i Virgili
Escola Tècnica Superior d'Enginyeria
Química

unir LA UNIVERSIDAD
EN INTERNET

Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y tecnología

Máster Interuniversitario en Mecánica de Fluidos
Computacional

Predicción de campos de velocidad alrededor de un cilindro mediante Redes Neuronales Artificiales

Trabajo fin de estudio presentado por: Solórzano Castillo, Byron

Directora: Salueña Pérez, Clara

Fecha: 16 de junio de 2021

Resumen

En el presente trabajo se propone una arquitectura de Red Neuronal Artificial (RNA) multicapa, para la predicción del campo de velocidad alrededor de un cilindro, en dos dimensiones, para números de Reynolds de hasta 300. El modelo de Inteligencia Artificial obtenido, basa la predicción o generalización de velocidad; en el número de Reynolds, las coordenadas de las celdas del dominio computacional, el tiempo o estados temporales y el campo de presión. Inicialmente en OpenFOAM, se establece y valida una simulación CFD, 2D, de flujo sobre un cilindro y se llevan a cabo varias simulaciones para generar la base de datos de entrenamiento, validación y prueba de las RNAs. De las configuraciones de RNAs estudiadas, resalta con mejores resultados una arquitectura, poco profunda, de dos capas ocultas con 30 neuronas por capa, siendo capaz de predecir con buena precisión las características del campo de velocidades, exhibiendo desarrollos espacio temporales coherentes. En las capas ocultas de la RNA se utilizan funciones de activación sigmoideas combinadas con una función de activación lineal en la salida, el algoritmo de aprendizaje es el de retropropagación, ampliamente utilizado en problemas ingenieriles. La RNA se somete a prueba, generando “simulaciones sintéticas” para números de Reynolds que no participaron en el entrenamiento. Se demostró que las predicciones concuerdan aceptablemente con los resultados CFD para números de Reynolds mayores a 5. El modelo de RNA generaliza (interpola y extrapola) efectivamente los campos de velocidad y los comportamientos intrínsecos del flujo.

Palabras Clave: Simulación CFD, Flujo alrededor de un cilindro, Redes Neuronales Artificiales, Campos de Velocidad.

Abstract

A multilayer Artificial Neural Network (ANN) architecture is proposed in this work to predict the velocity field around a cylinder in two dimensions for Reynolds number values up to 300. The generated Artificial Intelligence model predicts or generalizes velocity based on the Reynolds number, the coordinates of the cells of the computational domain, the time or temporal states, and the pressure field. Initially, a 2D CFD simulation of flow around a cylinder is created and validated in OpenFOAM. Several simulations are carried out to generate the training, validation, and testing databases for the ANNs. Based on the ANN configurations investigated, a shallow, two-hidden-layer architecture with 30 neurons per layer outperforms the others. It can predict the properties of the velocity field with high accuracy and demonstrate coherent spatiotemporal developments. The hidden layers of ANN employ sigmoidal activation functions in conjunction with a linear activation function at the output; the learning technique is backpropagation, which is frequently employed in engineering applications. The ANN is tested by generating "synthetic simulations" for Reynolds numbers that do not participate in the training. It is demonstrated that for Reynolds numbers greater than 5, the predictions correspond reasonably well with the CFD results. The ANN model generalizes (interpolates and extrapolates) velocity fields and intrinsic flow behaviors effectively.

Keywords: CFD Simulation, Flow around a cylinder, Artificial Neural Networks, Velocity fields.

Contenido

Resumen	2
Abstract.....	3
Contenido.....	4
Índice de figuras.....	6
Índice de tablas.....	10
Agradecimientos	11
Capítulo 1. Introducción	12
Motivación	12
Planteamiento del trabajo	13
Estructura del trabajo	13
Capítulo 2. Contexto y Estado del arte	15
Generalidades sobre Flujo sobre un cilindro.....	15
Estado del arte sobre simulaciones CFD, 2D, de flujo sobre un cilindro.....	20
Generalidades sobre Redes Neuronales Artificiales.....	22
Estado del arte sobre Redes Neuronales Artificiales vinculados a casos CFD.....	26
Capítulo 3. Objetivos y Metodología	29
Objetivos.....	29
Objetivo general	29
Objetivos específicos	29
Metodología.....	29
Capítulo 4. Desarrollo	33
Simulación CFD, 2D, de flujo sobre un cilindro.....	33
Pre-procesamiento de datos para el entrenamiento de las RNAs.....	49
RNA para el aprendizaje y predicción de resultados de velocidad.....	52

Evaluación de las arquitecturas de RNAs seleccionadas.	63
Resultados de la RNA con mejores prestaciones.	68
Capítulo 5. Conclusiones y trabajo futuro	83
Conclusiones	83
Trabajo futuro	85
Referencias	86
Anexos	95
Anexo 1. Diagramas de pastel indicando los porcentajes de celdas agrupadas por rangos de error relativo.	95
Anexo 2. Reportes gráficos de regresiones resultantes de la RNAs de 1 capa estudiadas. Vistas en el entorno de aplicación de RNA.	102
Anexo 3. Reportes gráficos de regresiones resultantes de la RNAs de 2 capas estudiadas. Vistas en el entorno de aplicación de RNA.	108
Anexo 4. Detalles de archivos de programación para la simulación OpenFOAM	113

Índice de figuras

Figura 1. Números de Strouhal en función del número de Reynolds, para un cilindro liso. ...	18
Figura 2. Evolución del coeficiente de arrastre para Re de hasta 100.....	19
Figura 3. Evolución del coeficiente de arrastre para $100 < Re < 10^6$	20
Figura 4. Representación de una neurona del modelo McCulloch-Pitts.....	22
Figura 5. Esquema de una RNA simple o de una sola capa.....	24
Figura 6. Representación de una RNA multicapa.....	24
Figura 7. Representación de una RNA recurrente.....	25
Figura 8. Representación esquemática de la metodología propuesta para el desarrollo del trabajo de fin de estudio.	32
Figura 9. Regímenes de flujo alrededor de un cilindro circular liso en corriente constante. ...	34
Figura 10. Detalle del dominio computacional asumido para la simulación.	34
Figura 11. Detalle de las partes del dominio computacional previo al mallado.	35
Figura 12. Detalle de la malla en las cercanías del cilindro.....	37
Figura 13. Residuales de presión y de velocidades para las simulaciones de flujo sobre un cilindro, con números de Reynolds de 60 y 100.	39
Figura 14. Residuales de presión y de velocidades para las simulaciones de flujo sobre un cilindro, con números de Reynolds de 200 y 300.	39
Figura 15. Densidad espectral de Cl para las diversas simulaciones evaluadas.....	40
Figura 16. Señales de Cl obtenidas de la simulación de flujo sobre un cilindro con Re = 60 y 100.....	45
Figura 17. Señales de Cl obtenidas de la simulación de flujo sobre un cilindro con Re=200 y 300.....	45
Figura 18. Señales de Cp obtenidas de la simulación de flujo sobre un cilindro con Re= 60 y 100.....	46
Figura 19. Señales de Cp obtenidas de la simulación de flujo sobre cilindro con Re=200 y 300.....	46

Figura 20. Detalle gráfico de obtención de las coordenadas representativas para cada celda del dominio computacional, aplicada sobre un extracto del dominio.....	50
Figura 21. Detalle de la región del dominio computacional, de la cual se extrae la información de interés.....	50
Figura 22. Detalle de subdominios: excluido y considerado para la extracción de información, previo al entrenamiento de las RNAs.....	51
Figura 23. Detalle de organización de los datos, previo a la aplicación de las RNAs.....	52
Figura 24. Ejemplos de arquitecturas de RNAs propuestas: a) RNA de una capa oculta, b) RNA de dos capas ocultas. Ambas arquitecturas con número de neuronas múltiplos de 5.	54
Figura 25. Matrices de entrada y salida, previo al entrenamiento de las RNAs.	55
Figura 26. Importación de datos para las capas de entrada y salida. Vista en el entorno de aplicación de RNAs.	57
Figura 27. Configuración de métodos y parámetros para la RNA. Vista en el entorno de aplicación de RNA.....	57
Figura 28. Arquitectura de la RNA creada. Vista en el entorno de aplicación de RNA.	58
Figura 29. Asignación de datos de entrada y de salida para el entrenamiento de la RNA. Vista en el entorno de aplicación de RNA.	58
Figura 30. Asignación de parámetros de entrenamiento para la RNA. Vista en el entorno de aplicación de RNA.....	58
Figura 31. Detalle de configuración y de progreso de los diversos parámetros de la RNA durante la ejecución del entrenamiento. Vista en el entorno de aplicación de RNA.....	59
Figura 32. Detalle de fin de entrenamiento. Vista en el entorno de aplicación de RNA.	60
Figura 33. Reporte gráfico de correlaciones resultantes de la RNA ejemplificada. Vista en el entorno de aplicación de RNA.....	61
Figura 34. Errores absolutos máximos de u , al aplicar las RNAs en evaluación.....	66
Figura 35. Errores relativos máximos de u , al aplicar las RNAs en evaluación.	66
Figura 36. Errores absolutos máximos de v , al aplicar las RNAs en evaluación.....	67

Figura 37. Errores relativos máximos de v , al aplicar las RNAs en evaluación.....	67
Figura 38. Campos instantáneos de velocidad para $Re = 5$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados homólogos por la RNA.	69
Figura 39. Campos instantáneos de velocidad para $Re = 60$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados homólogos por la RNA.	70
Figura 40. Campos instantáneos de velocidad para $Re = 100$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados homólogos por la RNA.	70
Figura 41. Campos instantáneos de velocidad para $Re = 140$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados predichos de por RNA.	71
Figura 42. Campos instantáneos de velocidad para $Re = 200$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados predichos por la RNA.	72
Figura 43. Campos instantáneos de velocidad para $Re = 260$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados predichos por la RNA.	72
Figura 44. Campos instantáneos de velocidad para $Re = 300$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados predichos por la RNA.	73
Figura 45. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 5$	74
Figura 46. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 60$	75
Figura 47. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 100$	76

Figura 48. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 140$77

Figura 49. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 200$78

Figura 50. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 260$79

Figura 51. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 300$80

Índice de tablas

Tabla 1. Representación de la evolución del flujo sobre un cilindro, para diferentes Re	17
Tabla 2. Detalle de subdivisión de las diversas zonas del dominio computacional propuesto.	36
Tabla 3. Detalle de algunos valores de velocidad de flujo para las simulaciones.....	37
Tabla 4. Valores de frecuencia de desprendimiento de vórtices en la estela para las simulaciones evaluadas.	40
Tabla 5. Campos de velocidad y líneas de corriente del flujo sobre un cilindro para $Re = 60$.	41
Tabla 6. Campos de velocidad y líneas de corriente del flujo sobre un cilindro para $Re = 100$	42
Tabla 7. Campos de velocidad y líneas de corriente del flujo sobre un cilindro para $Re = 200$	43
Tabla 8. Campos de velocidad y líneas de corriente del flujo sobre un cilindro para $Re = 300$	44
Tabla 9. Comparación de resultados de las simulaciones evaluadas con lo expuesto en la literatura consultada.	48
Tabla 10. Valores mínimos y máximos de las diferentes variables.	53
Tabla 11. Resultados de las RNAs entrenadas.....	62
Tabla 12. Tiempo de cada época de entrenamiento para las diversas RNAs probadas.	62
Tabla 13. Comparación de errores máximos, mínimos y medios; absolutos y relativos, de las RNA en validación.....	65
Tabla 14. Comparación de cantidad de celdas con errores relativos promedios para diferentes rangos.	82

Agradecimientos

A Dios por la existencia y la oportunidad brindada.

A mi familia por ser la motivación diaria para mejorar como persona y como profesional.

En especial, a mi esposa e hijos, por acompañarme y apoyarme sin importar las circunstancias.

A mi viejo, Agustín Solórzano, por sembrar sus principios, su lucha, sus conocimientos y sus ganas de salir adelante en mí. En su memoria dedico cada logro.

A la Dra. Clara, excelente persona y tutora, con dotes de gran docente.

Al colectivo docente de la Rovira i Virgili y de la Universidad Internacional de la Rioja.

A todos, muchas gracias.

Capítulo 1. Introducción

Motivación

Actualmente la inteligencia artificial, IA, está presente en el desarrollo de múltiples aplicaciones, en casi todos los campos de la ingeniería, desde pequeños procesos cotidianos, pasando por la identificación y búsqueda para corporaciones mundialmente conocidas, hasta su aplicación en la solución de problemas de control diversos.

Las Redes Neuronales Artificiales, RNA, como parte de la inteligencia artificial, son reconocidas por su capacidad de predicción o aproximación “universal”. Una RNA del tipo “backpropagation” con al menos una capa oculta (simple) y suficientes neuronas, es capaz de aproximar cualquier tipo de relación continua entre variables de entrada y salida. Lo anterior convierte a las RNA en poderosas e interesantes herramientas multipropósito.

Una RNA se puede considerar como un componente de software, que emite una respuesta, ante un conjunto de entradas, basadas en el reconocimiento de patrones obtenidos de un entrenamiento con datos experimentales o teóricos, siendo esta forma de adquirir el conocimiento una de sus características destacables y aprovechadas hoy en día.

En Mecánica de Fluidos Computacional, la aplicación de las RNA no ha sido la excepción y actualmente existen varios investigadores trabajando en el uso de las propiedades de aproximación de las RNA para predecir turbulencia, transferencia de calor, campos de velocidad y presión, entre otros. El éxito reciente y creciente del aprendizaje profundo proporciona varias herramientas de análisis y exploración de datos para el posible modelado de la dinámica de fluidos mediante estos datos de alta fidelidad.

Lo mencionado motiva el presente trabajo, cuyo propósito es utilizar RNAs para el aprendizaje de resultados (campo de velocidades) procedentes de un caso específico de simulación CFD, en 2D, con el fin de generar un algoritmo que pueda reproducir y predecir dicha información y a su vez, inferir resultados para entradas diferentes a las de entrenamiento.

De ser así, la RNA en conjunto con herramientas de filtrado y representación de datos (como ParaView) se lograría obtener una herramienta compuesta de análisis y síntesis de las simulaciones base, que agilicen la obtención e interpretación de resultados para nuevas condiciones de cálculo.

El hecho de lograr reproducir y representar aceptablemente simulaciones CFD mediante IA, permite la creación de aplicativos versátiles y de sencilla ejecución que podrían destinarse a los ámbitos de la docencia o un laboratorio virtual donde no sea necesario resolver modelos en derivadas parciales, o el uso de software CFD, minorando notablemente el requerimiento de procesador y el tiempo de cálculo.

Planteamiento del trabajo

El trabajo consiste en aplicar RNAs en un caso canónico de la Mecánica de Fluidos Computacional, específicamente en el flujo sobre un cilindro, para el aprendizaje y pronóstico de campos de velocidad dentro de varios números de Reynolds, hasta 300, con el fin de evaluar la aplicabilidad y fiabilidad de dicha técnica de IA como futuro recurso de generalización y exposición de resultados CFD.

El caso CFD utilizado, aunque es ampliamente estudiado, representa un fenómeno muy complejo que hasta hoy en día es objeto de interés, existiendo múltiples investigaciones numéricas como experimentales. Esa exuberante información exige revisar contenido sobre simulaciones, en literatura especializada, para definir qué mallado, qué modelo de turbulencia y qué métodos de solución, representan de mejor manera dicho caso de estudio.

Igualmente, las RNAs han sido ampliamente difundidas, utilizadas y evaluadas en diversos fenómenos de ingeniería, brindando resultados de alta aceptabilidad. Revisar esas experiencias de otros autores, análogas o semejantes al presente trabajo, permiten conceptualizar, estructurar, caracterizar y definir una RNA y sus datos de entrenamiento, validación y prueba, que conlleven a la consecución de resultados satisfactorios.

El análisis del nivel de predicción y/o generalización de la RNA, objeto de estudio, frente a un caso CFD, es un pequeño aporte para futuras aplicaciones de Inteligencia Artificial y Mecánica de Fluidos Computacional.

Estructura del trabajo

El trabajo en sus diferentes capítulos aborda lo siguiente:

En el *Capítulo 1* se expone la introducción del presente estudio, así como los elementos que motivan su desarrollo. De forma breve se menciona el caso CFD que servirá de patrón de entrenamiento para la RNA y los datos de interés a generalizar. También se describe a cortos rasgos la intención del trabajo y la forma de cómo será abordado.

En el *Capítulo 2* se presenta el estado del arte sobre los principales temas que permiten justificar el desarrollo metodológico, para cumplir los objetivos.

En primera instancia, se muestra el resultado de la revisión literaria sobre Simulación CFD, 2D, de flujo sobre un cilindro, se exponen conceptos de especial importancia y un resumen del análisis realizado a diversas investigaciones vinculadas con el fenómeno en estudio.

En segundo lugar, se describe el resultado de revisión literaria sobre Redes Neuronales Artificiales y su aplicación específica en casos CFD, detallándose conceptos y aspectos relevantes. Sobre la aplicación de RNAs en casos vinculados con la Mecánica de Fluidos Computacional, se conforma un resumen basado en información actual proveniente de revistas especializadas.

En el *Capítulo 3* se exhiben los objetivos de investigación seguidos del respectivo proceso metodológico detallado.

En el *Capítulo 4* se encuentran los resultados, expuestos en 5 apartados secuenciales que van desde la concepción y obtención de la simulación CFD de flujo sobre un cilindro, hasta la aplicación y evaluación de las Redes Neuronales Artificiales para la predicción de los campos de velocidad. Los resultados y sus apartados son referentes a:

- ✓ Simulación CFD, 2D, de flujo sobre un cilindro.
- ✓ Preprocesamiento de información para aplicarlas al aprendizaje mediante RNAs.
- ✓ Aprendizaje de campos de velocidad mediante RNAs.
- ✓ Evaluación de las RNAs para selección de la mejor opción.
- ✓ Análisis de resultados de la RNA de mejores prestaciones.

En el *Capítulo 5*, se exponen las conclusiones y trabajo futuro, especialmente enfocado en: los resultados de la RNA de mejores prestaciones, los errores de predicción, la breve comparativa temporal entre simulaciones CFD y “sintética” y en la aplicabilidad de la RNA en tareas de predicción del fenómeno estudiado.

Las conclusiones y líneas futuras develan la calidad del presente trabajo, las potencialidades de la RNA lograda y el diverso material que se puede generar para dar continuidad a este interesante trabajo de titulación.

Capítulo 2. Contexto y Estado del arte

Este apartado resume conceptos y aspectos básicos de los temas principales que conforman el trabajo de titulación (Flujo sobre un cilindro y Redes Neuronales Artificiales). También expone breves estados del arte de dichos temas, basados en la información indagada en artículos científicos. En caso de requerir profundizar algún tema en específico, se recomienda revisar el contenido de las referencias expuestas.

Generalidades sobre Flujo sobre un cilindro.

El flujo alrededor de un cilindro es considerado un problema fundamental en mecánica de fluidos que brinda la oportunidad de observar los patrones de flujo que se suscitan alrededor de un cuerpo liso con un espesor finito (Panton, 1981). Es un problema de importancia práctica puesto que existen varios casos típicos en donde se evidencia este tipo de flujo (Huq et al., 2013), vinculados con la meteorología, astronomía e ingeniería (Piñol & Grau Vidal, 1996). Este flujo viscoso, es un problema cuyo tratamiento analítico no es tarea fácil (Potter et al., 2002). En este caso particular, la forma del cilindro (por continuidad) hace que el fluido se acelere en la parte anterior y se desacelere en la posterior (De las Heras, 2012).

El parámetro más influyente para describir un flujo permanente e incompresible en el que los efectos de la gravedad, térmicos y de tensión superficial son insignificantes (como el flujo sobre un cilindro), es el número de Reynolds (Re) (Huq et al., 2013); otros parámetros como la rugosidad de la pared y la intensidad de fluctuación de la velocidad de la corriente libre, son ocasionalmente importantes (Potter et al., 2002). En este caso, Re (relación entre fuerzas inerciales y viscosas) se determina en base a la relación de aspecto (Mott & Untener, 2015).

$$Re = \frac{\rho \cdot v \cdot D}{\nu}$$

Donde Re es el número de Reynolds, D el diámetro del cilindro, ρ la densidad del fluido, v es la velocidad del flujo en la entrada libre y ν es la viscosidad cinemática del fluido.

El flujo sobre este objeto romo, exhibe un fenómeno interesante que ocurre para una amplia gama de Re (Park et al., 1998); vórtices se forman desde el objeto, regular y alternadamente desde lados opuestos del cilindro (Potter et al., 2002). Para Re muy bajos ($Re < 5$) no hay separación de flujo. El flujo en su paso por el cilindro se divide fluyendo simétricamente alrededor de la parte superior e inferior para luego unirse en la parte posterior del cilindro

(Huq et al., 2013). La capa límite es netamente laminar y la única resistencia al avance es la del rozamiento viscoso que resulta ser ínfima, por lo que la estela que se produce tras el cilindro, no oscila, es estacionaria y tiene un grosor mínimo (De las Heras, 2012).

Los autores (De las Heras, 2012; Piñol & Grau Vidal, 1996) mencionan que con el incremento de Re ($5 < Re < 50$), se crea una pequeña zona de separación detrás del cilindro donde se generan un par de remolinos simétricos, creando una región de recirculación llamada burbuja. Estos vórtices no se vierten en el flujo y la región de separación no se extiende aguas abajo.

Para que el flujo permanezca adherido a la superficie del cilindro, debe moverse hacia regiones de presión cada vez más alta a medida que avance hacia el punto de estancamiento posterior. A Re suficientemente altos ($Re > 10$), el flujo de movimiento lento de la capa límite cerca de la superficie no puede avanzar hacia la región de alta presión cerca del punto de estancamiento posterior, de modo que se separa del cuerpo (Potter et al., 2002).

Experimentalmente se ha demostrado que para Re aproximadamente mayores que 50 se forman remolinos aguas abajo del cilindro (Piñol & Grau Vidal, 1996) y dichos vórtices son acompañados por turbulencia a partir de $Re = 300$ (Potter et al., 2002). Los vórtices se emiten alternadamente desde la parte superior e inferior del cilindro con frecuencia definida. Los vórtices avanzan en la estela a una velocidad cercana al 80 % de la velocidad potencial, en dos hileras (Piñol & Grau Vidal, 1996). El flujo aguas abajo se conoce como calle de Kármán.






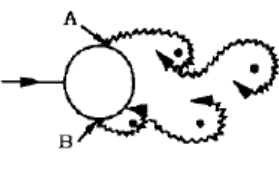
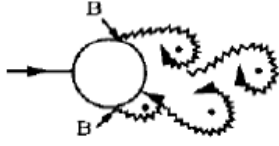


Además, (Panton, 1981) explica que en un $Re = 200$, la calle de vórtices se vuelve inestable y se dobla en la dirección axial (en sentido transversal). Para Re de hasta 10^5 , la capa límite es netamente laminar, pero contrariamente la estela es totalmente turbulenta, apareciendo estructuras 3D (De las Heras, 2012).

Por debajo de $Re = 3 \times 10^5$, (Panton, 1981) establece el régimen subcrítico, donde la capa límite del cilindro se mantiene laminar y se separa en la mitad frontal del cilindro con un ángulo poco profundo. En cambio, para $3 \times 10^5 < Re < 1.5 \times 10^6$ (régimen supercrítico) la capa límite se vuelve turbulenta cada vez más cerca del punto de estancamiento. El desprendimiento de los vórtices no es completamente simétrico y las capas límite que se desarrollan a ambos lados del cilindro se alternan entre laminar y turbulenta (Huq et al., 2013).

Para Re del régimen llamado de transición superior ($1.5 \times 10^6 < Re < 4 \times 10^6$) la capa límite es completamente turbulenta por un lado mientras que por el otro lado es parcialmente laminar

y parcialmente turbulenta (Huq et al., 2013). Finalmente, (De las Heras, 2012) expone que para Re aún mayores ($Re > 4 \times 10^6$), conocido como régimen transcrito, la capa límite es turbulenta casi en su totalidad y el punto de desprendimiento se atrasa reduciendo el ancho de la estela. La Tabla 1 detalla el comportamiento de la estela para diferentes Re .

Tabla 1. Representación de la evolución del flujo sobre un cilindro, para diferentes Re .

	Sin separación. Flujo progresivo.	$Re < 5$
	Existencia de un par fijo de vórtices simétricos adjuntos.	$5 < Re < 40$
	Estela laminar de vórtices. (Calle de vórtices de Von Kármán).	$40 < Re < 200$
	Transición a la turbulencia en la estela.	$200 < Re < 300$
	Estela completamente turbulenta. A: Separación de capa límite laminar.	$300 < Re < 3 \times 10^5$ Régimen subcrítico.
	A: Separación de capa límite laminar. B: Separación de capa límite turbulenta; pero con capa límite laminar.	$3 \times 10^5 < Re < 3.5 \times 10^5$ Régimen crítico (transición inferior)
	B: Separación de capa límite turbulenta; capa límite parcialmente laminar y parcialmente turbulenta.	$3.5 \times 10^5 < Re < 1.5 \times 10^6$ Régimen supercrítico.
	C: Capa límite completamente turbulenta en un lado.	$1.5 \times 10^6 < Re < 4 \times 10^6$ Transición superior.
	C: Capa límite completamente turbulenta en ambos lados.	$Re > 4 \times 10^6$ Régimen transcrito.

Fuente: (Huq et al., 2013).

El desprendimiento alterno de vórtices con sentidos opuestos de rotación, genera una circulación de la velocidad en torno al cilindro que se invierte constantemente en el tiempo. Este comportamiento en la estela, presenta una oscilación regular con un ciclo correspondiente a la distancia entre dos vórtices del mismo signo. La frecuencia de tal oscilación se llama número de Strouhal (Panton, 1981) y se define por la siguiente ecuación:

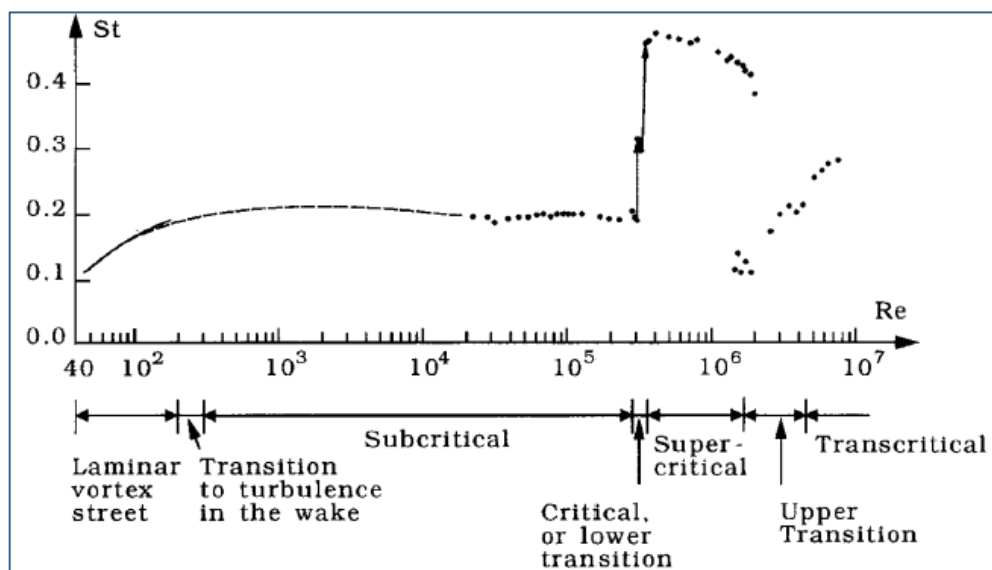
$$St = \frac{f \cdot D}{v}$$

Donde St es el número de Strouhal, f es la frecuencia de las oscilaciones, D el diámetro del cilindro (longitud característica) y v representa la velocidad del flujo.

Autores como (Panton, 1981; White, 2010) afirman que experimentalmente se observa que St varía levemente con Re , manteniéndose cercano a 0.21, siendo este valor una buena estimación en $200 < Re < 3 \times 10^5$, ver Figura 1. Por lo que se considera que la frecuencia es directamente proporcional a la velocidad del flujo en este intervalo de Re (Potter et al., 2002).

Existen particularidades en los valores de St , por ejemplo, (Panton, 1981) afirma que en $200 < Re < 400$, St pierde su carácter regular y bien definido, pero cerca a $Re = 400$ se restablece al valor de 0.2. También, para $3 \times 10^5 < Re < 3 \times 10^6$, existe un salto en St (ver Figura 1) cambiando notablemente su valor (Huq et al., 2013). Para Re mayores el desprendimiento de vórtices se torna algo irregular pero aún se observa un gran componente espectral en la frecuencia de St igual a 0.2 (Panton, 1981).

Figura 1. Números de Strouhal en función del número de Reynolds, para un cilindro liso.



Fuente: (Huq et al., 2013).

Según (Piñol & Grau Vidal, 1996), el patrón de líneas de corriente causado por la emisión de remolinos, genera una fuerza de arrastre sobre el cilindro con una frecuencia igual a la de lanzamiento de dichos remolinos. El coeficiente de arrastre se define de la siguiente forma:

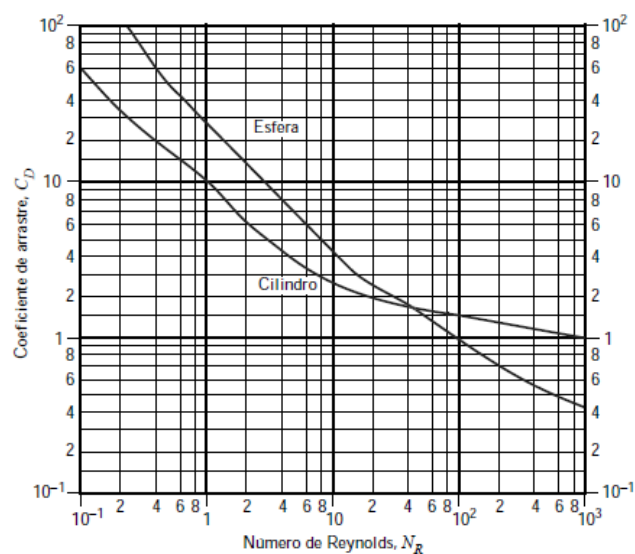
$$C_D = \frac{2 \cdot F_D}{\rho \cdot U^2 \cdot A}$$

Donde C_D es el coeficiente de arrastre, F_D es la fuerza de arrastre en la dirección del flujo, ρ la densidad del fluido, y A el área de referencia (sección transversal).

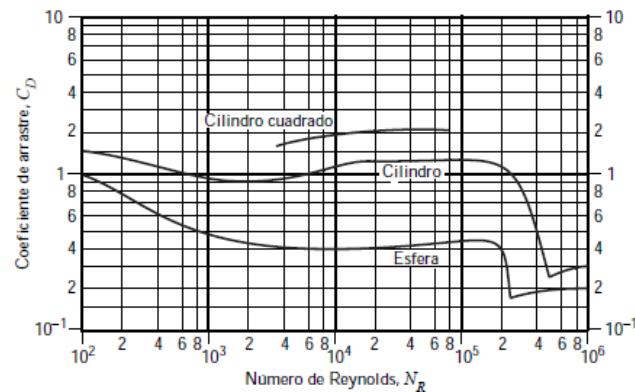
Para el flujo sobre un cilindro con $Re < 1$, el arrastre casi en su totalidad se debe a la fricción. En $Re = 0.1$ el C_D es cercano a 60, reduciéndose a 10 cuando $Re = 10$ y se reduce a 1 para $Re = 1000$. (Panton, 1981), afirma que para $100 < Re < 200$, el C_D promedio en el tiempo es cercano a 1, luego, al aumentar Re se mantiene casi constante. Las fuerzas viscosas y la vorticidad están confinadas cerca de la superficie del cilindro en una región de la capa límite.

(Mott & Untener, 2015; Panton, 1981) exponen que para $10^3 < Re < 10^5$, C_D varía desde 0.90 hasta 1.30. Para Re aún mayores, el arrastre por presión es predominante, en Re cercanos a 3×10^5 , C_D decrece bruscamente desde 1.2 hasta 0.3. La caída rápida del C_D se debe al cambio en el patrón de flujo donde la capa límite pasa de laminar a turbulenta. Al mismo tiempo, el punto sobre el cilindro donde se da la separación de flujo se desplaza hacia atrás, minorando el tamaño de la estela, resultando que la fuerza de presión total actuante sobre el cilindro disminuya abruptamente. La Figura 2 y Figura 3 muestran la evolución de C_D para diversos Re .

Figura 2. Evolución del coeficiente de arrastre para Re de hasta 100.



Fuente: (Mott & Untener, 2015)

Figura 3. Evolución del coeficiente de arrastre para $100 < Re < 10^6$.

Fuente: (Mott & Untener, 2015)

Estado del arte sobre simulaciones CFD, 2D, de flujo sobre un cilindro.

En el análisis CFD de flujo sobre un cilindro existen un sinnúmero de artículos develando el interés científico de este caso de flujo externo, incompresible y estacionario. El incremento notable de la capacidad de cómputo de los Pc's ha influenciado en los diferentes métodos y algoritmos de solución y con ellos, la precisión de resultados.

Realizar una revisión profunda de los avances sobre simulaciones de flujo sobre un cilindro, resultaría un trabajo muy extenso debido a la riqueza de la información existente. Por ello, en este apartado se pretende abordar de manera somera algunos aspectos relevantes.

La presencia de estudios de este tipo de flujo, en 2 dimensiones (2D), es pequeña, comparada con la cantidad de trabajos realizados en 3 dimensiones (3D). De los temas indagados, la mayoría de casos 2D, para $Re < 300$ se concentran en los años 70's y 80's. En los últimos años existe mayor énfasis en estudios para casos 3D, con Re superiores, dentro regímenes como el subcrítico hasta los transcíticos. Sin embargo, de manera metodológica, algunos estudios 3D, exponen también resultados 2D para discernir diferencias producidas por la variación de Re . Dichos resultados 2D, son de interés en el presente estudio y por ello son citados.

De lo consultado, los dominios computacionales más frecuentes son cuadriláteros y circulares. Los trabajos de (Dennis & Chang, 1970; Tuann & Olson, 1978) con dominios rectangulares y los de (Braza et al., 1986; Fornberg, 1980) con dominios circulares, exponen resultados de: St , Cl , Cd , Cp y otros, concordantes con estudios experimentales para Re de hasta 1000.

Un aspecto importante en las simulaciones es la configuración del dominio computacional, de tal manera que sus dimensiones no afecten los resultados y tampoco aumenten la carga computacional. Ejemplo de este tipo de estudios es el de (Piñol & Grau Vidal, 1996) que

establece que un dominio de $X = 25D$ y $Y = 25D$, es suficiente para lograr buenos resultados, 2D, para Re de hasta 200, utilizando el método de diferencias finitas.

En cuanto a las aproximaciones de la turbulencia, para resolver el problema 2D, existen múltiples estudios:

- Sin modelo de turbulencia (laminar), en este grupo están los trabajos de: (Park et al., 1998), (Farhoud et al., 2012), (Ben & Dou, 2015) y (Kornbleuth, 2016).
- Con simulación numérica directa (DNS, por sus siglas en inglés) están los trabajos de: (Henderson, 1995), (Rapaka & Sarkar, 2016), y los trabajos de (Jiang, 2020; Jiang, Cheng, & An, 2017; Jiang, Cheng, Draper, et al., 2016, 2017a, 2017b; Jiang, Cheng, Tong, et al., 2016; Jiang & Cheng, 2018).
- Con simulación de grandes remolinos (LES, por sus siglas en inglés) se encuentran los estudios de: (Breuer, 2000), (Tutar & Hold, 2001), (Franke & Frank, 2002) (Shao & Zhang, 2006), (Parnaudeau et al., 2008), (Meyer et al., 2010), (Rapaka & Sarkar, 2016). (Rapaka & Sarkar, 2016), (Zhang et al., 2016), (Yao et al., 2017) y (You et al., 2017).
- Realizando un promedio temporal (Reynolds-Averaged Navier-Stokes Equations, RANS), están los trabajos de: (Shao & Zhang, 2006) y (Rajani et al., 2009).
- Combinaciones de las anteriores y otros métodos, se encuentran los estudios de: (Wu et al., 2004), (Ding et al., 2004), (Harichandan & Roy, 2010), (Kozlov et al., 2011), (X. Wang et al., 2014) y (Chopra & Mittal, 2019).

También se observa que varios de los estudios DNS, 2D y 3D, de los últimos 5 años (como los de Jiang *et al*) son desarrollados en el aplicativo OpenFOAM, con dominios computacionales rectangulares y mallados refinados, permitiendo analizar a detalle la estela, así como estimar con precisión diversos coeficientes (C_d , C_l , C_p) y otras características importantes (St y ángulo de desprendimiento de vórtices). Otro aspecto de interés, es el tipo de resolovedor utilizado (IcoFoam), resolviendo las ecuaciones laminar incompresibles de Navier-Stokes usando el algoritmo de presión implícita con división de operadores (PISO, por sus siglas en inglés).

Lo antes expuesto permite concluir que el flujo sobre un cilindro es un tema sobre el cual se proponen y evalúan varios métodos de simulación. Para distintos números de Reynolds, las simulaciones DNS y LES son las más empleadas, brindando resultados cercanos a los obtenidos experimentalmente. Al parecer, el desarrollo de este tipo de simulaciones se debe a que posibilitan un mejor entendimiento de la estela y el desprendimiento de vórtices.

Por otro lado, las simulaciones de tipo: laminar, RANS e híbridas son totalmente adecuadas, pero de lo consultado, son aplicadas en menor medida.

Se denota la presencia del software OpenFOAM y el tipo de resolvidor (PISO) utilizado en estudios de años recientes.

Es claro que quedan pendientes de análisis muchos aspectos de las simulaciones, 2D, de flujo sobre un cilindro, como: tipo, forma y convergencia del mallado, dimensiones del dominio computacional, condiciones de contorno, métodos de cálculo y más, pero con esta breve revisión se identifican algunas de las principales líneas de investigación relacionadas.

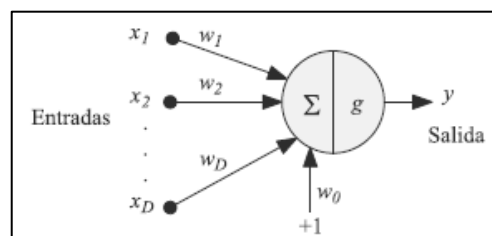
Generalidades sobre Redes Neuronales Artificiales.

Una Red Neuronal Artificial (RNA) es una técnica de Inteligencia Artificial y hace referencia a la aplicación de modelos matemáticos que permiten procesar información de manera similar a la ejecutada por el cerebro humano. Generalizando a través de algoritmos de entrenamiento y aprendizaje datos simples hasta muy complejos de obtener computacionalmente (Ponce, 2011).

Según (Yegnanarayana, 2005) una RNA general se compone de un cierto número de elementos llamados neuronas, las cuales pueden recibir señales con valores de entrada y devolver una señal de salida ponderada. La salida de valores se ve determinada por el peso de las señales de entrada y se conoce como valor de activación. Este valor puede obtenerse mediante el uso de varios modelos. Entre los principales están los propuestos por McCulloch-Pitts, Rosenblatt's (Perceptrón) y Widrow (Adaline o Elemento Lineal Adaptable).

Según (Palma & Marín, 2007), el modelo general mayormente utilizado en RNA corresponde al de McCulloch-Pitts, el mismo que se representa mediante la Figura 4:

Figura 4. Representación de una neurona del modelo McCulloch-Pitts.



Fuente: (Palma & Marín, 2007).

La suma ponderada establecida por este modelo se puede obtener mediante:

$$a = \sum_{i=1}^D w_i x_i + w_o$$

Donde a es el valor de activación, x_i es la entrada i , w_i es el peso sináptico de la entrada i y w_o es el umbral o sesgo.

Luego, el valor de salida y se deriva de esta suma ponderada al aplicar una función de transferencia o activación:

$$y = g(a) = g\left(\sum_{i=1}^D w_i x_i + w_o\right)$$

Donde y es valor de salida y $g(a)$ es la función de transferencia.

El modelo de McCulloch-Pitts usa una función de transferencia del tipo escalón; sin embargo, son de uso práctico las funciones lineal, sigmoidea, tangente hiperbólica o gaussiana (Palma & Marín, 2007).

(Yegnanarayana, 2005) expone que la manera como las unidades de procesamiento se encuentran conectadas entre sí se denomina Topología de una RNA. Es decir, la forma en que éstas se organizan y el número de conexiones presentes en la red.

La manera de organización de las RNAs se da generalmente mediante capas de unidades de procesamiento, las mismas que presentan dinámicas de activación y funciones de salidas en común. El tipo de conexión puede ser de una capa a otra (intercapas), dentro de una sola capa (intracapa) o entre capas e intracapas; además de ello, las RNA pueden ordenarse en una red retroalimentada.

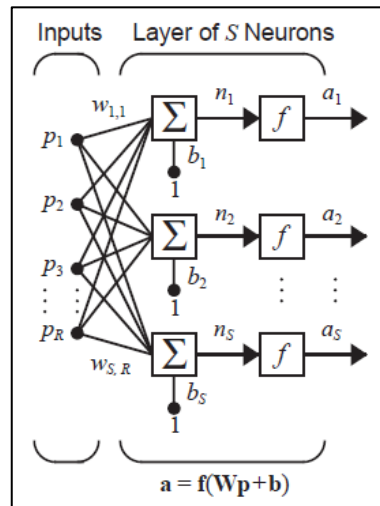
Por su parte (Haykin, 2005), define la Arquitectura de una RNA como una estructura o patrón de conexionado de una RNA. En una RNA los nodos se conectan mediante sinapsis, dichas conexiones sinápticas estructuradas definen el comportamiento de una RNA. Las conexiones sinápticas son direccionales, donde la información se propaga en un único sentido.

Es posible identificar tres clases principales de arquitecturas de red.

1. *Red directa o simple, de una sola capa.* Con una arquitectura en la que cada entrada se conecta directamente con cada neurona. A su vez, cada neurona posee sus propios valores de sesgo, sumatorio, función de transferencia y valor de activación. Matemáticamente, las salidas se pueden obtener mediante una matriz que engloba

los pesos de los valores de entrada, el vector que posee dichos valores y un vector con los valores de sesgo o umbral. La Figura 5 detalla lo expuesto.

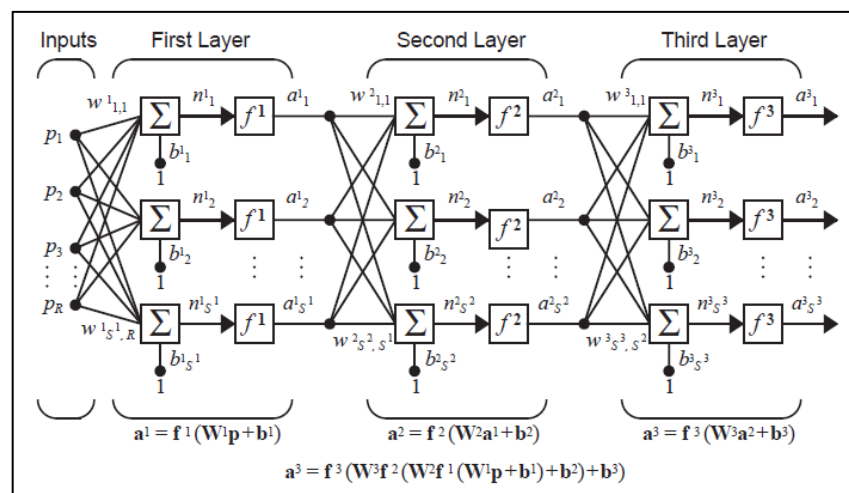
Figura 5. Esquema de una RNA simple o de una sola capa.



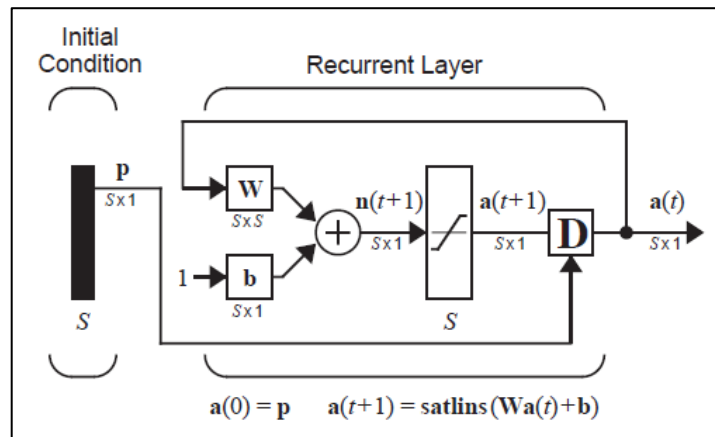
Fuente: (Haykin, 2005).

2. *Red directa multicapa.* Esta arquitectura conecta redes de una sola capa entre sí. Lográndose varias capas simples en donde cada elemento del vector de salida se conecta con cada elemento del vector de entrada de la capa siguiente. Ver Figura 6.
3. *Red recurrente.* Esta arquitectura se caracteriza principalmente por tener una retroalimentación. Donde algunos de los valores de salida pueden reintegrarse a la entrada con el fin de corregir el peso sináptico que ingresa a la red inicial Ver Figura 7. Este tipo de arquitecturas poseen notables ventajas respecto a las redes directas debido a su comportamiento temporal.

Figura 6. Representación de una RNA multicapa.



