



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA MECÁNICA

## DETECCIÓN DE FALLAS EN EQUIPOS TÉRMICOS A PARTIR DE TRANSFORMADORES Y MECANISMOS DE ATENCIÓN

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO

DAVID IGNACIO LIZANA PARDO

PROFESOR GUÍA:

ENRIQUE ANDRÉS LÓPEZ DROGUETT

MIEMBROS DE LA COMISIÓN:

VIVIANA ISABEL MERUANE NARANJO

RAFAEL OMAR RUIZ GARCÍA

SANTIAGO DE CHILE

2022

## **DETECCIÓN DE FALLAS EN EQUIPOS TÉRMICOS A PARTIR DE TRANSFORMADORES Y MECANISMOS DE ATENCIÓN**

El presente informe tiene por objetivo presentar los resultados más importantes obtenidos del desarrollo, entrenamiento, aplicación y evaluación de modelos neuronales artificiales de series temporales de tipo Transformador, LSTM, GRU y RNN diseñados bajo dos arquitecturas propuestas basadas en los papers “Attention is all you need” y “An image worth 16x16 words” orientados para tareas clasificación de identificación de estados de operación y tareas de regresión sobre estimación del tiempo de vida útil remante a partir de muestras de ventanas temporales de entrada, que provienen de un registro de parámetros de operación de un intercambiador de calor de un sistema rehervidor de generación de dióxido de cloro de una planta de celulosa, labor realizada como trabajo de título por el alumno David Ignacio Lizana Pardo de la carrera de ingeniería civil mecánica de la Universidad de Chile efectuado durante el segundo semestre académico de primavera del año 2021 y el primer semestre académico del otoño del año 2022.

El presente documento describe como motivación principal del tema, promover un avance de esta clase de tecnología dentro del área de ingeniería de mantenimiento industrial de tipo predictivo orientado hacia el pronóstico y diagnóstico de falla de activos críticos dentro de la cadena productiva. Para luego continuar con una recopilación de los conceptos básicos e ideas fundamentales relacionadas a la estructura y funcionamiento de las celdas unitarias características de cada tipo de modelo neuronal bajo estudio, junto a sus principales ventajas y desventajas, así como también sobre los artículos científicos a partir de los cuales nacen las arquitecturas propuestas para los modelos. En base a esto, posteriormente se describe en detalle la estructura de los modelos neuronales propuestos exponiendo en profundidad la estructura de sus subcomponentes y tipos de conexiones entre cada elemento, denotando las funciones específicas que desempeñan dentro de la red. Ya con esto, se procede a describir la estrategia de trabajo adoptada por el estudiante para efectuar el diseño, entrenamiento, evaluación y análisis de sensibilidad aplicado sobre cada tipo de modelo neuronal diseñados bajo las dos arquitecturas propuestas, especificando los plazos de tiempo asignados para cada una de estas actividades a lo largo de los 6 meses de trabajo. A partir de entonces, se continúa con la descripción del caso de estudio y recursos, detallando la estructura y el principal problema del sistema de rehervidor de la planta de celulosa a partir del cual se obtiene el registro parámetros operaciones base de los modelos neuronales. Posteriormente se procede con la presentación de resultados a la par el análisis de estos datos, que para el caso clasificación se realiza a través de los indicadores de rendimiento Exactitud, Precisión, Exhaustividad, Área bajo la curva ROC, mientras que para el caso de regresión por medio de los indicadores de Error cuadrático medio, Error absoluto medio y Coeficiente de determinación.

A partir del análisis de resultados se desprende que la arquitectura más eficiente para tareas de clasificación y regresión corresponde a la arquitectura basada en el paper “An image worth 16x16 words” por su mayor capacidad de extraer características centradas en variables de operación de un mismo tiempo, por otro lado, dentro de los tipos de modelos neuronales para el caso de clasificación el orden creciente de desempeño logrado por cada estructura corresponde respectivamente al modelo de tipo Transformador, LSTM, GRU y RNN, pues el desempeño depende más de la identificación de patrones sobre las ventanas temporales de entrada. Mientras para el caso de regresión el orden creciente de desempeño corresponde al modelo de tipo RNN, LSTM, GRU y Transformador, ya que el rendimiento depende más del desarrollo de características centrada en las últimas variables de operación de la ventana temporal de entrada, finalmente gracias el análisis de sensibilidad se corrobora que estos comportamientos son invariantes según el número de parámetros de operación que componen el codificador de las estructuras.

## **Agradecimientos**

Quiero agradecer a mi padre Mario Lizana, a mi madre Jessica Pardo y a mi hermana Elizabeth Lizana por haberme dado su apoyo incondicional durante todos estos años de duro trabajo y grandes sacrificios en esta etapa de mi vida, siendo el mayor impulso para sobreponerme ante las dificultades enfrentadas y un gran motivo para superar los desafíos relacionados con el cumplimiento de mis objetivos académicos y profesionales a lo largo de la carrera, que finalmente hoy dan como fruto mi alegría y orgullo, este momento no solo me pertenece a mí sino también a ellos. Muchas gracias por su apoyo incondicional, sin su compañía en este camino no hubiera podido llegar a donde estoy y no sería en quien hoy me he convertido, este triunfo que he alcanzado también es de ustedes padres.

Por sobre todo agradezco a mis abuelos María Caris y Héctor Pardo, quienes me enseñaron muchas cosas desde pequeño con su gran sabiduría mientras me criaban, entregándome los valores y principios que rigen mi vida que me encaminan cada día a llegar más alto, gracias por sus cuidados desde pequeño, jamás los olvidaré y siempre los llevaré en mi corazón a todas partes.

Deseo expresar también mi gratitud a mis amigos que han sido como hermanos para mí, quienes estuvieron a mi lado dándome ánimo y otorgándome grandes consejos para continuar en estos últimos pasos. Agradezco a Franco Lobos por su gran compañía brindada durante los peores momentos de mi vida alentándome a salir adelante a pesar de todo y brindándome fuerzas para seguir luchando, a Fernando Miranda por ser un fuerte pilar en mi formación académica representando un modelo profesional a seguir y del cual aprendí mucho durante los años críticos de mi carrera, y finalmente a Juan Chaparro por ser un fiel compañero de traspasos durante múltiples trabajos extensos para la universidad que por medio de conversaciones profundas y juegos en línea se hacían mucho más amenas las horas.

Por parte del cuerpo docente de la facultad de ciencias físicas y matemáticas de la universidad de Chile le doy las gracias al profesor Enrique López Droguett por la orientación brindada en el transcurso del desarrollo del presente proyecto, a la profesora Viviana Meruane por otorgarme nuevos puntos de vista que concluyeron en nuevos temas brindando un mayor sustento y solidez al trabajo de título elaborado, y al profesor Rafael Ruiz por ser un gran ejemplo de vocación y pasión por la ingeniería y la buena docencia.

---

## **Tabla de contenido**

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Problema base y solución propuesta	1
1.2	Motivación	2
1.3	Objetivos de trabajo	2
1.4	Alcances	3
<b>2</b>	<b>Antecedentes</b>	<b>5</b>
2.1	Red neuronal Transformador	5
2.2	Red neuronal recurrente simple (RNN)	12
2.3	Red neuronal memoria larga corto plazo (LSTM)	16
2.4	Red neuronal unidad recurrente cerrada (GRU)	20
2.5	Visión de transformador (VIT)	24
<b>3</b>	<b>Propuesta de diseño de modelos neuronales</b>	<b>27</b>
3.1	Arquitecturas base propuestas	27
3.2	Estructura de modelos neuronales	29
3.3	Selección inicial de hiperparámetros	31
3.4	Composición de capa características por tipo de modelo	31
3.5	Análisis de sensibilidad	32
3.6	Evaluación de modelo propuestos	32
3.6.1	Indicadores de rendimiento	32
<b>4</b>	<b>Metodología</b>	<b>35</b>
<b>5</b>	<b>Estudio de caso</b>	<b>38</b>
<b>6</b>	<b>Recursos</b>	<b>41</b>
6.1	Recursos no pecuniarios	41
6.2	Recursos pecuniarios	44
<b>7</b>	<b>Resultados y Discusión</b>	<b>46</b>
<b>7.1</b>	<b>Resultados de modelos de clasificación</b>	<b>47</b>
7.1.1	Función de pérdida de proceso de aprendizaje automático	47
7.1.2	Matriz de confusión	49
7.1.3	Curva característica de funcionamiento del receptor	51
7.1.4	Indicadores de rendimiento por tipo de modelo	53
7.1.5	Gráficos de comparativos de rendimiento:	54
<b>7.2</b>	<b>Resultados de modelos de regresión</b>	<b>61</b>
7.2.1	Función de pérdida de proceso de aprendizaje automático	61
7.2.2	Diagrama de dispersión valor predicho vs valor real	63
7.2.3	Diagrama de dispersión ordenado valor predicho vs valor real	64
7.2.4	Indicadores de rendimiento por tipo de modelo	65
7.2.5	Gráficos comparativos de rendimiento	66
<b>7.3</b>	<b>Análisis de sensibilidad sobre modelos de clasificación</b>	<b>72</b>
7.3.1	Función de pérdida de proceso de aprendizaje automático	72
7.3.2	Matriz de confusión	73
7.3.3	Curva característica de funcionamiento del receptor	74
7.3.4	Indicadores de rendimiento por variantes de modelo	75
7.3.5	Gráficos comparativos de rendimiento	79
<b>7.4</b>	<b>Análisis de sensibilidad sobre modelos de regresión</b>	<b>86</b>
7.4.1	Función de pérdida de proceso de aprendizaje automático	86
7.4.2	Diagrama de dispersión valor predicho vs valor real:	87
7.4.3	Diagrama de dispersión ordenado valor predicho vs valor real:	88
7.4.4	Indicadores de rendimiento por variantes de modelo:	89
7.4.5	Gráficos comparativos de rendimiento	93
<b>8</b>	<b>Conclusiones</b>	<b>98</b>
	<b>Bibliografía</b>	<b>100</b>

---

<b>Anexos .....</b>	<b>101</b>
<b>Anexo A: Registro de resultados de modelos de clasificación por iteración de aprendizaje .....</b>	<b>101</b>
Anexo A.1: Arquitectura “Attention is all you need” .....	101
Anexo A.2: Arquitectura “An image is worth 16x16 words”: .....	103
<b>Anexo B: Registro de resultados de modelos de regresión por iteración de aprendizaje .....</b>	<b>105</b>
Anexo B.1: Arquitectura “Attention is all you need” .....	105
Anexo B.2: Arquitectura “An image is worth 16x16 words” .....	107
<b>Anexo C: Funciones de pérdida de modelos alternativos para clasificación .....</b>	<b>109</b>
<b>Anexo D: Matrices de confusión de modelos alternativos para clasificación.....</b>	<b>113</b>
<b>Anexo E: Curva característica de modelos alternativos para clasificación .....</b>	<b>117</b>
<b>Anexo F: Registro de resultados de modelos alternativos en clasificación.....</b>	<b>121</b>
Anexo F.1: Arquitectura “Attention is all you need” .....	121
Anexo F.2: Arquitectura “An image is worth 16x16 words” .....	131
<b>Anexo G: Funciones de pérdida para regresión de modelos alternativos .....</b>	<b>141</b>
<b>Anexo H: Diagrama de dispersión de modelos alternativos .....</b>	<b>145</b>
<b>Anexo I: Diagrama de dispersión ordenado de modelos alternativos .....</b>	<b>149</b>
<b>Anexo J: Registro de resultados de modelos alternativos en regresión .....</b>	<b>153</b>
Anexo J.1: Arquitectura “Attention is all you need”: .....	153
Anexo J.2: Arquitectura “An image is worth 16x16 words” .....	163

---

## Índice de tablas

<b>Tabla 1:</b> Selección de hiperparámetros de modelos neuronales propuestos .....	31
<b>Tabla 2:</b> Número de celdas distintivas de la capa característica según tipo de modelo .....	31
<b>Tabla 3:</b> Sensores de monitoreo y variables de operación registradas del sistema .....	40
<b>Tabla 4:</b> Características globales de las variables de operación registradas .....	45
<b>Tabla 5:</b> Propiedades generales de etiquetas de estados y vida útil remanente.....	45
<b>Tabla 6:</b> Indicadores de rendimiento en clasificación de modelos bajo arquitectura “Attention is all you need”.....	53
<b>Tabla 7:</b> Indicadores de rendimiento en clasificación de modelos bajo arquitectura “An image is worth 16x16 words” .....	53
<b>Tabla 8:</b> Indicadores de rendimiento en regresión de modelos bajo arquitectura “Attention is all you need”. .....	65
<b>Tabla 9:</b> Indicadores de rendimiento en regresión de modelos bajo arquitectura “An image is worth 16x16 words”.....	65
<b>Tabla 10:</b> Indicadores de rendimiento en clasificación de modelos con 2 codificadores bajo arquitectura “Attention is all you need”.....	75
<b>Tabla 11:</b> Indicadores de rendimiento en clasificación de modelos con 4 codificadores bajo arquitectura “Attention is all you need”.....	75
<b>Tabla 12:</b> Indicadores de rendimiento en clasificación de modelos con 6 codificadores bajo arquitectura “Attention is all you need”.....	76
<b>Tabla 13:</b> Indicadores de rendimiento en clasificación de modelos con 8 codificadores bajo la arquitectura “Attention is all you need”.....	76
<b>Tabla 14:</b> Indicadores de rendimiento en clasificación de modelos con 10 codificadores bajo arquitectura “Attention is all you need”.....	76
<b>Tabla 15:</b> Indicadores de rendimiento en clasificación de modelos con 2 codificadores bajo arquitectura “An image is worth 16x16 words” .....	77
<b>Tabla 16:</b> Indicadores de rendimiento en clasificación de modelos con 4 codificadores bajo arquitectura “An image is worth 16x16 words” .....	77
<b>Tabla 17:</b> Indicadores de rendimiento en clasificación de modelos con 6 codificadores bajo arquitectura “An image is worth 16x16 words” .....	77
<b>Tabla 18:</b> Indicadores de rendimiento en clasificación de modelos con 8 codificadores bajo arquitectura “An image is worth 16x16 words” .....	78
<b>Tabla 19:</b> Indicadores de rendimiento en clasificación de modelos con 10 codificadores bajo arquitectura “An image is worth 16x16 words” .....	78
<b>Tabla 20:</b> Indicadores de rendimiento en regresión de modelos con 2 codificadores bajo arquitectura “Attention is all you need”.....	89
<b>Tabla 21:</b> Indicadores de rendimiento en regresión de modelos con 4 codificadores bajo arquitectura “Attention is all you need”.....	89
<b>Tabla 22:</b> Indicadores de rendimiento en regresión de modelos con 6 codificadores bajo arquitectura “Attention is all you need”.....	90
<b>Tabla 23:</b> Indicadores de rendimiento en regresión de modelos con 8 codificadores bajo arquitectura “Attention is all you need”.....	90
<b>Tabla 24:</b> Indicadores de rendimiento en regresión de modelos con 10 codificadores bajo arquitectura “Attention is all you need”.....	90
<b>Tabla 25:</b> Indicadores de rendimiento en regresión de modelos con 2 codificadores bajo arquitectura “An image is worth 16x16 words” .....	91
<b>Tabla 26:</b> Indicadores de rendimiento en regresión de modelos con 4 codificadores bajo arquitectura “An image is worth 16x16 words” .....	91
<b>Tabla 27:</b> Indicadores de rendimiento en regresión de modelos con 6 codificadores bajo arquitectura “An image is worth 16x16 words” .....	91
<b>Tabla 28:</b> Indicadores de rendimiento en regresión de modelos con 8 codificadores bajo arquitectura “An image is worth 16x16 words” .....	92
<b>Tabla 29:</b> Indicadores de rendimiento en regresión de modelos con 10 codificadores bajo arquitectura “An image is worth 16x16 words” .....	92



---

## Índice de figuras

<b>Figura 1:</b> Arquitectura del modelo Transformador.....	6
<b>Figura 2:</b> Celda de mecanismo de atención.....	7
<b>Figura 3:</b> Mecanismo de atención enmascarada.....	8
<b>Figura 4:</b> Conjunto de atención múltiple.....	10
<b>Figura 5:</b> Arquitectura de red neuronal recurrente simple.....	12
<b>Figura 6:</b> Estructura de la celda unitaria de red neuronal recurrente simple.....	13
<b>Figura 7:</b> Arquitectura de red neuronal de Memoria larga a corto plazo.....	16
<b>Figura 8:</b> Estructura de la celda unitaria de red neuronal recurrente simple.....	17
<b>Figura 9:</b> Arquitectura de red neuronal unidad recurrente cerrada.....	20
<b>Figura 10:</b> Estructura de la celda unitaria de las redes de unidad recurrente cerrada.....	21
<b>Figura 11:</b> Arquitectura de red neuronal de la visión transformador.....	24
<b>Figura 12:</b> Estructura de codificador de transformador.....	26
<b>Figura 13:</b> Estructura nodal de perceptrón multicapa.....	26
<b>Figura 14:</b> Esquemas representativos de las arquitecturas neuronales propuestas.....	27
<b>Figura 15:</b> Modelo de clasificación bajo arquitectura “Attention is all you need”.....	29
<b>Figura 16:</b> Modelo de clasificación bajo arquitectura “An image is worth 16x16 words”.....	29
<b>Figura 17:</b> Modelo de regresión bajo arquitectura “Attention is all you need”.....	30
<b>Figura 18:</b> Modelo de regresión bajo arquitectura “An image is worth 16x16 words”.....	30
<b>Figura 19:</b> Diagrama de estrategia de trabajo del trabajo de título.....	37
<b>Figura 20:</b> Diagrama del sistema rehervidor de generación de dióxido de cloro.....	38
<b>Figura 21:</b> Incrustaciones de sulfato en el interior de los tubos del rehervidor.....	40
<b>Figura 22:</b> Modelo de notebook de trabajo [Fuente; Pagina Web “www.solotodo.cl”].....	41
<b>Figura 23:</b> Comparación de funciones de pérdida en clasificación de cada tipo de modelo neuronal.....	47
<b>Figura 24:</b> Comparación de matrices de confusión de clasificación de cada tipo de modelo neuronal.....	49
<b>Figura 25:</b> Comparación de curvas características de clasificación de cada tipo de modelos bajo estudio.....	51
<b>Figura 26:</b> Gráfico comparativo del tiempo de entrenamiento según tipo de modelo y arquitectura.....	54
<b>Figura 27:</b> Gráfico comparativo del indicador de rendimiento de exactitud según tipo de modelo y arquitectura.....	55
<b>Figura 28:</b> Gráfico comparativo del indicador de rendimiento de precisión según tipo de modelo y arquitectura.....	56
<b>Figura 29:</b> Gráfico comparativo del indicador de rendimiento de exhaustividad según tipo de modelo y arquitectura.....	57
<b>Figura 30:</b> Gráfico comparativo del indicador de rendimiento de puntaje valor F1 según tipo de modelo y arquitectura.....	58
<b>Figura 31:</b> Gráfico comparativo del indicador de rendimiento área bajo curva ROC según tipo de modelo y arquitectura.....	59
<b>Figura 32:</b> Comparación de funciones de pérdida en regresión de cada tipo de modelos neuronal.....	61
<b>Figura 33:</b> Comparación de gráfico de dispersión de valor predicho contra valor real de la vida útil remanente del equipo térmico según tipo de modelo neuronal.....	63
<b>Figura 34:</b> Comparación de gráfico de dispersión ordenado de valor predicho contra valor real de la vida útil remanente del equipo térmico según tipo de modelo neuronal.....	64
<b>Figura 35:</b> Gráfico comparativo del tiempo de entrenamiento según tipo de modelo y arquitectura.....	66
<b>Figura 36:</b> Gráfico comparativo del indicador de rendimiento error cuadrático medio según tipo de modelo y arquitectura.....	67
<b>Figura 37:</b> Gráfico comparativo del indicador de rendimiento error absoluto medio según tipo de modelo y arquitectura.....	69
<b>Figura 38:</b> Gráfico comparativo del indicador de rendimiento coeficiente de determinación según tipo de modelo y arquitectura.....	70
<b>Figura 39:</b> Gráfico comparativo de tiempo de entrenamiento según tipo de modelo, arquitectura y número de codificadores que los conforman.....	79
<b>Figura 40:</b> Gráfico comparativo del indicador de rendimiento Exactitud según tipo de modelo, arquitectura y número de codificadores que los conforman.....	80

---

<b>Figura 41:</b> Gráfico comparativo del indicador de rendimiento precisión según tipo de modelo, arquitectura y número de codificadores que los conforman. ....	81
<b>Figura 42:</b> Gráfico comparativo del indicador de rendimiento exhaustividad según tipo de modelo, arquitectura y número de codificadores que los conforman. ....	82
<b>Figura 43:</b> Gráfico comparativo del indicador de rendimiento puntaje valor F1 según tipo de modelo, arquitectura y número de codificadores que los conforman. ....	83
<b>Figura 44:</b> Gráfico comparativo del indicador de rendimiento Área bajo curva característica según tipo de modelo, arquitectura y número de codificadores que los conforman. ....	84
<b>Figura 45:</b> Gráfico comparativo de tiempo de entrenamiento según tipo de modelo, arquitectura y número de codificadores que los conforman. ....	93
<b>Figura 46:</b> Gráfico comparativo del indicador de rendimiento error cuadrático medio según tipo de modelo, arquitectura y número de codificadores que los conforman. ....	94
<b>Figura 47:</b> Gráfico comparativo del indicador de rendimiento error absoluto medio según tipo de modelo, arquitectura y número de codificadores que los conforman. ....	95
<b>Figura 48:</b> Gráfico comparativo del indicador de rendimiento coeficiente de determinación según tipo de modelo, arquitectura y número de codificadores que los conforman. ....	96
<b>Figura 49:</b> Comparación de funciones de pérdida en clasificación de modelos variantes del modelo transformador. ....	109
<b>Figura 50:</b> Comparación de funciones de pérdida en clasificación de modelos variantes del modelo LSTM. ....	110
<b>Figura 51:</b> Comparación de funciones de pérdida en clasificación de modelos variantes del modelo GRU. ....	111
<b>Figura 52:</b> Comparación de funciones de pérdida en clasificación de modelos variantes del modelo RNN. ....	112
<b>Figura 53:</b> Comparación de matrices de confusión de clasificación de variantes del modelo transformador. ....	113
<b>Figura 54:</b> Comparación de matrices de confusión de clasificación de variantes del modelo LSTM. ....	114
<b>Figura 55:</b> Comparación de matrices de confusión de clasificación de variantes del modelo GRU. ....	115
<b>Figura 56:</b> Comparación de matrices de confusión de clasificación de variantes del modelo RNN. ....	116
<b>Figura 57:</b> Comparación de curvas características de variantes del modelo transformador. ....	117
<b>Figura 58:</b> Comparación de curvas características de variantes del modelo LSTM. ....	118
<b>Figura 59:</b> Comparación de curvas características de variantes del modelo GRU. ....	119
<b>Figura 60:</b> Comparación de curvas características de variantes del modelo RNN. ....	120
<b>Figura 61:</b> Comparación de funciones de pérdida en regresión de variantes del modelo transformador. ....	141
<b>Figura 62:</b> Comparación de funciones de pérdida en regresión de variantes del modelo LSTM. ....	142
<b>Figura 63:</b> Comparación de funciones de pérdida en regresión de variantes del modelo GRU. ....	143
<b>Figura 64:</b> Comparación de funciones de pérdida en regresión de variantes del modelo RNN. ....	144
<b>Figura 65:</b> Comparación de gráfico de dispersión de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo transformador. ....	145
<b>Figura 66:</b> Comparación de gráfico de dispersión de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo LSTM. ....	146
<b>Figura 67:</b> Comparación de gráfico de dispersión de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo GRU. ....	147
<b>Figura 68:</b> Comparación de gráfico de dispersión de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo RNN. ....	148
<b>Figura 69:</b> Comparación de gráfico de dispersión ordenado de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo transformador. ....	149
<b>Figura 70:</b> Comparación de gráfico de dispersión ordenado de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo LSTM. ....	150
<b>Figura 71:</b> Comparación de gráfico de dispersión ordenado de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo GRU. ....	151
<b>Figura 72:</b> Comparación de gráfico de dispersión ordenado de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo RNN. ....	152

# 1 Introducción

## 1.1 Problema base y solución propuesta

Dentro del sector industrial se ha hecho constantemente presente el riesgo de pérdidas humanas dentro de los procesos productivos que pueden colocar en riesgo las vidas de los trabajadores a través de accidentes laborales provocados por un error humano o de manera intencionada, facilitando dentro de las instalaciones el surgimiento de siniestros como incendios, inundaciones, fuga de material tóxico o descargas eléctricas de alta tensión, por otro lado además se tiene el peligro de pérdidas de activos físicos de la empresa que comprometen la estabilidad financiera de la organización a causa de fallas catastróficas en equipos y maquinaria ligada a la cadena productiva, o generando gastos innecesarios en repuestos por diagnósticos erróneos de estado y deterioro acelerado de mecanismos por mal uso de los instrumentos.

Es a partir de esto que surge la necesidad de automatizar los procesos industriales con el objetivo de mejorar constantemente la eficiencia de las actividades reduciendo la intervención humana en estos a la par de mitigar riesgos fatales que esto involucra generando condiciones de trabajo más seguras, permitiendo llevar un mayor control de las operaciones por medio del monitoreo continuo de las variables implicadas, garantizándonos un diagnóstico más preciso del estado de los equipos que repercute por una parte en una disminución de los costos de suministros de mantenimiento y aminora el riesgo de fallo de maquinaria e interrupciones no previstas de sistemas.

Los sistemas automatizados son mecanismos de control que se regulan y se desempeñan en la ejecución de tareas específicas sin necesidad de la intervención humana destacándose por ser capaces de llevar a cabo funciones que sobrepasan la capacidad humana (mayor velocidad, labor continua o mejor precisión), estos sistemas están conformados por dos subsistemas; el sistema actuador integrado por los elementos que permiten que la máquina realice la acción deseada y el sistema de mando que genera las órdenes precisas para que se ejecuten las acciones. Si bien el diseño del sistema actuador está restringido a la actividad que desempeña la máquina dentro del proceso productivo, el sistema de mando no posee esta limitante por lo que puede estar confeccionada por tecnología programable diseñada para adaptarse a las variaciones de configuración de la labor a través de un software, relés electromagnéticos que funcionan como un interruptor controlador de la apertura o cierre de circuitos eléctricos independientes, tarjetas electrónicas conformadas por placa de circuito impreso que conectan los componentes electromecánicos para que un sistema funcione como se desea o módulos lógicos neumáticos que se caracterizan por ser un conjunto de placas que utilizan gases comprimido para definir una acción. De acuerdo con los tipos de tecnologías programables nos encontramos con una gran variedad de opciones, entre las más populares están los controladores lógicos programables (PLC), robot control (RC), control numérico computarizado (CNC) y las redes neuronales artificiales.

Sobre la base de la tecnología programable de las redes neuronales en el contexto de la actual transición de la automatización de los procesos industriales, estos han destacado por presentar un gran potencial en aplicaciones de mantenimiento de tipo predictivo del campo de la ingeniería, entre los cuales encontramos el pronóstico y diagnóstico de falla de activos críticos de la cadena producción.

De la mano con esto último, se nos presenta el caso de estudio de la automatización del proceso de pronóstico y diagnóstico de falla de un sistema rehervidor de generación de dióxido de cloro en una planta de celulosa, caso a partir de cual nace el presente tema de trabajo de título como una forma de dar solución al problema, basado en la construcción de un modelo eficiente de aprendizaje automático orientado a predecir su estado de funcionamiento operacional o a estimar el tiempo de vida útil remanente del activo,

utilizando como entrada datos de su registro de parámetros de operación, fruto de años de funcionamiento del sistema. En cuanto al diseño del modelo, se ha decidido por desarrollar el modelo neuronal bajo una nueva estructura de DeepLearning llamada “Transformador”, introducida por el paper “Attention Is All You Need” [1] en el año 2017, que surge como una innovación tecnológica dirigida al área de neurolingüística, que presenta la gran ventaja de lograr un desempeño que rivaliza con los modelos de transducción de secuencias predominantes en el área, entre los cuales se encuentran los modelos de tipo LSTM, GRU y RNN, con la gran diferencia que modelo Transformador se diferencia del resto de modelos por presidir por completo de recurrencias y convoluciones dentro de su estructura, además de no disponer una unidad explícita de memoria. Cabe decir que la arquitectura de los modelos de tipo transformador corresponden a una estructura de aprendizaje automático autorregresivo conformado principalmente por dos grandes subestructuras; un codificador y un decodificador, ambas utilizan como unidad fundamental de procesamiento celdas de mecanismos de atención, que aprovechan exclusivamente la atención propia sobre los datos de entrada entregados y los datos de salida generados para extraer características útiles que garanticen definir una respuesta óptima, entre las ventajas que presenta esta arquitectura frente al resto de tipo de modelos de series temporales están: mayor memoria de largo y corto plazo, menores tiempos y costos de entrenamiento por su capacidad de paralelizar el procesamiento de la información y presentan resultados con mayor interpretación para el ser humano.

## 1.2 Motivación

El interés personal del alumno por desarrollar un trabajo de título centrado en los modelos neuronales de series temporales se debe a que este tipo de herramientas de aprendizaje automático poseen un gran potencial predictivo que puede bien aprovechado dentro de las actividades de pronóstico y diagnóstico de falla del área de mantenimiento industrial, revelando en este último tiempo un gran avance en la evolución y adaptación de la inteligencia artificial puesta en favor de la humanidad para solucionar problemas peligrosos y complejos, por su parte este tema cobra mayor interés en particular dentro de nuestro país Chile, puesto que esta disciplina científica dentro del campo de la ingeniería no ha logrado un gran desarrollo en comparación a otros países del primer mundo, por lo tanto, fomentar el progreso de este tipo de tecnología para instaurarla en la industria actual, le significaría grandes beneficios como lo es la automatización de procesos, reducción del error humano, incremento de la vida útil de los activos, aumento de productividad y calidad del servicio o producto final.

## 1.3 Objetivos de trabajo

### A) Objetivo general

Desarrollar la arquitectura, estructura e implementación de un modelo neuronal de tipo transformador orientado a la detección de estados de funcionamiento anómalos de un intercambiador de calor y a la estimación del tiempo de vida útil remanente del equipo térmico, a partir del registro temporal de parámetros operacionales de un sistema rehervidor de generación de dióxido de cloro de una planta de celulosa.

## B) Objetivos específicos

1. Recopilar los conceptos básicos y definiciones fundamentales de los modelos neuronales de series temporales de tipo Transformador, LSTM, GRU y RNN, junto a la arquitectura representada en los artículos científicos “Attention is all you need” y “An image worth 16x16 words”.
2. Analizar las características del registro de parámetros del equipo térmico para acondicionar la información contenida a fin de facilitar el proceso de aprendizaje de la red neuronal.
3. Construir el prototipo de modelo de aprendizaje de clasificación y regresión en lenguaje de programación Python mediante la herramienta online gratuita Google Colab, ejecutando el entrenamiento, validación, testeo y evaluación de la arquitectura de transformador desarrollada.
4. Construir los tres tipos de modelos neuronales comparativos que se especializan en series temporales (LSTM, GRU y RNN) con arquitectura análoga al modelo transformador definido a fin de comparar sus rendimientos dentro de la actividad de clasificación y regresión.
5. Optimizar el rendimiento de los modelos construidos modificando los hiperparámetros de la arquitectura de la red y la configuración del proceso de entrenamiento.
6. Realizar un estudio del impacto que conlleva un cambio del número de parámetro que compone al codificador sobre el rendimiento logrado por cada modelo diseñado.

## 1.4 Alcances

El proyecto se aboca al diseño de la arquitectura y optimización de rendimiento del modelo de regresión y clasificación de tipo transformador, así como también del desarrollo de los modelos comparativos análogos de tipo LSTM, GRU y RNN, para lo cual se consideran los siguientes alcances en la realización del presente trabajo:

- La alimentación del modelo utiliza como fuente de información exclusivamente el registro de monitoreo de los parámetros operacionales relacionadas al estado de funcionamiento del sistema rehervidor de generación de dióxido de cloro de una planta de celulosa, bridado por el profesor guía Enrique López.
- La salida del modelo se restringe únicamente a predecir el estado actual de funcionamiento del equipo (defectuoso o nominal) ó a estimar el tiempo de vida útil remanente (horas) antes de su próxima falla operacional.
- El diseño del modelo transformador se basa principalmente en la arquitectura de transformador representada en los artículos científicos “Attention is all you need” y “An image worth 16x16 words”, utilizando las estructuras de redes neuronales ahí descritas (Codificadores, Decodificadores, Mecanismos de atención, Dense, Softmax, Retroalimentación y Normalización)

## Introducción

---

- El código del modelo se compilará en lenguaje Python utilizando el entorno de Google Colab.
- Los modelos entrenados deben cumplir como mínimo una eficiencia de predicción del 70% dentro del grupo de datos prueba para efectos de validar la eficiencia del modelo para su posterior análisis de desempeño.

## 2 Antecedentes

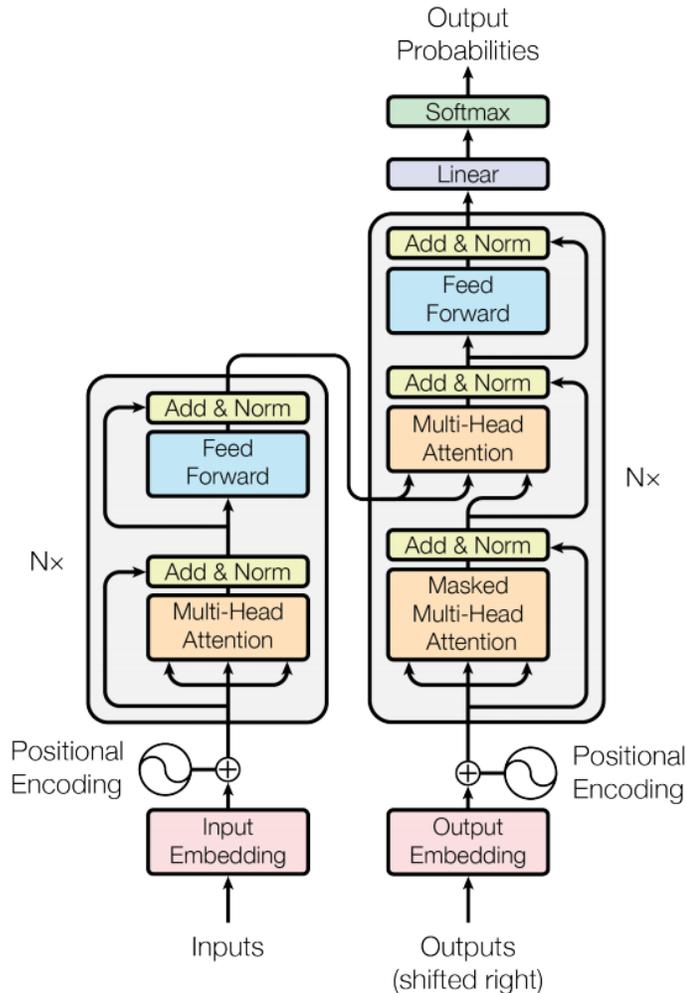
### 2.1 Red neuronal Transformador

La arquitectura del transformador fue introducida en el paper “Attention Is All You Need” [1] publicado en 2017 por Vaswani et al., desarrollado en colaboración con parte del personal de Google Brain, Google Research y la universidad de Toronto. Los transformadores son redes neuronales basadas en el funcionamiento natural de las neuronas biológicas, fueron creados como un modelo alternativo a los modelos predominantes de redes neuronales recurrentes como las redes Recurrent Neural Nets (RNN), Long Short Term Memory (LSTM) y Gated Recurrent Unit (GRU) en el área de programación neurolingüística (PNL), área que se enfoca en la traducción automática de textos, resumen de documentos, reconocimiento de entidades nombradas, generación de escritura automática con contextualización, chatbots y predicción de la siguiente oración.

El documento “Attention Is All You Need” tiene por objetivo describir en detalle el funcionamiento matemático y computacional de cada una de las estructuras que componen esta innovadora arquitectura, además de presentar sus principales beneficios y ventajas competitivas, dentro de las cuales figura el alto rendimiento desempeñado por este tipo de modelos en la traducción de texto en comparación con el rendimiento de los anteriores modelos recurrentes, para lo cual utilizando la arquitectura de tipo transformador se desarrollaron dos tipos de traductores: inglés-alemán e inglés-francés, entrenando cada modelo durante un periodo de 3.5 días utilizando 8 unidades de procesamiento gráfico (GPU), obteniendo una puntuación de 28,4 BLEU para el primero y una puntuación de 41 BLEU para el segundo. Superando los mejores resultados existentes hasta esa fecha, demostrando que este tipo de modelos son lo suficientemente potentes para lograr el rendimiento de las redes recurrentes utilizando solo una pequeña fracción de los costos de capacitación de los mejores modelos de la literatura y dentro de un periodo menor de entrenamiento.

Actualmente los transformadores se han convertido rápidamente en el modelo de elección para los problemas de PNL, reemplazando a los previos modelos de redes neuronales recurrentes, incluso ha llevado al desarrollo de variantes de este tipo de arquitectura integrándolas a sistemas previamente entrenados como BERT (Representaciones de codificador bidireccional de Transformers) y GPT (Transformador preentrenado generativo), que han sido entrenados con enormes conjuntos de datos de lenguaje general, como Wikipedia Corpus y Common Crawl, y pueden ser adaptado a tareas lingüísticas específicas. Por otro lado, cabe decir que este tipo de arquitectura se ha expandido al campo de reconocimiento de imágenes de las redes neuronales, adaptándose en nuevas estructuras como los modelos de visualización ViT (Visión Transformer).

## Arquitectura del modelo Transformador



**Figura 1:** Arquitectura del modelo Transformador [Fuente: Paper “Attention is all you need”].

Los transformadores poseen una estructura de autoencoder representada en la Figura 1 por la zona gris denotada en la ilustración como  $N \times$ , la cual a su vez está conformada por dos importantes subsecciones; por un lado tenemos la sección del codificador representada por la porción izquierda del autoencoder, que posee la función de comprimir la información de entrada extrayendo las características contenidas más importantes que posteriormente se guardan en un espacio de variables latentes, mientras por otro lado tenemos a la sección del decodificador representada por la porción derecha del autoencoder, que posee la función de descomprimir la información del espacio latente a partir de la cual extrae nuevas características útiles para construir en la salida una correcta respuesta (identificación de clase o regresión de un valor).

¿Como realiza esto el autoencoder? El codificador a partir de una secuencia de entrada de representaciones de símbolos  $X = (X_1, \dots, X_n)$  por medio de matrices de pesos  $W$  y distintas funciones lineales  $f$  mapea una secuencia de representaciones continuas  $Z = (Z_1, \dots, Z_n)$ . Luego una vez obtenida esta

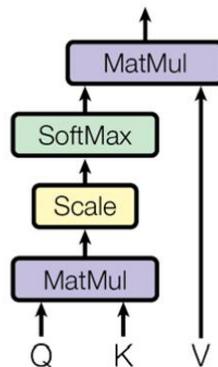
representación continua o espacio latente  $Z$  el decodificador tiene la misma función de generar una secuencia de salida de representaciones de símbolos  $Y = (Y_1, \dots, Y_m)$  de manera secuencial un elemento a la vez. Para esto último el modelo es autorregresivo, esto quiere decir que los símbolos  $Y_{i-1}$  generados previamente sirven como entrada adicional a la entrada  $X$  para generar el siguiente símbolo  $Y_i$ , sin embargo, esta nueva entrada adicional ingresa al modelo por el decodificador a diferencia de la entrada  $X$  que ingresa por el codificador.

La principal razón por la que el autoencoder es capaz de desarrollar una respuesta a partir de su entrada se debe a que tanto el codificador y decodificador están compuestos a su vez por tres distintos tipos de bloques de procesamiento; el primer bloque corresponde a múltiples celdas de mecanismos de atención representadas por la zona anaranjada de la Figura 1 que permite a la red centrarse en la información relevante contenida en la entrada, el segundo bloque corresponde a capas de retroalimentación que permiten el refinamiento de las características extraídas y el último bloque corresponde a capas de normalización que facilitan el procesamiento de los datos.

## Componentes principales del modelo

### 1) Celda de mecanismo de atención:

Scaled Dot-Product Attention



*Figura 2:* Celda de mecanismo de atención [Fuente: Paper “Attention is all you need”].

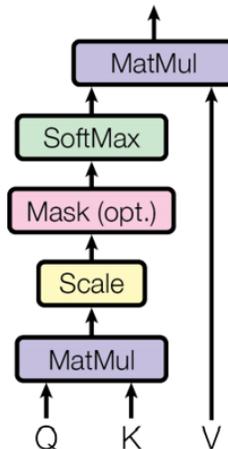
La celda de mecanismo de atención es la unidad de procesamiento fundamental de los transformadores, esta unidad se compone por tres tipos de matrices; llamadas Consulta  $Q$ , Clave  $K$  y Valor  $V$  como se pueden observar en la Figura 2, cada una de estas matrices se crean al multiplicar matrices de valores de pesos  $W_q$ ,  $W_k$  y  $W_v$  sobre la matriz de información de entrada  $X$ , luego la matriz  $Q$  y la matriz traspuesta de  $K$  se multiplican punto a punto creando la matriz  $QK^T$ , sobre la cual se aplica una operación de normalización y escala de valores dividiendo cada valor dentro de esta matriz por el valor de la dimensionalidad de la matriz  $K$ , denotado a partir de ahora como  $d_k$ , posteriormente sobre la matriz producto de esta operación se aplica la función Softmax para pasar cada valor dentro de la matriz a un valor de probabilidad, con esto se obtiene una matriz de atención  $A$  con valores dentro del rango  $[0, 1]$ , cada uno de estos valores corresponden a los pesos de atención que se les debe asignar a cada uno de los valores de la matriz  $V$ , por lo tanto la última

operación a realizar es una multiplicación entre la matriz  $V$  y la matriz traspuesta de  $A$  para finalmente obtener los valores que componen la matriz  $M$ , esta última matriz  $M$  representa la información más útil que se encuentra contenida en la entrada  $X$  para desarrollar una respuesta óptima  $Y$ , suprimiendo aquellos datos que no son útiles otorgándoles valores cercanos al cero. A continuación, se expone la representación matemática de la operación ejecutada por la celda de mecanismo de atención:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{dk}}\right) V$$

**2) Celda de mecanismo de atención enmascarada:**

Scaled Dot-Product Attention



*Figura 3: Mecanismo de atención enmascarada [Fuente: Paper “Attention is all you need”].*

La celda de mecanismo de atención enmascarada corresponde a un tipo especial de celda de mecanismo de atención que se ubica específicamente en la entrada autorregresiva del decodificador, este tipo de celda se diferencia de la anterior ya que posee un bloque extra entre la capa Softmax y la capa de normalización, llamada capa máscara denotada en la Figura 3 por el bloque Mask, que tiene la función de impedir que el programa fabrique respuestas predefinidas utilizando únicamente la información que ingresa desde el codificador proveniente de la entrada, obligando a que el programa “aprenda” de la retroalimentación de sus propias respuestas previas.

Debido a que el modelo es autorregresivo, es decir, agrupa y consume sus respuestas previas como una nueva entrada adicional para generar la siguiente respuesta, esto fuerza a que las respuestas se generen de una a una de manera secuencial en el tiempo, por lo tanto, necesitamos una forma de prevenir el condicionamiento de las futuras respuestas basadas exclusivas en la información de entrada inicial y necesitamos que estas nuevas respuestas se generen considerando las respuestas de instantes previos, para lograr la autoregresividad se utiliza una matriz triangular inferior incompleta llamada máscara compuesta por valores infinitos negativos en la zona triangular

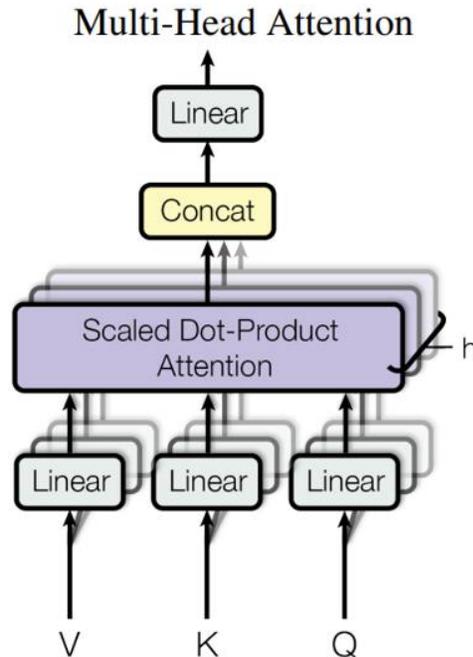
superior de la matriz y ceros en la parte triangular inferior de la matriz, que a medida que se van generando las nuevas respuestas se van agregando filas compuestas de ceros e infinitos negativos dentro de la matriz máscara completando su distribución triangular, cabe destacar que los valores infinitos negativos de la matriz máscara cumplen el rol de transformarse en valores nulos cuando se aplica la función Softmax mientras el resto de los valores se transforman en valores de probabilidades, con esto en cuenta la matriz máscara se suma a la matriz de puntajes escalados de las respuestas previas para obtener finalmente la matriz de puntajes enmascarados, un depósito de respuestas previas que se utilizarán como entrada adicional para generar la siguiente respuesta.

$$\begin{array}{ccc}
 \text{Puntajes escalados} & & \text{Máscara} & & \text{Puntajes Enmascarados} \\
 \begin{bmatrix} 0.7 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.2 & 0.1 \\ 0.1 & 0.3 & 0.6 & 0.1 \\ 0.1 & 0.3 & 0.3 & 0.3 \end{bmatrix} & + & \begin{bmatrix} 0 & -\infty & -\infty & -\infty \\ 0 & 0 & -\infty & -\infty \\ 0 & 0 & 0 & -\infty \\ 0 & 0 & 0 & 0 \end{bmatrix} & = & \begin{bmatrix} 0.7 & -\infty & -\infty & -\infty \\ 0.1 & 0.6 & -\infty & -\infty \\ 0.1 & 0.3 & 0.6 & -\infty \\ 0.1 & 0.3 & 0.3 & 0.3 \end{bmatrix}
 \end{array}$$

$$\begin{array}{cc}
 \text{Puntajes Enmascarados} & \text{Matriz Probabilística} \\
 \text{Softmax} \left( \begin{bmatrix} 0.7 & -\infty & -\infty & -\infty \\ 0.1 & 0.6 & -\infty & -\infty \\ 0.1 & 0.3 & 0.6 & -\infty \\ 0.1 & 0.3 & 0.3 & 0.3 \end{bmatrix} \right) & = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.37 & 0.62 & 0 & 0 \\ 0.26 & 0.31 & 0.43 & 0 \\ 0.21 & 0.26 & 0.26 & 0.26 \end{bmatrix}
 \end{array}$$

$$\begin{array}{ccccc}
 & \textit{Start} & \textit{I} & \textit{am} & \textit{fine} \\
 \textit{Start} & \begin{bmatrix} 0.7 & \# & \# & \# \end{bmatrix} & & & \\
 \textit{I} & \begin{bmatrix} 0.1 & 0.6 & \# & \# \end{bmatrix} & & & \\
 \textit{am} & \begin{bmatrix} 0.1 & 0.3 & 0.6 & \# \end{bmatrix} & & & \\
 \textit{fine} & \begin{bmatrix} 0.1 & 0.3 & 0.3 & 0.3 \end{bmatrix} & & & 
 \end{array}$$

### 3) Conjunto de atención multicabeza:



*Figura 4: Conjunto de atención múltiple [Fuente: Paper “Attention is all you need”].*

El conjunto de atención multicabeza corresponde a una distribución en paralelo de celdas de mecanismo de atención, a las cuales se les distribuye la información de entrada inicial para que cada una de estas celdas produzca su propia matriz M, luego todas estas matrices resultantes son concatenadas y multiplicada punto a punto con una matriz de pesos  $W^O$ , con el fin de obtener una única matriz M de promedio ponderado. Cabe destacar que como hay una variante de los mecanismos de atención enmascarada también hay un conjunto de atención multicabeza enmascarado.

$$MultiHead(Q, K, V) = Concat(head1, \dots, headj)W^O$$

$$Donde; head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

- 4) **Capas de normalización y adición:** Corresponde a una capa de tipo Dense (nodos de operaciones lineal) que tiene por función de activación cualquier función de estandarización de datos en un rango común, usualmente entre  $[0,1]$ , estas funciones de activación pueden ser trigonométricas, exponenciales o funciones de unidad lineal rectificadas (ReLU) [2]. Además, también se le envía y agrega los datos de entrada previo al procesamiento por el conjunto de atención multicabeza para que el impacto del algoritmo de propagación de errores hacia atrás utilizada para optimizar los pesos sea más eficiente, a esta ruta se le suele llamar conexión residual.

- 5) **Capa de retroalimentación:** Corresponde a una serie de capas Dense que tienen por objetivo procesar aún más la información de los vectores resultantes con fin de que la información contenida tenga una mejor interpretación para la red neuronal, es decir, facilitan el procesamiento matemático en la red neuronal.
- 6) **Codificador:** El codificador está conformado por varias sucesiones de secuencias de conjuntos de atención multicabeza, capas de normalización y adición, capas de retroalimentación y una última capa de normalización y adición. El objetivo del codificador consiste en sintetizar en un vector de latencia de menor longitud que el vector de entrada, extrayendo las características más importantes que se desprendan del vector de entrada, que le sea útil para predecir la etiqueta correspondiente.
- 7) **Decodificador:** El decodificador tiene la función de generar la sucesión de respuestas de salida utilizando la entrada original y sus respuestas previas, el decodificador se divide en 2 grandes bloques:
  - a) **Bloque decodificador enmascarado:** Se compone de varias sucesiones de secuencias de conjuntos de atención multicabeza enmascarado y capas de normalización y adición, que tiene por objetivo almacenar y procesar la entrada adicional compuesta de las respuestas previas entregadas.
  - b) **Bloque decodificador de salida:** Se compone de varias sucesiones de secuencias de conjuntos de atención multicabeza, capas de normalización y adición, capas de retroalimentación y una última capa de normalización y adición. Cada una de estas secuencias recibe en paralelo la salida del codificador y por otro lado se transmiten una a una el vector procesado con la información proveniente del bloque decodificador enmascarado, con estas dos entradas generan una salida que corresponde a un vector de probabilidad para cada etiqueta, que finalmente con una capa Softmax se entregará la etiqueta de mayor probabilidad en el vector.

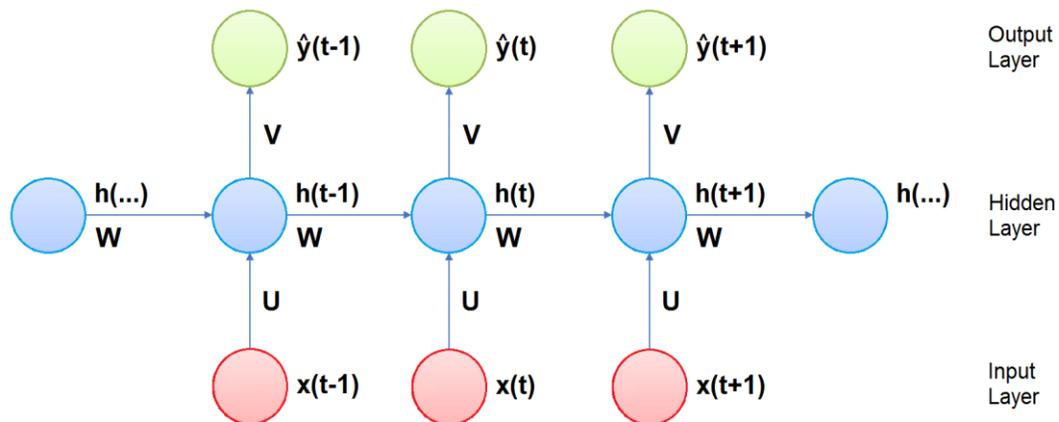
### Ventajas de los modelos transformadores:

- a) **Mayor memoria de corto y largo plazo:** Los transformadores a diferencia de los modelos recurrentes no transforman la información extraída de las características de los datos de entrada en un vector de latencia común para toda la cadena de entrada, sino que generan un vector de latencia para cada uno de los elementos de la cadena de entrada, lo que les permite procesar secuencias de datos de entrada de mayor longitud sin mayor riesgo de perder información relevante contenida en esta.
- b) **Paralelización de procesamiento:** Gracias a la estructura del codificador el procesamiento de la información de entrada no posee una naturaleza inherentemente secuencial como es el caso de los modelos recurrentes, en cambio el procesamiento de cada elemento de la entrada en los transformadores se realiza de manera simultánea.
- c) **Resultados fácilmente interpretables:** Puesto que la unidad fundamental del modelo son los mecanismos de atención inspirados en el funcionamiento de las neuronas de nuestro cerebro, presentan un comportamiento similar de suprimir aquellos detalles de la información que nos llega a nuestros sentidos y resaltar aquellos que son importantes para generar una buena respuesta.

## 2.2 Red neuronal recurrente simple (RNN)

El concepto de la arquitectura de las redes recurrentes simples fue introducido en el paper “Aprendizaje de representaciones mediante retropropagación de errores” [9] de David Rumelhart en octubre de 1986, el documento presenta la idea de una arquitectura de red que basa su extracción de característica de series temporales mediante una estructura de nodos ordenados en capas secuenciales sucesivas, similar a las redes neuronales tradicionales su estructura se divide en capa de entrada, capas ocultas y capa de salida, donde cada conexión entre neuronas tiene su correspondiente conjunto de parámetros de peso entrenables  $U(t)$ ,  $W(t)$  y  $V(t)$ , sin embargo, su principal diferencia con respecto a las redes tradicionales radica en que para el caso de las redes neuronales recurrentes a cada neurona se le asigna un intervalo de tiempo específico de la información de entrada para procesar, las neuronas interactúan entre si enviándose en una dirección dependiente del tiempo la información extraída de su correspondiente tiempo asignado  $X(t-1)$ , enviándola hacia la neurona encargada de procesar el siguiente tramo temporal de la información de entrada  $X(t)$ , esta transmisión de información se realiza mediante la capa oculta  $h(t)$  que en cada iteración de transmisión va evolucionando adquiriendo las características más relevantes que se desprenden de cada uno de los tiempos discretizados de la información de entrada, además cabe decir que de la misma manera que a cada neurona se le asigna su respectivo tramo temporal de la información de entrada  $X(t)$ , cada una de estas neuronas también emite su propia salida  $Y(t)$ , una respuesta que se crea solamente considerando la información de las características extraídas por las neuronas de los tiempos previos  $h(t-1)$  y la información de la entrada de su tiempo correspondiente  $X(t)$ .

En consecuencia, esto significa que todas las neuronas están completamente conectadas con las neuronas de la capa oculta previa  $h(t-1)$  y posterior  $h(t)$ , junto a los datos de la entrada relacionadas al paso de tiempo asignado  $X(t)$  y la entrega de una respuesta  $Y(t)$  que solo considera para su producción los datos procesados de las neuronas previas y su entrada de tiempo correspondiente, este orden se cumple para la gran mayoría de las neuronas de la arquitectura exceptuando el caso de las neuronas de entrada y salida que se encuentran conectadas al inicio de la capa oculta  $h(t)$  ó al final de esta capa oculta  $h(t-1)$ , como se puede observar en la Figura 5.

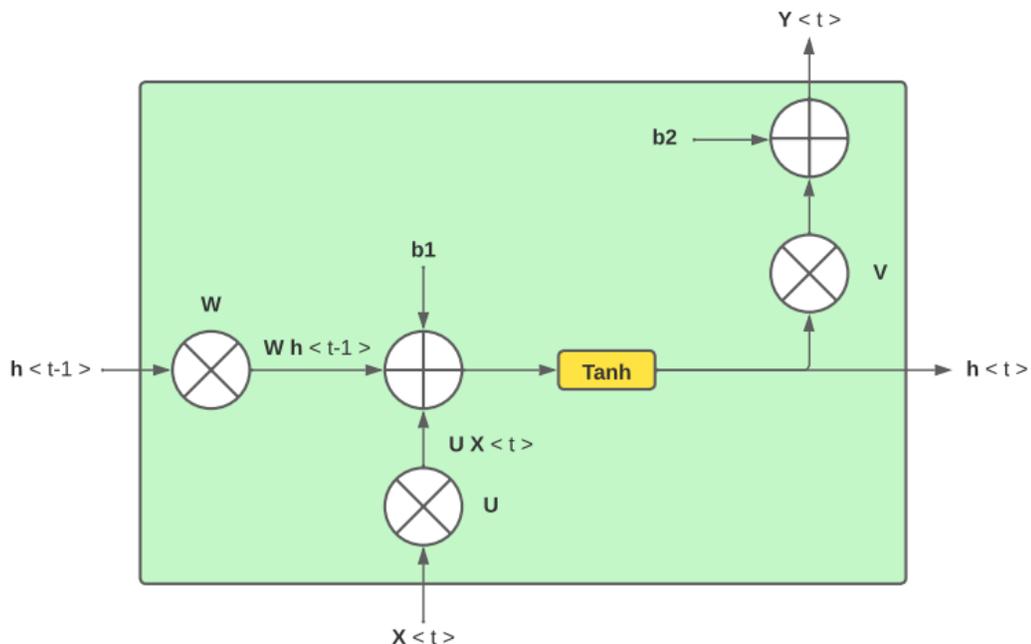


**Figura 5:** Arquitectura de red neuronal recurrente simple

[Fuente: Pagina Web “Time Series Forecasting with Deep Learning and Attention Mechanism”].

Dentro de arquitectura de las redes neuronales recurrentes simples las neuronas se ordenan en secuencia bajo una conexión unidireccional en la cual cada neurona que procesa información del tiempo  $t$  se encarga posteriormente de transmitir sus datos generados a la neurona que procesa la información del siguiente tiempo  $t+1$ , por lo tanto, la salida de la capa oculta  $h(t-1)$  de un paso de tiempo es parte de la entrada del siguiente paso de tiempo, lo que indica que la activación de las neuronas se calcula en orden de tiempo y de forma secuencial.

### Celda RNN simple



*Figura 6:* Estructura de la celda unitaria de red neuronal recurrente simple

[Fuente: Pagina Web “Building your Recurrent Neural Network - Step by Step”].

### Componentes principales del modelo

- 1) **Datos de entrada  $X(t)$ :** Corresponde a un conjunto de datos procedentes de una serie temporal de muestras de medición ordenados cronológicamente ligados a un rango de tiempo específico dentro de la sucesión temporal, siendo una fracción de la información de alimentación del modelo.
- 2) **Parámetros de pesos  $U, W, V$ :** el vector de entrada en el tiempo  $t$  está conectado a las neuronas de la capa oculta del tiempo  $t$  mediante una matriz de peso  $U$ , las neuronas de la capa oculta están conectadas a las neuronas del tiempo  $t-1$  mediante una matriz de pesos  $W$ , y las neuronas de la capa oculta están conectadas al vector de salida del tiempo  $t$  mediante una matriz de peso  $V$ ; todas las matrices de peso son constantes para cada paso de tiempo.

- 3) **Capa oculta previa  $h(t-1)$ :** Corresponde al vector de información que contiene las características más importantes extraídas del procesamiento de todos los datos de entradas previos generados por las neuronas anteriores, esta es la única conexión que existe entre la celda actual con la celda predecesora.
- 4) **Capa oculta posterior  $h(t)$ :** Corresponde al nuevo vector de información que contiene tanto las características más relevantes del procesamiento de los datos de entrada previos como también la adición de la nueva información de las características más importantes extraídos del rango de datos de entrada asignados para esta particular neurona, es una especie de memoria de la red que se calcula en función de la entrada actual y el estado oculto del paso de tiempo anterior, esta es la única conexión que existe entre la celda actual con la celda posterior.

$$h(t) = \tan(W \cdot h(t - 1) + U \cdot X(t) + b1)$$

- 5) **Respuesta de salida  $Y(t)$ :** Corresponde al vector de salida de la celda actual que para su definición considera solamente características extraídas y desarrolladas hasta esta instancia, es decir, no considera la información de las capas ocultas futuras.

$$Y(t) = V \cdot h(t) + b2$$

### Ventajas de las redes neuronales recurrentes:

- a) **Mayor adaptabilidad ante datos incompletos:** El rendimiento de las redes neuronales recurrentes no se ve afectado significativamente por la presencia de valores perdidos dentro de los datos de alimentación del modelo.
- b) **Mayor memoria de corto plazo:** Las redes neuronales recurrentes entregan muy buenos resultados al pronosticar respuestas que esencialmente no requieran de extensas muestras de series temporales para poder definir las.
- c) **Mayor capacidad de extracción de características:** Las redes neuronales recurrentes colocan mayor atención sobre los últimos datos de entrada en ser procesados de las muestras de la serie temporal, permitiéndoles extraer características más complejas tanto del contenido de los datos como también con respecto a su orden secuencial de estos.

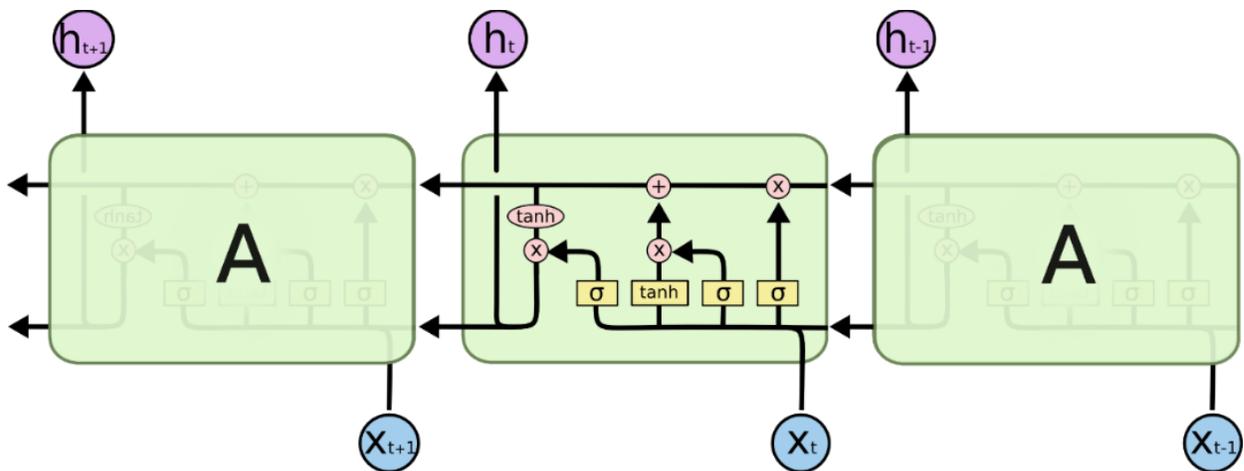
### **Desventajas de las redes neuronales recurrentes:**

- a) **Desvanecimiento de gradiente:** Cuando se entrena este tipo de arquitecturas en series de tiempo prolongado, las redes neuronales recurrentes suelen sufrir el problema del gradiente que se desvanece o del gradiente explosivo, lo que significa que los parámetros en las capas ocultas no cambian tanto o conducen a inestabilidad numérica y comportamiento caótico. Esto sucede porque el gradiente de la función de costo incluye la potencia de los pesos  $W(t)$ , lo que afecta gravemente a su capacidad de memorización a largo plazo.
- b) **Mala memoria a largo plazo:** Las redes neuronales recurrentes sufren de una memoria débil siendo incapaces de tener en cuenta varios sucesos del pasado que ocurrieron en el inicio de la muestra de serie temporal que se le ha entregado, que podría llegar a perjudicar negativamente en su capacidad de predicción del futuro.
- c) **Aumento de tiempos de entrenamiento y procesamiento:** debido a su característico procesamiento secuencial de los datos, esto le provee de mayores periodos de tiempo para terminar cada iteración de entrenamiento y posteriormente para definir una buena respuesta, además el entrenamiento de una red neuronal recurrente es difícil de paralelizar y también es computacionalmente más costoso.

## 2.3 Red neuronal memoria larga corto plazo (LSTM)

Las redes neuronales de memoria larga corto plazo (LSTM) fueron introducidas en el paper “Long Short-term Memory” [7] desarrollado por Hochreiter y Schmidhuber en diciembre de 1997, como una innovación para superar el problema de desaparición de gradiente que sufren las RNN, para ello en lugar de utilizar una capa oculta como celda de almacenamiento de características temporales se utiliza una unidad LSTM, la cual es una estructura que está compuesta por un estado de celda  $C(t)$ , una puerta de olvido  $f(t)$ , una puerta de entrada  $i(t)$  y una puerta de salida  $h(t)$ , esta nueva configuración mejora el flujo del gradiente dentro de la red permitiendo facilitar el proceso de optimización de pesos durante el entrenamiento de la red, mientras los tipos de vectores de memoria le otorga a este tipo de red una extensión de la memoria a largo plazo sobre los datos de entrada, mitigando el olvido de las características extraídas de los primeros datos de la información de entrada.

Tal como se puede observar en la Figura 7, las unidades LSTM se organizan bajo un orden en serie para procesar los datos de entrada de forma segmentada, a cada unidad LSTM se le asigna un único y específico intervalo de tiempo dentro de los datos de entrada para extraer las características que sean útiles para definir un buena respuesta, de la misma forma que en el caso precursor de las redes neuronales RNN el vector de entrada del tiempo  $t$  se encuentra conectado a la unidad LSTM del tiempo  $t$  por una matriz de peso  $U$ , también la unidad LSTM del tiempo  $t$  se encuentra conectada a la unidad LSTM del tiempo  $t-1$  y  $t+1$  por una matriz de pesos  $W$  y el vector de salida del tiempo  $t$  por un matriz de pesos  $V$ , sin embargo, para este caso las matrices  $W$  y  $U$  se dividen en submatrices con respecto las secciones de puerta de olvido ( $W_f, U_f$ ), puerta de entrada ( $W_i, W_g, U_i, U_g$ ) y puerta de salida ( $W_o, U_o$ ), por otro lado a diferencia de las redes RNN las unidades LSTM se relacionan entre sí por medio de dos conexiones entre las unidades LSTM que representan los dos tipos de memoria de la red neuronal; el estado de celda  $C(t)$  que sirve de componente de memoria de largo plazo y la puerta de salida  $h(t)$  que sirve de componente de memoria de corto plazo.

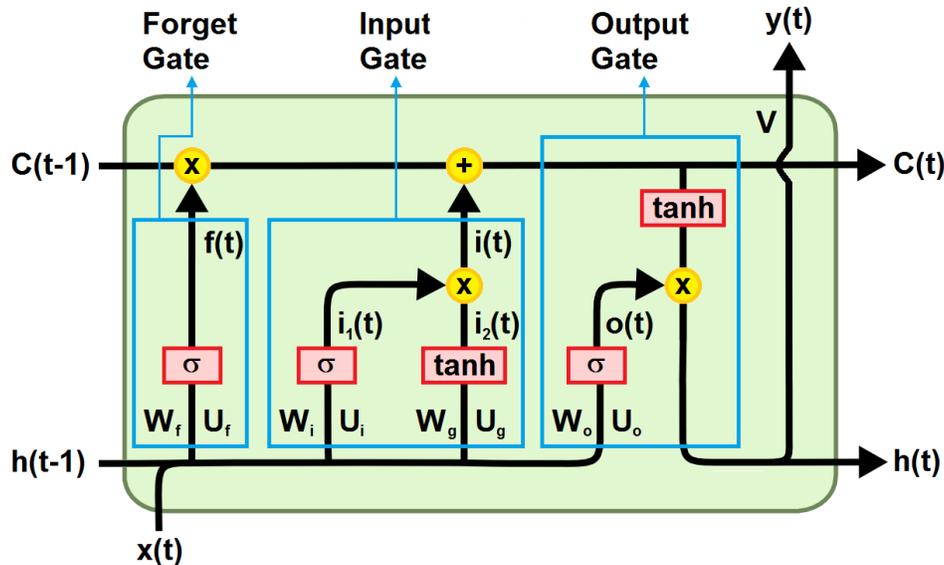


**Figura 7:** Arquitectura de red neuronal de Memoria larga a corto plazo

[Fuente: Pagina Web “<https://towardsdatascience.com/>

[time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc](https://towardsdatascience.com/time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc)”].

## Celda LSTM



**Figura 8:** Estructura de la celda unitaria de red neuronal recurrente simple

[Fuente: Pagina Web "<https://towardsdatascience.com>

[/time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc](https://towardsdatascience.com/time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc)"].

## Componentes principales del modelo

- 1) **Datos de entrada  $X(t)$ :** Corresponde a un conjunto de datos procedentes de una serie temporal de muestras de medición ordenados cronológicamente ligados a un rango de tiempo específico dentro de la sucesión temporal, siendo una fracción de la información de alimentación del modelo.
- 2) **Parámetros de pesos  $U, W, V$ :** Estas son las matrices de pesos que unen la puerta de salida del tiempo previo  $t-1$  y el vector de entrada del tiempo  $t$  con los componentes de la puerta de olvido del tiempo  $t$ , puerta de entrada del tiempo  $t$ , la puerta de salida del  $t$  y la respuesta de salida. El vector de entrada del tiempo  $t$  se conecta con las neuronas de su correspondiente puerta mediante una matriz de pesos  $U$ , donde la matriz  $U_f$  se relaciona con la puerta de olvido, las matrices  $U_i$  y  $U_g$  a la puerta de entrada y la matriz  $U_o$  a la puerta de salida, mientras la puerta de salida previa  $t-1$  se encuentra conectada a su correspondiente puerta mediante una matriz de pesos  $W$ , donde la matriz  $W_f$  se relaciona con la puerta de olvido, las matrices  $W_i$  y  $W_g$  a la puerta de entrada y la matriz  $W_o$  a la puerta de salida. Finalmente, una vez creada la salida de la puerta de salida  $h(t)$  esta se conecta con el vector de salida que entrega la unidad LSTM mediante una matriz de pesos  $V$ , cabe destacar que todas las matrices de peso aquí descritas son constantes para cada paso de tiempo y estos valores son actualizados durante el proceso de entrenamiento con el fin de mejorar la respuesta objetivo de la red neuronal.

- 3) **Estado de celda  $C(t)$ :** Corresponde al componente que tiene como función transferir la información relevante extraída durante el procesamiento de los datos, de forma que las características extraídas de los pasos de tiempo anteriores llegan a cada paso de tiempo posterior, reduciendo los efectos del desvanecimiento de gradiente mejorando la memoria a largo plazo de la red neuronal. Durante el entrenamiento de las redes LSTM, las puertas aprenden a determinar qué información es importante mantener para entregar una buena respuesta u olvidar para no dificultar esta acción, que posteriormente se agrega al estado de celda  $C(t)$  o se elimina de esta según corresponda, de esta manera se genera una recuperación de los datos transferidos en la memoria para paliar el desvanecimiento de gradiente.

$$C(t) = f(t) \cdot C(t - 1) + i(t)$$

- 4) **Puerta de olvido  $f(t)$ :** Corresponde al componente que tiene por objetivo decidir qué información procedente del pasado  $C(t-1)$  se debe eliminar o se debe mantener en la memoria futura de la red  $C(t)$ .

$$f(t) = \sigma(X(t) \cdot Uf + h(t - 1) \cdot Wf)$$

- 5) **Puerta de entrada  $i(t)$ :** Corresponde al componente responsable de la actualización del estado de celda  $C(t)$ , agregando nuevas características que se desprende del procesamiento de la información acumulada hasta ese instante  $t$ , es decir tanto del estado de celda anterior  $C(t-1)$  como del vector de entrada  $X(t)$ .

$$i1(t) = \sigma(X(t) \cdot Ui + h(t - 1) \cdot Wi)$$

$$i2(t) = \tanh(X(t) \cdot Ug + h(t - 1) \cdot Wg)$$

$$i(t) = i1(t) \cdot i2(t)$$

- 6) **Puerta de salida  $h(t)$ :** Corresponde al componente que tiene por labor decidir la relevancia que tendrá el siguiente estado de celda sobre la siguiente unidad LSTM.

$$o(t) = \sigma(X(t) \cdot Uo + h(t - 1) \cdot Wo)$$

$$h(t) = \tanh(C(t)) \cdot o(t)$$

- 7) **Respuesta de salida  $Y(t)$ :** Corresponde al vector de salida de la unidad LSTM actual que para su definición considera solamente características extraídas y desarrolladas hasta ese instante de tiempo  $t$ .

$$Y(t) = h(t) \cdot V$$

### **Ventajas de las redes de memoria larga a corto plazo:**

- a) **Desvanecimiento de gradiente reducido:** A partir de la nueva arquitectura que adoptan las unidades LSTM explícitamente diseñado para evitar el desvanecimiento de gradiente, ahora pueden almacenar la información desarrollada a partir de las muestras de entrada bajo dos distintos vectores de memoria; memoria de largo plazo  $C(t)$  y memoria de corto plazo  $h(t)$ , generando estabilidad numérica ante cambios en los parámetros de entrenamiento y mejorando el flujo de gradiente de la red.
- b) **Mayor memoria de largo plazo:** Gracias a la arquitectura de las puertas de entrada, olvido y salida, junto a los dos tipos de componentes de memoria, este tipo de red neuronal poseen una mayor capacidad para retener en su memoria las características extraídas de la información de entrada inicial, otorgándoles facilidad para recordar secuencias más largas y complejas, siendo capaces de modelar dependencias de secuencia a largo plazo.

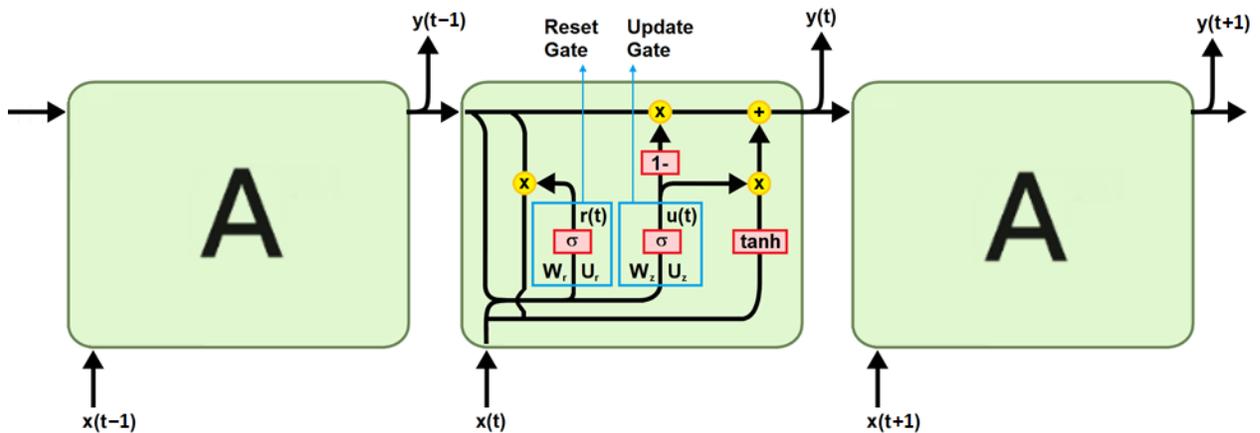
### **Desventajas de las redes de memoria larga a corto plazo:**

- a) **Aumento de la complejidad de cómputo:** Debido a su mayor complejidad estructural al poseer tres distintos tipos de puertas y dos tipos de memoria en su estructura, sumado a que la relación entre cada una de sus partes de la unidad LSTM están jerarquizadas bajo un bajo un orden secuencial, esto le confiere a cada unidad LSTM un mayor número de parámetros de pesos de entrenamiento y a su vez mayor tiempo de procesamiento de la información de entrada, que se traduce finalmente en extensos periodos de tiempo para finalizar cada iteración durante la etapa de entrenamiento de la red.
- b) **Mayores requerimientos de memoria computacional:** En comparación a las redes RNN, GRU y Transformadores, las redes LSTM poseen mayores requisitos en cuanto a la memoria del computador que ejecutará el programa, esto se debe principalmente a la gran cantidad de parámetros asociado a la unidad LSTM

## 2.4 Red neuronal unidad recurrente cerrada (GRU)

La unidad recurrente cerrada (GRU) corresponde a un tipo especial de red neuronal recurrente introducido en el paper “Gated Feedback Recurrent Neural Networks” [8] por Kyunghyun Cho el 17 de junio del 2015, este tipo de estructura de manera análoga a las redes tipo LSTM adoptan una nueva arquitectura para resolver el problema de desvanecimiento de gradiente con el objetivo de mejorar el proceso de entrenamiento de la red y mejorar la memoria a largo plazo de la red conservando las características procesadas de los datos iniciales, para esto la celda unitaria GRU se conforma de un estado oculto  $h(t)$  que sirve como almacenamiento de la memoria de la red más dos tipos de puertas diferentes; la puerta de reinicio  $r(t)$  y la puerta de actualización  $u(t)$ , encargados respectivamente de decidir cuanta información de los pasos de tiempo previos deben olvidarse o preservarse en la memoria de la red.

Las unidades GRU se agrupan bajo un orden secuencial para procesar los datos de entrada de forma fragmentada como se puede ver en la Figura 9, orden en el cual la unidad GRU del tiempo  $t$  se encuentra interrelacionado a la unidad del tiempo previo GRU  $t-1$  y a la unidad GRU del tiempo posterior  $t+1$  por medio del estado oculto previo  $h(t-1)$  y el estado oculto actual  $h(t)$  conectados a través de una matriz de pesos  $W(t)$ , por otro lado cada unidad GRU está conectado por medio de una matriz de pesos  $U(t)$  a su propio vector de entrada de tiempo  $t$  asignado, tanto la matriz de pesos  $W(t)$  como la matriz  $U(t)$  se dividen en las submatrices  $W_r(t)$  y  $W_z(t)$  relacionada a la puerta de reinicio mientras las submatrices  $W_z(t)$  y  $U_z(t)$  se relacionan a la puerta de actualización, finalmente cabe destacar que cada celda GRU se encarga de generar su propia respuesta  $Y(t)$  aplicando una matriz de pesos  $V(t)$  sobre el estado oculto  $h(t)$ .

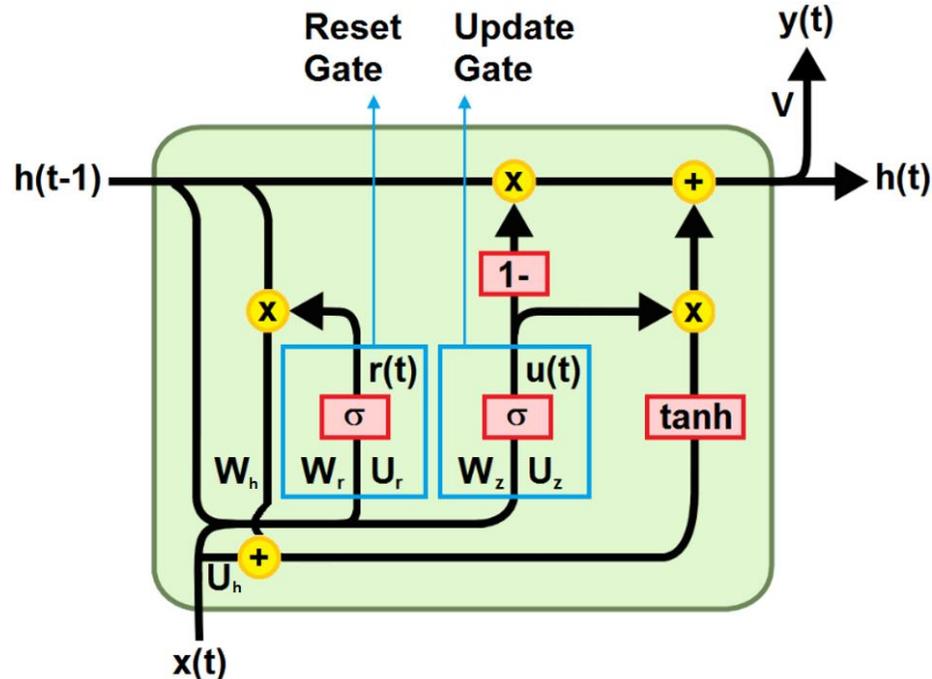


**Figura 9:** Arquitectura de red neuronal unidad recurrente cerrada

[Fuente: Pagina Web “<https://towardsdatascience.com/>

*time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc”*].

## Celda GRU



**Figura 10:** Estructura de la celda unitaria de las redes de unidad recurrente cerrada

[Fuente: Pagina Web "<https://towardsdatascience.com/>

*time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc*"].

## Componentes principales del modelo

- 1) **Datos de entrada  $X(t)$ :** Corresponde a un conjunto de datos procedentes de una serie temporal de muestras de medición ordenados cronológicamente ligados a un rango de tiempo específico dentro de la sucesión temporal, siendo una fracción de la información de alimentación del modelo.
- 2) **Parámetros de pesos  $U, W, V$ :** Estas son las matrices de pesos que unen el estado oculto previo y el vector de entrada del tiempo actual con los componentes de la puerta de reinicio del tiempo  $t$ , la puerta de actualización del tiempo  $t$ , la memoria actual y la respuesta de la unidad. El vector de entrada del tiempo  $t$  se conecta con las neuronas de su correspondiente puerta mediante una matriz de pesos  $U$ , donde la matriz  $U_r$  se relaciona con la puerta de reinicio y la matriz de pesos  $U_z$  se relaciona con la puerta de actualización, mientras el estado oculto anterior  $h(t-1)$  se encuentra conectada a su correspondiente puerta mediante una matriz de pesos  $W$ , donde la matriz  $W_r$  se relaciona a la puerta de reinicio y la matriz  $W_z$  se relaciona a la puerta de actualización. Mientras tanto para definir la memoria actual  $h(t)$  se utilizan otras dos matrices de pesos  $U_h$  y  $W_h$ , las cuales

se aplican correspondientemente sobre el vector de entrada  $X(t)$  y el vector de salida de la puerta de reinicio  $r(t)$ . Finalmente, una vez creado el estado oculto actual  $h(t)$  esta se conecta con el vector de salida que entrega la unidad GRU mediante una matriz de pesos  $V$ , cabe destacar que todas las matrices de peso aquí descritas son constantes para cada paso de tiempo y estos valores son actualizados durante el proceso de entrenamiento con el fin de mejorar la respuesta objetivo de la red neuronal.

- 3) **Memoria previa  $h(t-1)$ :** Corresponde al vector de memoria de los pasos de tiempo previos que almacena las características más relevantes de las muestras temporales anteriores del vector de entrada, tales características representan toda información que impacte en la definición de una buena respuesta del modelo de red neuronal, esta es la única conexión que existe entre la unidad GRU actual con la unidad GRU predecesora.

- 4) **Memoria actual  $\dot{h}(t)$ :** Corresponde a un vector de memoria preliminar que almacena la información filtrada del pasado a la que se le aplicó el proceso de supresión de características que no contribuyen a la definición de una buena respuesta.

$$\dot{h}(t) = \tanh( X(t) \cdot U_h + ( r(t) \cdot h(t - 1) ) W_h )$$

- 5) **Puerta de reinicio  $r(t)$ :** Corresponde a la sección de la unidad GRU responsable de determinar cómo combinar los nuevos datos de entrada con la memoria previa, determinando cuanta información procesada de los pasos de tiempo anteriores se debe olvidar.

$$r(t) = \sigma( X(t) \cdot U_r + h(t - 1) \cdot W_r )$$

- 6) **Puerta de actualización  $u(t)$ :** Corresponde a la estructura de la unidad GRU que posee la función de determinar cuanta información procedente de los tiempos anteriores vale la pena transferir hacia la memoria futura y que posteriormente se agregará al vector de memoria  $h(t)$ .

$$u(t) = \sigma( X(t) \cdot U_z + h(t - 1) \cdot W_z )$$

- 7) **Memoria final  $h(t)$ :** Es el vector de memoria que almacena la información relevante para desarrollar una correcta respuesta, adquirida desde los pasos de tiempo previos hasta el paso de tiempo actual que posteriormente será transmitida a la siguiente unidad GRU, este vector de memoria  $h(t)$  es la única conexión existente entre la unidad GRU actual y la siguiente unidad GRU.

$$h(t) = ( 1 - u(t) ) \cdot h(t - 1) + u(t) \cdot \dot{h}(t)$$

- 8) **Respuesta de salida  $Y(t)$ :** Corresponde al vector de salida de la unidad GRU actual que para su definición considera solamente características extraídas y desarrolladas hasta ese instante de tiempo  $t$ .

$$Y(t) = h(t) \cdot V$$

### **Ventajas de las redes unidad recurrente cerrada:**

- a) **Desvanecimiento de gradiente reducido:** Las redes GRU introducen una nueva arquitectura para combatir el desvanecimiento de gradiente que sufren las RNN, utilizando una estructura muy similar a las puertas que conforman las redes LSTM, pero en una versión más simplificada y compacta, quitando el vector de memoria de largo plazo, a cambio de utilizar solo un estado oculto como almacenamiento. De acuerdo con las pruebas realizadas con este tipo de redes se ha demostrado que gracias a esta nueva arquitectura simplificada han logrado una estabilidad numérica durante el entrenamiento muy similar a las redes LSTM.
  
- b) **Mayor memoria de largo plazo:** Gracias a la arquitectura de las puertas de reinicio y de actualización, junto a su estado oculto que utiliza como almacenamiento de memoria, las redes GRU poseen una gran capacidad de almacenamiento y un mayor grado de desarrollo de características a partir de extensas series temporales de entrada, disminuyendo la pérdida de información de las características que se pueden extraer del inicio del vector de entrada.

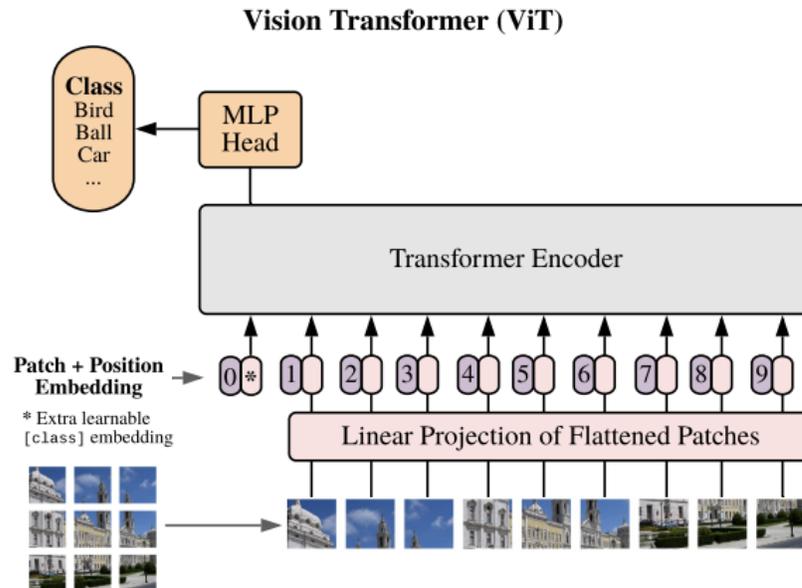
### **Desventajas de las redes unidad recurrente cerrada:**

- a) **Bajo rendimiento de entrenamiento:** A partir de la arquitectura simplificada de las redes GRU, este tipo de red se desempeña bajo una tasa de convergencia lenta y una baja eficiencia de aprendizaje, lo que resulta en que requieran una mayor cantidad de iteraciones durante el proceso de entrenamiento (épocas) para poder definir un modelo robusto, reflejando mayores periodos de tiempo para entrenar la red.

## 2.5 Visión de transformador (ViT)

La visión de transformador corresponde a una arquitectura de una red neuronal de tipo transformador que se especializa en el campo de reconocimiento de imágenes que se introdujo en el paper “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale” [5] por Google Research y Brain Team publicado el 22 de octubre del año 2020, en el documento se describe una red neuronal de clasificación que recibe como datos de entrada una imagen a color presentando un objeto que posteriormente la red neuronal procesará para seleccionar dentro de un lista de categorías definidas a qué clase corresponde el objeto contenido en la imagen.

La arquitectura de la visión de transformador como se puede observar en la Figura 11 está compuesta por una primera capa de subdivisión de la imagen inicial en múltiples parches cuadrados (16 x 16 parches), luego esta misma capa extrae las características más importantes que contiene cada parche almacenándolo en un vector lineal diferente para cada uno de los parches (256 vectores), la segunda capa corresponde a una capa de proyección lineal donde toma los vectores de características y los incrusta en un nuevo espacio vectorial de diferente dimensión, la tercera capa posicional es la encargada de agregar el vector de posición al vector de características de la nueva dimensión que sirve como identificación de la ubicación exacta del parche en la imagen original, la cuarta capa corresponde a un codificador de transformador que busca resaltar la información importante contenida en cada vector suprimiendo aquella menos útil para luego desarrollar nuevas características globales de todos los parches que serán almacenadas dentro de un único vector de latencia de menor dimensión, luego el vector de latencia ingresa a la quinta capa de multiperceptrón que se encarga de procesar y refinar aún más la información extraída con el fin de entregar un vector de probabilidades que representa la relación de la imagen con cada una de las clases existentes, finalmente en la salida del modelo se selecciona aquella clase con mayor valor de probabilidad.



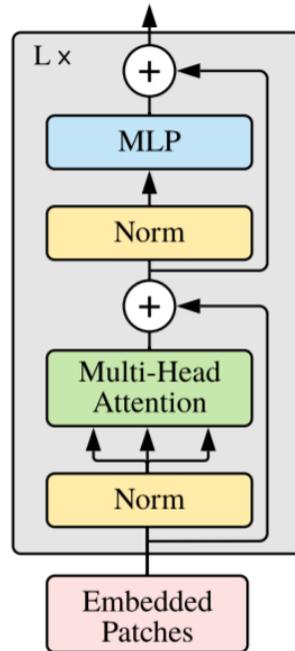
**Figura 11:** Arquitectura de red neuronal de la visión transformador

[Fuente: Paper “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”].

## **Componentes principales del modelo**

- 1) **Datos de entrada  $X(t)$ :** Corresponde a una imagen a color de resolución de 224 píxeles x 224 píxeles, la cual expone un objeto relacionado a una de las clases relacionadas al banco de datos JFT-300M o ImageNet21K, los cuales son un conjunto de imágenes utilizando para el entrenamiento de modelos de aprendizaje profundo para tareas de visión artificial.
- 2) **Capa de recorte de imagen:** Es la sección encargada de fragmentar la imagen en recuadros de tamaño de 14 píxeles x 14 píxeles resultando en la creación de 16 x 16 parches, a partir de los cuales mediante una capa lineal de nodos se extraen las características más importantes de cada parche y luego son guardadas en sus correspondientes vectores de características.
- 3) **Capa de proyección lineal:** La función de esta capa es transformar el vector de características original en un nuevo vector de características con una dimensión diferente, refinando la información contenida en el vector.
- 4) **Capa de posicionamiento:** Es el componente que tiene por función darle a cada vector de características el contexto de la ubicación de su respectivo parche en la imagen original, para eso se agrega al vector de característica un vector posicional que contiene la información tanto de la ubicación del propio parche como también la relación espacial con el resto de los parches.
- 5) **Codificador de transformador:** Corresponde al componente que se encarga del desarrollo de nuevas características a partir de los todos los vectores de características de cada parche, sintetizándola en único vector de latencia que se destaca por poseer una dimensión menor a los vectores de entrada comprimiendo la información procesada a su estado fundamental, el codificador de transformador como se puede observar en la Figura 12 consta de múltiples bloques de atención multicabeza, normalización y capas multiperceptrón completamente conectadas con conexiones residuales (de salto), cada una estas capas cumple respectivamente la función de resaltar la información contenida en los vectores de características que contribuyen a elaborar a una buena respuesta, suprimiendo aquellos datos de entrada menos útiles, como también la función de regular los valores de los datos desarrollados para que adopten una forma mucho más fácil de manipular a la hora de optimizar la red y para desarrollar nuevas características que sean esenciales para definir una correcta respuesta.

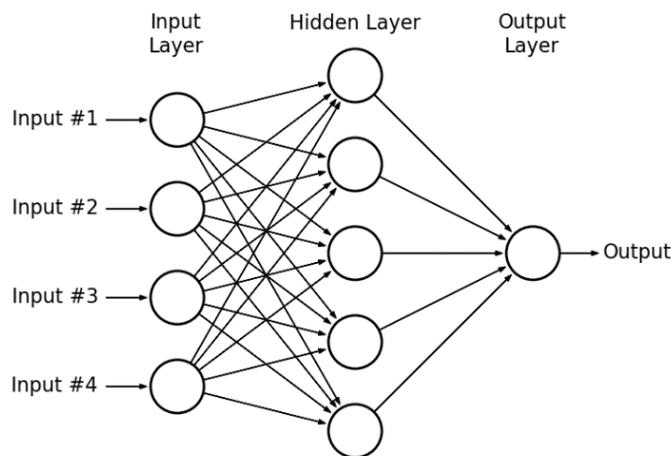
**Transformer Encoder**



*Figura 12: Estructura de codificador de transformador*

[Fuente: Paper “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”].

- 6) **Capa multiperceptrón:** Corresponde a una red de clasificación que selecciona la clase a la que corresponde el objeto contenido en la imagen de entrada a partir del vector de latencia que contienen las características desarrolladas por codificador, esta sección se compone de múltiples capas dense que decrecen en número a medida que se avanza en la red con el fin de discriminar entre todas las categorías de clases existentes o etiquetas a cuál corresponde la imagen.



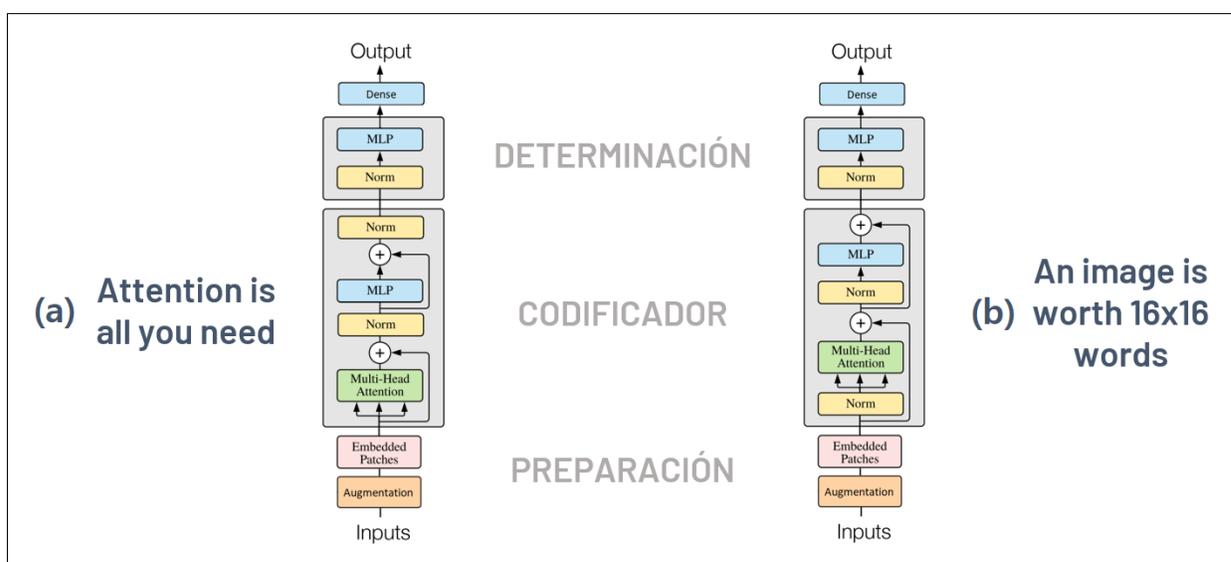
*Figura 13: Estructura nodal de perceptrón multicapa*

[Fuente: Paper “Assessment of artificial neural network for bathymetry estimation using high resolution satellite imagery in shallow lakes: case study el Burullus lake” [6] ].

### 3 Propuesta de diseño de modelos neuronales

#### 3.1 Arquitecturas base propuestas

A continuación, se muestra un esquema representativo de las dos arquitecturas base propuestas para el estudio a partir de las cuales se constituyen los modelos neuronales a prueba de tipo Transformador, LSTM, GRU y RNN.



**Figura 14:** Esquemas representativos de las arquitecturas neuronales propuestas (a) Arquitectura propuesta “Attention is all you need” (b) Arquitectura propuesta “An image is worth 16x16 words”

[Fuente: Elaboración Propia].

En la Figura 14.a se observa la arquitectura propuesta “Attention is all you need”, la cual corresponde a una arquitectura construida fundamentalmente en base a la estructura, organización y configuración del diseño original descrito en el paper fuente “Attention is all your need” que proporcionó el concepto de los modelos neuronales transformadores aplicándolos sobre el campo del procesamiento del lenguaje natural, caracterizando al modelo descrito por ser especialista en extraer características del orden secuencial de los datos de entrada y asociación de elementos.

Por otra parte, la Figura 14.b muestra la arquitectura “An image is worth 16x16 words”, la cual se basa principalmente en el paper fuente “An image is worth 16x16 words” que busca trasladar el concepto de los modelos transformadores al campo de la óptica de identificación de objetos a partir de imágenes y fotografías, destacando el rendimiento de estos modelos por extraer características de cada pixel que conforma la imagen (o secciones que conforman la fotografía) a la par de extraer información de la relación espacial que tiene con el resto de los pixeles que le rodean (o secciones que le rodea).

Ambas arquitecturas propuestas se subdividen en 3 grandes bloques de acuerdo con su función dentro de la estructura:

- I) **Bloque de preparación:** esta sección tiene el objetivo de ejecutar un pretratamiento sobre las ventanas temporales de entrada para facilitar el aprendizaje automático de los modelos, se encuentra compuesta por una capa de aumento que tiene por función ser la primera medida para evitar el sobreajuste durante el entrenamiento aplicando una normalización, giros de imagen en 90° y rotación sobre el eje Y sobre algunas muestras de entrada de forma aleatoria, seguida de esta capa se encuentra la capa de parches incrustados que consiste en una secuencia de capas de recorte, proyección lineal y posicionamiento, tiene por función generar recortes temporales a partir de la imagen original a la par de extraer las características útiles contenidas en ellas, e identificarlas espacialmente con un nuevo vector de mayor dimensión que guarda tanto la información de la ubicación original del parche dentro de la imagen temporal inicial como también la relación y características con el resto de los parches que lo rodean.
- I) **Bloque de codificación:** esta sección cumple la función de extraer y comprimir la información más útil para la elaboración de una buena respuesta, esta sección se compone de dos capas normalizadoras que cumplen el rol de suavizar gradientes permitiendo entrenamientos más rápidos y facilitando una mejor precisión de generalización, seguida por una capa característica conformada por unidades características (en la Figura representada por la capa de atención multicabezal compuestas de unidades de mecanismos de atención para el caso del modelo de tipo transformador), y por último integra una capa multiperceptrón que cumple el rol de retroalimentación y refinamiento de la información de procesada.
- I) **Bloque de determinación:** esta sección cumple el rol de definir la respuesta del modelo, siendo la predicción del estado de operación del equipo para el caso del modelo de clasificación ó estimando la vida útil remanente del equipo para el caso del modelo regresión, se compone de una capa de normalización que estandariza la información procesada proveniente del codificador, seguida de una capa multiperceptrón que refina aún más la información extraída y genera nuevas características, para finalmente enviarle esos datos a una última capa Dense, responsable de entregar la respuesta final del modelo.

Cabe destacar que ambas arquitecturas propuestas poseen los mismos componentes y elementos (número, cantidad de nodos, tipo y forma de capas) como también similares tipos de conexiones (en serie, paralelos y residual), sin embargo, lo que los hace distintos entre sí es una diferente organización en cuanto al orden de las capas normalizadoras de la sección del codificador del modelo y los puntos que une la conexión residual.

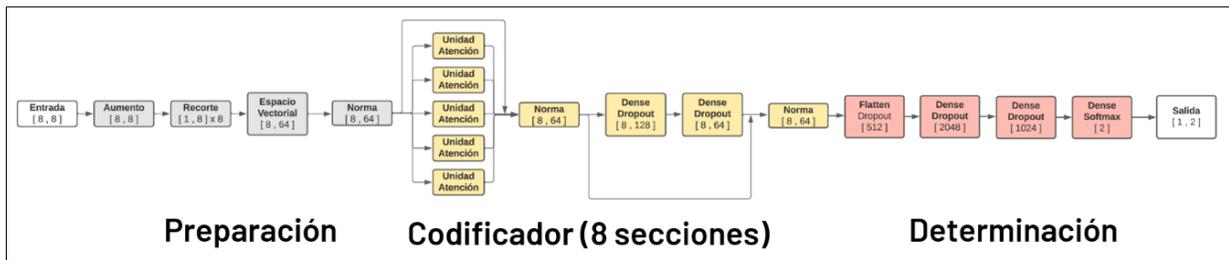
En la arquitectura “Attention is all you need” el codificador se componen en primer lugar por una capa característica (representado en la imagen para el caso particular del transformador como una capa de atención multicabeza), que posteriormente a su producto se le agrega la información previa a esta última capa por medio de una conexión residual, en segundo lugar se encuentra una capa normalizadora, luego en tercer lugar se encuentra una capa de retroalimentación de tipo perceptrón multicapa (MLP), que posteriormente a su producto se le agrega la información previa a esta última capa por medio de una conexión residual, y finalmente en cuarto lugar se encuentra una capa de normalización.

En cambio, en la arquitectura “An image is worth 16x16 words” el codificador se componen en primer lugar por una capa normalizadora, seguido en segundo lugar por una capa característica a la que posteriormente se le añade al producto de esta última capa una conexión residual de la información previa a la capa normalizadora, luego en tercer lugar se encuentra una capa normalizadora, seguido en cuarto lugar por una capa de retroalimentación de tipo perceptrón multicapa (MLP) a la que posteriormente a su producto se le agrega una conexión residual proveniente de la información extraída previo al paso por la última capa normalizadora.

### 3.2 Estructura de modelos neuronales

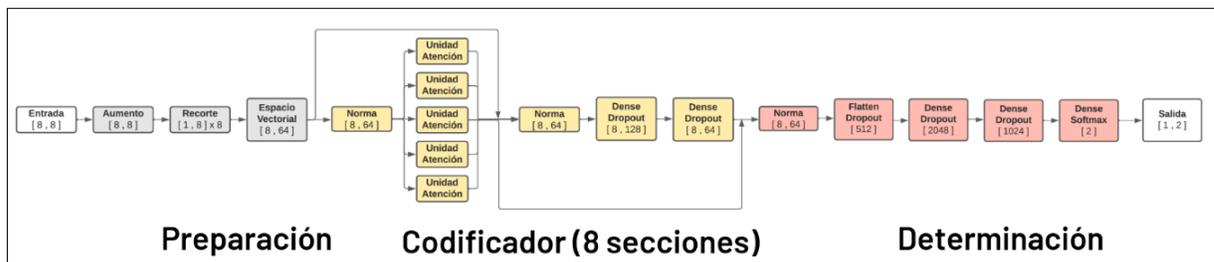
A continuación, se describen con mayor profundidad los elementos tensoriales que conforman cada una de las capas de las arquitecturas previamente expuestas en la sección anterior tanto para el caso de clasificación como también para el caso de regresión:

#### Estructura para modelo neuronales de clasificación



*Figura 15: Modelo de clasificación bajo arquitectura “Attention is all you need”*

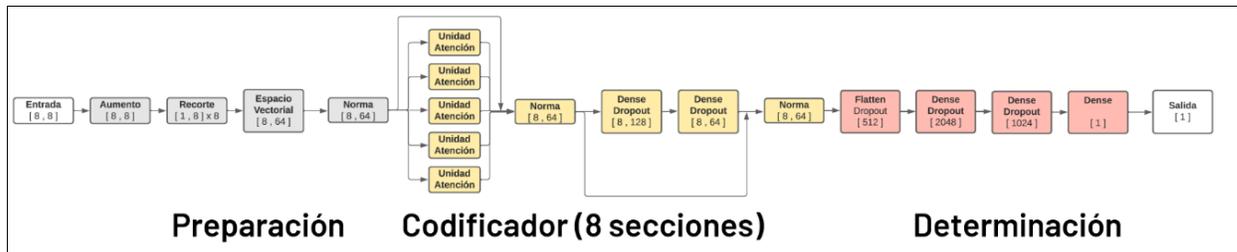
[Fuente: Elaboración Propia].



*Figura 16: Modelo de clasificación bajo arquitectura “An image is worth 16x16 words”*

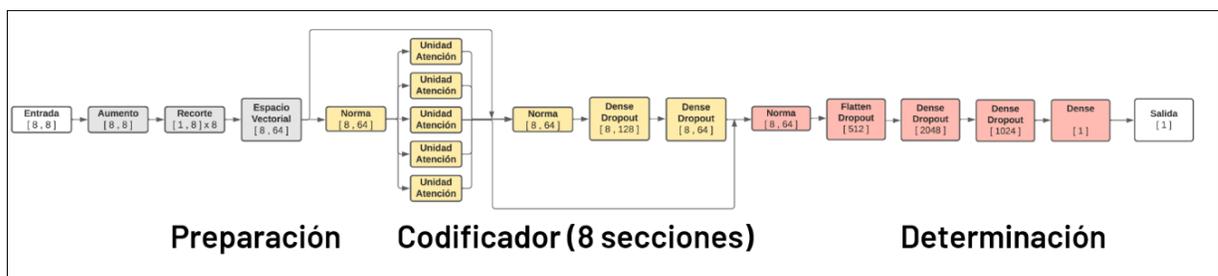
[Fuente: Elaboración Propia].

## Estructura para modelos neuronales de regresión



*Figura 17: Modelo de regresión bajo arquitectura "Attention is all you need"*

[Fuente: Elaboración Propia].



*Figura 18: Modelo de regresión bajo arquitectura "An image is worth 16x16 words"*

[Fuente: Elaboración Propia].

De acuerdo con las figuras previamente expuesta, se puede observar que el bloque de preparación funciona dentro del modelo solo una única vez, a diferencia del bloque del codificador que se reitera una cantidad de 8 veces seguidas bajo un orden estrictamente en serie, para luego entregar la información procesada al bloque de determinación que al igual que el primer bloque no se repite.

Sobre la base de la Figura 17 y Figura 18 se desprende que el formato de entrada del modelo consiste en ventanas temporales de tamaño [8,8], que al ingresar al bloque de preparación mantiene sus dimensiones en la capa de aumento, sin embargo, luego estas son recortadas en 8 parches de dimensiones [1,8] en la capa de recorte, para más tarde dentro de la capa de espacio vectorial cada parche se convierte en un vector de dimensiones [1,64], que agrupándolos todos en una matriz nos entrega un tensor de dimensiones [8, 64], luego al ingresar al bloque codificador se mantienen las dimensiones del tensor hasta la primera capa Dense que conforma el multiperceptrón donde se expanden las dimensiones del tensor a [8,128] y en la segunda capa Dense que integra al multiperceptrón el tensor vuelve a las dimensiones previas [8,64], luego al ingresar al bloque de determinación se linealiza el formato del tensor bajo una lista de valores numéricos de dimensión [512], el cual es expandido en la primera capa Dense a [2048] elementos, luego en la segunda capa Dense es reducido a una lista de [1024] y finalmente la información procesada es enviada a una última capa Dense, que según sea el caso, para un modelo de clasificación corresponde a una capa dense de dos nodos con función de activación Softmax que entrega un vector de salida de dimensiones [1,2], mientras para el modelo de regresión corresponde a una capa Dense de un solo nodo con función de activación lineal que entrega un vector de dimensión de dimensiones [1].

### 3.3 Selección inicial de hiperparámetros

Para la definición de la arquitectura “attention is all you need” y la arquitectura “An image is worth 16x16 words” se seleccionaron los siguientes valores iniciales de hiperparámetros para el proceso de entrenamiento, validación, prueba y evaluación de los modelos neuronales propuestos.

*Tabla 1: Selección de hiperparámetros de modelos neuronales propuestos.*

Hiperparámetro	Modelo de clasificación	Modelo de regresión
Formato entrada	Ventanas temporales	Ventanas temporales
Formato salida	One hot encoding	Rango unitario [0,1]
Número de parámetros	4.000.000	4.000.000
Distribución de datos	Training = 70% Validation = 15% Test = 15%	Training = 70% Validation = 15% Test = 15%
Tipo de optimizador	AdamW	Adam
Tasa de aprendizaje	0.001	0.001
Decaimiento de pesos	0.0001	-
Número de épocas	50	100
Lote de entrenamiento	256	256
Dimensión proyección	64	64
Función de pérdida	Maximizar exactitud	Minimizar RMSE

### 3.4 Composición de capa características por tipo de modelo

En las secciones anteriores las arquitecturas representadas se enfocaban principalmente de exponer la estructura del modelo neuronal de tipo transformador, sin embargo, bajo esta misma secuencia de componentes se conforman también el resto de los modelos LSTM, GRU y RNN, con la única diferencia en que difiere la cantidad de celdas distintivas que conforman su respectiva capa característica que funcionan entre sí bajo un orden estrictamente en paralelo, debido a que cada celda distintiva representa un cantidad discreta fija de parámetros entrenables es que con el objetivo de mantener una similar cantidad de parámetros entrenables entre los modelos Transformador, LSTM, GRU y RNN, es que se determinó que cada tipo de modelo dispone de una cantidad diferente de celdas distintivas que conforman su capa característica. A continuación, en la siguiente tabla se expone la cantidad de celdas distintivas que conforma cada capa característica del codificador según el tipo de modelo neuronal:

*Tabla 2: Número de celdas distintivas de la capa característica según tipo de modelo.*

Tipo de modelo neuronal	Unidades por capa característica
Transformador	5 celdas de mecanismos de atención
LSTM	3 celdas de memorias a corto largo plazo
GRU	3 celdas de unidades recurrentes cerradas
RNN	7 celdas de redes neuronales recurrentes

### 3.5 Análisis de sensibilidad

De acuerdo al plazo de tiempo designado para efectuar el presente trabajo de título se consideró que, para efectuar un análisis de sensibilidad sobre cada uno de los cuatro tipos de modelos neuronales bajo estudio de tipo transformador, LSTM, GRU y RNN respectivamente, se decidió por evaluar el impacto que implica un cambio en el número de parámetros netos que conforma cada modelo sobre el desempeño y rendimiento logrado en las tareas de clasificación y regresión, más aún con el fin de acotar las posibilidades que este término representa; el análisis de sensibilidad se centrará específicamente en modificar el número de codificadores que forman parte de la sección central tanto de la primera arquitectura propuesta “Attention is all you need” como de la segunda arquitectura propuesta “An image is worth 16x16 words”, por lo tanto a razón de cumplir con este objetivo se requiere de la construcción, entrenamiento, prueba y evaluación de nuevas arquitecturas alternativas o modelos variantes basados de los modelos originales de tipo transformador, LSTM, GRU y RNN previamente diseñados, con la única diferencia que estos nuevos modelos disponen en su estructura una diferente cantidad de codificadores en su sección central, entre aquellos grupos de codificadores que se podrán a prueba están; 2 codificadores, 4 codificador, 6 codificadores, 8 codificadores y 10 codificadores.

### 3.6 Evaluación de modelo propuestos

Una vez codificados los modelos neuronales de tipo Transformador, LSTM, GRU y RNN bajo las arquitecturas “Attention is all you need” y “An image worth 16x16 words”, se evaluará el desempeño logrado de cada modelo por su calidad de proceso de entrenamiento como también el desempeño obtenido durante la fase de pruebas bajo tareas de predicción de estados operativos por clasificación y estimación del RUL por regresión, para el caso de clasificación se utilizará la evolución de la función de pérdida durante el entrenamiento, la matriz de confusión típica y la usual curva característica del receptor, junto a la evaluación de los indicadores de rendimiento de Exactitud, Precisión, Exhaustividad, Puntaje Valor F1 y Área bajo la curva ROC, mientras por otro parte para el caso de regresión se utilizará la evolución de la función de pérdida durante el entrenamiento, diagramas de dispersión de valor predicho frente a valor real y Gráficos de dispersión ordenados de valor predicho frente a valor real, junto a evaluar los indicadores de rendimiento del error cuadrático medio, error absoluto medio y coeficiente de determinación.

#### 3.6.1 Indicadores de rendimiento

A continuación, se detalla una descripción sobre cada indicador de rendimiento a través de los cuales se evaluará el desempeño logrado por cada modelo neuronal diseñado:

#### Indicadores de rendimiento de clasificación

- I. **Caja de confusión:** Es una herramienta que permite la visualización explícita de dos clases (positivo y negativo) predichas por el modelo neuronal frente a las clases verdaderas que indican las muestras, exhibiéndolos en 4 grandes subtipos:

- **Verdaderos positivos (TP):** casos positivos que el modelo identifica como tales.
- **Verdaderos negativos (TN):** casos negativos que el modelo identifica como tales
- **Falsos positivos (FP):** casos positivos que el modelo identifica como negativos.
- **Falsos negativos (FN):** casos negativos que el modelo identifica como positivos

II. **Exactitud:** Corresponde a la relación de las cantidades de predicciones verdaderas (verdaderos positivos y verdaderos negativos) frente a la cantidad total de muestras evaluadas. Este indicador mide el porcentaje de casos que el modelo acierta.

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN}$$

III. **Precisión:** Corresponde a la relación de las cantidades verdaderos positivas frente a la cantidad total de positivos (verdadero positivo y verdadero negativo), representa la calidad del modelo de machine learning, exponiendo la proporción de casos positivos que el modelo puede identificar del total.

$$Precisión = \frac{TP}{TP + FP}$$

IV. **Exhaustividad:** Corresponde a la relación de verdaderos positivos frente a la cantidad de verdaderos positivos y falsos negativos, que nos indica la probabilidad de que una predicción positiva del modelo sea cierta.

$$Exhaustividad = \frac{TP}{TP + FN}$$

V. **Puntaje Valor F1:** Corresponde a la media armónica entre la precisión y la exhaustividad, se utiliza para representar el rendimiento combinado de los indicadores de precisión y exhaustividad.

$$Puntaje\ valor\ F1 = 2 \left( \frac{Precisión \cdot Exhaustividad}{Precisión + Exhaustividad} \right)$$

VI. **Área bajo la curva ROC (AUC):** Representa la capacidad discriminativa del modelo neuronal para discernir sobre las distintas clases, para este caso de estudio sobre los estados operacionales nominal y anormal. Se define por un valor entre 0 y 1 que representa la probabilidad general de que la predicción del modelo sea acertada.

$$AUC = \int_0^1 ROC(r) dr$$

### **Indicadores de rendimiento de regresión**

- I. Error cuadrático medio (RMSE):** Corresponde al promedio de los errores elevados al cuadrado y representa que tan bien un modelo de regresión puede ajustarse a un conjunto de datos, se destaca por penalizar más fuertemente los valores atípicos que incurren en valor de errores mayor.