Fuente: (Haykin, 2005).

Figura 7. Representación de una RNA recurrente.

Fuente: (Haykin, 2005).

Según (Palma & Marín, 2007), las RNAs reciben datos de entrada que se transforman para dar una salida, con el objetivo de clasificarlos o ajustar una función. Para ello se supone que la red con un número suficiente de neuronas, podrá ajustar cualquier función continua con una cierta precisión con tan sólo escoger los valores adecuados para los parámetros ajustables.

El proceso de entrenamiento o aprendizaje de una RNA, consiste en establecer un ajuste a la red para que los valores de entrada arrojen una salida deseada. Según (Ponce, 2011), generalmente el ajuste se realiza sobre los pesos sinápticos, los cuales se pueden conocer con anterioridad o mediante técnicas de retroalimentación. El entrenamiento de una red mediante el ajuste de pesos se puede realizar a través de la regla Hebbiana o la regla Delta.

La regla Hebbiana establece que, si dos unidades j y k están activas simultáneamente, la conexión de las mismas se debe fortalecer mediante la modificación del peso. Esto es:

$$\Delta\omega_{jk} = \gamma y_j y_k$$

Donde γ es la constante de proporcionalidad positiva que representa la tasa de aprendizaje.

En cambio, la regla Delta presenta un ajuste de los pesos a través de la diferencia entre la activación actual y la deseada. Esta regla se define como:

$$\Delta\omega_{jk} = \gamma y_j (d_k - y_k)$$

Donde: d_k es la activación deseada.

Varios autores consultados como (Berzal, 2018; Caicedo B & López S, 2009; Haykin, 2005), coinciden en que la RNA de alimentación directa "Feed forward" es la más popular en aplicaciones de ingeniería. Esta RNA se caracteriza por una estructura y un análisis matemático

relativamente simple. La red de propagación de errores hacia atrás (del inglés backpropagation) es la primera y más común RNA de retroalimentación porque posee un esquema de aprendizaje matemáticamente estricto para entrenar la red y garantizar el mapeo entre entradas y salidas. Durante la etapa de retroalimentación, se suministra un conjunto de datos de entrada a los nodos de entrada y la información se transfiere hacia adelante a través de la red a los nodos en la capa de salida. Los nodos realizan transformaciones de entrada y salida no lineales mediante la función de activación sigmoidea. Cada ciclo de corrección de propagación hacia atrás y hacia adelante para reducir la pérdida se denomina *época de entrenamiento* (La, 2019).

Estado del arte sobre Redes Neuronales Artificiales vinculados a casos CFD.

La Inteligencia Artificial (IA) con sus diversos modelos de aprendizaje ha tenido un desarrollo importante a partir del notable aumento de la capacidad de cómputo de los ordenadores actuales, algo similar a lo que ocurre con la dinámica de fluidos computacional (CFD).

De lo indagado, se evidencian varias investigaciones donde se aplica IA para identificar y reproducir resultados CFD, que tratan especialmente sobre: transferencia de calor, turbulencia, campos de presión y de velocidad, y coeficientes de sustentación y arrastre. Estos estudios están presentes notoriamente desde los años 90 con técnicas de IA como:

- ✓ Fuzzy ARTMAP, modelos de aprendizaje supervisado basados en la Teoría de Resonancia Adaptativa (ART, por sus siglas en inglés) incorporando lógica difusa (Hagan et al., 2014).
- ✓ Redes Neuronales de Función de Base Radial (RBF, por sus siglas en inglés). Modelo híbrido que incorpora tanto el aprendizaje supervisado como el no supervisado (Caicedo B & López S, 2009).
- ✓ Las Redes Neuronales Artificiales Multicapas, sin realimentación (MLFFNN, por sus siglas en inglés). Modelo en donde la información avanza en un solo sentido desde la capa de entradas hasta la capa de salida (Berzal, 2018).
- ✓ Redes Neuronales Convolucionales (CNN, por sus siglas en inglés). Modelos de aprendizaje profundo basados en el uso de convoluciones. Son ampliamente utilizadas para problemas de procesamiento de imágenes (Berzal, 2018).
- ✓ Redes Neuronales Informadas por la Física (PINN, por sus siglas en inglés). Modelos de aprendizaje que codifican el problema que rige las ecuaciones, como parte del

entrenamiento de la RNA. Son una herramienta para resolver varios problemas, como el cálculo de ecuaciones diferenciales en derivadas parciales lineales y no lineales, completar tareas de asimilación de datos y cuantificación de incertidumbre (Markidis, 2021).

- ✓ Otros tipos de Redes Neuronales, por ejemplo: La Descomposición Ortogonal Adecuada (POD, por sus siglas en inglés) denominada Red Neuronal Lineal; Redes Neuronales Basadas en Retrasos (DBN, por sus siglas en inglés), entre otros.

Respecto a la predicción de parámetros pertenecientes al flujo sobre un cilindro, no se constató mucha diversidad. Los tratados analizados abordan los siguientes temas:

- Reconocimiento de patrones incrustados en el campo de velocidad de la estela turbulenta detrás de un cilindro circular, mediante Fuzzy ARTMAP (Ferre et al., 1996).
- Aprendizaje de la dinámica del campo de velocidades de la estela turbulenta generada detrás de un cilindro, mediante Fuzzy ARTMAP (Giralt et al., 2000).
- Detección de estructuras en la estela turbulenta de un cilindro, mediante DBN (López Peña et al., 2002).
- Reconocimiento del número de Reynolds y campo de presión del flujo alrededor de un cilindro, mediante una combinación de Sensado Comprimido (CS, por sus siglas en inglés) y aprendizaje automático (ML, siglas de Machine Learning) (Bright et al., 2013).
- Estimación de campos de presión y velocidad sobre un cilindro, así como de las fuerzas de elevación y arrastre. Mediante PINN (Raissi et al., 2018).
- Reconocimiento de la transferencia de calor por convección natural en el flujo alrededor de un cilindro circular horizontal frío con temperatura superficial constante. Mediante MLFFNN con retro propagación de error (Tahavvor & Yaghoubi, 2008).
- Predicción de transferencia de calor generada por convección forzada inestable sobre un cilindro cuadrado. Mediante MLFFNN con retro propagación de error (Dey & Das, 2016).
- Predicción del campo de velocidad alrededor de un cilindro basado en el campo de presión. Mediante CNN (Jin et al., 2018).
- Identificación de vórtices del campo de flujo alrededor de un cilindro. Mediante CNN (Z. Ye et al., 2019).
- Predicción de datos de campo de flujo turbulento en la estela del flujo sobre un cilindro. Mediante CNN (Fukami et al., 2019).

- Detección de características de flujo para modelar la distribución de presión alrededor de un cilindro. Mediante CNN (S. Ye et al., 2020).
- Predicción de arrastre en formas 2D arbitrarias (incluido un cilindro) para flujos del tipo laminar (Viquerat & Hachem, 2020).

Con el detalle anterior y en base a los resultados expuestos en cada artículo, es evidente el éxito de las técnicas de la IA para reproducir los diferentes fenómenos suscitados en el flujo sobre un cilindro. Los primeros estudios (observados) sobre este tema datan del año 2003. En el caso de las RNA del tipo MLFFNN no existen aplicaciones en las que se realice detección y/o predicción de campos de velocidad. Sin embargo, existen múltiples trabajos vinculados a resultados CFD, por ejemplo:

- Identificación y reproducción de flujo turbulento, utilizando datos de una Simulación de Grandes Remolinos (LES, por sus siglas en inglés) de un flujo de canal plano (Sarghini et al., 2003).
- Predicción y simulación de la cinemática del movimiento de aleteo inestable y sus fuerzas aerodinámicas correspondientes (Kurtulus, 2009).
- Predicción de la transferencia de calor desde una superficie con flujo de calor constante sometido a un flujo anular oscilante (Akdag et al., 2009).
- Predicción de los números de Nusselt para el análisis de transferencia de calor en un tubo divergente convergente (Taymaz & Islamoglu, 2009).
- Predicción de formas funcionales para el modelado de flujos turbulentos y de transición, con el objetivo de desarrollar modelos de cierre (Duraisamy et al., 2015).
- Predicción del comportamiento de flujo no lineal en medios porosos (Wu et al., 2004).

Los estudios en los que se aplica el método MLFFNN y CNN han ganado actualmente mucho prestigio, debido al carácter de predictor “universal” de funciones y a la alta capacidad de procesar imágenes, respectivamente. Estos aspectos; datos e imágenes son fundamentales en el procesamiento y posprocesamiento CFD.

La vinculación de las RNAs y en general de la IA con la Dinámica de Fluidos Computacional es un campo poco explorado y hace unos pocos años parecía de poca relevancia y aplicabilidad, sin embargo, hoy existe un panorama diferente, brindando novedosas y diversas alternativas para abordar los problemas de alta complejidad característicos de la Mecánica de Fluidos.

Capítulo 3. Objetivos y Metodología

Objetivos

Objetivo general

Aplicar Redes Neuronales Artificiales para predecir campos de velocidad de flujo alrededor de un cilindro.

Objetivos específicos

Desarrollar una simulación CFD de flujo sobre un cilindro, con resultados válidos.

Procesar los resultados de las simulaciones CFD, para su uso en el aprendizaje mediante Redes Neuronales Artificiales.

Definir los parámetros de implementación, entrenamiento, validación y prueba de las Redes Neuronales Artificiales, cuya función objetivo sea la velocidad de flujo.

Obtener un modelo acertado de Red Neuronal Artificial para la predicción de datos y evaluar sus resultados.

Metodología

La metodología de trabajo propuesta para dar fiel cumplimiento a los objetivos es:

- ✓ Consultar, organizar y sintetizar información teórica referente al flujo sobre un cilindro. La información proveniente especialmente de libros y textos académicos de Mecánica de Fluidos, permitirá comprender de mejor manera el fenómeno en estudio para distinguir sus parámetros y comportamientos característicos inherentes.
- ✓ Consultar, organizar y resumir información proveniente de estudios, artículos científicos, sobre simulaciones CFD, 2D, de flujo sobre un cilindro. Esta información de revistas especializadas, servirá, a más de establecer el estado del arte, para definir aspectos de la simulación CFD a desarrollar, tales como: geometría del dominio computacional, tipo, estructura y densidad del mallado, modelo de turbulencia y métodos de solución. Lo anterior, garantizará la calidad de los resultados CFD, mismos que se evaluarán ante los expuestos en las investigaciones consultadas.

- ✓ Configurar una simulación, 2D, de flujo sobre un cilindro en el aplicativo OpenFOAM® para su ejecución con números de Reynolds entre 5 y 300. Con ello se logrará el banco de datos para su posterior tratamiento y uso con Redes Neuronales Artificiales. Además, con las herramientas del mencionado software CFD, se extraerá información relevante como: residuales, coeficientes, entre otros; para la evaluación de resultados.
- ✓ Evaluar los resultados obtenidos de las simulaciones CFD, 2D, de flujo sobre un cilindro con datos de simulaciones similares expuestos en la literatura investigada. Mediante la comparación de coeficientes (de arrastre y sustentación), la frecuencia de lanzamiento de vórtices (número de Strouhal) y residuales (de presión y velocidad) se validará el comportamiento y el desarrollo temporal de la simulación propuesta. Si los resultados de simulaciones para números de Reynolds de 60, 100, 200 y 300, (representativos en todo el rango de Re estudiado) son coherentes y con diferencias aceptables, se dará como válida la simulación concebida.
- ✓ Validada la simulación, ejecutar simulaciones para números de Reynolds en intervalos de 10 en 10 ($Re = 10, 20, 30, 40, 50 \dots 270, 280, 290$ y 300) para lograr un banco de información organizado y depurado, suficiente para entrenar y validar las RNAs.
- ✓ Clasificar la información en dos grupos: grupo de entrenamiento, validación y prueba, y otro grupo de evaluación de resultados. El primer grupo servirá para entrenar las RNAs y el segundo grupo para validarla.
- ✓ Consultar, organizar y sintetizar información teórica sobre Redes Neuronales Artificiales. Esta información, permitirá contextualizar y comprender las RNAs, sus clases, tipologías, arquitecturas, configuraciones y aplicaciones.
- ✓ Consultar, organizar y resumir información proveniente de artículos científicos, sobre aplicación de Inteligencia Artificial, especialmente RNAs, en casos de flujo sobre un cilindro y otros casos CFD. Esta información especializada, servirá para conformar el estado del arte y establecer aspectos de aplicación de las RNAs, tales como: tipo de RNA, arquitectura (número de capas y neuronas), variables de entrada, variables de salida, tipo de entrenamiento, tipo de evaluación de errores, criterios de parada, número de épocas, entre otros. Garantizando la aplicación de esta técnica de inteligencia artificial.
- ✓ Aplicar, por medio de MATLAB®, Redes Neuronales Artificiales para el aprendizaje de los campos de velocidad. Las RNAs se entrenarán, validarán y probarán a partir de la

información obtenida de las simulaciones CFD y agrupada exclusivamente para este proceso. Los datos a considerar son:

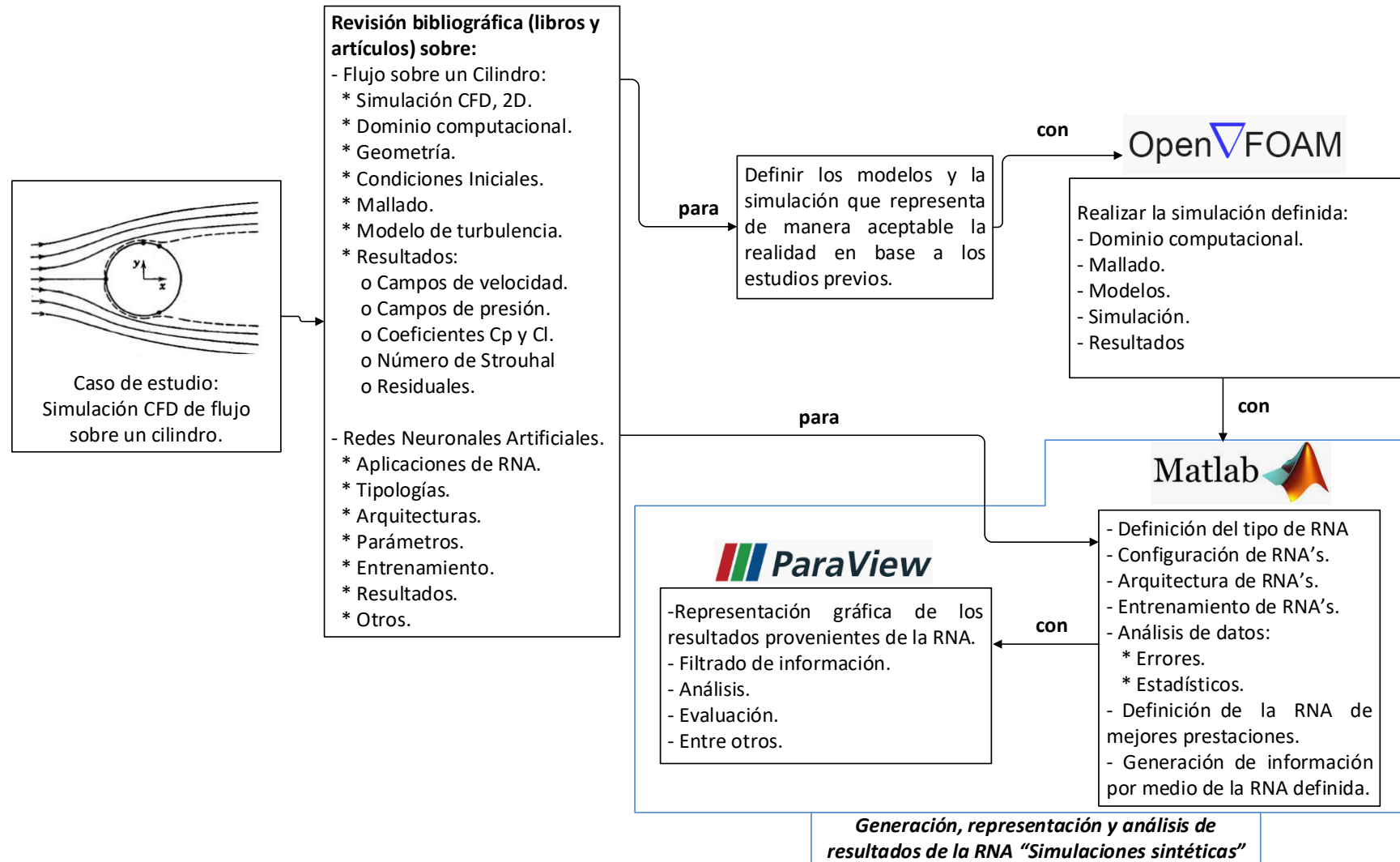
- El número de Reynolds.
- Las coordenadas (x, y) de las celdas del dominio computacional.
- Periodo de los pasos temporales de simulación.
- Los valores de las componentes de velocidad (u, v) para cada celda en los distintos pasos temporales.
- Presión en los nodos, para cada celda en los distintos pasos temporales.

Se probarán distintas arquitecturas de RNA', cambiando el número de capas y de neuronas por capa. Todas estas variantes mantendrán el mismo tipo de RNA, así como los criterios de entrenamiento, validación y prueba.

- ✓ De entre las diversas arquitecturas probadas, definir la RNA de mejores prestaciones. Lo anterior, se basará principalmente en la comparación de los reportes de correlación entre los datos generados por cada RNA y sus similares provenientes de las simulaciones, usados en las etapas de entrenamiento, validación y prueba.
- ✓ Definida la mejor arquitectura de RNA, ejecutarla para predecir los valores de velocidad (u y v) de cada celda del dominio computacional con valores de números de Reynolds pertenecientes al grupo de validación. Dicha información se organizará y vinculará principalmente con el Software ParaView® para su post procesado, visualización como si se tratase de una verdadera simulación CFD.
- ✓ Evaluar los resultados provenientes de la RNA (simulaciones sintéticas) con sus semejantes (simulaciones CFD). Se pondrá a prueba la capacidad de generalización de la RNA en casos de interpolación y extrapolación para números de Reynolds del grupo de validación. Mediante la comparación del comportamiento temporal y espacial de la velocidad de flujo, se establecerá errores absolutos y relativos promedios, para cada celda del dominio computacional y con ellos el nivel de predicción logrado.
- ✓ Discutir y concluir sobre los diversos resultados obtenidos y la aplicabilidad de la RNA en la generalización del caso CFD en estudio.

En la Figura 8 muestra mediante un esquema lo más relevante del proceso metodológico antes descrito.

Figura 8. Representación esquemática de la metodología propuesta para el desarrollo del trabajo de fin de estudio.



Fuente: Elaboración propia.

Capítulo 4. Desarrollo

A continuación, se detalla la aplicación de RNAs en el aprendizaje del campo de velocidad de flujo alrededor de un cilindro, para Re de hasta 300, con el fin de lograr un modelo capaz de reproducir dicha información y a su vez inferir resultados diferentes los de entrenamiento.

Conforme a la metodología, en base a la revisión de artículos especializados, se desarrolla en el aplicativo OpenFOAM® la simulación de flujo sobre un cilindro circular, cuyos resultados se evalúan y validan. Seguido, se procede a la definición y extracción del conjunto de datos de entrenamiento de la RNA (entradas y salidas). Estos datos se procesan para que sean receptados por el entorno de aplicación de RNAs, "Neural Network Toolbox de MATLAB".

Ya en el entorno de entrenamiento, se realizan varias arquitecturas de RNAs, para establecer, de entre ellas, la de mejores prestaciones. Se ejecuta la RNA seleccionada y con los datos obtenidos se generan archivos adecuados que permitan visualizar y analizar, en ParaView, las denominadas "simulaciones sintéticas", para finalmente discutir sobre la capacidad de predicción y generalización de la RNA frente al fenómeno CFD en estudio.

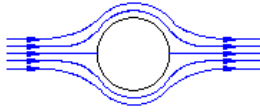



Simulación CFD, 2D, de flujo sobre un cilindro

Este apartado expone la configuración y ejecución de la simulación de flujo sobre un cilindro circular, cuyos resultados serán utilizados en el entrenamiento de las RNAs.

Inicialmente, se define trabajar las simulaciones para Re de hasta 300, pudiendo aplicar un modelo laminar, en este rango de Re se puede estudiar el desprendimiento de estelas hasta la transición turbulenta. Como se puede ver en la Figura 9.

Como se mencionó en el estado del arte, varios estudios utilizan dominios rectangulares, ya sea para bajos o altos números de Reynolds y en base a ello se define el dominio y el mallado para el presente trabajo. El modelo del dominio computacional (mallado y geometría) es tomado específicamente de los escritos (Jiang, 2020; Jiang, Cheng, Draper, et al., 2016), donde son denominados como "estándar" puesto que se eligieron en base a una verificación de dependencia de parámetros, en la que se examinó la idoneidad de la resolución de la malla y el tamaño del dominio mediante una serie de simulaciones 2D. Cumpliendo con lo mencionado por (Posdziech & Grundmann, 2001): buena resolución, adecuada extensión de dominios computacionales y correctas condiciones de borde.

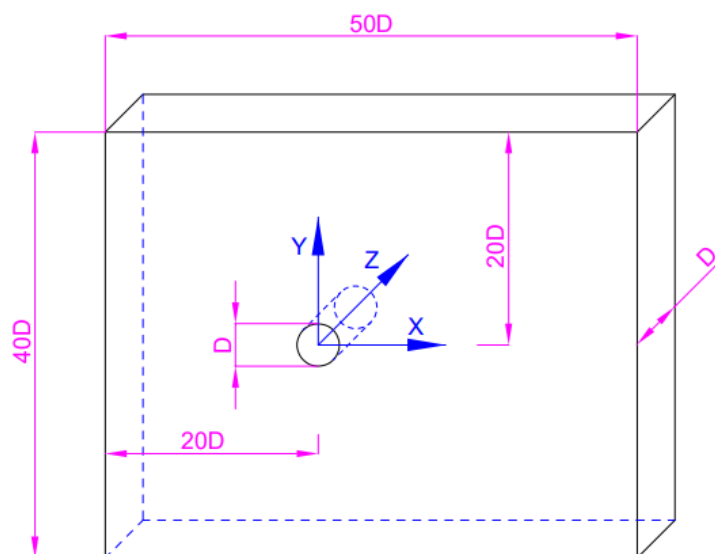
Figura 9. Regímenes de flujo alrededor de un cilindro circular liso en corriente constante.

	Sin separación. Flujo progresivo	$Re < 5$
	Un par fijo de vórtices simétricos	$5 < Re < 40$
	Calle laminar de vórtices	$40 < Re < 200$
	Transición a la turbulencia en la estela	$200 < Re < 300$

Fuente: (Sumer & Fredsøe, 1997).

Con base en el diámetro del cilindro (D), el dominio tiene las siguientes dimensiones: $X = 50D$, $Y = 40D$ y $Z = D$. El cilindro se ubica a $20D$ desde el borde izquierdo (entrada), quedando una longitud de túnel aguas arriba de $20D$ y una longitud de túnel aguas abajo de $30D$. El cilindro está centrado respecto a la altura del dominio, como se muestra en la Figura 10. Se propone medida en la dirección Z , ya que OpenFOAM® lo requiere, pero esta dimensión no se considera en la simulación, por tratarse de un estudio en 2D.

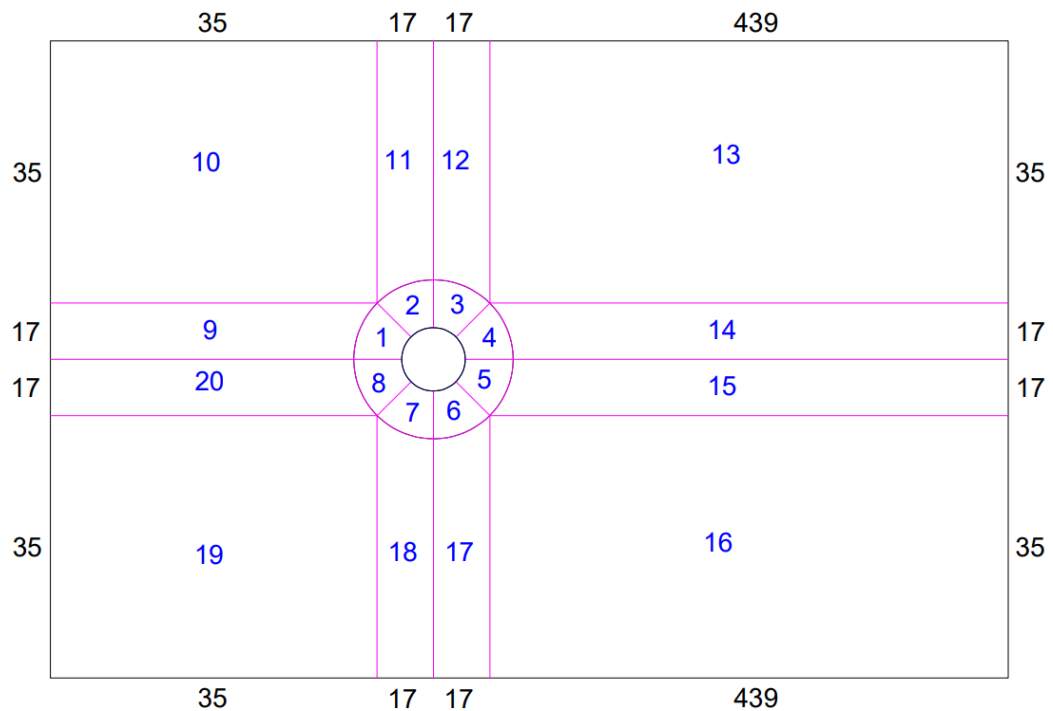
Figura 10. Detalle del dominio computacional asumido para la simulación.



Fuente: Elaboración propia.

Para el mallado en OpenFOAM®, se genera un subdominio concéntrico al cilindro, de diámetro igual a 2.5D. La sección entre el subdominio y el cilindro se divide en ocho partes equidistantes y cada parte a su vez en 17 divisiones, quedando el perímetro del cilindro discretizado en 136 nodos. El resto del dominio se configura de tal manera que coincida con las ocho divisiones de la parte central, resultando un total de 20 zonas. Números de color azul en la Figura 11.

Figura 11. Detalle de las partes del dominio computacional previo al mallado.



Fuente: Elaboración propia.

El mallado parte desde la zona del cilindro hasta el círculo concéntrico con una tasa de expansión de celda a celda igual a 1.1 y se configura de tal forma que el tamaño radial de la primera capa de malla junto al cilindro sea igual a 0.001D. Resultando entonces 46 divisiones en el sentido radial, entre círculos.

Posterior al círculo concéntrico, en las zonas nueve hasta la 12 y 17 hasta la 20 (ver Figura 11) se continúa radialmente con la misma tasa de expansión de celda a celda, 1.1, mientras que en las zonas 13 hasta la 16 la tasa de expansión de celda a celda igual a 1.

La configuración del mallado antes descrito se implementa en OpenFOAM®, con la ayuda del aplicativo “Scripts/blockMesh grading calculation” consultado en: https://openfoamwiki.net/index.php/Scripts/blockMesh_grading_calculation.

Las partes en que queda subdividida cada zona del dominio computacional se muestran en la Figura 11 (con números de color negro), para más detalle ver la Tabla 2.

Tabla 2. Detalle de subdivisión de las diversas zonas del dominio computacional propuesto.

Zona	Divisiones		Celdas	Zona	Divisiones		Celdas
	Radiales	Periféricas			x	y	
1	46	17	782	9	35	17	595
2	46	17	782	10	35	35	1225
3	46	17	782	11	17	35	595
4	46	17	782	12	17	35	595
5	46	17	782	13	439	35	15365
6	46	17	782	14	439	17	7463
7	46	17	782	15	439	17	7463
8	46	17	782	16	439	35	15365
Totalidad de celdas (zona central):			6256	17	17	35	595
				18	17	35	595
				19	35	35	1225
				20	35	17	595
				Totalidad de celdas (alrededores)			51676
Totalidad de celdas del dominio: 57932							

Fuente: Elaboración propia.

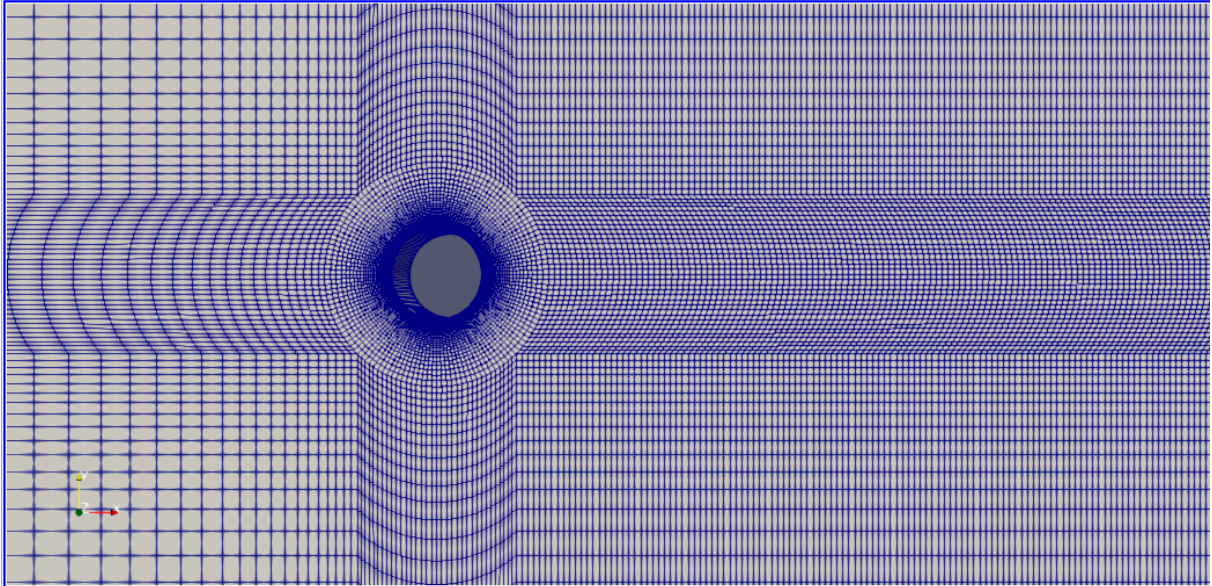
La malla generada posee la totalidad de 57932 celdas, conformadas por 58612 nodos y se detalla en la Figura 12.

En cuanto al modelo numérico, se utiliza el resolvidor denominado “IcoFoam”, el cual, según la página oficial de OpenFOAM®, resuelve las ecuaciones laminar incompresibles de Navier-Stokes utilizando el algoritmo PISO (presión implícita con división de operadores) para problemas transitorios. En la simulación, las condiciones de contorno son:

- ✓ En la entrada, “inlet” (lado izquierdo), se especifica una velocidad fija de flujo uniforme.
- ✓ En la salida, “outlet” (lado derecho), se aplica la condición de frontera del tipo Neumann, siendo la gradiente de velocidad normal igual a cero, y la presión con un valor de referencia de cero.
- ✓ En las fronteras superior e inferior, “top” y “bottom”, se aplica una condición de simetría, ya que no se considera que el obstáculo se encuentra en un medio infinito.

- ✓ Se ignoran las regiones frontal y posterior “front” y “back”, por tratarse de una simulación en dos dimensiones.
- ✓ Se aplica la condición de No Slip, para la superficie del cilindro.

Figura 12. Detalle de la malla en las cercanías del cilindro.



Fuente: Elaboración propia.

Para las diversas simulaciones se asume un diámetro de cilindro igual a 0.1m y una viscosidad cinemática de 0.002564103 m²/s propuesta en (Casacuberta et al., 2014), con estos valores y la ecuación del número de Reynolds, se obtienen las velocidades de flujo. En la Tabla 3 se muestran algunos de los valores calculados.

Tabla 3. Detalle de algunos valores de velocidad de flujo para las simulaciones.

Número de Reynolds (Re)	Diámetro del cilindro (D)	Viscosidad Cinemática (ν)	Velocidad de flujo (v)
(-)	(m)	(m ² /s)	(m/s)
5	0.1	0.002564103	0.1282
10			0.2564
50			1.2821
100			2.5641
200			5.1282
300			7.6923

Fuente: Elaboración propia.

Inicialmente se realizan cuatro simulaciones, para Re de 60, 100, 200 y 300, con el propósito de analizar los residuales y evaluar: el coeficiente de arrastre (C_l), el coeficiente de sustentación (C_d) y el número de Strouhal (St).

El paso temporal propuesto es 0.00001, con ello el número de Courant (Co) se mantiene por debajo de 0.5 en todo el dominio computacional.

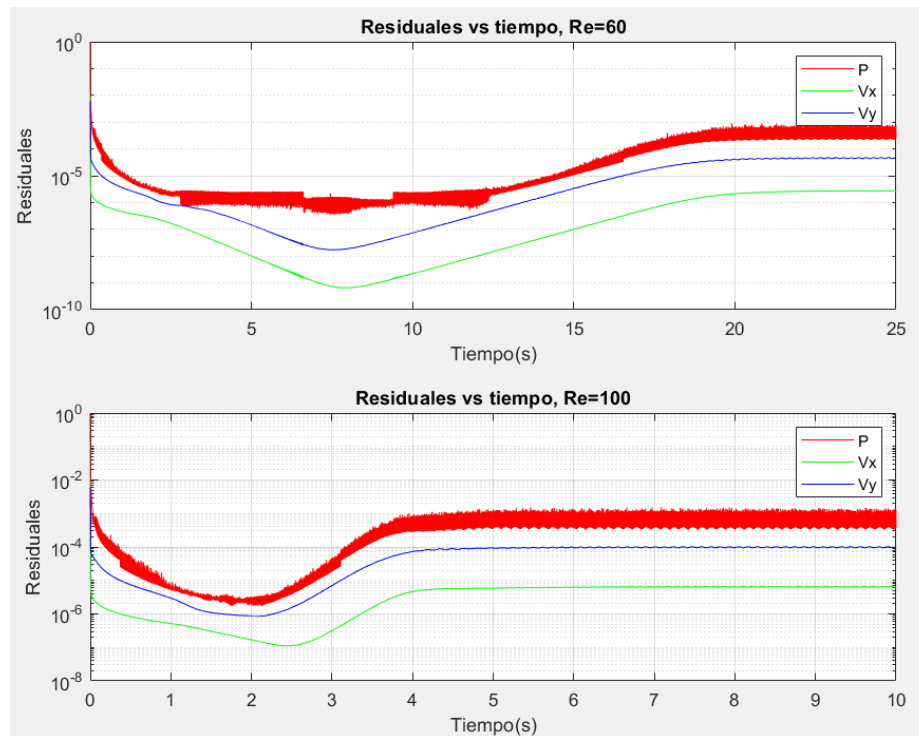
Las simulaciones se realizan con un computador Intel Core™ i3-3110M CPU @ 2.4GHz, 8GB RAM. Donde, para un caso típico simulado se usan tres procesadores en paralelo, lo que conlleva aproximadamente 25.3h de tiempo de reloj para la simulación numérica de hasta 500000 pasos de tiempo adimensionales.

En la programación se configura que los estados temporales de presión y velocidad se generen cada 10000 pasos temporales, mientras que para C_l y C_d sean cada 1000 pasos. Los coeficientes C_l y C_d sobre el cilindro son calculados en OpenFOAM®, a través de la librería “*libforces.so*”. El detalle de toda la programación se muestra en el Anexo 4.

Las simulaciones se ejecutan sin contratiempo alguno, reportando adecuadamente los datos de interés. En la Figura 13 se observa que, en la simulación para $Re = 60$, deben transcurrir alrededor de 2000000 de pasos de tiempo para lograr la estabilidad en los residuales de presión y velocidades, con valores por debajo de los 10^{-3} .

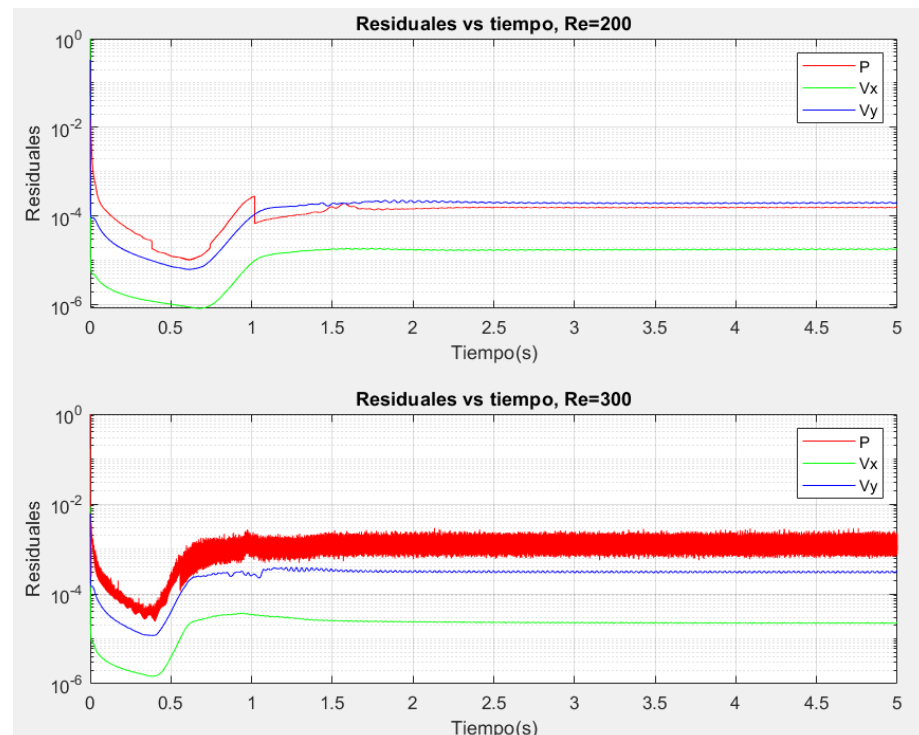
Para el resto de simulaciones de validación, el tiempo para lograr la estabilidad se reduce, llegando a ser aproximadamente: 400000 pasos temporales para $Re = 100$, 150000 pasos temporales para $Re = 200$ y finalmente, 100000 pasos temporales para $Re = 300$.

Figura 13. Residuales de presión y de velocidades para las simulaciones de flujo sobre un cilindro, con números de Reynolds de 60 y 100.



Fuente: Elaboración propia.

Figura 14. Residuales de presión y de velocidades para las simulaciones de flujo sobre un cilindro, con números de Reynolds de 200 y 300.

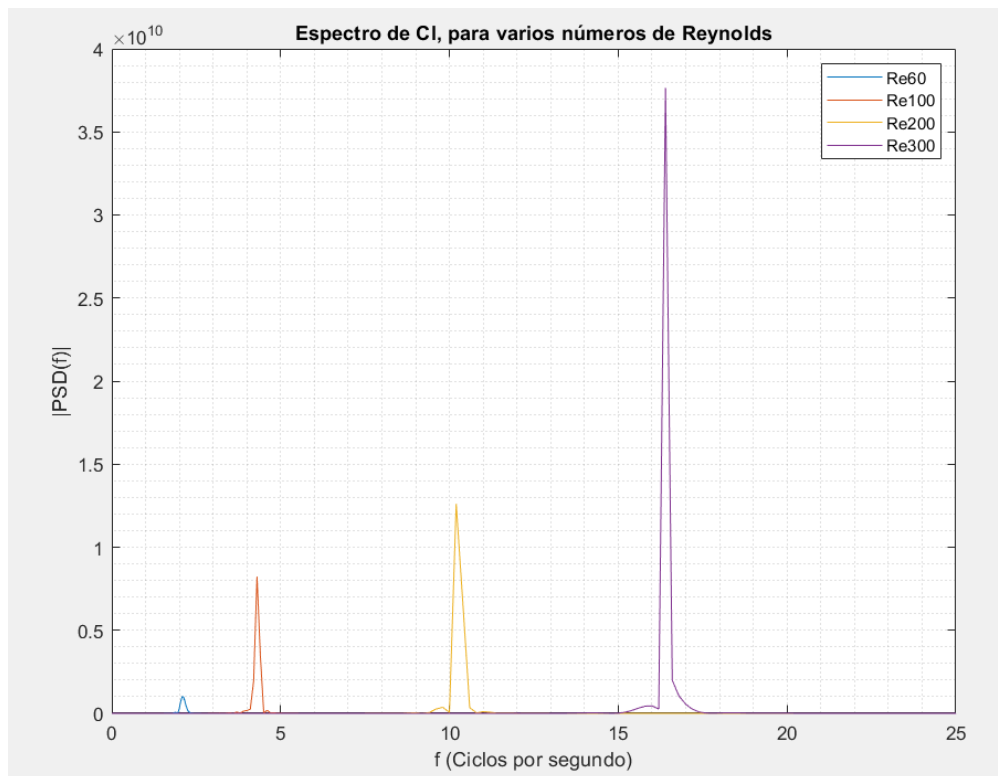


Fuente: Elaboración propia.

En las Tabla 5 hasta la Tabla 8, se muestran estados temporales de las simulaciones evaluadas, observándose el comportamiento típico del flujo sobre un cilindro, reportado en la literatura.

Los resultados de C_l y C_d , provenientes de OpenFOAM®, se procesan en MATLAB®, donde se generan las gráficas de interés y en base a la serie temporal de C_l se calcula, para cada simulación, el espectro de la señal y con ello el número de Strouhal. La Figura 15 muestra la densidad espectral (PSD) de las diferentes simulaciones evaluadas.

Figura 15. Densidad espectral de C_l para las diversas simulaciones evaluadas.



Fuente: Elaboración propia.

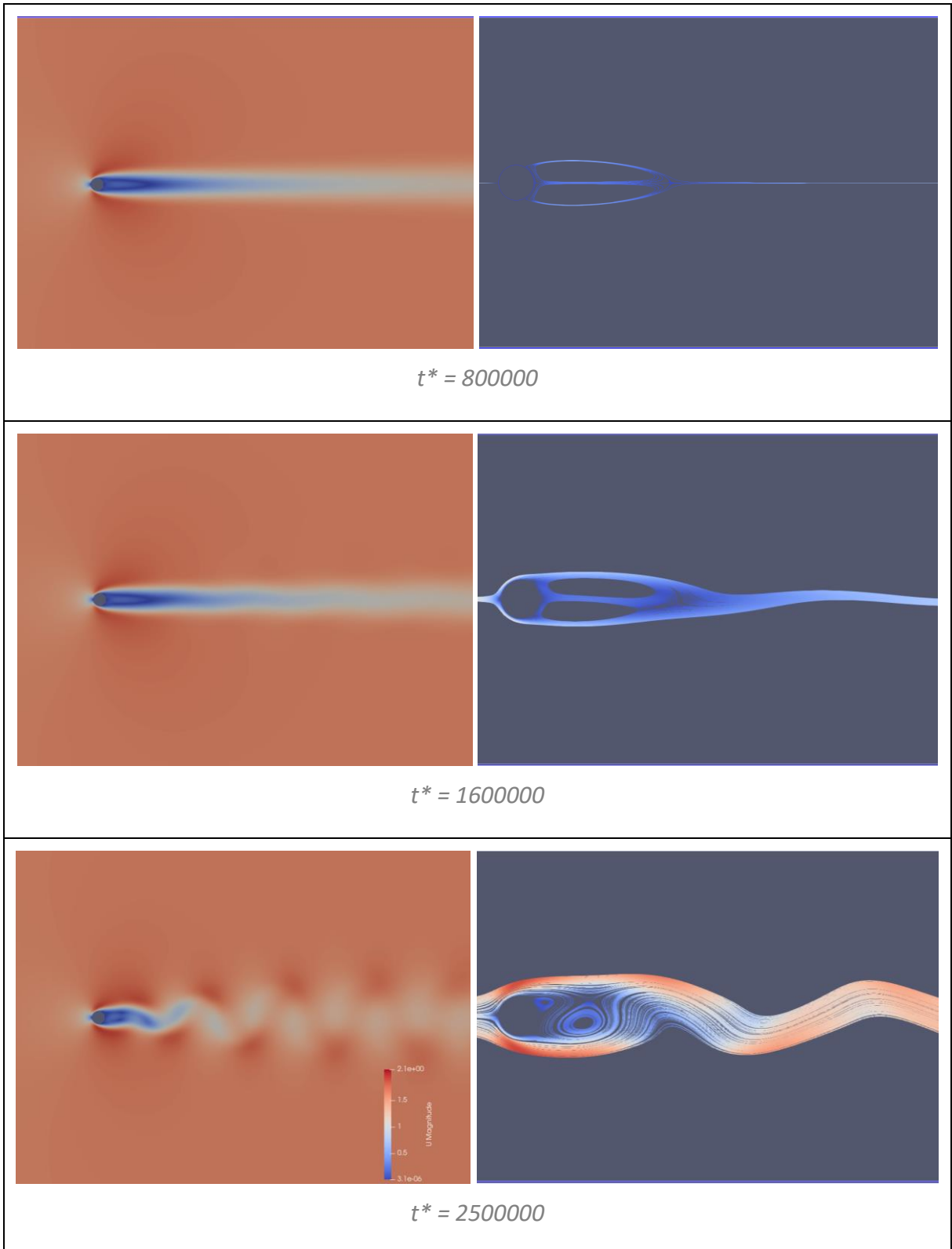
Resultado de la densidad espectral de las señales de los C_l , se obtienen las frecuencias de desprendimiento de los vórtices en la estela, mostrados en la Tabla 4.

Tabla 4. Valores de frecuencia de desprendimiento de vórtices en la estela para las simulaciones evaluadas.