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T (Y_t - \hat{Y}_t)^2}$$

- II. Error absoluto medio (MAE):** Corresponde al promedio de diferencia absoluta entre el valor objetivo y el valor predicho por el modelo, es más robusto para los valores atípicos y no penaliza los errores tan extremadamente como el RMSE.

$$MAE = \frac{1}{T} \sum_{t=1}^T |Y_t - \hat{Y}_t|$$

- III. Coeficiente de determinación (R<sup>2</sup>):** Corresponde a la proporción de la varianza total de la variable estimada por la regresión, representa la bondad del ajuste del modelo a la variable que se busca estimar, presenta la ventaja de estar libre de escala en comparación al RMSE y MAE.

$$R^2 = \frac{\sum_{t=1}^T (\hat{Y}_t - \bar{Y})^2}{\sum_{t=1}^T (Y_t - \bar{Y})^2}$$

## 4 Metodología

Para el desarrollo del tema de trabajo de título se procede a ejecutar los siguientes pasos:

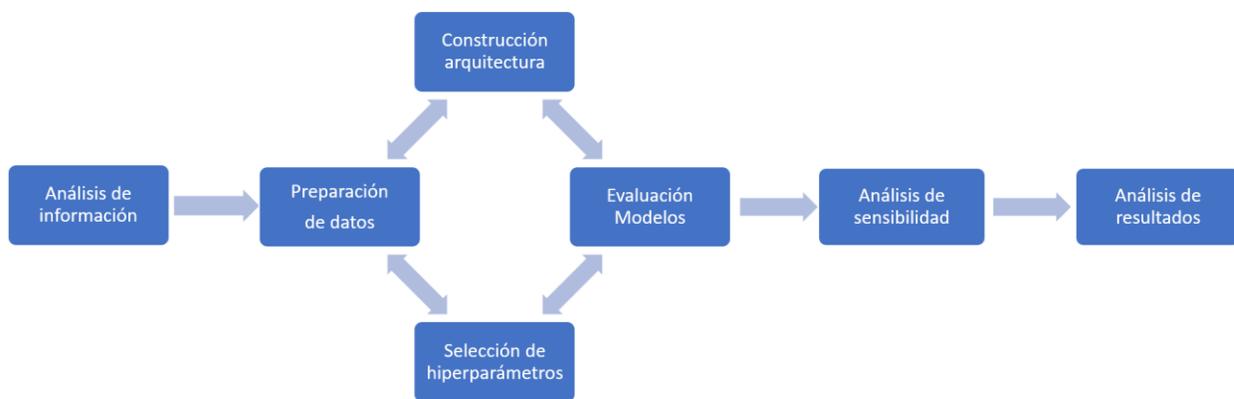
- 1) **Análisis de la información:** En esta etapa se efectúa un análisis de la información contenida en el banco de datos “Reboiler\_Raw Data\_With Timestamps” bajo el formato Excell provisto por el profesor guía Enrique López Droguett, primero se separan las variables de entrada y salida destinadas para el modelo neuronal, después se observa el comportamiento que poseen individualmente cada una de las variables de estudio por sí solas a lo largo del registro, posteriormente se verifica que el contenido de los datos se encuentren completos para cada una de las variables de operación como también que no falte ninguna etiqueta de estado asignado a cada instancia de tiempo de lo contrario se deben completar los datos ausentes con valores razonables dentro de la continuidad temporal de las variables, luego se procede a determinar el promedio, la desviación estándar, valor mínimo, valor máximo de cada una de las variables de operación medidas incorporadas en el Dataset de alimentación de la red neuronal.
- 2) **Acondicionamiento de datos:** En esta etapa inicialmente se sube el archivo Excel a un repositorio online en GitHub con el objetivo de que posteriormente a partir de este reservorio en línea se puedan cargar los datos contenidos en el documento Excel dentro del programa Python empleando únicamente la dirección online y el nombre de la carpeta en GitHub. Una vez cargada la información en el programa Python del ambiente Google Colab utilizando las extensiones Pandas y Numpy se normalizan los valores de las 8 columnas de las variables de operación del banco de datos bajo una escala de valores en el rango [0,1] utilizando el estimador MinMaxEscaler de la biblioteca Scikit-learn, como también se establece el formato de “ventana temporal” para la información de entrada del modelo neuronal definiendo el rango de tiempo que abarcará la matriz de los 8 parámetros de funcionamiento mientras a su vez se asigna cada ventana temporal su correspondiente etiqueta de estado operacional representativo del periodo de tiempo abarcado, por otro lado se modifica el formato binario de los estados operacionales a un formato de tipo one hot encoding para las etiquetas del modelo neuronal utilizando la función `to_categorical` de la biblioteca de Keras de TensorFlow, por último a partir de estos conjuntos se conforman los grupos de datos de entrenamiento, validación y prueba, verificando que tales grupos poseen una distribución uniforme de las etiquetas de estados (nominal y defectuoso).
- 3) **Selección de hiperparámetros:** Se define un primer límite del número de parámetros entrenables al que apuntan los modelos neuronales a construir, se fijan las dimensiones de las ventanas temporales de las muestras de entrada, se establece el número de clases a predecir por el modelo junto a la escala numérica de salida del RUL, se fijan las dimensiones de los parches temporales que serán procesados por separado y en paralelo por el codificador del transformador, se establece el número de dimensiones que tiene el espacio vectorial de la capa de proyección lineal, se selecciona un optimizador escogiendo también su frecuencia de aprendizaje y el decaimiento de pesos recomendados, se seleccionan las métricas de rendimiento y la función de pérdida a utilizar para calcular el error tanto para el caso de clasificación como para el caso de regresión, se establece la configuración de los puntos de control (función de monitoreo) para guardar el mejor modelo del entrenamiento, se define el tamaño de lotes (Batch size) de muestras de entrenamiento a partir del cual se calcula el error y

posteriormente se actualizan los pesos, se fijan la cantidad de épocas a ejecutar por ciclo de entrenamiento.

- 4) **Construcción de arquitectura:** Se construye un esquema conceptual de la arquitectura global de la red neuronal, utilizando la biblioteca Keras de Python se codifica la capa de "aumento" que evita el sobre ajuste durante el entrenamiento, se codifica la capa de recorte de parches de la imagen original, se codifica la capa de proyección lineal de los parches, se diseñan las conexiones entre cada una de las capas que conforman la red seleccionando entre conexiones de tipo en serie, paralelo o residuales, se define el número de repeticiones secuenciales de cada tipo de bloque de capas de la red neuronal, se definen las capas normalizadoras de los datos ubicadas a lo largo de la red, se especifica el número de celdas unitarias que componen la capa característica que diferencia el funcionamiento entre los cuatro modelos (transformador, RNN, GRU y LSTM), se codifica la función cronómetro que almacenará el tiempo que lleva la ejecución del entrenamiento de cada modelo, se establece la distribución y la configuración de las capas Dense y las capas Dropout que forman parte del multiperceptrón abarcando los porcentaje de pérdida, función de activación y la cantidad de nodos por cada capa.
- 5) **Evaluación de modelos:** Se define la función que despliega el diagrama de flujo de datos representativo de la arquitectura de red neuronal, se grafica la evolución de la función de perdida tanto del grupo de datos de entrenamiento como del grupo de datos de validación a lo largo de las épocas del proceso de aprendizaje, según el tipo de salida que posee el modelo se extraen sus indicadores de rendimiento correspondientes al aplicar el modelo entrenado sobre el selecto grupo de datos de prueba; para el caso del modelo de clasificación se extrae la exactitud, exhaustividad, precisión, el gráfico de la curva característica de funcionamiento del receptor (ROC) junto al área bajo la curva (AUC), se extrae la matriz de confusión y se almacena el tiempo de entrenamiento del modelo, por otra parte, para el caso del modelo de regresión se extrae el error cuadrático medio (RMSE), error absoluto medio (MAE), coeficiente de determinación (R<sup>2</sup>), se extrae el gráfico de dispersión de predicción del RUL, se extrae el gráfico ordenado de predicción del RUL y se guarda el tiempo de entrenamiento del modelo. Para la evaluación de rendimiento de cada modelo se busca realizar este procedimiento de entrenamiento y testeo de las redes neuronales una cantidad mínima de 10 veces con el fin de obtener resultados consistentes y sólidos con respecto al modelo estocástico de las redes neuronales.
- 6) **Análisis de sensibilidad:** Se seleccionan y definen nuevos hiperparámetros a partir de los cuales se diseñan nuevas arquitecturas base de los cuatro modelos iniciales para estudiar el impacto sobre el rendimiento de la red neuronal frente a un cambio de sus valores originales. En base a esta selección de hiperparámetros se construyen múltiples arquitecturas alternativas o modelos variantes que serán sometidos al proceso de entrenamiento, validación y prueba con fin de recopilar suficientes datos sobre su nuevo rendimiento alcanzado.
- 7) **Análisis de resultados:** Se efectúa una recopilación y síntesis de los principales resultados obtenidos por cada tipo de modelo en estudio (transformador, RNN, GRU y LSTM), en función de estos datos se realiza una comparación entre el funcionamiento característico de cada modelo con respecto al rendimiento logrado en la operación de clasificación y regresión, identificando sus ventajas y desventajas para cada caso, en base a esto se establecen los límites de cada modelo y se extraen las principales conclusiones del trabajo de título.

- 8) **Elaboración de informe:** Consiste en la elaboración de un documento en el cual se presenta el tema de trabajo de título seguido de sus antecedentes donde se presentan las definiciones básicas entorno a los modelos neuronales elaborados, se exponen los objetivos a lograr a partir de este proceso de entrenamiento y prueba de arquitecturas, se redacta la metodología donde se especifica en detalle cada paso efectuado, se describen los resultados más importantes destacando las principales conclusiones que se desprenden del presente trabajo de título, citando todas las fuentes bibliográficas que sirvieron de apoyo y guía para la estructura y desarrollo de los modelos neuronales.

Para el desarrollo del trabajo de título y el cumplimiento de los objetivos planteados, se utilizó la siguiente estrategia de trabajo para el desarrollo de cada una de las actividades expuestas en la metodología:



*Figura 19:* Diagrama de estrategia de trabajo del trabajo de título [Fuente: Elaboración propia].

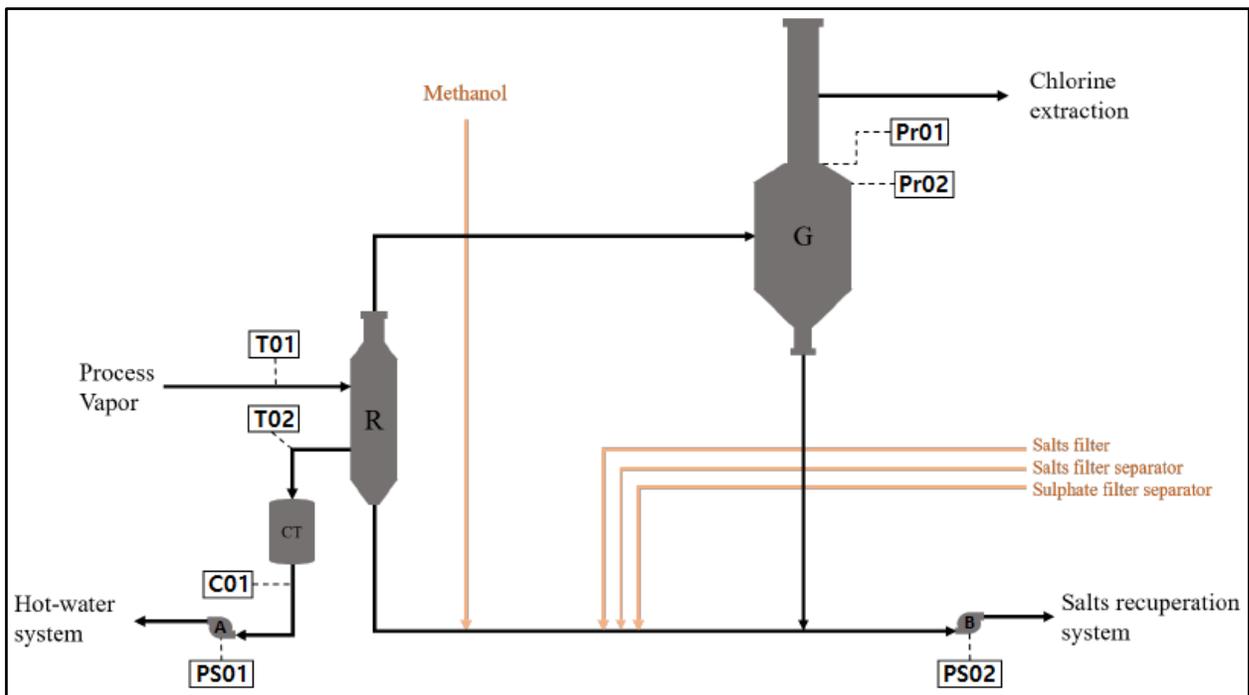
Como podemos observar en la Figura 19, la estrategia de trabajo considera como actividad inicial realizar el análisis de información de la base de datos, para luego dar paso a la ejecución en paralelo de las actividades de preparación de datos, construcción de arquitectura, selección de hiperparámetros y evaluación de los modelos, constituyendo un ciclo reiterativo entre estos procesos con el objetivo de realizar modificaciones en cada una de estas áreas de trabajo para que impacten positivamente en el rendimiento logrado por cada uno de los modelos neuronales creados (Transformador, GRU, RNN y LSTM). Posteriormente una vez diseñados los modelos definitivos bajo estudio se prosigue con el análisis de sensibilidad de las configuraciones originales establecidas, para lo cual se debe realizar el entrenamiento y testeo de nuevos modelos neuronales alternativos, por último, se ejecuta la tarea de análisis de los resultados obtenidos a lo largo de todas estas experiencias a modo de recopilación y síntesis de la información más importante que se desprende del presente trabajo. Cabe destacar que si bien en el diagrama de estrategia de trabajo no se expone explícitamente la actividad de elaboración de informe se debe a que esta actividad se ha desarrollado a lo largo de todo el semestre durante la ejecución de todas estas actividades expuestas en el diagrama.

## 5 Estudio de caso

El siguiente capítulo describe el caso de estudio sobre el cual se aplicarán y evaluarán los modelos neuronales diseñados de tipo Transformador, LSTM, GRU y RNN bajo las dos arquitecturas base propuestas que luego dará a lugar a un análisis de sensibilidad sobre sus parámetros.

El enfoque de los modelos neuronales por mejorar su rendimiento en tareas de clasificación y regresión, nacen de la necesidad de la industria por dar nuevas soluciones a problemas críticos que afectan directamente a la cadena de producción que den como resultado operaciones con mayor eficiencia o una reducción de costos considerable en su ejecución, en particular para el presente trabajo de título, los modelos neuronales propuestos están orientados a la identificación del estado operativo y estimación de la vida útil remanente de un intercambiador de calor que forma parte un sistema rehervidor de generación de dióxido de cloro de una planta de celulosa, que por medio de estas redes neuronales artificiales se busca potenciar el mantenimiento de tipo predictivo para la conservación del activo.

A continuación, en la Figura 20 se expone el diagrama de flujo del sistema hervidor de generación de dióxido de cloro, en donde se exponen los principales subsistemas que lo componen junto a la distribución de los sensores de medición a partir de los cuales se construyó el registro temporal base del presente trabajo de redes neuronales.

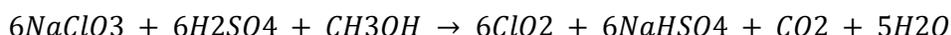


**Figura 20:** Diagrama del sistema rehervidor de generación de dióxido de cloro

[Fuente: Paper "System level prognostics and health management: a graph neural network-based framework" [2] ]

A partir del diagrama previamente expuesto, se puede observar que el sistema rehervidor de generación de dióxido de cloro se compone de los siguientes subsistemas:

**I) Generador (G):** Consiste en un equipo desinfectante que mediante una reacción química que emplea como reactantes clorato de sodio ( $\text{NaClO}_3$ ), ácido sulfúrico ( $\text{H}_2\text{SO}_4$ ) y metanol ( $\text{CH}_3\text{OH}$ ) que por medio de un aumento de temperatura de calor proveniente de vapor de agua genera agua condensada ( $\text{H}_2\text{O}$ ), dióxido de carbono ( $\text{CO}_2$ ), bisulfato sódico ( $\text{NaHSO}_4$ ) y dióxido de cloro ( $\text{ClO}_2$ ), siendo este último el producto objetivo de la reacción que se utiliza para blanquear la pasta de papel, o en otras palabras, para retirar una sustancia resinosa llamada ligsina que se adhiere a las capas de la celulosa y vuelve al papel débil, menos brillante y envejecza más rápido.



**II) Intercambiador de calor de carcasa y tubos (R):** Corresponde a equipo térmico que permite transferir energía térmica de un medio a otro en espacio reducidos, transfiriendo calor desde el fluido que circula por el interior de los tubos hacia aquel que circula por el interior del casco o coraza, dentro del sistema del rehervidor posee el rol de evaporar el agua condensada obtenida del generador para luego reingresarla otra vez al generador junto a los reactantes, con el objetivo de que el vapor de agua libere calor generando un aumento de temperatura entre los reactantes que garantiza la reacción química deseada.

**III) Tanque de condensación (CT):** Es un depósito que almacena el agua líquida de baja temperatura que proviene de los tubos del intercambiador de calor que desempeñó el rol de ceder parte de su energía térmica para evaporar el agua que deriva del generador que circula por el interior de la carcasa del equipo, cabe destacar que esta agua condensada que acumulada en el tanque de condensación posteriormente es redirigida al sistema de agua caliente de la planta de celulosa.

**IV) Bomba del sistema de agua caliente(A):** Corresponde a una máquina que cumple la función de extraer el agua del tanque de condensación, impulsando al líquido hacia el sistema de agua caliente de la planta de celulosa por medio de un aumento de la presión hidráulica de este fluido.

**V) Bomba del sistema de recuperación de sales(B):** Consiste en una bomba de inyección que controla el flujo de sales derivados de la generación de dióxido de cloro hacia un sistema de recuperación de sales de la planta de celulosa.

El problema más común del sistema rehervidor que afecta directamente a la generación de dióxido de cloro corresponde a las incrustaciones por corrosión dentro del intercambiador de calor (como se expone en la Figura 21), causado por las altas temperaturas de funcionamiento que aceleran los procesos de precipitación y la concentración de sales sulfatadas al interior de los tubos del equipo térmico, dando lugar a depósitos que forman una capa aislante que dificulta tanto el intercambio de calor como la circulación del fluido de trabajo a través de ellos, disminuyendo la eficiencia del equipo e incluso afectando al estado operativo del sistema de nominal a defectuoso.



**Figura 21:** Incrustaciones de sulfato en el interior de los tubos del rehervidor

[Fuente: Paper “System level prognostics and health management: a graph neural network based framework” [2] ]

Sobre la base del problema de incrustación por corrosión que sufre el sistema de generación de dióxido de cloro, con el objetivo de detectar a tiempo grandes depósitos de sulfatos que afecten negativamente al estado operativo se monitorean los siguientes parámetros de funcionamiento a través de diferentes sensores de medición distribuidos en los distintos subsistemas que participan del proceso:

**Tabla 3:** Sensores de monitoreo y variables de operación registradas del sistema.

Clase de sensor	Símbolo	Descripción de tipo de medición	Unidad
Conductividad	C01	Conductividad térmica del agua condensada del proceso	uS /cm
Potencia bomba 1	PS01	Carga de potencia de la bomba de reinyección 1	%
Potencia bomba 2	PS02	Carga de potencia de la bomba de reinyección 2	%
Potencia bomba 3	PS03	Carga de potencia de la bomba de reinyección 3	%
Presión 1	Pr01	Presión en el cabezal superior del generador de vapor	kPa
Presión 2	Pr02	Presión diferencial en el generador por pérdidas de vacío	kPa
Temperatura 1	T01	Temperatura del agua de entrada del intercambiador de calor	°C
Temperatura 2	T02	Temperatura del agua de salida del intercambiador de calor	°C
Estado del equipo	.	Condición de operación del sistema	.

## 6 Recursos

En la siguiente sección se exhiben los principales recursos utilizados para la elaboración del presente trabajo de título, en donde se detalla en profundidad la información del banco de datos “Reboiler\_Raw Data\_With Timestamps” provisto por el profesor Guía y empleado como fuente de alimentación para el entrenamiento y testeo de los modelos neuronales diseñados, también se especifican las características técnicas del computador utilizado durante este proceso junto a los programas aplicados para ejecutar el código.

### 6.1 Recursos no pecuniarios

#### A) Equipos de trabajo:

El ordenador portátil utilizado para el desarrollo de los modelos neuronales de aprendizaje automático corresponde a un notebook modelo Nitro 5 marca Acer de 12 GB de Ram con procesador AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx, tarjeta gráfica NVIDIA GeForce GTX 1650 de 4GB, de velocidad base de 2,30 GHz, de 4 núcleos y 8 procesadores lógicos.

Cabe destacar que, para el caso extremo de requerir ejecutar entrenamientos en paralelo de diferentes modelos neuronales a causa de falta de tiempo, se consideró utilizar los computadores de la institución que el departamento de mecánica de la facultad de ciencias físicas y matemáticas de la universidad de Chile dispone para todos los alumnos de pertenecientes a la especialidad, los cuales se encuentran ubicados en el cuarto piso del edificio de Beauchef 851.



*Figura 22: Modelo de notebook de trabajo [Fuente; Pagina Web “www.solotodo.cl”]*

## **B) Softwares empleados:**

Para el desarrollo y ejecución de los modelos neuronales bajo estudio que abarca el presente trabajo de título se emplean los siguientes softwares informáticos, plataformas de almacenamiento online y formatos de datos:

- **Excel:** Es un programa informático de tipo hoja de cálculo desarrollado y distribuido por Microsoft Corp que caracteriza el formato nativo de los datos del documento “Reboiler\_Raw Data\_With Timestamps”, el cual aloja la información que sirve como fuente de alimentación para los modelos neuronales entrenados.
- **Python:** Corresponde a un lenguaje de programación multiplataforma y multiparadigma (puede tratar con programación imperativa, programación funcional y orientación de objetos), se caracteriza por ser flexible y diseñado para ser fácil de leer permitiendo desarrollar aplicaciones en cualquier sistema operativo siendo accesible para casi todo el mundo, presenta diversos paquetes y módulos en su biblioteca que fomenta su adaptabilidad y la reutilización del código. En virtud del diseño y ejecución de los modelos neuronales (LSTM, GRU, RNN y transformador) se selecciona Python como el lenguaje de programación base para el código del programa.
- **Python Colab:** Es un entorno gratuito de Jupyter Notebook que se ejecuta completamente en la nube de Google, te permite programar y ejecutar Python en tu navegador web local permitiendo hasta un máximo de 5 pestañas en ejecución simultáneamente con una única cuenta durante un periodo de 12 horas, por otro lado, presenta la ventaja de brindar la opción de compartir los códigos realizados promoviendo el trabajo en equipo a la hora de programar. Para la ejecución del código este se debe escribir en un cuaderno, el cual es un documento que contiene un código ejecutable y también elementos de texto enriquecido, que para ejecutarse se conecta a una VM de Google Compute Engine (la infraestructura de máquinas virtuales de Google en la nube) que cuenta con 12 GB de RAM y 50 GB de almacenamiento en disco disponibles para su uso. De acuerdo con el presente trabajo de título se seleccionó este entorno de Python Colab para ejecutar el programa diseñado dado la gran capacidad de recursos que presenta y para posteriormente disponer de un espacio virtual en donde programar el código y ejecutar el programa.
- **GitHub:** Es una plataforma web gratuita de alojamiento de códigos en repositorios públicos e ilimitados destinado para proyectos de código abierto que busca mantener un control de las versiones de cada programa y promover la colaboración entre los desarrolladores, en función de la elaboración de los modelos neuronales se crea un repositorio personal en GitHub que sirve como ruta de acceso para permitir la carga de los datos del documento de excell “Reboiler\_Raw Data\_With Timestamps” en Python Colab.
- **Google Drive:** Es una plataforma de almacenamiento online que provee Google en su versión gratuita que incluye una capacidad de almacenamiento máxima de 15 GB, permite almacenar, crear, modificar, compartir y acceder a documentos, archivos y carpetas de todo tipo, para fines del trabajo de título sirve como repositorio para las copias de seguridad de las distintas versiones del código de los modelos neuronales.

### C) Extensión de bibliotecas de Python:

Dentro de la siguiente subsección se presentan los conjuntos de módulos y paquetes de herramientas que forman parte de las extensiones que dispone Python que se utilizaron para la codificación de las redes neuronales involucradas en el presente estudio:

- **Numpy**: Es una librería de Python especializada en el cálculo y análisis numérico de masivas cantidades de datos que se caracteriza por incorporar la clase de objeto “array” que facilita la representación de colecciones de datos en diferentes escalas bajo un formato de matriz multidimensional. Dentro del código programado de las redes neuronales (LSTM, GRU, RNN y Transformador), Numpy cumple el rol de participar en la generación de las ventanas temporales junto a sus respectivas etiquetas, conversión de los datos desde formato “DataFrame” a “Array”, la reestructuración de las dimensiones base de las ventanas temporales, cálculo de muestras totales X e Y y despliegue de las características de las matrices y vectores creados como dimensiones, valores numéricos, cantidad de datos, tipo de datos, distribución de la cantidad de muestras y etiquetas de estado dentro de los grupos de entrenamiento, validación y testeo.
- **Pandas**: Es una biblioteca de software de código abierto dependiente de la librería Numpy diseñado para el lenguaje de programación Python ofreciendo herramientas y funciones que se especializan en la organización, manipulación y análisis de datos que dispongan de estructuras de datos de tipo “Dataframe”, “Panel” y “Series”. Esta biblioteca dentro del código diseñado para el presente trabajo de título cumple la función de carga de datos desde Github a Python Colab, despliegue de las columnas y filas de tablas de datos del documento “Reboiler\_Raw Data\_With Timestamps”, análisis de las características de la información contenida (Máximo, Mínimo, promedio, desviación estándar, número de datos) y generación del conjunto de ventanas temporales junto a sus etiquetas respectivas para el proceso de entrenamiento y testeo de las redes neuronales.
- **Scikit-learn**: La biblioteca Scikit-learn, también llamada sklearn, corresponde a un conjunto de potentes herramientas diseñadas para la ciencia de datos y el aprendizaje automático en Python especializándose en el análisis de datos y modelado estadístico que cumplen un rol fundamental en el área de mantenimiento predictivo de la ingeniería, contando con algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad, además posee la ventaja de ser compatible con las librerías de Numpy, Pandas y Matplotlib. Con el propósito de desarrollar los modelos neuronales relacionados al trabajo de título, scikit-learn desempeña el rol de normalizar los datos numéricos contenidos en el documento “Reboiler\_Raw Data\_With Timestamps” por medio de la función MinMaxScaler, creación de los grupos de muestras (entrenamiento, validación y prueba) y la importación de las métricas de machine learning para los modelos de clasificación y regresión.
- **Tensorflow y Keras**: Ambas corresponden a bibliotecas de código abierto destinados para la creación de sistemas de aprendizaje automático de Deep Learning y computación numérica elaborado por Google Brain permitiendo crear gráficos de flujo de datos, estructuras que describen cómo los datos se mueven a través de un gráfico, series de nodos

de procesamiento y creación de la arquitectura de las redes neuronales. De acuerdo con el código de las redes neuronales diseñado en el presente trabajo de título estas bibliotecas se utilizan para convertir las etiquetas nativas de formato binario a formato one hot encoding mediante la función `to_categorical`, además se encargan de organizar y configurar cada una de las capas y conexiones que conforman los modelos neuronales, por otro lado, también cumplen el rol de desplegar el gráfico de la función de pérdida y configurar tanto el optimizador como los puntos de guardado de los modelos neuronales, para la ejecución de estos procesos los principales comandos que destacan de la biblioteca están:

- `Layers.Input`
  - `Layers.Normalization`
  - `Layers.Resizing`
  - `Layers.RandomFlip`
  - `Layers.RandomZoom`
  - `Tensorflow.image.shape`
  - `Tensorflow.image.extract_patches`
  - `Layers.Dense`
  - `Layers.Dropout`
  - `Layers.Reshape`
  - `Layers.Embedding`
  - `Layers.MultiHeadAttention`, `Layers.RNN`, `Layers.GRU`, `Layers.LSTM`
  - `Layers.Flatten`
  - `Layers.Add()`
  - `Keras.callbacks.ModelCheckpoint`
  - `Keras.callbacks.ModelCheckpoint`
- **Matplotlib**: Es una librería multiplataforma de trazado de Python que se especializa en la generación de gráficos 2D a partir de datos contenidos en listas o arrays, se ha diseñado para facilitar la visualización de datos y el comportamiento de funciones. En favor del desarrollo de los modelos neuronales del trabajo de título, esta librería dentro del código del programa se encarga desplegar las ventanas temporales procesadas por las primeras capas de las redes neuronales y exponer los gráficos comparativos del tiempo de vida remanente predicho por los modelos neuronales frente al valor verdadero del mismo.

## 6.2 Recursos pecuniarios

### A) Fuente base de información:

Sobre la base del estudio del presente trabajo de título, el profesor guía Enrique López proporcionó el documento Excel “Reboiler\_Raw Data\_With Timestamps” para la alimentación de los modelos neuronales, el cual corresponde a un registro temporal de las variables de operación de un sistema rehervidor de generación de dióxido de cloro de una planta de celulosa [3], este registro abarca el monitoreo de 8 variables de operación de los distintos subsistemas que componen al rehervidor, junto su estado representativo de funcionamiento de la instancia de medición, el historial comprende un registro de datos desde el 24 octubre del 2006 hasta el 6 de septiembre del 2018 bajo una frecuencia de medición cada 2 horas, sumando un total de 42206 muestras temporales junto a sus respectivas etiquetas de estado.

A continuación, se presenta una síntesis de los valores numéricos contenidos en el banco de datos “Reboiler\_Raw Data\_With Timestamps”, exponiendo en la siguiente Tabla 5 las características globales que se desprenden de cada variable de operación medida, mientras en la Tabla 6 se muestran las propiedades generales de los estados de funcionamiento establecidos para cada instancia de tiempo de medición y el valor de la vida útil remanente estimada desde la última falla del sistema.

### **Datos brutos de entrada X:**

Los parámetros de operación registrados en el banco de datos en formato Excell con los cuales se construyen las ventanas temporales que sirven como entrada para el entrenamiento, validación y testeo de los modelos neuronales, presentan las siguientes características:

*Tabla 4: Características globales de las variables de operación registradas*

Propiedad	Conductividad	Carga Motor 1	Carga Motor 2	Carga Motor 3	Presión 1	Presión 2	Temperatura 1	Temperatura 2
<b>Conteo</b>	42206	42206	42206	42206	42206	42206	42206	42206
<b>Promedio</b>	4.36 e+29	46.02	56.43	40.73	22.48	22.25	78.13	100.84
<b>Desviación</b>	3.68 e+31	34.41	21.84	9.88	15.59	15.08	14.43	16.34
<b>Mínimo</b>	0	0	0	0	6.75	6.30	2.42	4.21
<b>25%</b>	2.70	0	40.06	38.05	18.07	18.00	77.97	100.25
<b>50%</b>	3.60	69.07	67.59	42.46	19.75	19.60	79.39	101.05
<b>75%</b>	4.70	72.93	73.16	45.56	20.58	20.50	81.49	105.41
<b>Máximo</b>	4.00e+33	105.57	86.44	65.91	123.52	108.40	137.73	160.44

### **Datos brutos de salida Y:**

Los datos de salida objetivo para el caso de los modelos clasificación corresponde a predecir la condición operativa del sistema (defectuoso o nominal), mientras para el caso de los modelos de regresión estos apuntan a estimar el tiempo de vida útil remanente del equipo (RUL) desde la última falla del sistema que removió su condición nominal de funcionamiento, ambos tipos de datos presentan las siguientes características:

*Tabla 5: Propiedades generales de etiquetas de estados y vida útil remanente*

Propiedad	Clases binaria {0,1}	Tiempo vida remanente [Horas]
<b>Conteo</b>	42206	42206
<b>Promedio</b>	0.06	3924.46
<b>Desviación</b>	0.24	5361.75
<b>Mínimo</b>	0	0
<b>25%</b>	0	256
<b>50%</b>	0	1648
<b>75%</b>	0	4982
<b>Máximo</b>	1	21634

## 7 Resultados y Discusión

En el siguiente capítulo se presentan los resultados que se desprenden de la ejecución del proceso de entrenamiento de aprendizaje automático de los modelos neuronales bajo estudio, abarcando también la evaluación del desempeño logrado por cada uno de estos sobre tareas de identificación del estado operativo del sistema por método de clasificación o estimación de la vida útil remanente del equipo por método de regresión, para luego continuar con los resultados que se desprenden del desarrollo del análisis de sensibilidad de los parámetros netos que conforman las arquitecturas originales diseñadas de tipo Transformador, LSTM, GRU y RNN, tanto para la arquitectura “Attention is all you need” como también para la arquitectura “An Image is Worth 16x16 Words”.

A continuación, los primeros resultados a exponer corresponden a aquellos que se desprenden directamente del proceso de aprendizaje automático; para el caso de clasificación se exponen las funciones de pérdidas durante el entrenamiento, las matrices de confusión usuales obtenidas y las curvas características de funcionamiento del receptor, por otra parte, para el caso de regresión se exponen las funciones de pérdidas durante el entrenamiento, el diagrama de dispersión valor predicho-valor real y el gráfico de dispersión ordenado valor predicho - valor real.

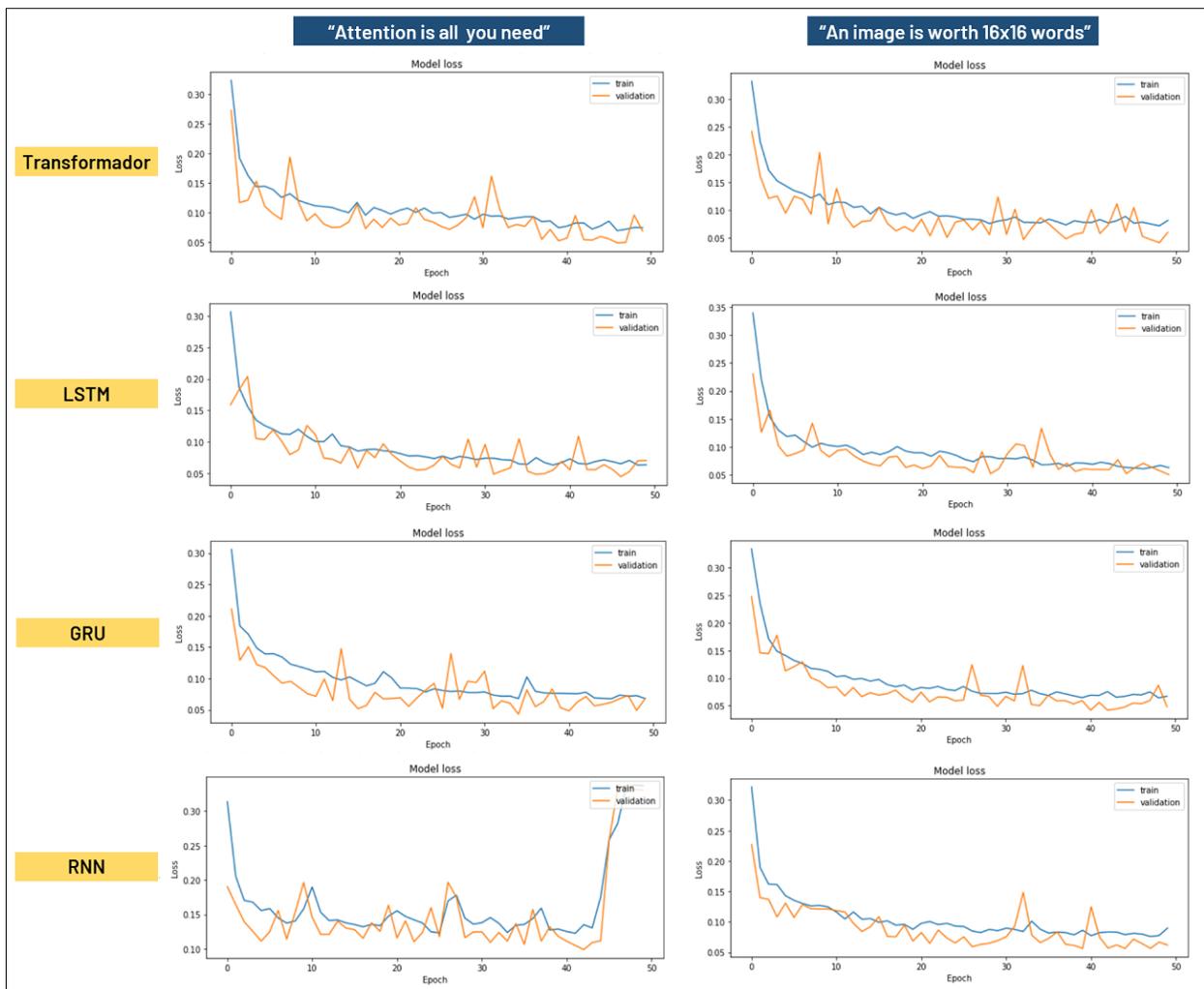
Posteriormente los segundos resultados corresponden al desempeño logrado por cada modelo, dentro del caso de clasificación se exponen los indicadores de rendimiento globales de puntajes de exactitud global, precisión, exhaustividad, valor F1, área bajo la curva característica operativa del receptor, parámetros entrenables netos y tiempo total de entrenamiento, mientras para el caso de regresión se exponen los indicadores de rendimiento globales de puntajes de error cuadrático medio (RMSE), el error absoluto medio (MAE) y el coeficiente de determinación (R<sup>2</sup>), parámetros entrenables netos y tiempo total de entrenamiento. Sobre la base de estos datos luego se muestran gráficos comparativos que representan el desempeño logrado por cada modelo con relación al número de parámetros netos que los conforman.

Finalmente, esta sección culmina describiendo los resultados que se desprenden del análisis de sensibilidad de los parámetros de entrenamiento que conforman los modelos neuronales de tipo Transformador, LSTM, GRU y RNN bajo estudio, revelando el impacto que tiene el cambio del número de parámetros que conforma cada modelo sobre el rendimiento alcanzado en tareas de clasificación del estado operativo y regresión de la vida útil remanente del equipo, exponiendo los indicadores de rendimiento y los gráficos comparativos de desempeño obtenidos por cada modelo variante puesto a prueba.

## 7.1 Resultados de modelos de clasificación

### 7.1.1 Función de pérdida de proceso de aprendizaje automático

A continuación, se presenta la evolución del indicador de rendimiento de “Exactitud” sobre el grupo de datos de entrenamiento y validación, obtenido a lo largo de 50 épocas del proceso de aprendizaje automático de los modelos neuronales de tipo transformador, LSTM, GRU y RNN bajo las dos arquitecturas propuestas para la clasificación del estado operativo del equipo térmico.



*Figura 23: Comparación de funciones de pérdida en clasificación de cada tipo de modelo neuronal*

Sobre la base de la imagen comparativa de la función de pérdida de los modelos neuronales puestos a prueba en la Figura 23, se observa que para ningún modelo neuronal entrenado existe sobreajuste de los datos durante el proceso de aprendizaje, debido a que todos los gráficos presentan una curva de la función de pérdida del grupo de datos de entrenamiento y validación que a lo largo de las épocas convergen y finalmente decaen simultáneamente hasta el valor mínimo de Exactitud, aunque cabe resaltar que el modelo neuronal RNN bajo la arquitectura “Attention is all you need” presenta un comportamiento que no decae en las épocas finales del entrenamiento, esto se debe a que las capas normalizadoras ubicadas en segundo y cuarto orden del codificador en la arquitectura “Attention is all you need” reducen las marcadas diferencias entre los valores obtenidos al final de este bloque, de los cuales el optimizador aprovecha para ajustar los parámetros por medio del cálculo del gradiente de la función de pérdida, más aún se enfatiza este problema por el desvanecimiento de gradiente que presentan fuertemente este tipo de modelos neuronales ante largas series temporales, perdiendo gradualmente información de las instancias temporales iniciales, dificultando al optimizador modificar eficientemente la gran cantidad de parámetros de pesos entre los nodos con el objetivo de mejorar el rendimiento de la red. Gracias a esto mismo, según los resultados de las funciones de pérdida de los ocho modelos entrenados aquél que presenta los peores resultados sobre su función de pérdida corresponde al modelo RNN bajo la arquitectura “Attention is all you need”.

Por otra parte, a partir de la Figura 23 se revela que tanto para los modelos bajo la arquitectura “Attention is all you need” como también para los modelos bajo la arquitectura “An image is worth 16x16 words”, en ambos casos dentro del mismo tipo de modelo neuronal presentan comportamientos similares durante el proceso de aprendizaje automático, siendo levemente más errático el comportamiento de la arquitectura “An image is worth 16x16 words”, esto se debe a que al poseer las capas normalizadoras en primer y tercer orden dentro del bloque del codificador, el codificador entrega valores más irregulares que resulta en un gradiente mucho más pronunciado, implicando en un ajuste más drástico de los pesos nodales o parámetros de entrenamiento por parte del optimizador.

### 7.1.2 Matriz de confusión

A continuación, se exponen las matrices de confusiones usuales que se desprenden del proceso de prueba de cada modelo tipo de red neuronal bajo las dos arquitecturas propuestas, a partir de los cuales se extrae la información de los indicadores de rendimiento de: exactitud, exhaustividad, precisión, puntaje valor F1, gráfico y área de la curva característica operativa del receptor.

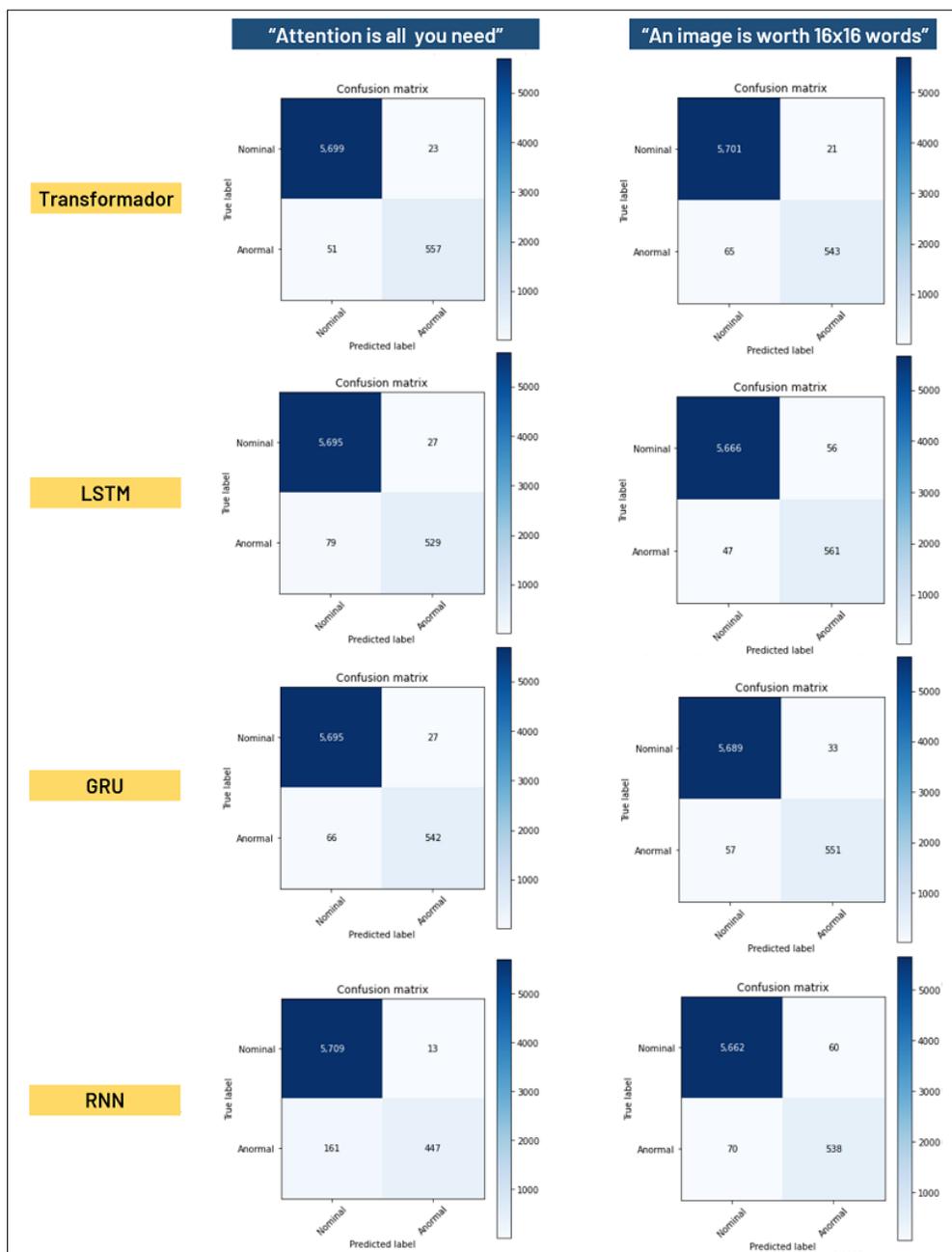


Figura 24: Comparación de matrices de confusión de clasificación de cada tipo de modelo neuronal

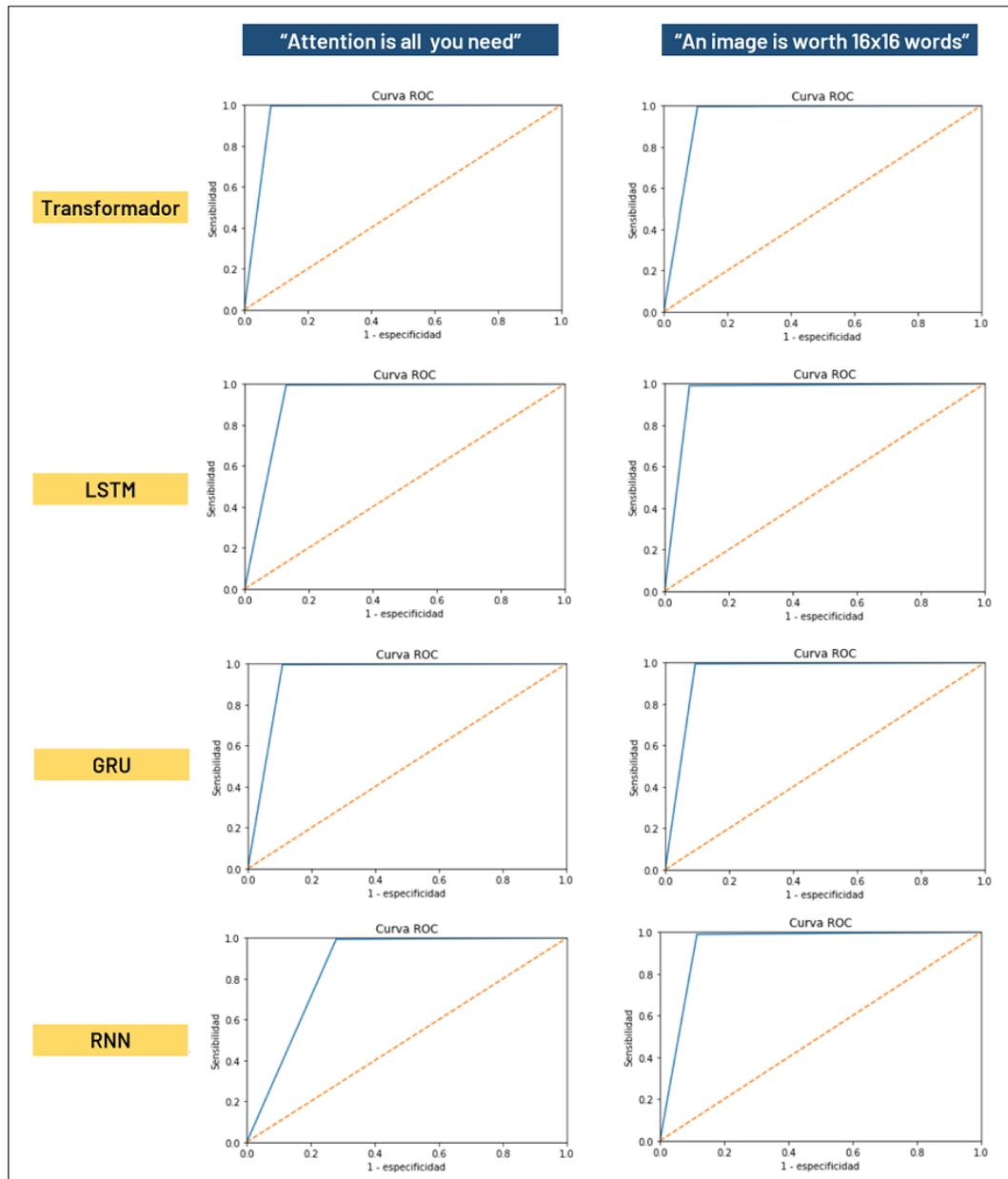
A partir de las matrices de confusión expuestas en la Figura 24, se observa que la mayor cantidad de aciertos en la identificación de estados de operación por parte de los modelos neuronales se concentra en detectar el estado nominal de funcionamiento del equipo térmico, esto se debe a raíz de que el estado nominal corresponde a la etiqueta predominante por alrededor del 90% de las muestras, por lo tanto es claro que los modelos para maximizar rápidamente su desempeño es lo primero que aprenden a detectar durante las primeras iteraciones de épocas del aprendizaje automático, luego en segundo lugar se prioriza maximizar la identificación de los estados de funcionamiento anormales, de los cuales corresponden al 10% restante que componen las muestras, siendo este el principal problema para la identificación de este estado operativo por la menor proporción de muestras representativas con las que deben de aprender cada modelo, esto se puede ver en las matrices de confusión al notar que los modelos neuronales entrenados tienen a confundir más muestras de estados operacionales defectuosos con nominales que estados operacionales nominales con defectuosos.

De acuerdo con las matrices de confusión se desprende que el desempeño de las dos arquitecturas propuestas con las que se basó el diseño de los cuatro tipos de modelos neuronales transformador, LSTM, GRU y RNN, no exponen grandes diferencias entre los resultados obtenidos por las diferentes arquitecturas bajo un mismo tipo de red neuronal, por lo que desde ya nos indica que el cambio del orden las capas de normalización en el codificador no repercuten drásticamente en el rendimiento de los tipos de modelos.

También, cabe destacar que a partir de las matrices de confusión se revela que aquel tipo de modelo neuronal que peor identifica los casos de funcionamiento anormales resulta ser nuevamente el modelo RNN bajo la arquitectura “Attention is all you need”, sin embargo, también corresponde al modelo que más estados nominales identifica dentro de los modelos entrenados.

### 7.1.3 Curva característica de funcionamiento del receptor

A continuación, se exponen las curvas características operativas del receptor habituales que se desprenden del proceso de entrenamiento de cada tipo de modelo neuronal bajo las dos arquitecturas propuestas enfocadas en la clasificación de estados de operación.



**Figura 25:** Comparación de curvas características de clasificación de cada tipo de modelos bajo estudio.

En función de la imagen comparativa de las curvas características de la Figura 25, se observa que todos los modelos neuronales entrenados puestos a pruebas presentan una forma análoga de la curva característica operativa del receptor (ROC), una curva que asciende linealmente para luego cambiar abruptamente su dirección hacia una línea horizontal, este comportamiento se presenta independiente del tipo de modelo neuronal o la arquitectura a la que está sujeta su estructura.

No obstante, los modelos construidos bajo la arquitectura “An image worth 16x16 words” presentan un leve mejora en su curvas característica que reflejan un mayor rendimiento en la identificación de estados operacionales que aquellos modelos neuronales bajo la arquitectura “Attention is all you need”, debido a que esta última arquitectura se especializa más en extraer características temporales de la secuencia de datos que en la identificación de patrones en la serie temporal que son fundamentales en la identificación del estado operativo.

Cabe destacar que nuevamente el peor rendimiento logrado resulta ser por parte del modelo neuronal de tipo RNN que independiente bajo que arquitectura se encuentre sujeta la estructura presenta una curva más decaída, el responsable de este comportamiento se debe en parte al problema de desvanecimiento de gradiente que afecta más negativamente a este tipo de modelo neuronal RNN en comparación al resto de los tipos de modelos neuronales Transformador, GRU, LSTM, y por otra parte también se debe a que el funcionamiento de su capa característica es mucho más simple y limitada que el resto de los modelos, mientras estos otros presentan una curva características con resultados similares independiente de la arquitectura.

### 7.1.4 Indicadores de rendimiento por tipo de modelo

En el siguiente apartado se exponen las tablas de los indicadores de desempeño promedio de las 10 iteraciones de aprendizaje automático ejecutado sobre cada modelo neuronal bajo estudio, presentando los resultados de forma separada según la arquitectura a la que se encuentra sujeta la estructura base con la que se construyeron los modelos de tipo Transformador, LSTM, GRU y RNN, exponiendo los indicadores de rendimiento de exactitud, exhaustividad, precisión, puntaje valor F1 y valor del área bajo la curva ROC, junto al periodo de tiempo de ejecución de entrenamiento y la cantidad neta de parámetros que componen cada estructura. Para más detalle sobre el registro de estos valores dentro de las 10 iteraciones ejecutadas diríjase a la subsección Anexo A: Registro de resultados de modelos de clasificación por iteración de aprendizaje.

#### 7.1.4.1 Arquitectura “Attention is all you need”

*Tabla 6: Indicadores de rendimiento en clasificación de modelos bajo arquitectura “Attention is all you need”.*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.950.277	4.178.117	3.984.581	3.978.949
Tiempo [h]	1,1	1,5	1,3	1,2
Exactitud [%]	98,5	98,5	98,4	96,9
Precisión [%]	95,3	92,2	92,1	93,6
Exhaustividad [%]	88,6	91,9	91,5	73,3
Puntaje valor F1 [%]	91,8	92,0	91,7	82,0
Área curva ROC [%]	94,1	95,5	95,3	86,4

#### 7.1.4.2 Arquitectura “An image is worth 16x16 words”

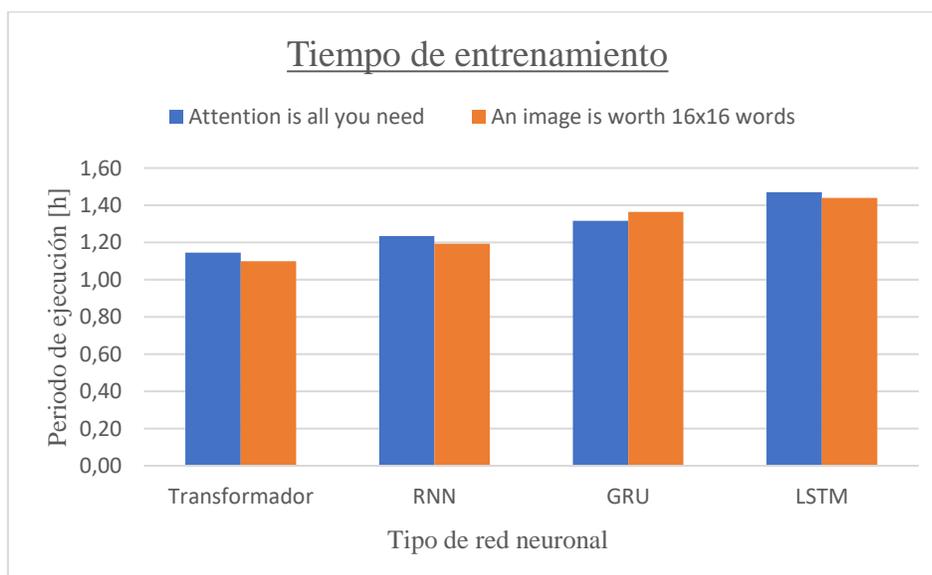
*Tabla 7: Indicadores de rendimiento en clasificación de modelos bajo arquitectura “An image is worth 16x16 words”.*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.950.277	4.178.117	3.984.581	3.978.949
Tiempo [h]	1,10	1,44	1,36	1,19
Exactitud [%]	98,5	98,3	98,6	98,1
Precisión [%]	94,8	90,6	93,4	90,5
Exhaustividad [%]	89,7	92,4	91,7	89,3
Puntaje valor F1 [%]	92,1	91,4	92,5	89,8
Área curva ROC [%]	94,7	95,7	95,5	94,1

### 7.1.5 Gráficos de comparativos de rendimiento:

A partir del contenido de las tablas previamente expuestas se elaboran los siguientes gráficos de rendimiento con el fin de facilitar la evaluación comparativa de los resultados entre los diferentes tipos de modelos neuronales y las distintas arquitecturas propuestas, además para quitar la ventaja de mayor capacidad de procesamiento de características que supone una mayor cantidad de parámetros que conforma la capa característica de cada tipo de modelo, es que se ha decidido presentar el desempeño de cada tipo de modelo neuronal a razón del valor del indicador de rendimiento dividido por la cantidad de parámetros neto del modelo.

#### Tiempo de entrenamiento:



*Figura 26:* Gráfico comparativo del tiempo de entrenamiento según tipo de modelo y arquitectura.

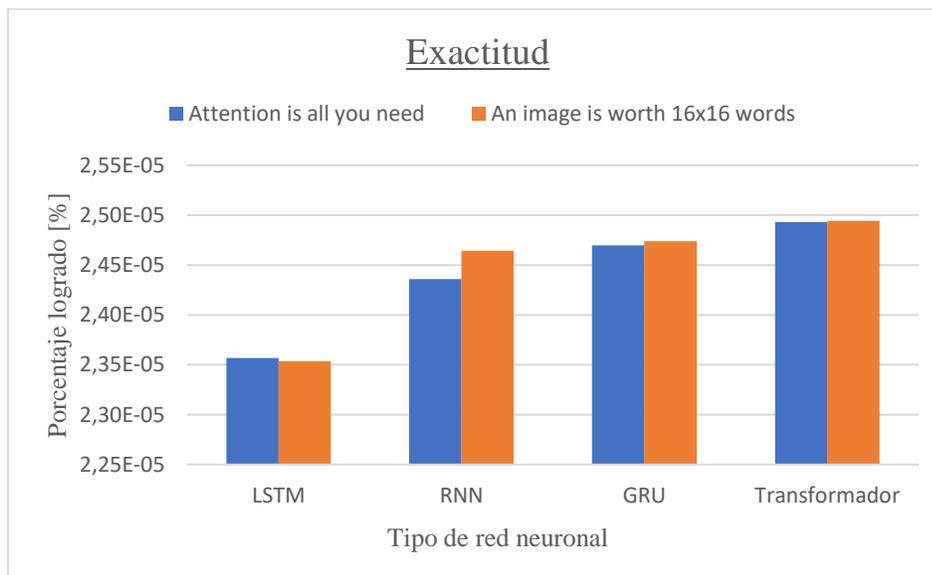
De acuerdo con el gráfico comparativo de los tiempos de ejecución de entrenamiento para modelos de clasificación de la Figura 26, se desprende que el modelo neuronal más rápido en ejecutar su proceso de entrenamiento corresponde al tipo de modelo Transformador, debido a que el funcionamiento de su capa característica se centra en la identificación de patrones relevantes para posteriormente en base a esto aplicar un filtro de datos que repercute en un procesamiento parcial de las secuencias temporales de entrada, agilizando el desarrollo de la respuesta del codificador.

En segundo lugar, se encuentra el tipo de modelo RNN, debido a que el funcionamiento y composición de su celda distintiva es la más simple y básica dentro de los cuatro modelos, exponiendo una única unidad de memoria y una menor capacidad de elaboración de características complejas, que acorta los tiempos de ejecución de la capa característica, sin embargo, este tipo de modelo pierde tiempo al procesar la serie temporal completa de entrada.

En tercer lugar, se encuentra el tipo de modelo GRU, que dentro de su capa característica presenta una celda distintiva con un nivel de complejidad de funcionamiento y composición intermedia entre el modelo RNN y modelo LSTM, siendo muy similar a una celda RNN, pero con mayor número de submódulos y conexiones que no sobrepasa a la del modelo LSTM, por lo tanto, esto le otorga un mayor tiempo de entrenamiento para procesar la serie temporal completa de entrada.

Finalmente el modelo neuronal más lento en ejecutar su proceso de entrenamiento corresponde al modelo LSTM, ya que se debe en primera instancia a que el funcionamiento de su capa característica extrae características procesando la serie temporal completa, y por otro lado se debe a que las celdas LSTM resultan ser más complejas en cuanto a un mayor número de submódulos y conexiones, agregando además dos componentes de memoria adicionales (Largo y corto plazo), lo que le confiere a su capa característica un mayor número de parámetros de entrenamiento por ajustar.

### **Indicador de rendimiento de exactitud:**



*Figura 27:* Gráfico comparativo del indicador de rendimiento de exactitud según tipo de modelo y arquitectura.

De acuerdo con el gráfico comparativo de exactitud para modelos de clasificación de la Figura 27, se desprende que el modelo neuronal más eficiente en distinguir estados nominales y defectuosos verdaderos corresponde al tipo de modelo Transformador, ya que su capa característica se especializa en la identificación de patrones distintivos en las ventanas temporales para identificar el estado operativo defectuoso de las muestras, suprimiendo el resto de la información que no contribuye con la entrega de una buena respuesta de clasificación.

En segundo lugar, se encuentra el tipo de modelo GRU, ya que su celda distintiva por un lado permite desarrollar características más complejas por su mayor número de subcomponentes (puerta de actualización y puerta de reinicio) que le permite extraer información más útil para la predicción, y por otro lado de acuerdo a la particular estructura que posee la celda distintiva de los modelos GRU diseñada para reducir el desvanecimiento de gradiente, le permite no perder grandes cantidades de información de las primeras instancias de medición de las ventanas temporales, pudiendo así elaborar características

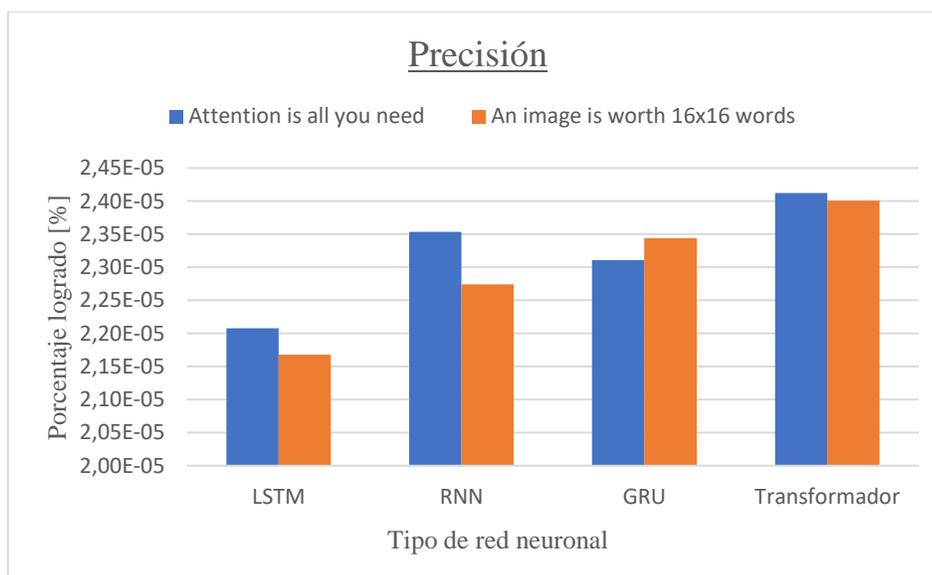
considerando el panorama completo de los datos de las ventanas temporales facilitando la identificación del estado operativo de la muestra.

En tercer lugar, se encuentra el tipo de modelo RNN, ya que por la estructura más simple que presenta su celda distintiva, posee una menor capacidad para desarrollar características complejas y tiende a ser más afectado por el desvanecimiento de gradiente, perdiendo parte de la información de las primeras instancias de medición de las ventanas temporales, aunque esto también le permite enfocarse especialmente en las últimas instancias temporales de la ventana temporal de muestra que resultan útiles para definir una buena respuesta del estado operativo.

Por último, se encuentra el modelo LSTM como aquel que tiene el peor desempeño en exactitud, esto se debe al mayor grado de complejidad de su celda distintiva dentro de la cual destacan las unidades de memoria de largo plazo y corto plazo que dificultan al modelo para enfocarse en extraer características útiles en un rango de tiempo de la ventana temporal para definir una buena respuesta, ralentizando la mejora de su desempeño en cada época de entrenamiento.

Cabe destacar que a partir del gráfico comparativo del indicador de rendimiento de exactitud se revela que independiente de la arquitectura a la que está sujeta el tipo de modelo neuronal estos presentan el mismo orden creciente de desempeño en cuanto al indicador de rendimiento exactitud, aunque la arquitectura “An image worth 16 x 16 words” presenta mejores resultados ya que se especializa en la detección de objetos en imágenes, que favorece a la detección de patrones en las ventanas temporales para determinar el estado operativo.

### **Indicador de rendimiento de precisión:**

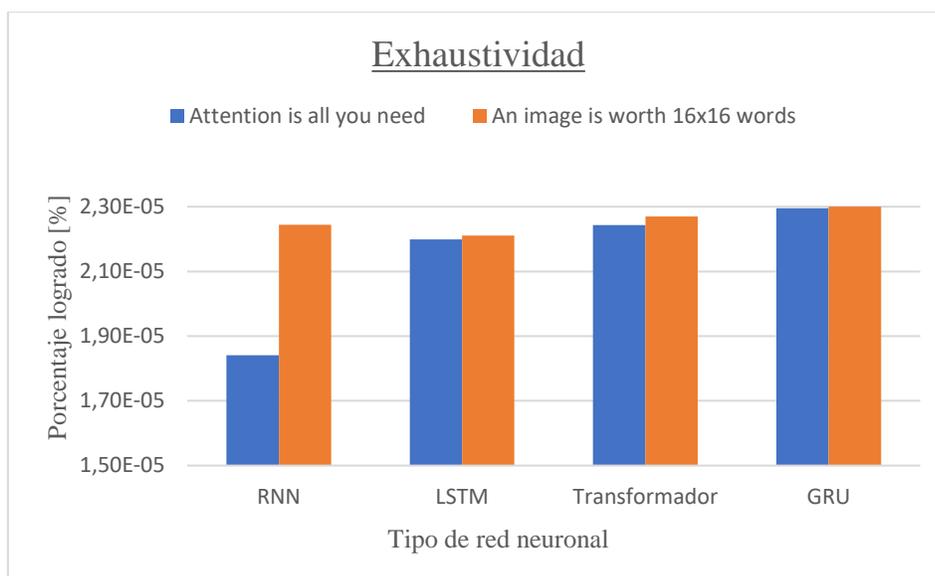


**Figura 28:** Gráfico comparativo del indicador de rendimiento de precisión según tipo de modelo y arquitectura.

De acuerdo con el gráfico comparativo de precisión para modelos de clasificación de la Figura 28, se observa un comportamiento similar al expuesto en gráfico comparativo del indicador de rendimiento de exactitud, ya que por las mismas razones el modelo neuronal que tiene mayor probabilidad de que su

predicción de un estado operativo defectuoso sea cierta, corresponde al modelo transformador, ya que su capa característica le confiere una ventaja por sobre el resto de tipos de modelos para concentrarse en la detección de patrones que sean distintivos de los estados operativos defectuoso, mientras los demás modelos neuronales quedan rezagados porque su capa característica se concentra en elaborar características que resultan menos útiles para la detección de estos estados operacionales deficientes.

### **Indicador de rendimiento de exhaustividad:**



*Figura 29: Gráfico comparativo del indicador de rendimiento de exhaustividad según tipo de modelo y arquitectura.*

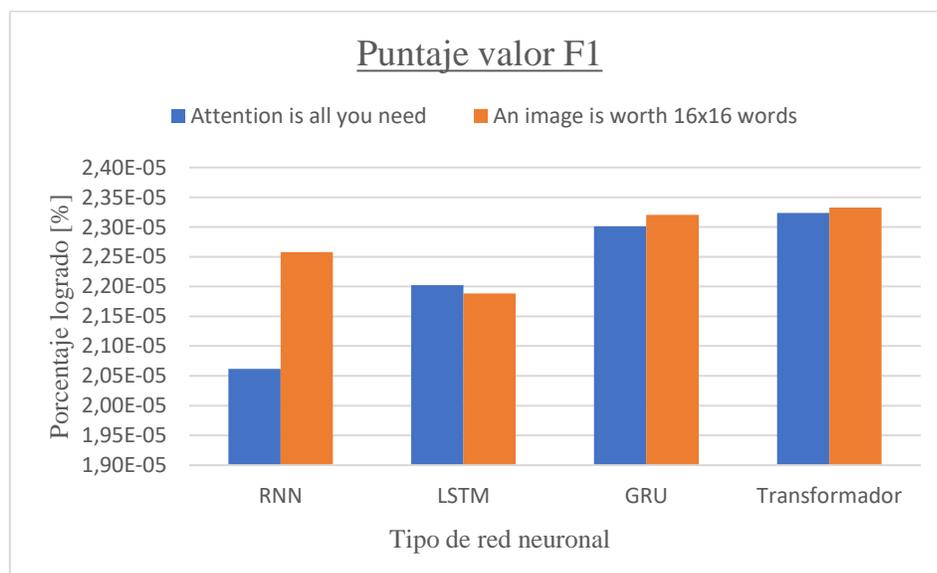
En función del gráfico comparativo de exhaustividad para modelos de clasificación de la Figura 29, se deduce que el modelo neuronal que tiene mayor probabilidad de que su predicción de un estado operativo nominal sea cierta, corresponde al modelo neuronal de tipo GRU, ya que su modelo rival el transformador se enfoca demasiado en identificar los estados defectuosos de las muestras minoritarias, descuidando la identificación del estado operativo predominante en las muestras, mientras el modelo neuronal GRU por no contar con este potente filtro de datos es más consistente enfocándose en el panorama global de datos que componen las ventanas temporales para identificar el estado nominal en las muestras.

Por el otro extremo del gráfico de la Figura 29, los modelos LSTM y RNN exponen un menor rendimiento de exhaustividad, quedando el modelo LSTM rezagado por presentar la desventaja de poseer dos componentes distintos de memorias en su celda distintiva que perjudica la tarea de clasificación de estados operacionales, mientras el modelo RNN es castigado por poseer una menor capacidad de desarrollo de características complejas que faciliten la identificación del estado nominal, además se suman los efectos negativos que se desprenden del desvanecimiento de gradiente.

De acuerdo con el gráfico comparativo de exhaustividad para modelos de clasificación de la Figura 29, se puede observar que el orden creciente de los valores de exhaustividad obtenidos sí depende de la arquitectura a la que están sujetos los modelos, porque si bien para el modelo Transformador y GRU se cumple el orden creciente independiente de las arquitecturas, el problema recae en el orden de los modelos

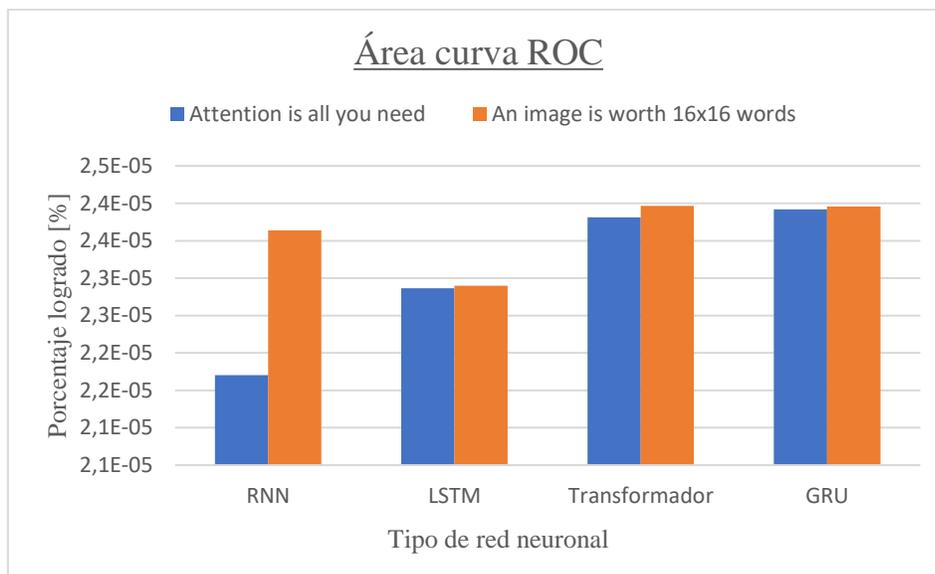
RNN y LSTM, pues el orden creciente de las magnitud de exhaustividad obtenidas se invierten de acuerdo a la arquitectura escogida, aunque como las mayores magnitudes poseen valores muy similares esto se atribuye a la variabilidad estadística sujeta al limitado número de iteraciones de aprendizaje evaluadas. Además, se puede notar que globalmente la arquitectura “An image worth 16x16 words” presenta mejores resultados que la arquitectura “Attention is all you need”, ya que al igual que para el caso del gráfico comparativo del indicador de rendimiento de precisión este tipo de arquitectura saca mayor provecho por especializarse en la detección de objetos en imágenes facilitando la identificación del estado operativo nominal. Particularmente resalta el bajo rendimiento del modelo RNN bajo la arquitectura “Attention is all you need” a causa de la acción conjunta de las desventajas del funcionamiento del tipo de modelo como de arquitectura en la que se basa.

### **Indicador de rendimiento de puntaje valor F1:**



**Figura 30:** Gráfico comparativo del indicador de rendimiento de puntaje valor F1 según tipo de modelo y arquitectura.

El gráfico comparativo del indicador puntaje valor F1 deriva de la combinación de los indicadores de rendimiento de precisión y exhaustividad, lo que resulta más práctico para observar el rendimiento general de la acción de ambos indicadores. En base al gráfico comparativo del puntaje valor F1 de la Figura 30, se desprende que el modelo neuronal que presenta la predicción más confiable corresponde al modelo Transformador, seguido muy de cerca por las predicciones del modelo GRU, mientras los menos confiables corresponderían al modelo RNN y modelo LSTM.

**Indicador de rendimiento de área bajo la curva ROC:**

**Figura 31:** Gráfico comparativo del indicador de rendimiento área bajo curva ROC según tipo de modelo y arquitectura.

De acuerdo con el gráfico comparativo del área bajo la curva características para modelos de clasificación de la Figura 31, se observa que el modelo neuronal con mayor capacidad discriminativa entre los estados de operación nominal y defectuoso sobre las muestras de ventanas temporales de entrada corresponde al modelo neuronal de tipo GRU, gracias a su gran capacidad de elaborar características complejas otorgada por la estructura de su celda distintiva y por otro lado gracias a su gran memoria de largo plazo que aplica sobre cada una de las instancias de medición que componen las ventanas temporales de entrada, evitando así la pérdida de información relevante para definir una buena respuesta.

En segundo lugar, se presenta el modelo Transformador exponiendo valores muy cercanos al modelo GRU, esto se debe a que este tipo de modelo presenta la ventaja de aplicar un potente filtro de información previo al desarrollo de características que le ayuda a detectar patrones dentro de los datos de las ventanas temporales que resultan útiles para identificar el estado operativo de la muestra. Aunque cabe resaltar que se sospecha en base a los resultados de los indicadores de rendimiento previos, que muy probablemente por el limitado número de iteración de aprendizaje evaluadas esta sea una desafortunada coincidencia estadística que posiciona por muy poco este tipo de modelo en el segundo lugar, pues se presume que este modelo debería haber mostrado el mejor desempeño para el indicador de rendimiento área bajo la curva ROC.

En tercer lugar, se encuentra el modelo LSTM que, a pesar de poseer una mayor complejidad en su estructura y funcionamiento en su celda distintiva, el hecho de manejar dos unidades de memoria de largo plazo y corto plazos al final resultan contraproducentes, dificultando la elaboración de características útiles para la identificación de estados operativos y subsecuentemente genera una ralentización en la optimización de los parámetros críticos del modelo para mejorar su rendimiento, resultando en un aprendizaje más lento para alcanzar mayor desempeño.

En último lugar se encuentra el modelo RNN, quedando rezagado por el resto de los tipos de modelos neuronales por poseer una celda distintiva que es más afectada negativamente por el desvanecimiento de gradiente y por su baja capacidad de elaborar características más complejas que faciliten la identificación de estados.

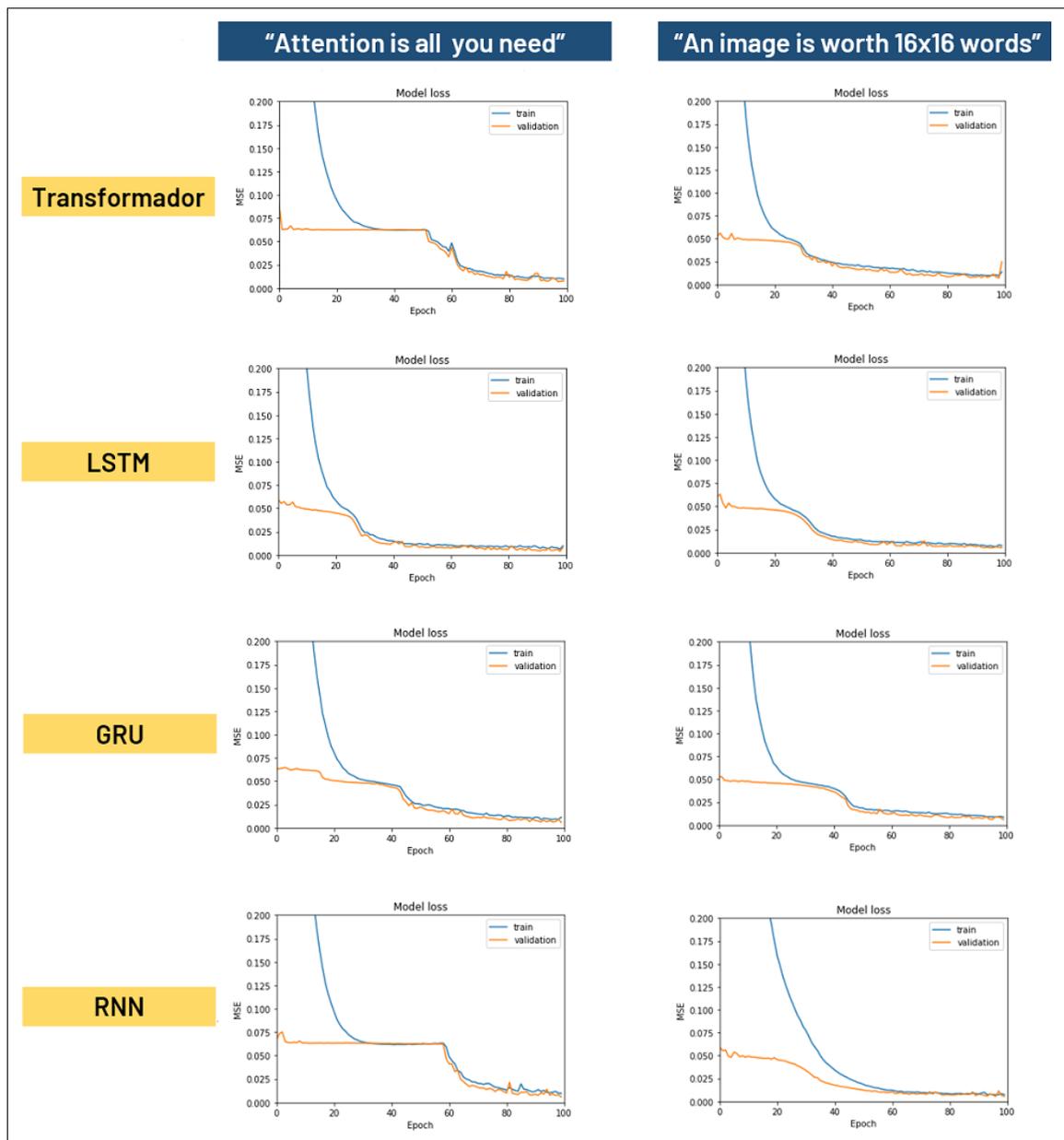
De acuerdo con el gráfico comparativo del área bajo la curva ROC para modelos de clasificación de la Figura 31, se puede observar que el orden creciente de los valores del área bajo la curva ROC obtenidos sí depende de la arquitectura a la que están sujeta los modelos, porque si bien para el modelo Transformador y GRU se cumple el orden creciente de forma independiente de la arquitecturas, el problema recae en el orden de los modelos RNN y LSTM, pues el orden creciente de las magnitud de exhaustividad obtenidas se invierten de acuerdo a la arquitectura escogida.-

El gráfico comparativo describe que la arquitectura “An image worth 16x16 words” presenta mejores resultados que la arquitectura “Attention is all you need” para discernir entre estados operacionales, ya que este tipo de arquitectura al especializarse en la detección de objetos en imágenes se le facilita la acción de identificar estados operativos a partir de ventanas temporales. Cabe destacar el menor rendimiento del modelo RNN bajo la arquitectura “Attention is all you need” es causada por la acción conjunta de las desventajas del básico funcionamiento del tipo de modelo como de la poco conveniente arquitectura en la que se basa.

## 7.2 Resultados de modelos de regresión

### 7.2.1 Función de pérdida de proceso de aprendizaje automático

A continuación, se presenta la evolución del indicador de rendimiento del “Error cuadrático medio” sobre el grupo de datos de entrenamiento y validación, obtenido a lo largo de 100 épocas del proceso de aprendizaje automático de los modelos neuronales de tipo transformador, LSTM, GRU y RNN bajo las dos arquitecturas propuestas para la estimación de la vida útil remanente del equipo térmico bajo estudio.



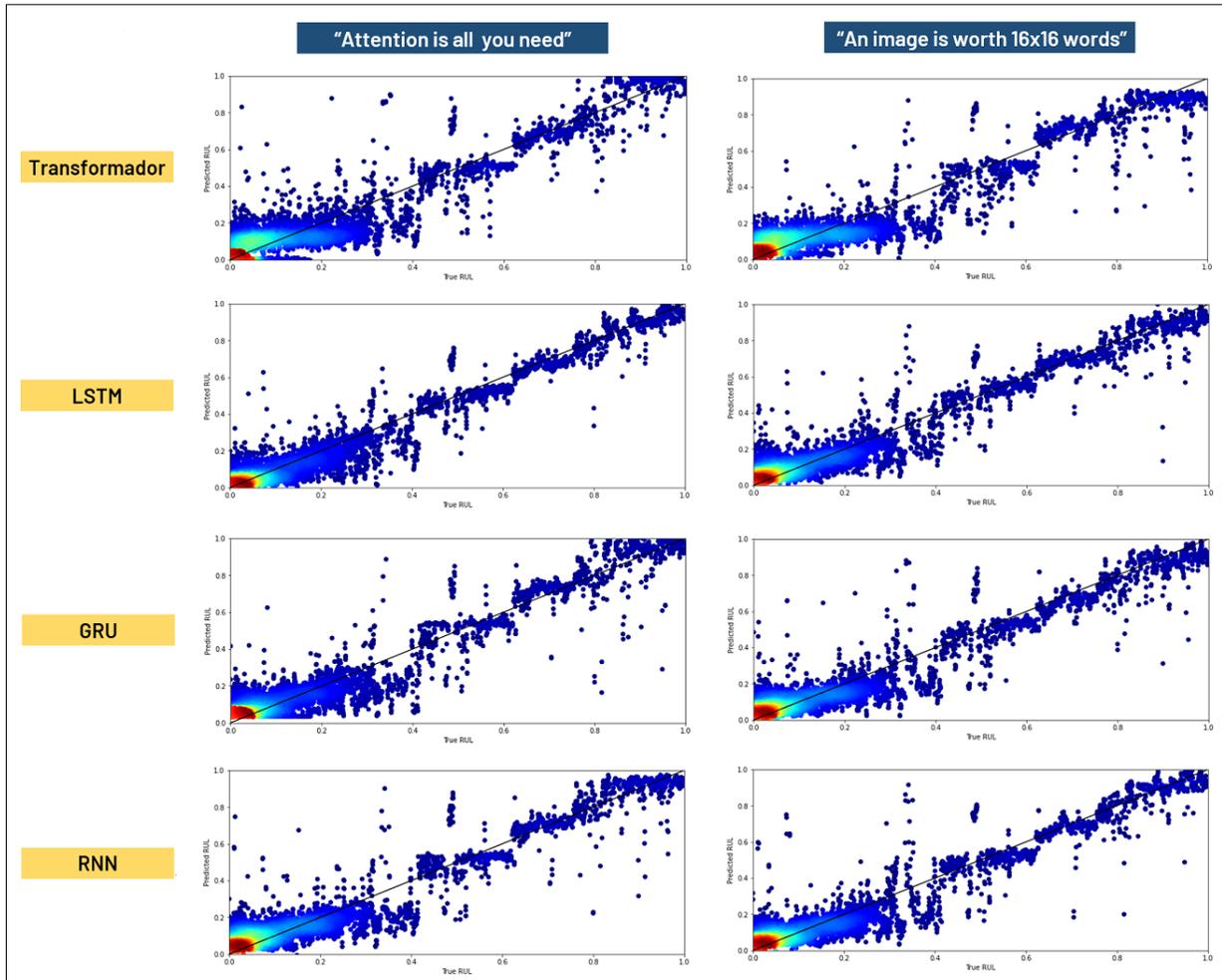
*Figura 32:* Comparación de funciones de pérdida en regresión de cada tipo de modelos neuronal.

Sobre la base de la imagen comparativa de la función de pérdida de los modelos neuronales entrenados de la Figura 32, se desprende que no existe sobreajuste de datos durante el proceso de aprendizaje automático en ningún modelo entrenado, pues todos los gráficos de función de pérdida exhiben un comportamiento del grupo de datos de entrenamiento y validación que en el transcurso de las épocas convergen y decaen simultáneamente hasta el valor menor del Error cuadrático medio.

Por otra parte, también se expone que tanto para los modelos bajo la arquitectura “Attention is all you need” como también para los modelos bajo la arquitectura “An image worth 16x16 words” dentro de la categoría de un mismo tipo de modelo neuronal (Transformador, LSTM, GRU y RNN) presentan curvas con comportamientos similares durante el proceso de aprendizaje automático, llegando a valores muy cercanos al término de las épocas de entrenamiento, e incluso independiente de la arquitectura que se escoja las curvas exhiben un comportamiento muy estable, reflejando que sin importar el orden de las capas normalizadoras en el codificador no impacta en gran medida en la manera en que se entrenan los tipos de modelos bajo estudio. Aunque cabe destacar que dentro de las pequeñas diferencias que posee los modelos de distintas arquitecturas, aquellos modelos bajo la arquitectura “An image worth 16x16 words” poseen curvas levemente más suaves que aquellos bajo la arquitectura “Attention is all you need”, gracias al particular orden de sus capas de normalización en el codificador.

## 7.2.2 Diagrama de dispersión valor predicho vs valor real

A continuación, se presenta los gráficos de dispersión usuales que derivan de los valores predichos por los modelos neuronales entrenados frente a los valores reales de la vida útil remanente (RUL) del equipo térmico definidos en el registro de variables operacionales:

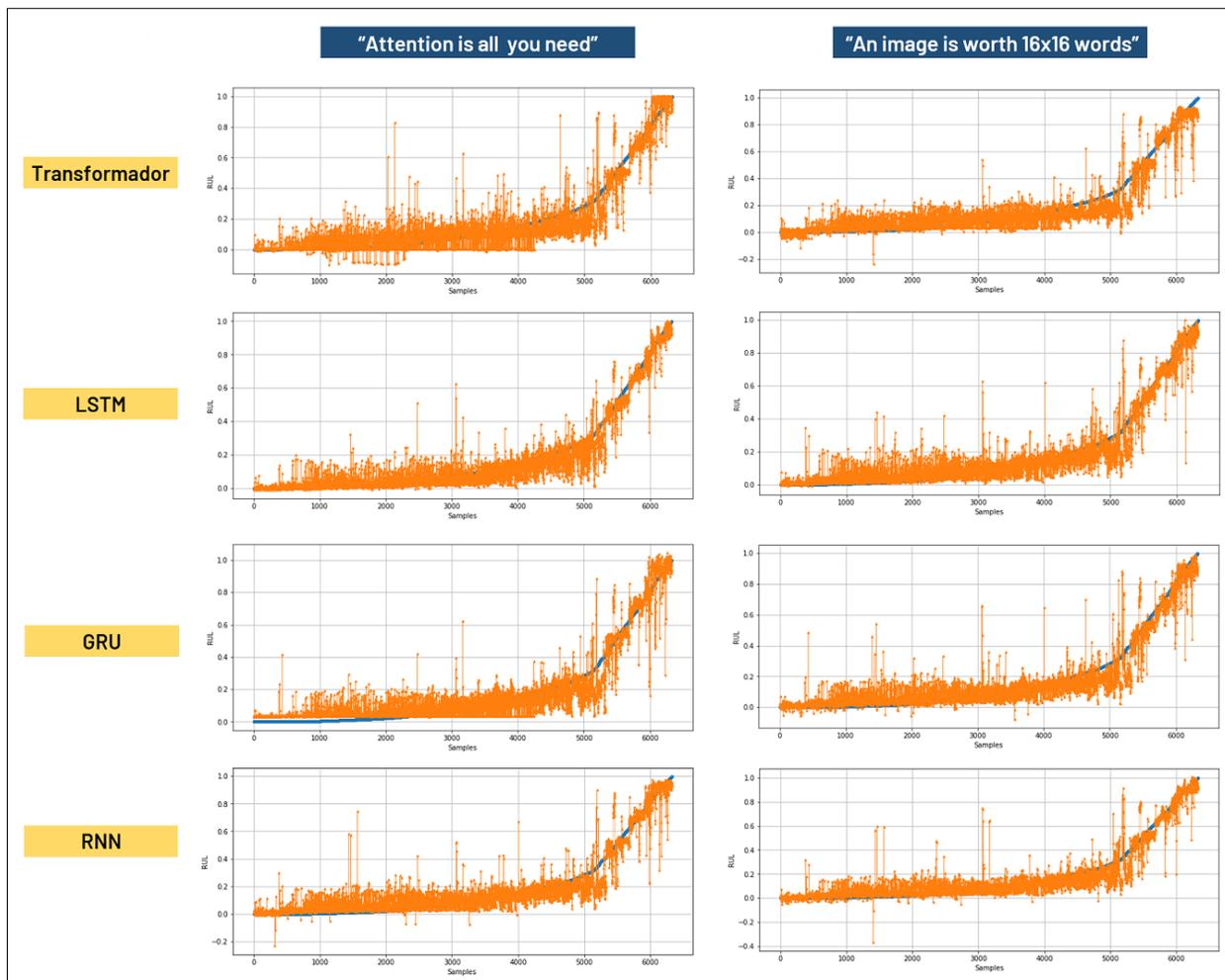


**Figura 33:** Comparación de gráfico de dispersión de valor predicho contra valor real de la vida útil remanente del equipo térmico según tipo de modelo neuronal.

En función de los gráficos de dispersión presentados en la Figura 33, se puede observar que dentro de la categoría de un mismo tipo de modelo neuronal independiente de la arquitectura que se seleccione, estos presentan un diagrama de dispersión análogo, concentrando la mayor parte de los valores estimados por los modelos entrenados cerca del origen del gráfico, esto se debe principalmente a que más del 75% de los valores RUL a estimar se encuentran más próximos al valor mínimo de RUL ( 0 horas ) que al valor máximo de RUL ( 21634 horas ) del registro, mientras el resto de predicciones minoritarias muestran la tendencia de ubicarse próximos a los valores verdaderos del RUL.

### 7.2.3 Diagrama de dispersión ordenado valor predicho vs valor real

A continuación, se presenta los gráficos de dispersión ordenados que derivan de los valores predichos por los modelos neuronales entrenados frente a los valores reales de la vida útil remanente (RUL) del equipo térmico definidos en el registro de variables operacionales:



**Figura 34:** Comparación de gráfico de dispersión ordenado de valor predicho contra valor real de la vida útil remanente del equipo térmico según tipo de modelo neuronal.

De acuerdo con los gráficos de dispersión ordenados presentados en la Figura 34, se puede observar que nuevamente dentro de la categoría de un mismo tipo de modelo neuronal independiente de la arquitectura que se seleccione, estos presentan un comportamiento similar del diagrama de dispersión ordenado, a partir del cual se desprende que todos los modelos neuronales entrenados para regresión presentan mayores problemas para estimar los valores RUL más altos que se ubican entre el rango  $[0.4,1]$  del extremo superior de la curva presentada, lo cual es causado por la baja proporción de muestras representativas de este rango de valores.

## 7.2.4 Indicadores de rendimiento por tipo de modelo

En el siguiente apartado se exponen las tablas de los indicadores de desempeño promedio de las 10 iteraciones de aprendizaje automático ejecutado sobre cada modelo neuronal bajo estudio, presentando los resultados de forma separada según la arquitectura a la que se encuentra sujeta la estructura base con la que se construyeron los modelos de tipo Transformador, LSTM, GRU y RNN, exponiendo los indicadores de rendimiento de error cuadrático medio, valor absoluto medio y coeficiente de determinación, junto al periodo de tiempo de ejecución de entrenamiento y la cantidad neta de parámetros que componen cada estructura. Para más detalle sobre el registro de estos valores dentro de las 10 iteraciones ejecutadas diríjase a la subsección Anexo B: Registro de resultados de modelos de regresión por iteración de aprendizaje.

### 7.2.4.1 Arquitectura “Attention is all you need”

*Tabla 8: Indicadores de rendimiento en regresión de modelos bajo arquitectura “Attention is all you need”.*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.949.252	4.177.092	3.983.556	3.977.924
Tiempo [h]	2,17	2,76	2,47	2,25
RMSE[.]	0,15	0,10	0,11	0,12
R2 [.]	0,54	0,83	0,74	0,74
MAE [.]	0,10	0,06	0,07	0,08

### 7.2.4.2 Arquitectura “An image is worth 16x16 words”

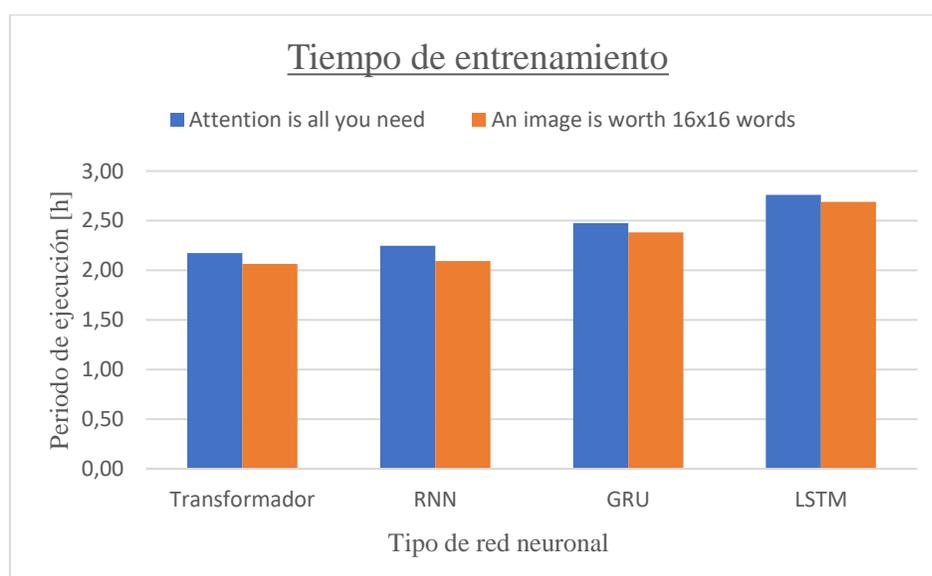
*Tabla 9: Indicadores de rendimiento en regresión de modelos bajo arquitectura “An image is worth 16x16 words”.*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.949.252	4.177.092	3.983.556	3.977.924
Tiempo [h]	2,06	2,69	2,38	2,09
RMSE[.]	0,09	0,07	0,07	0,07
R2 [.]	0,85	0,91	0,91	0,91
MAE [.]	0,06	0,05	0,05	0,05

### 7.2.5 Gráficos comparativos de rendimiento

A partir del contenido de las tablas previamente expuestas se elaboran los siguientes gráficos de rendimiento con el fin de facilitar la evaluación comparativa de los resultados entre los diferentes tipos de modelos neuronales y las distintas arquitecturas propuestas, además para quitar la ventaja de mayor capacidad de procesamiento de características que supone una mayor cantidad de parámetros que conforma la capa característica de cada tipo de modelo, es que se ha decidido presentar el desempeño de cada tipo de modelo neuronal a razón del valor del indicador de rendimiento dividido por la cantidad de parámetros neto del modelo.

#### Tiempo de ejecución de entrenamiento:



**Figura 35:** Gráfico comparativo del tiempo de entrenamiento según tipo de modelo y arquitectura.

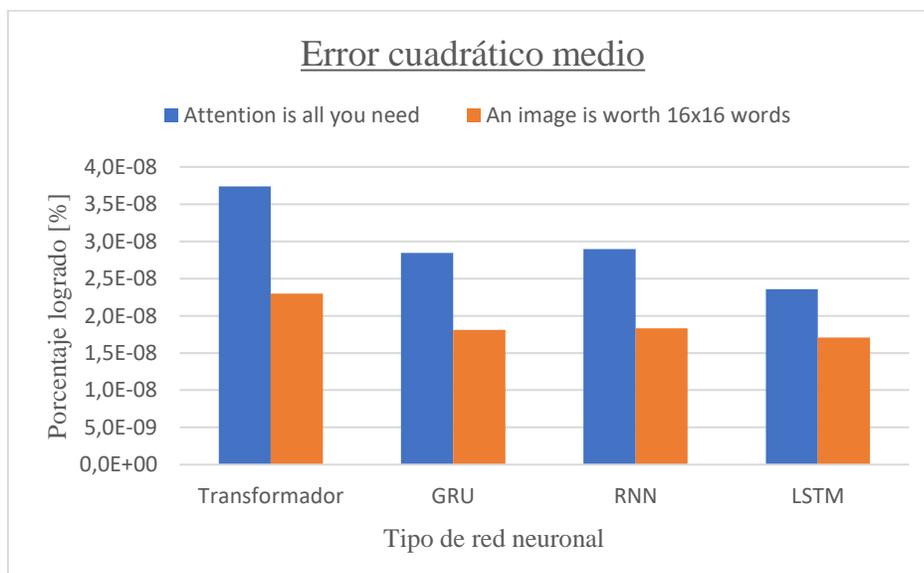
De acuerdo con el gráfico comparativo del tiempo de ejecución de entrenamiento en clasificación de la Figura 35, se puede observar que el modelo neuronal que más rápido realiza su proceso de entrenamiento corresponde al modelo tipo Transformador, debido a que el funcionamiento de su celda distintiva (mecanismos de atención) se enfoca exclusivamente de identificar patrones relevantes, a partir de los cuales posteriormente aplica un filtro global de datos, suprimiendo aquellos valores que no son importantes para desarrollar características útiles que contribuyan con una buena predicción del valor del RUL de las muestras, permitiendo un procesamiento parcialmente de la serie temporal de entrada, acelerando la generación de una respuesta del modelo y acortando los tiempos de ejecución de las épocas de entrenamiento.

En segundo lugar, se encuentra el tipo de modelo RNN, debido a que el funcionamiento y composición de su celda distintiva es la más simple dentro de los tipos de modelo bajo estudio, a costo de un rango de memoria de corto plazo y menor capacidad para desarrollar características complejas, aunque esto le confiere un menor tiempo de procesamiento de datos que resulta finalmente en una pequeña reducción del tiempo del desarrollo de respuesta del modelo y del periodo de tiempo de ejecución del entrenamiento, sin embargo, este tipo de modelos neuronales de igual forma no posee un prefiltro de la información como el modelo transformador, perdiendo tiempo al procesar la serie temporal completa de entrada.

En tercer lugar, se encuentra el tipo de modelo GRU, esto se debe a que presenta en su capa característica una celda distintiva que posee un grado de complejidad de funcionamiento y composición en un nivel intermedio entre el modelo RNN y el modelo LSTM, en teoría posee los mismos vectores de salida que el modelo RNN, pero al poseer un número mayor de subcomponentes al interior de su celda distintiva (puerta de reinicio y puerta de actualización) le permite construir características más complejas a costo de un mayor tiempo de ejecución, aunque cabe destacar que esta estructura no es suficiente para alcanzar a superar la capacidad de procesamiento del modelo LSTM, por ende esto provoca que el periodo de tiempo de ejecución de esta capa característica sea mayor al modelo RNN, pero menor al modelo LSTM, lo cual se refleja finalmente en el tiempo neto de ejecución del entrenamiento del modelo neuronal.

Por último el modelo neuronal más lento en ejecutar su proceso de entrenamiento corresponde al modelo LSTM, esto se debe a que posee en su capa característica la más compleja celda distintiva de entre los tipos de modelos neuronales bajo estudio, contando con una mayor cantidad de subcomponentes y conexiones que le permite extraer características más elaboradas a costa de requerir un mayor tiempo de procesamiento de datos, lo que repercute finalmente en un mayor periodo de tiempo de ejecución de las épocas de entrenamiento.

### **Indicador de rendimiento de error cuadrático medio:**



**Figura 36:** Gráfico comparativo del indicador de rendimiento error cuadrático medio según tipo de modelo y arquitectura.

De acuerdo con el gráfico comparativo del error cuadrático medio de los modelos para regresión de la Figura 36, se desprende que el modelo neuronal menos eficiente para estimar el RUL de las muestras de entrada corresponde al modelo Transformador, debido a que su celda distintiva (mecanismos de atención) y por consecuencia su capa característica, se enfoca exclusivamente a realizar un filtro de la información relevante de los datos de la ventana temporal de entrada en vez de desarrollar nuevas características útiles para estimar el valor del RUL de la muestra, como sí lo hacen en esta misma etapa el resto de los tipos de modelos neuronales, almacenando estas nuevas características en su respectiva unidad de memoria, de esta forma el modelo transformador pierde capacidad de elaboración de características útiles, propiedad fundamental para la operación de regresión requerida.

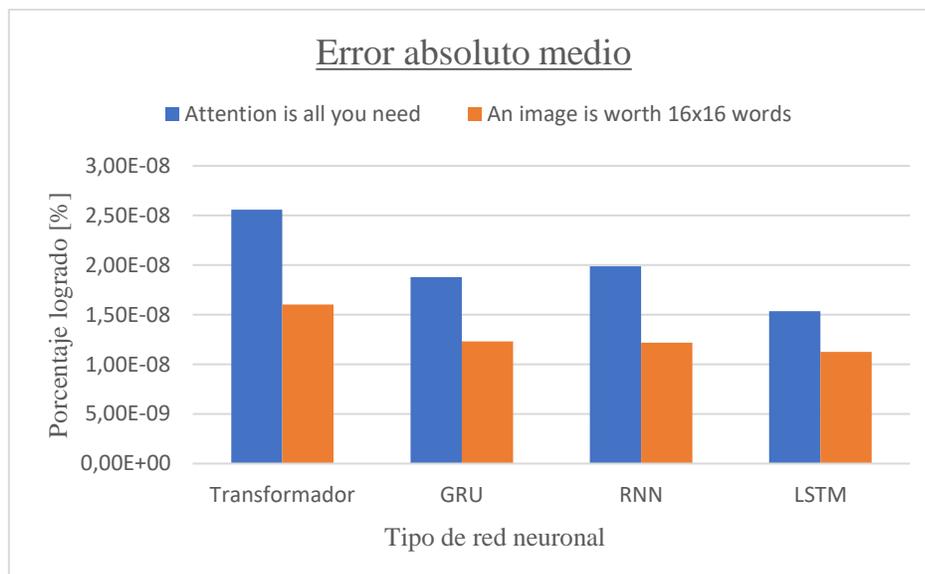
En tercer lugar, se tiene el tipo de modelo GRU, ya que a pesar de que su celda distintiva le otorga una mayor capacidad de elaboración de características complejas y una reducción del efecto de desvanecimiento del gradiente que mejora la memoria de largo plazo sobre los datos de entrada iniciales, esto último en particular resulta contraproducente para la tarea de regresión, ya que al tener una atención más uniforme sobre todos los datos de la ventana temporal de entrada, esto dificulta que el modelo neuronal pueda prestar especial atención sobre las últimas instancias de las ventanas temporales que resultan estar más ligadas con el valor del RUL a predecir.

En segundo lugar, se tiene el tipo de modelo neuronal RNN, a causa de que este tipo de modelo neuronal presenta la celda distintiva más simple en comparación al resto de modelos, lo subsecuentemente genera que este tipo de modelos se vea más perjudicado por el efecto del desvanecimiento del gradiente reduciendo la memoria del modelo sobre los datos iniciales, sin embargo, para este caso repercute positivamente en su desempeño, ya que esto significa que el modelo neuronal RNN concentra una mayor atención sobre las últimas instancias de la ventana temporal de entrada para la elaboración de características, datos que resultan ser especialmente más útiles para predecir el valor de la vida útil remanente del equipo térmico.

Finalmente, el mejor tipo de modelo neuronal para estimar el RUL de las muestras de entrada corresponde al modelo neuronal de tipo LSTM, gracias a que presenta más capacidad para elaborar características complejas considerando simultáneamente el panorama global de los datos de la ventana temporal como también exclusivamente las últimas instancias de la ventana temporal, estas propiedades son otorgadas por su mayor número de subestructuras (puerta de entrada, puerta de olvido y puerta de salida) y por sus dos unidades de memoria de largo plazo y memoria de corto plazo presentes en su celda distintiva.

Cabe destacar que a partir del gráfico comparativo del error cuadrático medio de los modelos de regresión, se observa que independiente de la arquitectura a la que están sujetas los tipos de modelos neuronales estos presentan el mismo orden creciente de desempeño, no obstante, se puede ver que los modelos bajo la arquitectura “Attention is all you need” presentan peores resultados que la arquitectura “An image worth 16 x 16 words”, esto se debe a que la primera arquitectura se especializa en extraer características de la relación de sucesión de los datos de las series temporales, en este caso de la evolución temporal individual de los parámetros de funcionamiento, mientras la segunda se enfoca más en la elaboración de características a partir de patrones globales de la ventana de tiempo a procesar, que para este caso se centra en extraer características de cada instancia de medición que compone la ventana temporal.

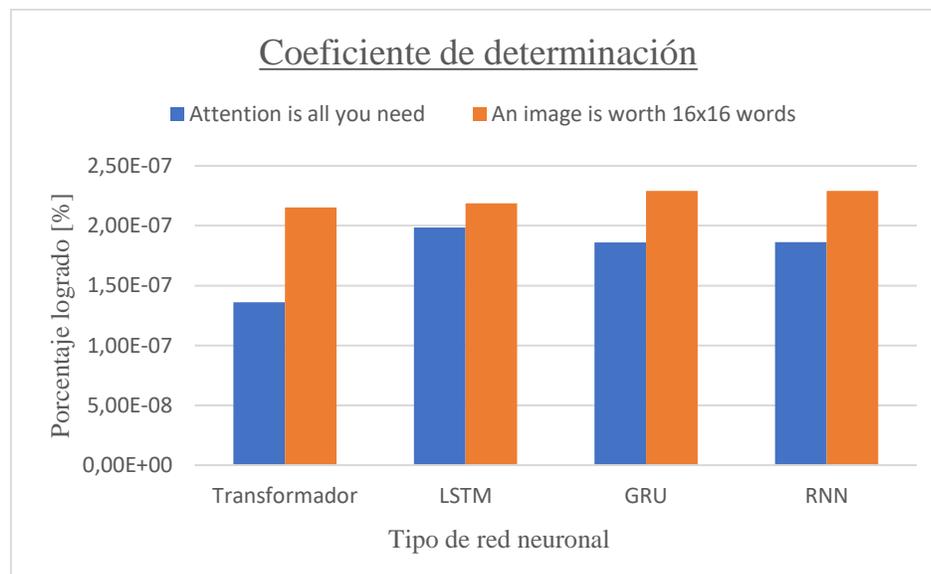
**Indicador de rendimiento de error absoluto medio:**



**Figura 37:** Gráfico comparativo del indicador de rendimiento error absoluto medio según tipo de modelo y arquitectura.

A partir del gráfico comparativo del error absoluto medio de los modelos para regresión de la Figura 37, expone el mismo comportamiento de desempeño expuesto en gráfico comparativo previo del error cuadrático medio, esto se debe a que la definición de ambos indicadores están sujetos a formulas muy similares entre sí, por ende las justificaciones previamente expuestas sobre comportamiento de los resultados de cada tipo de modelo neuronal y las diferentes arquitecturas, cobran la misma validez para explicar el comportamiento del indicador de rendimiento de error absoluto medio.

### Indicador de rendimiento de coeficiente de determinación



**Figura 38:** Gráfico comparativo del indicador de rendimiento coeficiente de determinación según tipo de modelo y arquitectura.

De acuerdo con el gráfico comparativo del coeficiente de determinación de los modelos para regresión de la Figura 38, se observa que el tipo de modelo neuronal que presenta la menor calidad de ajuste para replicar los valores del RUL de las muestras, corresponde al modelo de tipo transformador, a causa de una menor capacidad para elaborar características en comparación al resto de modelos, ya que en su capa característica no se enfoca exclusivamente a realizar esta acción sino a realizar un filtro de información, lo que resulta en un estimación numérica más alejada del valor verdadero del RUL de las muestras.

En tercer lugar, se tiene el tipo de modelo LSTM, porque si bien su sofisticada celda distintiva le favorece para disminuir el indicador de rendimiento del error cuadrático medio, esto no le favorece para acercar de manera uniforme cada valor predicho a sus valores objetivo, pues gracias a su gran capacidad de elaboración de características tenderá a desarrollar un método de procesamiento de datos que considerará más las diferencias entre las muestras de entrada, resultando en varias predicciones puntuales que se alejan demasiado de sus valores objetivos, aumentando el porcentaje de variación numérica y por ende disminuyendo su valor de coeficiente de determinación.

En segundo lugar, se tiene el tipo de modelo neuronal GRU, debido a que su celda distintiva presenta una estructura de funcionamiento que reduce el efecto del desvanecimiento de gradiente a un nivel intermedio entre el modelo LSTM y el modelo RNN.

Finalmente, aquel tipo de modelo neuronal que mejor ajusta sus predicciones con los valores reales del RUL corresponde al modelo RNN, ya que al presentar la más simple de las celdas distintivas, tiende a desarrollar un método de procesamiento de datos que toman en cuenta menos las diferencias entre las muestras de entrada, resultando en predicciones uniformemente distanciadas de sus valores reales del RUL, disminuyendo el porcentaje de variación numérica y por ende aumentando su valor de coeficiente de determinación.

Sobre la base del gráfico comparativo del coeficiente de determinación de los modelos de regresión, se deduce que el orden creciente de desempeño de cada tipo de modelo es independiente de su arquitectura base, es decir, ya sea que se seleccione la arquitectura “Attention is all you need” o la arquitectura “An image worth 16x16 words”, el modelo con el mejor desempeño de su coeficiente de determinación corresponde al modelo RNN, seguido del modelo GRU, continuando con el modelo LSTM y al final se encuentra el modelo Transformador. Cabe destacar que, a diferencia de los otros indicadores de rendimiento, aquí los modelos neuronales que presentan un mejor desempeño con respecto al coeficiente de determinación corresponden a aquellos modelos sujetos a la arquitectura “An image worth 16x16 words”, esto se debe a que como la función de pérdida del proceso de entrenamiento se enfoca principalmente en disminuir el valor del error cuadrático medio que resulta de la comparación de sus valores estimados y los valores verdades del RUL, los modelos bajo la arquitectura “Attention is all you need” al presentar la propiedad de especializarse en extraer características complejas del orden de los datos de las series temporales y no a partir de los datos de un mismo tiempo en la ventana temporal como la arquitectura “An image worth 16x16 words”, tendrán la tendencia de disminuir drásticamente el valor de su error cuadrático medio aplicando la estrategia de aproximar la mayoría de los valores predichos a su valores reales dejando un grupo minoritario de valores predichos que se alejan bastante de sus valores reales, por el contrario el otro modelo tendrá la tendencia de disminuir gradualmente el valor de su error cuadrático medio aplicando la estrategia de aproximar uniformemente los valores predichos a los valores reales del RUL, disminuyendo así la variabilidad de los valores de la predicciones.

## 7.3 Análisis de sensibilidad sobre modelos de clasificación

### 7.3.1 Función de pérdida de proceso de aprendizaje automático

Con el objetivo de sintetizar el contenido de la actual sección de “Resultado y Discusión”, se ha decidido presentar los gráficos de función de pérdida de los modelos alternativos en la subsección de Anexo C: Funciones de pérdida de modelos alternativos para clasificación, en la cual se presenta la evolución del indicador de rendimiento de “Exactitud” del grupo de datos de entrenamiento y validación obtenidos a lo largo de 50 épocas de entrenamiento que se desprenden del aprendizaje automático de los nuevos modelos alternativos para clasificación, que en comparación a los modelos neuronales iniciales de tipo Transformador, LSTM, GRU y RNN bajos las dos arquitecturas propuestas “Attention is all you need ” y “An image worth 16x16 words”, estos nuevos modelos presentan variaciones del número de parámetros que componen la estructura del codificador, modificando la cantidad de bloques secuenciales que lo constituyen.

A partir de las funciones de pérdida exhibidas se observa que no hay ningún modelo neuronal entrenado que sufra de un sobreajuste durante el entrenamiento, puesto que independiente de la arquitectura, tipo de modelo o variación del modelo que se seleccione, se sigue manteniendo la tendencia de que la curva de aprendizaje del grupo de datos de entrenamiento converge con la curva de aprendizaje del grupo de datos de validación para luego decaer simultáneamente, por otro lado también se desprende que si bien los modelos de tipo transformador mantienen más consistentemente el comportamiento de la curva de la función de pérdida independiente del número de parámetros que integra su codificador, para el resto de los tipos de modelos LSTM, GRU, RNN se muestra que a medida que aumenta el número de parámetros que compone su codificador, las funciones de pérdida se vuelve más irregulares, exhibiendo mayor cantidad de peaks a lo largo de las épocas, lo cual se debe a que estos tipos modelos neuronales poseen la capacidad de elaborar características más complejas por su celda distintiva, que al aumentar la cantidad de codificadores secuenciales, aumenta el número de estas celdas distintivas permitiendo que aumente el grado de desarrollo y complejidad de las características que el codificador entrega, impactando con mayor fuerza un cambio en el valor de los parámetros de peso de la red a causa del optimizador, produciendo así cambios de desempeño más drásticos a través de las épocas de entrenamiento.

Cabe destacar que nuevamente el modelo de tipo RNN bajo la arquitectura “Attention is all you need” presenta la peor de las curvas de aprendizaje con respecto al resto de tipos de modelos, además específicamente en los modelos RNN con mayor número de parámetros compuestos por una cantidad de 8 bloques codificadores y 10 bloques codificadores respectivamente, presentan una curva de aprendizaje con comportamiento que no decae en las épocas finales del entrenamiento, sino que ascienden divergiendo la función de pérdida del grupo de datos de entrenamiento y la función de pérdida del grupo de datos de validación, esto se debe a que el aumento del número de capas característica acentúa el efecto del desvanecimiento de gradiente, reduciendo aún más su memoria de corto plazo, dificultando más en cada época que el optimizador modifique correctamente los valores de los parámetros de pesos para mejorar el rendimiento.

### 7.3.2 Matriz de confusión

De acuerdo con sintetizar el contenido de la actual sección de “Resultado y Discusión”, se ha decidido presentar las matrices de confusión de los modelos alternativos en la subsección de Anexo D: Matrices de confusión de modelos alternativos para clasificación, en donde se presentan las matrices de confusiones usuales que se desprenden del proceso de prueba de cada tipo de modelo neuronal alternativo bajo las dos arquitecturas propuestas, a partir de los cuales se extrae la información de los indicadores de rendimiento de: exactitud, exhaustividad, precisión, puntaje valor F1, gráfico y área de la curva característica operativa del receptor.

De acuerdo con estos resultados, podemos ver que el desempeño logrado por los tipo de modelos neuronales Transformador, LSTM, GRU y RNN diseñados bajo las dos arquitecturas “Attention is all you need” y “An image worth 16x16 words” para identificar estados de operación, no muestra directamente una relación proporcional con la cantidad de parámetros que componen sus codificador, ya que se puede ver que los valores de las matrices de confusión son similares entre sí para todos los modelos alternativos entrenados.

### **7.3.3 Curva característica de funcionamiento del receptor**

Con el objetivo de sintetizar el contenido de la actual sección de “Resultado y Discusión”, se ha decidido presentar las curvas características operativas del receptor de los modelos alternativos en la subsección de Anexo E: Curva característica de modelos alternativos para clasificación, en donde se presentan las curvas ROC usuales que se desprenden del proceso de entrenamiento de cada tipo de modelo neuronal alternativo bajo las dos arquitecturas propuestas enfocadas en la clasificación de estados de operación.

De acuerdo con estos resultados, se puede ver que las curvas ROC de todos los tipos de modelos neuronales diseñados bajo cualquier arquitectura propuesta, muestran el mismo comportamiento lineal previamente descrito independientemente de la cantidad de parámetros que conforman su codificador, aun que dentro de todos los tipos de modelos aquel que presenta una curva ROC menos regular corresponde al modelo RNN, pues el valor de la pendiente del tramo inicial de la curva ROC varía más en este tipo de modelo que en el resto, esto se debe principalmente a que es el tipo de modelo más afectado por desvanecimiento de gradiente, aumentado más la variabilidad del fruto de su entrenamiento.