Número de Reynolds	Frecuencia de vórtices en la estela
60	2.080
100	4.300
200	10.199
300	16.399

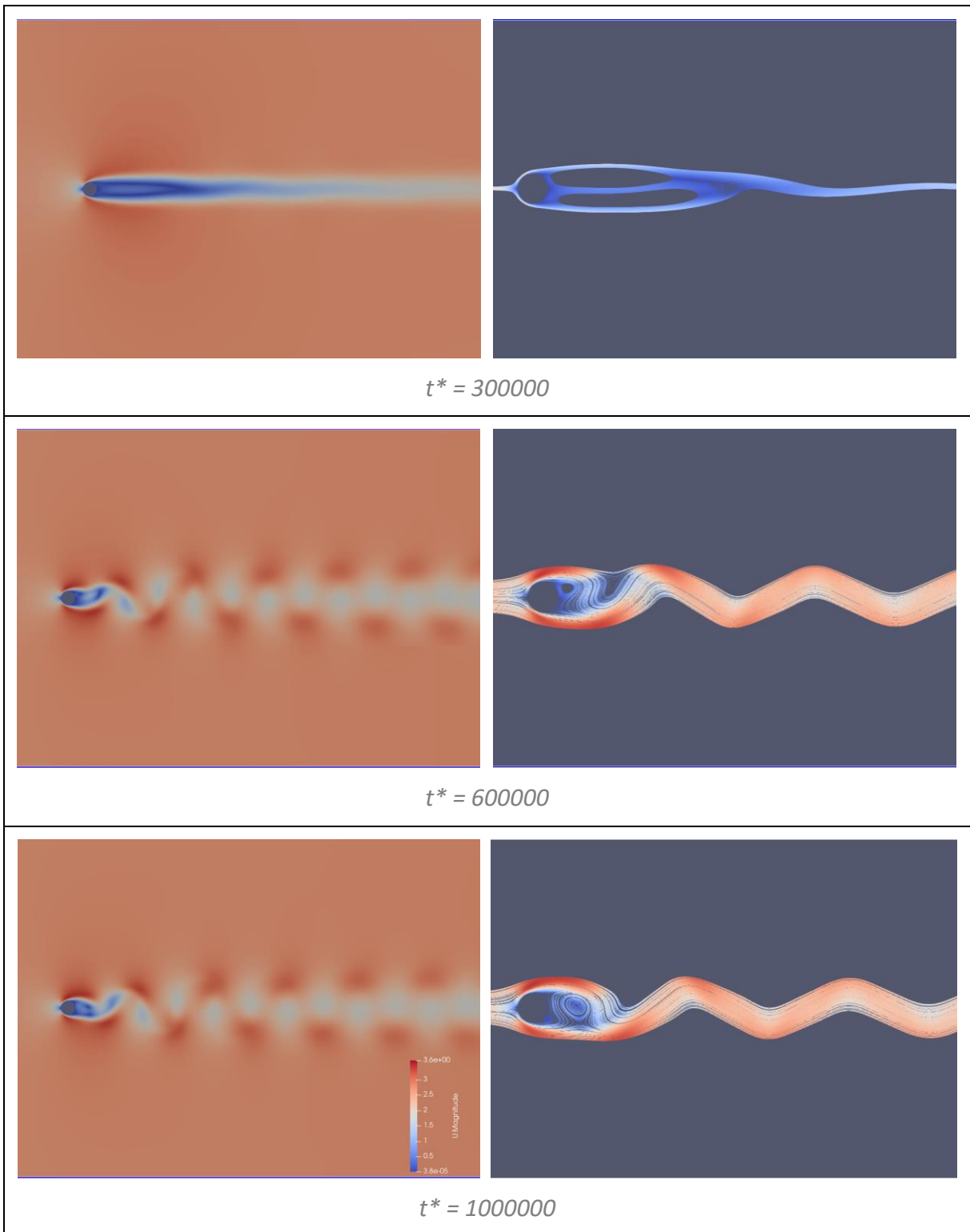
Fuente: Elaboración propia.

Tabla 5. Campos de velocidad y líneas de corriente del flujo sobre un cilindro para $Re = 60$.



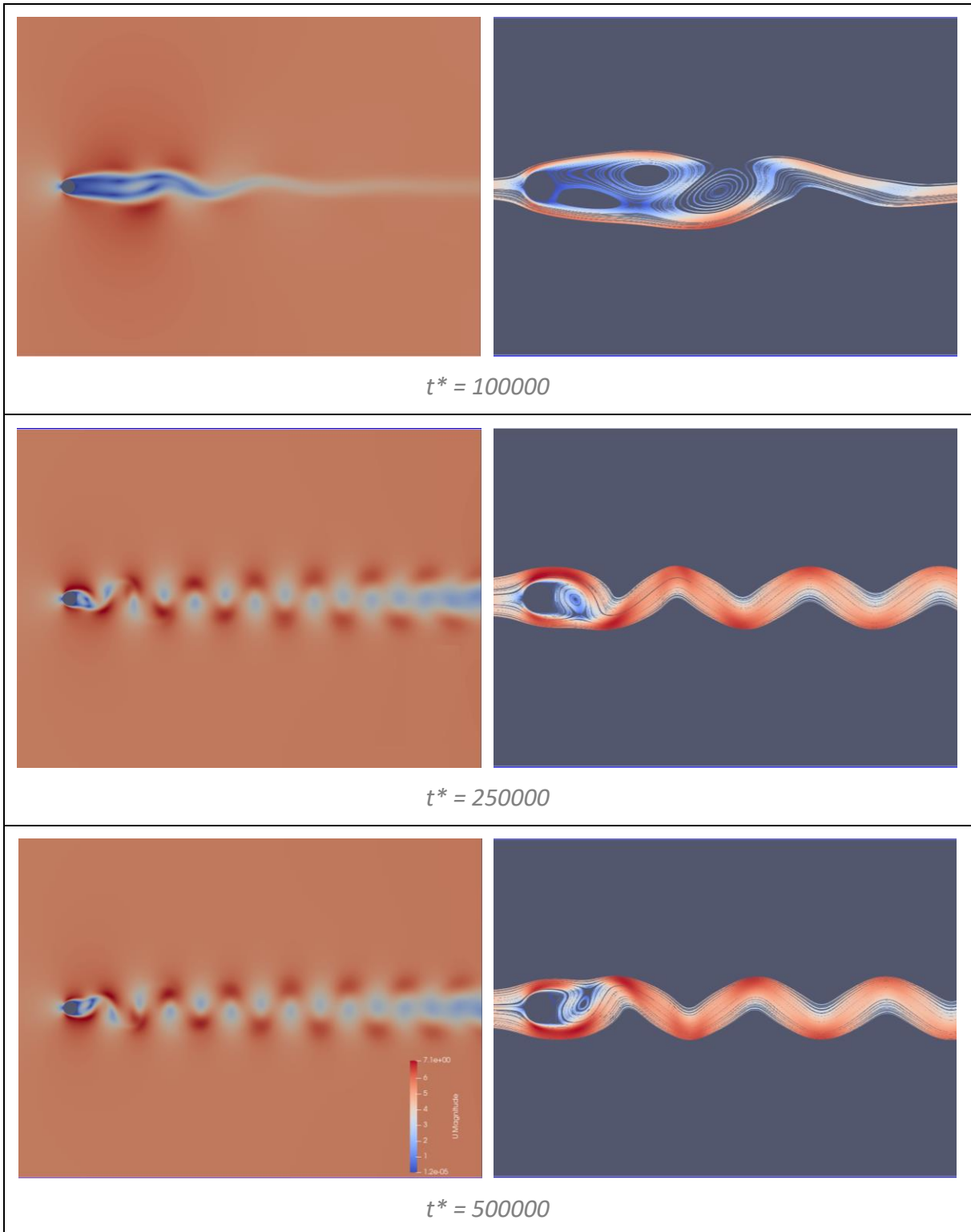
Fuente: Elaboración propia.

Tabla 6. Campos de velocidad y líneas de corriente del flujo sobre un cilindro para $Re = 100$.



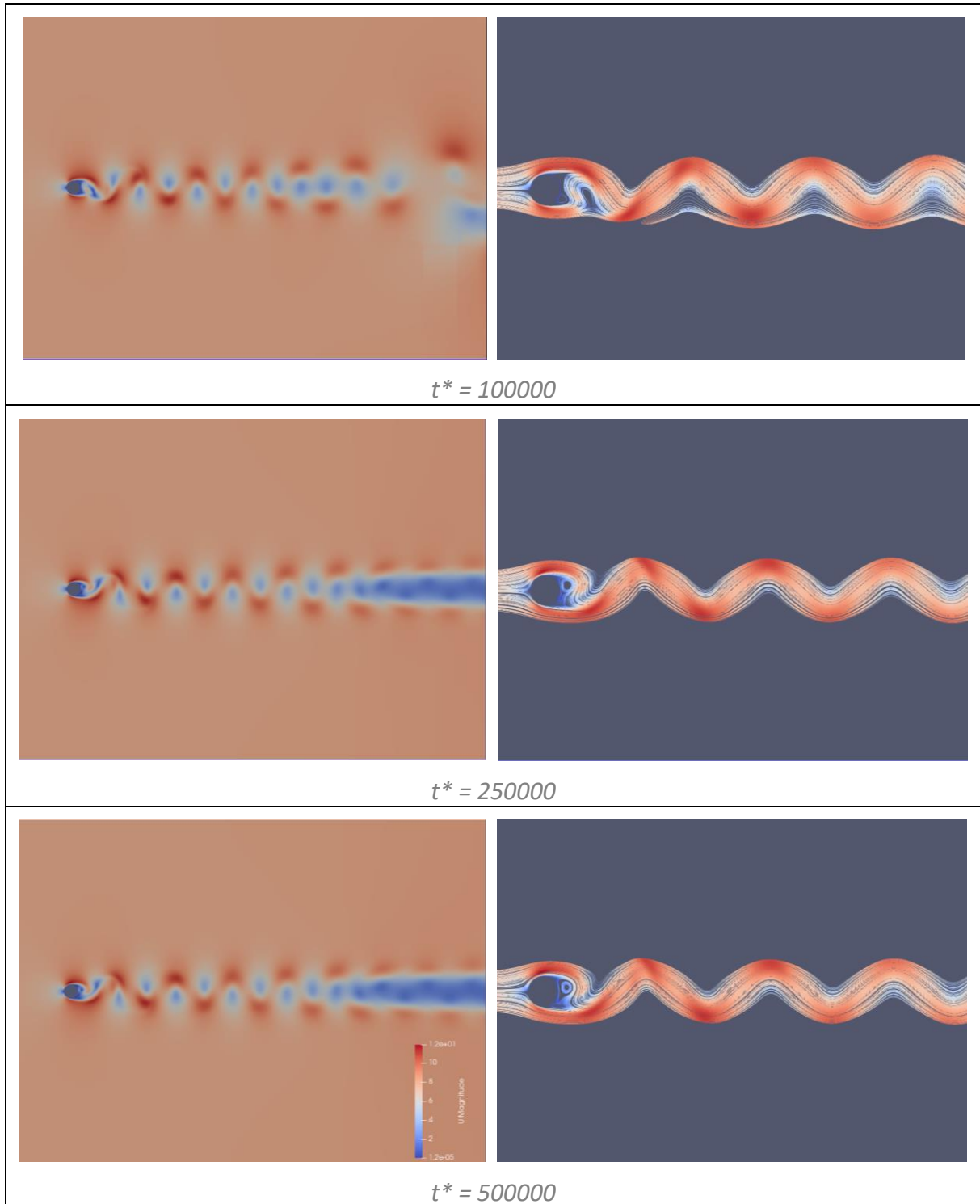
Fuente: Elaboración propia.

Tabla 7. Campos de velocidad y líneas de corriente del flujo sobre un cilindro para $Re = 200$.



Fuente: Elaboración propia.

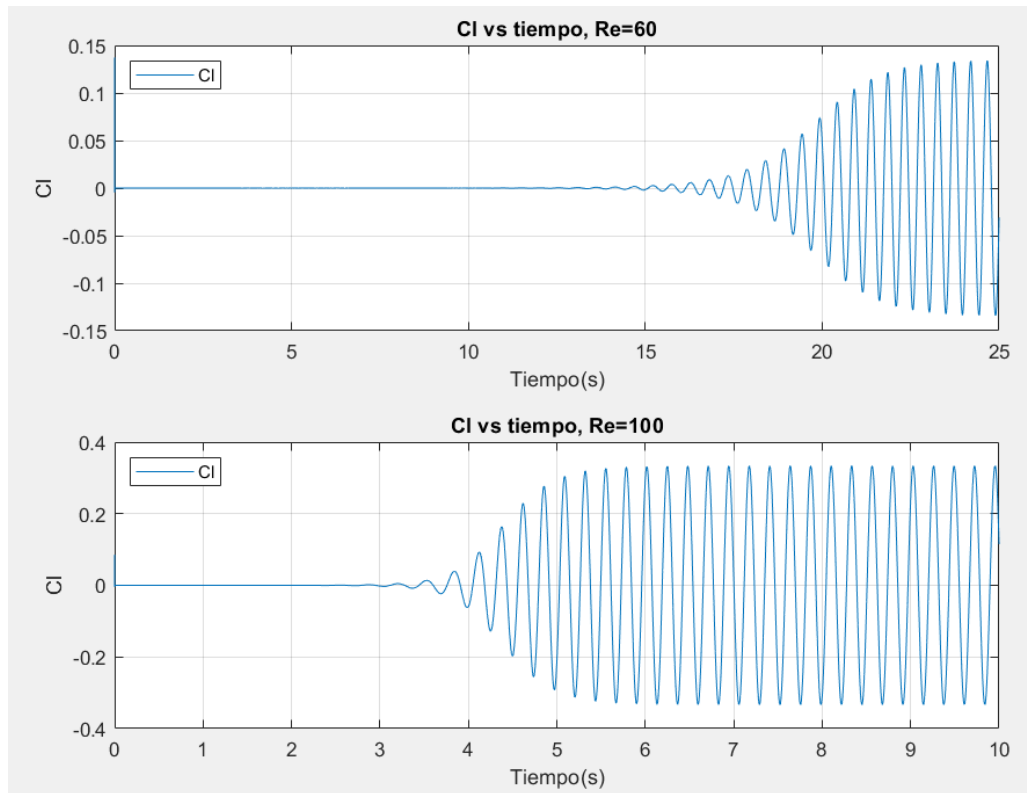
Tabla 8. Campos de velocidad y líneas de corriente del flujo sobre un cilindro para $Re = 300$.



Fuente: Elaboración propia.

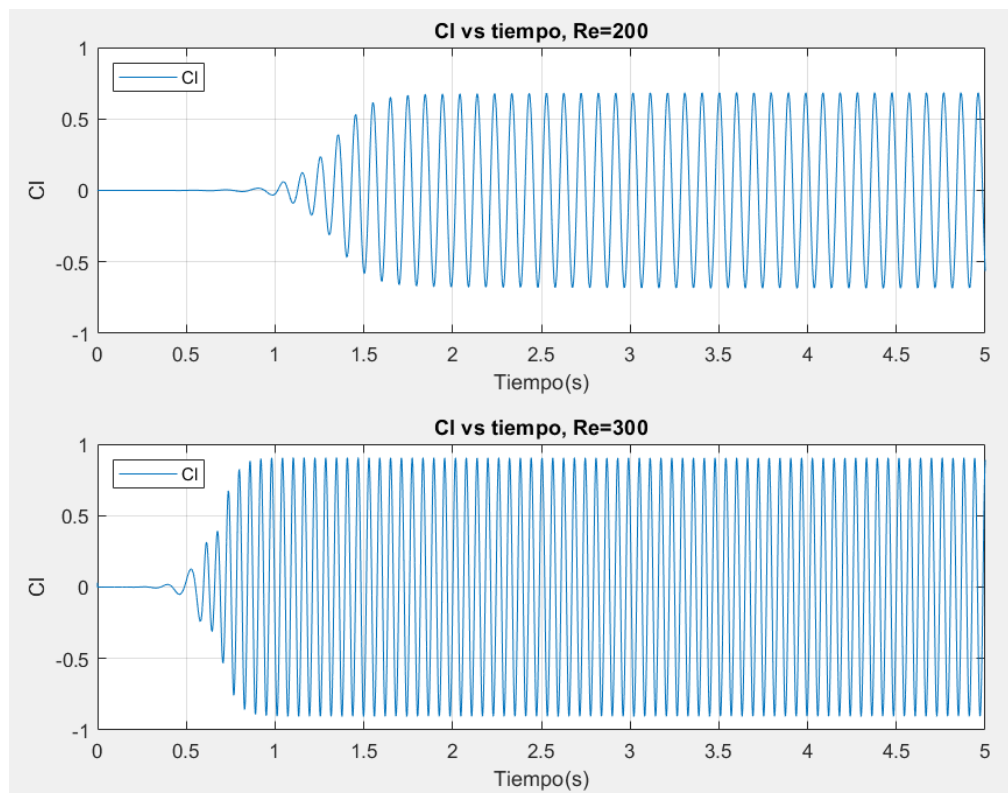
En base a lo expuesto por (Norberg, 2001; Rapaka & Sarkar, 2016), para la obtención de C_L , se calcula el r.m.s. (root-mean-square) de la señal, mientras que, para el caso de C_D se calcula el valor medio de la señal, todo lo anterior, luego del transitorio inicial. Las señales de los coeficientes C_D y C_L de los casos evaluados se muestran en la Figura 16 hasta la Figura 19.

Figura 16. Señales de C_l obtenidas de la simulación de flujo sobre un cilindro con $Re = 60$ y 100 .



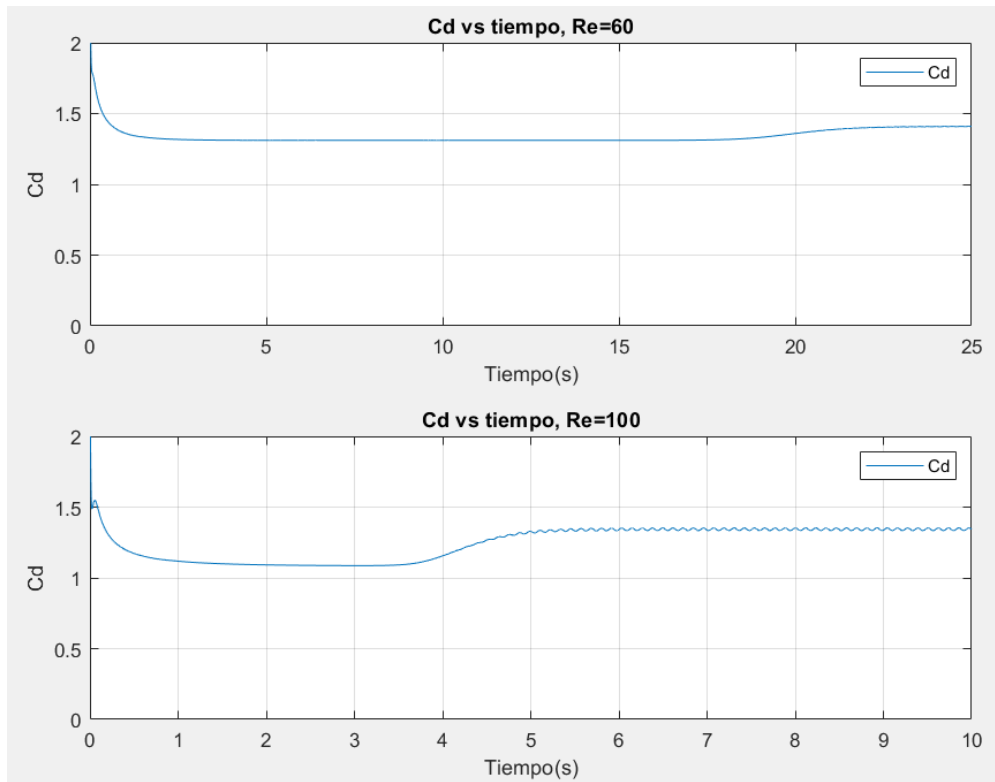
Fuente: Elaboración propia.

Figura 17. Señales de C_l obtenidas de la simulación de flujo sobre un cilindro con $Re=200$ y 300 .



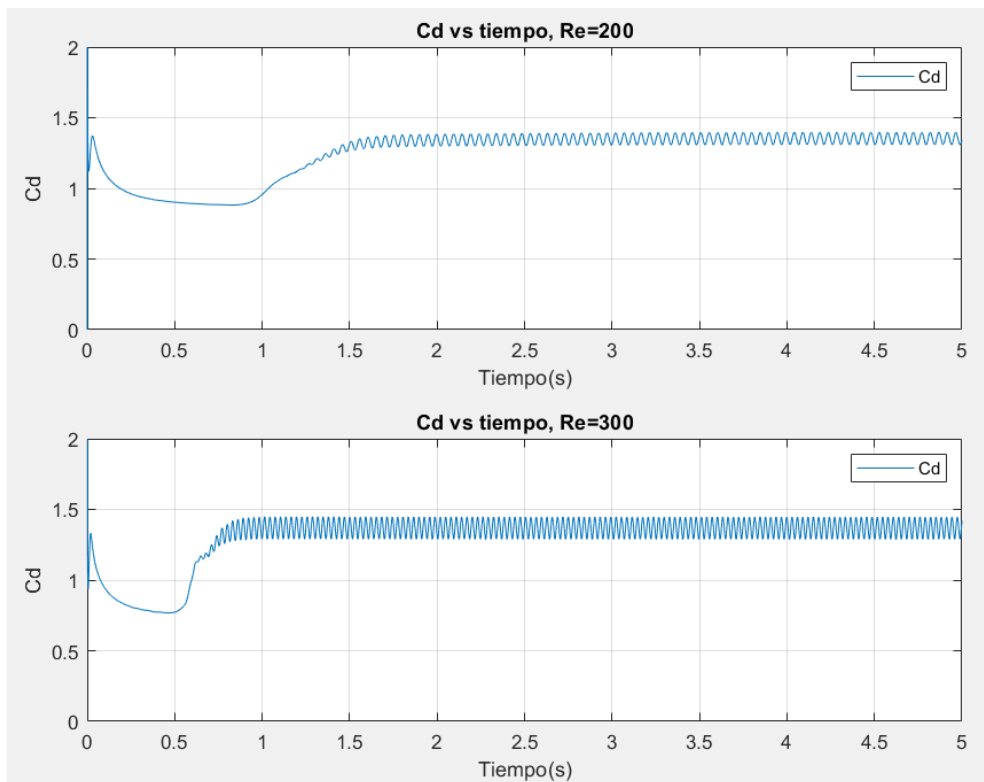
Fuente: Elaboración propia.

Figura 18. Señales de C_p obtenidas de la simulación de flujo sobre un cilindro con $Re=60$ y 100 .



Fuente: Elaboración propia.

Figura 19. Señales de C_p obtenidas de la simulación de flujo sobre cilindro con $Re=200$ y 300 .



Fuente: Elaboración propia.

Como se mencionó, se determinan los valores de C_L , C_D y St para los casos de prueba con el fin de validar la simulación propuesta. Los resultados se comparan con los obtenidos por diversos autores de artículos en revistas especializadas, ver Tabla 9, donde se observa la similitud de valores y un buen acuerdo de resultados.

De esta manera se da como aceptable la simulación de flujo sobre un cilindro propuesta y el respectivo código se expone en el Anexo 4.

Con dicho código se ejecutan un total de 31 simulaciones, para Re desde 5 hasta 300, en intervalos de 10 en 10. Con el fin de generar Información para el entrenamiento, validación y análisis de las RNAs que predecirán los campos de velocidad, objeto del presente trabajo de investigación.

Tabla 9. Comparación de resultados de las simulaciones evaluadas con lo expuesto en la literatura consultada.

Parámetros	Coeficiente de arrastre (C_D)				Coeficiente de Sustentación (C_L)				Número de Strouhal (St)			
	Re = 60	Re = 100	Re = 200	Re = 300	Re = 60	Re = 100	Re = 200	Re = 300	Re = 60	Re = 100	Re = 200	Re = 300
(Jiang, Cheng, Draper, et al., 2016)	--	--	1.342	1.377	--	--	0.474	0.642	0,137	0,166	0,197	0,210
(Posdziech & Grundmann, 2001)	1.387	1.325	1.327	1.363	--	--	0.475	--	0.135	0.164	0.196	0.209
(Braza et al., 1986)	1.421	1.291	1.309	1.220	--	0.25	0.75	--	0.139	0.164	0.181	0.205
(Rapaka & Sarkar, 2016)	--	1.361	--	1.370	--	--	--	--	--	0.168	--	0.211
(Henderson, 1995)	1.416	1.356	1.359	1.395	--	--	--	--	--	--	--	--
(Norberg, 2001)	--	--	--	--	0.112	0.208	0.457	0.669	0.136	0.165	0.195	0.201
(Harichandan & Roy, 2010)	--	1.352	1.325	--	--	0.278	0.602	--	--	0.161	0.192	--
(Dennis & Chang, 1970)	1.303	1.27	--	--	--	--	--	--	--	--	--	--
(Ding et al., 2004)	--	1.325	1.327	--	--	0.28	0.60	--	--	0.164	0.196	--
(Jiang & Cheng, 2017)	--	--	--	--	--	--	--	--	0.170	0.178	0.174	0.174
(Piñol & Grau Vidal, 1996)	1.341	1.212	1.046	--	0.07	0.15	0.27	--	0.129	0.156	0.181	--
(Kozlov et al., 2011)	1.521	1.426	1.415	--	0,116	0,261	0,534	--	0.145	0.172	0.202	--
	1.415	1.348	1.280	--	0,116	0,261	0.508	--	0.133	0.162	0.183	--
(Park et al., 1998)	1.39	1.33	--	--	0.134	0.332	--	--	0.136	0.165	--	--
(Rajani et al., 2009)	1.330	1.335	1.336	1.366	--	0.179	0.424	0.604	0.133	0.156	0.195	0.215
(Tritton, 1959)	1.47	1.24	--	--	--	--	--	--	--	--	--	--
(Chopra & Mittal, 2019)	1.383	1.319	--	--	0,088	0,225	--	--	0.134	0.164	--	--
Presente estudio	1.397	1.327	1.349	1.369	0.093	0.236	0.481	0.639	0.134	0.166	0.196	0.211

Fuente: Elaboración propia.

Pre-procesamiento de datos para el entrenamiento de las RNAs.

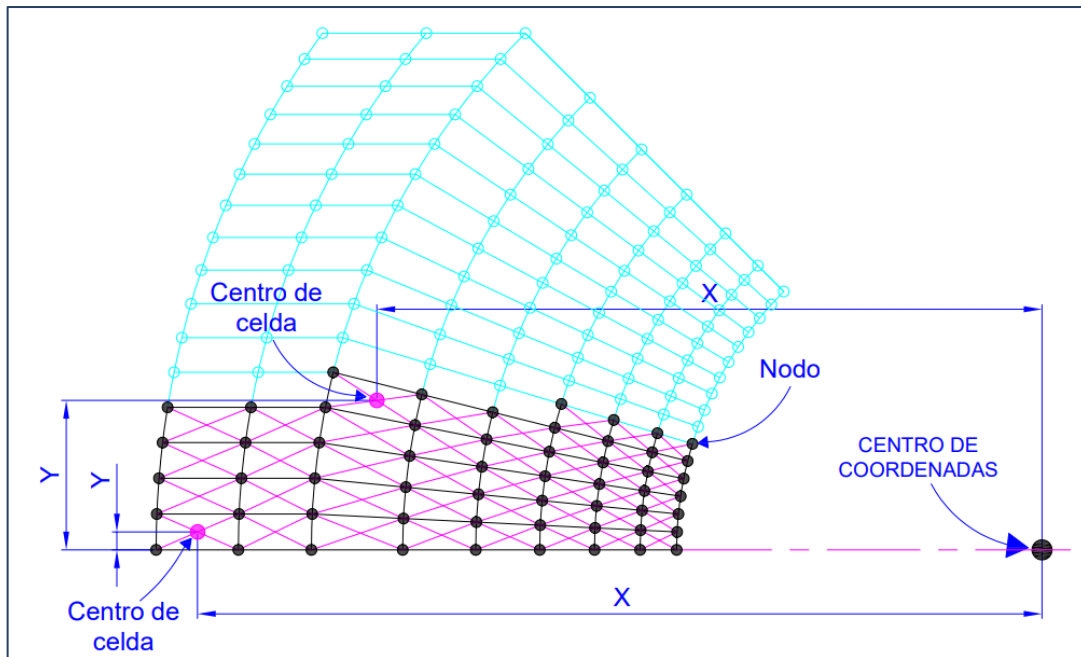
Con la información generada en las simulaciones, se procede a pre-procesar los datos de tal manera que puedan ser aceptados por el entorno de entrenamiento de las RNAs.

Los datos de interés son: Re , las coordenadas (X e Y), el tiempo de ejecución, las velocidades (u y v) y la presión. Debido a la gran cantidad de datos y a la escasa capacidad de cómputo (de los CPU que posee el autor del trabajo) se propone delimitar los datos a lo siguiente:

- ✓ De las 31 simulaciones generadas, se utilizan los datos de 15 de ellas, que van con Re desde 10 hasta 300 en intervalos de 20 en 20 ($Re = 10, 30, 50, \dots, 270, 290$), el resto de simulaciones ($Re = 5, 20, 40, \dots, 280, 300$) servirán para evaluar los resultados de las RNAs.
- ✓ Los datos de interés se extraen de las celdas del dominio computacional, no de los nodos, reduciendo así la información a procesar. Entonces, inicialmente se procesan 57932 datos de presión y de velocidad en sus componentes X e Y. En la componente Z no existe información por tratarse de un caso 2D.
- ✓ Para reducir aún más la información, del dominio computacional se excluyen zonas del campo lejano, ya que no cambian significativamente su estado de presión y velocidad durante las simulaciones. Utilizándose entonces, un sub dominio cercano al cilindro que se extiende por toda la zona de desarrollo de la estela. Ver Figura 21.
- ✓ Finalmente, de los estados temporales reportados en cada simulación, se utilizan los correspondientes al periodo de 0 a 5 s (0 – 500000 pasos temporales) con avances de 0.1 s (10000 pasos temporales). Lo expuesto se justifica, debido a que en este lapso la mayoría de simulaciones (desde $Re = 80$ hasta $Re = 300$) logra superar el transitorio inicial, desarrollando completamente su estela. De lo anterior, se cuenta con los campos de presión y velocidad para los tiempos: 0.1 s, 0.2 s, 0.3 s, ...1.0 s, 1.1 s, ...4.8 s, 4.9 s y 5.0 s. En total 50 estados temporales por cada simulación.

Respecto a la información de cada celda, esta es reportada por OpenFOAM® en carpetas, una para cada 0.1 s de simulación, pero no existe reporte de las coordenadas de las celdas, puesto que se conforman por un grupo de cuatro nodos (cuadriláteros). Siendo necesario definir un punto de referencia de celda. Para ello, se procede en MATLAB® a establecer las coordenadas del punto medio de las celdas a partir de las coordenadas de los nodos que las componen. La Figura 20 muestra cómo se definen los puntos representativos de las celdas.

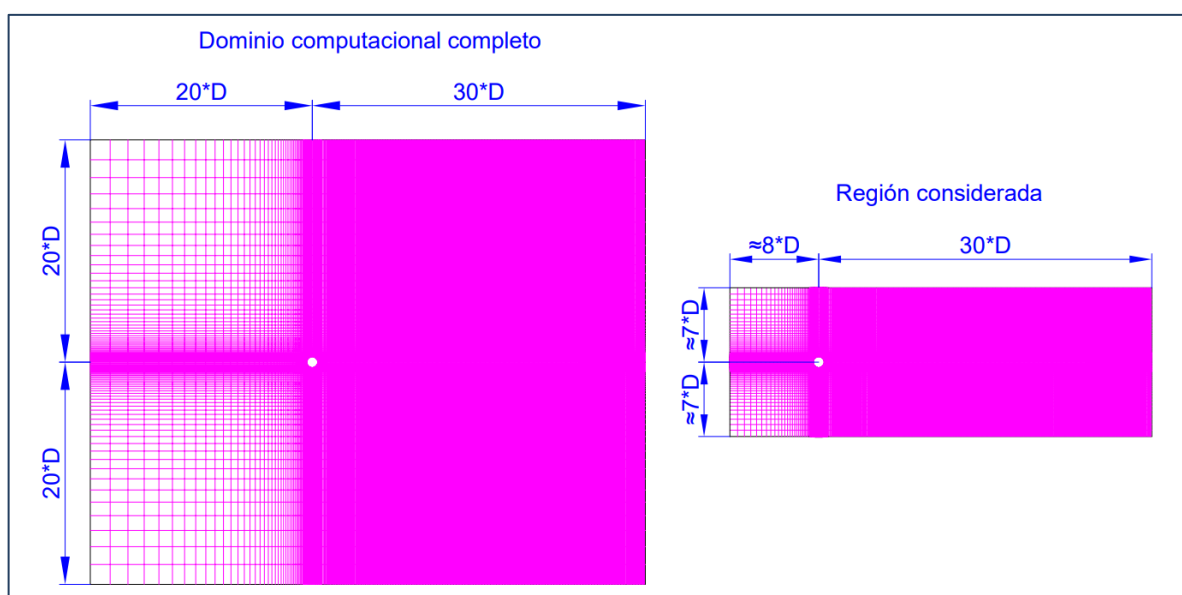
Figura 20. Detalle gráfico de obtención de las coordenadas representativas para cada celda del dominio computacional, aplicada sobre un extracto del dominio.



Fuente: Elaboración propia.

El sub dominio de donde se extrae la información posee las siguientes dimensiones: $X = 38D$ e $Y = 14D$. El borde izquierdo de esta zona queda aproximadamente a $8D$ del centro del cilindro, se mantiene la longitud de túnel aguas abajo de $30D$ y el cilindro está centrado respecto a la altura de este subdominio, como se muestra en la Figura 21.

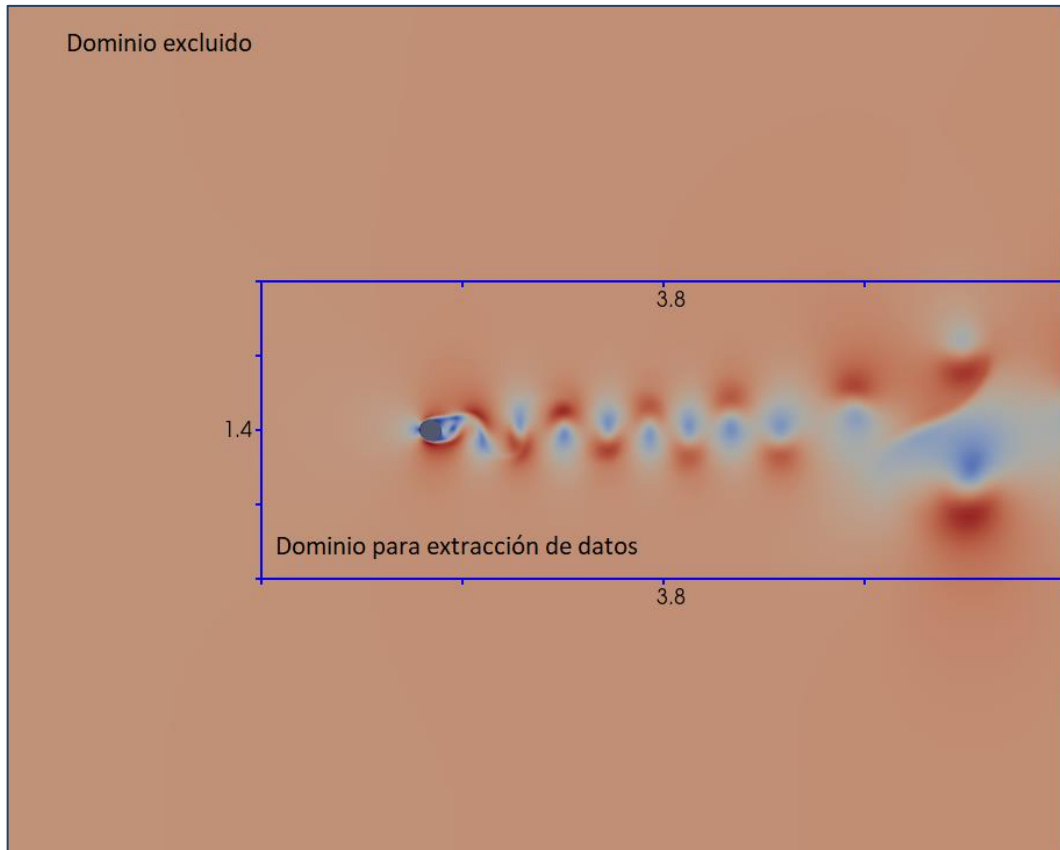
Figura 21. Detalle de la región del dominio computacional, de la cual se extrae la información de interés.



Fuente: Elaboración propia.

Luego de la adopción del subdominio, se extrae la información mediante MATLAB® quedando un total de 44940 celdas utilizadas y 12992 celdas excluidas. La Figura 22 muestra la división del dominio computacional dentro de un campo de velocidad para $Re = 300$, $t = 0.9$ s.

Figura 22. Detalle de subdominios: excluido y considerado para la extracción de información, previo al entrenamiento de las RNAs.



Fuente: Elaboración propia.

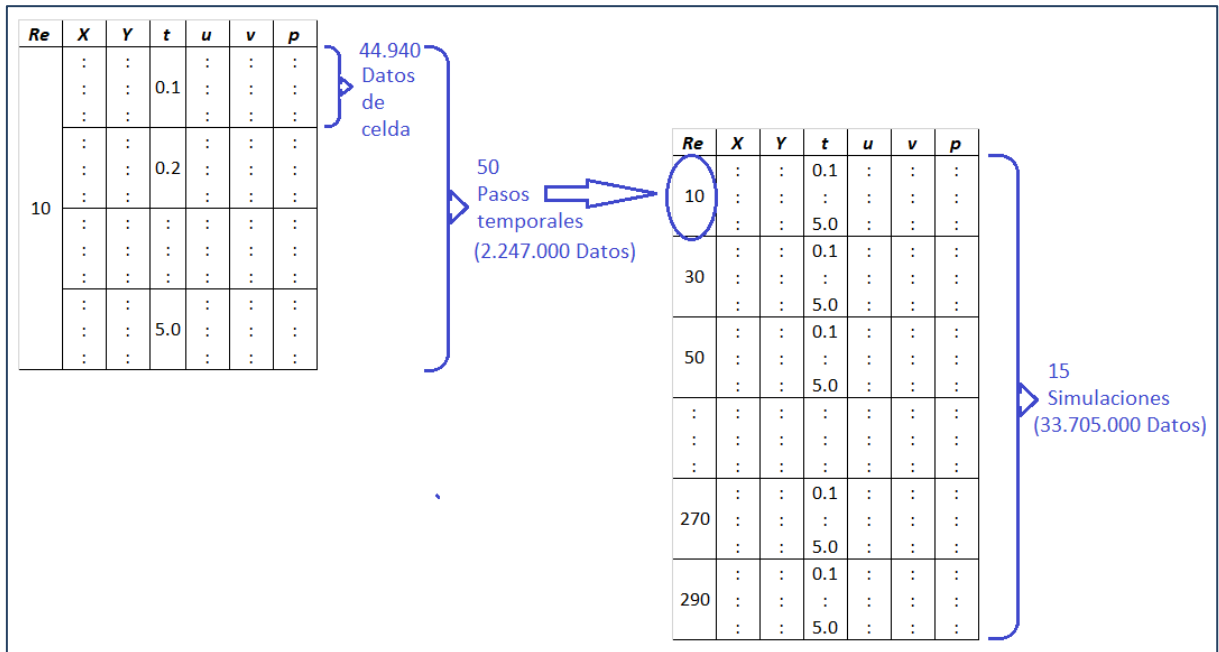
Con las consideraciones previas, se estructura y ordena la información para el entrenamiento. Mediante MATLAB® se crea una tabla en cuyas columnas se ubican las variables en el siguiente orden:

1. Número de Reynolds.
2. Coordenada X del punto central de celda.
3. Coordenada Y del punto central de celda.
4. Paso temporal
5. Componente X de velocidad de flujo en cada celda.
6. Componente Y de velocidad de flujo en cada celda.
7. Presión en cada celda.

En cambio, en las filas se emplazan los 44940 datos de celda, para los 50 pasos temporales de cada una de las 15 simulaciones, dando un total de 33705000 filas. Los datos de las filas están ordenados de forma ascendente primeramente respecto a Re y luego al tiempo.

Las coordenadas de las celdas avanzan desde la zona uno hasta la 20. La Figura 23 muestra la estructura y orden de los datos.

Figura 23. Detalle de organización de los datos, previo a la aplicación de las RNAs.



Fuente: Elaboración propia.

De esta manera queda lista la información para su aplicación en las RNAs.

RNA para el aprendizaje y predicción de resultados de velocidad.

Una vez estructurados y organizados los datos de interés, son utilizados en el proceso de entrenamiento de RNAs para predecir los valores de velocidad en cada celda del dominio computacional, para los diferentes pasos de tiempo y los distintos Re.

En este estudio, en la aplicación de las RNAs se procede de la siguiente manera:

1. Para comparar valores con diferentes órdenes de magnitud, se normaliza los datos entre -1 y 1, esto favorece la aplicación de varias RNAs ya que son más eficientes al entrenarse con datos en este rango o rangos similares (Ledesma, 2006).
2. Con base a trabajos similares y recomendaciones de literatura especializada, se define una estructura de partida de las RNAs, el modelo y algoritmo de entrenamiento; las funciones de activación y las condiciones de parada.

3. Se crea dos subgrupos de datos; las entradas y las salidas, dispuestas de la manera requerida por el software de aplicación de las RNAs.
4. Se entrena las diversas configuraciones de RNAs y se evalúa los indicadores resultantes para establecer cuál RNA es la de mejores características. Debido a que durante el entrenamiento existen procedimientos aleatorios, se realizan múltiples “experimentos” hasta en alguno de ellos conseguir un menor error al evaluar el conjunto de datos de entrenamiento (Ledesma, 2006).
5. Se evalúa la capacidad de predicción de resultados de la configuración obtenida.

La normalización puede considerarse como una técnica de pre-procesamiento, en este caso los datos de entrada como de salida quedarán delimitados en el rango de -1 y 1, para lo cual se aplica la fórmula obtenida en (Caicedo B & López S, 2009):

$$x_n = 2 \cdot \left(\frac{x - x_{min}}{x_{max} - x_{min}} \right) - 1$$

Donde se normaliza un valor x , de cada variable, a partir de los valores máximo y mínimo del conjunto de datos de dicha variable. Los valores máximos y mínimos se muestran en Tabla 10.

Tabla 10. Valores mínimos y máximos de las diferentes variables.

Variables	<i>Re</i>	<i>X</i>	<i>Y</i>	<i>t</i>	<i>u</i>	<i>v</i>	<i>P</i>
Mínimo	10	-0.7713	-0.6471	0.1	-7.2741	-7.5969	-54.2561
Máximo	290	2.9967	0.6471	5.0	12.7592	7.6328	33.3033

Fuente: Elaboración propia.

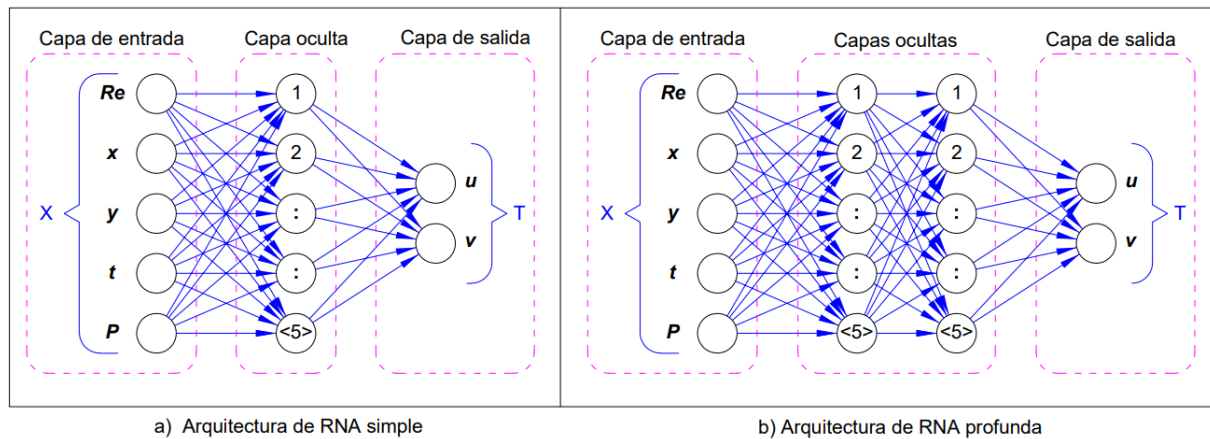
Una vez lograda la RNA deseada y generada la información de salida esta se “desnormalizará” para obtener los datos dentro de los rangos originales.

Respecto a la arquitectura de la RNA, esta se establece eligiendo únicamente el número de capas ocultas y su número de neuronas, puesto que el número de neuronas de entrada (X) y de salida (T) corresponden a las variables de interés predefinidas. Se parte con una configuración de RNA poco profunda o simple, pasando a configuraciones denominadas como profundas (Berzal, 2018).

Se proponen tres arquitecturas base, diferenciándose en el número de capas ocultas, de uno hasta tres. Las conexiones en este tipo de red avanzan en una sola dirección entre capas, donde todas las neuronas en cada capa están conectadas a todas las neuronas de la siguiente

capa (Hammelmann et al., 2016). En las capas ocultas se varía el número de neuronas de cinco en cinco, similar a lo desarrollado por (Islamoglu & Kurt, 2004). En la Figura 24 se detalla las arquitecturas de las RNAs propuestas.

Figura 24. Ejemplos de arquitecturas de RNAs propuestas: a) RNA de una capa oculta, b) RNA de dos capas ocultas. Ambas arquitecturas con número de neuronas múltiplos de 5.



Fuente: Elaboración propia.

Para lo antes expuesto, también se considera que las redes “feedforward” simples son capaces de aprender virtualmente cualquier relación entrada-salida, pero una RNA profunda puede aprender relaciones complejas más rápidamente. Para la mayoría de los problemas, es mejor comenzar con una RNA con una capa oculta y luego aumentar a dos capas ocultas, si el rendimiento con de la RNA simple no es satisfactorio (MathWorks, 2021).

Similar a los trabajos de (Dey & Das, 2016; Islamoglu & Kurt, 2004; Kurtulus, 2009; Pacheco-Vega et al., 2001; Samarasinghe, 2016; Sarghini et al., 2003; Taymaz & Islamoglu, 2009) las RNAs se entrenan con funciones de activación sigmoideas en las capas ocultas y función de activación lineal en la salida (Y. Wang et al., 2020), que están entre las más comunes en este tipo de aplicaciones (Zeng, 1998).

Debido a la naturaleza de los datos, donde las variables de salida oscilan entre valores positivos y negativos (ver Tabla 10) se utiliza la función de transferencia sigmoidea tangente hiperbólica, la cual se ajusta perfectamente a la normalización aplicada (Berzal, 2018; Kim et al., 2020; Zeng, 1998) .

El entrenamiento se realiza con el método de Levenberg Marquardt con parada anticipada para optimizar la red (Islamoglu & Kurt, 2004; Pacheco-Vega et al., 2001). Al tratarse de RNAs con pocas salidas (dos en todos los casos), el método de Levenberg Marquardt, según

(Ledesma, 2006; MathWorks, 2021), es frecuentemente superior que otros métodos de entrenamiento a costa de gasto computacional.

Entre los tipos de RNAs existentes, la de alimentación directa se considera de las más comunes en aplicaciones de ingeniería (Pacheco-Vega et al., 2001; Zeng, 1998) siendo el tipo de RNA que se utiliza en este estudio.

Al tratarse de un entrenamiento supervisado, se usa el método de cálculo del gradiente de retropropagación (del inglés backpropagation), lo que a su vez convierte en especialmente útiles a las funciones de activación sigmoidales (Berzal, 2018). Las RNAs con retropropagación plantean un esquema de aprendizaje matemáticamente estricto para entrenar la red, los pesos y sesgos son modificados para minimizar el error y garantizar el mapeo entre entradas y salidas (Islamoglu & Kurt, 2004; Pacheco-Vega et al., 2001).

Como paso previo a la configuración y entrenamiento de las RNAs, en el entorno de MATLAB®, se conforman los datos, dividiéndolos en dos matrices; la matriz de entrada "X" (5x33705000) y la matriz de salida "T" (2x33705000). Ver Figura 25.

Figura 25. Matrices de entrada y salida, previo al entrenamiento de las RNAs.



Name	Value
T	2x33705000 double
X	5x33705000 double

Fuente: Elaboración propia.

El siguiente paso es configurar y entrenar cada una de las RNAs. Estableciéndose las siguientes condiciones para el entrenamiento y parada del proceso:

- ✓ Los datos se dividen aleatoria y automáticamente, en tres conjuntos: entrenamiento (60%), validación (20%) y prueba (20%). El conjunto de entrenamiento es usado para enseñar a la RNA, dicho entrenamiento prosigue hasta que la RNA continúe mejorando en el conjunto de validación. El conjunto de prueba se utiliza para obtener una medida de precisión de la red, pero no tiene ningún efecto en el entrenamiento.
- ✓ El número máximo de épocas (ciclos de corrección de propagación hacia atrás y hacia adelante para reducir la pérdida) para entrenar es: 10000.
- ✓ El tiempo máximo de entrenamiento es indefinido, puesto que existen otros criterios de parada, por ejemplo, el número de épocas antes detallado.

- ✓ El objetivo de rendimiento “performance” es 0. El cual se mide en términos de error cuadrático medio (MSE, por sus siglas en inglés) y se muestra en escala logarítmica.
- ✓ La cantidad de fallos máximos de validación es 6. Este valor, utiliza el conjunto de validación para detener tempranamente el entrenamiento si el rendimiento de la RNA no mejora o sigue siendo el mismo para épocas consecutivas. Esto significa que el entrenamiento para después de 6 aumentos consecutivos en el error de validación y el mejor rendimiento se toma de la época con el error de validación más bajo.
- ✓ El gradiente mínimo de rendimiento es $1e-7$ (valor por defecto).
- ✓ El valor que afecta la magnitud del cambio de los pesos sinápticos es μ (mu), teniendo un valor inicial por defecto de 0.001, el valor adaptativo mu aumenta en un factor de 10 hasta que el cambio anterior da como resultado un valor de rendimiento reducido. Seguidamente, se realiza el cambio en la red y mu se reduce en un factor de 0.1. El valor máximo para mu es $1e10$ (MathWorks, 2021).
- ✓ Los pesos y los sesgos se configuran por defecto, de manera automática por el aplicativo; estando lista la red para el entrenamiento.

En resumen, el entrenamiento se detiene cuando ocurre alguna de estas condiciones:

- Se alcanza el número máximo de épocas.
- El rendimiento se reduce al mínimo objetivo.
- El gradiente de rendimiento cae por debajo del mínimo.
- mu excede el máximo.
- El rendimiento de la validación ha aumentado más que 6 veces desde la última vez que disminuyó (al usar la validación).

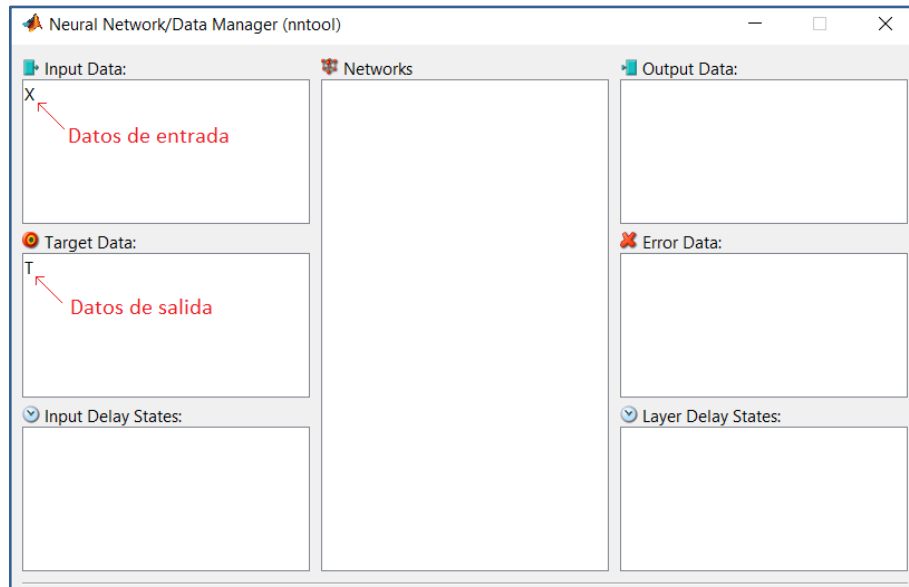
Con lo descrito, se ejecutan las RNAs y son evaluadas a priori con base en la regresión, específicamente en el valor de R (entre datos reales y predichos). Las configuraciones con los mejores valores de correlación se seleccionan para evaluar resultados.

Ejemplificando lo expuesto en este apartado, se detalla el proceso de creación y entrenamiento para una RNA profunda, de dos capas ocultas; con 10 neuronas en cada capa:

1. Se importa los datos para las capas de entrada y de salida. Ver Figura 26.

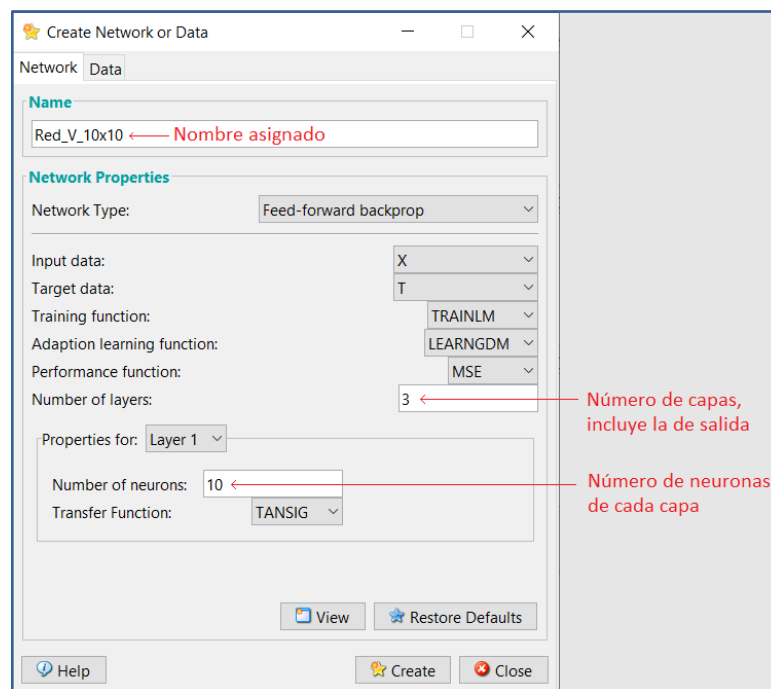
- Se crea la RNA, seleccionando la tipología, arquitectura y parámetros adicionales, ya descritos. Ver Figura 27.

Figura 26. Importación de datos para las capas de entrada y salida. Vista en el entorno de aplicación de RNAs.



Fuente: Elaboración propia.

Figura 27. Configuración de métodos y parámetros para la RNA. Vista en el entorno de aplicación de RNA.

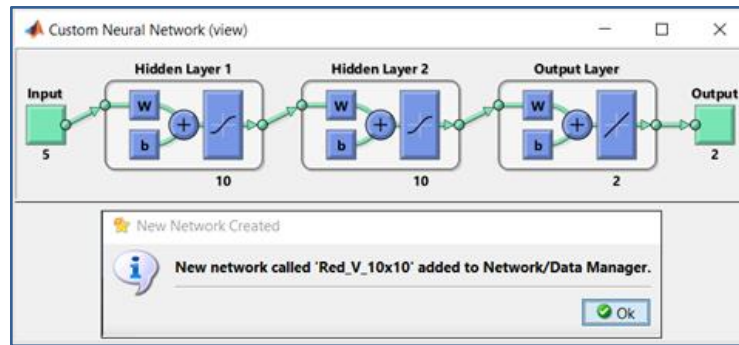


Fuente: Elaboración propia.

- Se crea la red neuronal y se visualiza su arquitectura. Ver Figura 28.

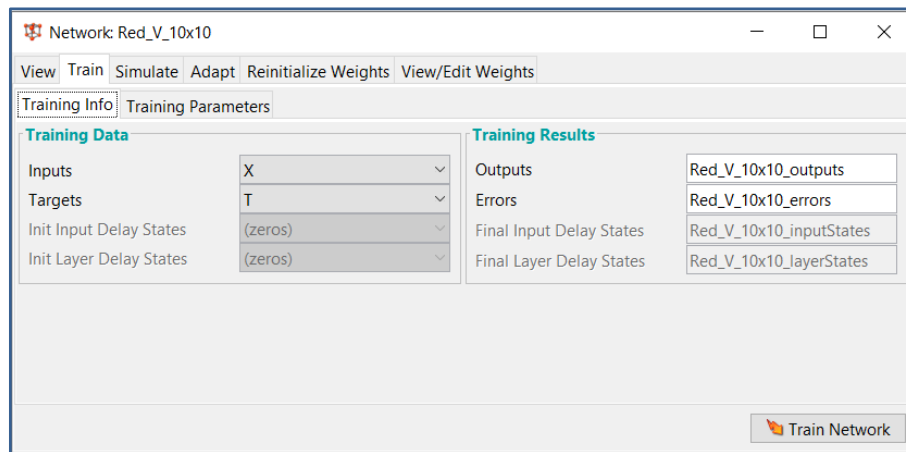
- Se asigna los datos de entrada y de salida, importados en el paso 1. Ver Figura 29.

Figura 28. Arquitectura de la RNA creada. Vista en el entorno de aplicación de RNA.



Fuente: Elaboración propia.

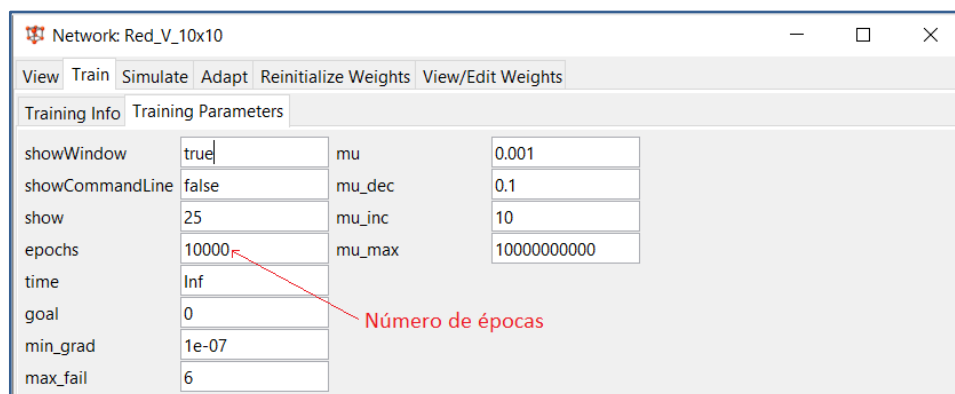
Figura 29. Asignación de datos de entrada y de salida para el entrenamiento de la RNA. Vista en el entorno de aplicación de RNA.



Fuente: Elaboración propia.

- Se configura los parámetros de entrenamiento. Ver Figura 30.

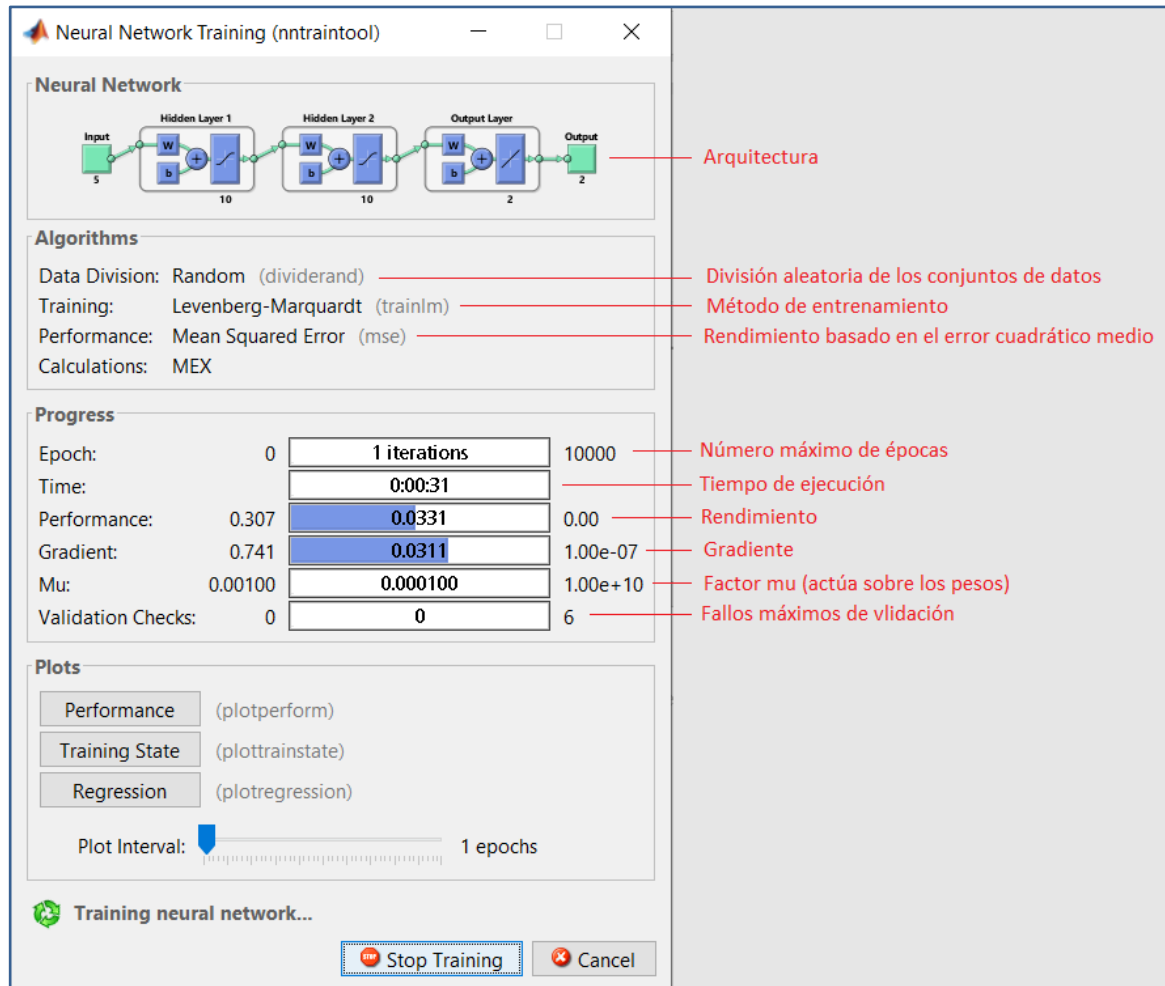
Figura 30. Asignación de parámetros de entrenamiento para la RNA. Vista en el entorno de aplicación de RNA.



Fuente: Elaboración propia.

6. Se ejecuta el entrenamiento y se controla su desarrollo hasta que se cumpla alguno de los diversos criterios de parada. Ver Figura 31.

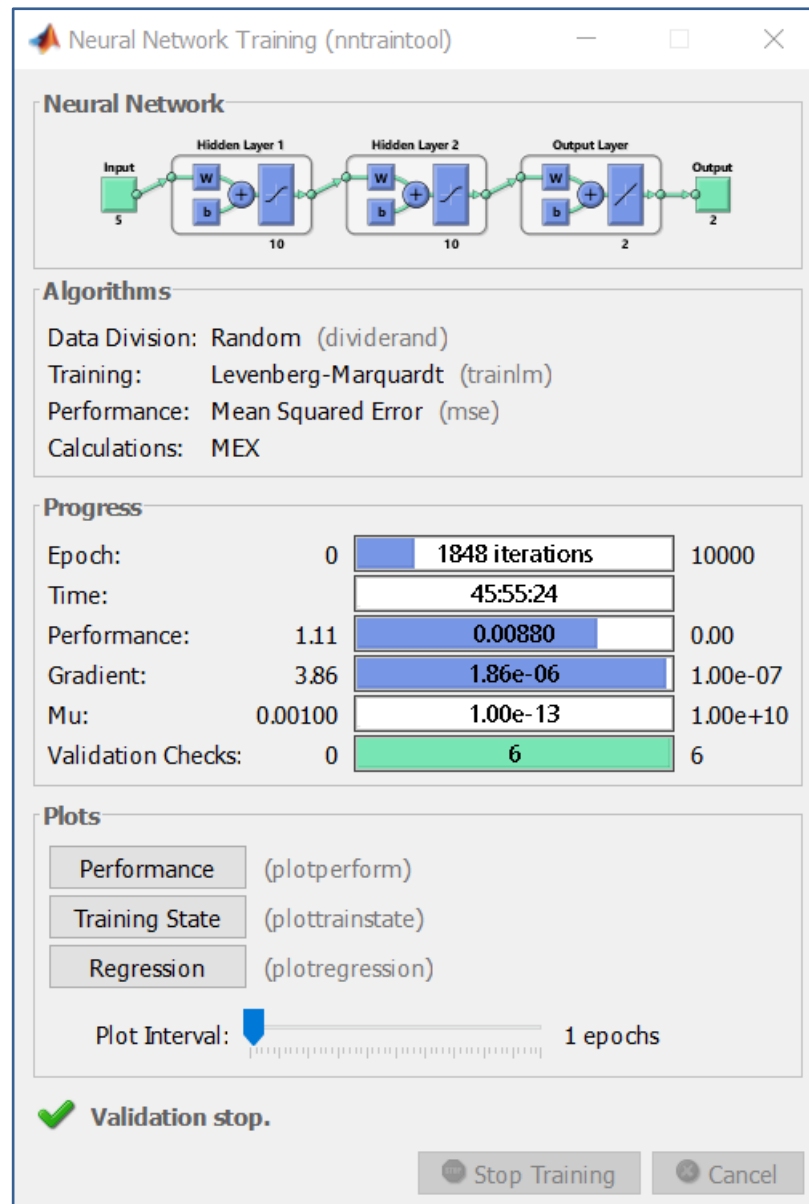
Figura 31. Detalle de configuración y de progreso de los diversos parámetros de la RNA durante la ejecución del entrenamiento. Vista en el entorno de aplicación de RNA.



Fuente: Elaboración propia.

En este caso la RNA culmina su entrenamiento a las 1848 épocas, lo que implica un tiempo aproximado de 46 h, con un rendimiento de 0.0088 y una gradiente de $1.86e-6$. Siendo el número de validaciones de aumento del rendimiento el criterio de parada cumplido. La Figura 32 muestra el detalle de finalización del entrenamiento.

Figura 32. Detalle de fin de entrenamiento. Vista en el entorno de aplicación de RNA.

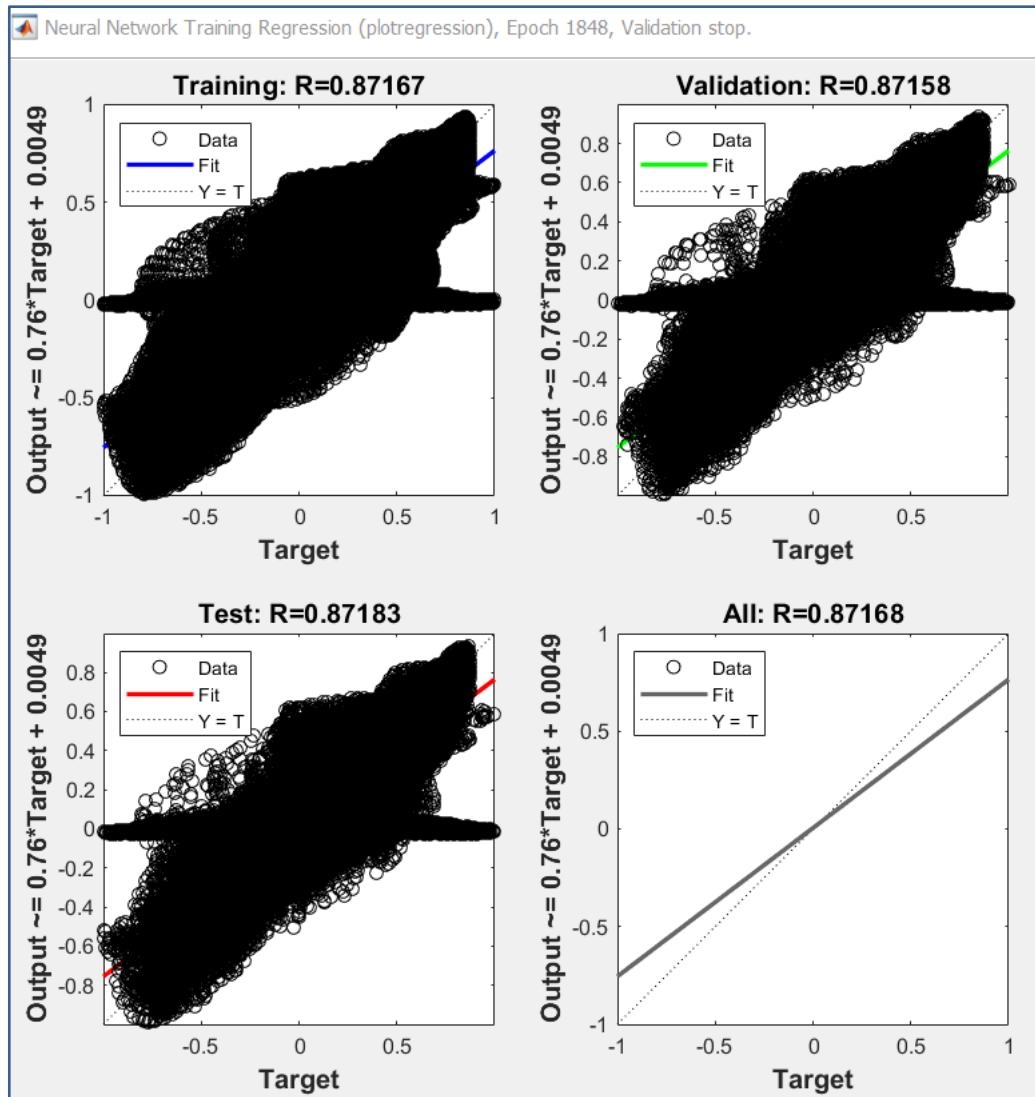


Fuente: Elaboración propia.

7. Se genera la gráfica de regresión y se analiza los resultados. En la Figura 33 se observa que todas las correlaciones son positivas, imperfectas, superiores y cercanas a 0.87. Las cuatro correlaciones mostradas en el reporte (de izquierda a derecha y de arriba a abajo) son: del conjunto de datos de entrenamiento, del conjunto de validación, del conjunto de prueba y del total de los datos.

Con base a los valores de correlación alcanzados, se puede afirmar que la RNA lograda no posee un sobre ajuste y que logra generalizar y predecir los datos de velocidades con la misma capacidad lograda en la fase entrenamiento y validación.

Figura 33. Reporte gráfico de correlaciones resultantes de la RNA ejemplificada. Vista en el entorno de aplicación de RNA.



Fuente: Elaboración propia.

8. Finalmente, se importa la RNA entrenada, en un archivo “.mat” para su futura ejecución. Considerando lo recomendado por (Caicedo B & López S, 2009) se empieza a probar RNAs con un número pequeño de neuronas en la capa oculta, cinco en este caso. También se evita utilizar más de tres capas ocultas ya que (Caicedo B & López S, 2009) , también menciona que, para una gran cantidad de aplicaciones complejas, tres capas son suficientes, pero el hecho de usar más de una capa oculta, aumenta drásticamente la carga computacional de la red. Se ejecutan un total de 12 RNAs cuyos resultados se exponen en la Tabla 11. Debido al considerable gasto computacional y el tiempo que este conlleva, no se probó RNAs con tres capas ocultas.

Tabla 11. Resultados de las RNAs entrenadas.

Arquitectura de capas ocultas		Épocas	Tiempo	Suceso de parada	Rendimiento	Correlación
Capas	Neuronas/Capa		(h:min)		(MSE)	(R)
1	5	408	1:55	Validación	0.0147	0.774
	10	10000	89:18	Épocas	0.0115	0.828
	15	686	8:36	Validación	0.0110	0.836
	20	954	15:24	Validación	0.0109	0.838
	25	5015	105:17	Validación	0.0102	0.850
	30	945	26:26	Validación	0.0096	0.859
2	5	584	5:10	Validación	0.0118	0.823
	10	1.848	45:55	Validación	0.0088	0.872
	15	3926	181:06	Validación	0.0082	0.881
	20	2120	219:48	Validación	0.0080	0.885
	25	1933	336:35	Validación	0.0077	0.888
	30	2117	658:39	Validación	0.0075	0.892

Fuente: Elaboración propia.

Los entrenamientos se desarrollaron con una computadora Intel Core™ i7-10750H CPU @ 2.60 GHz, 16 GB RAM, de ellos se puede observar que las redes con arquitecturas de dos capas ocultas entregan mejores ajustes lineales y menores errores. También se observa que conforme aumenta el número de neuronas en las capas ocultas mejora el desempeño de las RNAs, pero a costa de un notablemente mayor gasto computacional. La Tabla 12 muestra el tiempo utilizado en cada época de entrenamiento para las arquitecturas probadas.

Tabla 12. Tiempo de cada época de entrenamiento para las diversas RNAs probadas.

Tiempo de cada época de entrenamiento (en segundos).							
	Neuronas/Capa	5	10	15	20	25	30
Capas	1	17	32	45	58	76	101
	2	32	89	166	373	627	1120

Fuente: Elaboración propia.

En función de los resultados mostrados, se decide seleccionar las RNAs de dos capas ocultas, con 25 y 30 neuronas en cada capa para la evaluación de predicción de datos, de donde se obtendrá la mejor versión de RNA del presente estudio. Resultados más detallados de los entrenamientos son mostrados en el Anexo 2 y Anexo 3.

Evaluación de las arquitecturas de RNAs seleccionadas.

En este apartado se procede a evaluar las dos mejores RNAs obtenidas, declaradas en el apartado anterior y resaltadas en la Tabla 11. Previo al proceso de evaluación, se expone los criterios del por qué se seleccionaron dichas RNAs:

- ✓ Son las versiones que presentan los mejores valores de rendimiento (Ver Tabla 11), en base al error cuadrático medio (MSE).
- ✓ Son las que brindan los mejores valores de correlación (R) resultantes de la regresión lineal entre la respuesta de la RNA y los objetivos correspondientes (Ver Tabla 11).
- ✓ No existen diferencias notables de los valores obtenidos de R y MSE entre estas arquitecturas (Ver Tabla 11).
- ✓ La relación de gasto computacional entre estas dos arquitecturas es aproximadamente de 1.8. En la Tabla 12, se observa que la arquitectura de dos capas ocultas con 25 neuronas en cada capa (25x25), requiere aproximadamente 627 segundos en cada época, frente a los 1120 segundos empleados por la arquitectura de dos capas ocultas con 30 neuronas en cada capa (30x30). Con lo anterior, existe la posibilidad de que, en la evaluación de resultados, no haya diferencias notorias entre las arquitecturas, siendo así, la RNA de menor gasto computacional se presentaría como resultado final del presente estudio.

Se compara la precisión de predicción de las RNAs mediante el cálculo de errores medios, absoluto y relativo, de velocidad de flujo (para cada celda en todos los pasos temporales), entre simulaciones CFD con Re que no participaron en el entrenamiento y sus similares obtenidos desde las RNAs “simulaciones sintéticas”.

Cabe recalcar que se evalúan los campos de velocidad compuestos por los valores de velocidad horizontal y vertical, u y v respectivamente.

En el cálculo de errores de cada celda del dominio computacional, se usan relaciones similares a las expuestas por (Jin et al., 2018). El error absoluto Ea_i (Ea_u o Ea_v) y el error relativo Er_i (Er_u o Er_v) se definen mediante las ecuaciones siguientes:

$$Ea_i(x) = \sqrt{\frac{\int_T (u_i^*(x, t) - \hat{u}_i^*(x, t))^2 dt}{T}}$$

$$Er_i(x) = \frac{\frac{1}{T} \cdot \int_T |u_i^*(x, t) - \hat{u}_i^*(x, t)| dt}{\max_{t \in [0, T]} |u_i^*(x, t)|}$$

Donde u_i^* (u^* o v^*) son las componentes de velocidad reportadas por las simulaciones CFD para cada celda (x) en cada paso temporal (t), mientras que \hat{u}_i^* (\hat{u}^* o \hat{v}^*) representan los valores de velocidad predichos por la RNA, igualmente, para cada celda y paso temporal. T es el total de pasos temporales de la simulación o predicción en análisis.

En el cálculo del error relativo se excluyen los puntos cercanos a la pared del cilindro por poseer valores de velocidad muy cercanos o iguales a cero.

Las simulaciones evaluadas son para números de Reynolds de 5, 60, 100, 140, 200, 260, 300; dentro de los primeros 50 pasos temporales (0, 0.1, 0.2, ...4.9, 5 s). Los casos con Re de 5 y 300 evalúan la capacidad de extrapolación de las RNAs, mientras que el resto de casos son para evaluar la capacidad de interpolación. La Tabla 13 muestra los valores máximos, mínimos y medios de los errores absolutos y relativos obtenidos.

De los valores reportados se puede observar y deducir lo siguiente:

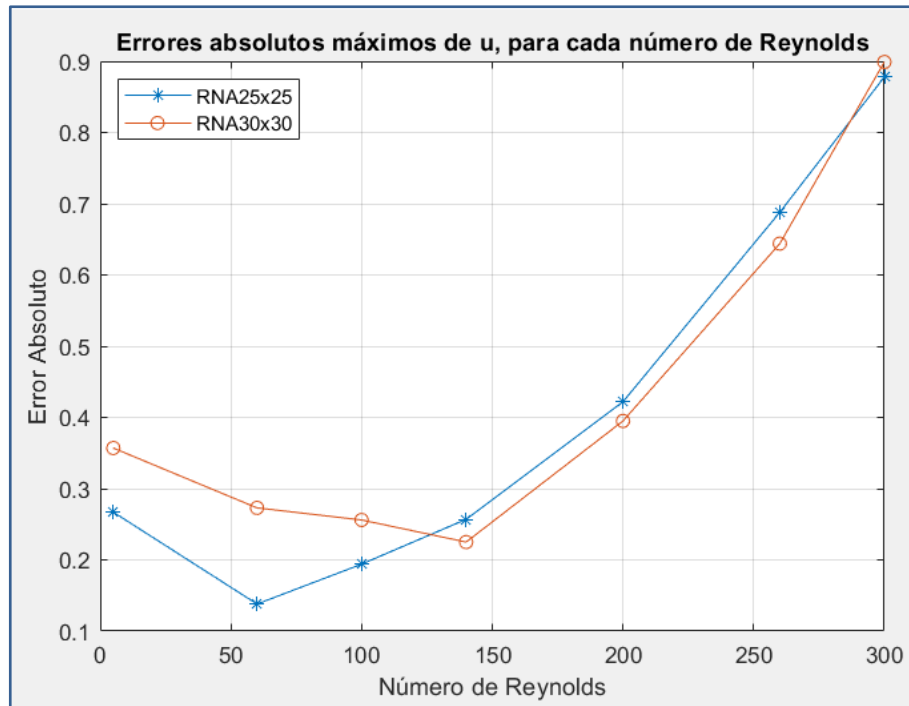
- ✓ La arquitectura denominada “25x25” presenta menores errores absolutos máximos de predicción de valores de velocidad horizontal (u) para números de Reynolds de 5, 60, 100 y 300. Para números de Reynolds de 140, 200 y 260; la arquitectura denominada “30x30” presenta menores valores de error. Ver Figura 34.
- ✓ La arquitectura “25x25” presenta menores errores relativos máximos en la predicción de velocidad u para números de Reynolds de 5 y 300. Para Reynolds de 60, 100, 140, 200 y 260; la arquitectura “30x30” presenta menores errores. Ver Figura 35.
- ✓ La arquitectura “30x30” presenta menores errores absolutos máximos en la predicción de velocidad vertical (v) para casi todos los números de Reynolds, a excepción de $Re = 100$, pero con una diferencia ínfima. Ver Figura 36 y Tabla 13.
- ✓ La arquitectura “30x30” presenta menores errores relativos máximos en la predicción de velocidad vertical (v) para casi todos los números de Reynolds, a excepción de $Re = 5$. En este caso la diferencia es notable. Ver Figura 37.

Tabla 13. Comparación de errores máximos, mínimos y medios; absolutos y relativos, de las RNA en validación.

RNA	Re	Errores											
		E _u			E _r			E _v			E _r		
		Max	Min	Med	Max	Min	Med	Max	Min	Med	Max	Min	Med
25x25	5	0.267	0.003	0.027	0.652	0.004	0.058	0.393	0.0004	0.015	5347	0	40.7
	60	0.138	0.001	0.020	0.190	0.0002	0.004	0.151	0.0006	0.012	26.71	0.003	0.102
	100	0.194	0.002	0.044	0.083	0.0004	0.004	0.169	0.001	0.062	4.197	0.001	0.028
	140	0.257	0.004	0.054	0.088	0.0003	0.004	0.507	0.002	0.147	2.103	0.003	0.037
	200	0.422	0.007	0.075	0.066	0.0003	0.003	1.033	0.002	0.249	1.2713	0.003	0.035
	260	0.688	0.007	0.107	0.058	0.0003	0.003	1.458	0.003	0.340	1.981	0.003	0.034
	300	0.878	0.007	0.150	0.058	0.0002	0.004	1.821	0.005	0.397	1.983	0.003	0.035
30x30	5	0.357	0.003	0.024	1.216	0.005	0.048	0.171	0.001	0.016	27005	0	88
	60	0.273	0.003	0.021	0.147	0.0004	0.004	0.139	0.0004	0.012	15.509	0.003	0.113
	100	0.256	0.004	0.045	0.047	0.0003	0.005	0.170	0.0007	0.061	1.911	0.001	0.027
	140	0.225	0.006	0.054	0.051	0.0003	0.004	0.504	0.002	0.146	0.883	0.002	0.037
	200	0.395	0.007	0.076	0.056	0.0002	0.003	1.019	0.001	0.247	0.425	0.002	0.034
	260	0.644	0.007	0.104	0.053	0.0003	0.003	1.445	0.001	0.328	0.303	0.002	0.032
	300	0.899	0.007	0.147	0.059	0.0003	0.004	1.798	0.002	0.393	0.281	0.003	0.033

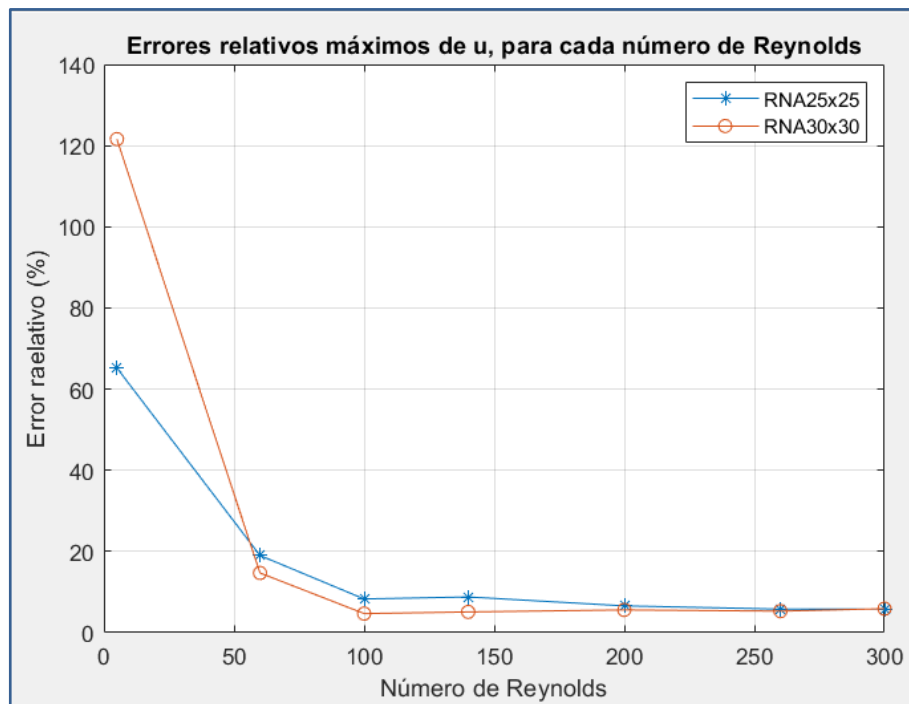
Fuente: Elaboración propia.

Figura 34. Errores absolutos máximos de u , al aplicar las RNAs en evaluación.

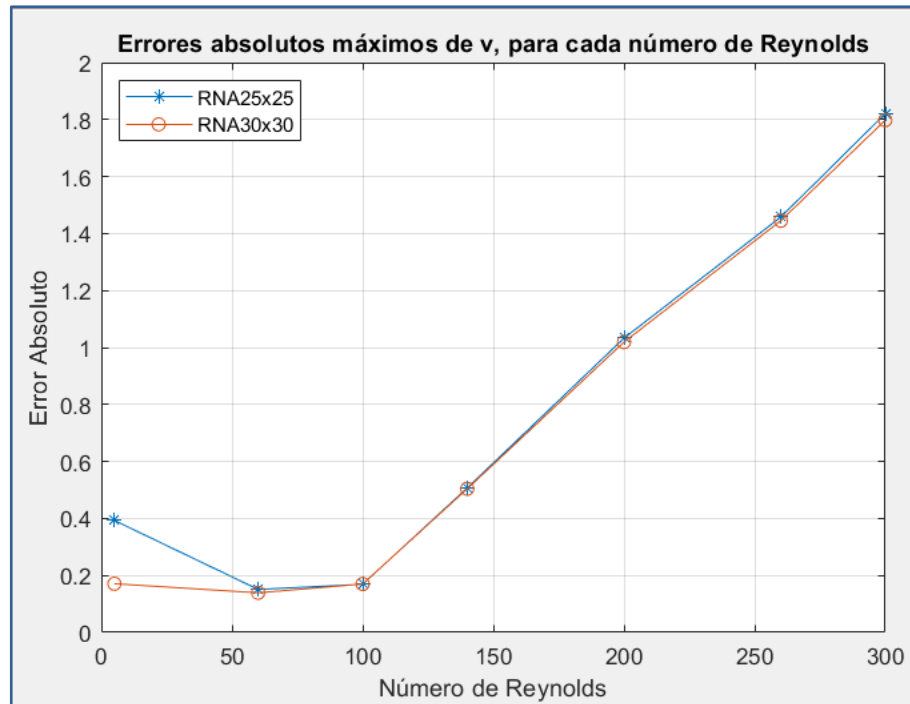


Fuente: Elaboración propia.

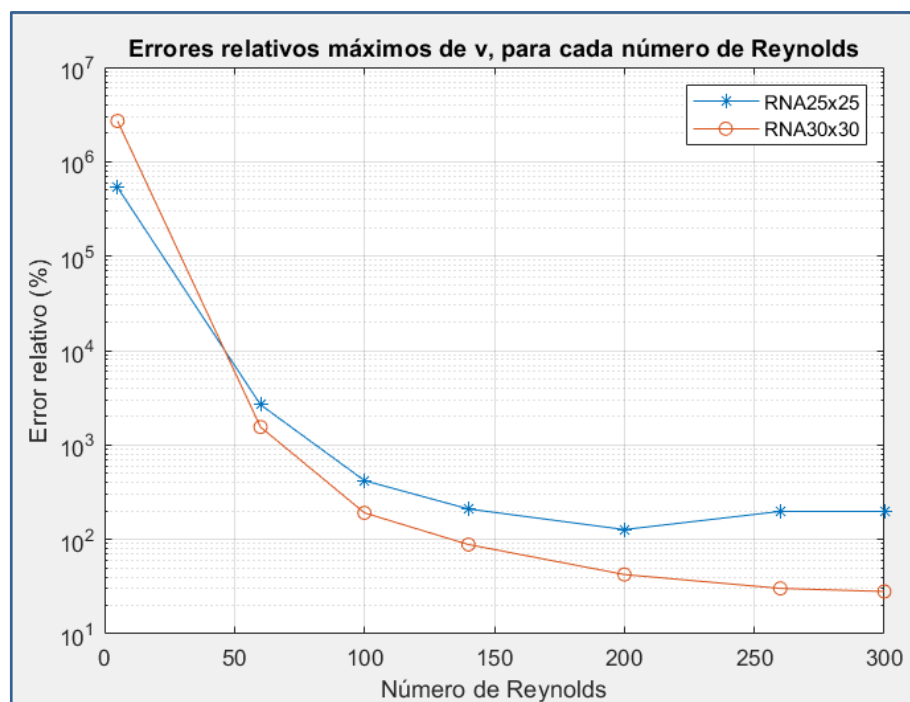
Figura 35. Errores relativos máximos de u , al aplicar las RNAs en evaluación.



Fuente: Elaboración propia.

Figura 36. Errores absolutos máximos de v , al aplicar las RNAs en evaluación.

Fuente: Elaboración propia.

Figura 37. Errores relativos máximos de v , al aplicar las RNAs en evaluación.

Fuente: Elaboración propia.

De los datos y gráficas de la componente horizontal de velocidad, se puede concluir que para Re iguales a 60, 100, 140, 200 y 260 la RNA con arquitectura de dos capas ocultas con 25 neuronas en cada capa "25x25", predice con un nivel de precisión muy próximo (diferencia

máxima de 4.3%) a su similar de dos capas ocultas con 30 neuronas en cada capa, incluso presenta mejor capacidad de extrapolación, puesto que para números de Reynolds de 5 y 300 se obtienen menores errores relativos máximos (con diferencia máxima de 56.4%).

En cambio, de lo expuesto sobre la componente vertical de velocidad, se concluye que para Re iguales a 60, 100, 140, 200, 260 y 300 la RNA con arquitectura de dos capas ocultas con 30 neuronas en cada capa "30x30", predice datos notablemente mejor que su similar de dos capas ocultas con 25 neuronas en cada capa (diferencia mínima de 170.2% y máxima de 1120%), a excepción de $Re = 5$ donde obtienen errores relativos máximos muy elevados.

Los errores relativos mínimos y medios, para ambas componentes de velocidad, reportados en la Tabla 13, son muy similares en ambas arquitecturas de RNAs y poseen valores aceptables entre el 3.3% y 11.3%.