### 7.3.4 Indicadores de rendimiento por variantes de modelo

En el siguiente apartado se exponen las tablas de los indicadores de desempeño promedio de las 10 iteraciones de aprendizaje automático ejecutado sobre cada modelo neuronal alternativo bajo estudio, presentando los resultados de forma separada según la arquitectura a la que se encuentra sujeta la estructura base con la que se construyeron los modelos de tipo Transformador, LSTM, GRU y RNN, exponiendo los indicadores de rendimiento de exactitud, exhaustividad, precisión, puntaje valor F1 y valor del área bajo la curva ROC, junto al periodo de tiempo de ejecución de entrenamiento y la cantidad neta de parámetros que componen cada estructura. Para más detalle sobre el registro de estos valores dentro de las 10 iteraciones ejecutadas diríjase a la subsección de Anexo F: Registro de resultados de modelos alternativos en clasificación.

#### 7.3.4.1 Arquitectura “Attention is all you need”:

##### a) Modelo neuronal alternativo de dos codificadores

*Tabla 10: Indicadores de rendimiento en clasificación de modelos con 2 codificadores bajo arquitectura “Attention is all you need”.*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.351.621	3.408.581	3.360.197	3.358.789
Tiempo [h]	0,48	0,51	0,49	0,48
Exactitud [%]	98,54	98,46	98,59	98,06
Precisión [%]	94,86	92,26	93,93	91,77
Exhaustividad [%]	89,80	91,76	91,30	87,68
Puntaje valor F1 [%]	92,19	91,98	92,56	89,64
Área curva ROC [%]	94,64	95,47	95,33	93,42

##### b) Modelo neuronal alternativo de cuatro codificadores

*Tabla 11: Indicadores de rendimiento en clasificación de modelos con 4 codificadores bajo arquitectura “Attention is all you need”.*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.551.173	3.665.093	3.568.325	3.565.509
Tiempo [h]	0,78	0,88	0,84	0,79
Exactitud [%]	98,59	98,59	98,50	98,04
Precisión [%]	93,32	94,06	93,35	92,73
Exhaustividad [%]	92,04	91,19	90,87	86,46
Puntaje valor F1 [%]	92,62	92,56	92,06	89,39
Área curva ROC [%]	95,66	95,28	95,09	92,87

c) **Modelo neuronal alternativo de seis codificadores**

*Tabla 12: Indicadores de rendimiento en clasificación de modelos con 6 codificadores bajo arquitectura "Attention is all you need".*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.750.725	3.921.605	3.776.453	3.772.229
Tiempo [h]	0,84	1,17	0,98	0,87
Exactitud [%]	98,45	98,55	98,44	97,35
Precisión [%]	95,10	93,11	93,81	90,50
Exhaustividad [%]	88,52	91,74	89,74	81,02
Puntaje valor F1 [%]	91,63	92,39	91,68	85,36
Área curva ROC [%]	94,01	95,51	94,54	89,66

d) **Modelo neuronal alternativo de ocho codificadores**

*Tabla 13: Indicadores de rendimiento en clasificación de modelos con 8 codificadores bajo la arquitectura "Attention is all you need".*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.950.277	4.178.117	3.984.581	3.978.949
Tiempo [h]	1,1	1,5	1,3	1,2
Exactitud [%]	98,5	98,5	98,4	96,9
Precisión [%]	95,3	92,2	92,1	93,6
Exhaustividad [%]	88,6	91,9	91,5	73,3
Puntaje valor F1 [%]	91,8	92,0	91,7	82,0
Área curva ROC [%]	94,1	95,5	95,3	86,4

e) **Modelo neuronal alternativo de diez codificadores**

*Tabla 14: Indicadores de rendimiento en clasificación de modelos con 10 codificadores bajo arquitectura "Attention is all you need".*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	4.149.829	4.434.629	4.192.709	4.185.669
Tiempo [h]	1,4	1,7	1,6	1,3
Exactitud [%]	98,2	98,5	97,5	96,7
Precisión [%]	94,1	91,8	92,5	93,8
Exhaustividad [%]	86,7	92,3	80,7	70,1
Puntaje valor F1 [%]	90,2	92,0	86,0	80,1
Área curva ROC [%]	93,1	95,7	90,0	84,8

### 7.3.4.2 Arquitectura “An image is worth 16x16 words”:

#### a) Modelo neuronal alternativo de dos codificadores

*Tabla 15: Indicadores de rendimiento en clasificación de modelos con 2 codificadores bajo arquitectura “An image is worth 16x16 words”*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.351.621	3.408.581	3.360.197	3.358.789
Tiempo [h]	0,45	0,54	0,52	0,46
Exactitud [%]	98,46	98,34	98,43	98,14
Precisión [%]	94,50	92,57	93,30	91,64
Exhaustividad [%]	89,33	90,08	90,23	88,80
Puntaje valor F1 [%]	91,77	91,23	91,69	90,13
Área curva ROC [%]	94,38	94,65	94,77	93,96

#### b) Modelo neuronal alternativo de cuatro codificadores

*Tabla 16: Indicadores de rendimiento en clasificación de modelos con 4 codificadores bajo arquitectura “An image is worth 16x16 words”*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.551.173	3.665.093	3.568.325	3.565.509
Tiempo [h]	0,74	0,81	0,80	0,75
Exactitud [%]	98,50	98,48	98,53	98,08
Precisión [%]	92,75	92,10	93,32	92,33
Exhaustividad [%]	91,51	92,27	91,30	87,45
Puntaje valor F1 [%]	92,10	92,12	92,27	89,76
Área curva ROC [%]	95,38	95,71	95,30	93,33

#### c) Modelo neuronal alternativo de seis codificadores

*Tabla 17: Indicadores de rendimiento en clasificación de modelos con 6 codificadores bajo arquitectura “An image is worth 16x16 words”*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.750.725	3.921.605	3.776.453	3.772.229
Tiempo [h]	0,85	1,17	1,11	0,99
Exactitud [%]	98,44	98,32	98,53	98,07
Precisión [%]	95,07	90,70	93,76	91,17
Exhaustividad [%]	89,09	92,01	90,82	88,55
Puntaje valor F1 [%]	91,80	91,30	92,24	89,81
Área curva ROC [%]	94,36	95,50	95,09	93,82

**d) Modelo neuronal alternativo de ocho codificadores**

*Tabla 18: Indicadores de rendimiento en clasificación de modelos con 8 codificadores bajo arquitectura "An image is worth 16x16 words"*

<b>Indicador de rendimiento</b>	<b>Transformador</b>	<b>LSTM</b>	<b>GRU</b>	<b>RNN</b>
<b>Parámetros [.]</b>	3.950.277	4.178.117	3.984.581	3.978.949
<b>Tiempo [h]</b>	1,10	1,44	1,36	1,19
<b>Exactitud [%]</b>	98,5	98,3	98,6	98,1
<b>Precisión [%]</b>	94,8	90,6	93,4	90,5
<b>Exhaustividad [%]</b>	89,7	92,4	91,7	89,3
<b>Puntaje valor F1 [%]</b>	92,1	91,4	92,5	89,8
<b>Área curva ROC [%]</b>	94,7	95,7	95,5	94,1

**e) Modelo neuronal alternativo de diez codificadores**

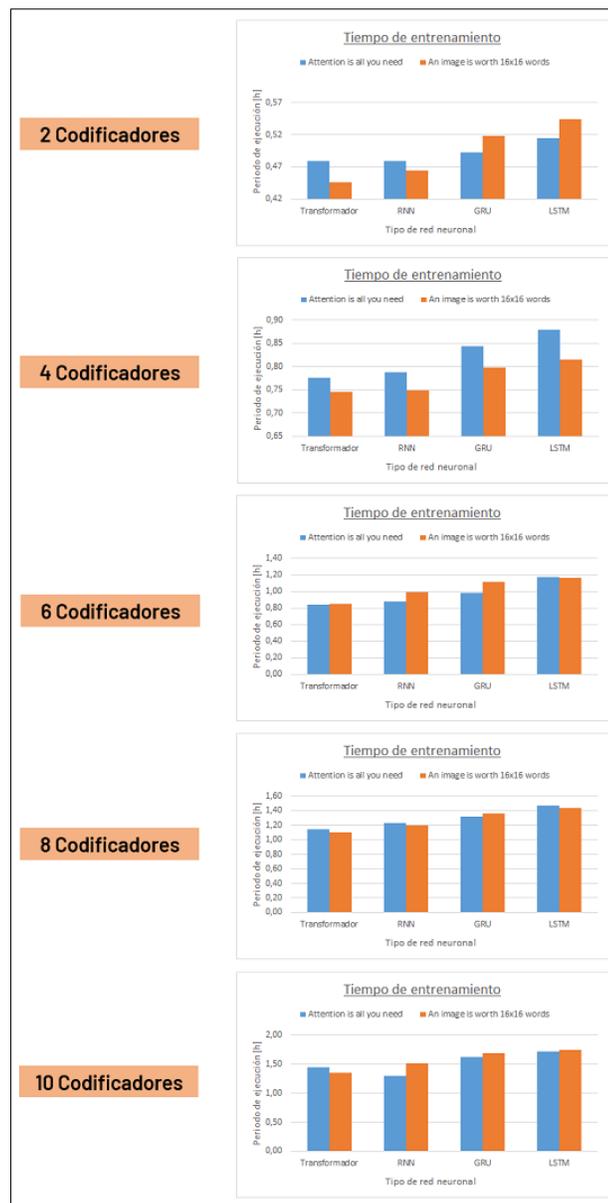
*Tabla 19: Indicadores de rendimiento en clasificación de modelos con 10 codificadores bajo arquitectura "An image is worth 16x16 words"*

<b>Indicador de rendimiento</b>	<b>Transformador</b>	<b>LSTM</b>	<b>GRU</b>	<b>RNN</b>
<b>Parámetros [.]</b>	4.149.829	4.434.629	4.192.709	4.185.669
<b>Tiempo [h]</b>	1,34	1,74	1,69	1,52
<b>Exactitud [%]</b>	98,5	98,4	98,5	98,1
<b>Precisión [%]</b>	93,8	91,6	93,6	90,9
<b>Exhaustividad [%]</b>	90,0	91,8	90,8	88,8
<b>Puntaje valor F1 [%]</b>	91,8	91,7	92,1	89,8
<b>Área curva ROC [%]</b>	94,7	95,5	95,1	93,9

### 7.3.5 Gráficos comparativos de rendimiento

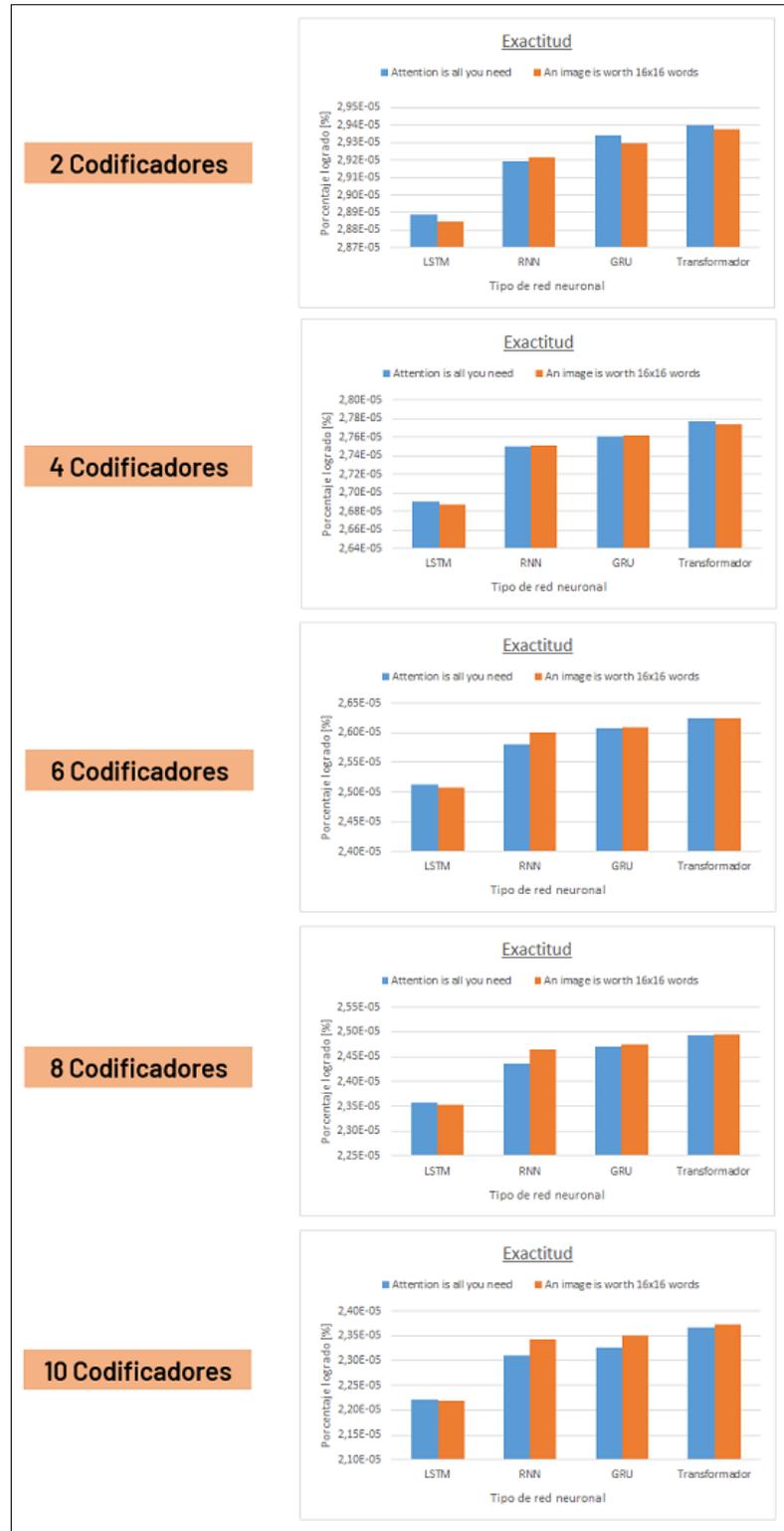
A partir del contenido de las tablas del punto anterior se elaboran los siguientes gráficos de rendimiento con el fin de facilitar la evaluación comparativa de los resultados de los diferentes tipos de modelos neuronales alternativos, bajo distintas arquitecturas y conformados por un variado número de parámetros, además para quitar la ventaja de mayor capacidad de procesamiento de características que supone una mayor cantidad de parámetros que conforma la capa característica de cada tipo de modelo, es que se ha decidido presentar el desempeño de cada tipo de modelo neuronal a razón del valor del indicador de rendimiento dividido por la cantidad de parámetros neto del modelo.

#### **Tiempo de ejecución de entrenamiento de modelos alternativos**



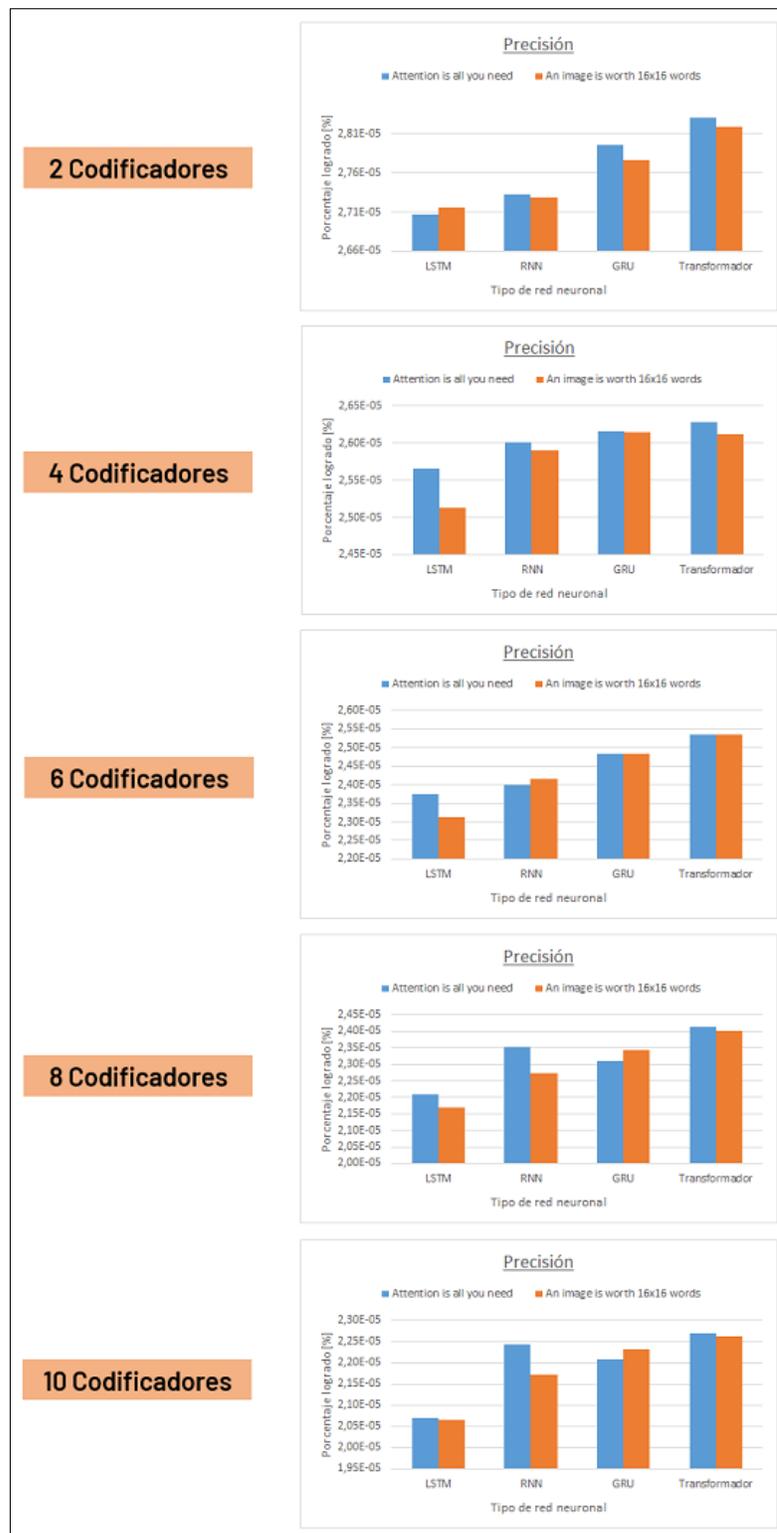
**Figura 39:** Gráfico comparativo de tiempo de entrenamiento según tipo de modelo, arquitectura y número de codificadores que los conforman.

**Indicador de rendimiento de exactitud de modelos alternativos**



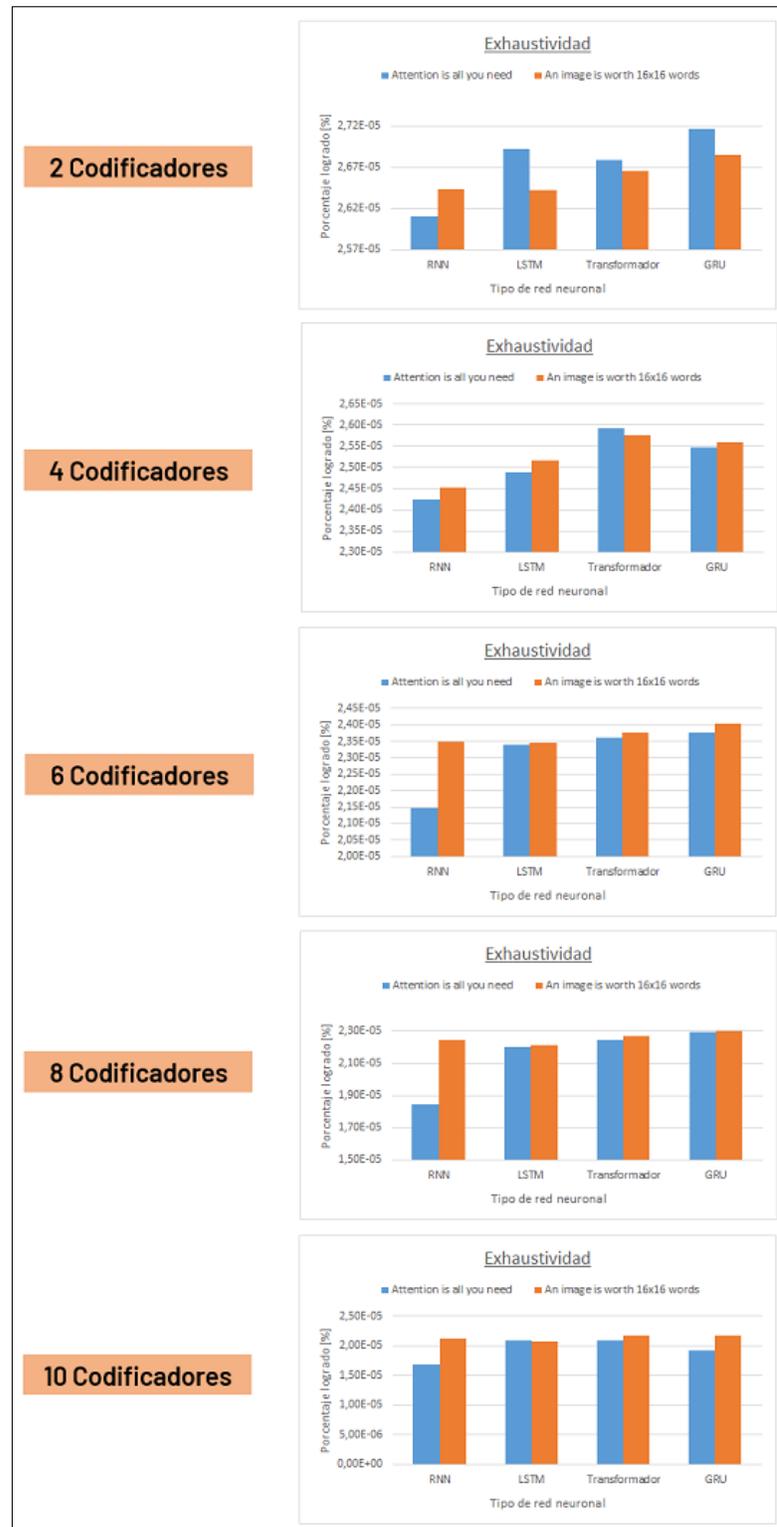
**Figura 40:** Gráfico comparativo del indicador de rendimiento Exactitud según tipo de modelo, arquitectura y número de codificadores que los conforman.

**Indicador de rendimiento de precisión de modelos alternativos**



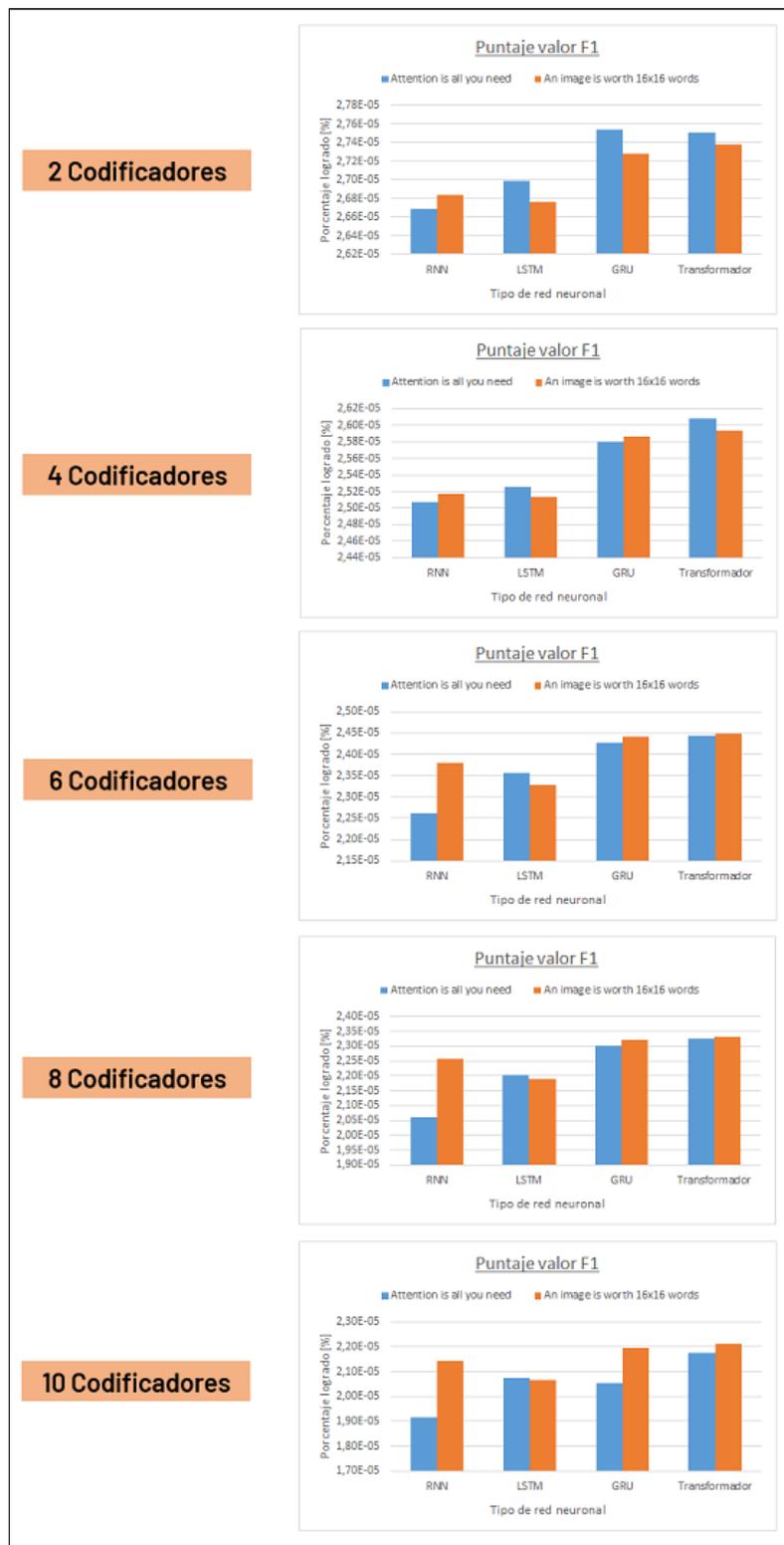
**Figura 41:** Gráfico comparativo del indicador de rendimiento precisión según tipo de modelo, arquitectura y número de codificadores que los conforman.

**Indicador de rendimiento de exhaustividad de modelos alternativos**



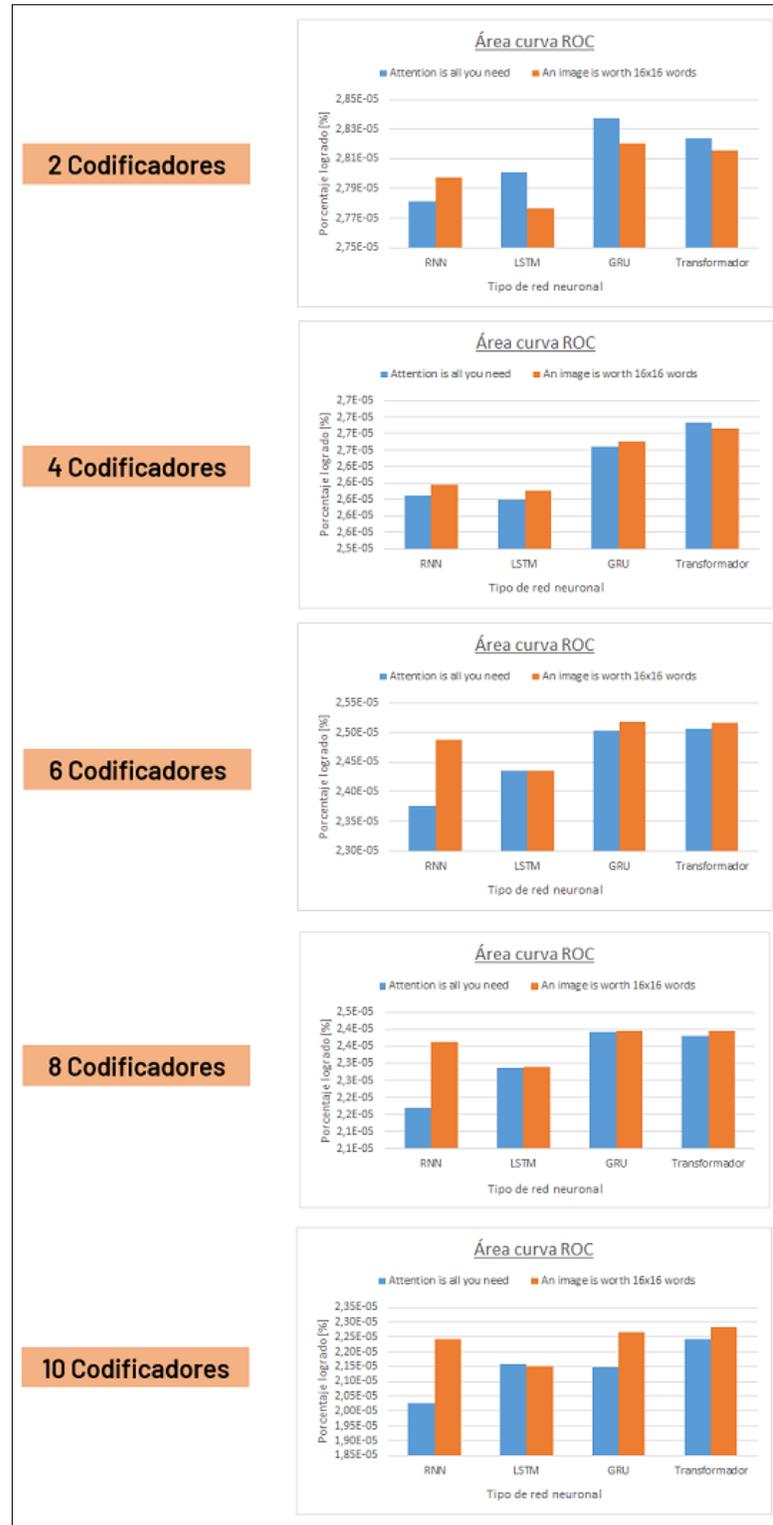
**Figura 42:** Gráfico comparativo del indicador de rendimiento exhaustividad según tipo de modelo, arquitectura y número de codificadores que los conforman.

**Indicador de rendimiento de puntaje valor F1 de modelos alternativos**



**Figura 43:** Gráfico comparativo del indicador de rendimiento puntaje valor F1 según tipo de modelo, arquitectura y número de codificadores que los conforman.

**Indicador de rendimiento de área bajo la curva ROC de modelos alternativos**



**Figura 44:** Gráfico comparativo del indicador de rendimiento Área bajo curva característica según tipo de modelo, arquitectura y número de codificadores que los conforman.

En función de los gráficos comparativos de los indicadores de rendimiento de tiempo de ejecución de entrenamiento, exactitud, exhaustividad, precisión, puntaje valor F1 y valor del área bajo la curva ROC según el número de bloques codificadores que conforma su estructura, en general para cada indicador de rendimiento se expone el mismo orden de desempeño por tipo de modelo que se describió anteriormente para el caso particular de los modelos de clasificación compuestos solamente por 8 bloques de codificadores en su estructura, esto quiere decir que las diferencias de desempeño entre tipos de modelos neuronales diferentes son independiente de la cantidad de bloques que conforma la estructura de su codificador, con la particular excepción que dentro del indicador de rendimiento “área bajo la curva ROC” se presenta en primer lugar de desempeño el modelo Transformador y en segundo lugar el modelo GRU, corroborando el supuesto que solamente por una desafortunada coincidencia estadística únicamente para los modelos compuestos por 8 codificadores se muestra una relación invertida de este orden.

Por otro lado, si se observa el desempeño según las dos arquitecturas propuestas, se encuentra que la tendencia global es que la arquitectura “An image worth 16x16 words” supere en rendimiento a la arquitectura “Attention is all you need” por las mismas razones descritas para los modelos originales, no obstante, también se observa que en los modelos constituidos por menores parámetros en su estructura, por ejemplo, los modelos que solo tienen 2 bloques de codificadores, prevalece el rendimiento de los modelos bajo la arquitectura “Attention is all you need”, esto se debe a que la otra arquitectura al contar con tan bajo número de parámetros no es capaz de identificar tan eficientemente patrones distintivos de los dos estados operacional como para rivalizar con la utilidad que representan las características que elabora la arquitectura “Attention is all you need”. Cabe destacar también que la variabilidad en los resultados que exponen los gráficos comparativos se adjudica principalmente al limitado número de iteraciones de aprendizaje evaluados.

## 7.4 Análisis de sensibilidad sobre modelos de regresión

### 7.4.1 Función de pérdida de proceso de aprendizaje automático

Con el objetivo de sintetizar el contenido de la actual sección de “Resultado y Discusión”, se ha decidido presentar los gráficos de función de pérdida de los modelos alternativos en la subsección Anexo G: Funciones de pérdida para regresión de modelos alternativos, en la cual se presenta la evolución del indicador de rendimiento de “Error cuadrático medio” del grupo de datos de entrenamiento y validación obtenidos a lo largo de 100 épocas de entrenamiento que se desprenden del aprendizaje automático de los nuevos modelos alternativos para regresión, que en comparación a los modelos neuronales iniciales de tipo Transformador, LSTM, GRU y RNN bajos las dos arquitecturas propuestas “Attention is all you need ” y “An image worth 16x16 words”, estos nuevos modelos presentan variaciones del número de parámetros que componen la estructura del codificador, modificando la cantidad de bloques secuenciales que lo constituyen.

A partir de las funciones de pérdida exhibidas se observa que no hay presencia de un sobreajuste de datos en ningún modelo neuronal entrenado, puesto que independiente de la arquitectura, tipo de modelo o variación del modelo que se seleccione, se sigue manteniendo la tendencia de que la curva de aprendizaje del grupo de datos de entrenamiento converge con la curva de aprendizaje del grupo de datos de validación para luego decaer simultáneamente.

De forma análoga al análisis previo hecho sobre los modelos compuestos únicamente por 8 codificadores, aquí se observa el mismo comportamiento que independiente de la arquitectura que se seleccione dentro del mismo tipo de modelo se presentan curvas de función de pérdida muy similares entre sí, como también se observa a partir de sus pequeñas diferencias que los modelos neuronales bajo la estructura “Attention is all you need” presentan gráficos con una curva de aprendizaje más irregular y menos suave que aquellos modelos bajo la arquitectura “An image worth 16x16 words”, fruto del orden específico de las capas de normalización en el codificador.

Por otro lado, si se observa bien detenidamente la forma en que evoluciona la curva de aprendizaje a través de las cantidades variables de bloques secuenciales que componen al codificador, es decir, a medida que aumenta el número de parámetros de la estructura, las funciones de pérdida se vuelven más irregulares, exhibiendo mayor cantidad de peaks a lo largo de las épocas, debido a que al aumentar la cantidad de parámetros, aumenta la cantidad de configuraciones posibles que el optimizador debe probar para mejorar el rendimiento de la exactitud del modelo.

### **7.4.2 Diagrama de dispersión valor predicho vs valor real:**

De acuerdo con sintetizar el contenido de la actual sección de “Resultado y Discusión”, se ha decidido presentar los gráficos de dispersión de los modelos alternativos en la subsección de Anexo H: Diagrama de dispersión de modelos alternativos, en donde se presentan los diagramas de dispersión típicos del proceso de prueba de cada tipo de modelo neuronal alternativo bajo las dos arquitecturas propuestas.

En función de los gráficos comparativos de dispersión de los nuevos modelos que presentan diferentes cantidades de bloques secuenciales de codificador, se presentan diagramas de dispersión con una distribución común de nube de puntos y similares zonas de concentración de valores predichos, lo que indica que la cantidad de parámetros que conforma el codificador no representa una gran ventaja para estimar el valor numérico de la vida útil remanente a predecir.

De mismo modo que se objetó anteriormente en el análisis de los modelos compuesto únicamente por 8 bloques codificadores en su estructura, dentro de la categoría de un mismo tipo de modelo neuronal ambas arquitecturas presentan gráficos de dispersión análogos, independiente de cantidad de bloques que conforma su codificador.

### **7.4.3 Diagrama de dispersión ordenado valor predicho vs valor real:**

De acuerdo con sintetizar el contenido de la actual sección de “Resultado y Discusión”, se ha decidido presentar los gráficos de dispersión de los modelos alternativos en la subsección de Anexo I: Diagrama de dispersión ordenado de modelos alternativos, en donde se presentan los diagramas de dispersión ordenados usuales del proceso de prueba de cada tipo de modelo neuronal alternativo bajo las dos arquitecturas propuestas.

A partir de los gráficos de dispersión ordenados que surgen del entrenamiento de los modelos variantes, se corrobora que dentro de una misma categoría de tipo de modelo neuronal se presenta un diagrama de dispersión ordenado común independiente de la arquitectura que se observe, más aún esta similitud en los gráficos se repite dentro de los nuevos modelos sujetos a diferentes cantidades de bloques codificadores en su estructura, por otra parte también se observa nuevamente de los gráficos de dispersión ordenados que todos los modelos entrenados presentan la desventaja de presentar mayores problemas para estimar los valores más altos del valores del RUL de las muestras, generado por una menor cantidad de muestras representativa de estos valores.

#### 7.4.4 Indicadores de rendimiento por variantes de modelo:

En el siguiente apartado se exponen las tablas de los indicadores de desempeño promedio de las 10 iteraciones de aprendizaje automático ejecutado sobre cada modelo neuronal alternativo bajo estudio, presentando los resultados de forma separada según la arquitectura a la que se encuentra sujeta la estructura base con la que se construyeron los modelos de tipo Transformador, LSTM, GRU y RNN, exponiendo los indicadores de rendimiento de error cuadrático medio, error absoluto medio y el coeficiente de determinación, junto al periodo de tiempo de ejecución de entrenamiento y la cantidad neta de parámetros que componen cada estructura. Para más detalle sobre el registro de estos valores dentro de las 10 iteraciones ejecutadas diríjase a la subsección Anexo J: Registro de resultados de modelos alternativos en regresión.

##### 7.4.4.1 Arquitectura “Attention is all you need”:

###### a) Modelo neuronal alternativo de dos codificadores

*Tabla 20: Indicadores de rendimiento en regresión de modelos con 2 codificadores bajo arquitectura “Attention is all you need”*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.350.596	3.407.556	3.359.172	3.357.764
Tiempo [h]	0,85	1,03	0,96	0,87
RMSE[.]	0,085	0,071	0,072	0,074
R2 [.]	0,87	0,91	0,91	0,91
MAE [.]	0,060	0,048	0,050	0,049

###### b) Modelo neuronal alternativo de cuatro codificadores

*Tabla 21: Indicadores de rendimiento en regresión de modelos con 4 codificadores bajo arquitectura “Attention is all you need”*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.550.148	3.664.068	3.567.300	3.564.484
Tiempo [h]	1,3	1,6	1,4	1,4
RMSE[.]	0,09	0,07	0,08	0,07
R2 [.]	0,85	0,92	0,90	0,92
MAE [.]	0,064	0,047	0,052	0,048

**c) Modelo neuronal alternativo de seis codificadores**

*Tabla 22: Indicadores de rendimiento en regresión de modelos con 6 codificadores bajo arquitectura "Attention is all you need"*

<b>Indicador de rendimiento</b>	<b>Transformador</b>	<b>LSTM</b>	<b>GRU</b>	<b>RNN</b>
<b>Parámetros [.]</b>	3.749.700	3.920.580	3.775.428	3.771.204
<b>Tiempo [h]</b>	1,63	2,07	1,94	1,71
<b>RMSE[.]</b>	0,095	0,068	0,073	0,071
<b>R2 [.]</b>	0,832	0,927	0,905	0,918
<b>MAE [.]</b>	0,067	0,045	0,051	0,047

**d) Modelo neuronal alternativo de ocho codificadores**

*Tabla 23: Indicadores de rendimiento en regresión de modelos con 8 codificadores bajo arquitectura "Attention is all you need"*

<b>Indicador de rendimiento</b>	<b>Transformador</b>	<b>LSTM</b>	<b>GRU</b>	<b>RNN</b>
<b>Parámetros [.]</b>	3.949.252	4.177.092	3.983.556	3.977.924
<b>Tiempo [h]</b>	2,17	2,76	2,47	2,25
<b>RMSE[.]</b>	0,15	0,10	0,11	0,12
<b>R2 [.]</b>	0,54	0,83	0,74	0,74
<b>MAE [.]</b>	0,10	0,06	0,07	0,08

**e) Modelo neuronal alternativo de diez codificadores**

*Tabla 24: Indicadores de rendimiento en regresión de modelos con 10 codificadores bajo arquitectura "Attention is all you need"*

<b>Indicador de rendimiento</b>	<b>Transformador</b>	<b>LSTM</b>	<b>GRU</b>	<b>RNN</b>
<b>Parámetros [.]</b>	4.148.804	4.433.604	4.191.684	4.184.644
<b>Tiempo [h]</b>	2,4	3,3	3,0	2,6
<b>RMSE[.]</b>	0,104	0,072	0,082	0,081
<b>R2 [.]</b>	0,81	0,91	0,88	0,89
<b>MAE [.]</b>	0,071	0,048	0,058	0,055

#### 7.4.4.2 Arquitectura “An image is worth 16x16 words”:

##### a) Modelo neuronal alternativo de dos codificadores

*Tabla 25: Indicadores de rendimiento en regresión de modelos con 2 codificadores bajo arquitectura “An image is worth 16x16 words”.*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.350.596	3.407.556	3.359.172	3.357.764
Tiempo [h]	0,88	0,97	0,90	0,92
RMSE[.]	0,089	0,074	0,078	0,073
R2 [.]	0,87	0,91	0,90	0,90
MAE [.]	0,062	0,050	0,053	0,050

##### b) Modelo neuronal alternativo de cuatro codificadores

*Tabla 26: Indicadores de rendimiento en regresión de modelos con 4 codificadores bajo arquitectura “An image is worth 16x16 words”.*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.550.148	3.664.068	3.567.300	3.564.484
Tiempo [h]	1,2	1,6	1,4	1,3
RMSE[.]	0,09	0,07	0,08	0,07
R2 [.]	0,85	0,92	0,90	0,91
MAE [.]	0,062	0,046	0,052	0,049

##### c) Modelo neuronal alternativo de seis codificadores

*Tabla 27: Indicadores de rendimiento en regresión de modelos con 6 codificadores bajo arquitectura “An image is worth 16x16 words”.*

Indicador de rendimiento	Transformador	LSTM	GRU	RNN
Parámetros [.]	3.749.700	3.920.580	3.775.428	3.771.204
Tiempo [h]	1,67	2,19	1,97	1,76
RMSE[.]	0,090	0,070	0,073	0,072
R2 [.]	0,854	0,921	0,911	0,827
MAE [.]	0,063	0,047	0,049	0,049

**d) Modelo neuronal alternativo de ocho codificadores**

*Tabla 28: Indicadores de rendimiento en regresión de modelos con 8 codificadores bajo arquitectura “An image is worth 16x16 words”.*

<b>Indicador de rendimiento</b>	<b>Transformador</b>	<b>LSTM</b>	<b>GRU</b>	<b>RNN</b>
<b>Parámetros [.]</b>	3.949.252	4.177.092	3.983.556	3.977.924
<b>Tiempo [h]</b>	2,06	2,69	2,38	2,09
<b>RMSE[.]</b>	0,09	0,07	0,07	0,07
<b>R2 [.]</b>	0,85	0,91	0,91	0,91
<b>MAE [.]</b>	0,06	0,05	0,05	0,05

**e) Modelo neuronal alternativo de diez codificadores**

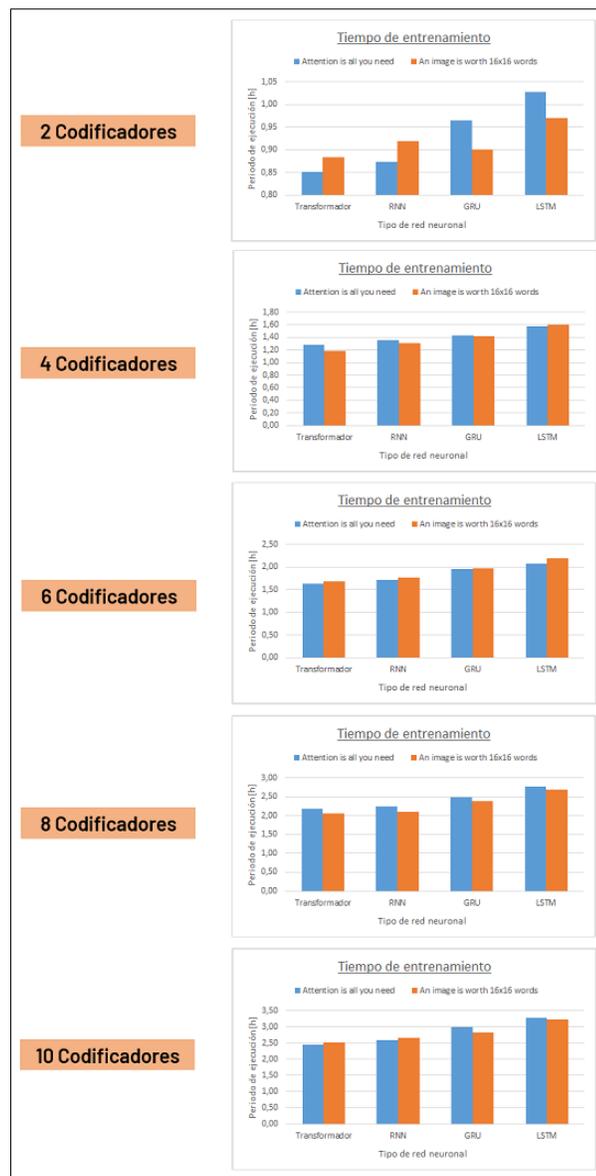
*Tabla 29: Indicadores de rendimiento en regresión de modelos con 10 codificadores bajo arquitectura “An image is worth 16x16 words”.*

<b>Indicador de rendimiento</b>	<b>Transformador</b>	<b>LSTM</b>	<b>GRU</b>	<b>RNN</b>
<b>Parámetros [.]</b>	4.148.804	4.433.604	4.191.684	4.184.644
<b>Tiempo [h]</b>	2,52	3,22	2,83	2,66
<b>RMSE[.]</b>	0,09	0,07	0,07	0,07
<b>R2 [.]</b>	0,85	0,91	0,91	0,91
<b>MAE [.]</b>	0,06	0,05	0,05	0,05

### 7.4.5 Gráficos comparativos de rendimiento

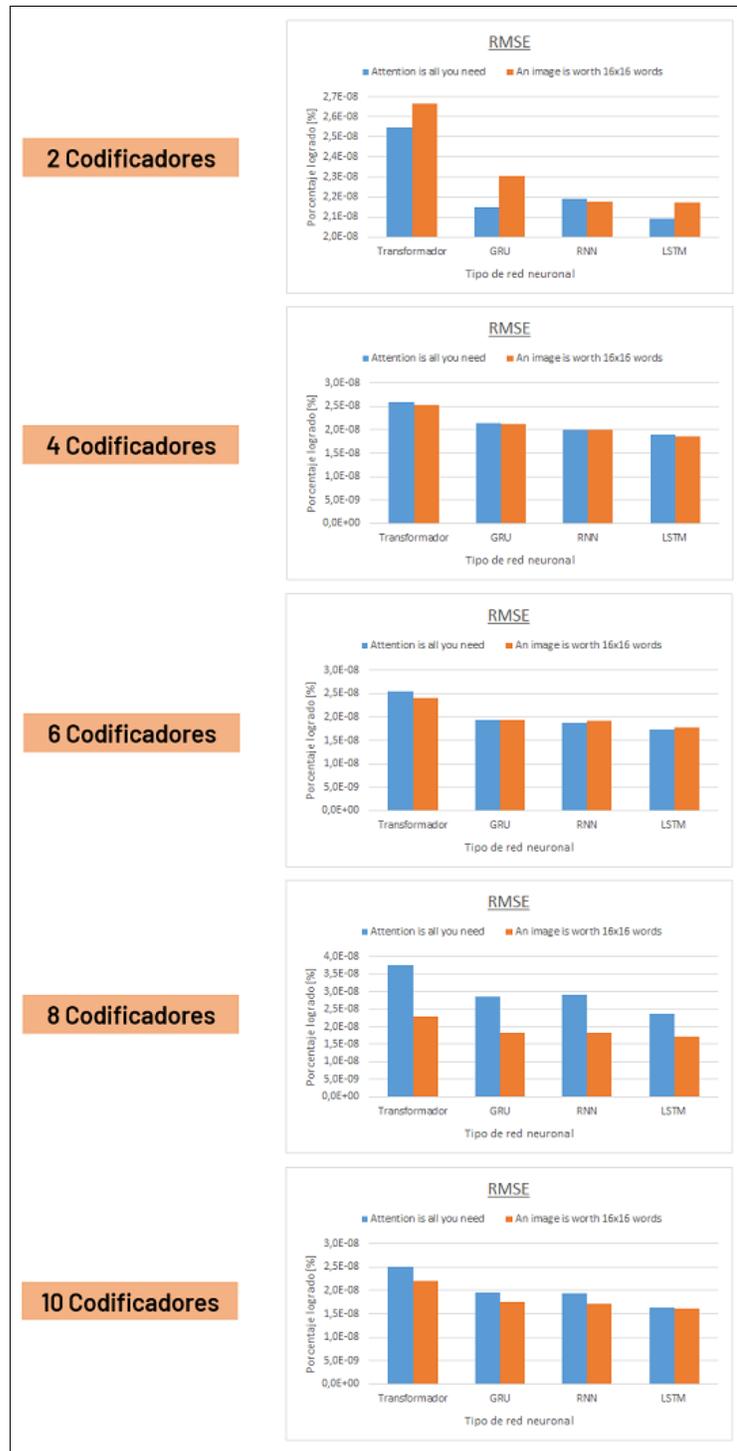
A partir del contenido de las tablas del punto anterior se elaboran los siguientes gráficos de rendimiento con el fin de facilitar la evaluación comparativa de los resultados de los diferentes tipos de modelos neuronales alternativos, bajo distintas arquitecturas y conformados por un variado número de parámetros, además para quitar la ventaja de mayor capacidad de procesamiento de características que supone una mayor cantidad de parámetros que conforma la capa característica de cada tipo de modelo, es que se ha decidido presentar el desempeño de cada tipo de modelo neuronal a razón del valor del indicador de rendimiento dividido por la cantidad de parámetros neto del modelo.

#### **Tiempo de ejecución de entrenamiento de modelos alternativos**



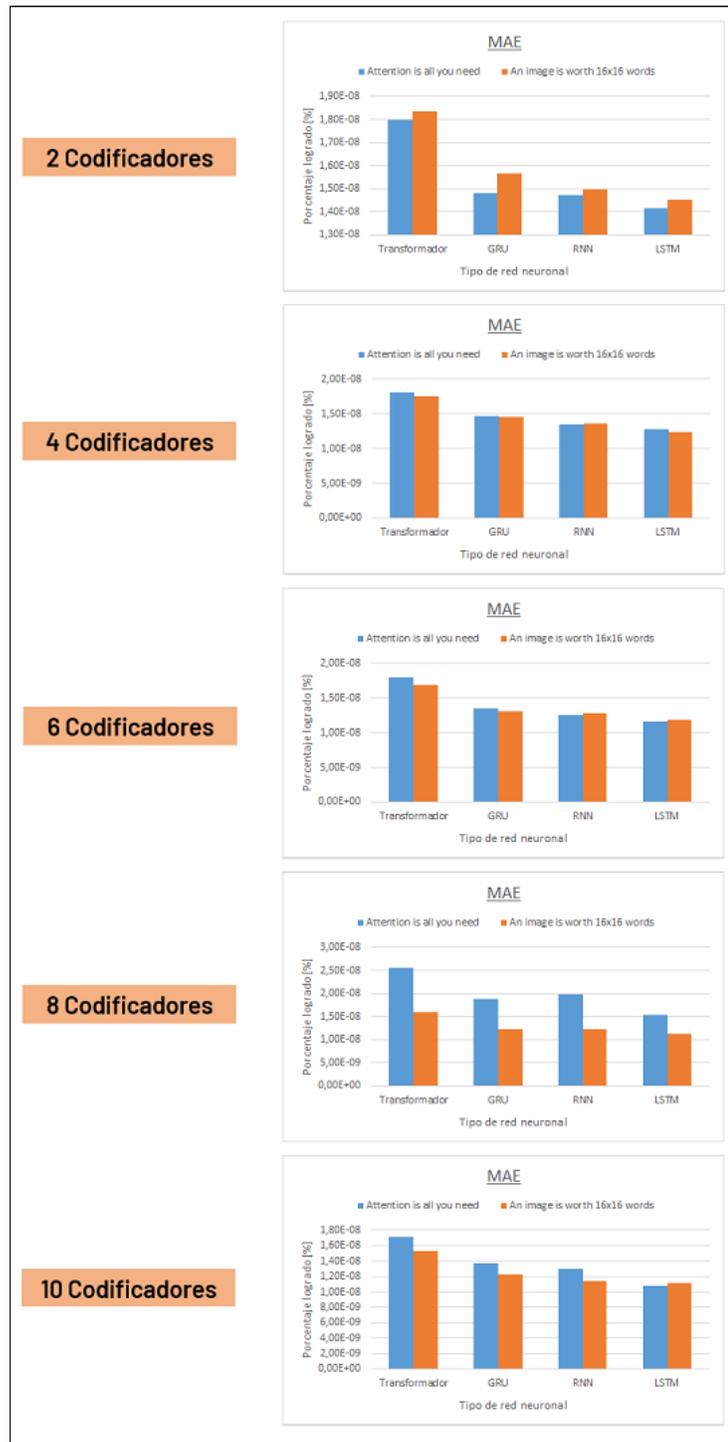
**Figura 45:** Gráfico comparativo de tiempo de entrenamiento según tipo de modelo, arquitectura y número de codificadores que los conforman.

**Indicador de rendimiento de error cuadrático medio de modelos alternativos**



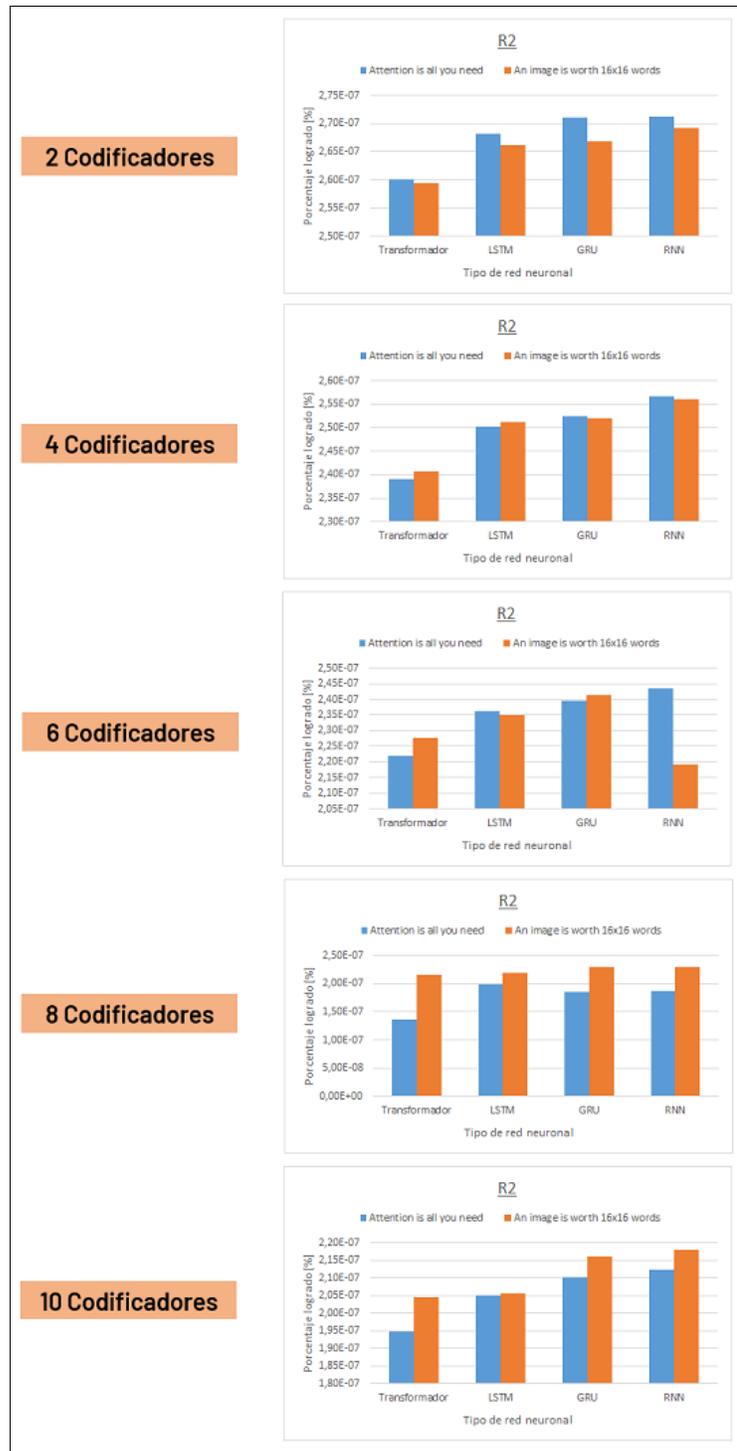
**Figura 46:** Gráfico comparativo del indicador de rendimiento error cuadrático medio según tipo de modelo, arquitectura y número de codificadores que los conforman.

**Indicador de rendimiento de error absoluto medio de modelos alternativos**



**Figura 47:** Gráfico comparativo del indicador de rendimiento error absoluto medio según tipo de modelo, arquitectura y número de codificadores que los conforman.

**Indicador de rendimiento de coeficiente de determinación de modelos alternativos**



**Figura 48:** Gráfico comparativo del indicador de rendimiento coeficiente de determinación según tipo de modelo, arquitectura y número de codificadores que los conforman.

En función de los gráficos comparativos de los indicadores de rendimiento de tiempo de ejecución de entrenamiento, error cuadrático medio, error absoluto medio y el coeficiente de determinación según el número de bloques codificadores que conforma su estructura, en general para cada indicador de rendimiento se expone el mismo orden de desempeño por tipo de modelo que se describió anteriormente para el caso particular de los modelos de regresión compuestos únicamente por 8 bloques de codificadores en su estructura, esto quiere decir que las diferencias de desempeño entre tipos de modelos neuronales diferentes son independiente de la cantidad de bloques que conforma la estructura de su codificador.