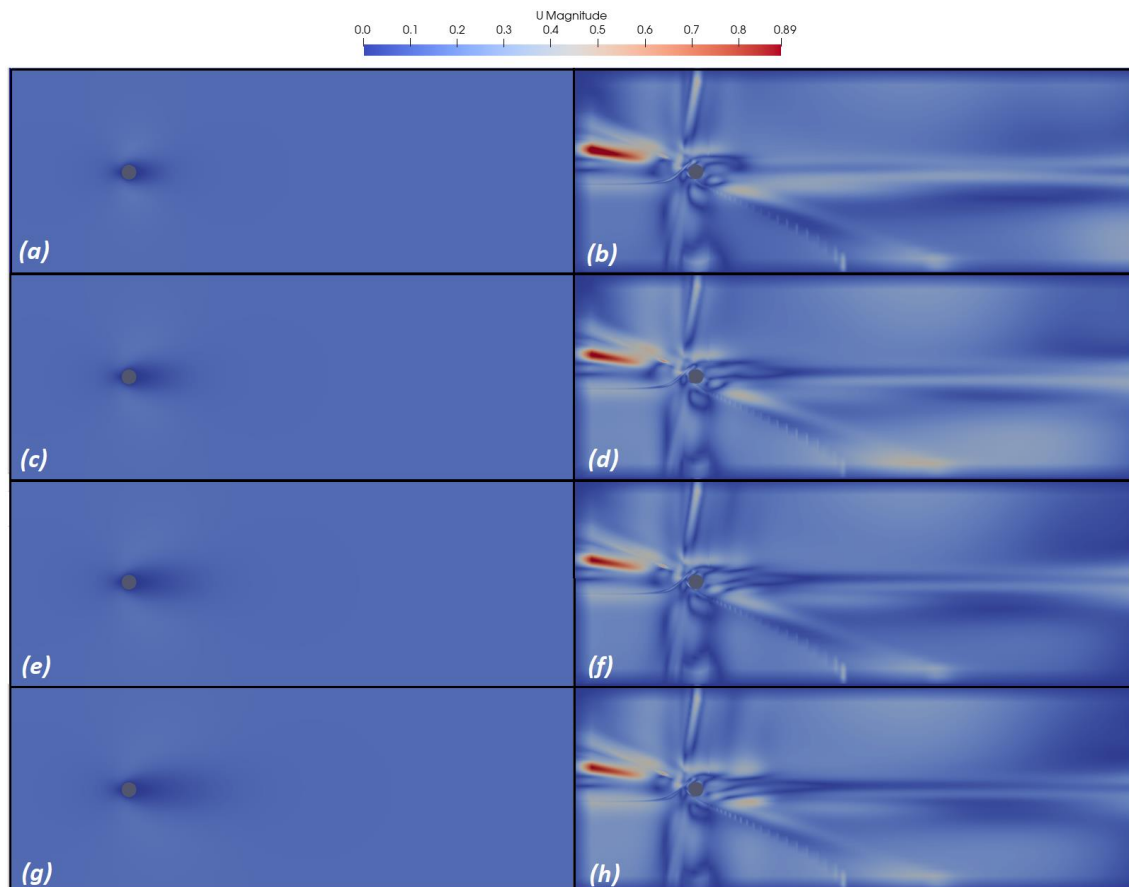
En base a las conclusiones anteriores, se selecciona como resultado final del presente estudio, a la arquitectura de dos capas ocultas con 30 neuronas en cada capa "30x30", lo anterior, por presentar menores errores relativos máximos en los resultados de predicción en ambas componentes de velocidad. Sin embargo, es importante denotar que su capacidad de extrapolación para bajos números de Reynolds es deficiente, por exhibir errores relativos elevados para el caso de $Re = 5$. Ver Tabla 13.

Resultados de la RNA con mejores prestaciones.

Con la mejor RNA se evalúan los campos de velocidad. Se exponen cuatro campos instantáneos de magnitud de velocidad, pertenecientes a cada una de las simulaciones CFD y sus homólogas "Sintéticas" para Re de 5, 60, 100, 140, 200, 260 y 300, en los tiempos $t = 1.2, 2.4, 3.6$ y 5 s. Posteriormente, se calculan los errores absolutos y relativos promedios en cada celda del dominio computacional y se indican en escala de colores sobre dicho dominio.

La Figura 38 detalla los campos de velocidad para el caso de flujo con un número de $Re = 5$, se observa que existen diferencias de valores en las regiones que rodean en cilindro y en especial en la entrada de flujo, coincidiendo con los errores expuestos en la Tabla 13. En cuanto a la estela, la simulación sintética no realiza lanzamiento de vórtices, concordando con los resultados CFD, sin embargo, existen diferencias de velocidad.

Figura 38. Campos instantáneos de velocidad para $Re = 5$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados homólogos por la RNA.

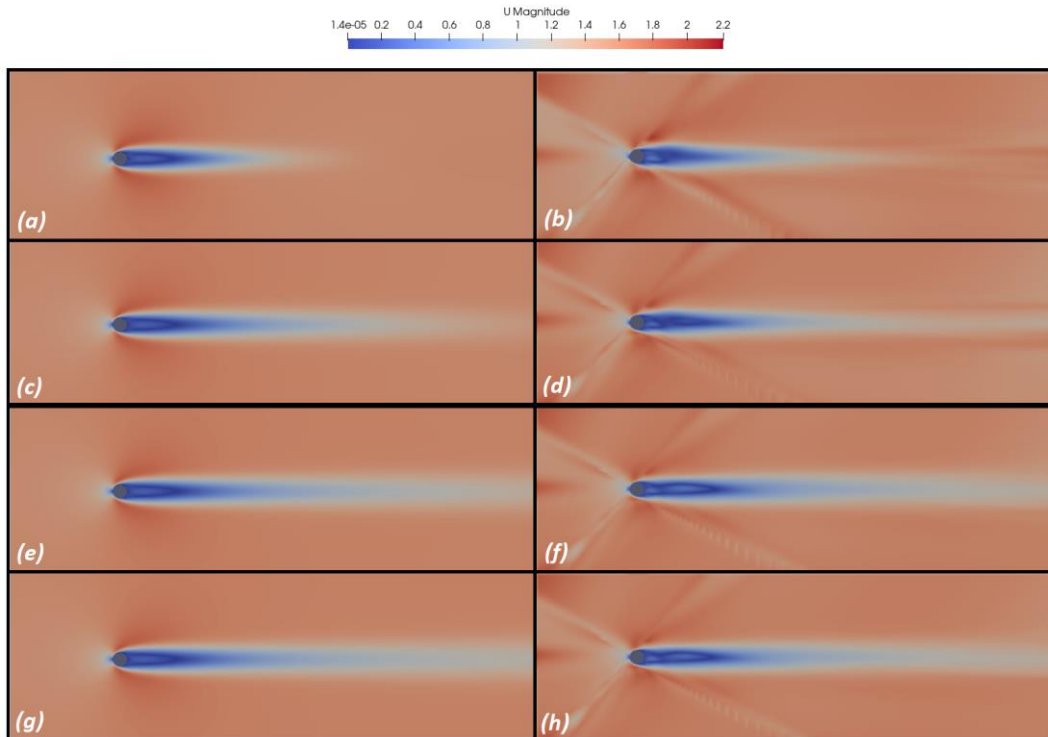


Fuente: Elaboración propia.

La Figura 39 muestra los campos de velocidad para el flujo con $Re = 60$, se aprecia que existen escasas diferencias de valores en las regiones que rodean en cilindro y a la entrada de flujo. La simulación sintética no realiza lanzamiento de vórtices en la estela, la longitud de estela es altamente concordante y su comportamiento temporal es correcto, este aspecto posee elevada similitud con los resultados CFD.

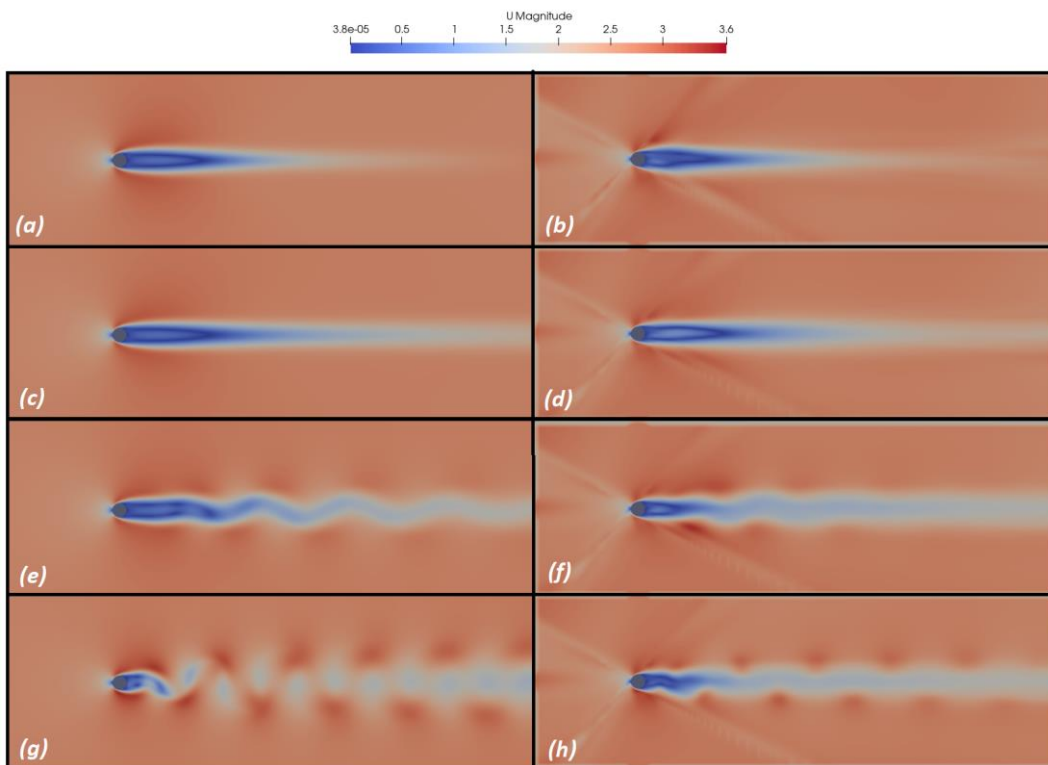
La Figura 40 expone los campos de velocidad para el flujo con $Re = 100$, al igual que en los casos anteriores se observan algunas diferencias de valores en las regiones que rodean en cilindro y a la entrada de flujo. La simulación sintética realiza lanzamiento de vórtices en la estela en los mismos instantes que su homóloga CFD. La estela presenta algunas diferencias, especialmente en el tiempo 3.6 s, figura (f), donde la zona de lanzamiento de vórtices, del resultado predicho, se reduce. La frecuencia de salida de vórtices es similar, aunque existen diferencias de continuidad y amplitud.

Figura 39. Campos instantáneos de velocidad para $Re = 60$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados homólogos por la RNA.



Fuente: Elaboración propia.

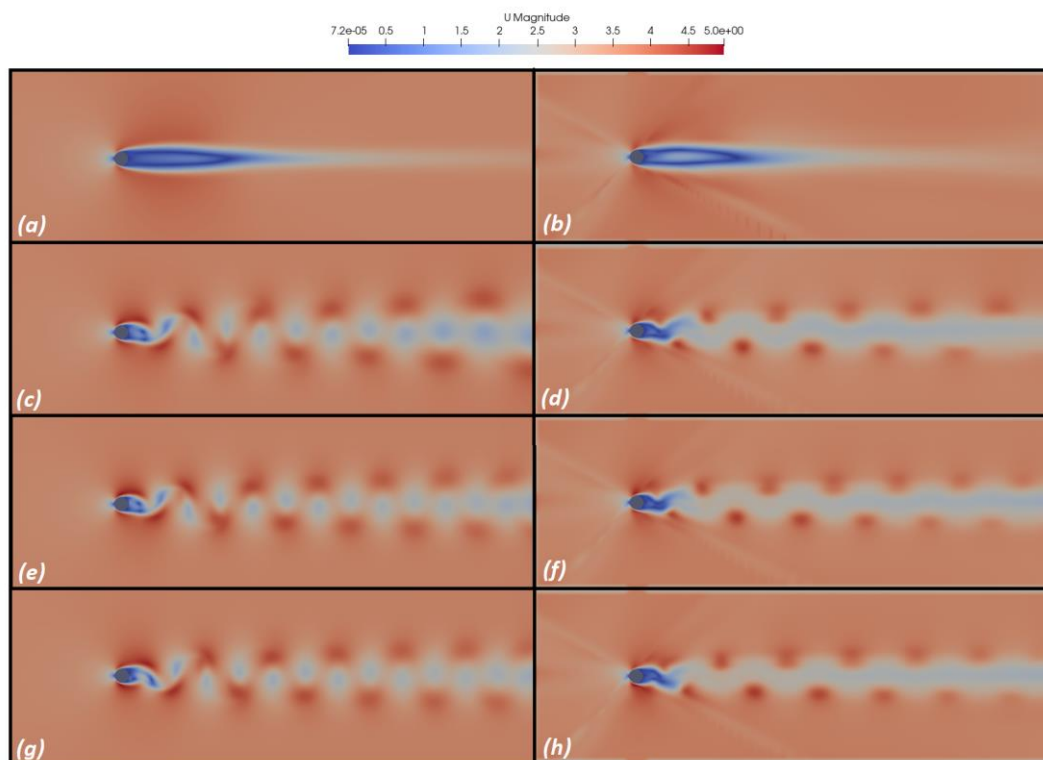
Figura 40. Campos instantáneos de velocidad para $Re = 100$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados homólogos por la RNA.



Fuente: Elaboración propia.

La Figura 41 detalla resultados para el flujo con $Re = 140$, donde persisten diferencias de valores en las regiones alrededor del cilindro y a la entrada de flujo. La simulación sintética realiza lanzamiento de vórtices en sincronismo con su homóloga CFD. La estela es semejante en los estados iniciales, con diferencias de amplitud y de continuidad para los estados de tiempo finales.

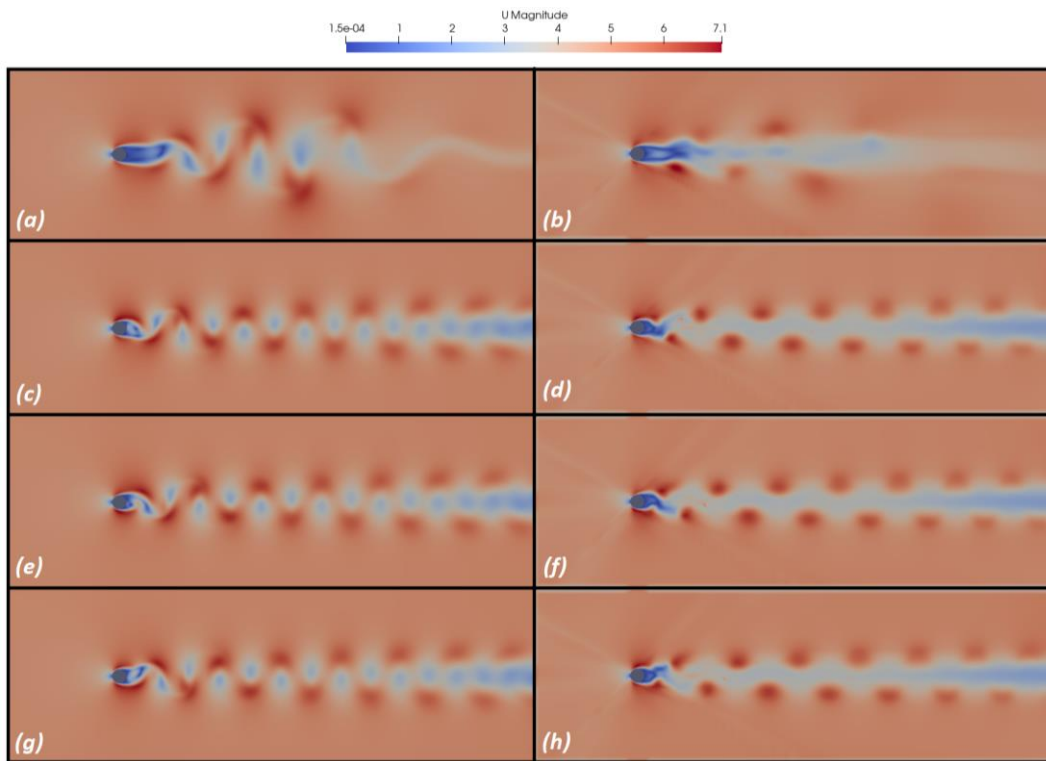
Figura 41. Campos instantáneos de velocidad para $Re = 140$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados predichos de por RNA.



Fuente: Elaboración propia.

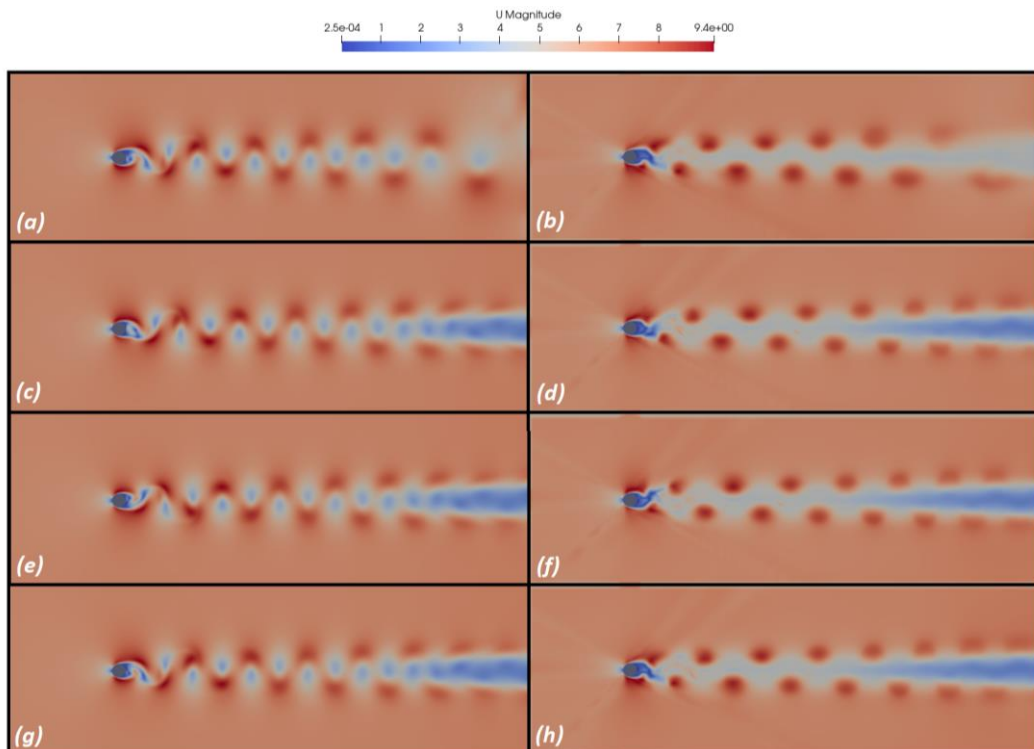
Los resultados para el flujo con Re de: 200 (Figura 42), 260 (Figura 43) y 300 (Figura 44) son muy similares en comportamiento. En estos casos persisten diferencias de valores en las regiones que rodean al cilindro y en la entrada de flujo, pero menos notorias. Las simulaciones sintéticas realizan el lanzamiento de vórtices en los mismos tiempos que sus homólogas CFD. Las regiones de lanzamiento de vórtices poseen buena similitud, pero aún se presentan diferencias de amplitud, continuidad y de valores. Es importante notar que, en la parte final del dominio, la estela presenta buena semejanza de valores. En las simulaciones sintéticas, también es notoria la disminución de la longitud de formación de vórtices, que según (Tamura et al., 1990), se presenta con el aumento de Re para el caso de simulación 2D de flujo sobre un cilindro. Concordando con el comportamiento real del fenómeno en estudio.

Figura 42. Campos instantáneos de velocidad para $Re = 200$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados predichos por la RNA.



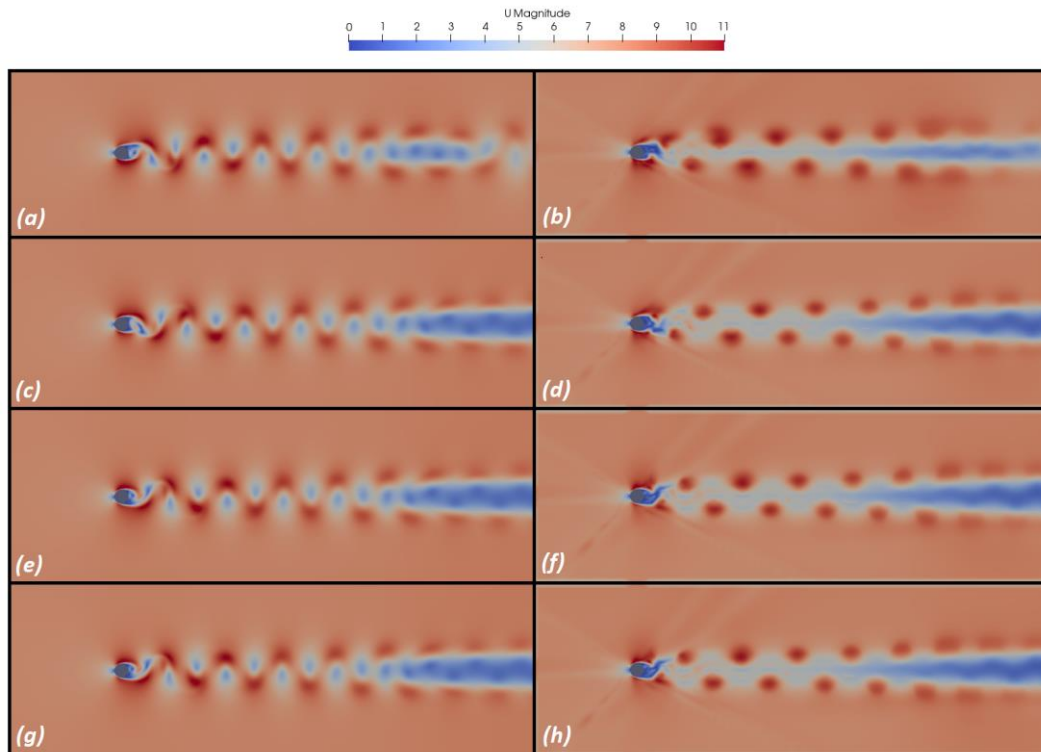
Fuente: Elaboración propia.

Figura 43. Campos instantáneos de velocidad para $Re = 260$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados predichos por la RNA.



Fuente: Elaboración propia.

Figura 44. Campos instantáneos de velocidad para $Re = 300$. En (a), (c), (e) y (g) resultados CFD de $t = 1.2, 2.4, 3.6$ y 5 s, respectivamente. En (b), (d), (f) y (h) resultados predichos por la RNA.



Fuente: Elaboración propia.

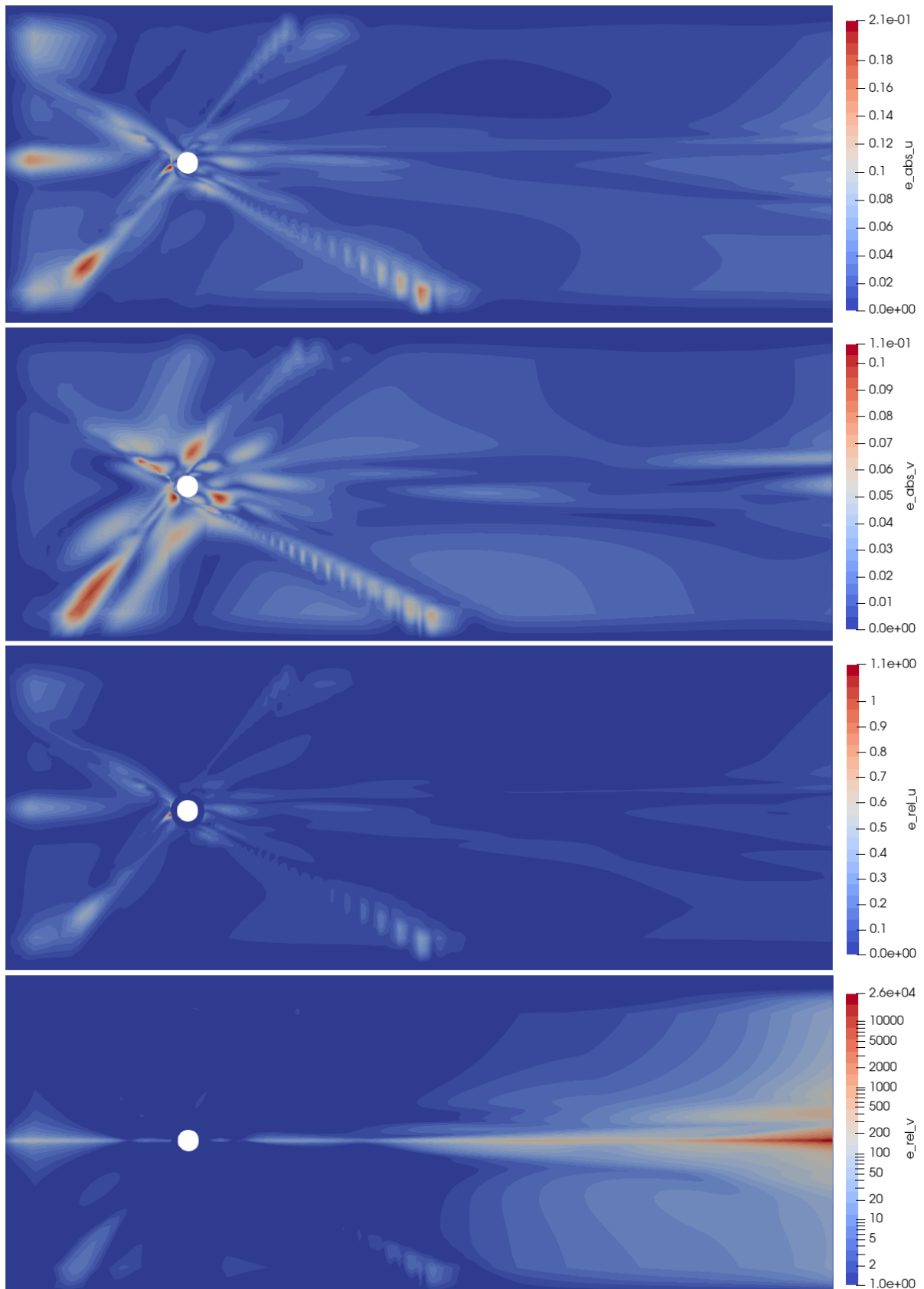
Para identificar los errores resultantes de la predicción de velocidad, se presenta para cada celda del dominio computacional los errores promedios temporales. Se generan cuatro campos de errores para cada caso de simulación; dos de errores absolutos y dos de errores relativos, correspondientes a las componentes de velocidad.

La Figura 45 detalla los errores de predicción para $Re = 5$, los errores relativos son elevados para la velocidad vertical y menos pronunciados para la velocidad horizontal. Corroborando los estados temporales de la Figura 38. Resultando ser una extrapolación errónea.

La Figura 46 muestra los errores promedios para $Re = 60$, con valores notables distribuidos alrededor del cilindro. Los errores relativos de la velocidad horizontal son menores al 6% en casi todo el dominio computacional, a excepción de un pequeño sector de la región inferior de lanzamiento de estela (error de hasta 120%).

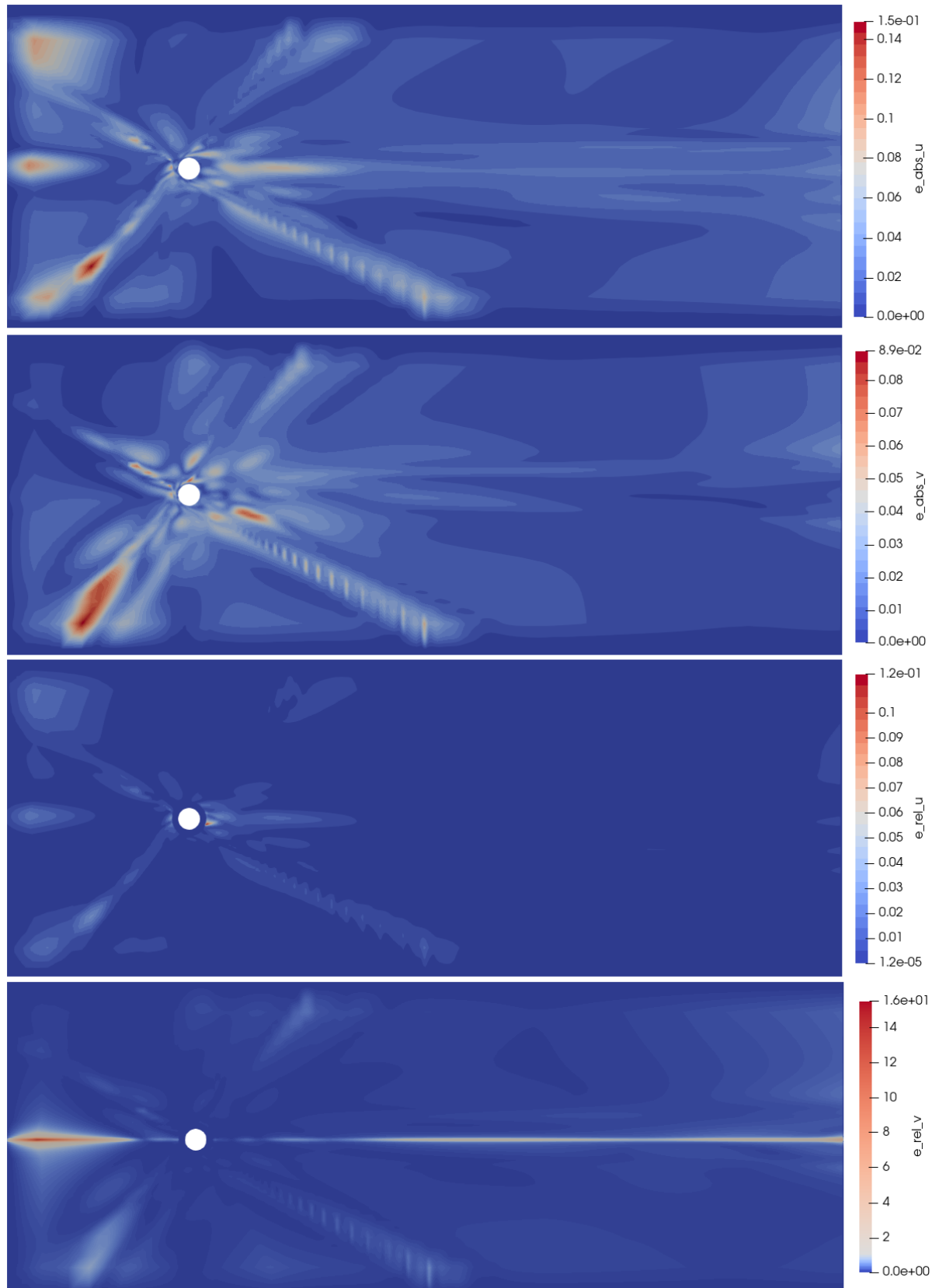
En el caso de la velocidad vertical, los errores son más pronunciados y se extienden por toda la parte longitudinal media del dominio computacional, con valores máximos (de hasta 1600%) a la entrada y salida de flujo.

Figura 45. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 5$.



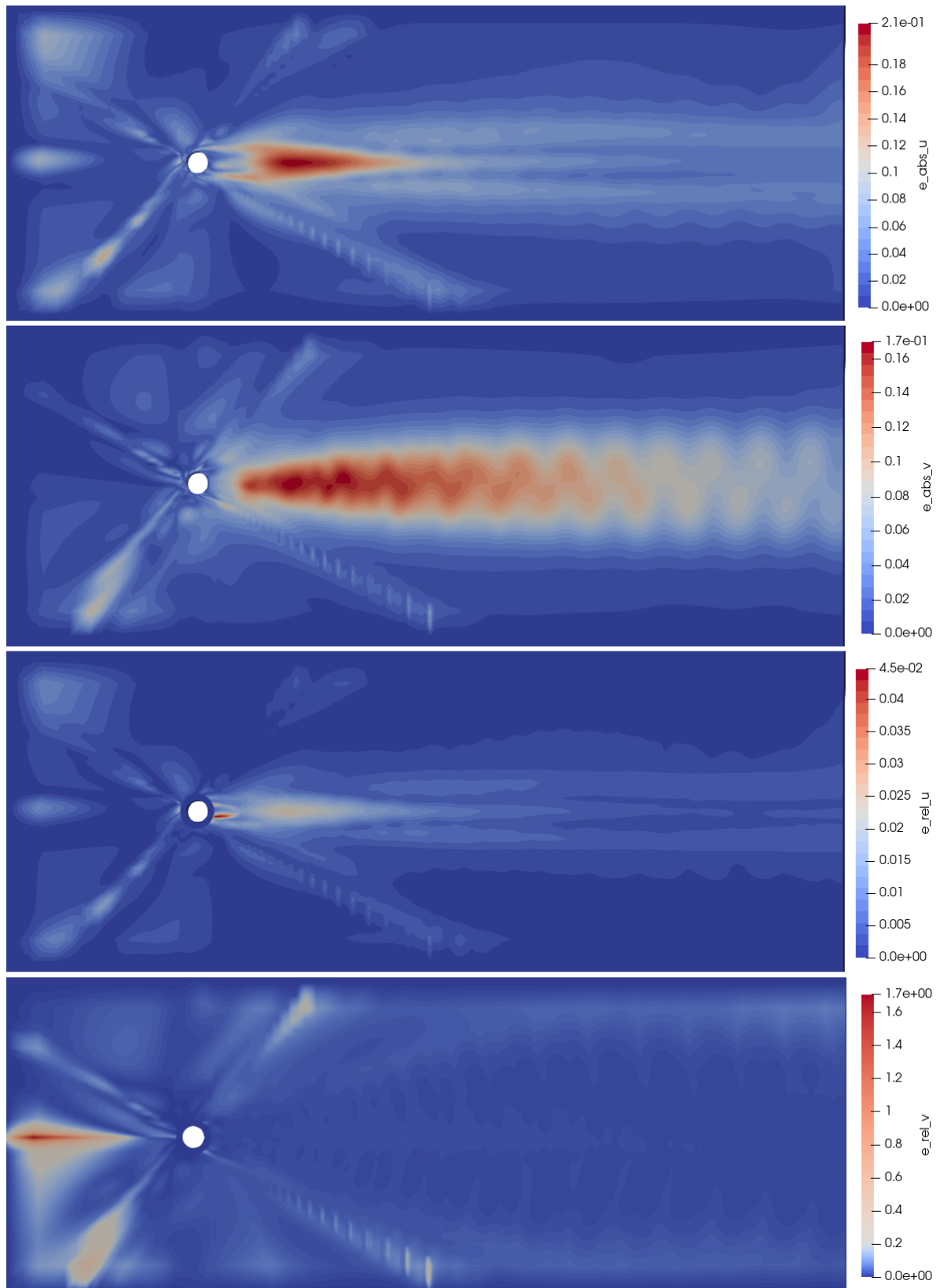
Fuente: Elaboración propia.

Figura 46. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 60$.



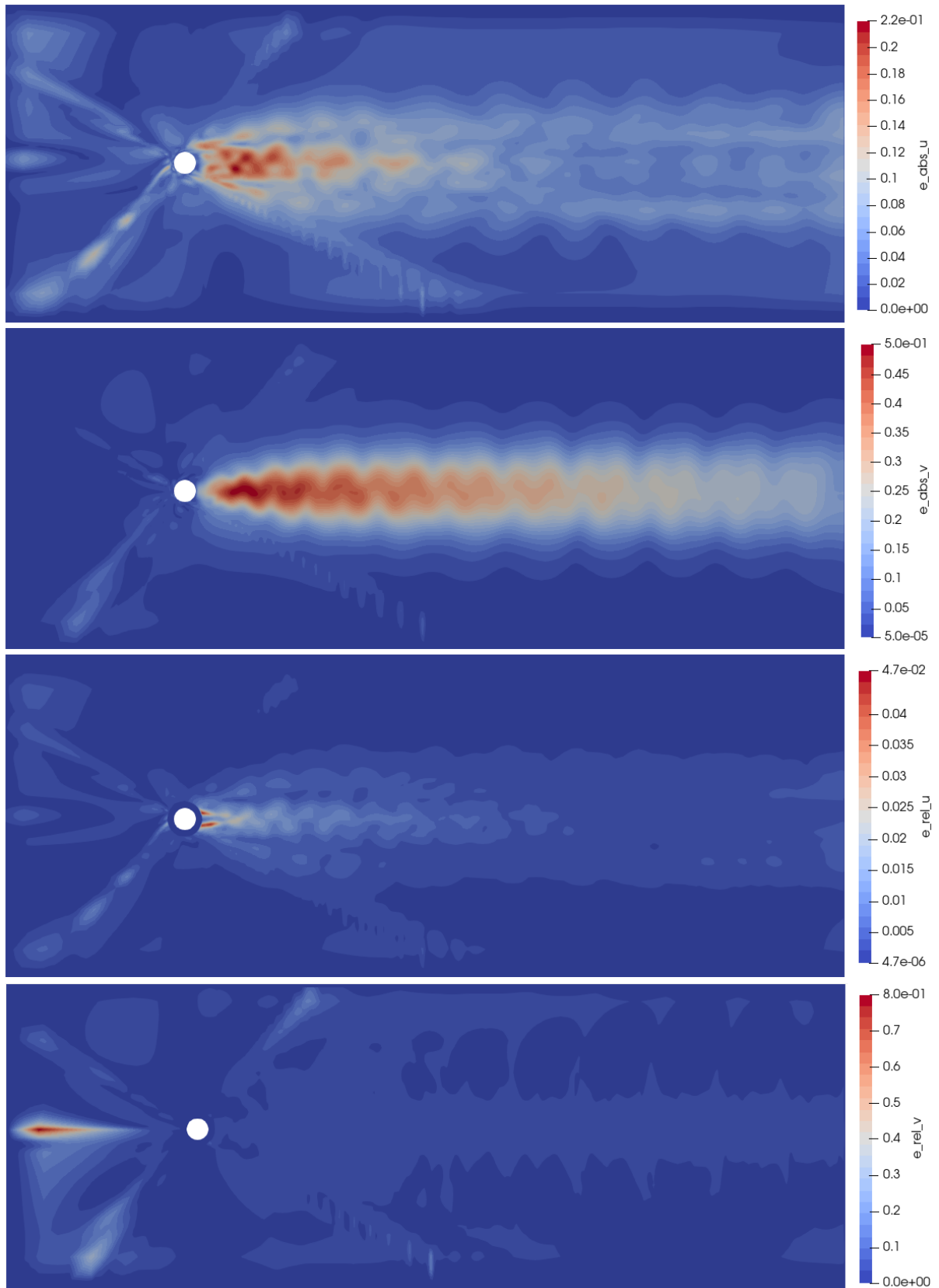
Fuente: Elaboración propia.

Figura 47. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 100$.



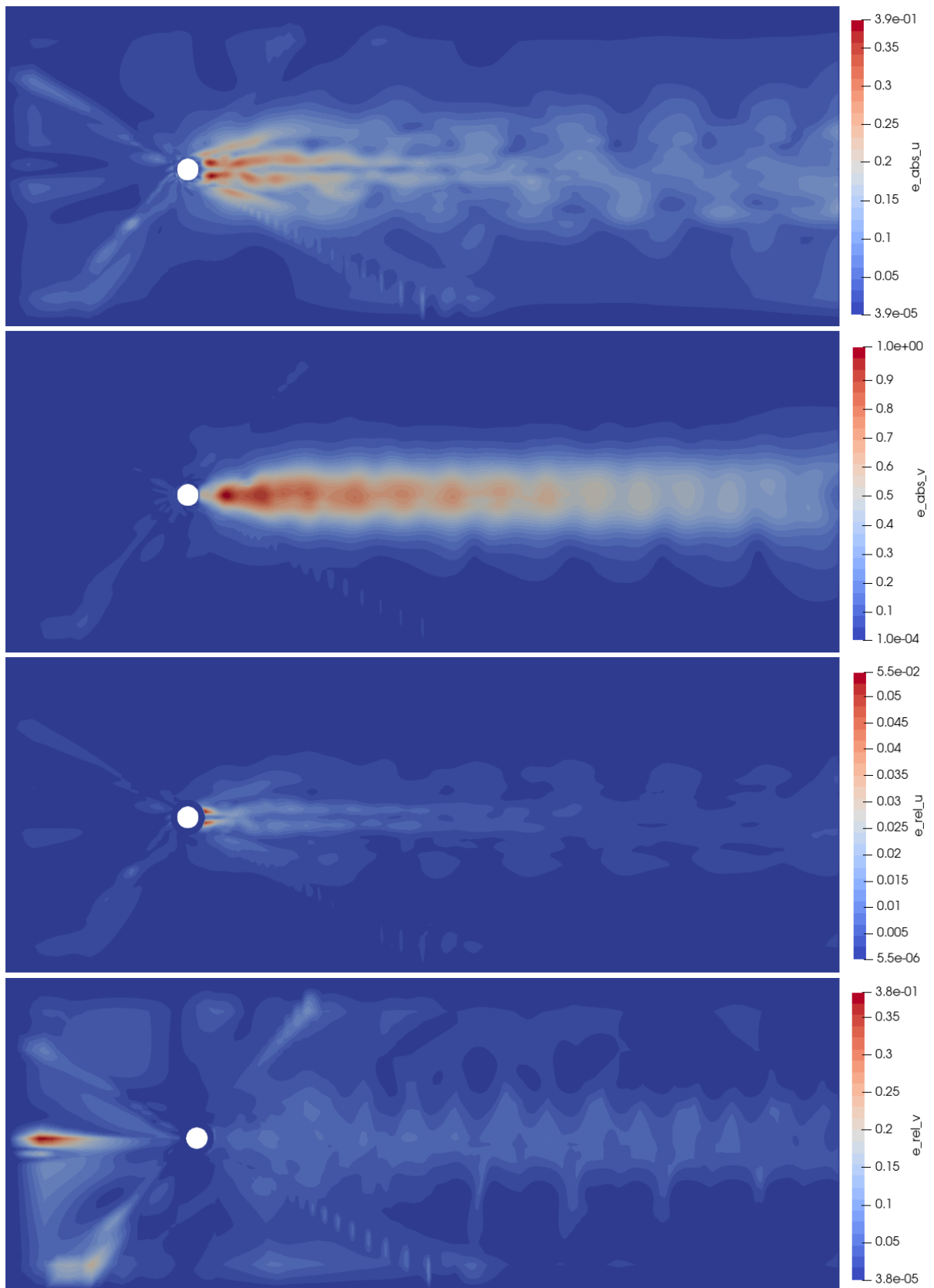
Fuente: Elaboración propia.

Figura 48. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 140$.



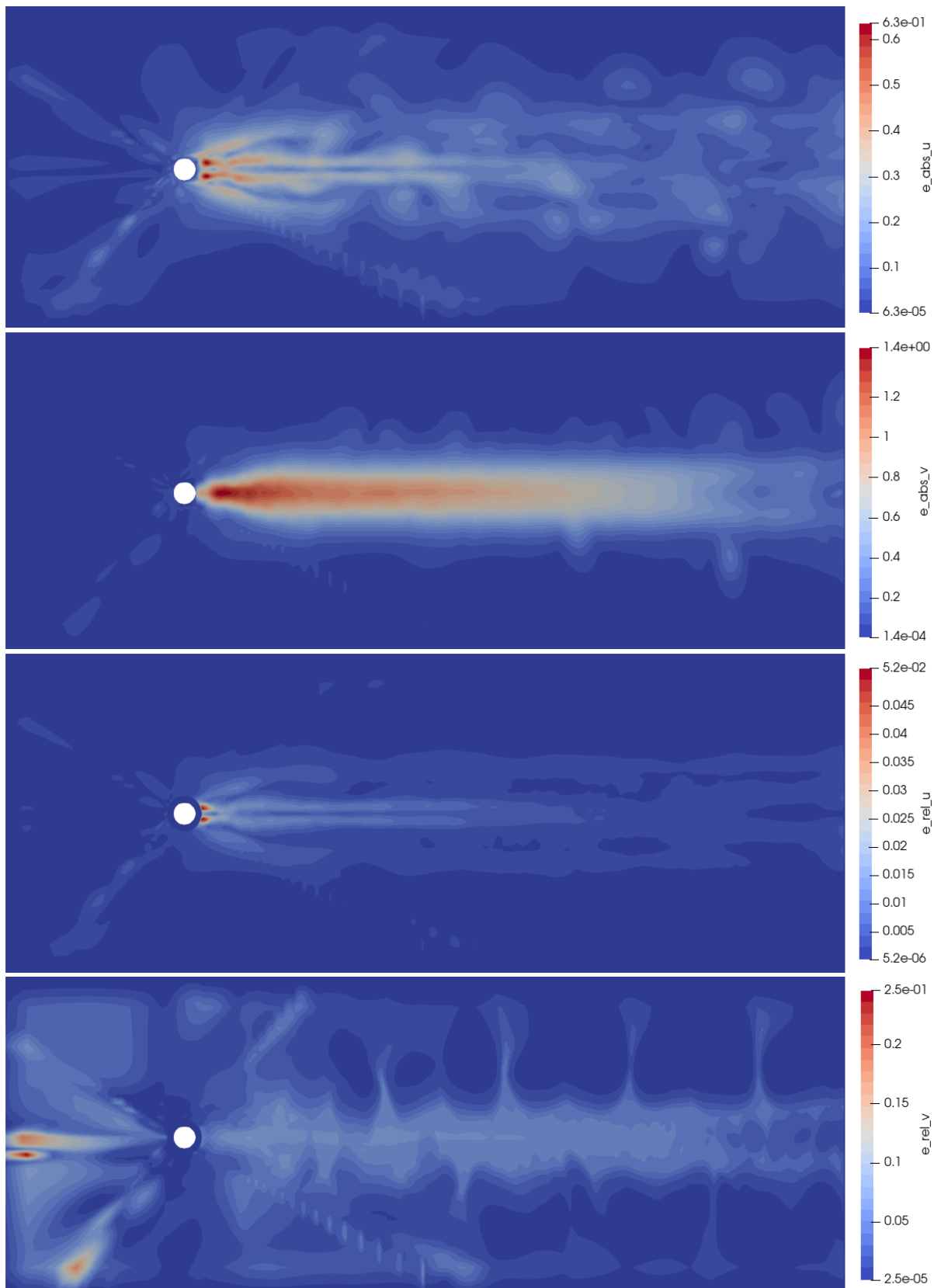
Fuente: Elaboración propia.