Por otro lado, si se observan los valores de los indicadores de rendimiento obtenidos por cada tipo de modelo neuronal bajo las dos arquitecturas propuestas, se corroborará que por lo general los modelos bajo la arquitectura “An image worth 16x16 words” poseen un mayor desempeño para estimar las horas de vida útil remanente en comparación a los modelos bajo la arquitectura “Attention is all you need”, aunque cabe destacar que para los modelos constituidos por menores parámetros en su estructura, por ejemplo, los modelos que están conformados solo por 2 bloques de codificadores, esta relación de desempeño se invierte, ya que la arquitectura “An image worth 16x16 words” al contar con tan bajo número de parámetros no es capaz de extraer de una misma instancia de variables de la ventana temporal las características suficientemente importantes como para rivalizar con la utilidad que representan las características que elabora la arquitectura “Attention is all you need” a partir de la evolución individual de los parámetros operacionales en la ventana temporal de entrada. Cabe destacar también que la variabilidad en los resultados que exponen los gráficos comparativos se adjudica principalmente al limitado número de iteraciones de aprendizaje evaluados.

## 8 Conclusiones

Sobre la base del análisis de resultados realizados en torno al desempeño de los modelos de tipo Transformador, LSTM, GRU y RNN diseñados bajo las dos arquitecturas propuestas “Attention is all you need” y “An image worth 16x16 words”, para posteriormente ser sometidos a un análisis de sensibilidad aplicado sobre el número de parámetros que componen su estructura, se concluye que; la arquitectura que destaca por su mayor rendimiento independiente de si se trata de la clasificación del estado operativo o de la regresión de la vida útil remanente del equipo térmico, este corresponde a la arquitectura “An image worth 16x16 words”, principalmente porque este tipo de arquitectura se centra más en la extracción de características a partir de variables de operación relacionadas a un mismo instante de tiempo de la ventana temporal, mientras la arquitectura “Attention all you need” se centra más en la extracción de características a partir de la evolución individual de cada variable de operación dentro de la ventana temporal.

Por otro lado, se obtuvo que dentro de los tipos de modelos neuronales de serie temporales bajo estudio (Transformador, LSTM, GRU y RNN), aquel tipo de modelo que presenta el mejor desempeño para el caso de clasificación bajo una curva de aprendizaje automático regido por una función de pérdida centrada en maximizar el indicador rendimiento de “Exactitud” y que muestra el mejor desempeño sobre su indicador de rendimiento más representativo de la calidad del modelo “área bajo la curva ROC”, corresponde al modelo neuronal de tipo Transformador, debido a que su celda distintiva de su capa característica le permite aplicar un filtro de información que ayuda a la detección de patrones distintivos de cada estado operativo, en segundo lugar se encuentra el modelo de tipo GRU, gracias a que su capa característica le permite extraer características complejas considerando el panorama global de las ventanas temporales sin perder mayor información de las instancias iniciales por el desvanecimiento de gradiente, en tercer lugar se encuentra el modelo LSTM, a causa de que excesiva complejidad en su celda distintiva le dificulta extraer características útiles para identificar estados, y finalmente el modelo con peor rendimiento es de tipo RNN, ya que lo afecta muy gravemente el desvanecimiento de gradiente, perdiendo grandes cantidades de datos de las muestras temporales.

, en tanto que el tipo de modelo que presenta el mejor desempeño para el caso de regresión bajo una curva de aprendizaje automático regido por una función de pérdida centrada en minimizar el indicador de “Error cuadrático medio” y que muestra el mejor desempeño sobre su indicador de rendimiento más representativo de la calidad del modelo “coeficiente de determinación”, corresponde al modelo neuronal de tipo RNN, gracias a que es afectado fuertemente por el desvanecimiento de gradiente le permite concentrar para extraer características en las últimas instancias de las ventanas temporales de entrada que resultan las más relacionadas al valor del RUL a predecir, en segundo lugar se encuentra el modelo GRU, debido a que su celda distintiva presenta una estructura de funcionamiento que reduce el efecto del desvanecimiento de gradiente a un nivel intermedio entre el modelo LSTM y el modelo RNN, luego en tercer lugar están los modelos de tipo LSTM, ya que a pesar de poseer una mejor memoria de corto plazo y memoria de largo plazo la complejidad de su celda distintiva no le permite mejorar rápidamente su desempeño, requiriendo un mayor número de épocas para obtener mejores resultados. Por último, el modelo con el peor desempeño corresponde al modelo Transformador, pues en su capa característica se dedica exclusivamente a aplicar el filtro de datos de entrada sin extraer nuevas características, perdiendo capacidad de extracción de características complejas útiles para la estimación del valor del RUL.

Mientras el análisis de sensibilidad aplicado sobre la cantidad de bloques secuenciales discretos que conforma el codificador corrobora que las diferencias de rendimiento y orden de desempeño entre distintos tipos de modelos neuronales o entre las distintas arquitecturas propuestas bajo las que se encuentran sujetas los modelos, son independientes de la cantidad de parámetros que conforma la estructura característica de funcionamiento de cada modelo.

## Conclusiones

---

Cabe destacar que en general los modelos entrenados demostraron un gran desempeño en las actividades de clasificación y regresión sujetas al caso de estudio, entregando valores de eficiencia de predicción que superan un 70% de lo estipulado inicialmente y tampoco presentaron problemas durante la fase de entrenamiento de estos modelos, demostrando siempre un funcionamiento sólido del código del programa que no exhibió errores de ejecución graves.

## Bibliografía

- [1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., & Gomez, A. et al. (2017). Attention Is All You Need. Retrieved 15 June 2021, from <https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>
- [2] Ntakouris, T. (2021). The Time Series Transformer. Retrieved 15 June 2021, from <https://towardsdatascience.com/the-time-series-transformer-2a521a0efad3>
- [3] Ruiz-Tagle Palazuelos, A. A. (2019). System-level prognostics and health management: a graph neural network-based framework. Retrieved 13 May 2022, from [https://repositorio.uchile.cl/bitstream/handle/2250/173988/cf-ruiz-tagle\\_ap.pdf?sequence=4&isAllowed=y](https://repositorio.uchile.cl/bitstream/handle/2250/173988/cf-ruiz-tagle_ap.pdf?sequence=4&isAllowed=y)
- [4] Wu, N., Green, B., Ben, X., & O'Banion, S. (2021). Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case. Retrieved 15 June 2021, from <https://arxiv.org/pdf/2001.08317.pdf>
- [5] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. Retrieved 13 May 2022, from <https://openreview.net/pdf?id=YicbFdNTTy&amp=1>
- [6] Mohamed, H., Negm, A., Zahran, M., & Saavedra, O. C. (2015, March). Assessment of artificial neural network for bathymetry estimation using High Resolution Satellite imagery in Shallow Lakes: case study El Burullus Lake. In International water technology conference (pp. 12-14). Retrieved 13 May 2022, from [https://www.academia.edu/29681500/Assessment\\_of\\_Artificial\\_Neural\\_Network\\_for\\_bathymetry\\_estimation\\_using\\_High\\_Resolution\\_Satellite\\_imagery\\_in\\_Shallow\\_Lakes\\_Case\\_Study\\_El\\_Burullus\\_Lake](https://www.academia.edu/29681500/Assessment_of_Artificial_Neural_Network_for_bathymetry_estimation_using_High_Resolution_Satellite_imagery_in_Shallow_Lakes_Case_Study_El_Burullus_Lake)
- [7] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780. Retrieved 13 May 2022, from <http://www.bioinf.jku.at/publications/older/2604.pdf>
- [8] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2015, June). Gated feedback recurrent neural networks. In International conference on machine learning (pp. 2067-2075). PMLR. Retrieved 15 June 2021, from <https://arxiv.org/pdf/1502.02367.pdf>
- [9] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation. California Univ San Diego La Jolla Inst for Cognitive Science. Retrieved 13 May 2022, from <http://concepts.psych.wisc.edu/papers/711/RumelhartBackprop.pdf>

## Anexos

### Anexo A: Registro de resultados de modelos de clasificación por iteración de aprendizaje

#### Anexo A.1: Arquitectura “Attention is all you need”

*Tabla 30: Datos de desempeño logrado en clasificación de cada modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por ocho módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
Tiempo de entrenamiento [s]	Intento 1	3763	5057	4577	4404
	Intento 2	4546	5118	4339	4658
	Intento 3	3943	5298	4519	4595
	Intento 4	4679	5359	4818	4413
	Intento 5	4267	5430	4879	4360
	Intento 6	3883	5298	4940	4475
	Intento 7	4184	5420	4884	4413
	Intento 8	4065	5605	5004	4882
	Intento 9	4123	4939	4706	3898
	Intento 10	3762	5431	4704	4312
Exactitud [%]	Intento 1	98,83	98,33	98,53	97,30
	Intento 2	98,40	98,70	98,58	96,79
	Intento 3	98,50	98,28	98,55	97,06
	Intento 4	98,69	98,45	98,37	96,41
	Intento 5	98,56	98,31	98,31	96,87
	Intento 6	98,07	98,47	98,04	96,75
	Intento 7	98,36	98,63	98,44	97,05
	Intento 8	98,53	98,59	98,45	96,82
	Intento 9	98,45	98,58	98,26	97,25
	Intento 10	98,37	98,37	98,53	96,98
Precisión [%]	Intento 1	96,03	95,14	95,25	90,69
	Intento 2	95,35	93,98	90,09	85,59
	Intento 3	98,49	90,97	96,91	97,10
	Intento 4	97,81	90,22	88,55	97,04
	Intento 5	96,41	90,73	92,97	96,38
	Intento 6	90,23	91,82	88,05	92,41
	Intento 7	91,72	95,46	91,65	95,66
	Intento 8	95,25	91,00	92,08	90,62

	Intento 9	96,36	91,91	88,79	97,17
	Intento 10	95,17	91,19	96,40	93,71
<b>Exhaustividad [%]</b>	Intento 1	91,61	87,01	89,14	80,10
	Intento 2	87,66	92,43	95,72	80,10
	Intento 3	85,69	91,12	87,66	71,55
	Intento 4	88,32	94,08	95,39	64,64
	Intento 5	88,32	91,78	89,14	70,07
	Intento 6	89,64	92,27	92,11	72,04
	Intento 7	91,12	89,97	92,11	72,53
	Intento 8	89,14	94,74	91,78	74,67
	Intento 9	87,17	93,42	93,75	73,52
	Intento 10	87,50	91,94	87,99	73,52
	<b>Puntaje valor F1 [%]</b>	Intento 1	93,77	90,89	92,10
Intento 2		91,35	93,20	92,82	82,75
Intento 3		91,64	91,04	92,06	82,39
Intento 4		92,83	92,11	91,84	77,59
Intento 5		92,19	91,25	91,02	81,14
Intento 6		89,93	92,04	90,03	80,96
Intento 7		91,42	92,63	91,88	82,51
Intento 8		92,10	92,83	91,93	81,88
Intento 9		91,54	92,66	91,20	83,71
Intento 10		91,17	91,56	92,00	82,40
<b>Área bajo la curva características [%]</b>	Intento 1	95,60	93,27	94,34	89,61
	Intento 2	93,61	95,90	97,30	89,33
	Intento 3	92,78	95,08	93,68	85,66
	Intento 4	94,06	96,50	97,04	82,21
	Intento 5	93,99	95,39	94,21	84,89
	Intento 6	94,30	95,70	95,39	85,71
	Intento 7	95,12	94,76	95,61	86,09
	Intento 8	94,34	96,87	95,47	86,92
	Intento 9	93,41	96,27	96,25	86,65
	Intento 10	93,51	95,50	93,82	86,50

## Anexo A.2: Arquitectura “An image is worth 16x16 words”:

**Tabla 31:** Datos de desempeño logrado en clasificación de cada modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por ocho módulos de codificadores.

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
<b>Tiempo de entrenamiento[s]</b>	Intento 1	4244	5269	4814	4352
	Intento 2	3880	4817	4757	3922
	Intento 3	4065	5430	5060	4300
	Intento 4	3824	4751	4818	4353
	Intento 5	3862	4814	4881	3979
	Intento 6	4124	5252	5001	4599
	Intento 7	3945	5155	5002	4351
	Intento 8	3881	5298	4939	4600
	Intento 9	3704	5484	4884	4412
	Intento 10	4066	5551	4941	4103
<b>Exactitud [%]</b>	Intento 1	98,64	98,14	98,48	97,84
	Intento 2	98,58	98,29	98,52	98,09
	Intento 3	98,55	98,26	98,64	98,07
	Intento 4	98,66	98,45	98,55	98,12
	Intento 5	98,47	98,44	98,42	98,14
	Intento 6	98,55	98,37	98,67	97,91
	Intento 7	98,50	98,09	98,74	98,12
	Intento 8	98,77	98,55	98,58	98,06
	Intento 9	98,21	98,31	98,66	98,03
	Intento 10	98,40	98,50	98,42	98,20
<b>Precisión [%]</b>	Intento 1	96,28	88,52	91,16	86,40
	Intento 2	94,35	88,70	98,31	90,38
	Intento 3	94,64	88,31	93,21	93,09
	Intento 4	96,61	89,97	93,73	89,89
	Intento 5	93,23	92,35	92,76	93,29
	Intento 6	98,68	90,92	93,96	89,53
	Intento 7	92,26	90,12	93,56	91,65
	Intento 8	95,53	91,75	94,35	86,69
	Intento 9	93,50	91,40	91,97	89,53
	Intento 10	93,33	93,70	90,84	94,42
<b>Exhaustividad [%]</b>	Intento 1	89,31	92,60	93,26	91,94
	Intento 2	90,62	94,24	86,02	89,64
	Intento 3	89,97	94,41	92,60	86,35
	Intento 4	89,14	94,41	90,95	90,62

Anexos

	Intento 5	90,62	91,28	90,62	86,84
	Intento 6	86,02	92,27	92,11	88,65
	Intento 7	92,11	89,97	93,26	88,49
	Intento 8	91,45	93,26	90,62	94,24
	Intento 9	87,50	90,95	94,24	89,97
	Intento 10	89,80	90,46	92,93	86,35
<b>Puntaje valor F1 [%]</b>	Intento 1	92,66	90,51	92,20	89,08
	Intento 2	92,45	91,39	91,75	90,01
	Intento 3	92,24	91,26	92,90	89,59
	Intento 4	92,73	92,13	92,32	90,25
	Intento 5	91,91	91,81	91,68	89,95
	Intento 6	91,92	91,59	93,02	89,22
	Intento 7	92,18	90,04	93,41	90,04
	Intento 8	93,45	92,50	92,45	90,31
	Intento 9	90,40	91,18	93,10	89,75
	Intento 10	91,53	92,05	91,87	90,21
<b>Área bajo la curva característica [%]</b>	Intento 1	94,47	95,66	95,94	94,31
	Intento 2	96,01	96,48	92,93	94,32
	Intento 3	94,71	96,54	95,94	92,83
	Intento 4	94,41	96,64	95,15	94,77
	Intento 5	94,96	95,24	94,94	93,09
	Intento 6	92,95	95,65	95,74	93,78
	Intento 7	95,64	94,46	96,29	93,82
	Intento 8	95,50	96,18	95,02	96,35
	Intento 9	93,43	95,02	96,68	94,42
	Intento 10	94,56	94,91	95,97	92,90

## Anexo B: Registro de resultados de modelos de regresión por iteración de aprendizaje

### Anexo B.1: Arquitectura “Attention is all you need”

*Tabla 32: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need”.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
Tiempo de entrenamiento [s]	Intento 1	9163	11075	9623	9880
	Intento 2	8443	11238	9441	9100
	Intento 3	9344	11554	11491	8862
	Intento 4	9824	11020	10864	8846
	Intento 5	8341	12209	11141	10124
	Intento 6	9343	12523	10045	10296
	Intento 7	9944	12332	9351	9579
	Intento 8	9396	11318	10956	9822
	Intento 9	8383	11853	11561	9691
	Intento 10	8563	10688	7395	9640
RMSE [.]	Intento 1	0,09	0,07	0,07	0,07
	Intento 2	0,09	0,07	0,07	0,07
	Intento 3	0,09	0,07	0,08	0,07
	Intento 4	0,09	0,07	0,08	0,08
	Intento 5	0,08	0,07	0,07	0,07
	Intento 6	0,09	0,07	0,07	0,07
	Intento 7	0,13	0,07	0,07	0,07
	Intento 8	0,09	0,07	0,08	0,07
	Intento 9	0,09	0,07	0,07	0,07
	Intento 10	0,08	0,07	0,08	0,07
R2 [.]	Intento 1	0,87	0,91	0,91	0,91
	Intento 2	0,86	0,92	0,92	0,91
	Intento 3	0,84	0,91	0,90	0,91
	Intento 4	0,88	0,91	0,90	0,90
	Intento 5	0,88	0,91	0,92	0,92
	Intento 6	0,86	0,92	0,90	0,92
	Intento 7	0,66	0,91	0,91	0,91
	Intento 8	0,88	0,93	0,90	0,91
	Intento 9	0,89	0,91	0,92	0,92
	Intento 10	0,89	0,91	0,89	0,91
MAE [.]	Intento 1	0,06	0,05	0,05	0,05

## Anexos

---

	Intento 2	0,06	0,05	0,05	0,05
	Intento 3	0,06	0,06	0,06	0,05
	Intento 4	0,06	0,05	0,05	0,05
	Intento 5	0,06	0,05	0,05	0,05
	Intento 6	0,06	0,05	0,05	0,05
	Intento 7	0,09	0,05	0,05	0,05
	Intento 8	0,06	0,05	0,05	0,05
	Intento 9	0,06	0,05	0,05	0,05
	Intento 10	0,06	0,05	0,05	0,05

## Anexo B.2: Arquitectura “An image is worth 16x16 words”

*Tabla 33: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words”.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
<b>Tiempo [s]</b>	Intento 1	8733	9921	7933	7337
	Intento 2	7121	8952	9871	8008
	Intento 3	6820	11138	8051	7337
	Intento 4	7662	10048	8172	8307
	Intento 5	8143	10770	8472	7216
	Intento 6	7658	9015	8773	7093
	Intento 7	7422	8987	8113	7583
	Intento 8	8202	8092	8294	6487
	Intento 9	5013	9930	8415	7878
	Intento 10	7541	9915	9618	8068
<b>RMSE [.]</b>	Intento 1	0,09	0,07	0,07	0,07
	Intento 2	0,09	0,07	0,08	0,07
	Intento 3	0,09	0,07	0,07	0,08
	Intento 4	0,08	0,07	0,08	0,07
	Intento 5	0,10	0,07	0,08	0,08
	Intento 6	0,10	0,07	0,07	0,08
	Intento 7	0,09	0,08	0,07	0,07
	Intento 8	0,09	0,07	0,07	0,08
	Intento 9	0,10	0,07	0,07	0,07
	Intento 10	0,09	0,07	0,07	0,07
<b>R2 [.]</b>	Intento 1	0,87	0,92	0,91	0,92
	Intento 2	0,84	0,91	0,90	0,92
	Intento 3	0,87	0,91	0,92	0,89
	Intento 4	0,87	0,91	0,91	0,92
	Intento 5	0,82	0,93	0,90	0,90
	Intento 6	0,85	0,90	0,92	0,91
	Intento 7	0,86	0,899	0,91	0,92
	Intento 8	0,83	0,92	0,93	0,91
	Intento 9	0,84	0,93	0,91	0,91
	Intento 10	0,86	0,91	0,92	0,92
<b>MAE [.]</b>	Intento 1	0,06	0,05	0,05	0,05
	Intento 2	0,06	0,05	0,06	0,05
	Intento 3	0,06	0,05	0,05	0,05
	Intento 4	0,06	0,05	0,05	0,05

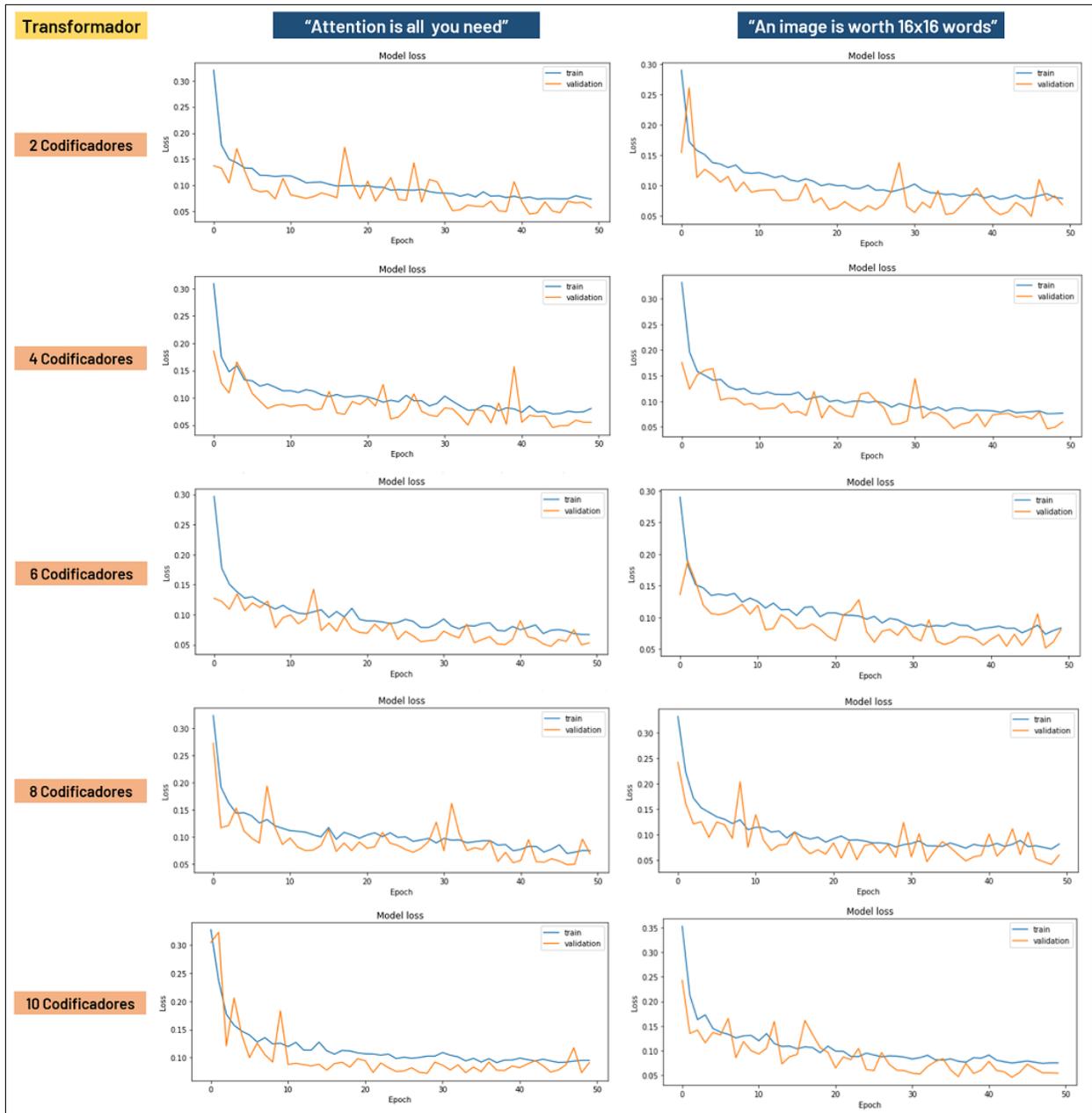
## Anexos

---

	Intento 5	0,07	0,04	0,05	0,05
	Intento 6	0,06	0,05	0,05	0,05
	Intento 7	0,06	0,05	0,05	0,05
	Intento 8	0,07	0,05	0,05	0,05
	Intento 9	0,07	0,05	0,05	0,05
	Intento 10	0,06	0,05	0,05	0,05

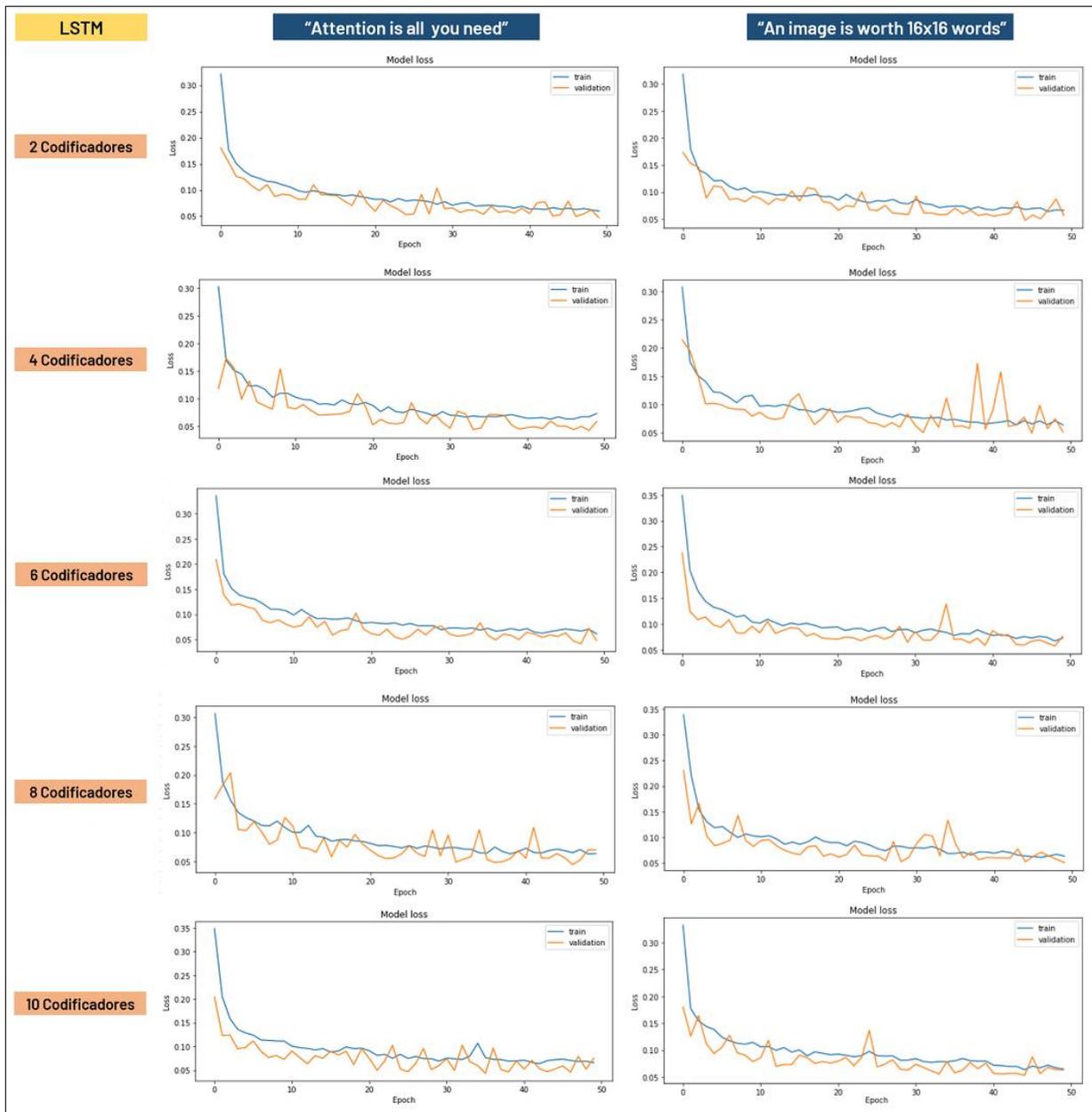
## Anexo C: Funciones de pérdida de modelos alternativos para clasificación

### Función de pérdida de modelos alternativos de tipo Transformador



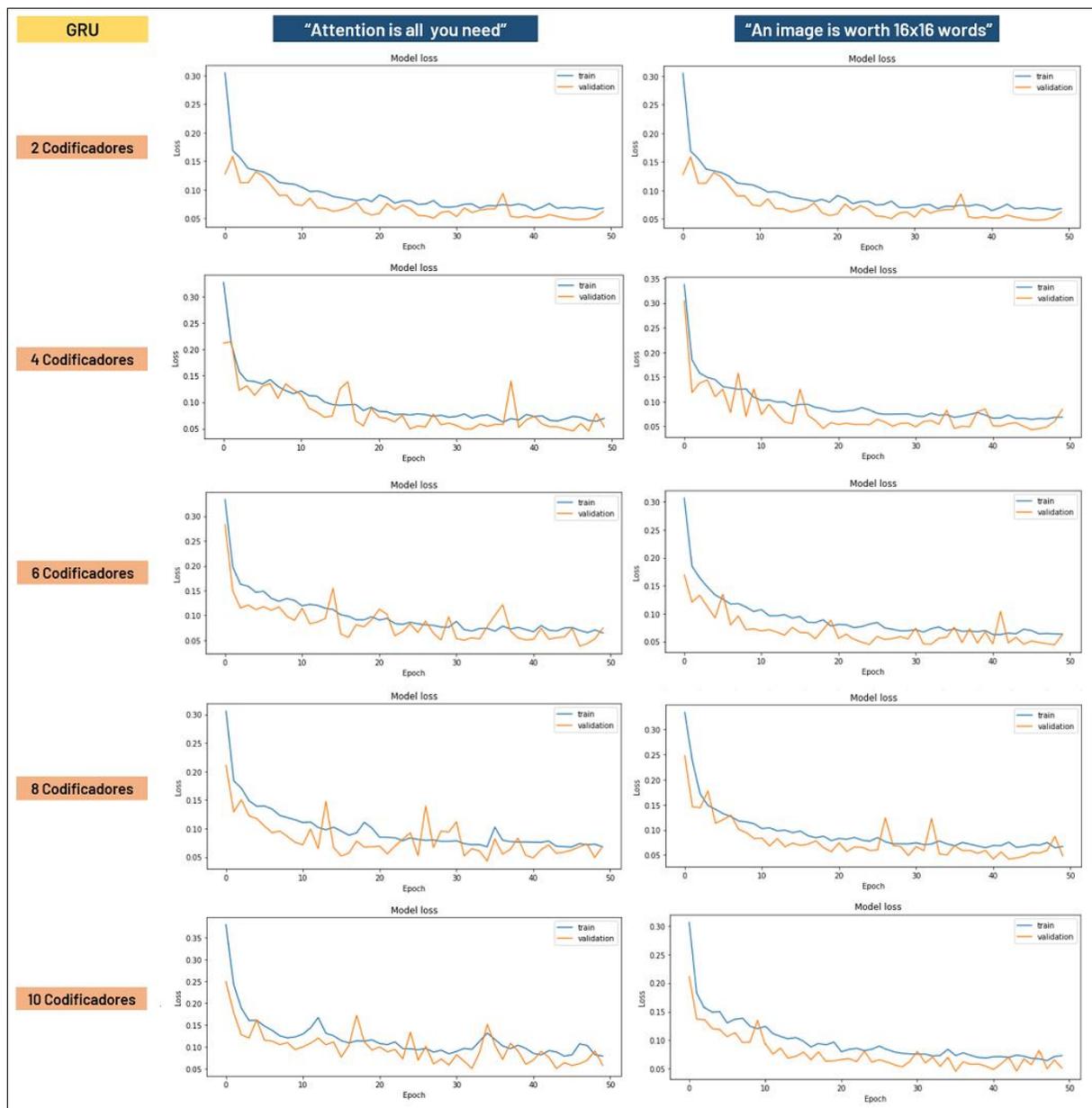
*Figura 49: Comparación de funciones de pérdida en clasificación de modelos variantes del modelo transformador.*

## Función de pérdida de modelos alternativos de tipo LSTM



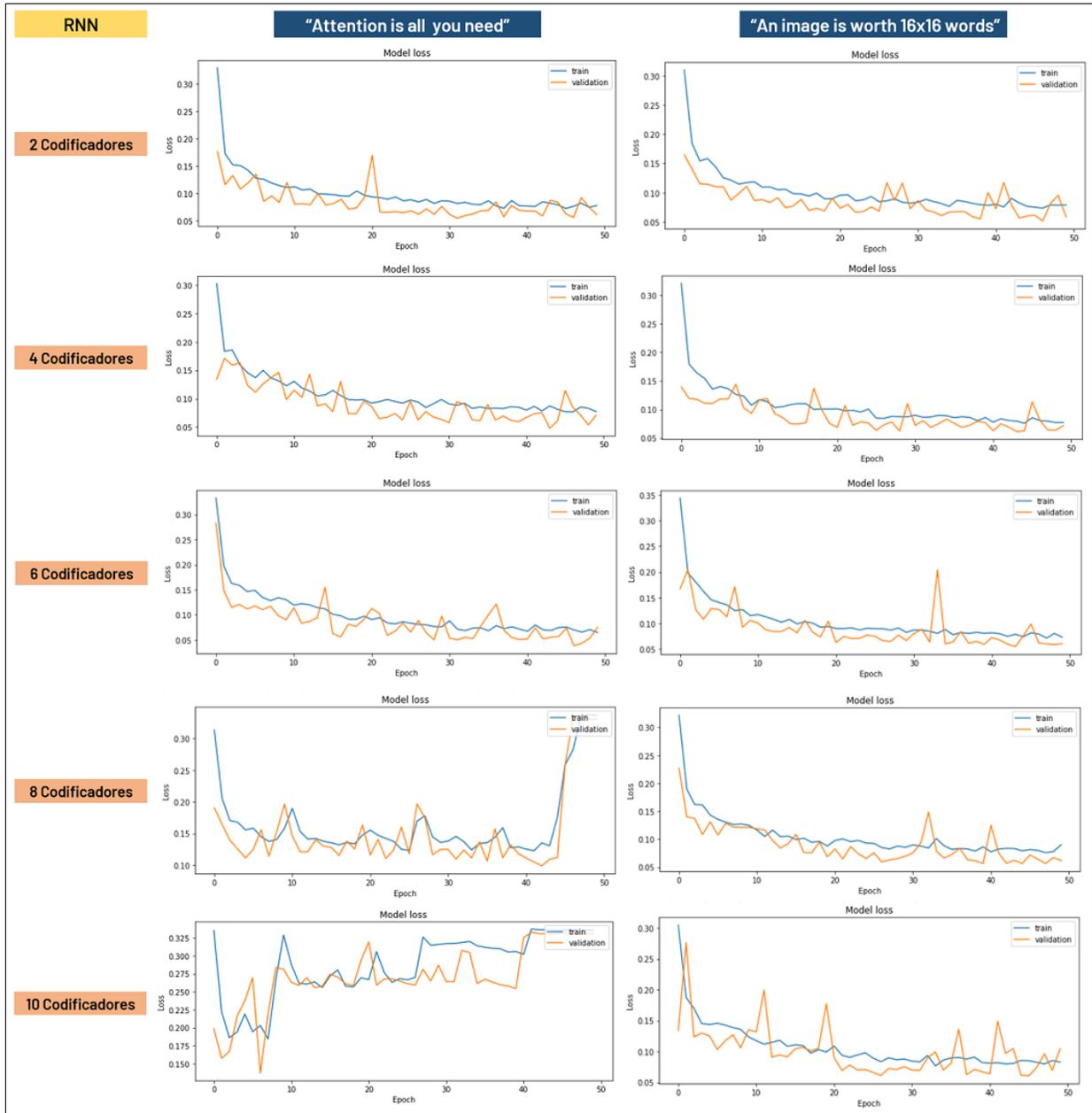
**Figura 50:** Comparación de funciones de pérdida en clasificación de modelos variantes del modelo LSTM.

## Función de pérdida de modelos alternativos de tipo GRU



*Figura 51: Comparación de funciones de pérdida en clasificación de modelos variantes del modelo GRU.*

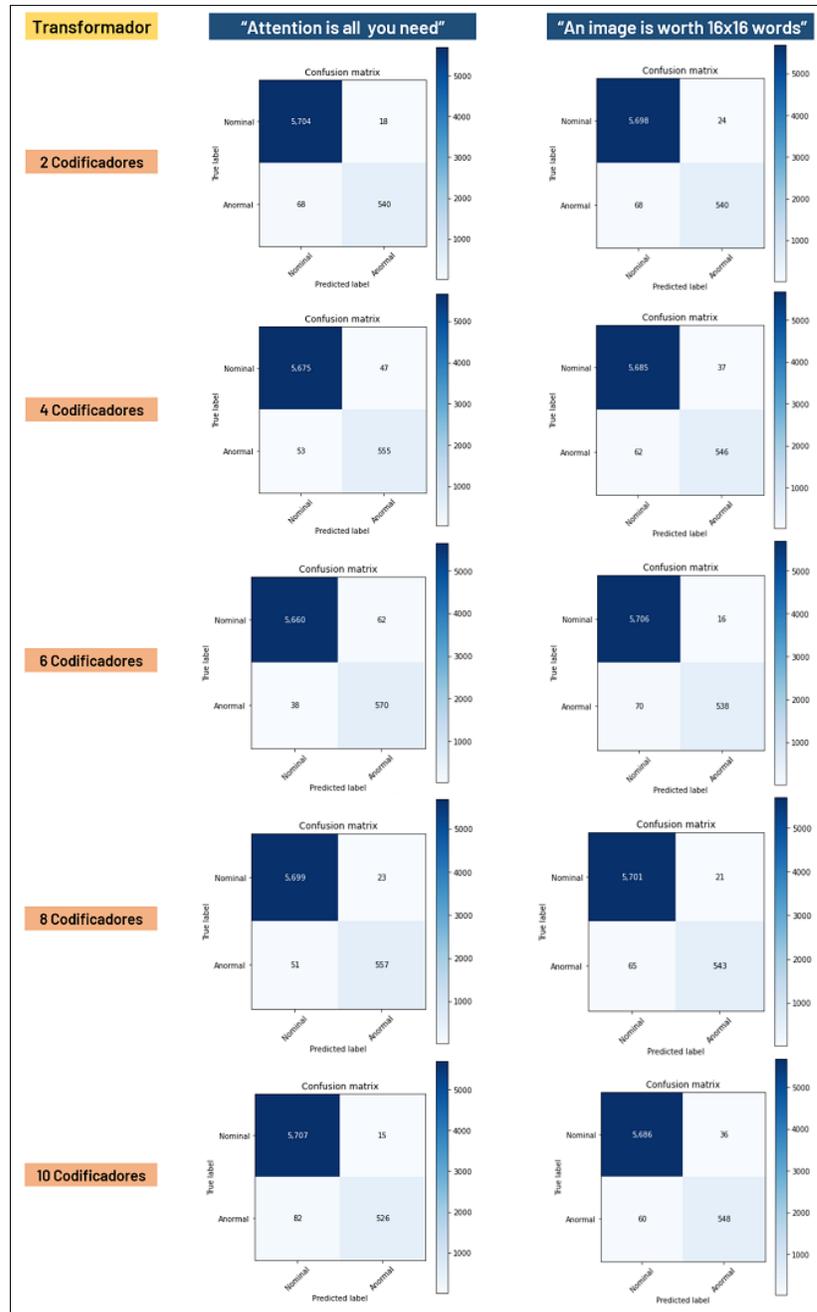
**Función de pérdida de modelos alternativos de tipo RNN**



*Figura 52: Comparación de funciones de pérdida en clasificación de modelos variantes del modelo RNN.*

## Anexo D: Matrices de confusión de modelos alternativos para clasificación

### Matriz de confusión de modelos alternativos de tipo Transformador



*Figura 53: Comparación de matrices de confusión de clasificación de variantes del modelo transformador.*

**Matriz de confusión de modelos alternativos de tipo LSTM**

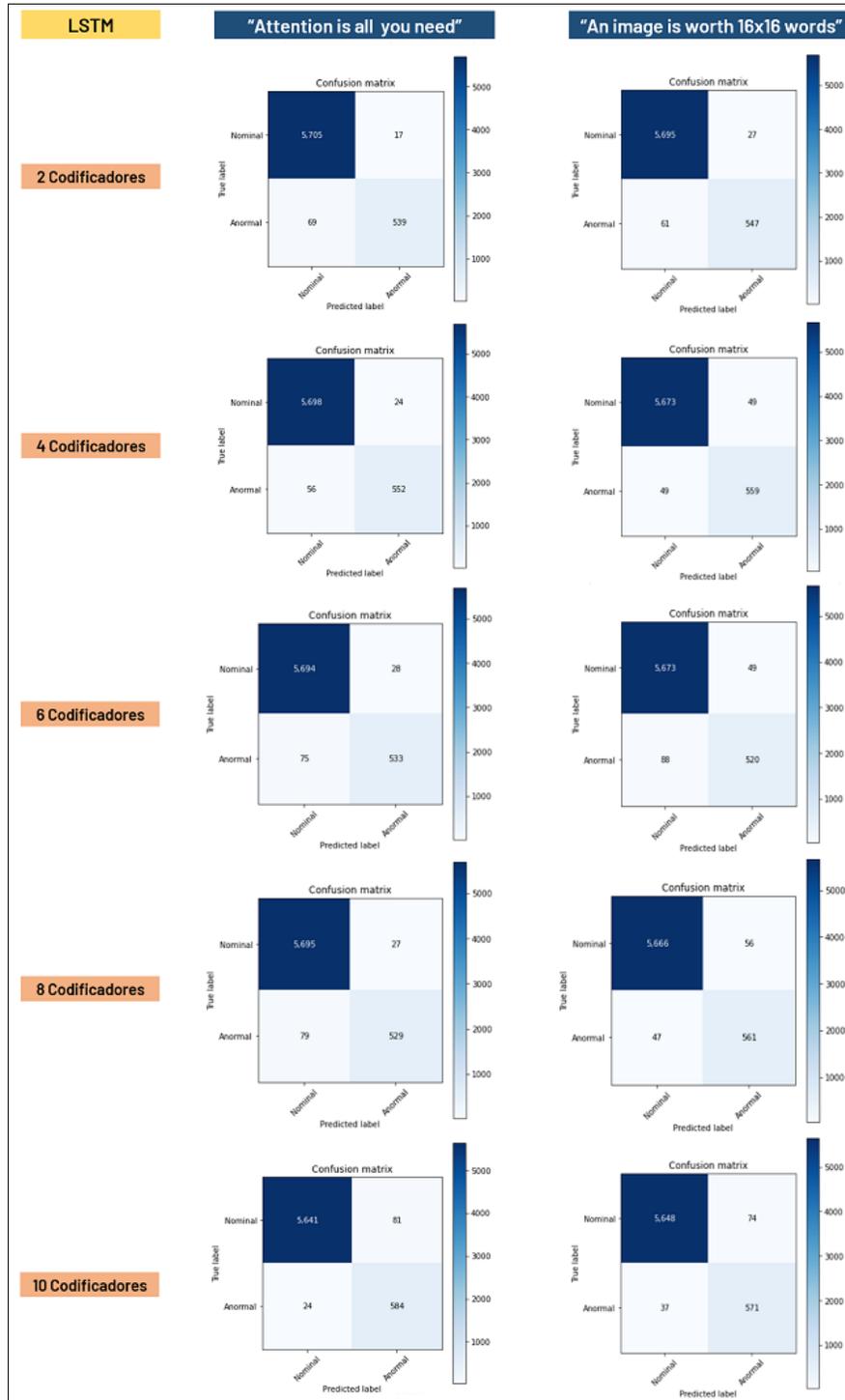


Figura 54: Comparación de matrices de confusión de clasificación de variantes del modelo LSTM.

**Matriz de confusión de modelos alternativos de tipo GRU**

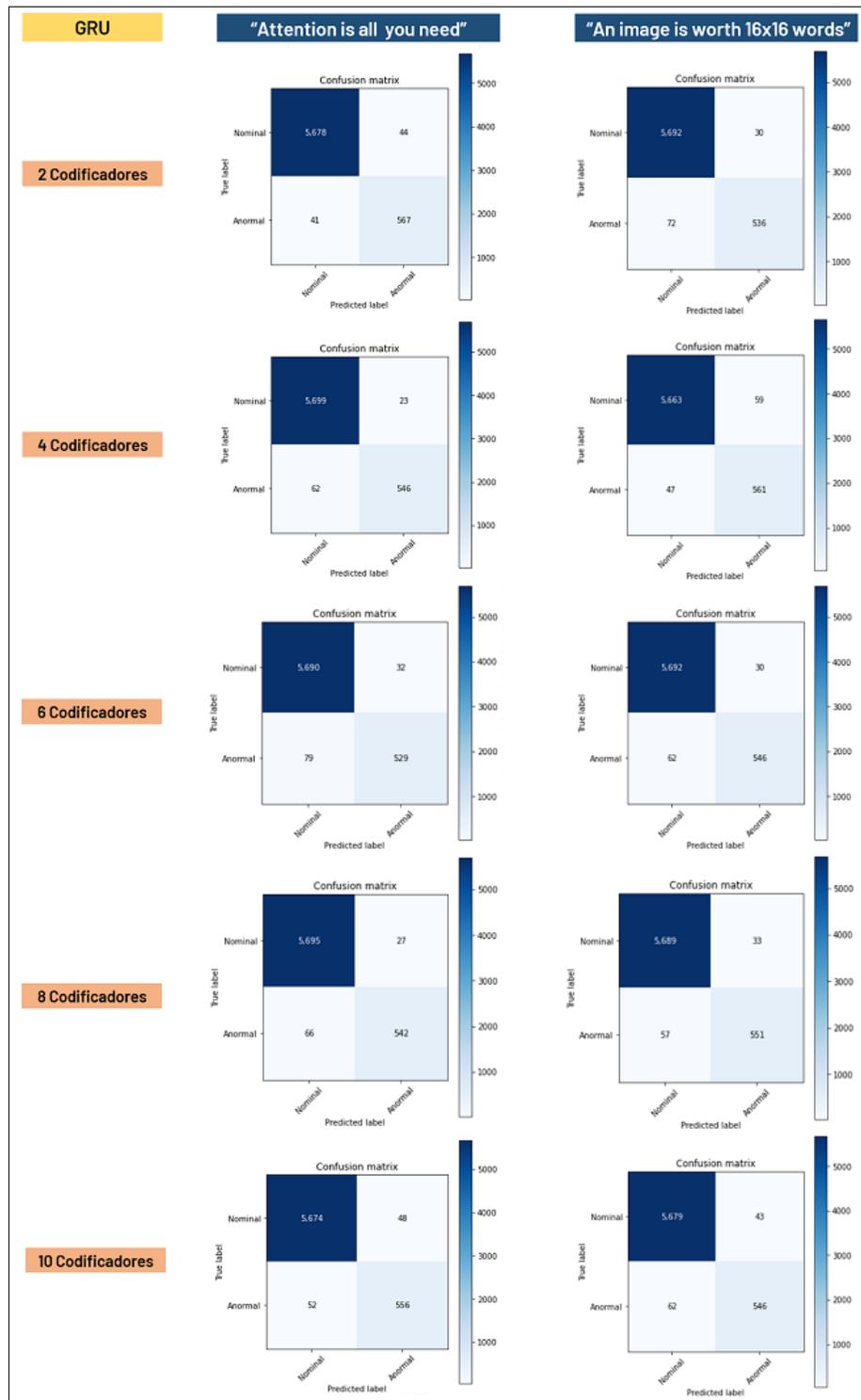


Figura 55: Comparación de matrices de confusión de clasificación de variantes del modelo GRU.

**Matriz de confusión de modelos alternativos de tipo RNN**

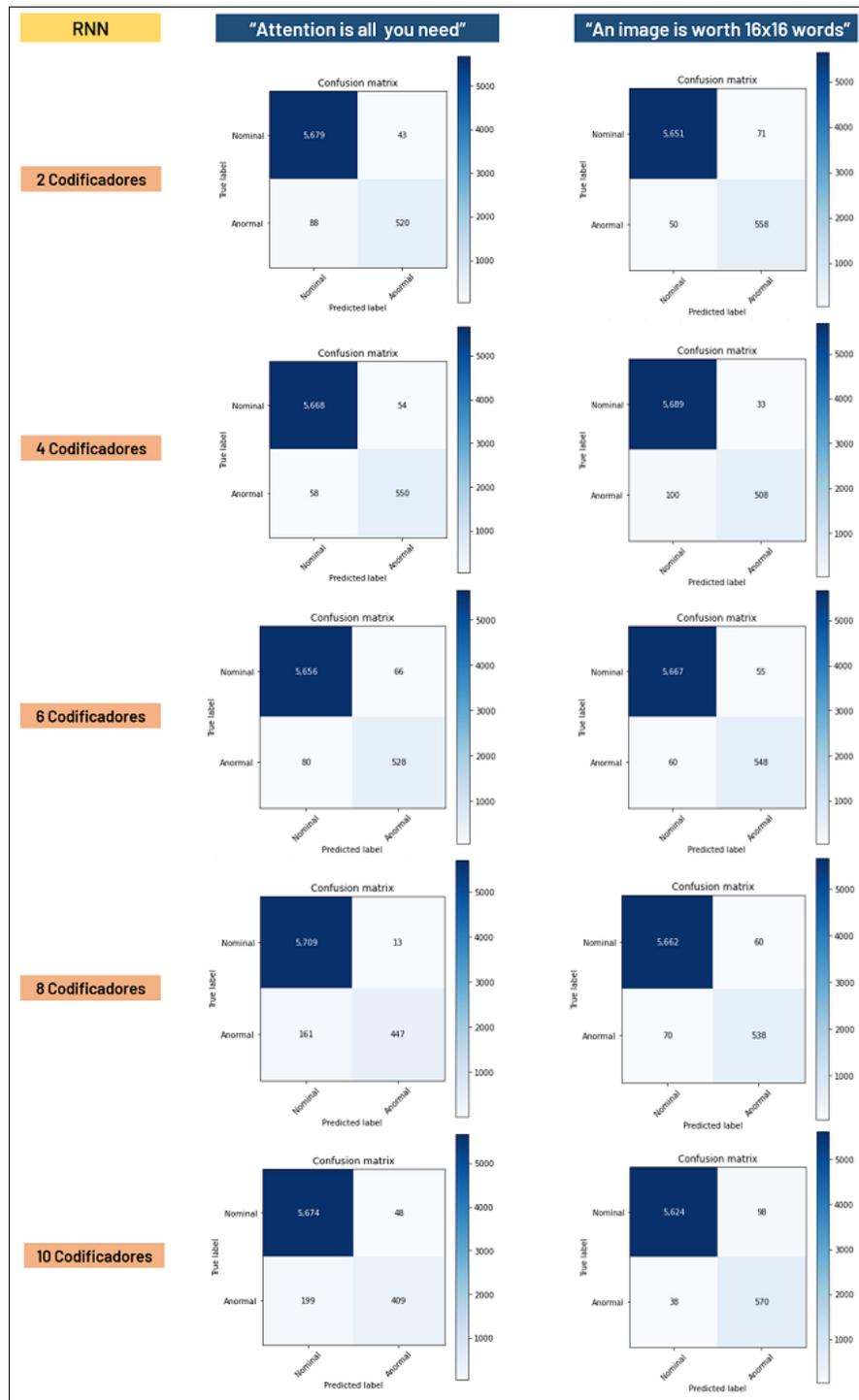
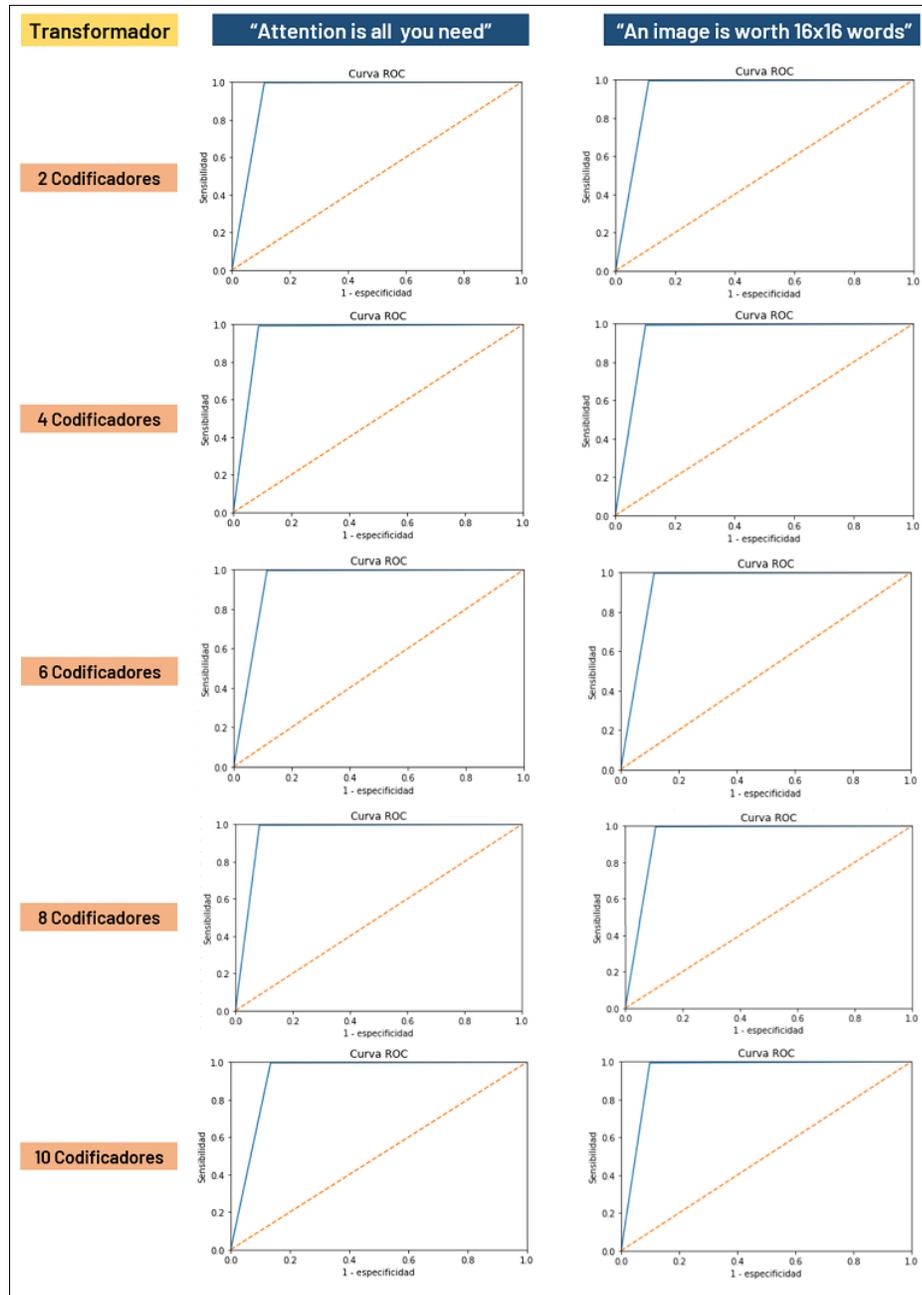


Figura 56: Comparación de matrices de confusión de clasificación de variantes del modelo RNN.