Figura 49. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 200$.



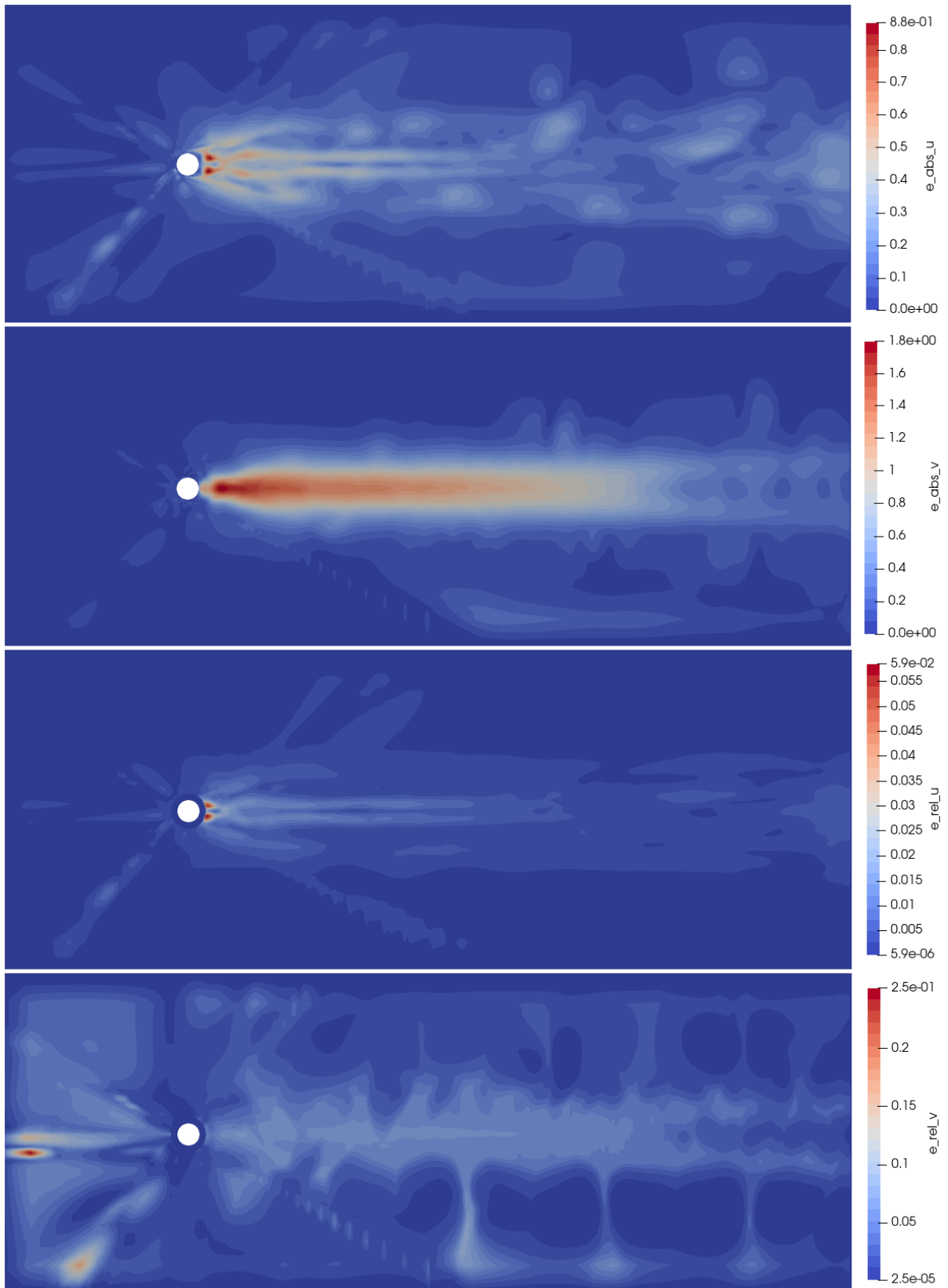
Fuente: Elaboración propia.

Figura 50. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 260$.



Fuente: Elaboración propia.

Figura 51. Representación de errores medios, absolutos y relativos, sobre el dominio computacional, de la predicción mediante RNA de flujo sobre un cilindro para $Re = 300$.



Fuente: Elaboración propia.

Para las predicciones de $Re = 100$ (Figura 47), $Re = 140$ (Figura 48), $Re = 200$ (Figura 49), $Re = 260$ (Figura 50) y $Re = 300$ (Figura 51); se observa que en la componente horizontal de velocidad los errores relativos están por debajo de 6%, siendo una aproximación con buen grado de precisión. Los mayores errores coinciden al encontrarse en la zona posterior del cilindro, específicamente en la región de lanzamiento de estelas.

En el caso de la velocidad vertical, los errores relativos son más pronunciados, pero con valores aceptables, menores al 20% en casi todo el dominio computacional. Para todos estos casos, existen valores máximos de error relativo en regiones aisladas a la entrada del fluido, los errores disminuyen al aumentar Re , (170%, 80%, 38%, 25%, 24.5% para $Re = 100, 140, 200, 260$ y 300 , respectivamente).

En cuanto a los errores absolutos, para todos los casos a excepción de $Re = 5$, se observa valores bajos. En el caso de la componente horizontal de velocidad, los valores máximos varían desde 0.15 hasta 0.88. En cambio, la componente vertical de velocidad presenta valores máximos que varían entre 0.089 hasta 1.8.

Lo anterior indica que la RNA posee buena capacidad de predicción, observándose en los errores absolutos, buena exactitud. Sin embargo, también son notorios en todos los casos, los errores en las regiones diagonales al cilindro. Aunque no se tiene explicación clara para estos errores, se podrían adjudicar al mallado, ya que en esas regiones se da la sucesión de malla, de circular a rectangular, favoreciendo a la generación de una especie de reflexiones muy notorias en el campo de velocidad a bajos Re .

En la Tabla 14 se tabula detalladamente, para cada caso de simulación, la cantidad de celdas que poseen errores dentro de rangos que varían 5%. Resulta evidente que la predicción para $Re = 5$ es deficiente, pero para Re superiores la RNA predice con buen nivel los valores de velocidades. En especial el caso de $Re = 100$ donde se puede asumir que hay una predicción perfecta en la componente horizontal de velocidad (100% de las celdas poseen errores relativos promedios por debajo del 5%) y una muy buena aproximación de la componente vertical de velocidad (95% de las celdas poseen errores relativos promedios por debajo del 5%, y el 4% de las celdas poseen errores entre 5 y 10%).

Tabla 14. Comparación de cantidad de celdas con errores relativos promedios para diferentes rangos.

Error relativo promedio	Número de Celdas para cada caso en estudio													
	Re = 5		Re = 60		Re = 100		Re = 140		Re = 200		Re = 260		Re = 300	
	u	v	u	v	u	v	u	v	u	v	u	v	u	v
<= 5%	28095	97	40011	17525	40044	37917	40043	37082	40033	32508	40032	30020	40006	28951
>5 – 10%	9865	491	28	12366	0	1597	1	2596	11	7376	12	9935	38	11041
>10 – 15%	1234	722	5	4209	0	194	0	161	0	81	0	58	0	37
>15 – 20%	474	788	0	2385	0	96	0	77	0	30	0	24	0	5
> 20%	376	37946	0	3559	0	240	0	128	0	49	0	7	0	10

Fuente: Elaboración propia.

La generalización de la componente horizontal de velocidad (u) es excelente a partir de $Re = 60$, donde se predice (de un total de 40044 celdas evaluadas) 40011 valores con un error promedio igual y por debajo del 5%, representando un 99.9% de las celdas del dominio computacional estudiado. Para $Re = 100$ la predicción es aún mejor, donde el 100% de los valores predichos poseen errores por debajo del 5%. De igual manera, hasta $Re = 300$ se mantiene la calidad de predicción de esta componente de velocidad.

Para la componente vertical de velocidad (v), la predicción es inferior. Presenta errores elevados para $Re = 5$ los cuales disminuyen notablemente a partir de $Re = 60$, donde alrededor del 15% de las celdas poseen valores de error promedio superiores al 15% y el 44% de las celdas un error promedio igual y por debajo del 5%. A partir de $Re = 100$ la predicción mejora notablemente puesto que el 99% de los valores predichos poseen errores por debajo del 10%, obteniéndose entre el 72 y 95% de las celdas con errores por debajo del 5%.

En las figuras que conforman el anexo 1, se detallan mediante diagramas de pasteles los datos de la Tabla 14, favoreciendo su observación y análisis.

Capítulo 5. Conclusiones y trabajo futuro

Conclusiones

La simulación CFD, 2D, de flujo sobre un cilindro, implementada en OpenFOAM, brinda resultados acertados. Los coeficientes de arrastre y sustentación, así como los números de Strouhal obtenidos, concuerdan con la información expuesta en la literatura especializada. Los desarrollos temporales de las simulaciones permiten observar con claridad el fenómeno de generación de estela y desprendimiento de vórtices, característicos del caso estudiado.

El procesamiento de los resultados provenientes de las simulaciones CFD, resultó exitoso. Las variables y los datos considerados para el entrenamiento son suficientes para permitir que la RNA generalice adecuadamente el campo de velocidades.

Los parámetros de configuración e implementación de las RNAs son acertados, la naturaleza de la mayoría de los datos de entrenamiento se acopla a las funciones tangente hiperbólicas (sigmoideas) de las capas ocultas, lo anterior, en conjunto con la normalización de datos favorecieron el proceso de entrenamiento. El uso del método de Levenberg Marquardt incidió en el gasto computacional pero su uso se justifica por el hecho de haber alcanzado buena precisión con una arquitectura de RNA relativamente pobre.

La capacidad de generalización de la RNA profunda, de dos capas ocultas, con 30 neuronas por capa, es notoria, exponiendo para Re entre 60 y 300 una caracterización muy buena del fenómeno. Los campos de velocidad predichos concuerdan con las simulaciones CFD. La precisión de las predicciones y la generalización del fenómeno están evaluadas y validadas.

Una media del 99.96% de las celdas del dominio computacional con errores relativos promedio iguales o inferiores al 5% para la componente horizontal de velocidad y, un similar del 76.58% para la componente vertical de velocidad, dan idea del gran aprendizaje de la RNA.

Las aproximaciones con mayores diferencias se concentran en pequeñas regiones ubicadas diagonalmente al cilindro, a lo largo de estela y a la entrada de flujo. El comportamiento del error en las regiones diagonales (en especial a bajos Re) no es claro, pueden deberse a un fenómeno secundario inducido por el tipo de mallado, puesto que, en esta región existe la transición de malla; de circular a rectangular. Aunque, también puede atribuirse a que la arquitectura de la RNA es relativamente pobre en cuanto al número de Neuronas, ya que en trabajos análogos (de diferente naturaleza) relacionados con CFD (Akdag et al., 2009; Dey &

Das, 2016; Islamoglu & Kurt, 2004; Kim et al., 2020; Kurtulus, 2009; Markopoulos et al., 2016; Sarghini et al., 2003; Taymaz & Islamoglu, 2009) se utilizan hasta 50 neuronas, con bases de datos más pequeñas, logrando valores de correlación superiores a los del presente trabajo.

La predicción de la componente vertical de velocidad es menos precisa, existe una media del 1% (para $Re \geq 100$) de datos del dominio computacional con valores de error relativo promedio superiores al 20%, aun teniendo errores absolutos bajos. Lo anterior puede deberse a que la magnitud de dicha componente es baja, cercana a cero en las zonas pronunciadas de error, donde aproximaciones con pequeñas diferencias absolutas resultan con errores relativos elevados.

En el estudio de predicción de campos de velocidad de flujo sobre un cilindro, expuesto por (Jin et al., 2018), utilizan Redes Neuronales Convolucionales, obteniendo las mayores diferencias de predicción en algunas zonas coincidentes a las del presente estudio, con errores relativos promedios de hasta el 6%, observándose en los reportes gráficos, que los errores de la componente de velocidad vertical son justamente los de mayor magnitud.

Las predicciones para $Re = 5$ y $Re = 300$, casos fuera del rango de entrenamiento, indican que la RNA aprende las características inherentes del flujo alrededor del cilindro, como la longitud de formación de la estela y el desprendimiento de vórtices. Aunque, para $Re = 5$ la RNA posee errores considerables en la componente vertical de velocidad, la componente horizontal posee un 70% de resultados con error relativo promedio menor al 5% y durante el desarrollo temporal de la “simulación sintética” no realiza lanzamiento de vórtices, coincidiendo con los resultados CFD. En cambio, para $Re = 300$ se evidencia una muy buena generalización, altamente coincidente en valores de velocidad y en el desarrollo temporal de la simulación, ratificando la capacidad de extrapolación para Re superiores a los del rango de entrenamiento.

La RNA no es capaz de predecir, extrapolar, con precisión el comportamiento del flujo para Re inferiores a los del entrenamiento, siendo necesario realizar entrenamientos con más datos de simulaciones dentro de este rango. Sin embargo, si se requiere una mayor precisión, una nueva arquitectura de RNA con más neuronas por capa sería conveniente. Se hace hincapié en la cantidad de neuronas por capa, en base a la tendencia de mejora de correlación observada en las diferentes arquitecturas de prueba ejecutadas (ver Tabla 11).

La limitante principal en la definición de la arquitectura de la RNA y en la propuesta de una mejor versión, fue la capacidad de cómputo disponible, donde el periodo de entrenamiento resultó muy extenso. El tiempo de entrenamiento mejoró con la reducción de datos, implicando la delimitación del dominio computacional de estudio, dejando al margen zonas de campo lejano y de entrada de flujo que también son de interés.

Por encima de las limitantes, el uso del modelo de RNA logrado tiene sus ventajas, entre ellas, en la ejecución de las denominadas “simulaciones sintéticas”, donde es necesario alrededor de 10 s (tiempo de computadora Intel Core™ i7-10750H CPU @ 2.60 GHz, 16 GB RAM) para presentar resultados, tiempo que resulta ínfimo ante el desarrollo de una simulación CFD.

Trabajo futuro

Una propuesta importante para trabajos futuros, es seguir estudiando o analizando la RNA lograda, puesto que de ella se puede obtener diversa información a más de la ya brindada. Por ejemplo, el desarrollo de “simulaciones sintéticas”:

- ✓ Para cualquier Re dentro del rango de entrenamiento (interpolación en base a Re) e incluso por encima y cercano de dicho rango (extrapolación en base a Re).
- ✓ Para pasos de tiempo distintos a los de entrenamiento (0.1 s) (Interpolación en base al tiempo).
- ✓ Para tiempos totales de simulación, diferentes a los de entrenamiento (5 s).
- ✓ Para el mismo dominio computacional, pero con una densidad de celdas diferente. Nuevas celdas cuyas coordenadas sean distintas a las consideradas en el entrenamiento (interpolación espacial).
- ✓ Para regiones externas y próximas al dominio computacional (extrapolación espacial).
- ✓ Para una combinación de parte o todos los casos anteriores.

Otra propuesta a futuro, consiste en aplicar técnicas de reducción de dimensionalidad y de datos para lograr un conjunto de valores de entrenamiento más compacto, que minorice los tiempos de entrenamiento y si es posible, optimice la calidad del aprendizaje.

Finalmente, se podría analizar la influencia del mallado en los resultados de la RNA, pues al parecer en este caso de estudio existiría un fenómeno secundario de reflexión o inducción sobre el campo de velocidades, captado por la RNA y especialmente notorio en Re bajos.

Referencias

- Akdag, U., Komur, M. A., & Ozguc, A. F. (2009). Estimation of heat transfer in oscillating annular flow using artificial neural networks. *Advances in Engineering Software*, 40(9), 864–870. <https://doi.org/10.1016/j.advengsoft.2009.01.010>
- Ben, A. Q., & Dou, H. S. (2015). Simulation and Stability Study of the Flow Around a Cylinder in Infinite Domain. In *Procedia Engineering* (Vol. 126). Elsevier B.V. <https://doi.org/10.1016/j.proeng.2015.11.180>
- Berzal, F. (2018). Redes Neuronales & Deep Learning. In *Departamento de Ciencias de la Computacion e IA*. Berzal Fernando. <http://libros.univalle.edu.co/index.php/programaeditorial/catalog/book/64>
- Braza, M., Chassaing, P., & Ha Minh, H. (1986). Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics*, 165(1986), 79–130. <https://doi.org/10.1017/S0022112086003014>
- Breuer, M. (2000). A challenging test case for large eddy simulation: High Reynolds number circular cylinder flow. *International Journal of Heat and Fluid Flow*, 21(5), 648–654. [https://doi.org/10.1016/S0142-727X\(00\)00056-4](https://doi.org/10.1016/S0142-727X(00)00056-4)
- Bright, I., Lin, G., & Kutz, J. N. (2013). Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. *Physics of Fluids*, 25(12). <https://doi.org/10.1063/1.4836815>
- Caicedo B, E. F., & López S, J. A. (2009). Una aproximación práctica a las redes neuronales artificiales. In *Una aproximación práctica a las redes neuronales artificiales*. Programa Editorial Universidad del Valle. <https://doi.org/10.25100/peu.64>
- Casacuberta, J., Gamez, P. J., & Rausch, G. (2014). *OpenFOAM guide for beginners*. ETSEIAT-UPC. https://the-foam-house5.webnode.es/_files/200000023-6f88b70855/Introduction.pdf
- Chopra, G., & Mittal, S. (2019). Drag coefficient and formation length at the onset of vortex shedding. *Physics of Fluids*, 31(1). <https://doi.org/10.1063/1.5075610>
- De las Heras, S. (2012). *Mecánica de Fluidos en Ingeniería* (O. de P. A. D. de la UPC (ed.)).

Oficina de Publicacions Acadèmiques Digitals de la UPC.

- Dennis, S. C. R., & Chang, G. zu. (1970). Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. *Journal of Fluid Mechanics*, 42(3), 471–489. <https://doi.org/10.1017/S0022112070001428>
- Dey, P., & Das, A. (2016). Numerical analysis and prediction of unsteady forced convection over a sharp and rounded edged square cylinder. *Journal of Applied Fluid Mechanics*, 9(3), 1189–1199. <https://doi.org/10.18869/acadpub.jafm.68.228.24156>
- Ding, H., Shu, C., Yeo, K. S., & Xu, D. (2004). Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method. *Computer Methods in Applied Mechanics and Engineering*, 193(9–11), 727–744. <https://doi.org/10.1016/j.cma.2003.11.002>
- Duraisamy, K., Zhang, Z. J., & Singh, A. P. (2015). New approaches in turbulence and transition modeling using data-driven techniques. *53rd AIAA Aerospace Sciences Meeting, January*, 1–14. <https://doi.org/10.2514/6.2015-1284>
- Farhoud, R. K., Amiralai, S., Jabbari, G., & Amiralai, S. (2012). Numerical Study of Unsteady Laminar Flow around a Circular Cylinder. *Journal of Civil Engineering and Urbanism*, 2(2), 63–67.
- Ferre, J., Rallo, R., Arenas, A., & Giralt, F. (1996). Identification of coherent structures in turbulent shear flows with a Fuzzy ARTMAP Neural Network. *International Journal of Neural Systems*, 7(5), 559–568.
- Fornberg, B. (1980). A numerical study of steady viscous flow past a circular cylinder. *J. Fluid Mech.*, 98(4, Jun.26, 1980), 819–855.
- Franke, J., & Frank, W. (2002). Large eddy simulation of the flow past a circular cylinder at $Re_D = 3900$. *Journal of Wind Engineering and Industrial Aerodynamics*, 90(10), 1191–1206. [https://doi.org/10.1016/S0167-6105\(02\)00232-5](https://doi.org/10.1016/S0167-6105(02)00232-5)
- Fukami, K., Fukagata, K., & Taira, K. (2019). Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870, 106–120. <https://doi.org/10.1017/jfm.2019.238>
- Giralt, F., Rallo, R., Arenas, A., Kopp, G. A., & Introduction, I. (2000). The simulation and

interpretation of free turbulence with a cognitive. *Physics of Fluids*, 12(7), 1826–1835.

- Hagan, M., Demuth, H., Beale, M., & De Jesús, O. (2014). *Neural network design* (M. Hagan (ed.); Second). Hagan, Martín. <https://doi.org/10.1109/ijcnn.1989.118439>
- Hammelman, J., Lobo, D., & Levin, M. (2016). *Artificial Neural Networks as Models of Robustness in Development and Regeneration: Stability of Memory During Morphological Remodeling*. 628, 45–65. https://doi.org/10.1007/978-3-319-28495-8_3
- Harichandan, A. B., & Roy, A. (2010). Numerical investigation of low Reynolds number flow past two and three circular cylinders using unstructured grid CFR scheme. *International Journal of Heat and Fluid Flow*, 31(2), 154–171. <https://doi.org/10.1016/j.ijheatfluidflow.2010.01.007>
- Haykin, S. (2005). *Neural Networks: A Comprehensive Foundation*. Pearson Education. <https://books.google.com.ec/books?id=bX4pAQAAAMAJ>
- Henderson, R. D. (1995). Details of the drag curve near the onset of vortex shedding. *Physics of Fluids*, 7(9), 2102–2104. <https://doi.org/10.1063/1.868459>
- Huq, S., Ahmed, S., Saeedi, M., & Nanjundiah, R. (2013). *Exploring OpenFOAM*. GitHub Repository. <https://nbviewer.jupyter.org/github/Comflics/Exploring-OpenFOAM/blob/master/Exploring-OpenFOAM.pdf>
- Islamoglu, Y., & Kurt, A. (2004). Heat transfer analysis using ANNs with experimental data for air flowing in corrugated channels. *International Journal of Heat and Mass Transfer*, 47(6–7), 1361–1365. <https://doi.org/10.1016/j.ijheatmasstransfer.2003.07.031>
- Jiang, H. (2020). Separation angle for flow past a circular cylinder in the subcritical regime. *Physics of Fluids*, 32(1). <https://doi.org/10.1063/1.5139479>
- Jiang, H., & Cheng, L. (2017). Strouhal-Reynolds number relationship for flow past a circular cylinder. *Journal of Fluid Mechanics*, 832, 170–188. <https://doi.org/10.1017/jfm.2017.685>
- Jiang, H., & Cheng, L. (2018). Hydrodynamic characteristics of flow past a square cylinder at moderate Reynolds numbers. *Physics of Fluids*, 30(10). <https://doi.org/10.1063/1.5050439>
- Jiang, H., Cheng, L., & An, H. (2017). On numerical aspects of simulating flow past a circular

- cylinder. *International Journal for Numerical Methods in Fluids*, 85(2), 113–132. <https://doi.org/10.1002/flid.4376>
- Jiang, H., Cheng, L., Draper, S., & An, H. (2017a). Prediction of the secondary wake instability of a circular cylinder with direct numerical simulation. *Computers & Fluids*, 149. <https://doi.org/10.1016/j.compfluid.2017.03.018>
- Jiang, H., Cheng, L., Draper, S., & An, H. (2017b). Two- and three-dimensional instabilities in the wake of a circular cylinder near a moving wall. *Journal of Fluid Mechanics*, 812(April 2019), 435–462. <https://doi.org/10.1017/jfm.2016.810>
- Jiang, H., Cheng, L., Draper, S., An, H., & Tong, F. (2016). Three-dimensional direct numerical simulation of wake transitions of a circular cylinder. *Journal of Fluid Mechanics*, 801, 353–391. <https://doi.org/10.1017/jfm.2016.446>
- Jiang, H., Cheng, L., Tong, F., Draper, S., & An, H. (2016). Stable state of mode A for flow past a circular cylinder. *Physics of Fluids*, 28(10). <https://doi.org/10.1063/1.4964379>
- Jin, X., Cheng, P., Chen, W. L., & Li, H. (2018). Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder. *Physics of Fluids*, 30(4). <https://doi.org/10.1063/1.5024595>
- Kim, J. H., Kim, Y., & Lu, W. (2020). Prediction of ice resistance for ice-going ships in level ice using artificial neural network technique. *Ocean Engineering*, 217(September), 108031. <https://doi.org/10.1016/j.oceaneng.2020.108031>
- Kornbleuth, M. (2016). *Studying the Viscous Flow Around a Cylinder Using OpenFoam*. 1–34.
- Kozlov, I. M., Dobrego, K. V., & Gnesdilov, N. N. (2011). Application of RES methods for computation of hydrodynamic flows by an example of a 2D flow past a circular cylinder for $Re=5-200$. *International Journal of Heat and Mass Transfer*, 54(4), 887–893. <https://doi.org/https://doi.org/10.1016/j.ijheatmasstransfer.2010.10.013>
- Kurtulus, D. F. (2009). Ability to forecast unsteady aerodynamic forces of flapping airfoils by artificial neural network. *Neural Computing and Applications*, 18(4), 359–368. <https://doi.org/10.1007/s00521-008-0186-2>
- La, F. (2019). A Closer Look at Neural Networks. *MSDN Magazine Issues*, 34(2). <https://docs.microsoft.com/en-us/archive/msdn-magazine/2019/february/artificially->

intelligent-a-closer-look-at-neural-networks

- Ledesma, S. (2006). Las Redes Neuronales implementación y consideraciones prácticas. *Fifth Mexican International Conference on Artificial Intelligence*, 50.
- López Peña, F., Duro, R. J., & Sánchez Simón, M. (2002). Detecting coherent structures in a turbulent wake by using delay based networks. *Computer Standards and Interfaces*, 24(2), 171–184. [https://doi.org/10.1016/S0920-5489\(02\)00011-9](https://doi.org/10.1016/S0920-5489(02)00011-9)
- Markidis, S. (2021). Physics-Informed Deep-Learning for Scientific Computing. *ArXiv Preprint ArXiv:2103.09655*.
- Markopoulos, A., Georgiopoulos, S., & Manolakos, D. (2016). On the use of back propagation and radial basis function neural networks in surface roughness prediction. *Journal of Industrial Engineering International*, 12, 389–400. <https://doi.org/10.1007/s40092-016-0146-x>
- MathWorks. (2021). *MathWorks*.
<https://es.mathworks.com/help/deeplearning/ref/trainlm.html>.
- Meyer, M., Hickel, S., & Adams, N. A. (2010). Assessment of Implicit Large-Eddy Simulation with a Conservative Immersed Interface Method for turbulent cylinder flow. *International Journal of Heat and Fluid Flow*, 31(3), 368–377. <https://doi.org/10.1016/j.ijheatfluidflow.2010.02.026>
- Mott, R. L., & Untener, J. A. (2015). *Applied Fluid Mechanics*. Pearson Education. <https://books.google.com.ec/books?id=unWxCQAAQBAJ>
- Norberg, C. (2001). Flow around a circular cylinder: Aspects of fluctuating lift. *Journal of Fluids and Structures*, 15(3), 459–469. <https://doi.org/https://doi.org/10.1006/jfls.2000.0367>
- Pacheco-Vega, A., Sen, M., Yang, K. T., & McClain, R. (2001). Neural network analysis of fin-tube refrigerating heat exchanger with limited experimental data. *International Journal of Heat and Mass Transfer*, 44, 763–770. [https://doi.org/10.1016/S0017-9310\(00\)00139-3](https://doi.org/10.1016/S0017-9310(00)00139-3)
- Palma, J., & Marín, R. (2007). Inteligencia Artificial. Técnicas, métodos y aplicaciones. In *Inteligencia Artificial* (Vol. 2, Issue 6). McGraw-Hill Education. <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Inteligencia+Artificial>

:+un+enfoco+moderno#0

- Panton, R. (1981). Newtonian Fluids. The Navier-Stokes Equations. In *Mathematics in Science and Engineering* (Fourth, Vol. 158, Issue C). Wiley. [https://doi.org/10.1016/S0076-5392\(08\)61790-7](https://doi.org/10.1016/S0076-5392(08)61790-7)
- Park, J., Kwon, K., & Choi, H. (1998). Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160. *KSME International Journal*, 12(6), 1200–1205. <https://doi.org/10.1007/BF02942594>
- Parnaudeau, P., Carlier, J., Heitz, D., & Lamballais, E. (2008). Experimental and numerical studies of the flow over a circular cylinder at Reynolds number 3900. *Physics of Fluids*, 20(8). <https://doi.org/10.1063/1.2957018>
- Piñol, S., & Grau Vidal, F. (1996). Flujo alrededor de un cilindro: efecto de la condición de contorno en la pared y de la anchura del dominio. *Revista Internacional de Métodos Numéricos Para Cálculo y Diseño En Ingeniería*, 12(3), 313-330–330.
- Ponce, P. (2011). *Inteligencia Artificial con Aplicaciones en Ingeniería*. Marcombo. <https://books.google.com.ec/books?id=myyQpwAACAAJ>
- Posdziech, O., & Grundmann, R. (2001). Numerical Simulation of the Flow Around an Infinitely Long Circular Cylinder in the Transition Regime. *Theoretical and Computational Fluid Dynamics*, 15(2), 121–141. <https://doi.org/10.1007/s001620100046>
- Potter, M. C., Wiggert, D. C., Hondzo, M., & Shih, T. I. P. (2002). *Mecánica de fluidos*. International Thomson. <https://books.google.com.ec/books?id=qofVwAEACAAJ>
- Raissi, M., Yazdani, A., & Karniadakis, G. E. (2018). *Hidden Fluid Mechanics: A Navier-Stokes Informed Deep Learning Framework for Assimilating Flow Visualization Data*. <http://arxiv.org/abs/1808.04327>
- Rajani, B. N., Kandasamy, A., & Majumdar, S. (2009). Numerical simulation of laminar flow past a circular cylinder. *Applied Mathematical Modelling*, 33(3), 1228–1247. <https://doi.org/10.1016/j.apm.2008.01.017>
- Rapaka, N. R., & Sarkar, S. (2016). An immersed boundary method for direct and large eddy simulation of stratified flows in complex geometry. *Journal of Computational Physics*, 322(June), 511–534. <https://doi.org/10.1016/j.jcp.2016.06.036>

- Samarasinghe, S. (2016). Order in the Black Box: Consistency and Robustness of Hidden Neuron Activation of Feed Forward Neural Networks and Its Use in Efficient Optimization of Network Structure. In S. Shanmuganathan & S. Samarasinghe (Eds.), *Artificial Neural Network Modelling* (pp. 15–43). Springer International Publishing. https://doi.org/10.1007/978-3-319-28495-8_2
- Sarghini, F., Felice, G., & Santini, S. (2003). Neural networks based subgrid scale modeling in large eddy simulations. *Computers and Fluids*, 32(1), 97–108. [https://doi.org/10.1016/S0045-7930\(01\)00098-6](https://doi.org/10.1016/S0045-7930(01)00098-6)
- Shao, J., & Zhang, C. (2006). Numerical analysis of the flow around a circular cylinder using RANS and LES. *International Journal of Computational Fluid Dynamics*, 20(5), 301–307. <https://doi.org/10.1080/10618560600898437>
- Sumer, B. M., & Fredsøe, J. (1997). *Hydrodynamics Around Cylindrical Structures*. WORLD SCIENTIFIC. <https://doi.org/10.1142/3316>
- Tahavvor, A. R., & Yaghoubi, M. (2008). Natural cooling of horizontal cylinder using Artificial Neural Network (ANN). *International Communications in Heat and Mass Transfer*, 35(9), 1196–1203. <https://doi.org/10.1016/j.icheatmasstransfer.2008.05.009>
- Tamura, T., Ohta, I., & Kuwahara, K. (1990). On the reliability of two-dimensional simulation for unsteady flows around a cylinder-type structure. *Journal of Wind Engineering and Industrial Aerodynamics*, 35, 275–298. [https://doi.org/https://doi.org/10.1016/0167-6105\(90\)90221-W](https://doi.org/https://doi.org/10.1016/0167-6105(90)90221-W)
- Taymaz, I., & Islamoglu, Y. (2009). Prediction of convection heat transfer in converging-diverging tube for laminar air flowing using back-propagation neural network. *International Communications in Heat and Mass Transfer*, 36(6), 614–617. <https://doi.org/10.1016/j.icheatmasstransfer.2009.03.009>
- Tritton, D. J. (1959). Experiments on the flow past a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics*, 6(4), 547–567. <https://doi.org/10.1017/S0022112059000829>
- Tuann, S., & Olson, M. D. (1978). Numerical studies of the flow around a circular cylinder by a finite element method. *Computers & Fluids*, 6(4), 219–240. [https://doi.org/https://doi.org/10.1016/0045-7930\(78\)90015-4](https://doi.org/https://doi.org/10.1016/0045-7930(78)90015-4)

- Tutar, M., & Hold, A. E. (2001). Computational modelling of flow around a circular cylinder in sub-critical flow regime with various turbulence models. *International Journal for Numerical Methods in Fluids*, 35(7), 763–784. [https://doi.org/10.1002/1097-0363\(20010415\)35:7<763::AID-FLD112>3.0.CO;2-S](https://doi.org/10.1002/1097-0363(20010415)35:7<763::AID-FLD112>3.0.CO;2-S)
- Viquerat, J., & Hachem, E. (2020). A supervised neural network for drag prediction of arbitrary 2D shapes in laminar flows at low Reynolds number. *Computers and Fluids*, 210, 104645. <https://doi.org/10.1016/j.compfluid.2020.104645>
- Wang, X., Shu, C., Wu, J., & Yang, L. M. (2014). A boundary condition-implemented immersed boundary-lattice Boltzmann method and its application for simulation of flows around a circular cylinder. *Advances in Applied Mathematics and Mechanics*, 6(6), 811–829. <https://doi.org/10.4208/aamm.2013.m-s2>
- Wang, Y., Zhang, S., Ma, Z., & Yang, Q. (2020). Artificial neural network model development for prediction of nonlinear flow in porous media. *Powder Technology*, 373, 274–288. <https://doi.org/10.1016/j.powtec.2020.06.048>
- White, F. (2010). *Fluid Mechanics*. McGraw-Hill Education. <https://books.google.com.ec/books?id=d48lQwAACAAJ>
- Wu, M. H., Wen, C. Y., Yen, R. H., Weng, M. C., & Wang, A. B. (2004). Experimental and numerical study of the separation angle for flow around a circular cylinder at low Reynolds number. *Journal of Fluid Mechanics*, 515, 233–260. <https://doi.org/10.1017/S0022112004000436>
- Yao, J., Feng, W., Chen, B., Zhou, W., & Cao, S. (2017). A Quadtree-gridding LBM with Immersed Boundary for Two-dimension Viscous Flows. *IOP Conference Series: Materials Science and Engineering*, 224(1). <https://doi.org/10.1088/1757-899X/224/1/012023>
- Ye, S., Zhang, Z., Song, X., Wang, Y., Chen, Y., & Huang, C. (2020). A flow feature detection method for modeling pressure distribution around a cylinder in non-uniform flows by using a convolutional neural network. *Scientific Reports*, 10(1), 1–10. <https://doi.org/10.1038/s41598-020-61450-z>
- Ye, Z., Chen, Q., Zhang, Y., Zou, J., & Zheng, Y. (2019). Identification of vortex structures in flow field images based on convolutional neural network and dynamic mode decomposition. *Traitement Du Signal*, 36(6), 501–506. <https://doi.org/10.18280/ts.360604>

Yegnanarayana, B. (2005). *Artificial Neural Networks*. Prentice Hall.

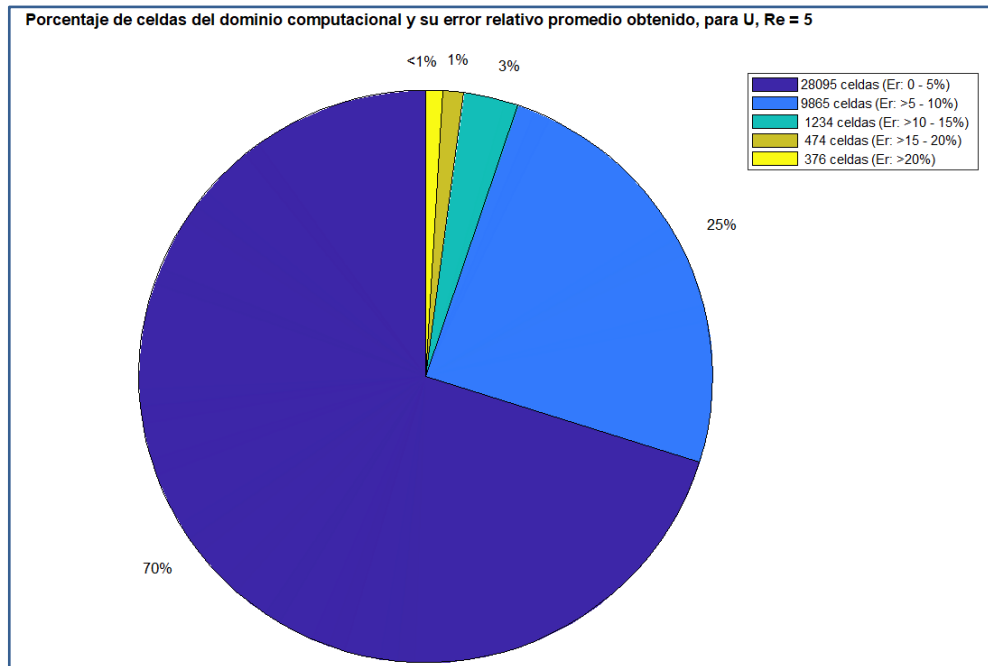
You, Y., Yan, Z., Zhong, Y., & Zhang, L. (2017). *Parameter analysis of numerical simulation on flow around a cylinder*. *135(Iccte)*, 162–167. <https://doi.org/10.2991/iccte-17.2017.28>

Zeng, P. (1998). Neural computing in mechanics. *Applied Mechanics Reviews*, *51*(2), 173–197. <https://doi.org/10.1115/1.3098995>

Zhang, K., Katsuchi, H., Zhou, D., Yamada, H., & Han, Z. (2016). Numerical study on the effect of shape modification to the flow around circular cylinders. *Journal of Wind Engineering and Industrial Aerodynamics*, *152*, 23–40. <https://doi.org/10.1016/j.jweia.2016.02.008>

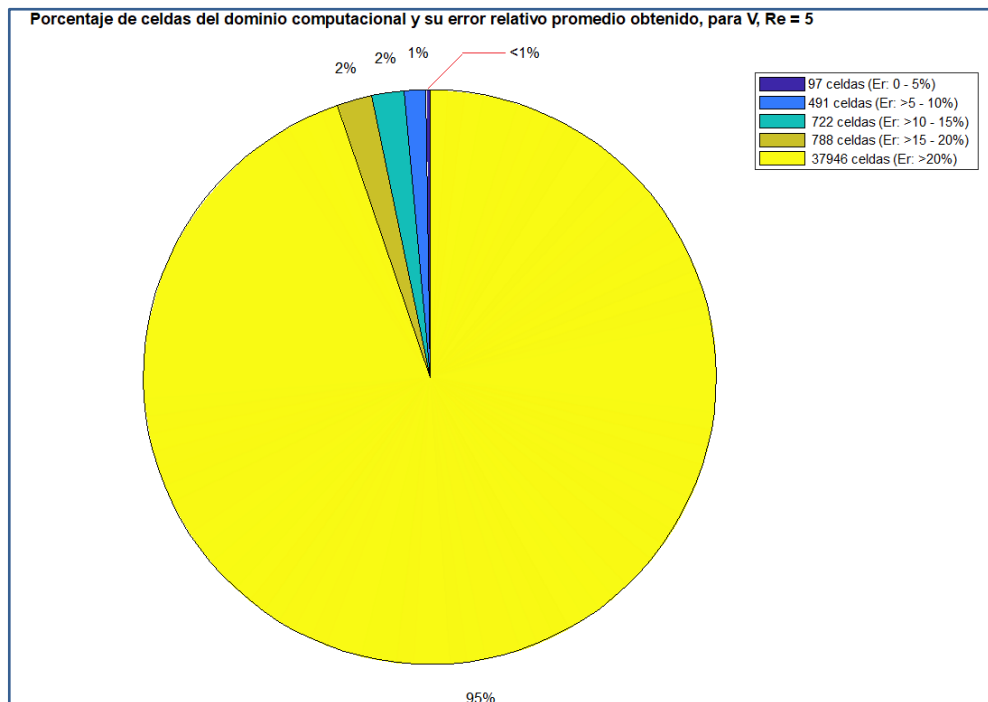
Anexo 1. Diagramas de pastel indicando los porcentajes de celdas agrupadas por rangos de error relativo.

Porcentajes de celdas agrupadas por rangos de error relativo promedio. Para U , $Re = 5$.



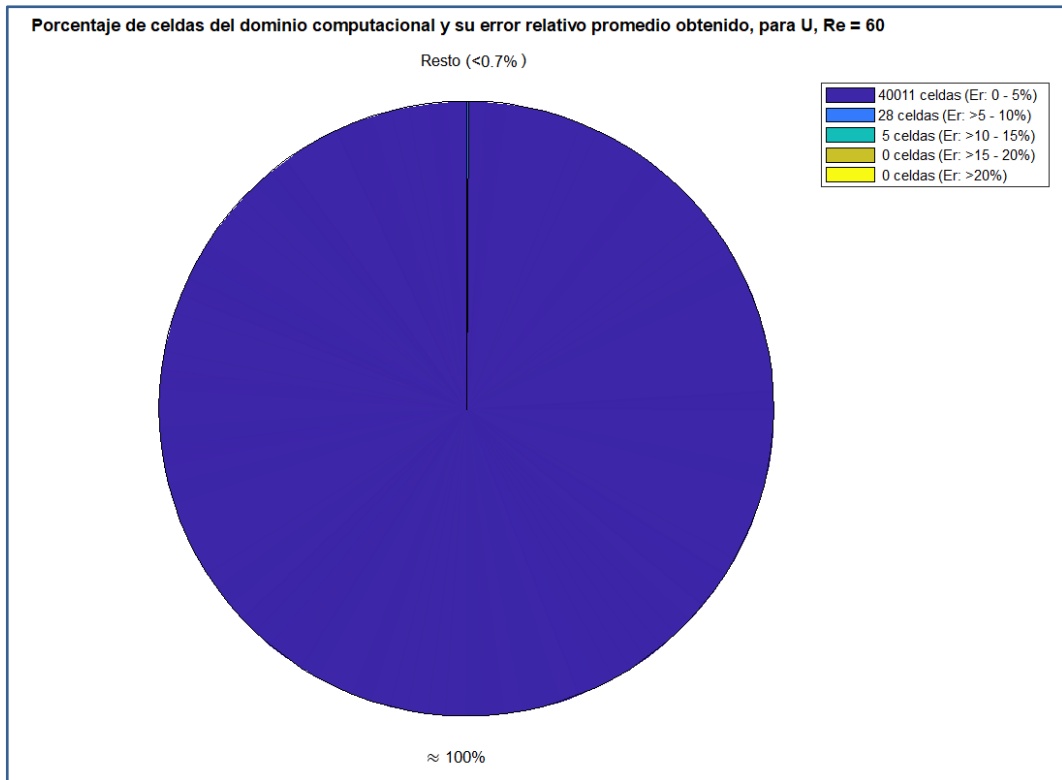
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rangos de error relativo promedio. Para V , $Re = 5$.