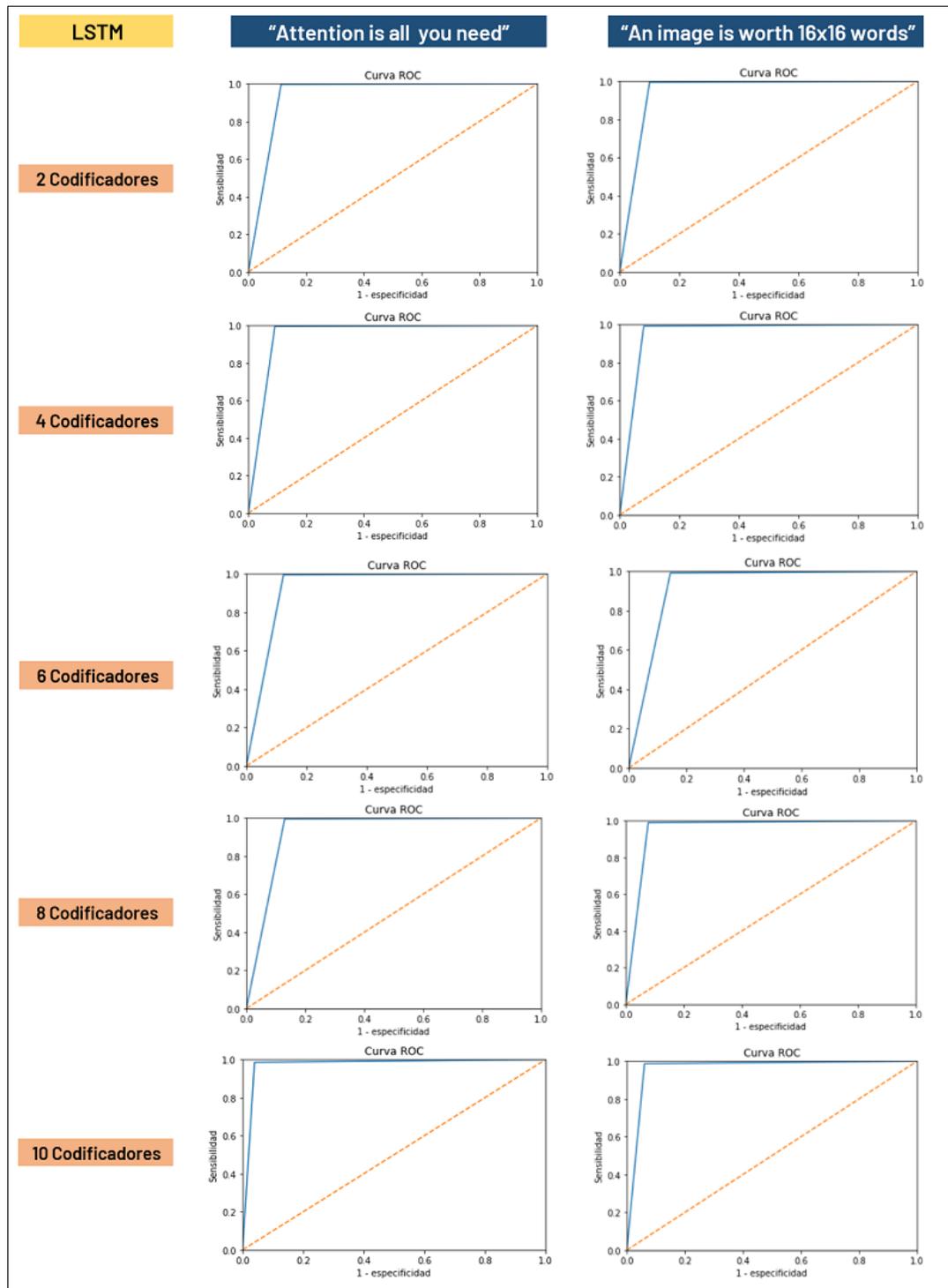
## Anexo E: Curva característica de modelos alternativos para clasificación

### Curva característica operativa del receptor de modelos alternativos de tipo Transformador



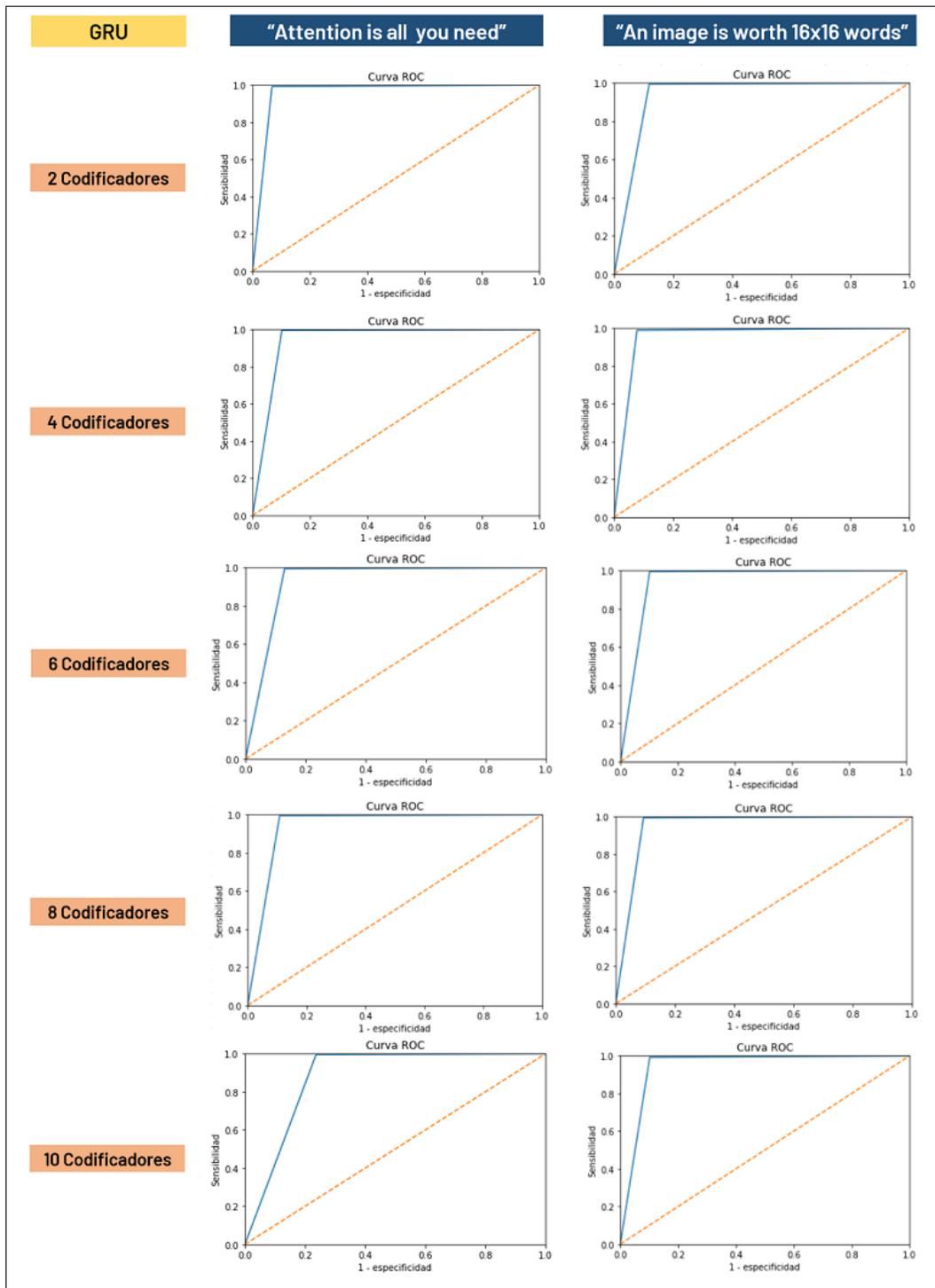
*Figura 57: Comparación de curvas características de variantes del modelo transformador.*

**Curva característica operativa del receptor de modelos alternativos de tipo LSTM**



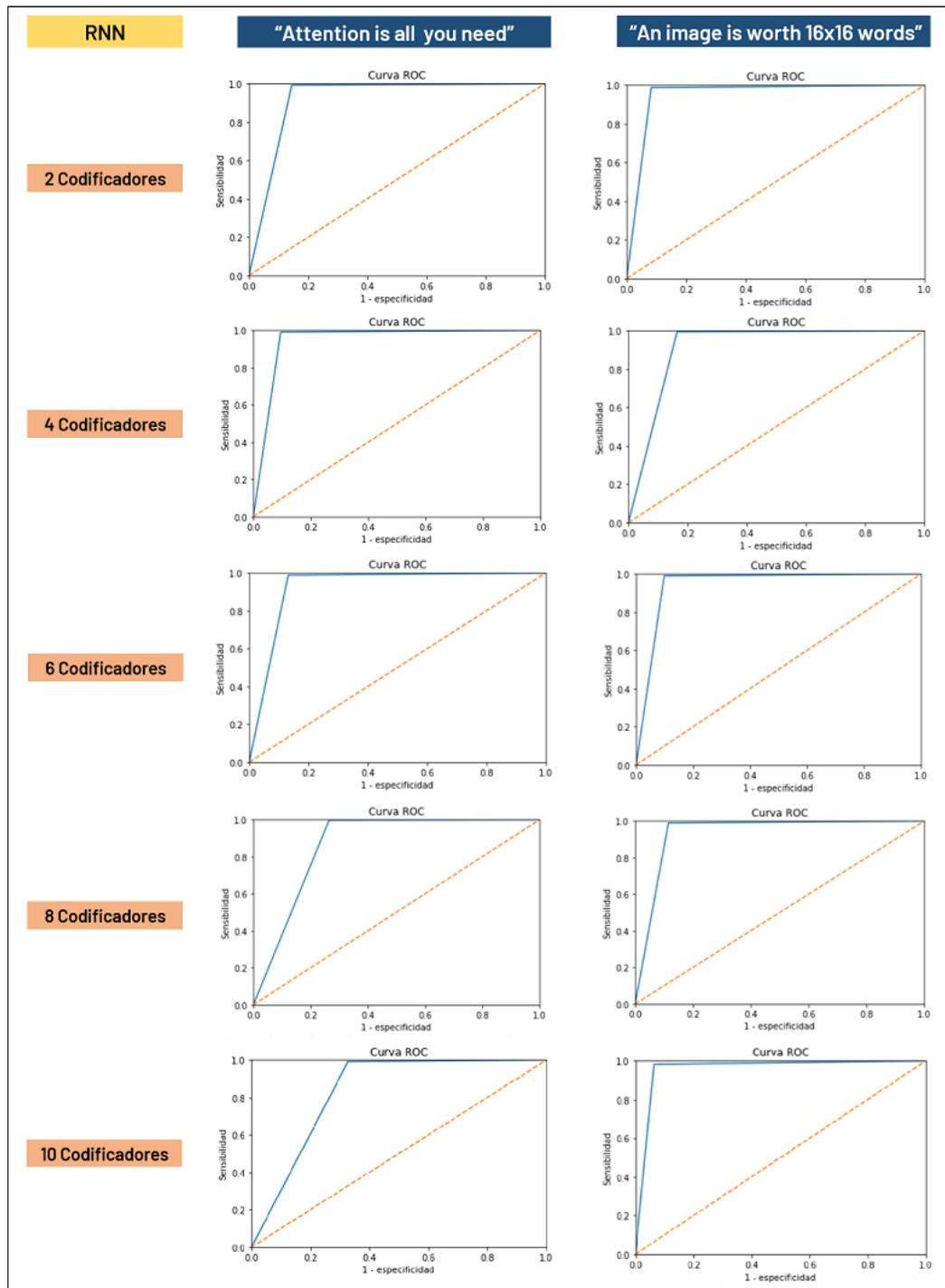
*Figura 58: Comparación de curvas características de variantes del modelo LSTM.*

**Curva característica operativa del receptor de modelos alternativos de tipo GRU**



*Figura 59: Comparación de curvas características de variantes del modelo GRU.*

**Curva característica operativa del receptor de modelos alternativos de tipo RNN**



*Figura 60: Comparación de curvas características de variantes del modelo RNN.*

## Anexo F: Registro de resultados de modelos alternativos en clasificación

### Anexo F.1: Arquitectura “Attention is all you need”

#### a) Modelo neuronal alternativo de dos codificadores

*Tabla 34: Datos de desempeño logrado en clasificación de cada tipo de modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por dos módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
Tiempo de entrenamiento [s]	Intento 1	1701	1845	1318	1737
	Intento 2	1709	1838	1786	1734
	Intento 3	1753	1960	1818	1724
	Intento 4	1676	1714	1834	1703
	Intento 5	1712	1963	1753	1914
	Intento 6	1725	1787	1984	1771
	Intento 7	1704	1835	1898	1702
	Intento 8	1683	1854	1563	1519
	Intento 9	1810	1925	1893	1731
	Intento 10	1753	1789	1878	1734
Exactitud [%]	Intento 1	98,64	98,64	98,66	97,93
	Intento 2	98,33	98,37	98,61	97,82
	Intento 3	98,64	98,28	98,42	97,88
	Intento 4	98,45	98,55	98,67	98,12
	Intento 5	98,52	98,55	98,58	97,76
	Intento 6	98,55	98,12	98,45	98,33
	Intento 7	98,69	98,36	98,55	98,20
	Intento 8	98,61	98,47	98,82	98,17
	Intento 9	98,56	98,66	98,56	98,07
	Intento 10	98,42	98,64	98,59	98,28
Precisión [%]	Intento 1	96,77	96,94	92,80	92,36
	Intento 2	91,42	92,58	92,07	90,80
	Intento 3	95,47	90,05	93,64	92,17
	Intento 4	91,94	92,16	97,64	95,03
	Intento 5	93,86	93,14	93,02	89,76
	Intento 6	98,50	89,12	90,87	94,66
	Intento 7	94,12	90,91	95,58	90,89
	Intento 8	91,94	91,82	95,71	89,55
	Intento 9	99,05	92,66	94,96	90,23
	Intento 10	95,52	93,21	93,03	92,22
Exhaustividad	Intento 1	88,82	88,65	93,26	85,53

Anexos

[%]	Intento 2	91,12	90,30	93,59	86,02
	Intento 3	90,13	92,27	89,64	85,20
	Intento 4	91,94	92,76	88,32	84,87
	Intento 5	90,46	91,61	92,11	86,51
	Intento 6	86,18	91,61	93,26	87,50
	Intento 7	92,11	92,11	88,98	90,30
	Intento 8	93,75	92,27	91,78	91,61
	Intento 9	85,86	93,42	89,80	89,64
	Intento 10	87,66	92,60	92,27	89,64
	<b>Puntaje valor F1</b> [%]	Intento 1	92,62	92,61	93,03
Intento 2		91,27	91,42	92,82	88,34
Intento 3		92,72	91,15	91,60	88,55
Intento 4		91,94	92,46	92,75	89,66
Intento 5		92,13	92,37	92,56	88,11
Intento 6		91,93	90,35	92,05	90,94
Intento 7		93,1	91,50	92,16	90,59
Intento 8		92,83	92,04	93,70	90,57
Intento 9		91,98	93,04	92,31	89,93
Intento 10		91,42	92,90	92,60	90,91
<b>Área bajo la curva característica</b> [%]	Intento 1	94,25	94,18	96,24	92,39
	Intento 2	95,10	94,76	96,36	92,55
	Intento 3	94,84	95,59	94,50	92,21
	Intento 4	95,54	95,96	94,05	92,20
	Intento 5	94,92	95,45	95,69	92,73
	Intento 6	93,02	95,21	96,13	93,49
	Intento 7	95,75	95,56	94,27	94,67
	Intento 8	96,44	95,70	95,67	95,24
	Intento 9	92,88	96,32	94,65	94,30
	Intento 10	93,61	95,94	95,77	94,42

## b) Modelo neuronal alternativo de cuatro codificadores

*Tabla 35: Datos de desempeño logrado en clasificación de cada tipo de modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por cuatro módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
<b>Tiempo de entrenamiento</b> [s]	Intento 1	2856	3112	3204	2698
	Intento 2	2796	3115	3054	2877
	Intento 3	2493	3116	2689	2546
	Intento 4	2437	3306	2724	2672
	Intento 5	2815	2727	3401	3089
	Intento 6	2726	3294	2826	2867
	Intento 7	2983	3455	3212	2780
	Intento 8	2898	3293	2827	2996
	Intento 9	2926	2942	3461	2773
	Intento 10	2989	3335	3009	3051
<b>Exactitud</b> [%]	Intento 1	98,42	98,74	98,66	98,23
	Intento 2	98,50	98,45	98,69	98,06
	Intento 3	98,44	99,02	98,53	98,04
	Intento 4	98,55	98,72	98,58	97,69
	Intento 5	98,61	98,25	98,48	98,04
	Intento 6	98,61	98,56	98,34	98,31
	Intento 7	98,93	98,82	98,23	97,79
	Intento 8	98,78	98,34	98,64	97,95
	Intento 9	98,55	98,36	98,55	98,31
	Intento 10	98,55	98,67	98,25	97,93
<b>Precisión</b> [%]	Intento 1	92,19	95,83	95,96	91,06
	Intento 2	96,55	97,22	93,39	98,21
	Intento 3	90,59	98,06	95,58	89,16
	Intento 4	95,91	93,41	92,88	90,81
	Intento 5	90,75	94,30	96,21	93,37
	Intento 6	93,05	92,31	93,59	90,86
	Intento 7	94,55	94,05	90,13	94,66
	Intento 8	94,32	90,89	91,83	93,77
	Intento 9	95,26	91,04	93,00	91,96
	Intento 10	90,06	93,52	90,94	93,44
<b>Exhaustividad</b> [%]	Intento 1	91,28	90,79	89,80	90,46
	Intento 2	87,50	86,35	92,93	81,25
	Intento 3	93,42	91,61	88,82	90,62
	Intento 4	88,65	93,26	92,27	84,54
	Intento 5	95,23	87,01	87,66	85,69

	Intento 6	92,43	92,76	88,82	91,61
	Intento 7	94,24	93,59	91,61	81,58
	Intento 8	92,93	91,94	94,24	84,21
	Intento 9	89,31	91,94	91,78	90,30
	Intento 10	95,39	92,60	90,79	84,38
<b>Puntaje valor F1 [%]</b>	Intento 1	91,74	93,24	92,78	90,76
	Intento 2	91,80	91,46	93,16	88,93
	Intento 3	91,98	94,73	92,07	89,89
	Intento 4	92,14	93,33	92,57	87,56
	Intento 5	92,94	90,50	91,74	89,37
	Intento 6	92,74	92,53	91,14	91,24
	Intento 7	94,40	93,82	90,86	87,63
	Intento 8	93,62	91,41	93,02	88,73
	Intento 9	92,19	91,49	92,38	91,12
	Intento 10	92,65	93,06	90,86	88,68
<b>Área bajo la curva característica [%]</b>	Intento 1	95,23	95,19	94,70	94,76
	Intento 2	93,58	93,04	96,11	90,55
	Intento 3	96,19	95,71	94,19	94,73
	Intento 4	94,12	96,28	95,76	91,82
	Intento 5	97,10	93,22	93,65	92,52
	Intento 6	95,85	95,97	94,08	95,32
	Intento 7	96,83	96,48	95,27	90,54
	Intento 8	96,17	95,48	96,68	91,81
	Intento 9	94,42	95,49	95,52	94,73
	Intento 10	97,14	95,96	94,91	91,87

**c) Modelo neuronal alternativo de seis codificadores**

*Tabla 36 : Datos de desempeño logrado en clasificación de cada tipo de modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por seis módulos de codificadores*

<b>Indicador de rendimiento</b>	<b>Tipo de red neuronal</b>	<b>Transformador</b>	<b>LSTM</b>	<b>GRU</b>	<b>RNN</b>
<b>Tiempo de entrenamiento [s]</b>	Intento 1	2374	4206	4029	3738
	Intento 2	3187	4073	3504	3140
	Intento 3	3059	4293	3296	3204
	Intento 4	2969	4201	3559	3007
	Intento 5	3147	3959	3297	3730
	Intento 6	3231	4108	3650	2729
	Intento 7	3141	4302	3196	3412
	Intento 8	3123	4208	4098	2785
	Intento 9	3145	4335	3323	2675
	Intento 10	2992	4352	3493	3002
<b>Exactitud [%]</b>	Intento 1	98,42	98,37	98,25	97,69
	Intento 2	98,36	98,82	98,74	97,41
	Intento 3	98,39	98,42	98,70	97,14
	Intento 4	98,64	98,44	98,44	97,57
	Intento 5	98,48	98,7	98,26	96,97
	Intento 6	98,40	98,39	98,55	97,27
	Intento 7	98,23	98,37	98,18	97,47
	Intento 8	98,45	98,45	98,28	97,22
	Intento 9	98,53	98,64	98,21	96,73
	Intento 10	98,55	98,88	98,75	97,99
<b>Precisión [%]</b>	Intento 1	90,19	95,01	94,30	88,89
	Intento 2	94,52	93,90	92,86	92,05
	Intento 3	96,68	92,19	96,63	89,76
	Intento 4	96,94	89,46	95,69	92,04
	Intento 5	99,61	93,98	90,29	94,64
	Intento 6	94,24	92,31	97,60	92,90
	Intento 7	91,20	90,40	91,43	90,88
	Intento 8	96,36	93,97	93,85	86,73
	Intento 9	96,73	93,94	90,24	88,78
	Intento 10	94,48	95,90	95,21	88,36
<b>Exhaustividad [%]</b>	Intento 1	93,75	87,66	87,01	86,84
	Intento 2	87,99	93,75	94,08	79,93
	Intento 3	86,18	91,28	89,64	79,28
	Intento 4	88,65	94,90	87,66	81,74

Anexos

	Intento 5	84,54	92,43	91,78	72,53
	Intento 6	88,82	90,79	87,01	77,47
	Intento 7	90,30	92,93	89,47	81,91
	Intento 8	87,17	89,64	87,83	83,88
	Intento 9	87,66	91,78	91,28	75,49
	Intento 10	90,13	92,27	91,61	91,12
<b>Puntaje valor F1 [%]</b>	Intento 1	91,94	91,19	90,50	87,85
	Intento 2	91,14	93,83	93,46	85,56
	Intento 3	91,13	91,74	93,00	84,19
	Intento 4	92,61	92,10	91,50	86,59
	Intento 5	91,46	93,20	91,03	82,12
	Intento 6	91,45	91,54	92,00	84,48
	Intento 7	90,74	91,65	90,44	86,16
	Intento 8	91,54	91,75	90,74	85,28
	Intento 9	91,98	92,85	90,76	81,60
	Intento 10	92,26	94,05	93,38	89,72
<b>Área bajo la curva característica [%]</b>	Intento 1	96,33	93,59	93,22	92,84
	Intento 2	93,73	96,55	96,65	89,60
	Intento 3	92,93	95,23	94,65	89,16
	Intento 4	94,18	96,86	93,62	86,59
	Intento 5	92,25	95,90	95,30	86,05
	Intento 6	94,12	94,99	93,39	88,42
	Intento 7	94,68	95,94	94,29	90,52
	Intento 8	93,41	94,51	93,61	91,26
	Intento 9	93,67	95,57	95,12	87,24
	Intento 10	94,79	95,93	95,56	94,92

#### d) Modelo neuronal alternativo de ocho codificadores

*Tabla 37: Datos de desempeño logrado en clasificación de cada tipo de modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por ocho módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
<b>Tiempo de entrenamiento [s]</b>	Intento 1	3763	5057	4577	4404
	Intento 2	4546	5118	4339	4658
	Intento 3	3943	5298	4519	4595
	Intento 4	4679	5359	4818	4413
	Intento 5	4267	5430	4879	4360
	Intento 6	3883	5298	4940	4475
	Intento 7	4184	5420	4884	4413
	Intento 8	4065	5605	5004	4882
	Intento 9	4123	4939	4706	3898
	Intento 10	3762	5431	4704	4312
<b>Exactitud [%]</b>	Intento 1	98,83	98,33	98,53	97,30
	Intento 2	98,40	98,70	98,58	96,79
	Intento 3	98,50	98,28	98,55	97,06
	Intento 4	98,69	98,45	98,37	96,41
	Intento 5	98,56	98,31	98,31	96,87
	Intento 6	98,07	98,47	98,04	96,75
	Intento 7	98,36	98,63	98,44	97,05
	Intento 8	98,53	98,59	98,45	96,82
	Intento 9	98,45	98,58	98,26	97,25
	Intento 10	98,37	98,37	98,53	96,98
<b>Precisión [%]</b>	Intento 1	96,03	95,14	95,25	90,69
	Intento 2	95,35	93,98	90,09	85,59
	Intento 3	98,49	90,97	96,91	97,10
	Intento 4	97,81	90,22	88,55	97,04
	Intento 5	96,41	90,73	92,97	96,38
	Intento 6	90,23	91,82	88,05	92,41
	Intento 7	91,72	95,46	91,65	95,66
	Intento 8	95,25	91,00	92,08	90,62
	Intento 9	96,36	91,91	88,79	97,17
	Intento 10	95,17	91,19	96,40	93,71
<b>Exhaustividad [%]</b>	Intento 1	91,61	87,01	89,14	80,10
	Intento 2	87,66	92,43	95,72	80,10
	Intento 3	85,69	91,12	87,66	71,55
	Intento 4	88,32	94,08	95,39	64,64

Anexos

	Intento 5	88,32	91,78	89,14	70,07
	Intento 6	89,64	92,27	92,11	72,04
	Intento 7	91,12	89,97	92,11	72,53
	Intento 8	89,14	94,74	91,78	74,67
	Intento 9	87,17	93,42	93,75	73,52
	Intento 10	87,50	91,94	87,99	73,52
<b>Puntaje valor F1 [%]</b>	Intento 1	93,77	90,89	92,10	85,07
	Intento 2	91,35	93,20	92,82	82,75
	Intento 3	91,64	91,04	92,06	82,39
	Intento 4	92,83	92,11	91,84	77,59
	Intento 5	92,19	91,25	91,02	81,14
	Intento 6	89,93	92,04	90,03	80,96
	Intento 7	91,42	92,63	91,88	82,51
	Intento 8	92,10	92,83	91,93	81,88
	Intento 9	91,54	92,66	91,20	83,71
	Intento 10	91,17	91,56	92,00	82,40
<b>Área bajo la curva característica [%]</b>	Intento 1	95,60	93,27	94,34	89,61
	Intento 2	93,61	95,90	97,30	89,33
	Intento 3	92,78	95,08	93,68	85,66
	Intento 4	94,06	96,50	97,04	82,21
	Intento 5	93,99	95,39	94,21	84,89
	Intento 6	94,30	95,70	95,39	85,71
	Intento 7	95,12	94,76	95,61	86,09
	Intento 8	94,34	96,87	95,47	86,92
	Intento 9	93,41	96,27	96,25	86,65
	Intento 10	93,51	95,50	93,82	86,50

e) **Modelo neuronal alternativo de diez codificadores**

*Tabla 38: Datos de desempeño logrado en clasificación de cada tipo de modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por diez módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
<b>Tiempo de entrenamiento [s]</b>	Intento 1	4938	6069	5248	4783
	Intento 2	5228	5798	5996	5522
	Intento 3	5262	5978	5939	5403
	Intento 4	5225	6585	5506	5891
	Intento 5	5346	6287	5873	356
	Intento 6	5222	6595	6117	3818
	Intento 7	5103	6485	5578	5160
	Intento 8	5286	6045	6058	4978
	Intento 9	5354	5855	5880	5465
	Intento 10	5221	6107	5912	5341
<b>Exactitud [%]</b>	Intento 1	98,47	98,34	97,61	96,10
	Intento 2	97,84	98,36	97,33	96,71
	Intento 3	98,18	98,53	96,78	96,98
	Intento 4	98,21	98,67	97,06	96,32
	Intento 5	97,96	98,4	97,39	96,60
	Intento 6	98,42	98,14	97,42	97,11
	Intento 7	98,17	98,53	98,21	96,92
	Intento 8	97,91	98,36	98,44	96,75
	Intento 9	98,45	98,70	96,60	96,57
	Intento 10	98,31	98,61	98,12	96,65
<b>Precisión [%]</b>	Intento 1	97,23	87,82	92,71	89,50
	Intento 2	88,42	90,13	94,89	98,08
	Intento 3	96,25	90,81	92,26	97,28
	Intento 4	93,65	95,96	88,09	99,21
	Intento 5	92,54	90,05	94,57	91,54
	Intento 6	96,69	89,77	97,44	95,31
	Intento 7	94,57	92,28	87,79	91,72
	Intento 8	89,67	94,06	96,02	94,47
	Intento 9	95,70	93,83	84,29	89,02
	Intento 10	96,47	93,62	97,29	92,31
<b>Exhaustividad [%]</b>	Intento 1	86,51	96,05	81,58	67,27
	Intento 2	89,14	93,09	76,32	67,11
	Intento 3	84,38	94,24	72,53	70,56
	Intento 4	87,34	89,97	80,26	62,17

Anexos

	Intento 5	85,69	93,70	77,30	71,22
	Intento 6	86,51	90,95	75,16	73,52
	Intento 7	85,86	92,43	94,57	74,67
	Intento 8	88,49	88,49	87,34	70,23
	Intento 9	87,83	92,60	79,44	73,36
	Intento 10	85,53	91,78	82,73	71,05
<b>Puntaje valor F1 [%]</b>	Intento 1	91,56	91,75	86,79	76,81
	Intento 2	88,78	91,59	84,59	79,69
	Intento 3	89,92	92,49	81,22	81,79
	Intento 4	90,38	92,87	83,99	76,44
	Intento 5	88,98	91,86	85,07	80,11
	Intento 6	91,32	90,36	84,87	83,01
	Intento 7	90,00	92,36	91,05	82,32
	Intento 8	89,07	91,19	91,47	80,57
	Intento 9	91,60	93,21	81,80	80,43
	Intento 10	90,67	92,69	89,42	80,30
<b>Área bajo la Curva característica [%]</b>	Intento 1	93,13	97,32	90,45	83,22
	Intento 2	93,95	96,00	87,94	83,48
	Intento 3	92,01	96,61	85,94	85,17
	Intento 4	93,35	94,78	89,55	81,06
	Intento 5	92,48	96,32	88,42	85,26
	Intento 6	93,10	94,93	87,48	86,57
	Intento 7	92,67	95,81	96,59	86,98
	Intento 8	93,70	93,95	93,48	84,90
	Intento 9	93,70	95,98	88,93	86,20
	Intento 10	92,60	95,56	91,24	85,21

## Anexo F.2: Arquitectura “An image is worth 16x16 words”

### a) Modelo neuronal alternativo de dos codificadores

*Tabla 39: Datos de desempeño logrado en clasificación de cada tipo de modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por dos módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
<b>Tiempo de entrenamiento [s]</b>	Intento 1	1588	1969	1873	1669
	Intento 2	1769	1958	1899	1721
	Intento 3	1562	1978	1974	1786
	Intento 4	1610	1943	1923	1671
	Intento 5	1565	1899	1759	1648
	Intento 6	1542	2045	1564	1652
	Intento 7	1769	1994	1914	1593
	Intento 8	1493	1853	1893	1669
	Intento 9	1547	1962	1872	1639
	Intento 10	1613	1989	1964	1652
<b>Exactitud [%]</b>	Intento 1	98,56	98,61	98,39	98,09
	Intento 2	98,55	98,44	98,25	98,42
	Intento 3	98,53	98,26	98,40	98,06
	Intento 4	98,33	98,26	98,55	97,88
	Intento 5	98,34	98,31	98,28	98,29
	Intento 6	98,55	98,42	98,21	98,06
	Intento 7	98,53	98,31	98,66	98,04
	Intento 8	98,63	98,04	98,66	98,42
	Intento 9	98,31	98,29	98,44	98,06
	Intento 10	98,29	98,42	98,47	98,04
<b>Precisión [%]</b>	Intento 1	93,45	95,30	94,70	88,71
	Intento 2	95,74	93,80	89,76	91,64
	Intento 3	91,20	97,52	90,82	94,50
	Intento 4	94,82	88,43	94,79	93,89
	Intento 5	91,85	93,26	95,78	91,81
	Intento 6	95,58	93,20	94,28	88,19
	Intento 7	98,31	91,54	91,84	93,37
	Intento 8	98,33	87,46	95,96	93,05
	Intento 9	95,63	92,52	91,93	90,48
	Intento 10	90,06	92,62	93,09	90,74
<b>Exhaustividad [%]</b>	Intento 1	91,45	89,97	88,16	91,78
	Intento 2	88,82	89,64	92,27	91,94

	Intento 3	93,75	84,05	92,76	84,70
	Intento 4	87,34	94,24	89,80	83,39
	Intento 5	90,79	88,82	85,86	90,30
	Intento 6	88,98	90,13	86,68	92,11
	Intento 7	86,18	90,79	94,41	85,69
	Intento 8	87,17	92,93	89,80	90,30
	Intento 9	86,35	89,47	91,78	89,14
	Intento 10	92,43	90,79	90,79	88,65
<b>Puntaje valor F1 [%]</b>	Intento 1	92,44	92,55	91,31	90,22
	Intento 2	92,15	91,67	91,00	91,79
	Intento 3	92,46	90,28	91,78	89,33
	Intento 4	90,92	91,24	92,23	88,33
	Intento 5	91,32	90,99	90,55	91,04
	Intento 6	92,16	91,64	90,32	90,10
	Intento 7	91,85	91,16	93,11	89,37
	Intento 8	92,41	90,11	92,78	91,65
	Intento 9	90,75	90,97	91,85	89,81
	Intento 10	91,23	91,69	91,92	89,68
<b>Área bajo la curva característica [%]</b>	Intento 1	95,38	94,75	93,82	95,27
	Intento 2	94,20	94,50	95,58	95,52
	Intento 3	96,39	91,91	95,88	92,09
	Intento 4	93,41	96,47	94,64	91,41
	Intento 5	94,97	94,07	92,73	94,72
	Intento 6	94,27	94,72	93,06	95,40
	Intento 7	93,01	94,95	96,76	92,52
	Intento 8	93,51	95,76	94,70	94,79
	Intento 9	92,96	94,35	95,46	94,07
	Intento 10	95,68	95,01	95,04	93,85

## b) Modelo neuronal alternativo de cuatro codificadores

*Tabla 40: Datos de desempeño logrado en clasificación de cada tipo de modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por cuatro módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
<b>Tiempo de entrenamiento[s]</b>	Intento 1	2734	2905	2878	2633
	Intento 2	2673	2937	2874	2699
	Intento 3	2702	2981	2993	2675
	Intento 4	2663	2497	2796	2596
	Intento 5	2756	3079	2890	2695
	Intento 6	2703	3004	2759	2842
	Intento 7	2631	2951	2893	2753
	Intento 8	2676	3001	2898	2624
	Intento 9	2593	2946	2906	2764
	Intento 10	2687	3032	2873	2705
<b>Exactitud [%]</b>	Intento 1	98,44	98,45	98,33	97,90
	Intento 2	98,26	98,34	98,42	98,15
	Intento 3	98,36	98,25	98,48	98,07
	Intento 4	98,66	98,28	98,70	97,96
	Intento 5	98,67	98,52	98,34	98,15
	Intento 6	98,42	98,55	98,75	98,07
	Intento 7	98,55	98,61	98,77	98,17
	Intento 8	98,52	98,37	98,47	98,10
	Intento 9	98,33	98,83	98,29	98,18
	Intento 10	98,75	98,63	98,74	98,09
<b>Precisión [%]</b>	Intento 1	93,65	91,94	90,48	93,90
	Intento 2	90,16	90,63	94,25	89,79
	Intento 3	91,04	87,71	92,81	93,71
	Intento 4	93,08	90,44	96,14	90,94
	Intento 5	94,26	94,62	90,76	95,55
	Intento 6	94,10	94,48	96,81	94,51
	Intento 7	93,14	93,05	95,22	93,62
	Intento 8	92,13	87,74	92,09	90,53
	Intento 9	94,04	97,17	90,19	92,43
	Intento 10	91,92	93,20	94,44	88,35
<b>Exhaustividad [%]</b>	Intento 1	89,80	91,94	92,27	83,55
	Intento 2	91,94	92,27	88,98	91,12
	Intento 3	91,94	95,07	91,28	85,69
	Intento 4	92,93	91,78	90,13	87,50
	Intento 5	91,78	89,64	92,11	84,70

	Intento 6	89,14	90,13	89,97	84,87
	Intento 7	91,61	92,43	91,78	86,84
	Intento 8	92,43	96,55	91,94	89,64
	Intento 9	88,16	90,46	92,27	88,32
	Intento 10	95,39	92,43	92,27	92,27
<b>Puntaje valor F1 [%]</b>	Intento 1	91,69	91,94	91,37	88,42
	Intento 2	91,04	91,44	91,54	90,45
	Intento 3	91,49	91,24	92,04	89,52
	Intento 4	93,00	91,10	93,04	89,19
	Intento 5	93,00	92,06	91,43	89,80
	Intento 6	91,55	92,26	93,27	89,43
	Intento 7	92,37	92,74	93,47	90,10
	Intento 8	92,28	91,93	92,02	90,08
	Intento 9	91,00	93,70	91,22	90,33
	Intento 10	93,62	92,82	93,34	90,27
<b>Área bajo la curva característica [%]</b>	Intento 1	94,58	95,54	95,62	91,49
	Intento 2	95,44	95,63	94,20	95,01
	Intento 3	95,49	96,83	95,27	92,54
	Intento 4	96,10	95,37	94,87	93,29
	Intento 5	95,59	94,55	95,55	92,14
	Intento 6	94,28	94,79	94,83	92,17
	Intento 7	95,45	95,85	95,64	93,11
	Intento 8	95,80	97,56	95,55	94,32
	Intento 9	93,78	95,09	95,60	93,78
	Intento 10	97,25	95,86	95,85	95,49

### c) Modelo neuronal alternativo de seis codificadores

**Tabla 41:** Datos de desempeño logrado en clasificación de cada tipo de modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por seis módulos de codificadores.

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
Tiempo de entrenamiento [s]	Intento 1	3101	4164	4035	3572
	Intento 2	3040	4195	4034	3551
	Intento 3	3043	4127	4139	3502
	Intento 4	3065	4151	4102	3625
	Intento 5	3021	4223	4049	3445
	Intento 6	3035	4206	3863	3453
	Intento 7	3029	4301	4031	3852
	Intento 8	3025	4225	3973	3494
	Intento 9	3051	4153	3919	3506
	Intento 10	3013	4256	3940	3535
Exactitud [%]	Intento 1	98,75	97,84	98,55	98,18
	Intento 2	98,64	98,50	98,56	97,99
	Intento 3	98,42	98,33	98,77	98,42
	Intento 4	98,21	98,26	98,85	97,95
	Intento 5	98,18	98,25	98,33	98,28
	Intento 6	98,56	98,59	98,69	97,95
	Intento 7	98,31	98,26	98,56	97,93
	Intento 8	98,59	97,93	98,21	97,95
	Intento 9	98,56	98,64	98,47	97,88
	Intento 10	98,21	98,58	98,34	98,15
Precisión [%]	Intento 1	96,65	91,39	94,79	90,88
	Intento 2	97,11	91,44	95,43	90,42
	Intento 3	98,42	89,59	91,93	95,68
	Intento 4	93,50	91,23	98,03	91,78
	Intento 5	93,63	87,03	93,13	94,95
	Intento 6	95,11	92,06	93,10	89,05
	Intento 7	92,24	90,29	92,73	88,78
	Intento 8	97,44	87,56	92,89	91,35
	Intento 9	96,24	94,85	94,28	89,24
	Intento 10	90,38	91,51	91,30	89,53
Exhaustividad [%]	Intento 1	90,13	85,53	89,80	90,13
	Intento 2	88,49	93,09	89,31	88,49
	Intento 3	90,84	93,42	95,56	87,50
	Intento 4	87,50	90,62	89,80	86,35

Anexos

	Intento 5	87,01	96,05	89,14	86,68
	Intento 6	89,64	93,42	93,26	89,64
	Intento 7	89,97	91,78	92,27	89,80
	Intento 8	87,66	91,45	88,16	86,84
	Intento 9	88,49	90,79	89,47	88,65
	Intento 10	91,12	93,91	91,45	91,45
<b>Puntaje valor F1 [%]</b>	Intento 1	93,28	88,36	92,23	90,5
	Intento 2	92,60	92,26	92,27	89,44
	Intento 3	92,93	91,47	93,71	91,41
	Intento 4	90,40	90,92	93,73	88,98
	Intento 5	90,20	91,32	91,09	90,63
	Intento 6	92,29	92,73	93,18	89,34
	Intento 7	91,09	91,03	92,50	89,29
	Intento 8	92,29	89,46	90,46	89,04
	Intento 9	92,20	92,77	91,81	88,94
	Intento 10	90,75	92,69	91,37	90,48
<b>Área bajo la curva característica [%]</b>	Intento 1	94,90	92,33	94,64	94,59
	Intento 2	94,10	96,08	94,43	93,75
	Intento 3	95,97	96,13	97,33	93,54
	Intento 4	93,43	94,85	94,81	92,76
	Intento 5	93,19	97,27	94,22	93,09
	Intento 6	94,57	96,28	96,26	94,23
	Intento 7	94,58	95,36	95,75	94,30
	Intento 8	93,71	95,03	93,72	92,98
	Intento 9	94,06	95,13	94,45	93,76
	Intento 10	95,04	96,49	95,26	95,16

### d) Modelo neuronal alternativo de ocho codificadores

*Tabla 42: Datos de desempeño logrado en clasificación de cada tipo de modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por ocho módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
<b>Tiempo de entrenamiento [s]</b>	Intento 1	4244	5269	4814	4352
	Intento 2	3880	4817	4757	3922
	Intento 3	4065	5430	5060	4300
	Intento 4	3824	4751	4818	4353
	Intento 5	3862	4814	4881	3979
	Intento 6	4124	5252	5001	4599
	Intento 7	3945	5155	5002	4351
	Intento 8	3881	5298	4939	4600
	Intento 9	3704	5484	4884	4412
	Intento 10	4066	5551	4941	4103
<b>Exactitud [%]</b>	Intento 1	98,64	98,14	98,48	97,84
	Intento 2	98,58	98,29	98,52	98,09
	Intento 3	98,55	98,26	98,64	98,07
	Intento 4	98,66	98,45	98,55	98,12
	Intento 5	98,47	98,44	98,42	98,14
	Intento 6	98,55	98,37	98,67	97,91
	Intento 7	98,50	98,09	98,74	98,12
	Intento 8	98,77	98,55	98,58	98,06
	Intento 9	98,21	98,31	98,66	98,03
	Intento 10	98,40	98,50	98,42	98,20
<b>Precisión [%]</b>	Intento 1	96,28	88,52	91,16	86,40
	Intento 2	94,35	88,70	98,31	90,38
	Intento 3	94,64	88,31	93,21	93,09
	Intento 4	96,61	89,97	93,73	89,89
	Intento 5	93,23	92,35	92,76	93,29
	Intento 6	98,68	90,92	93,96	89,53
	Intento 7	92,26	90,12	93,56	91,65
	Intento 8	95,53	91,75	94,35	86,69
	Intento 9	93,50	91,40	91,97	89,53
	Intento 10	93,33	93,70	90,84	94,42
<b>Exhaustividad [%]</b>	Intento 1	89,31	92,60	93,26	91,94
	Intento 2	90,62	94,24	86,02	89,64
	Intento 3	89,97	94,41	92,60	86,35
	Intento 4	89,14	94,41	90,95	90,62
	Intento 5	90,62	91,28	90,62	86,84

	Intento 6	86,02	92,27	92,11	88,65
	Intento 7	92,11	89,97	93,26	88,49
	Intento 8	91,45	93,26	90,62	94,24
	Intento 9	87,50	90,95	94,24	89,97
	Intento 10	89,80	90,46	92,93	86,35
<b>Puntaje valor F1 [%]</b>	Intento 1	92,66	90,51	92,20	89,08
	Intento 2	92,45	91,39	91,75	90,01
	Intento 3	92,24	91,26	92,90	89,59
	Intento 4	92,73	92,13	92,32	90,25
	Intento 5	91,91	91,81	91,68	89,95
	Intento 6	91,92	91,59	93,02	89,22
	Intento 7	92,18	90,04	93,41	90,04
	Intento 8	93,45	92,50	92,45	90,31
	Intento 9	90,40	91,18	93,10	89,75
	Intento 10	91,53	92,05	91,87	90,21
<b>Área bajo la curva característica [%]</b>	Intento 1	94,47	95,66	95,94	94,31
	Intento 2	96,01	96,48	92,93	94,32
	Intento 3	94,71	96,54	95,94	92,83
	Intento 4	94,41	96,64	95,15	94,77
	Intento 5	94,96	95,24	94,94	93,09
	Intento 6	92,95	95,65	95,74	93,78
	Intento 7	95,64	94,46	96,29	93,82
	Intento 8	95,50	96,18	95,02	96,35
	Intento 9	93,43	95,02	96,68	94,42
	Intento 10	94,56	94,91	95,97	92,90

e) **Modelo neuronal alternativo de diez codificadores**

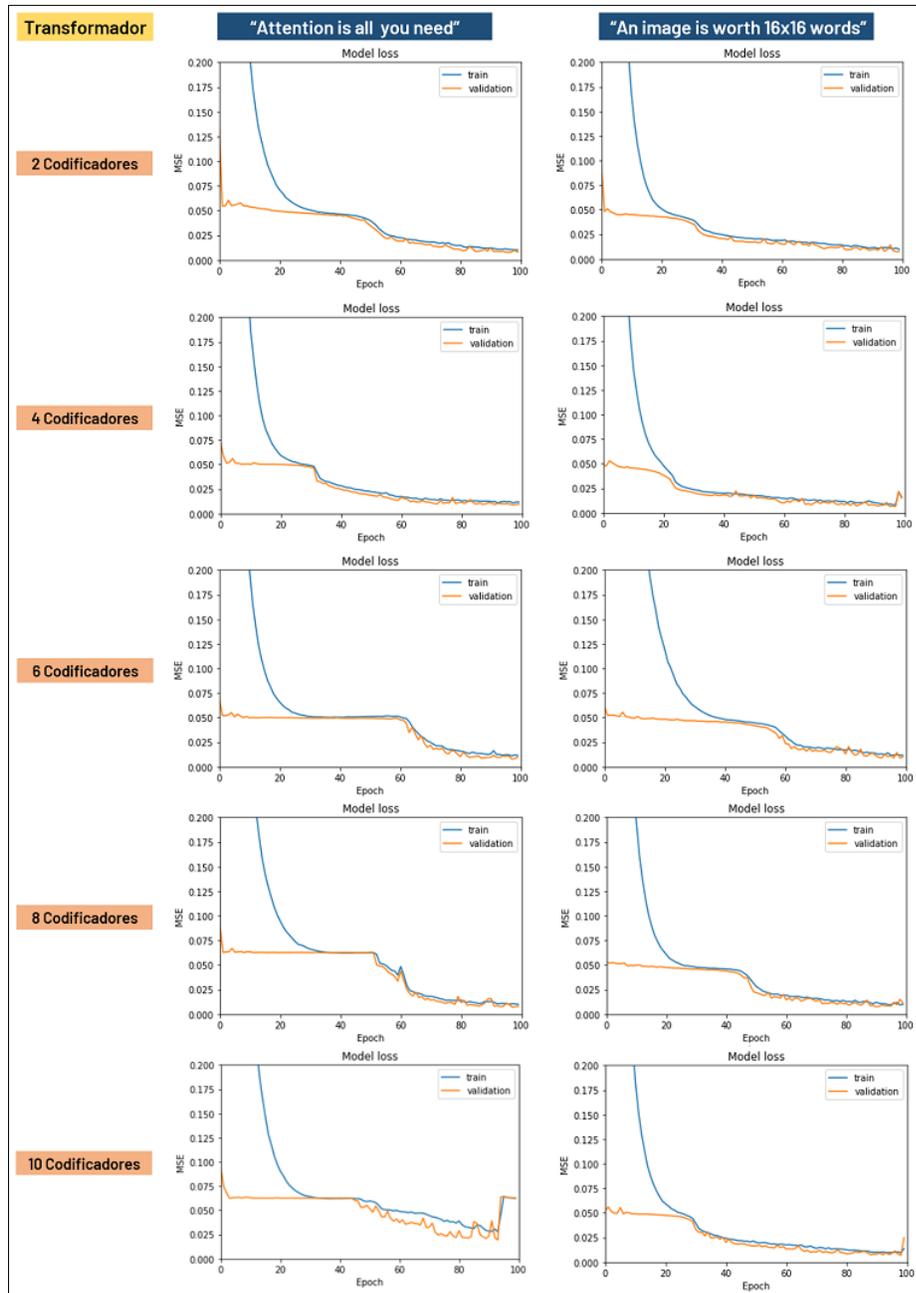
*Tabla 43: Datos de desempeño logrado en clasificación de cada tipo de modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por diez módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformador	LSTM	GRU	RNN
<b>Tiempo de entrenamiento [s]</b>	Intento 1	4623	6157	5844	5622
	Intento 2	4653	5698	6401	5758
	Intento 3	5287	6523	6063	5539
	Intento 4	5164	6448	6288	5520
	Intento 5	4660	6592	6009	5886
	Intento 6	4501	6113	5889	5834
	Intento 7	4743	6447	5934	5525
	Intento 8	4623	5930	6295	5887
	Intento 9	5104	6227	6242	4857
	Intento 10	4982	6473	5873	4156
<b>Exactitud [%]</b>	Intento 1	98,50	98,25	98,59	97,87
	Intento 2	98,67	98,23	98,66	98,34
	Intento 3	98,42	98,36	98,44	97,95
	Intento 4	98,48	98,31	98,29	97,85
	Intento 5	98,59	98,39	98,34	97,85
	Intento 6	98,40	98,64	98,45	97,98
	Intento 7	98,52	98,63	98,59	97,93
	Intento 8	98,37	98,47	98,56	98,37
	Intento 9	98,53	98,42	98,59	98,15
	Intento 10	98,17	98,26	98,59	98,29
<b>Precisión [%]</b>	Intento 1	95,89	88,53	93,32	91,86
	Intento 2	94,56	92,93	92,94	91,57
	Intento 3	97,57	92,00	90,72	91,35
	Intento 4	93,84	91,82	92,66	91,55
	Intento 5	93,03	91,34	92,70	85,33
	Intento 6	95,03	93,50	96,36	92,11
	Intento 7	94,31	93,63	93,47	88,65
	Intento 8	91,87	90,11	93,30	92,44
	Intento 9	90,81	91,64	92,75	90,85
	Intento 10	91,55	90,69	97,61	93,55
<b>Exhaustividad [%]</b>	Intento 1	88,16	93,91	91,94	85,36
	Intento 2	91,45	89,31	93,09	91,12
	Intento 3	85,69	90,79	93,26	86,84
	Intento 4	90,13	90,46	89,31	85,53

	Intento 5	92,27	91,94	89,80	93,75
	Intento 6	87,99	92,27	87,17	86,35
	Intento 7	89,97	91,94	91,78	89,97
	Intento 8	91,12	94,41	91,61	90,46
	Intento 9	94,24	91,94	92,60	89,80
	Intento 10	89,14	91,28	87,50	88,32
<b>Puntaje valor F1 [%]</b>	Intento 1	91,86	91,14	92,63	88,49
	Intento 2	92,98	90,65	93,02	91,34
	Intento 3	91,24	91,39	91,97	89,04
	Intento 4	91,95	91,14	90,95	88,44
	Intento 5	92,65	91,64	91,23	89,34
	Intento 6	91,37	92,88	91,54	89,13
	Intento 7	92,09	92,78	92,61	89,31
	Intento 8	91,49	92,21	92,45	91,44
	Intento 9	92,49	91,79	92,67	90,32
	Intento 10	90,33	90,98	92,28	90,86
<b>Área bajo la Curva característica [%]</b>	Intento 1	93,88	96,31	95,62	92,28
	Intento 2	95,44	94,24	96,17	95,11
	Intento 3	92,73	94,98	96,12	92,98
	Intento 4	94,75	94,80	94,28	92,34
	Intento 5	95,77	95,51	94,53	96,02
	Intento 6	93,75	95,79	93,41	92,78
	Intento 7	94,70	95,64	95,55	94,37
	Intento 8	95,13	96,65	95,46	94,84
	Intento 9	96,61	95,52	95,91	94,42
	Intento 10	94,14	95,14	93,64	93,84

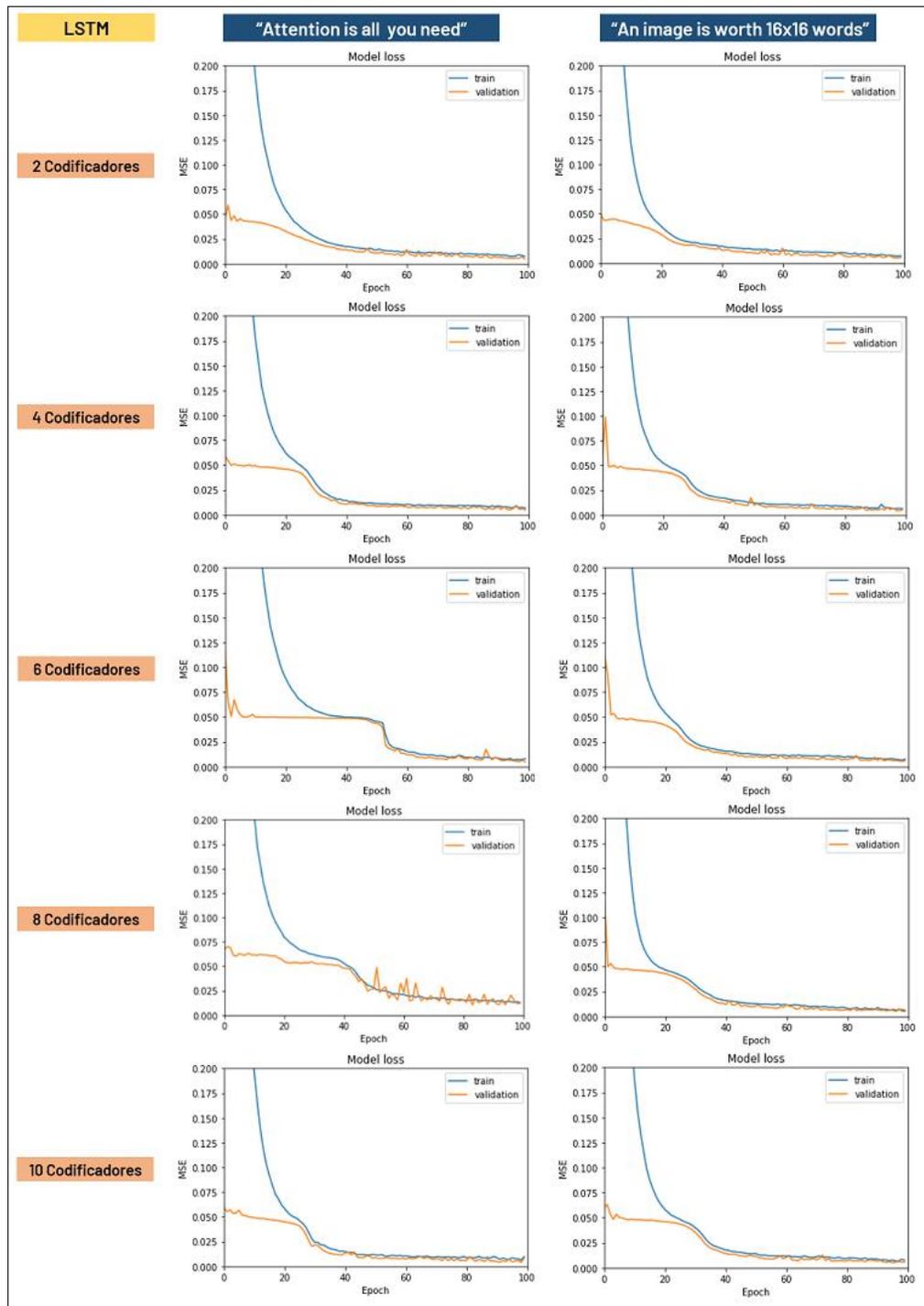
## Anexo G: Funciones de pérdida para regresión de modelos alternativos

### Función de pérdida de modelos alternativos de tipo Transformador



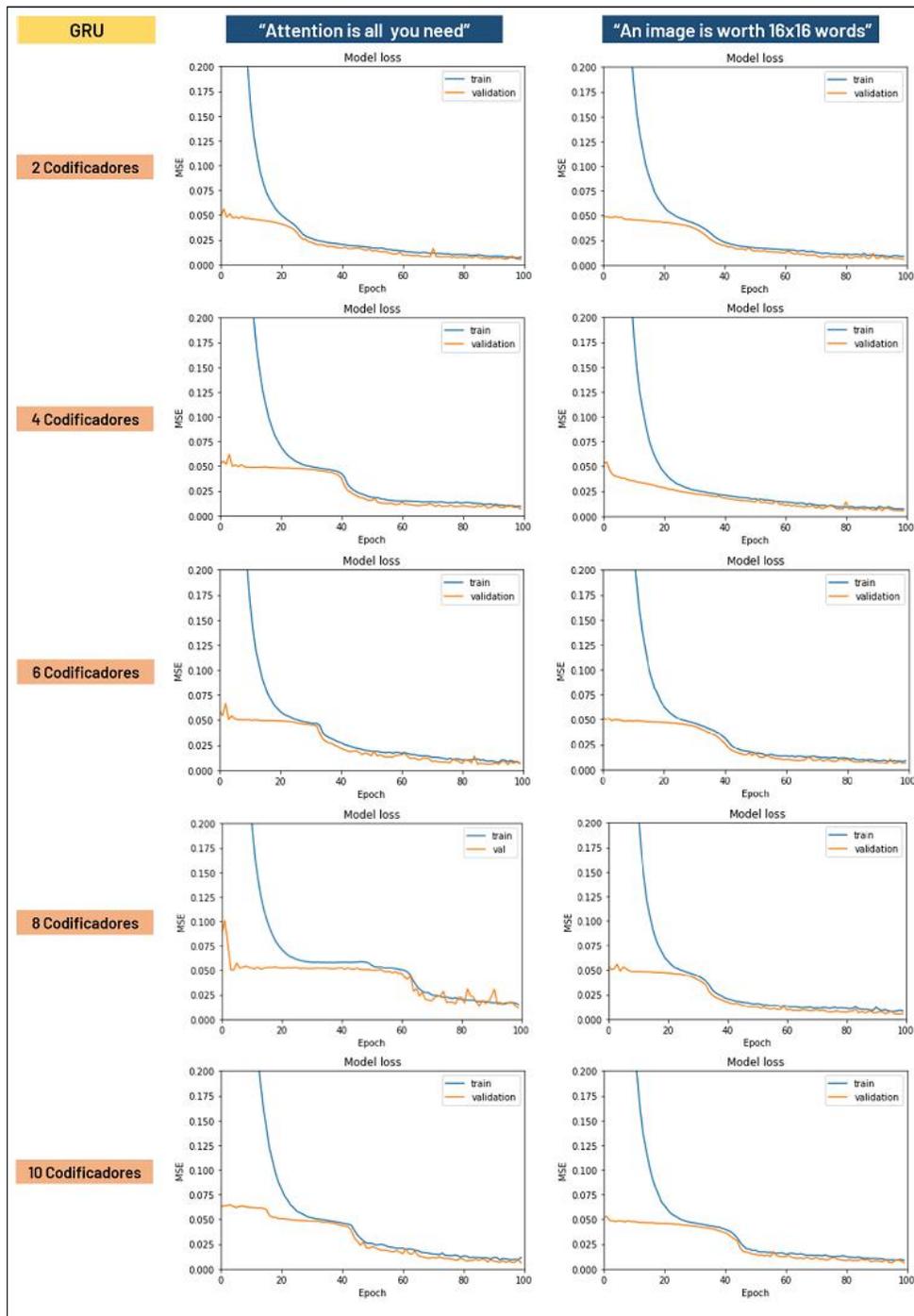
*Figura 61: Comparación de funciones de pérdida en regresión de variantes del modelo transformador*