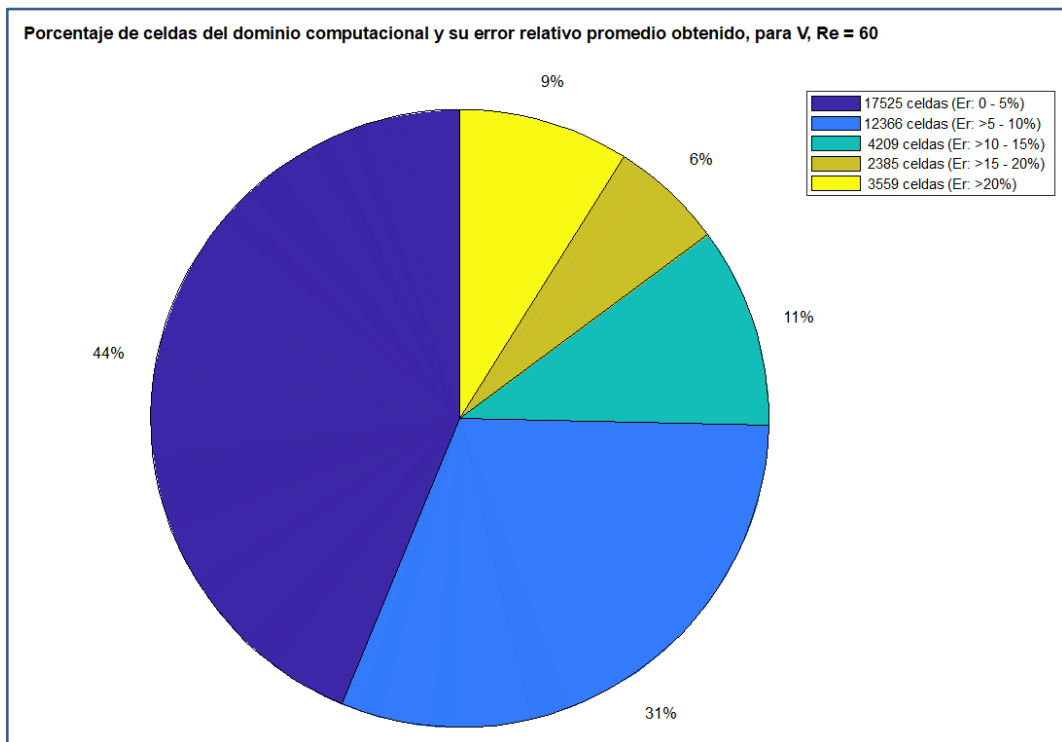
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rangos de error relativo promedio. Para U, Re = 60.



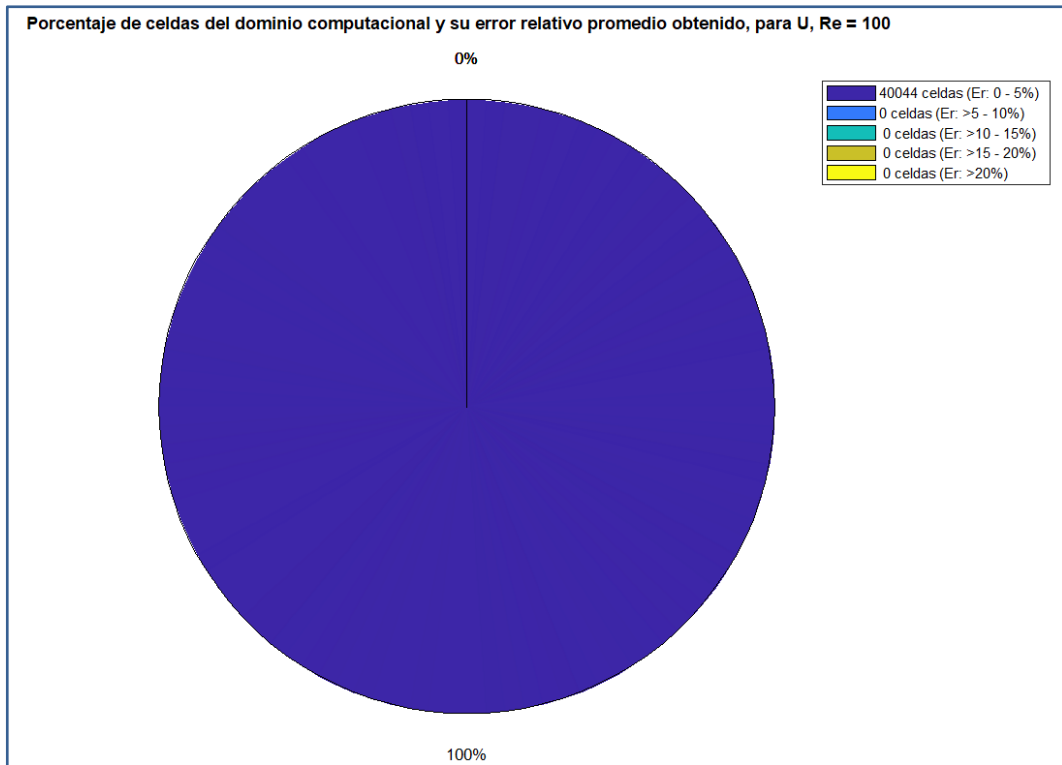
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rangos de error relativo promedio. Para V, Re = 60.



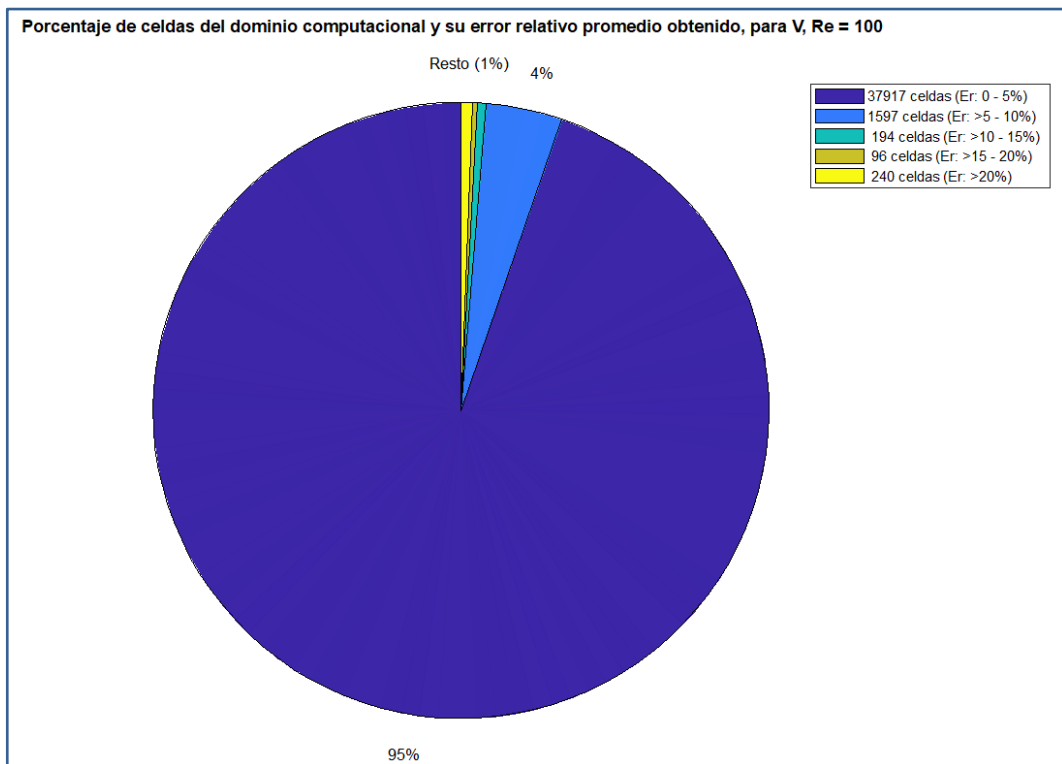
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rangos de error relativo promedio. Para U, Re = 100.



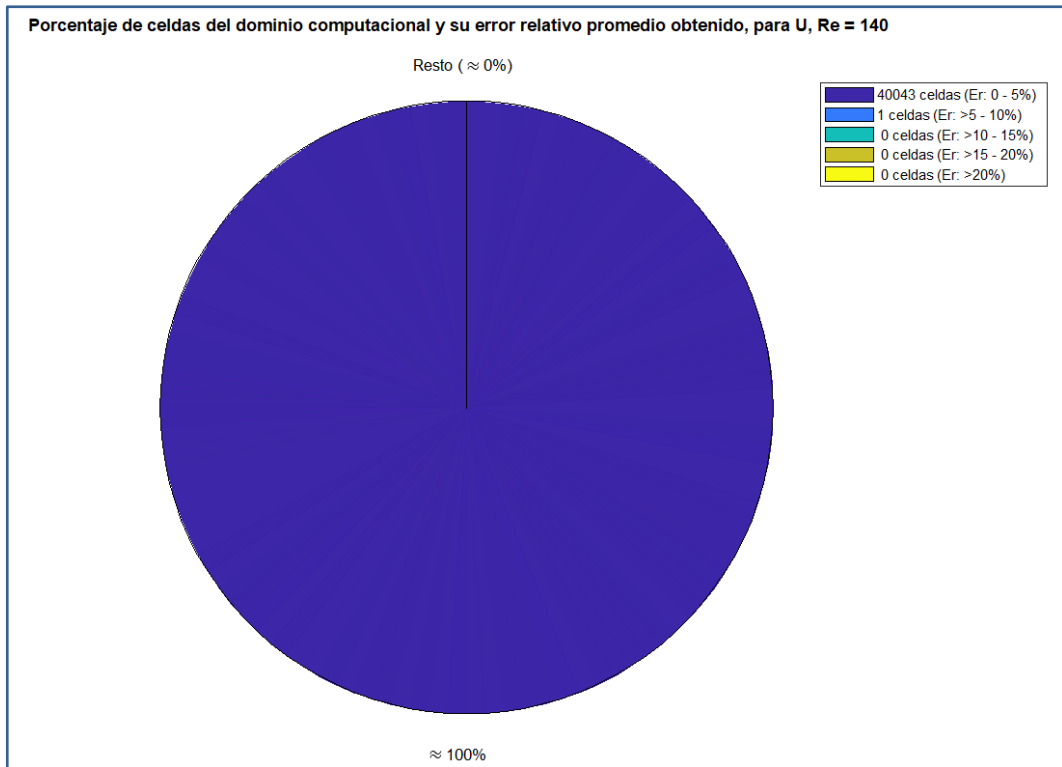
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rango de error relativo promedio. Para V, Re = 100.



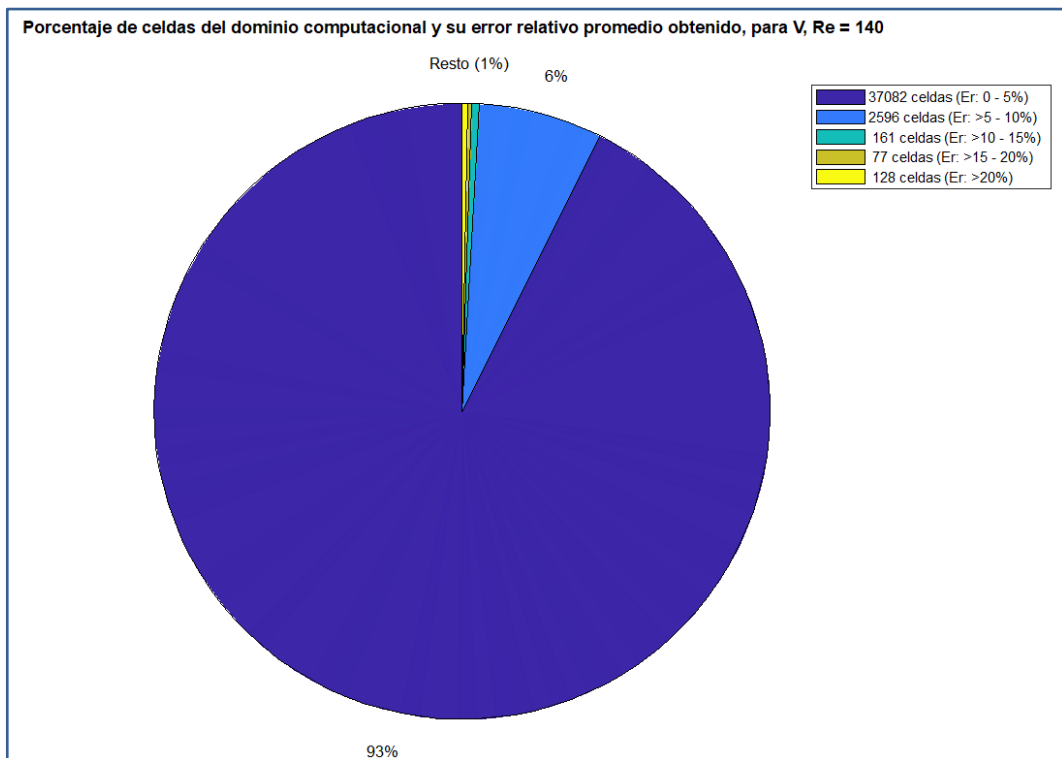
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rango de error relativo promedio. Para U, Re = 140.



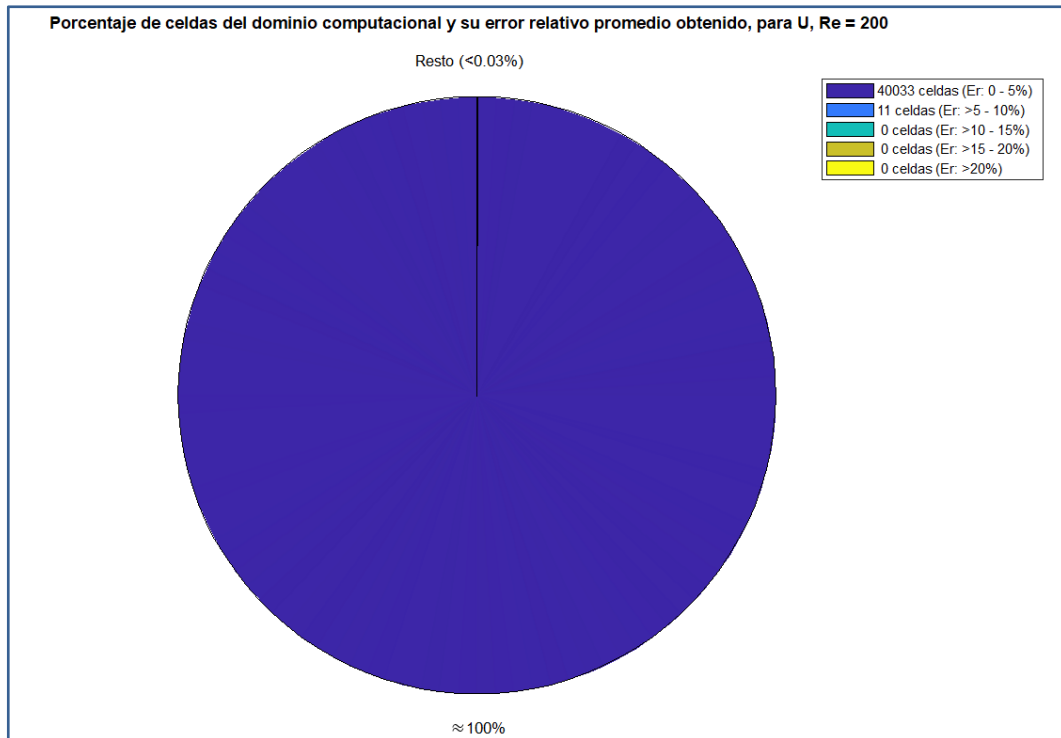
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rango de error relativo promedio. Para V, Re = 140.



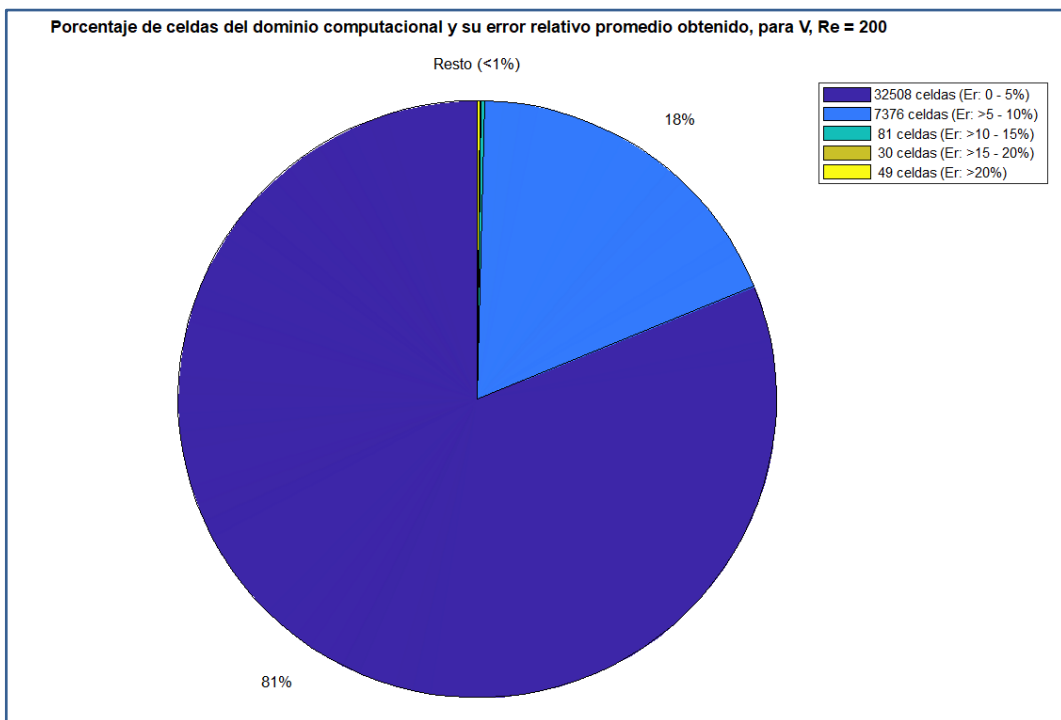
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rango de error relativo promedio. Para U, Re = 200.



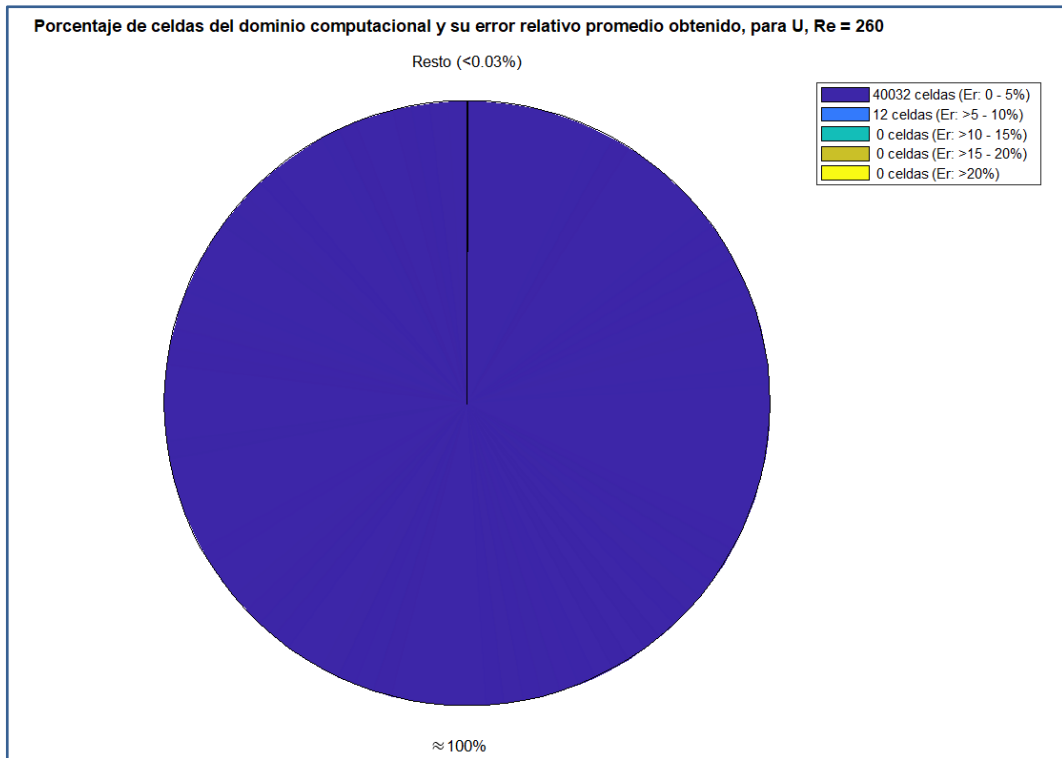
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rango de error relativo promedio. Para V, Re = 200.



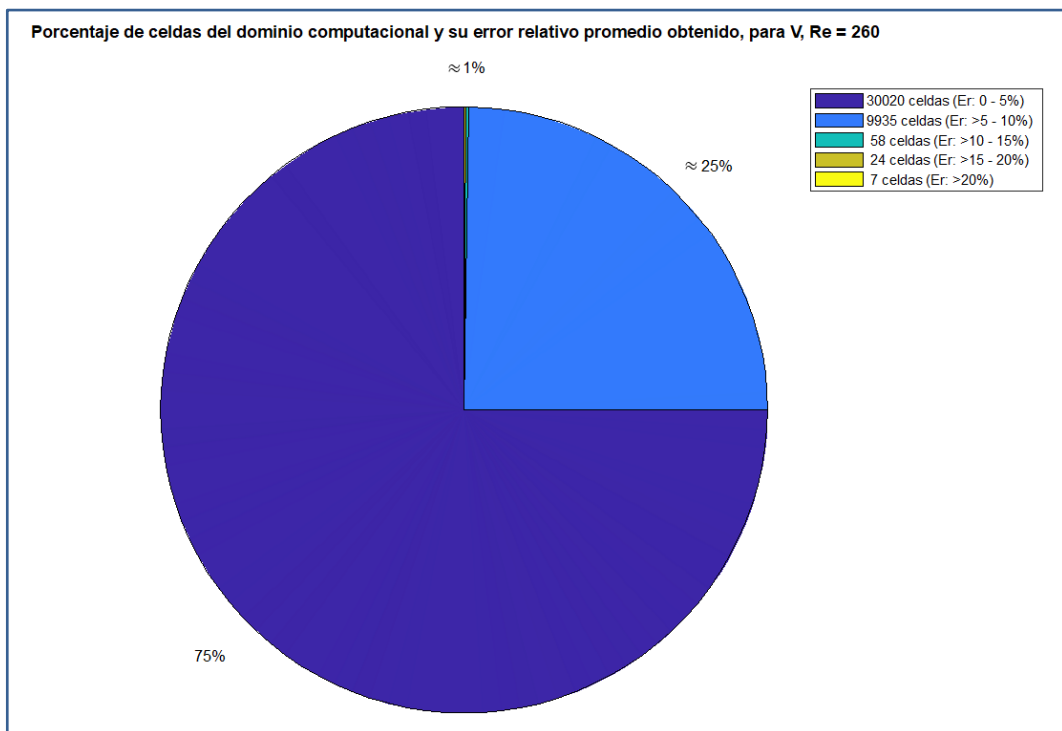
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rango de error relativo promedio. Para U, Re = 260.



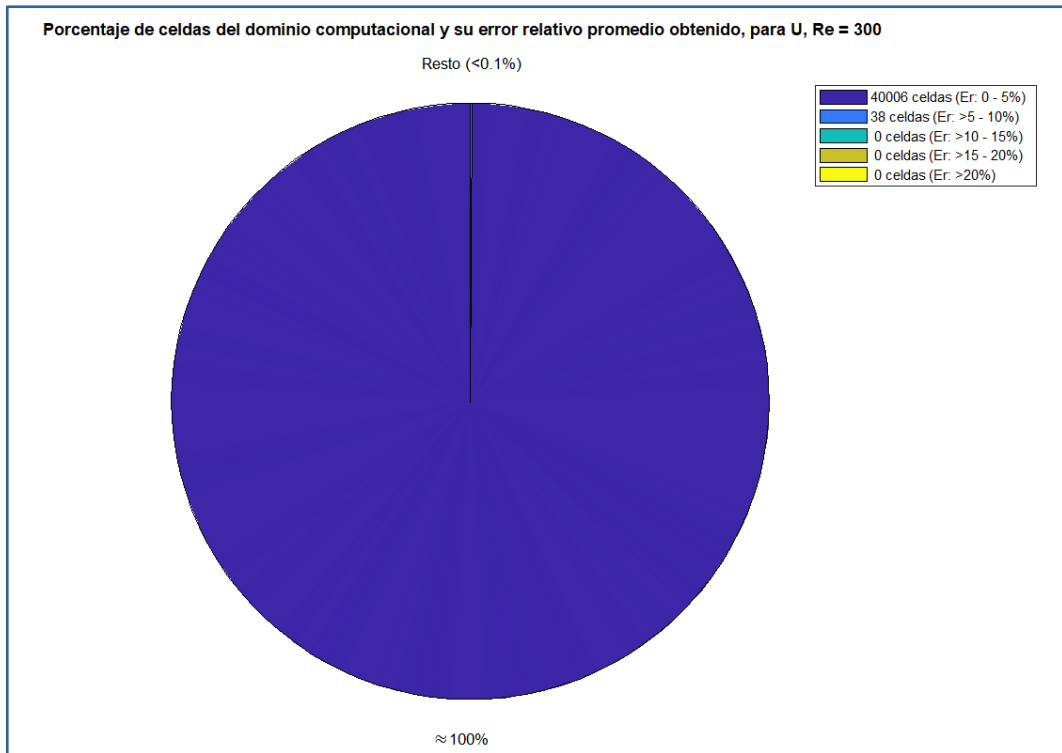
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rango de error relativo promedio. Para V, Re = 260.



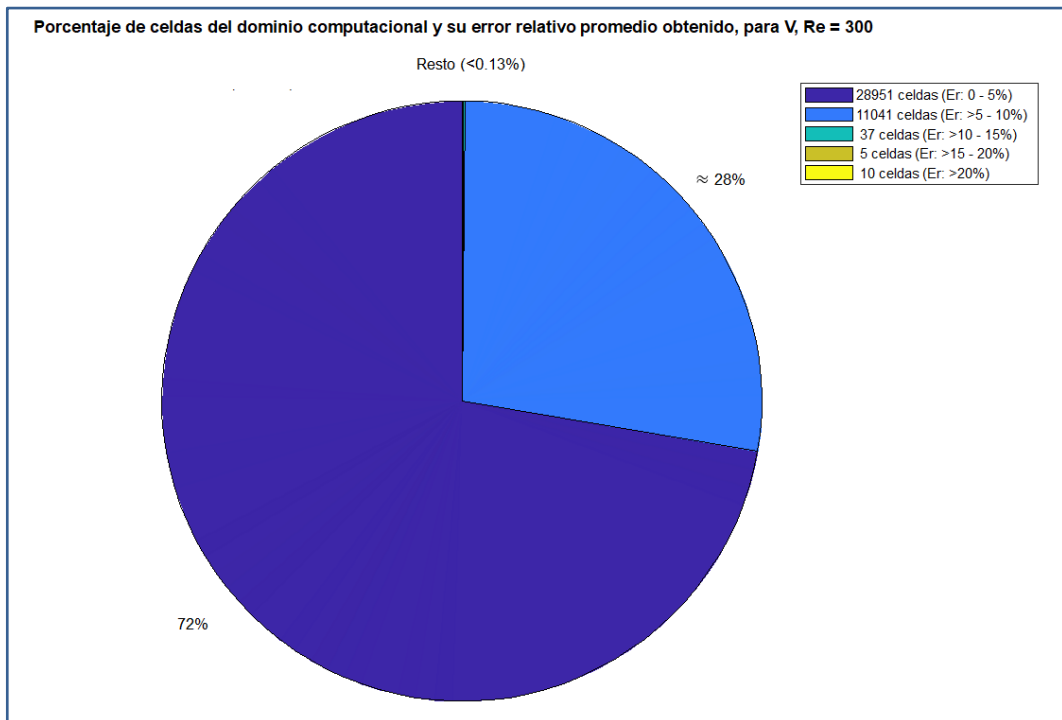
Fuente: Elaboración propia.

Porcentajes de celdas agrupadas por rango de error relativo promedio. Para U , $Re = 300$.



Fuente: Elaboración propia.

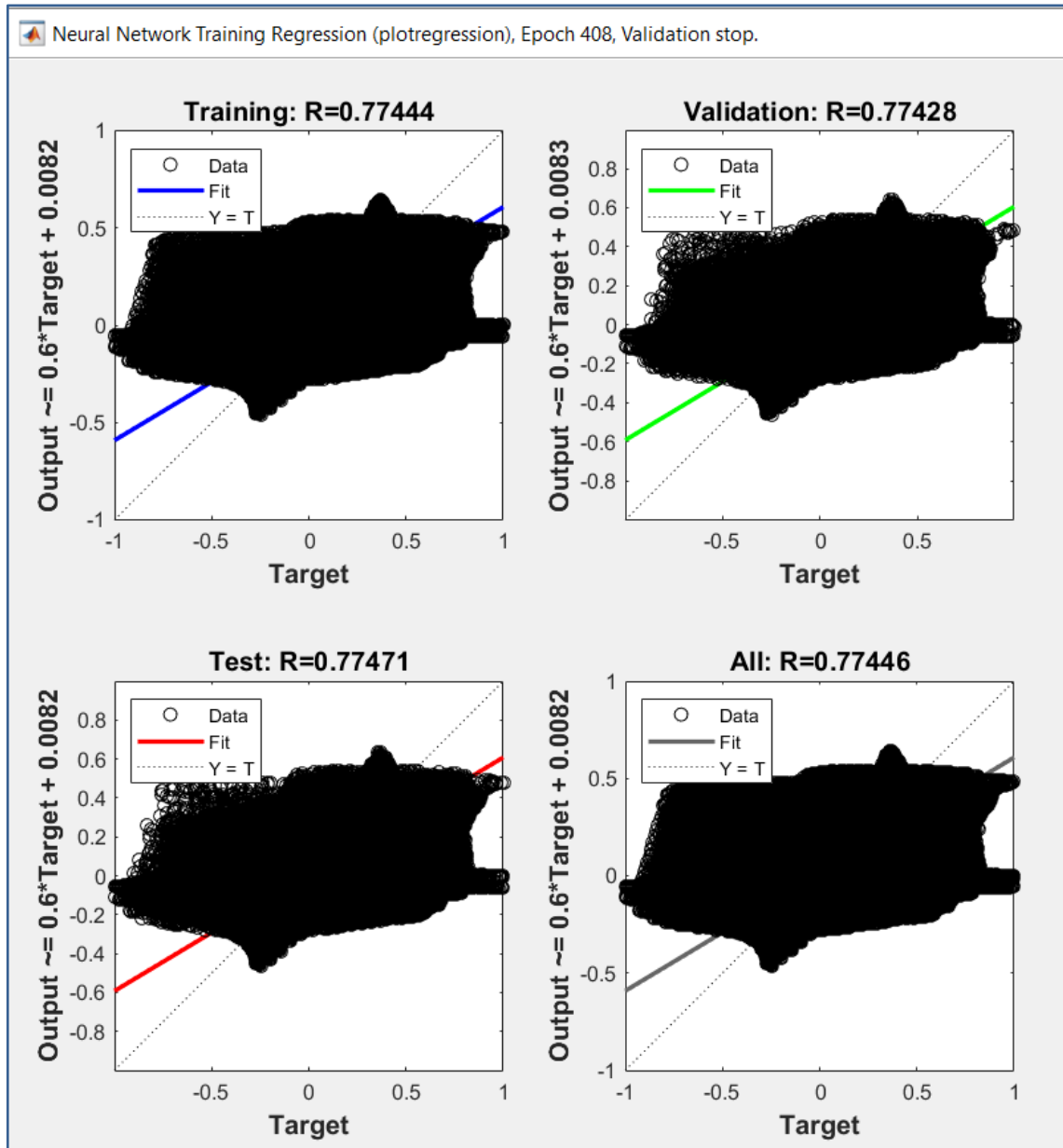
Porcentajes de celdas agrupadas por rango de error relativo promedio. Para V , $Re = 300$.



Fuente: Elaboración propia.

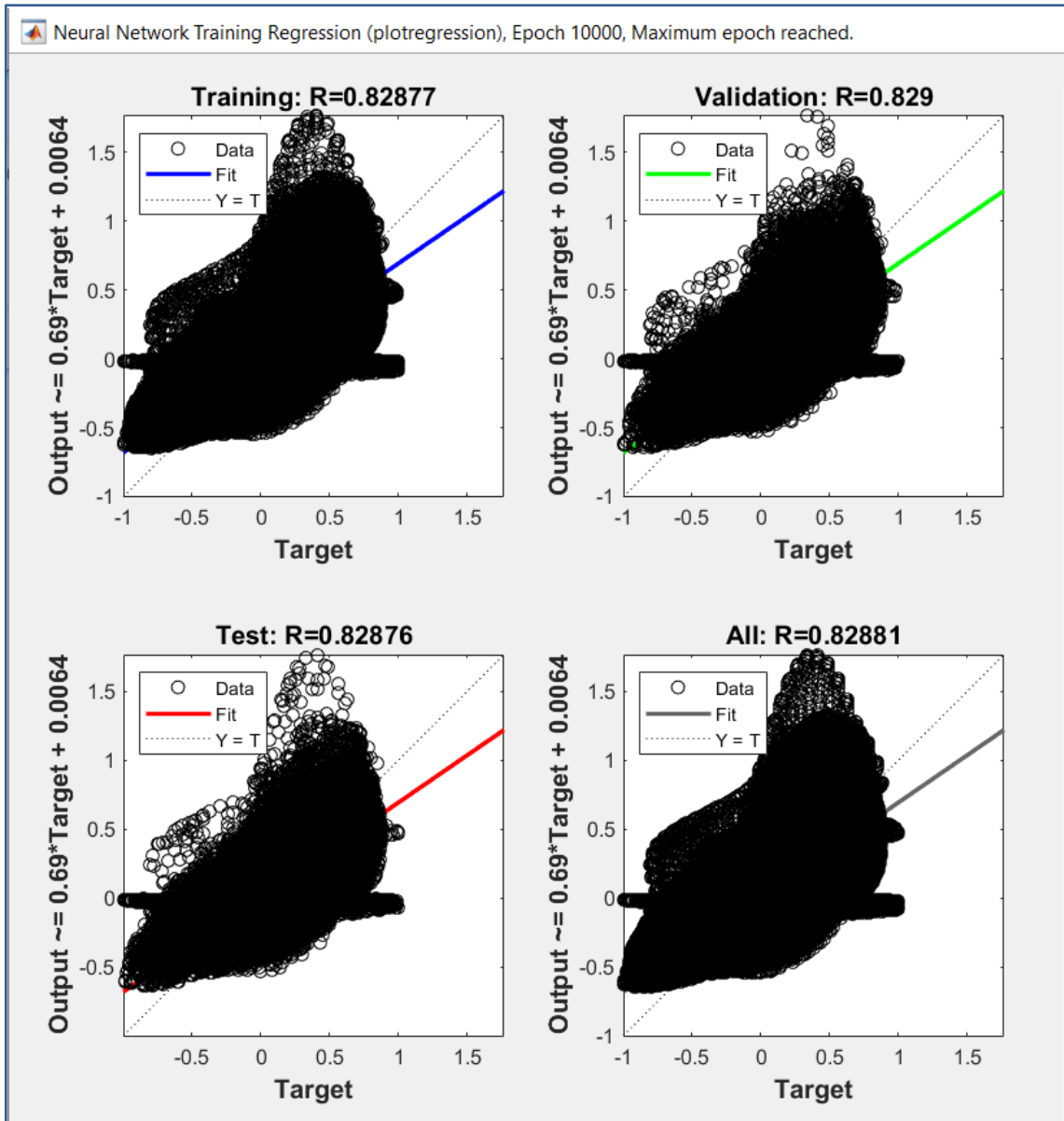
Anexo 2. Reportes gráficos de regresiones resultantes de la RNAs de 1 capa estudiadas. Vistas en el entorno de aplicación de RNA.

Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura simple de una capa y cinco neuronas.



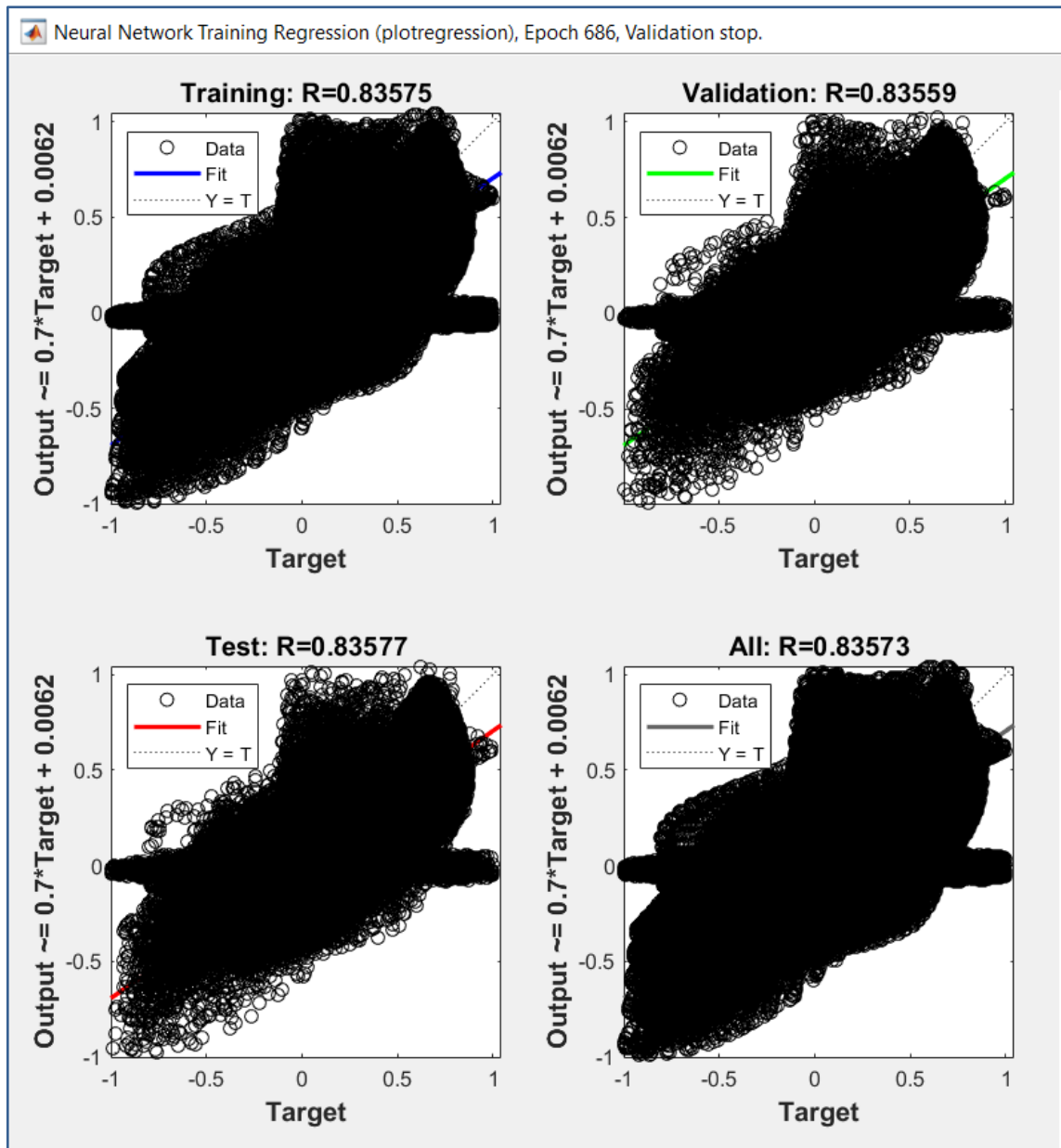
Fuente: Elaboración propia.

Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura simple de una capa y 10 neuronas.



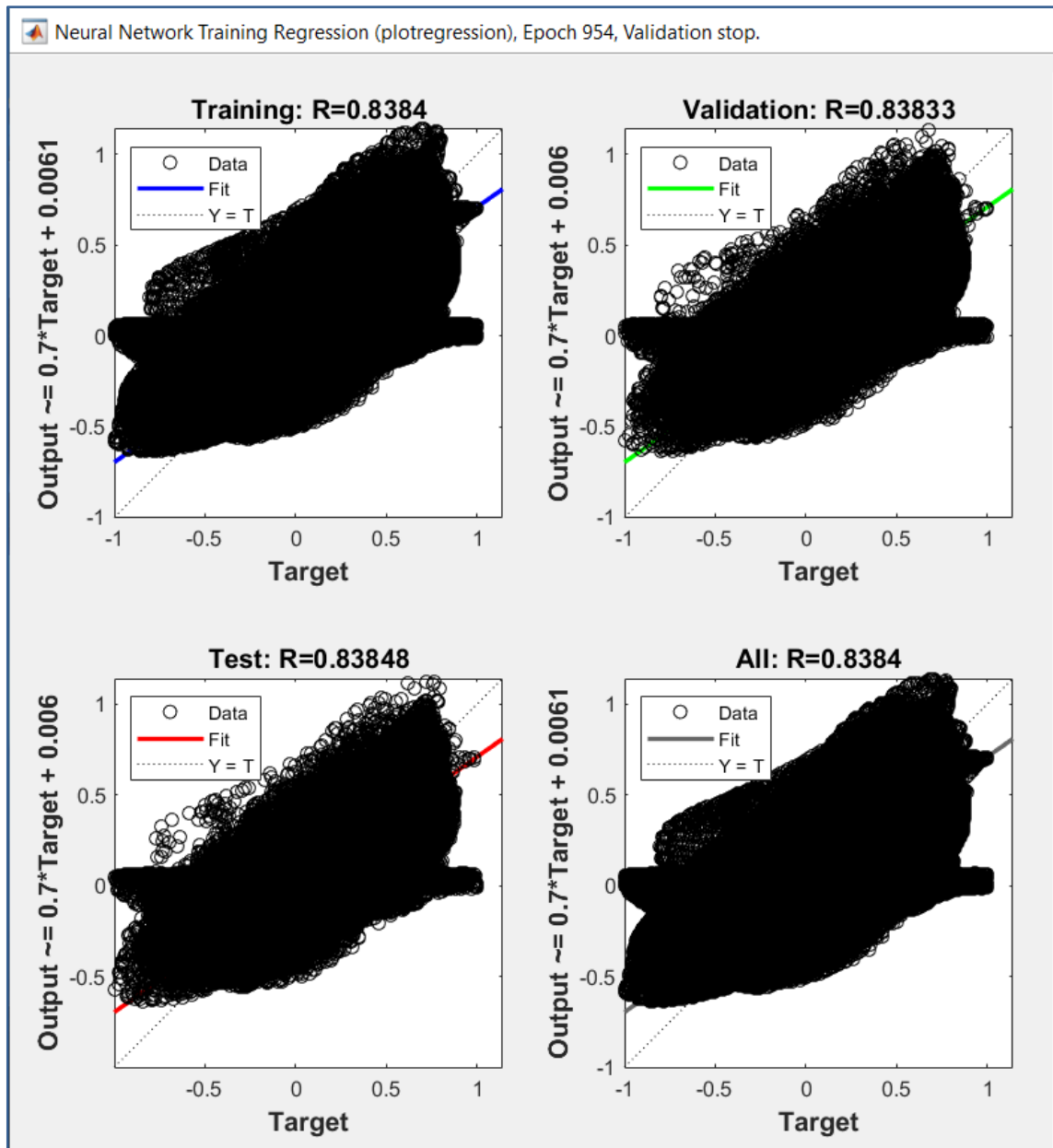
Fuente: Elaboración propia.

Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura simple de una capa y 15 neuronas.



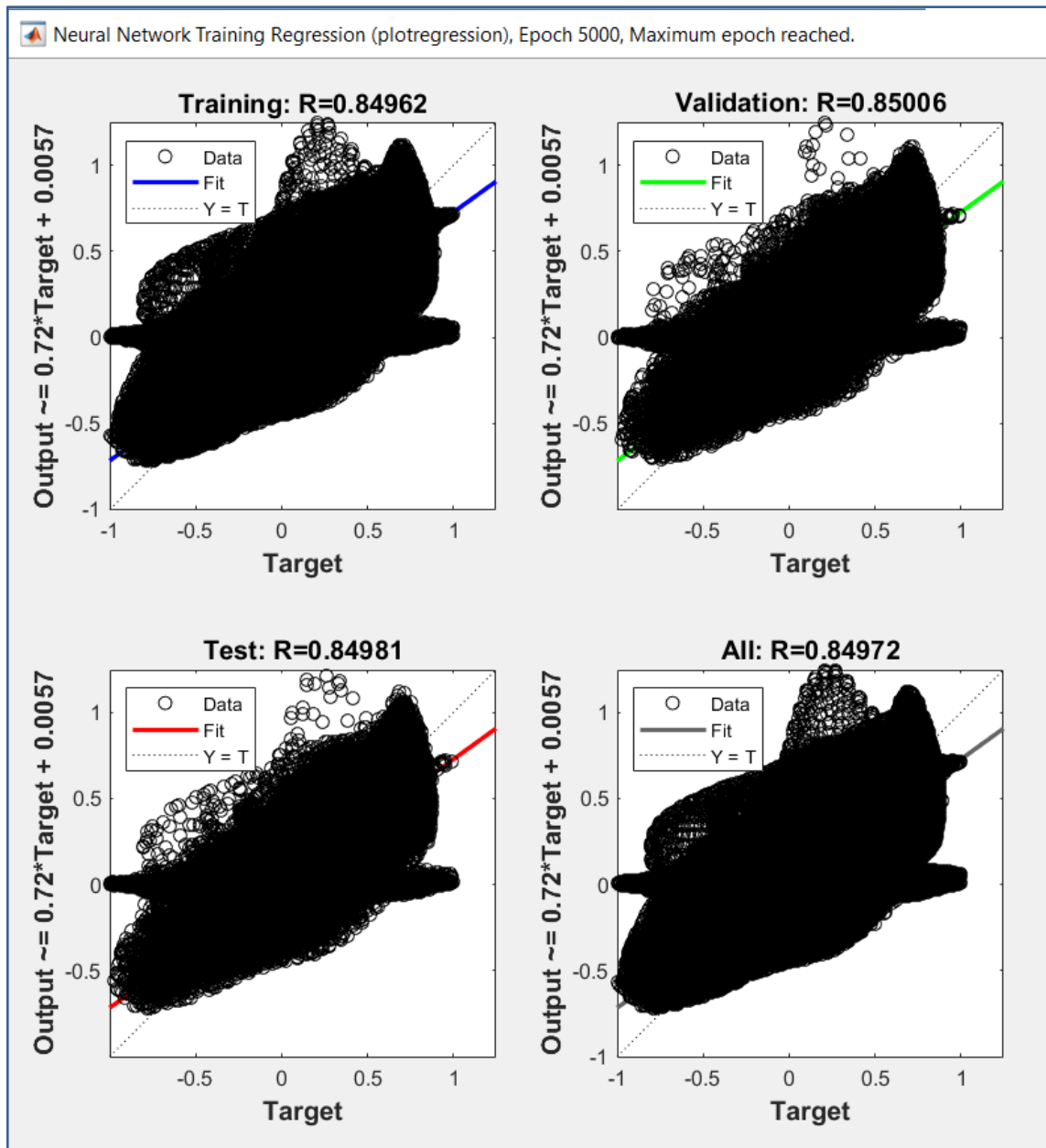
Fuente: Elaboración propia.

Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura simple de una capa y 20 neuronas.



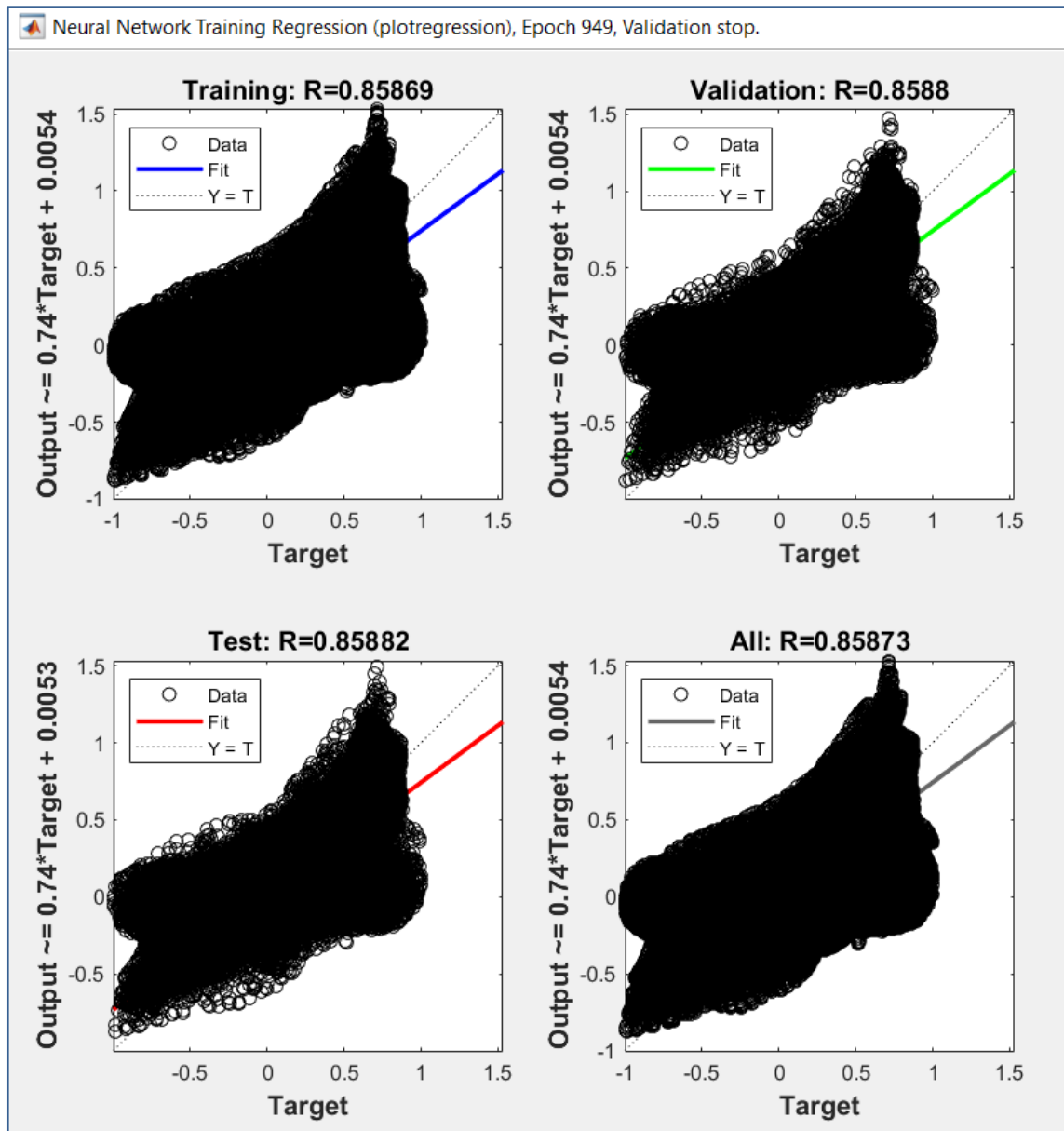
Fuente: Elaboración propia.

Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura simple de una capa y 25 neuronas.



Fuente: Elaboración propia.

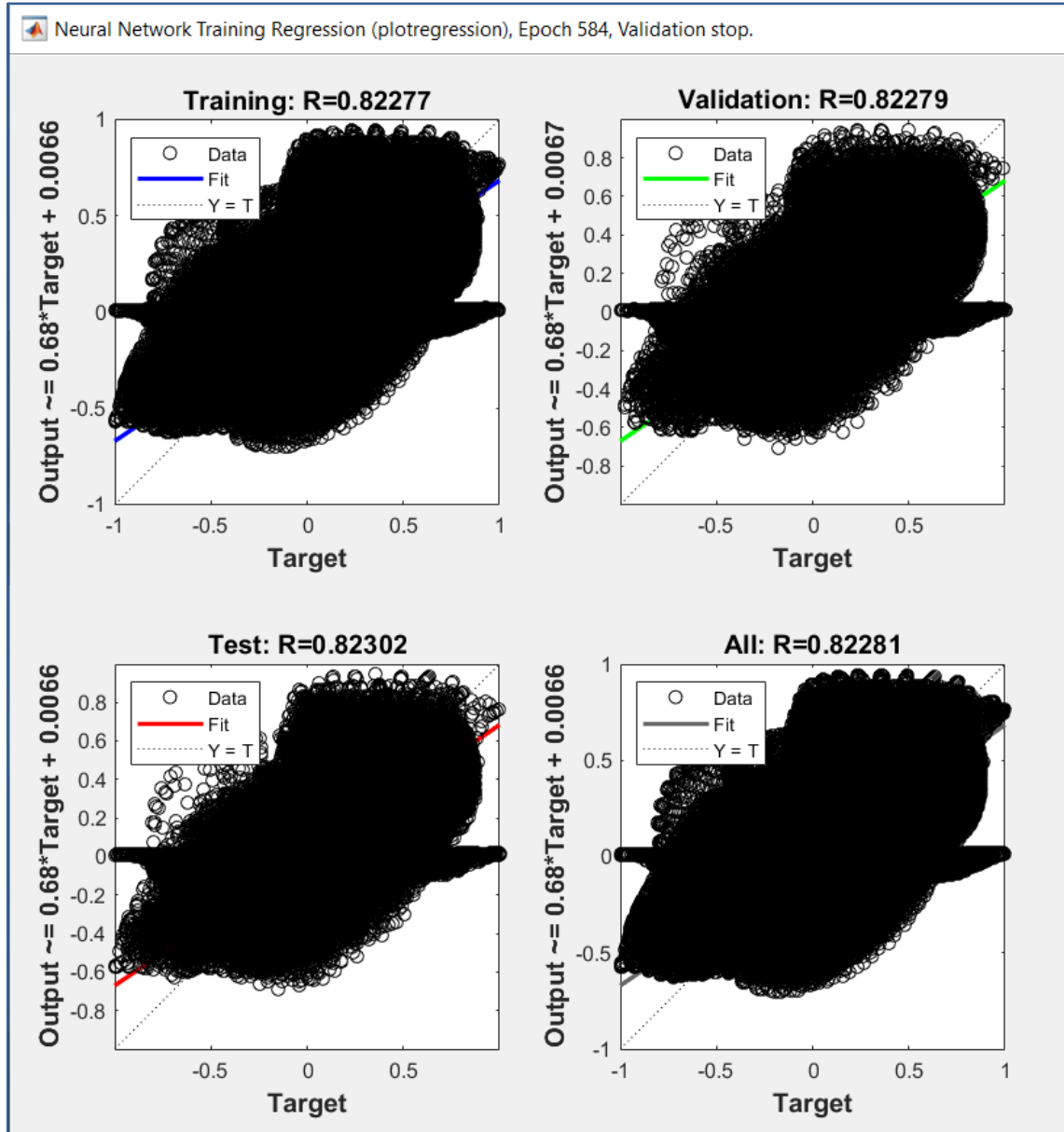
Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura simple de una capa y 30neuronas.



Fuente: Elaboración propia.

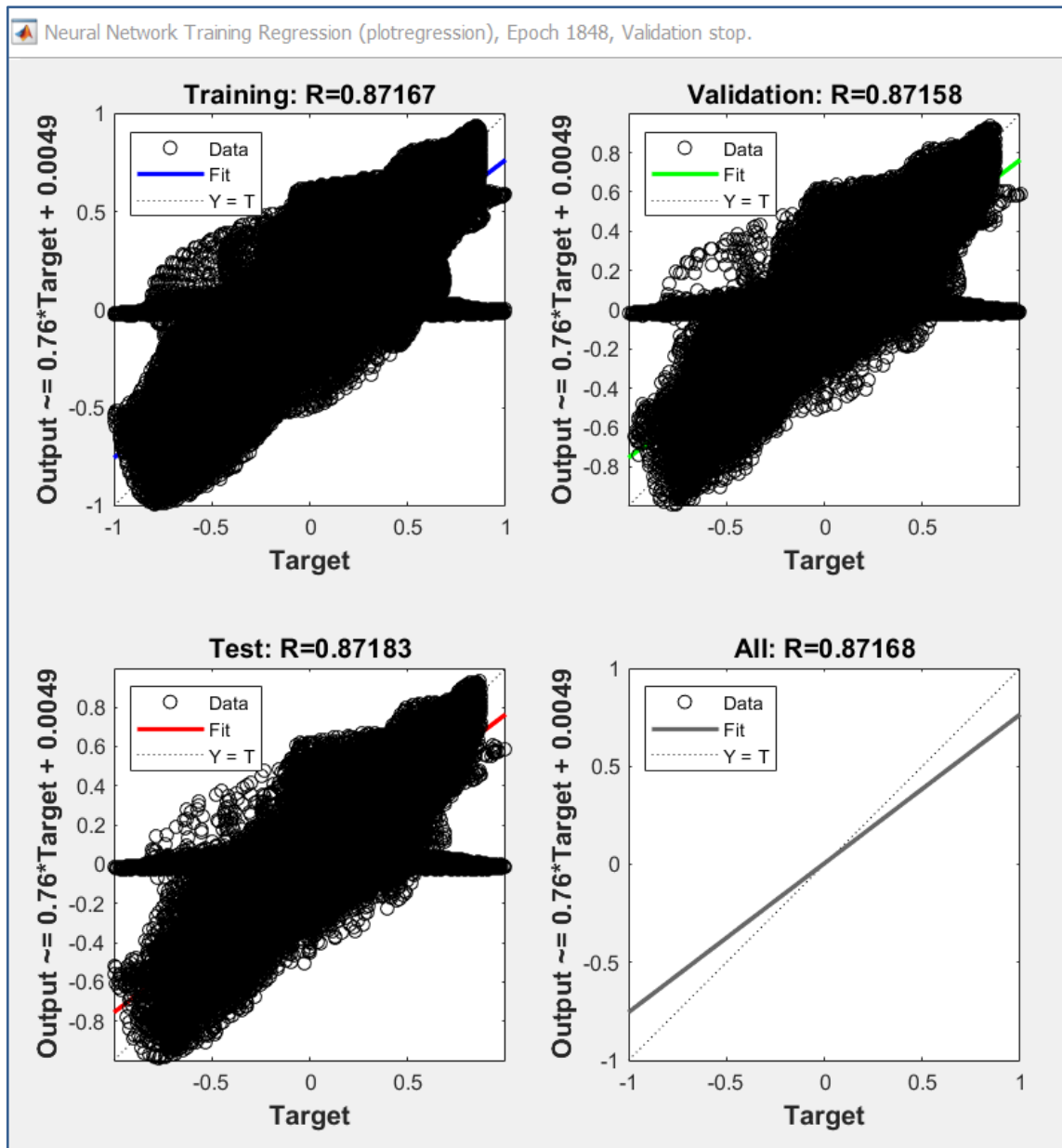
Anexo 3. Reportes gráficos de regresiones resultantes de la RNAs de 2 capas estudiadas. Vistas en el entorno de aplicación de RNA.

Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura de dos capas y cinco neuronas por capa.



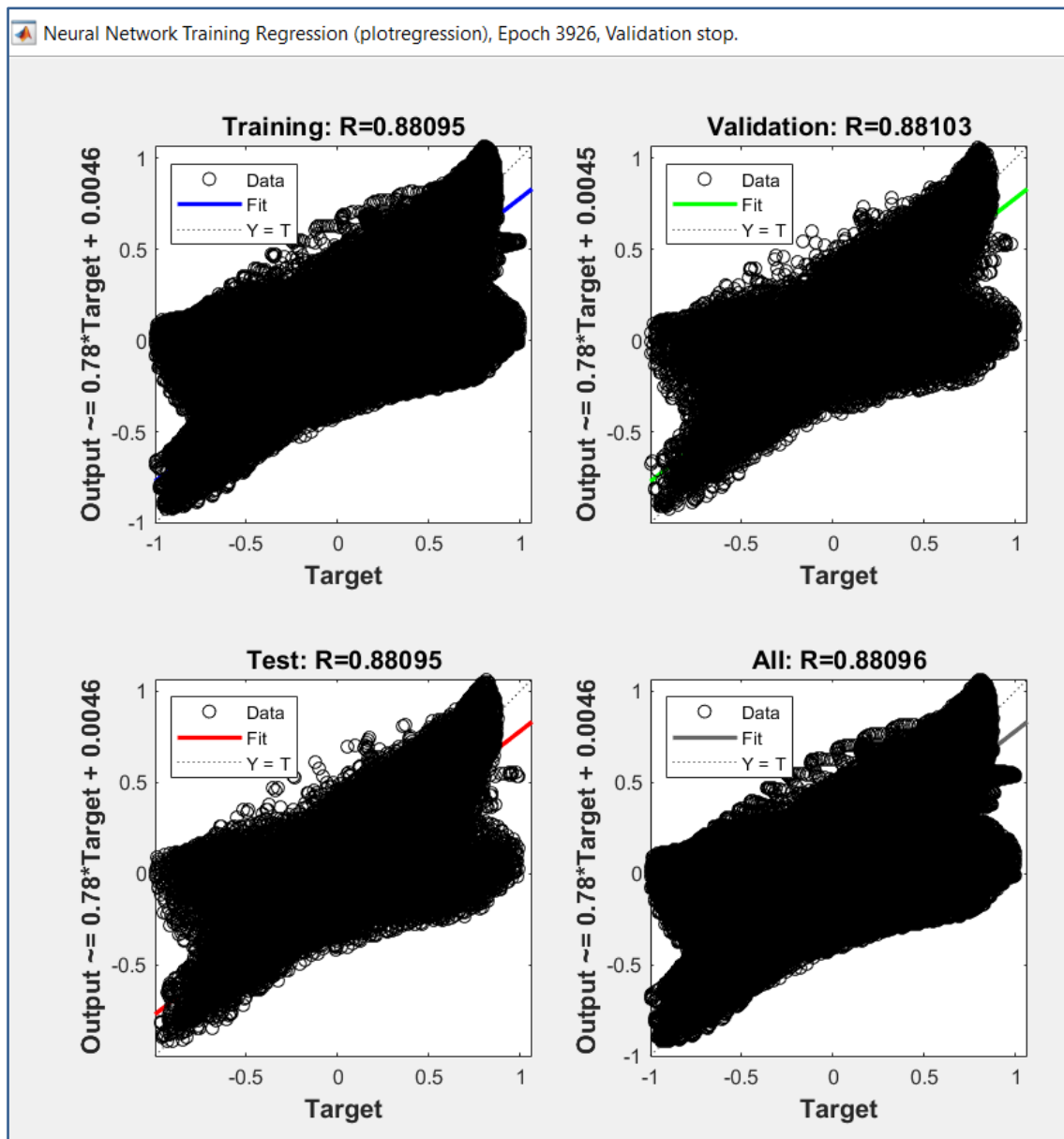
Fuente: Elaboración propia.

Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura de dos capas y 10 neuronas por capa.



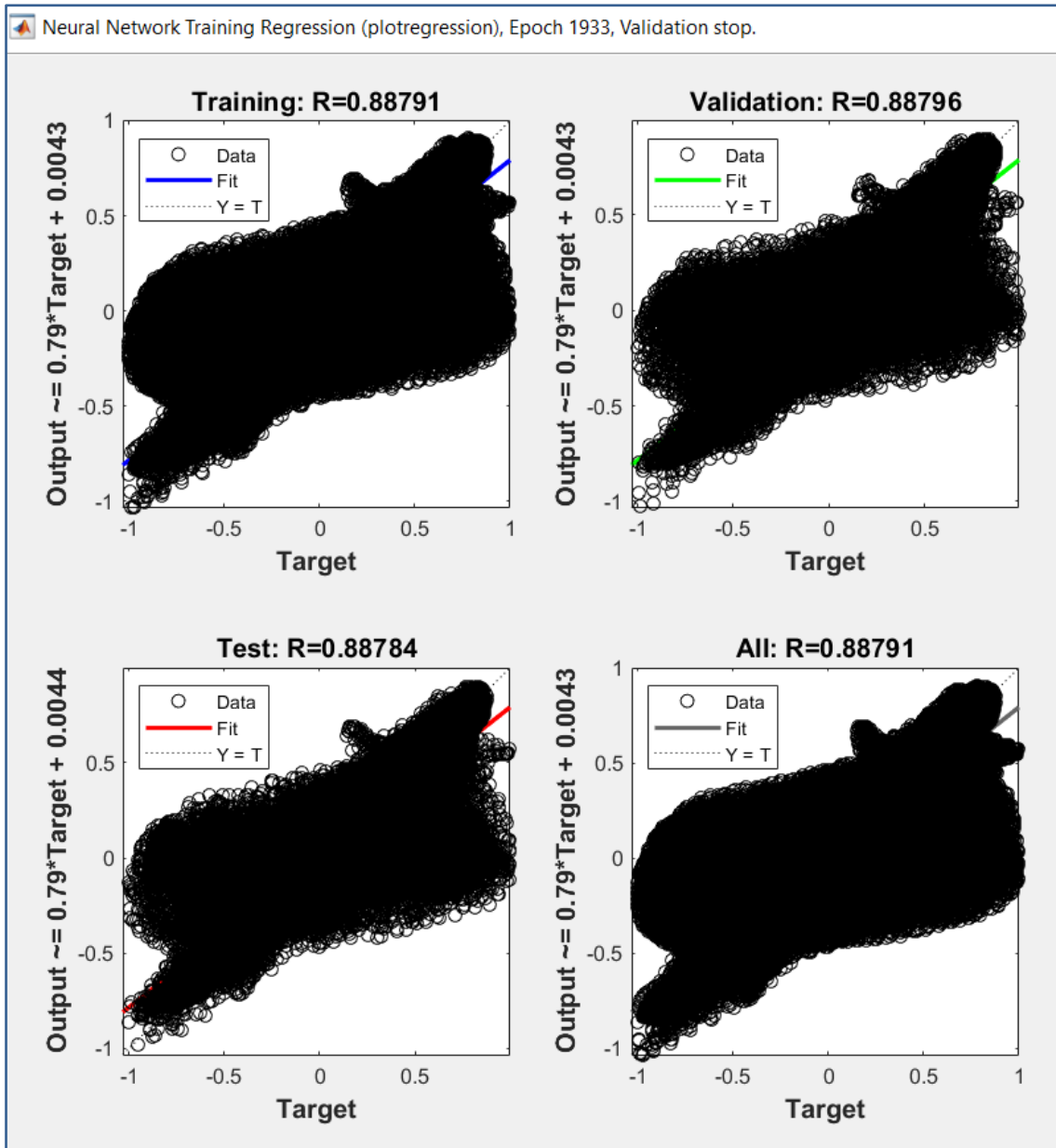
Fuente: Elaboración propia.

Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura de dos capas y 15 neuronas por capa.



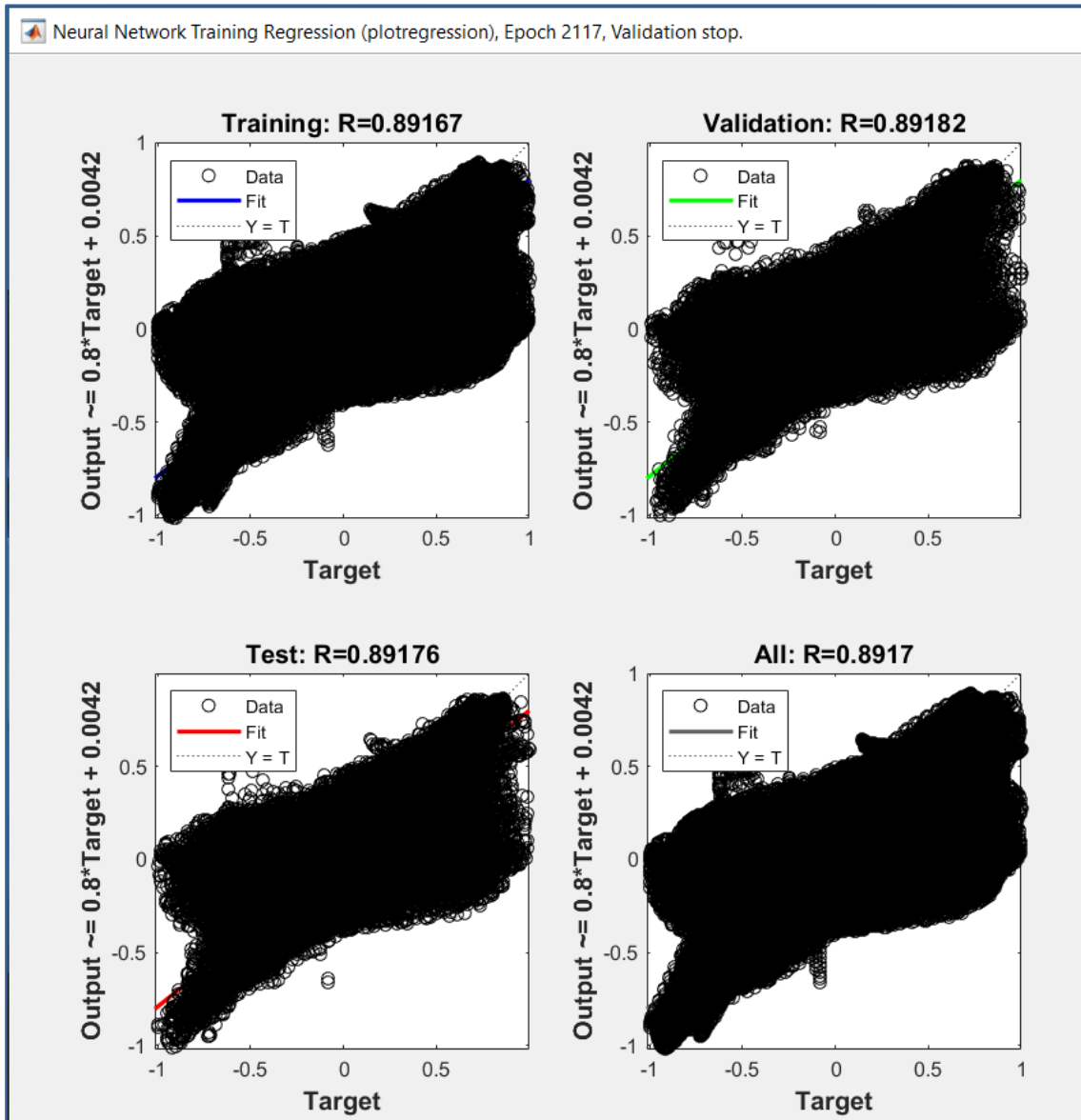
Fuente: Elaboración propia.

Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura de dos capas y 25 neuronas por capa.



Fuente: Elaboración propia.

Gráficos de regresión de entrenamiento validación y pruebas de RNA con arquitectura de dos capas y 30 neuronas por capa.



Fuente: Elaboración propia.

Anexo 4. Detalles de archivos de programación para la simulación OpenFOAM

Programación del dominio computacional, archivo "blockMeshDict"

```

1 |-----C++-----|
2 | |
3 | \\ // Field | OpenFOAM: The Open Source CFD Toolbox
4 | \\ // Operation | Version: 7
5 | \\ // Manipulation | Web: www.OpenFOAM.org
6 | |
7 |-----|
8 |FoamFile
9 | {
10 |     version      2.0;
11 |     format       ascii;
12 |     class        dictionary;
13 |     object       blockMeshDict;
14 | }
15 | // ***** //
16 |
17 |
18 |convertToMeters 1; // metros
19 |
20 |vertices
21 | (
22 |     // CIRCULO INTERNO BACK
23 |     (-0.05 0 -0.05) //Nodo 11 ----0
24 |     (-0.035355339 0.035355339 -0.05) //Nodo 16 ----1
25 |     (0 0.05 -0.05) //Nodo 10 ----2
26 |     (0.035355339 0.035355339 -0.05) //Nodo 5 ----3
27 |     (0.05 0 -0.05) //Nodo 0 ----4
28 |     (0.035355339 -0.035355339 -0.05) //Nodo 4 ----5
29 |     (0 -0.05 -0.05) //Nodo 45----6
30 |     (-0.035355339 -0.035355339 -0.05) //Nodo 48----7
31 |
32 |     // CIRCULO EXTERNO BACK
33 |     (-0.125 0 -0.05) //Nodo 12 ----8
34 |     (-0.0883883475 0.0883883475 -0.05) //Nodo 15 ----9
35 |     (0 0.125 -0.05) //Nodo 9 ----10
36 |     (0.0883883475 0.0883883475 -0.05) //Nodo 4 ----11
37 |     (0.125 0 -0.05) //Nodo 1 ----12
38 |     (0.0883883475 -0.0883883475 -0.05) //Nodo 39----13
39 |     (0 -0.125 -0.05) //Nodo 44----14
40 |     (-0.0883883475 -0.0883883475 -0.05) //Nodo 47----15
41 |
42 |     // DOMINIO CUADRADO BACK
43 |     (-2 0 -0.05) //Nodo 13 ----16
44 |     (-2 0.0883883475 -0.05) //Nodo 14 ----17
45 |
46 |     (-2 2 -0.05) //Nodo 17 ----18
47 |     (-0.0883883475 2 -0.05) //Nodo 18 ----19
48 |     (0 2 -0.05) //Nodo 8 ----20
49 |     (0.0883883475 2 -0.05) //Nodo 7 ----21
50 |     (3 2 -0.05) //Nodo ----22
51 |     (3 0.0883883475 -0.05) //Nodo 3 ----23
52 |     (3 0 -0.05) //Nodo 2 ----24
53 |     (3 -0.0883883475 -0.05) //Nodo 38----25
54 |     (3 -2 -0.05) //Nodo 41----26
55 |     (0.0883883475 -2 -0.05) //Nodo 42----27
56 |     (0 -2 -0.05) //Nodo 43----28
57 |     (-0.0883883475 -2 -0.05) //Nodo 50----29
58 |     (-2 -2 -0.05) //Nodo 49----30
59 |     (-2 -0.0883883475 -0.05) //Nodo 46----31
60 |
61 |     // CIRCULO INTERNO FRONT
62 |     (-0.05 0 0.05) //Nodo 30 ----32
63 |     (-0.035355339 0.035355339 0.05) //Nodo 35 ----33
64 |     (0 0.05 0.05) //Nodo 29 ----34
65 |     (0.035355339 0.035355339 0.05) //Nodo 24 ----35
66 |     (0.05 0 0.05) //Nodo 19 ----36
67 |     (0.035355339 -0.035355339 0.05) //Nodo 53----37
68 |     (0 -0.05 0.05) //Nodo 58----38
69 |     (-0.035355339 -0.035355339 0.05) //Nodo 61----39
70 |
71 |     // CIRCULO EXTERNO FRONT
72 |     (-0.125 0 0.05) //Nodo 31 ----40
73 |     (-0.0883883475 0.0883883475 0.05) //Nodo 34 ----41
74 |     (0 0.125 0.05) //Nodo 28 ----42
75 |     (0.0883883475 0.0883883475 0.05) //Nodo 23 ----43
76 |     (0.125 0 0.05) //Nodo 20 ----44
77 |     (0.0883883475 -0.0883883475 0.05) //Nodo 52----45
78 |     (0 -0.125 0.05) //Nodo 57----46
79 |     (-0.0883883475 -0.0883883475 0.05) //Nodo 60----47
80 |
81 |     // DOMINIO CUADRADO FRONT
82 |     (-2 0 0.05) //Nodo 32 ----48
83 |     (-2 0.0883883475 0.05) //Nodo 33 ----49
84 |     (-2 2 0.05) //Nodo 36 ----50
85 |     (-0.0883883475 2 0.05) //Nodo 37 ----51
86 |     (0 2 0.05) //Nodo 27 ----52
87 |
88 |     (-0.0883883475 2 0.05) //Nodo 37 ----51
89 |     (0 2 0.05) //Nodo 27 ----52
90 |     (0.0883883475 2 0.05) //Nodo 26 ----53
91 |     (3 2 0.05) //Nodo 25 ----54
92 |     (3 0.0883883475 0.05) //Nodo 22 ----55
93 |     (3 0 0.05) //Nodo 21 ----56
94 |     (3 -0.0883883475 0.05) //Nodo 51 ----57
95 |     (3 -2 0.05) //Nodo 54 ----58
96 |     (0.0883883475 -2 0.05) //Nodo 55 ----59
97 |     (0 -2 0.05) //Nodo 56 ----60
98 |     (-0.0883883475 -2 0.05) //Nodo 63 ----61
99 |     (-2 -2 0.05) //Nodo 62 ----62
100 |     (-2 -0.0883883475 0.05) //Nodo 59 ----63
101 |
102 |)
103 |blocks
104 | (
105 |     // PARTE SUPERIOR DEL CILINDRO
106 |     hex ( 1 9 8 0 33 41 40 32) ( 46 17 1) simpleGrading ( 66.26407608 1 1) // 0 ---- 6
107 |     hex ( 2 10 9 1 34 42 41 33) ( 46 17 1) simpleGrading ( 66.26407608 1 1) // 1 ---- 5
108 |     hex ( 3 11 10 2 35 43 42 34) (46 17 1) simpleGrading (66.26407608 1 1) // 2 ---- 0
109 |     hex ( 4 12 11 3 36 44 43 35) (46 17 1) simpleGrading ( 66.26407608 1 1) // 3 ---- 1
110 |
111 |     //PARTE INFERIOR DEL CILINDRO
112 |     hex ( 5 13 12 4 37 45 44 36) (46 17 1) simpleGrading (66.26407608 1 1) // 4 ----11
113 |     hex ( 6 14 13 5 38 46 45 37) (46 17 1) simpleGrading (66.26407608 1 1) // 5 ----10
114 |     hex ( 7 15 14 6 39 47 46 38) (46 17 1) simpleGrading (66.26407608 1 1) // 6 ----15
115 |     hex ( 8 15 7 32 40 47 39) (46 17 1) simpleGrading (66.26407608 1 1) // 7 ----16
116 |
117 |     // DOMINIO SUPERIOR
118 |     hex ( 9 17 16 8 41 49 48 40) ( 35 17 1) simpleGrading ( 25.54766986 1 1) // 8 ----7
119 |     hex ( 9 19 18 17 41 51 50 49) ( 35 35 1) simpleGrading ( 25.54766986 25.54766986 1) // 9 ----8
120 |     hex ( 10 20 19 9 42 52 51 41) ( 35 17 1) simpleGrading ( 25.54766986 1 1) // 10 ----9
121 |     hex ( 11 21 20 10 43 53 52 42) ( 35 17 1) simpleGrading ( 25.54766986 1 1) // 11 ----4
122 |
123 |     hex ( 11 23 22 21 43 55 54 53) (439 35 1) simpleGrading ( 1 25.54766986 1) // 12 ----3
124 |     hex ( 12 24 23 11 44 56 55 43 ) (439 17 1) simpleGrading ( 1 1 1) // 13----2
125 |

```

Predicción de campos de velocidad alrededor de un cilindro mediante Redes Neuronales Artificiales

```

131 hex (13 25 24 12 45 57 56 44) (439 17 1) simpleGrading (1 1 1) // 14 ----12
132 hex (13 27 26 25 45 59 58 57) (35 439 1) simpleGrading (25.54766986 1 1) // 15 ----13
133
134 hex (14 28 27 13 46 60 59 45) (35 17 1) simpleGrading (25.54766986 1 1) // 16 ----14
135 hex (15 29 28 14 47 61 60 46) (35 17 1) simpleGrading (25.54766986 1 1) // 17 ----19
136 hex (15 31 30 29 47 63 62 61) (35 35 1) simpleGrading (25.54766986 25.54766986 1) // 18 ----18
137 hex (8 16 31 15 40 48 63 47) (35 17 1) simpleGrading (25.54766986 1 1) // 19 ----17
138 );
139
140 edges
141 (
142
143 //ARCOS POSTERIOR INTERNOS
144 arc 0 1 (0 0.05 -0.05) // 0
145 arc 1 2 ( 0.035355339 0.035355339 -0.05) // 1
146
147 arc 2 3 ( 0.05 0 -0.05) // 2
148 arc 3 4 ( 0.035355339 -0.035355339 -0.05) // 3
149
150 arc 4 5 (0 -0.05 -0.05) // 4
151 arc 5 6 (-0.035355339 -0.035355339 -0.05) // 5
152 arc 6 7 (-0.05 0 -0.05) // 6
153 arc 7 0 (-0.035355339 0.035355339 -0.05) // 7
154
155 //ARCOS POSTERIOR EXTERNOS
156 arc 8 9 (0 0.125 0.05) // 8
157 arc 9 10 ( 0.0883883475 0.0883883475 -0.05) // 9
158 arc 10 11 ( 0.125 0 -0.05) // 10
159 arc 11 12 ( 0.0883883475 -0.0883883475 -0.05) // 11
160
161 arc 12 13 (0 -0.125 -0.05) // 12
162 arc 13 14 ( -0.0883883475 -0.0883883475 -0.05) // 13
163 arc 14 15 (-0.125 0 -0.05) // 14
164 arc 15 8 (-0.0883883475 0.0883883475 -0.05) // 15
165
166 //ARCOS FRONTAL INTERNOS
167 arc 32 33 (0 0.05 0.05) // 16
168 arc 33 34 ( 0.035355339 0.035355339 0.05) // 17
169 arc 34 35 ( 0.05 0 0.05) // 18
170 arc 35 36 ( 0.035355339 -0.035355339 0.05) // 19
171
172 arc 36 37 ( 0 -0.05 0.05) // 20
173 arc 37 38 (-0.035355339 -0.035355339 0.05) // 21
174 arc 38 39 (-0.05 0 0.05) // 22

```

```

177 //ARCOS FRONTAL EXTERNOS
178 arc 40 41 (0 0.125 0.05) // 24
179 arc 41 42 ( 0.0883883475 0.0883883475 0.05) // 25
180 arc 42 43 ( 0.125 0 0.05) // 26
181 arc 43 44 ( 0.0883883475 -0.0883883475 0.05) // 27
182
183 arc 44 45 (0 -0.125 0.05) // 28
184 arc 45 46 (-0.0883883475 -0.0883883475 0.05) // 29
185 arc 46 47 (-0.125 0 0.05) // 30
186 arc 47 40 (-0.0883883475 0.0883883475 0.05) // 31
187
188 );
189
190 boundary
191 (
192
193 top
194 (
195 type symmetryPlane;
196 faces
197 (
198 ( 51 19 18 50)
199 ( 52 20 19 51)
200 ( 53 21 20 52)
201 ( 54 22 21 53)
202 );
203 );
204
205 bottom
206 (
207 type symmetryPlane;
208 faces
209 (
210 (29 61 62 30)
211 (28 60 61 29)
212 (27 59 60 28)
213 (26 58 59 27)
214 );
215 );
216
217 );
218
219 );
220

```

```

288 (14 13 27 28) //16
289 (15 14 28 29) //17
290 (15 29 30 31) //18
291 (8 15 31 16) //19
292
293 );
294
295 front
296 (
297 type empty;
298
299 faces
300 (
301 //FRONT
302 (32 33 41 40) //0
303 (33 34 42 41) //1
304 (34 35 43 42) //2
305 (35 36 44 43) //3
306 (36 37 45 44) //4
307 (37 38 46 45) //5
308 (38 39 47 46) //6
309 (39 32 40 47) //7
310 (40 41 49 48) //8
311 (41 51 50 49) //9
312 (41 42 52 51) //10
313 (42 43 53 52) //11
314 (43 55 54 53) //12
315 (43 44 56 55) //13
316 (44 45 57 56) //14
317 (45 59 58 57) //15
318 (45 46 60 59) //16
319 (46 47 61 60) //17
320 (47 63 62 61) //18
321 (47 40 48 63) //19
322 );
323
324 );
325
326 );
327
328 );
329
330 // *****
331

```

Archivo "ControlDic" configurado en OpenFOAM

```

1 |----- C++ -----|
2 |=====|
3 | \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / / O p e r a t i o n | Version: 7 |
5 | \ \ / / A n d | Web: www.OpenFOAM.org |
6 | \ \ / / M a n i p u l a t i o n | |
7 |-----|
8 |FoamFile|
9 |{
10 |  version      2.0;
11 |  format       ascii;
12 |  class        dictionary;
13 |  object       controlDict;
14 |}
15 |
16 |// * * * * *
17 |application    icoFoam;
18 |startFrom      latestTime;
19 |startTime      0;
20 |stopAt         endTime;
21 |endTime        5;
22 |deltaT         0.0001;
23 |writeControl   adjustableRunTime;
24 |writeInterval  0.1;
25 |purgeWrite     0;
26 |writeFormat    ascii;
27 |writePrecision 6;
28 |writeCompression  uncompressed;
29 |timeFormat     general;
30 |runTimeModifiable true;
31 |adjustTimeStep off;
32 |maxCo          0.5;
33 |functions
34 |{
35 |  #includeFunc residuals
36 |  #includeFunc Q
37 |  // #includeFunc yPlus
38 |  #include "force/forceCoeffs"
39 |}
40 |// * * * * *

```

Archivo "decomposeParDict" configurado en OpenFOAM

```

3 |----- C++ -----|
4 |=====|
5 | \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
6 | \ \ / / O p e r a t i o n | Version: 7 |
7 | \ \ / / A n d | Web: www.OpenFOAM.org |
8 | \ \ / / M a n i p u l a t i o n | |
9 |-----|
10 |FoamFile|
11 |{
12 |  version      2.0;
13 |  format       ascii;
14 |  class        dictionary;
15 |  location     "system";
16 |  object       decomposeParDict;
17 |}
18 |// * * * * *
19 |numberOfSubdomains 3;
20 |method             simple;
21 |simpleCoeffs
22 |{
23 |  n          (3 1 1);
24 |  delta      0.001;
25 |}
26 |hierarchicalCoeffs
27 |{
28 |  n          (1 1 1);
29 |  delta      0.001;
30 |  order      xyz;
31 |}
32 |manualCoeffs
33 |{
34 |  dataFile   "";
35 |}
36 |distributed  no;
37 |roots       ();
38 |}

```

Archivo "fvSchemes" configurado en OpenFOAM

```

1 |----- C++ -----|
2 |=====|
3 | \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / / O p e r a t i o n | Website: https://openfoam.org |
5 | \ \ / / A n d | Version: 7 |
6 | \ \ / / M a n i p u l a t i o n | |
7 |-----|
8 |FoamFile|
9 |{
10 |  version      2.0;
11 |  format       ascii;
12 |  class        dictionary;
13 |  location     "system";
14 |  object       fvSchemes;
15 |}
16 |dtSchemes
17 |{
18 |  default Euler; // //1st order, stable, diffusive
19 |}
20 |gradSchemes
21 |{
22 |  default Gauss linear;
23 |  grad(p) Gauss linear;
24 |  grad(U) Gauss linear;
25 |}
26 |divSchemes
27 |{
28 |  default none;
29 |}
30 |laplacianSchemes
31 |{
32 |  default Gauss linear limited 1; // tenemos < 45 de no ortogonalidad
33 |}
34 |interpolationSchemes
35 |{
36 |  default linear;
37 |}
38 |snGradSchemes
39 |{
40 |  default limited 1;
41 |}
42 |fluxRequired
43 |{
44 |  default no;

```


Archivo "g" configurado en OpenFOAM

```
1 |----- C++ -----|
2 | ***** |
3 | \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / / O p e r a t i o n | Version: 7 |
5 | \ \ / / A n d | Web: www.OpenFOAM.org |
6 | \ \ / / M a n i p u l a t i o n | |
7 |-----|
8 | FoamFile |
9 | { |
10 |     version      2.0; |
11 |     format       ascii; |
12 |     class        uniformDimensionedVectorField; |
13 |     location     "constant"; |
14 |     object       g; |
15 | } |
16 | // ***** // |
17 | |
18 | dimensions      [0 1 -2 0 0 0]; |
19 | value           (0 9.81 0); |
20 | |
21 | // ***** // |
22 | |
```

Archivo "transportProperties" configurado en OpenFOAM

```
1 |----- C++ -----|
2 | ***** |
3 | \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / / O p e r a t i o n | Version: 7 |
5 | \ \ / / A n d | Web: www.OpenFOAM.org |
6 | \ \ / / M a n i p u l a t i o n | |
7 |-----|
8 | FoamFile |
9 | { |
10 |     version      2.0; |
11 |     format       ascii; |
12 |     class        dictionary; |
13 |     location     "constant"; |
14 |     object       transportProperties; |
15 | } |
16 | // ***** // |
17 | |
18 | transportModel  Newtonian; |
19 | nu              [0 2 -1 0 0 0] 0.002564103; //nu del agua a -T =16 - grados centigrados |
20 | |
21 | // ***** // |
22 | |
```

Archivo "turbulenceProperties" configurado en OpenFOAM

```
1 |----- C++ -----|
2 | ***** |
3 | \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / / O p e r a t i o n | Version: 7 |
5 | \ \ / / A n d | Web: www.OpenFOAM.org |
6 | \ \ / / M a n i p u l a t i o n | |
7 |-----|
8 | FoamFile |
9 | { |
10 |     version      2.0; |
11 |     format       ascii; |
12 |     class        dictionary; |
13 |     object       turbulenceProperties; |
14 | } |
15 | // ***** // |
16 | |
17 | |
18 | simulationType laminar; |
19 | |
20 | // ***** // |
21 | |
```

Archivo "P" configurado en OpenFOAM

```
1 |----- C++ -----|
2 | ***** |
3 | \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / / O p e r a t i o n | Version: 7 |
5 | \ \ / / A n d | Web: www.OpenFOAM.org |
6 | \ \ / / M a n i p u l a t i o n | |
7 |-----|
8 | FoamFile |
9 | { |
10 |     version      2.0; |
11 |     format       ascii; |
12 |     class        volScalarField; |
13 |     object       p; |
14 | } |
15 | // ***** // |
16 | |
17 | #include "include/initialConditions" |
18 | dimensions      [0 2 -2 0 0 0]; |
19 | internalField    uniform $Pressure; |
20 | boundaryField |
21 | { |
22 |     // - Set patchGroups for constraint patches |
23 |     #includeEtc "caseDicts/setConstraintTypes" |
24 |     inlet |
25 |     { |
26 |         type      zeroGradient; |
27 |     } |
28 |     outlet |
29 |     { |
30 |         type      fixedValue; |
31 |         value     $internalField; |
32 |     } |
33 |     cylinder |
34 |     { |
35 |         type      zeroGradient; |
36 |     } |
37 | #include "include/fab" |
38 | } |
39 | |
40 | // ***** // |
41 | |
```

Archivo "U" configurado en OpenFOAM

```
1 |----- C++ -----|
2 | ***** |
3 | \ \ / Field | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / Operation | Version: 7 |
5 | \ \ / And | Web: www.OpenFOAM.org |
6 | \ \ / Manipulation |
7 |-----|
8 | FoamFile
9 | {
10 |     version      2.0;
11 |     format        ascii;
12 |     class         volVectorField;
13 |     location      "0";
14 |     object        U;
15 | }
16 | ***** //
17 | #include "include/initialConditions"
18 | dimensions      [0 1 -1 0 0 0];
19 | internalField   uniform $Velocity;
20 | boundaryField
21 | {
22 |     //- Set patchGroups for constraint patches
23 |     #includeEtc "caseDicts/setConstraintTypes"
24 |     inlet
25 |     {
26 |         type      fixedValue;
27 |         value      $internalField;
28 |     }
29 |     outlet
30 |     {
31 |         type      zeroGradient;
32 |     }
33 |     cylinder
34 |     {
35 |         type      noSlip;
36 |     }
37 |     #include     "include/fab"
38 | }
39 | ***** //
```