## Función de pérdida de modelos alternativos de tipo LSTM



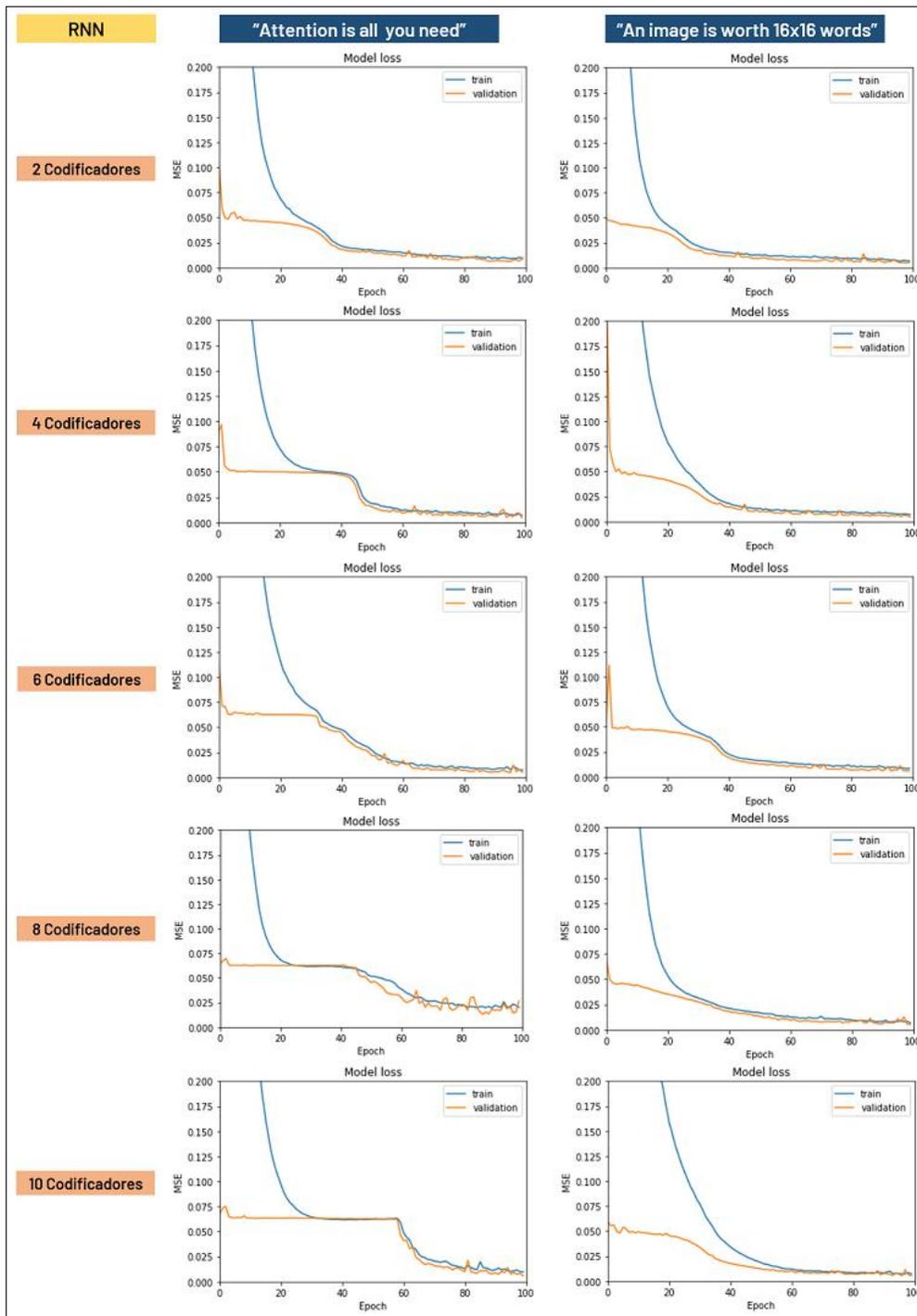
*Figura 62: Comparación de funciones de pérdida en regresión de variantes del modelo LSTM.*

## Función de pérdida de modelos alternativos de tipo GRU



*Figura 63: Comparación de funciones de pérdida en regresión de variantes del modelo GRU.*

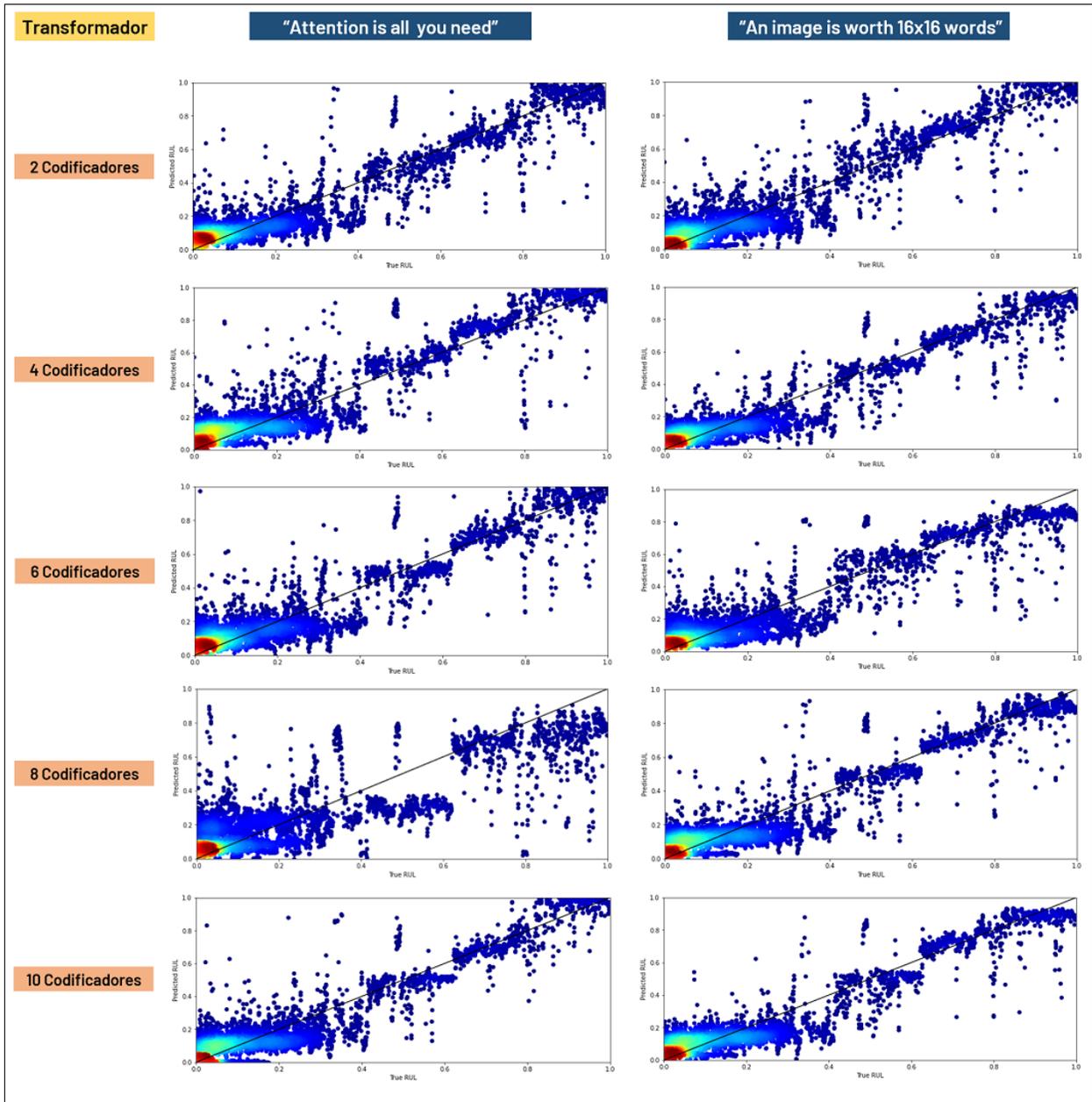
**Función de pérdida de modelos alternativos de tipo RNN**



*Figura 64: Comparación de funciones de pérdida en regresión de variantes del modelo RNN.*

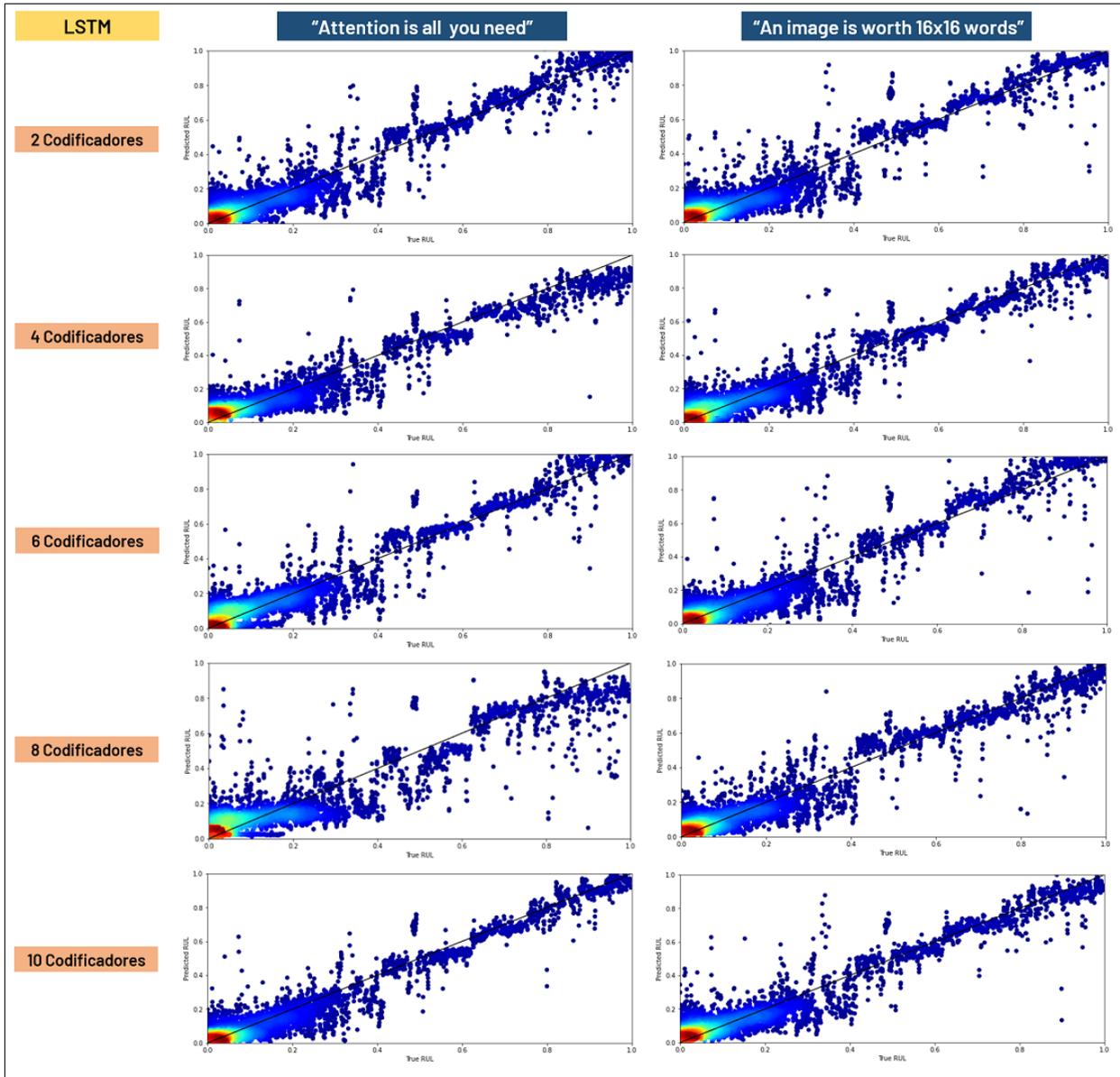
## Anexo H: Diagrama de dispersión de modelos alternativos

### Diagrama de dispersión de modelos alternativos de tipo Transformador



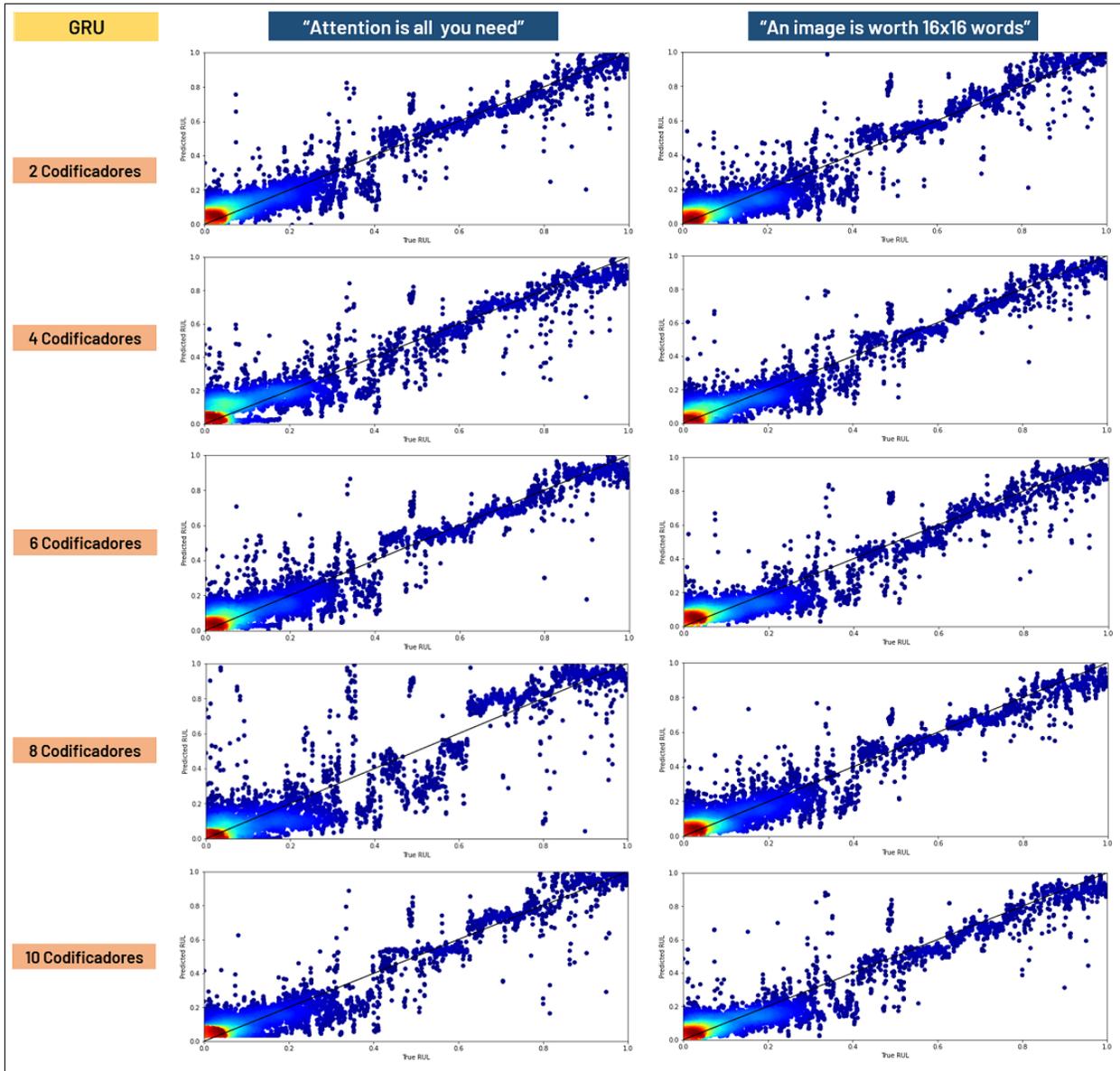
*Figura 65: Comparación de gráfico de dispersión de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo transformador*

**Diagrama de dispersión de modelos alternativos de tipo LSTM**



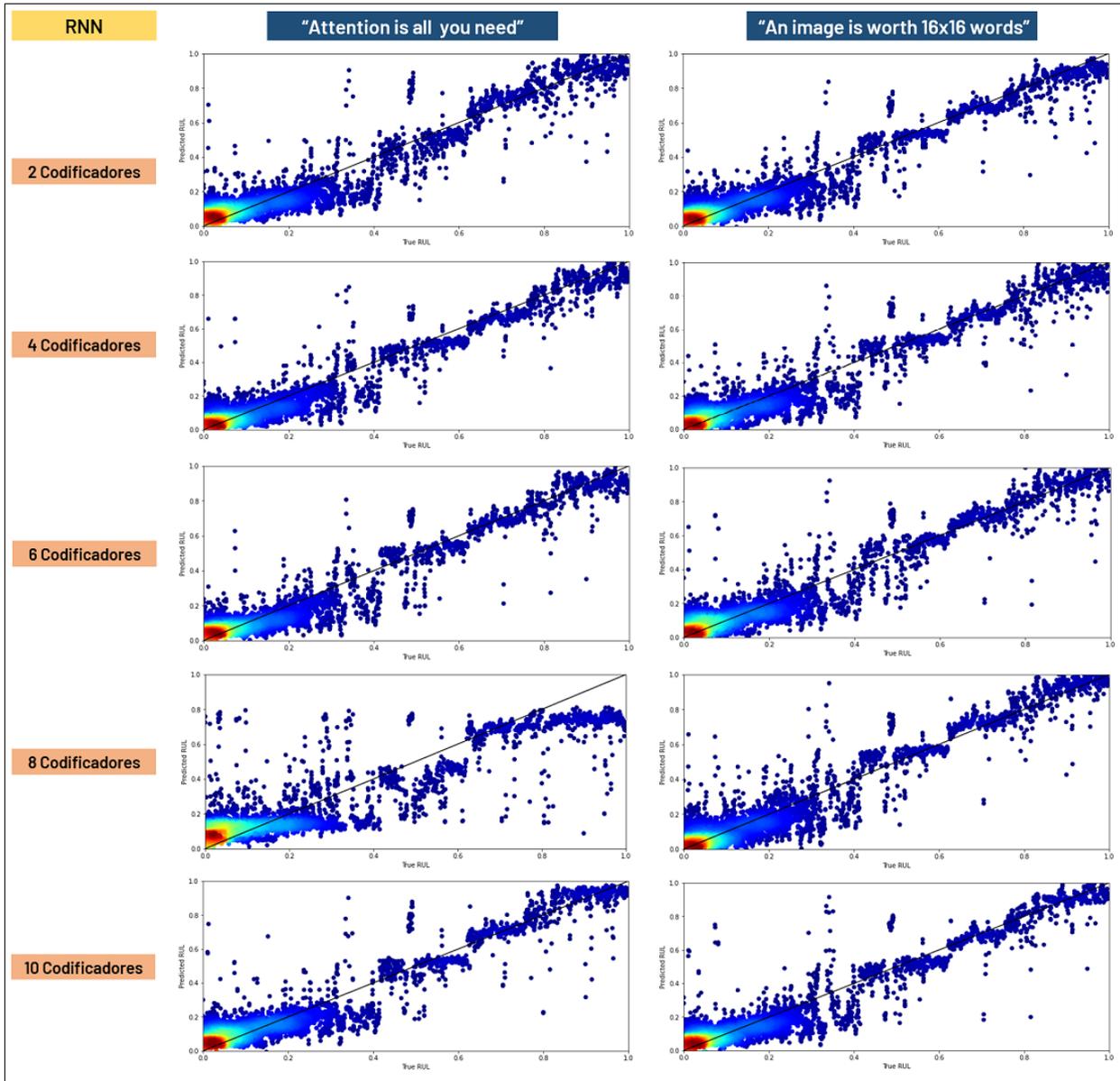
*Figura 66: Comparación de gráfico de dispersión de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo LSTM.*

**Diagrama de dispersión de modelos alternativos de tipo GRU**



*Figura 67:* Comparación de gráfico de dispersión de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo GRU.

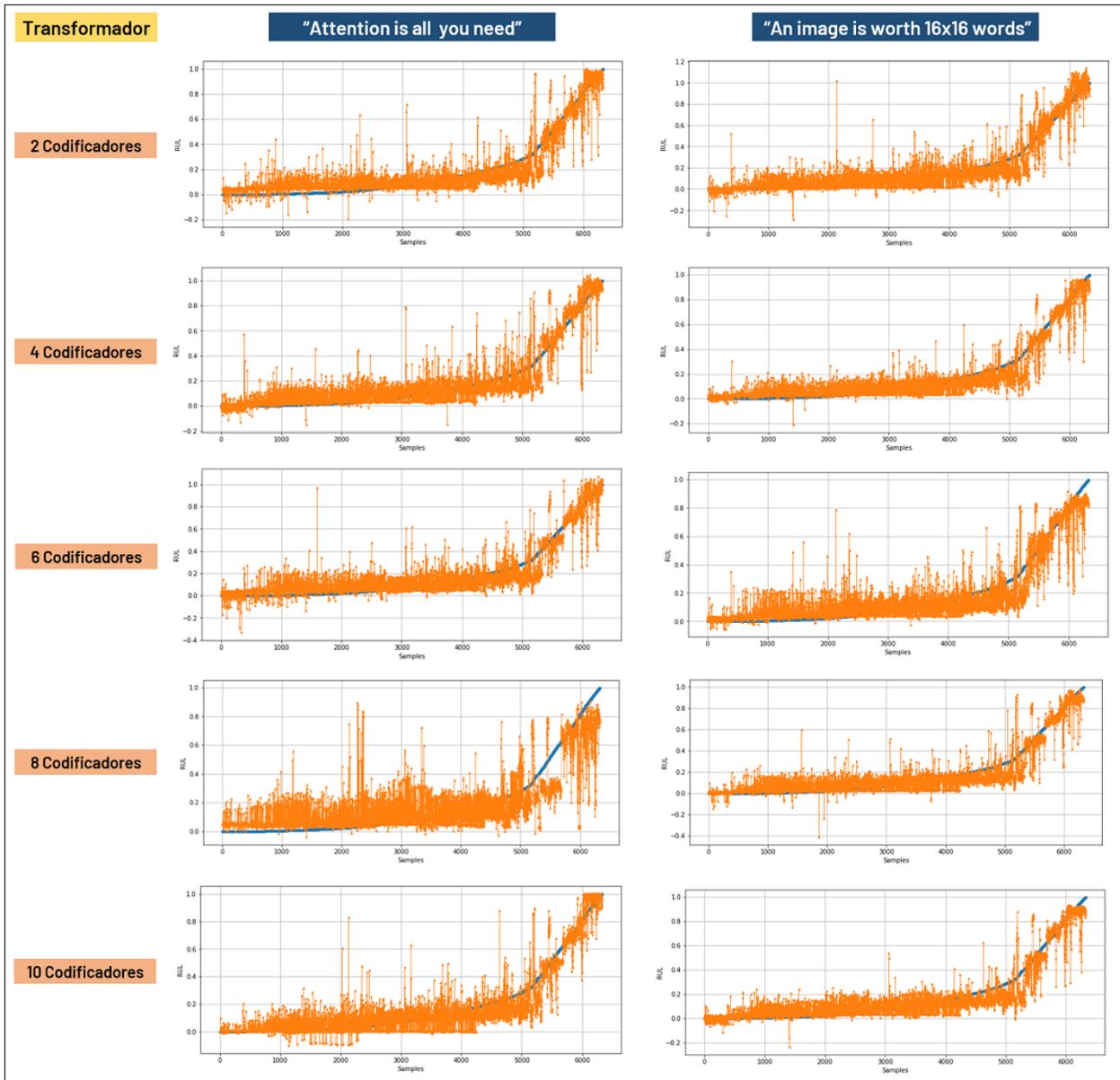
**Diagrama de dispersión de modelos alternativos de tipo RNN**



*Figura 68: Comparación de gráfico de dispersión de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo RNN.*

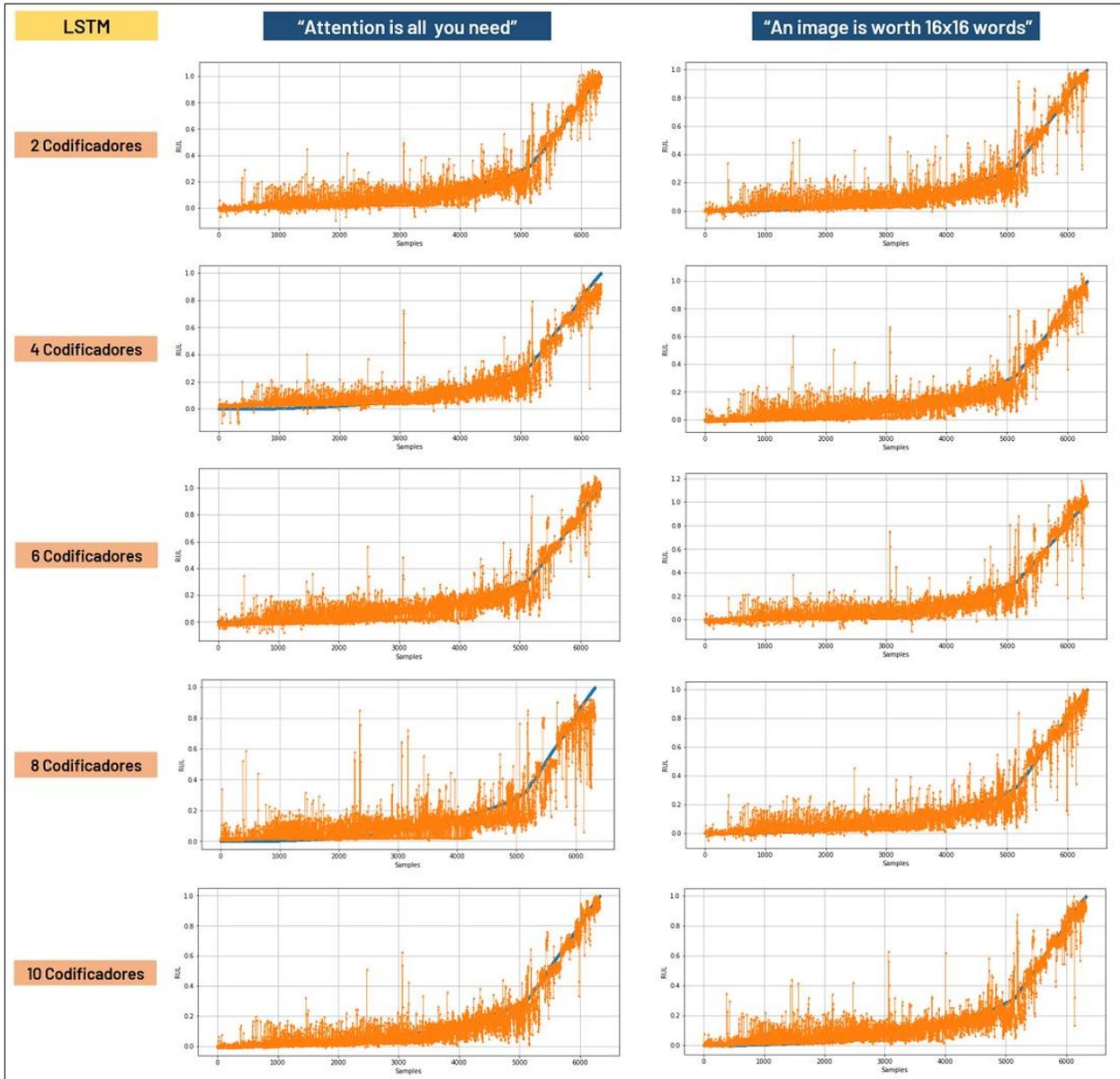
## Anexo I: Diagrama de dispersión ordenado de modelos alternativos

### Diagrama de dispersión ordenado de modelos alternativos de tipo Transformador



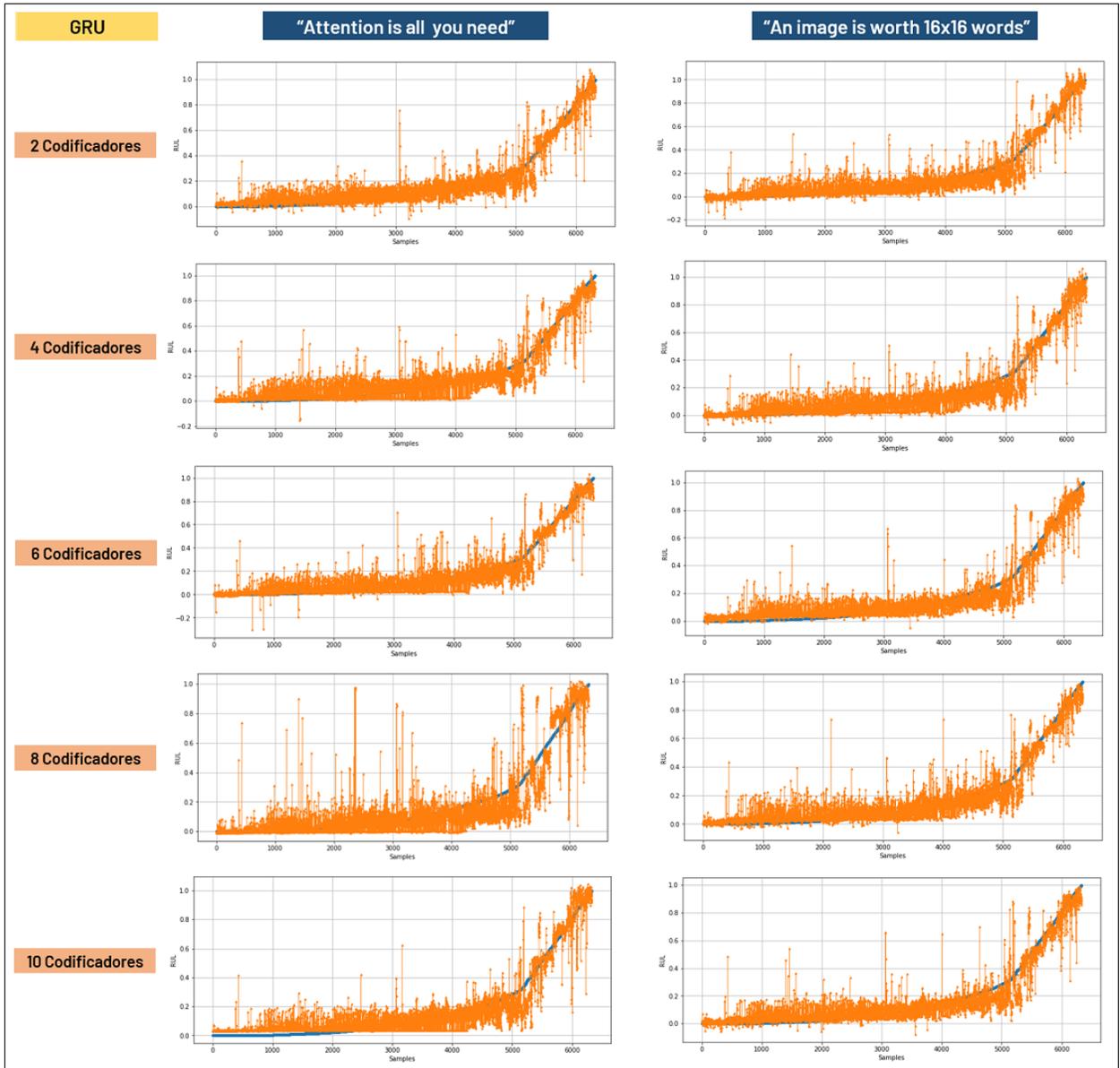
**Figura 69:** Comparación de gráfico de dispersión ordenado de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo transformador.

**Diagrama de dispersión ordenado de modelos alternativos de tipo LSTM**



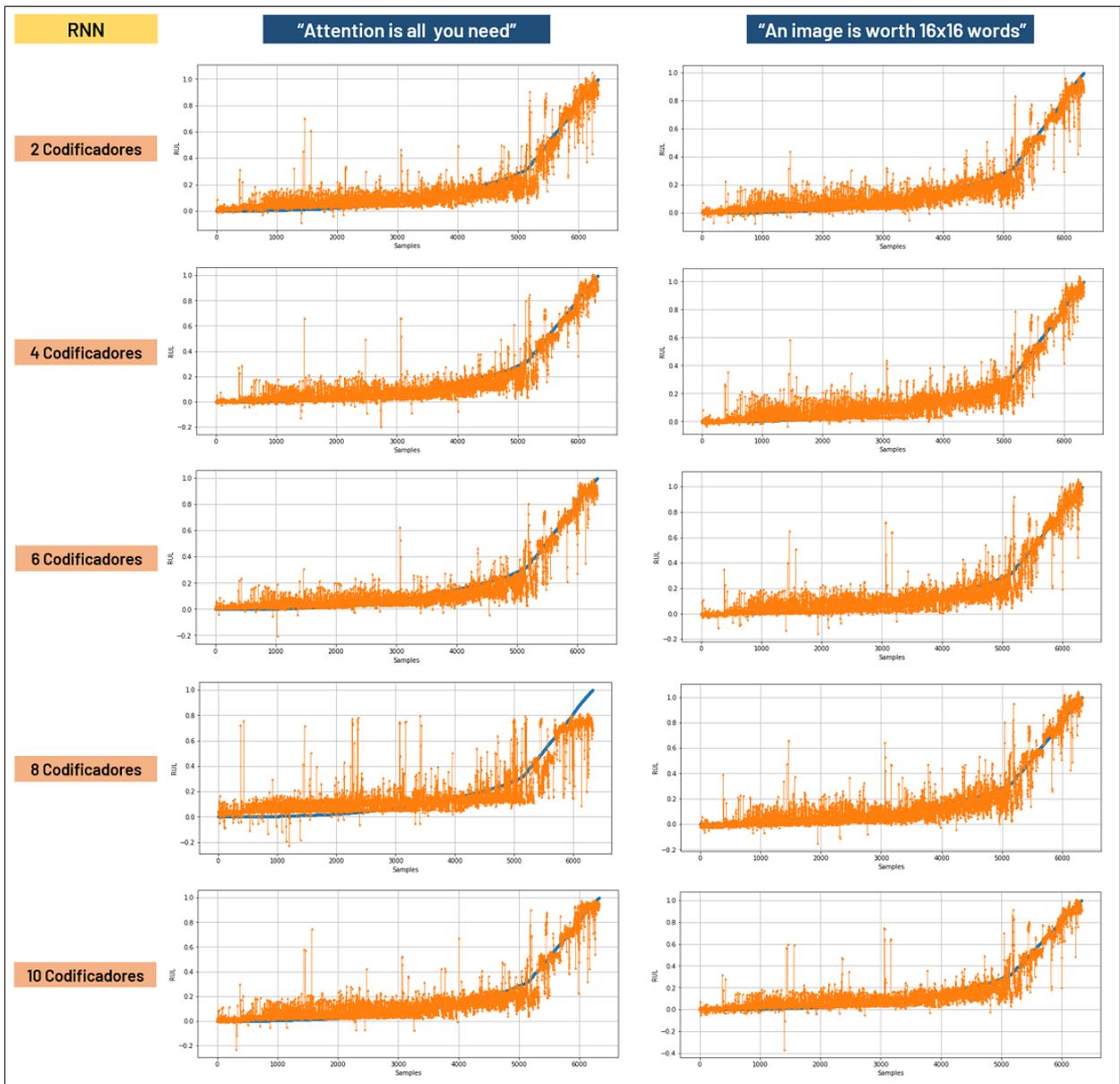
*Figura 70: Comparación de gráfico de dispersión ordenado de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo LSTM.*

**Diagrama de dispersión ordenado de modelos alternativos de tipo GRU**



*Figura 71:* Comparación de gráfico de dispersión ordenado de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo GRU.

**Diagrama de dispersión ordenado de modelos alternativos de tipo RNN**



*Figura 72: Comparación de gráfico de dispersión ordenado de valor predicho contra valor real de la vida útil remanente del equipo térmico de variantes del modelo RNN.*

## Anexo J: Registro de resultados de modelos alternativos en regresión

### Anexo J.1: Arquitectura “Attention is all you need”:

#### a) Modelo neuronal alternativo de dos codificadores

*Tabla 44: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por dos módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
Tiempo de entrenamiento [s]	Intento 1	3090	3336	3156	2855
	Intento 2	3236	3817	3722	3279
	Intento 3	3122	3698	3518	3279
	Intento 4	3243	3646	3459	3096
	Intento 5	3150	3708	3517	2916
	Intento 6	3060	3819	3638	3519
	Intento 7	3210	3537	3216	3340
	Intento 8	2967	3819	3632	3009
	Intento 9	2787	3638	3457	3100
	Intento 10	2765	3938	3398	3038
RMSE [.]	Intento 1	0,09	0,07	0,07	0,08
	Intento 2	0,09	0,07	0,07	0,07
	Intento 3	0,09	0,07	0,07	0,07
	Intento 4	0,09	0,07	0,08	0,08
	Intento 5	0,09	0,07	0,07	0,07
	Intento 6	0,09	0,07	0,07	0,08
	Intento 7	0,08	0,07	0,07	0,08
	Intento 8	0,09	0,07	0,08	0,07
	Intento 9	0,08	0,07	0,07	0,08
	Intento 10	0,09	0,07	0,07	0,07
R2 [.]	Intento 1	0,86	0,93	0,91	0,90
	Intento 2	0,85	0,92	0,92	0,93
	Intento 3	0,88	0,93	0,92	0,91
	Intento 4	0,86	0,91	0,89	0,91
	Intento 5	0,87	0,91	0,92	0,91
	Intento 6	0,88	0,91	0,91	0,92
	Intento 7	0,89	0,91	0,92	0,91
	Intento 8	0,86	0,93	0,90	0,92
	Intento 9	0,88	0,90	0,91	0,89

---

	Intento 10	0,89	0,90	0,93	0,91
<b>MAE</b> [.]	Intento 1	0,06	0,04	0,05	0,05
	Intento 2	0,06	0,05	0,05	0,05
	Intento 3	0,06	0,04	0,05	0,05
	Intento 4	0,06	0,05	0,06	0,05
	Intento 5	0,06	0,05	0,05	0,05
	Intento 6	0,06	0,05	0,05	0,05
	Intento 7	0,06	0,05	0,05	0,05
	Intento 8	0,06	0,05	0,05	0,04
	Intento 9	0,06	0,05	0,05	0,05
	Intento 10	0,06	0,05	0,05	0,05

## b) Modelo neuronal alternativo de cuatro codificadores

*Tabla 45: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por cuatro módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
<b>Tiempo de entrenamiento</b> [s]	Intento 1	4832	5874	5391	4972
	Intento 2	4837	5178	5406	4854
	Intento 3	4712	5763	5035	4916
	Intento 4	4230	5148	5511	4735
	Intento 5	4832	5327	5993	5215
	Intento 6	4712	5754	4736	5160
	Intento 7	4728	5213	3643	5177
	Intento 8	4171	6295	5274	4653
	Intento 9	4352	5628	5455	4916
	Intento 10	4772	6388	4969	4308
<b>RMSE</b> [.]	Intento 1	0,10	0,07	0,08	0,07
	Intento 2	0,09	0,07	0,08	0,07
	Intento 3	0,09	0,07	0,08	0,07
	Intento 4	0,09	0,08	0,07	0,07
	Intento 5	0,09	0,07	0,07	0,07
	Intento 6	0,09	0,06	0,08	0,08
	Intento 7	0,09	0,07	0,08	0,08
	Intento 8	0,094	0,07	0,08	0,07
	Intento 9	0,10	0,08	0,08	0,07
	Intento 10	0,10	0,07	0,07	0,07
<b>R2</b> [.]	Intento 1	0,87	0,91	0,91	0,92
	Intento 2	0,85	0,91	0,89	0,93
	Intento 3	0,86	0,94	0,89	0,93
	Intento 4	0,86	0,91	0,93	0,90
	Intento 5	0,85	0,92	0,91	0,93
	Intento 6	0,85	0,94	0,88	0,88
	Intento 7	0,86	0,92	0,87	0,91
	Intento 8	0,81	0,91	0,90	0,91
	Intento 9	0,85	0,90	0,91	0,92
	Intento 10	0,84	0,92	0,92	0,93
<b>MAE</b> [.]	Intento 1	0,07	0,05	0,05	0,05
	Intento 2	0,06	0,05	0,05	0,05
	Intento 3	0,07	0,04	0,06	0,04
	Intento 4	0,06	0,05	0,05	0,05

## Anexos

---

	Intento 5	0,07	0,04	0,05	0,05
	Intento 6	0,06	0,04	0,06	0,06
	Intento 7	0,06	0,05	0,06	0,05
	Intento 8	0,07	0,05	0,05	0,05
	Intento 9	0,07	0,05	0,05	0,05
	Intento 10	0,07	0,05	0,05	0,04

c) **Modelo neuronal alternativo de seis codificadores**

*Tabla 46: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por seis módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
Tiempo de entrenamiento [s]	Intento 1	4533	7985	7262	6363
	Intento 2	6159	7743	6484	6122
	Intento 3	5675	7716	6359	6546
	Intento 4	5975	7862	7447	5825
	Intento 5	6244	7261	7259	6122
	Intento 6	6651	7019	7080	4500
	Intento 7	5300	7075	6846	6789
	Intento 8	5795	7081	7387	6608
	Intento 9	6159	7080	7265	6730
	Intento 10	6208	7862	6544	6006
RMSE [.]	Intento 1	0,09	0,07	0,07	0,08
	Intento 2	0,10	0,07	0,07	0,06
	Intento 3	0,09	0,07	0,08	0,07
	Intento 4	0,10	0,07	0,07	0,08
	Intento 5	0,10	0,06	0,07	0,07
	Intento 6	0,09	0,07	0,07	0,07
	Intento 7	0,09	0,06	0,07	0,07
	Intento 8	0,09	0,07	0,08	0,06
	Intento 9	0,09	0,07	0,07	0,08
	Intento 10	0,11	0,07	0,07	0,07
R2 [.]	Intento 1	0,86	0,93	0,91	0,90
	Intento 2	0,81	0,93	0,92	0,94
	Intento 3	0,84	0,93	0,87	0,93
	Intento 4	0,83	0,92	0,91	0,92
	Intento 5	0,83	0,94	0,93	0,89
	Intento 6	0,87	0,92	0,91	0,91
	Intento 7	0,83	0,95	0,89	0,93
	Intento 8	0,84	0,92	0,90	0,93
	Intento 9	0,86	0,92	0,91	0,90
	Intento 10	0,77	0,92	0,92	0,93
MAE [.]	Intento 1	0,07	0,05	0,05	0,05
	Intento 2	0,07	0,05	0,05	0,04
	Intento 3	0,07	0,05	0,06	0,04
	Intento 4	0,07	0,05	0,05	0,05

## Anexos

---

	Intento 5	0,07	0,04	0,05	0,05
	Intento 6	0,06	0,05	0,05	0,05
	Intento 7	0,07	0,04	0,05	0,05
	Intento 8	0,07	0,05	0,05	0,04
	Intento 9	0,06	0,05	0,05	0,05
	Intento 10	0,08	0,05	0,05	0,04

### d) Modelo neuronal alternativo de ocho codificadores

*Tabla 47: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por ocho módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
Tiempo de entrenamiento [s]	Intento 1	8202	10116	8238	8461
	Intento 2	7916	9448	7782	7472
	Intento 3	8246	10361	9271	8474
	Intento 4	7783	9511	8612	8266
	Intento 5	7776	8963	8424	8024
	Intento 6	8020	10118	8066	8635
	Intento 7	7242	9925	9311	8019
	Intento 8	7362	9328	8670	7229
	Intento 9	7839	10721	10308	8267
	Intento 10	7843	10829	10416	8083
RMSE [%]	Intento 1	0,14	0,10	0,10	0,12
	Intento 2	0,14	0,10	0,10	0,11
	Intento 3	0,14	0,10	0,11	0,11
	Intento 4	0,14	0,10	0,13	0,11
	Intento 5	0,13	0,10	0,11	0,10
	Intento 6	0,15	0,09	0,10	0,12
	Intento 7	0,19	0,10	0,12	0,11
	Intento 8	0,15	0,10	0,15	0,11
	Intento 9	0,14	0,11	0,12	0,14
	Intento 10	0,16	0,10	0,10	0,12
R2 [%]	Intento 1	0,51	0,83	0,79	0,75
	Intento 2	0,57	0,81	0,79	0,69
	Intento 3	0,61	0,85	0,82	0,80
	Intento 4	0,69	0,82	0,69	0,69
	Intento 5	0,57	0,86	0,74	0,79
	Intento 6	0,55	0,83	0,86	0,72
	Intento 7	0,46	0,83	0,69	0,81
	Intento 8	0,42	0,85	0,53	0,77
	Intento 9	0,51	0,79	0,69	0,67
	Intento 10	0,49	0,84	0,81	0,71
MAE [%]	Intento 1	0,09	0,06	0,07	0,08
	Intento 2	0,09	0,07	0,07	0,08
	Intento 3	0,10	0,06	0,07	0,07
	Intento 4	0,10	0,06	0,08	0,09

## Anexos

---

	Intento 5	0,09	0,06	0,07	0,07
	Intento 6	0,10	0,06	0,06	0,08
	Intento 7	0,13	0,06	0,08	0,07
	Intento 8	0,11	0,06	0,09	0,07
	Intento 9	0,10	0,07	0,08	0,09
	Intento 10	0,10	0,07	0,07	0,09

### e) Modelo neuronal alternativo de diez codificadores

*Tabla 48: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “Attention is all you need” conformado por diez módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
<b>Tiempo de entrenamiento [s]</b>	Intento 1	8984	10954	10869	9100
	Intento 2	8684	11646	11133	9706
	Intento 3	9043	11938	11136	9653
	Intento 4	8203	11658	11315	9502
	Intento 5	8823	11707	11311	9282
	Intento 6	8443	11154	10646	9827
	Intento 7	8983	12149	10837	6737
	Intento 8	8944	12639	9867	9700
	Intento 9	8864	11558	9381	9941
	Intento 10	8742	12147	11115	9696
<b>RMSE [.]</b>	Intento 1	0,08	0,06	0,08	0,09
	Intento 2	0,08	0,12	0,10	0,09
	Intento 3	0,10	0,07	0,07	0,08
	Intento 4	0,12	0,07	0,09	0,07
	Intento 5	0,11	0,08	0,07	0,09
	Intento 6	0,09	0,06	0,07	0,09
	Intento 7	0,11	0,06	0,12	0,08
	Intento 8	0,14	0,07	0,07	0,07
	Intento 9	0,10	0,08	0,08	0,10
	Intento 10	0,11	0,06	0,08	0,08
<b>R2 [.]</b>	Intento 1	0,89	0,94	0,91	0,88
	Intento 2	0,89	0,77	0,81	0,88
	Intento 3	0,85	0,92	0,92	0,90
	Intento 4	0,72	0,93	0,87	0,92
	Intento 5	0,79	0,90	0,90	0,88
	Intento 6	0,83	0,93	0,92	0,87
	Intento 7	0,80	0,94	0,79	0,90
	Intento 8	0,70	0,91	0,93	0,93
	Intento 9	0,83	0,91	0,89	0,83
	Intento 10	0,78	0,94	0,89	0,90
<b>MAE [.]</b>	Intento 1	0,06	0,04	0,05	0,06
	Intento 2	0,06	0,08	0,07	0,06
	Intento 3	0,07	0,05	0,05	0,05
	Intento 4	0,08	0,05	0,06	0,05

## Anexos

---

	Intento 5	0,07	0,05	0,05	0,06
	Intento 6	0,07	0,04	0,05	0,06
	Intento 7	0,08	0,04	0,08	0,05
	Intento 8	0,09	0,05	0,05	0,05
	Intento 9	0,07	0,05	0,06	0,07
	Intento 10	0,07	0,04	0,06	0,05

## Anexo J.2: Arquitectura “An image is worth 16x16 words”

### a) Modelo neuronal alternativo de dos codificadores

*Tabla 49: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por dos módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
<b>Tiempo de entrenamiento</b> [s]	Intento 1	2802	3337	3035	3399
	Intento 2	3090	3336	3398	3338
	Intento 3	3210	3336	3396	2963
	Intento 4	3328	3700	2341	3277
	Intento 5	3150	3276	3519	3339
	Intento 6	3269	3638	3156	3520
	Intento 7	3570	3335	3486	3398
	Intento 8	2847	3577	3256	3338
	Intento 9	3390	3718	3578	3219
	Intento 10	3148	3638	3218	3277
<b>RMSE</b> [.]	Intento 1	0,09	0,07	0,08	0,07
	Intento 2	0,10	0,08	0,08	0,08
	Intento 3	0,09	0,08	0,08	0,08
	Intento 4	0,10	0,07	0,08	0,08
	Intento 5	0,09	0,07	0,08	0,08
	Intento 6	0,08	0,07	0,07	0,07
	Intento 7	0,09	0,08	0,08	0,07
	Intento 8	0,09	0,07	0,08	0,07
	Intento 9	0,08	0,07	0,08	0,07
	Intento 10	0,10	0,08	0,07	0,07
<b>R2</b> [.]	Intento 1	0,89	0,92	0,91	0,91
	Intento 2	0,86	0,91	0,90	0,89
	Intento 3	0,85	0,90	0,89	0,89
	Intento 4	0,84	0,90	0,88	0,89
	Intento 5	0,89	0,91	0,88	0,91
	Intento 6	0,88	0,91	0,91	0,925
	Intento 7	0,88	0,90	0,89	0,90
	Intento 8	0,87	0,91	0,91	0,90
	Intento 9	0,88	0,90	0,88	0,91
	Intento 10	0,85	0,91	0,92	0,91
<b>MAE</b> [.]	Intento 1	0,06	0,05	0,05	0,05
	Intento 2	0,07	0,05	0,05	0,05
	Intento 3	0,06	0,05	0,05	0,05

## Anexos

---

	Intento 4	0,07	0,05	0,06	0,05
	Intento 5	0,06	0,05	0,06	0,05
	Intento 6	0,06	0,05	0,05	0,04
	Intento 7	0,06	0,05	0,05	0,05
	Intento 8	0,06	0,05	0,05	0,05
	Intento 9	0,06	0,05	0,06	0,05
	Intento 10	0,07	0,05	0,05	0,05

**b) Modelo neuronal alternativo de cuatro codificadores**

*Tabla 50: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por cuatro módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
<b>Tiempo de entrenamiento</b> [s]	Intento 1	4712	5872	3815	4794
	Intento 2	4352	5993	5518	4374
	Intento 3	4769	5694	5330	4309
	Intento 4	4699	5148	5537	5098
	Intento 5	5138	5814	5510	4555
	Intento 6	4832	5814	4970	5323
	Intento 7	3569	6357	5260	4328
	Intento 8	5313	5749	4848	4976
	Intento 9	4643	5847	5209	4613
	Intento 10	496	5273	5027	4855
<b>RMSE</b> [.]	Intento 1	0,12	0,07	0,08	0,07
	Intento 2	0,08	0,06	0,07	0,07
	Intento 3	0,09	0,07	0,07	0,07
	Intento 4	0,09	0,07	0,08	0,07
	Intento 5	0,08	0,06	0,07	0,08
	Intento 6	0,08	0,07	0,08	0,07
	Intento 7	0,08	0,07	0,08	0,08
	Intento 8	0,10	0,07	0,08	0,07
	Intento 9	0,09	0,07	0,08	0,07
	Intento 10	0,09	0,07	0,08	0,07
<b>R2</b> [.]	Intento 1	0,72	0,91	0,90	0,93
	Intento 2	0,88	0,93	0,90	0,92
	Intento 3	0,86	0,93	0,91	0,92
	Intento 4	0,86	0,93	0,90	0,91
	Intento 5	0,89	0,93	0,90	0,90
	Intento 6	0,89	0,92	0,89	0,91
	Intento 7	0,89	0,91	0,91	0,90
	Intento 8	0,84	0,90	0,90	0,90
	Intento 9	0,86	0,91	0,89	0,91
	Intento 10	0,87	0,93	0,89	0,93
<b>MAE</b> [.]	Intento 1	0,08	0,05	0,05	0,05
	Intento 2	0,06	0,04	0,05	0,05
	Intento 3	0,06	0,04	0,05	0,04

## Anexos

---

	Intento 4	0,06	0,05	0,05	0,05
	Intento 5	0,06	0,04	0,05	0,05
	Intento 6	0,06	0,05	0,05	0,05
	Intento 7	0,06	0,05	0,05	0,05
	Intento 8	0,07	0,05	0,05	0,05
	Intento 9	0,06	0,05	0,05	0,05
	Intento 10	0,06	0,04	0,06	0,05

### c) Modelo neuronal alternativo de seis codificadores

*Tabla 51: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por seis módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
<b>Tiempo de entrenamiento</b> [s]	Intento 1	6303	7441	6846	5885
	Intento 2	5799	7804	7085	6548
	Intento 3	6310	8041	7144	6785
	Intento 4	6219	8164	6311	6546
	Intento 5	5614	8041	7087	6424
	Intento 6	5679	8045	7289	6426
	Intento 7	5915	7743	7146	6846
	Intento 8	6459	7326	7747	6847
	Intento 9	6279	7385	7390	4602
	Intento 10	5619	8765	7024	6547
<b>RMSE</b> [.]	Intento 1	0,10	0,07	0,08	0,07
	Intento 2	0,09	0,08	0,07	0,07
	Intento 3	0,09	0,07	0,07	0,08
	Intento 4	0,10	0,07	0,08	0,07
	Intento 5	0,09	0,07	0,07	0,08
	Intento 6	0,08	0,07	0,07	0,07
	Intento 7	0,08	0,07	0,08	0,07
	Intento 8	0,10	0,07	0,07	0,07
	Intento 9	0,09	0,07	0,07	0,08
	Intento 10	0,09	0,07	0,08	0,07
<b>R2</b> [.]	Intento 1	0,83	0,93	0,90	0,91
	Intento 2	0,89	0,92	0,92	0,90
	Intento 3	0,86	0,91	0,91	0,90
	Intento 4	0,77	0,93	0,91	0,93
	Intento 5	0,87	0,91	0,92	0,90
	Intento 6	0,88	0,92	0,92	0,91
	Intento 7	0,89	0,93	0,88	0,93
	Intento 8	0,83	0,93	0,93	0,91
	Intento 9	0,86	0,92	0,92	0,05
	Intento 10	0,87	0,93	0,91	0,92
<b>MAE</b> [.]	Intento 1	0,07	0,05	0,05	0,05
	Intento 2	0,06	0,05	0,05	0,05
	Intento 3	0,06	0,05	0,05	0,05
	Intento 4	0,07	0,04	0,05	0,05
	Intento 5	0,06	0,05	0,05	0,05

## Anexos

---

	Intento 6	0,06	0,05	0,05	0,05
	Intento 7	0,06	0,05	0,05	0,04
	Intento 8	0,07	0,04	0,05	0,05
	Intento 9	0,07	0,05	0,05	0,05
	Intento 10	0,06	0,05	0,05	0,05

### d) Modelo neuronal alternativo de ocho codificadores

*Tabla 52: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por ocho módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
<b>Tiempo de entrenamiento [s]</b>	Intento 1	8733	9921	7933	7337
	Intento 2	7121	8952	9871	8008
	Intento 3	6820	11138	8051	7337
	Intento 4	7662	10048	8172	8307
	Intento 5	8143	10770	8472	7216
	Intento 6	7658	9015	8773	7093
	Intento 7	7422	8987	8113	7583
	Intento 8	8202	8092	8294	6487
	Intento 9	5013	9930	8415	7878
	Intento 10	7541	9915	9618	8068
<b>RMSE [.]</b>	Intento 1	0,09	0,07	0,07	0,07
	Intento 2	0,09	0,07	0,08	0,07
	Intento 3	0,09	0,07	0,07	0,08
	Intento 4	0,08	0,07	0,08	0,07
	Intento 5	0,10	0,07	0,08	0,08
	Intento 6	0,10	0,07	0,07	0,08
	Intento 7	0,09	0,08	0,07	0,07
	Intento 8	0,09	0,07	0,07	0,08
	Intento 9	0,10	0,07	0,07	0,07
	Intento 10	0,09	0,07	0,07	0,07
<b>R2 [.]</b>	Intento 1	0,87	0,92	0,91	0,92
	Intento 2	0,84	0,91	0,90	0,92
	Intento 3	0,87	0,91	0,92	0,89
	Intento 4	0,87	0,91	0,91	0,92
	Intento 5	0,82	0,93	0,90	0,90
	Intento 6	0,85	0,90	0,92	0,91
	Intento 7	0,86	0,899	0,91	0,92
	Intento 8	0,83	0,92	0,93	0,91
	Intento 9	0,84	0,93	0,91	0,91
	Intento 10	0,86	0,91	0,92	0,92
<b>MAE [.]</b>	Intento 1	0,06	0,05	0,05	0,05
	Intento 2	0,06	0,05	0,06	0,05
	Intento 3	0,06	0,05	0,05	0,05
	Intento 4	0,06	0,05	0,05	0,05

## Anexos

---

	Intento 5	0,07	0,04	0,05	0,05
	Intento 6	0,06	0,05	0,05	0,05
	Intento 7	0,06	0,05	0,05	0,05
	Intento 8	0,07	0,05	0,05	0,05
	Intento 9	0,07	0,05	0,05	0,05
	Intento 10	0,06	0,05	0,05	0,05

### e) Modelo neuronal alternativo de diez codificadores

*Tabla 53: Datos de desempeño logrado en regresión de cada modelo por iteración de aprendizaje automático bajo la arquitectura “An image is worth 16x16 words” conformado por diez módulos de codificadores.*

Indicador de rendimiento	Tipo de red neuronal	Transformer	LSTM	GRU	RNN
<b>Tiempo de entrenamiento</b> [s]	Intento 1	9163	11075	9623	9880
	Intento 2	8443	11238	9441	9100
	Intento 3	9344	11554	11491	8862
	Intento 4	9824	11020	10864	8846
	Intento 5	8341	12209	11141	10124
	Intento 6	9343	12523	10045	10296
	Intento 7	9944	12332	9351	9579
	Intento 8	9396	11318	10956	9822
	Intento 9	8383	11853	11561	9691
	Intento 10	8563	10688	7395	9640
<b>RMSE</b> [.]	Intento 1	0,09	0,07	0,07	0,07
	Intento 2	0,09	0,07	0,07	0,07
	Intento 3	0,09	0,07	0,08	0,07
	Intento 4	0,09	0,07	0,08	0,08
	Intento 5	0,08	0,07	0,07	0,07
	Intento 6	0,09	0,07	0,07	0,07
	Intento 7	0,13	0,07	0,07	0,07
	Intento 8	0,09	0,07	0,08	0,07
	Intento 9	0,09	0,07	0,07	0,07
	Intento 10	0,08	0,07	0,08	0,07
<b>R2</b> [.]	Intento 1	0,87	0,91	0,91	0,91
	Intento 2	0,86	0,92	0,92	0,91
	Intento 3	0,84	0,91	0,90	0,91
	Intento 4	0,88	0,91	0,90	0,90
	Intento 5	0,88	0,91	0,92	0,92
	Intento 6	0,86	0,92	0,90	0,92
	Intento 7	0,66	0,91	0,91	0,91
	Intento 8	0,88	0,93	0,90	0,91
	Intento 9	0,89	0,91	0,92	0,92
	Intento 10	0,89	0,91	0,89	0,91
<b>MAE</b> [.]	Intento 1	0,06	0,05	0,05	0,05
	Intento 2	0,06	0,05	0,05	0,05
	Intento 3	0,06	0,06	0,06	0,05
	Intento 4	0,06	0,05	0,05	0,05

## Anexos

---

	Intento 5	0,06	0,05	0,05	0,05
	Intento 6	0,06	0,05	0,05	0,05
	Intento 7	0,09	0,05	0,05	0,05
	Intento 8	0,06	0,05	0,05	0,05
	Intento 9	0,06	0,05	0,05	0,05
	Intento 10	0,06	0,05	0,05	0,05